Intelligent System Design, Operation and Management to Govern on Electrical Energy to the Consumer

By

Loau Tawfak Abd Ali Al-Bahrani

A thesis submitted in fulfilment of the of the requirements for the degree of Doctor of Philosophy

School of Software and Electrical Engineering
Faculty Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia

July 2018

Abstract

The economic dispatch (ED) of generation is one of the most critical optimization problems in operation and management of power systems. The current Iraq power generating system (PGS) is suffering from several concurrent challenges. One of these challenges is ED of the power of large-scale thermal generating units (TGUs) with various power constraints, which made it non-economic.

The ED allows PGS analysts to schedule the committed online TGUs so as to meet the required load demand at a minimum operating cost while satisfying all online TGU and PGS inequality and equality constraints, such as ramp-rate limits, prohibited operating zones and valve-point loading effects. Practically, with these constraints, the fuel cost function of ED problem becomes multimodal and non-convex with highly non-linear characteristics.

Several evolutionary computation techniques (ECTs) sought to address such a complex problem. One popular type of ECT is the global particle swarm optimization (GPSO) algorithm, however, it is not capable of solving such a complex problem satisfactory.

This thesis introduces eight papers to address and solve the complex ED problem by proposing two novel algorithms called orthogonal PSO (OPSO) and multi-gradient PSO (MG-PSO) algorithms. In the OPSO algorithm, the *m* particles in a swarm are divided into two groups. The first group is an active group of best personal experience of *d* particles. The second group is a passive group of personal experience of remaining (*m* – *d*) particles. The target of creating two groups is to boost the diversity of the swarm's population. The *d* active group particles in each iteration undergo an orthogonal diagonalization process and are updated in such way that their position vectors are orthogonally diagonalized. Whereas, the passive group particles are not updated, as their contribution in obtaining correct direction is not significant. The particles in OPSO algorithm are updated using only one guide, thus avoiding the conflict between the two guides that occurs in the GPSO algorithm.

In the MG-PSO algorithm, multiple negative gradients are used in two phases (*Exploration* phase and *Exploitation* phase) by the *m* particles while searching for an optimum solution. In *Exploration* phase, a particle is called an *Explorer*. The *m Explorers* operate in several episodes. In each episode, the *m Explorers* use a different negative gradient to explore a new neighbourhood. The *m Explorers* enhance global search ability of the MG-PSO algorithm. In the *Exploitation* phase, a particle is called an *Exploiter*. The *m Exploiters* use only one negative gradient which is less than that of the *Exploration* phase to exploit the best neighbourhood. Thus, the *m Exploiters* enhance local search ability of the MG-PSO algorithm. This diversity in negative gradients helps the best particle to avoid from falling into a local minimum. The combination of two phases supplies a balance between *Exploration* and *Exploitation* in a search space, thus avoiding the loss of balance between the two guides that occurs in the GPSO algorithm.

The effectiveness of the OPSO and MG-PSO algorithms are verified using small, medium and large PGSs with several power constraints as well as a set of unimodal and multimodal benchmark functions with dimensions of 30 and 100 taken from the Congress on evolutionary computation 2015 (CEC 2015). Superior performance of the proposed OPSO and MG-PSO algorithms over the GPSO algorithm and several existing optimization techniques with several performance measures are shown in Papers A-H. In addition, by using an unpaired t-test, the statistical significance of the proposed OPSO and MG-PSO algorithms has been shown against several contending algorithms including top-ranked CEC 2015 algorithms.

Acknowledgment

First of all, many thanks to "ALLAH" who has blessed and helped me to accomplish this thesis.

I would like to thank Ministry of higher education and scientific research in Iraq and University of Diyala for giving me the opportunity to pursue my PhD study at Swinburne University of Technology. I am thankful also to the Cultural Office, Embassy of Iraq in Canberra for their understanding, help and co-operation.

I would like to express my sincere gratitude to my supervisor, senior lecturer Jagdish Chandra Patra, for the continuous support during my PhD study and related research.

I would also like to thank the Swinburne University of Technology for providing me with this opportunity and experience that improved my academic research skills and made me explore new directions in my experience.

It is my pleasure to thank the staff in the IT services and library for their instant support whenever I need.

I would like to thank all my colleagues and friends in Australia and Iraq who have participated with me in the discussion, support, encouragement during my PhD study.

Last but not the least, a huge thanks to my fantastic wife who followed me in this study in Australia. She has been supporting me spiritually throughout the research and writing this thesis and in my life, Abeer, in general. I will never be grateful enough gratitude for what she did. And thank you to my daughter, Narjas, my son, Mohammed and my father-in-law, Nidhal D. Salman, for assisting me when I need.

Declaration

“I, Loau Tawfak Abd Ali Al-Bahrani, declare that this PhD thesis including associated papers entitled “Intelligent system design, operation and management to govern on electrical energy to the consumer” is my work. It contains no material, which has been accepted for the award of any other degree or diploma, except where due reference is made in the text of the examinable outcome. To the best of my knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the examinable outcome.

9 July 2018
Loau Tawfak Abd Ali Al-Bahrani
School of Software and Electrical Engineering
Faculty of Science, Engineering and Technology
Swinburne University of Technology

Supervisor Certification

This is to certify that the above statement made by the candidate, Loau Tawfak Abd Ali Al-Bahrani, is correct to the best of my knowledge.

JcPatra
9 July 2018
Dr. Jagdish Chandra Patra
Senior Lecturer
Department of Telecommunications, Electrical, Robotics and Biomedical Engineering
Faculty of Science, Engineering and Technology
Swinburne University of Technology

Copyright Notices

Notice #1:

Notice #2:

I warrant that I have obtained, where necessary, permission from the copyright owners to use any third party copyright material reproduced in the thesis (such as artwork, images, unpublished documents), or to use of any own published work (such as journal articles) in which the copyright is held by another party (such as publisher, co-author).

Copies of the signed authorship indication form and publisher permission of each paper are appended in Appendices 2 and 3.

9 July 2018

Loau Tawfak Abd Ali Al-Bahrani
School of Software and Electrical Engineering
Faculty Science, Engineering and Technology
Swinburne University of Technology

Table of Contents

List of Figures

Figure # **Description** **Page #**

List of Tables

List of the Author's Publications (Published, Accepted for Publication and under review)

Ten international journal and top-tier peer-reviewed international conference Papers are produced during my PhD study. Eight Papers were published, and one Paper is accepted for publication. Another Paper was submitted to an international journal (under review). The author's publications, published, accepted for publication and under review, are given below.

1. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1-7.

2. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO Algorithm for Economic Dispatch of Power under Power Grid Constraints," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Hong Kong, 2015, pp. 14-19.

3. **L. T. Al-Bahrani**, J. C. Patra, and R. Kowalczyk, "Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, 2016, pp. 660-667.

4. **L. T. Al-Bahrani**, J. C. Patra, and R. Kowalczyk, "Multi-gradient PSO algorithm for economic dispatch of thermal generating units in smart grid," in *Proceedings of IEEE PES Innovative Smart Grid Technologies 2016 Asian Conference (ISGT 2016 Asia)*, Melbourne, Australia, 2016, pp. 258-263.

5. **L. T. Al-Bahrani** and J. C. Patra, "A novel orthogonal PSO algorithm based on orthogonal diagonalization," *Swarm and Evolutionary Computation*, vol. xxx, pp. 1-23, 2017. In press.

6. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid," *Applied Soft Computing,* vol. 58, pp. 401-426, 2017.

7. **L. T. Al-Bahrani** and J. C. Patra, "Multi-gradient PSO algorithm for solving non-convex cost function of thermal generating units under various power constraints in smart power grid," *in Proceedings of first MoHESR and HCED Iraqi Scholars Conference (ISCA2017),* Melbourne, Australia, 2017, pp. 258-270.

8. **L. T. Al-Bahrani** and J. C. Patra, "Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid," *Energy*, vol. 147, pp. 1070-1091, 2018.

9. **L. T. Al-Bahrani** and J. C. Patra, "Multi-gradient PSO algorithm with enhanced exploration and exploitation," *Applied Soft Computing* (2018). Under review.

10. **L. T. Al-Bahrani** and J. C. Patra, "Solving economic dispatch problem under valve-point loading effects and generation constrains using a multi-gradient PSO algorithm," *IEEE International Joint Conference on Neural Networks (IJCNN)*, Rio de Janerio, Brazil, 2018. Accepted for presentation.

List of Abbreviations

A list of the abbreviations used in this thesis is given below.

Description	Abbreviation
Ant colony optimization	ACO
Average execution time	AET
Bat algorithm	BA
Becterial foraging	BF
Bee colony optimization	BCO
Best fitness value	BFV
Carbon oxide	CO_2
Chaotic bat algorithm	CBA
Colonial competitive differential evolution	CCDE
Co-swarm shrinking hypersphere particle swarm optimization	CSHPSO
Cuckoo search algorithm	CSA
Differential evolution	DE
Differential evolution with greedy randomized adaptive search procedure	DE-GRASP
Differential evolution with particle swarm optimization	DE-PSO-DE
Differential evolution with greedy randomized adaptive search	DE-GRAS
Differential evolution with harmony search	DHS
Dynamic particle swarm optimization with escaping prey	DPSOEP
Economic dispatch	ED
Evolutionary computation techniques	ECTs
Evolutionary differential evolution	EDE
Feasible operating zones	FOZs
Flower pollination algorithm	FPA
Fully decentralized approach	FDA
Fuzzy based hybrid particle swarm optimization-differential evolution	FBHPSO-DE
Genetic algorithm	GA
Global particle swarm optimization	GPSO
Harmony search	HS
Hybrid genetic algorithm with Becterial foraging	HGBF
Hybrid immune with genetic algorithm	HIGA
Hybrid grey wolf optimizer	HGWO
Hybrid symbiotic organisms search	HSOS
Improved differential evolution	IDE

Improved particle swarm optimization	IPSO
Improved random drift particle swarm optimization	IRDPSO
Iteration particle swarm optimization with time varying acceleration coefficients	IPSO-TVAC
Linear programming	LP
Local particle swarm optimization	LPSO
Lighting flash algorithm	LFA
Multi-gradient particle swarm optimization	MG-PSO
Mixed-integer programming	MIP
Mean fitness value	MFV
Modified particle swarm optimization	MPSO
Mixed-integer quadratic programming	MIQP
National dispatch centre	NDC
Nitrogin oxides	NO_X
Orthogonal vector	OV
Orthogonal diagonalization	OD
Opposition-based greedy heuristic search	OGHS
Oppositional invasive weed algorithm	OIWA
Oppositional grey wolf algorithm	OGWA
Power generating system	PGS
Power generating system	PGU
Prohibited operating zones	POZs
Particle swarm optimization	PSO
Particle swarm optimization with gravitational search algorithm	PSO-GSA
Parallel hurricane optimization algorithm	PHOA
Parallel augmented Lagrangian relaxation	PALR
Quadratic programming	QP
Ramp-rate limits	RRLs
Smart power grid	SPG
Sequential quadratic programming	SQP
Sulfur doxide	SO_2
Shuffled differential evolution	SDE
Synergic predator prey optimization	SPPO
The Congress of evolutionary computation	CEC
Two-phase mixed integer programming	TPMIP
Thermal generating unit	TGU
Theta particle swarm optimization	θ-MBA
Tournament-based harmony search	THS
Valve-point loading	VPL
Worst fitness value	WFV

Chapter 1: Introduction

1.1 Chapter overview

In this Chapter, an introduction of this study is addressed as follows. The background is given in Section 1.2. The current issues in economic dispatch problem are presented in Section 1.3. In section 1.4, motivation and research questions are presented. Limitations of data availability are addressed in Section 1.5. List of the author's publications as part of thesis is presented in Section 1.6. Contribution of this thesis is presented in Section 1.7. In Section 1.8, outline of this thesis is presented. Finally, Chapter summary is presented in Section 1.9.

1.2 Background

I was sponsored by the government of Iraq to study the problems and issues in Iraq power generating system (PGS). The current Iraq PGS is mainly divided into five operating regions based on the operation and control, as shown in a single line diagram in Figure 1.1. The five operating regions namely are North, Diyala and Anbar, Baghdad, Middle and South regions [1]. Each operating region has several power stations, and each power station consists of a number of power generating units (PGUs). These five regions are operated, controlled and managed as a unified and interconnected PGS by the national dispatch centre (NDC). Besides, there is a local dispatch centre in each operating region associated directly with the NDC. The PGS is connected directly to a large power grid of 132 kV ultra-high voltage network through substations, and then the electrical power is transmitted by different types of the transmission networks to the consumer. The consumer in this thesis represents residential, commercial and industrial electrical loads.

Since Iraq has abundant fossil fuels (oil and natural gas reserves), the PGS largely depends on fossil fuels powered by a large-scale of thermal generating units (TGUs) to generate electricity. The statistical data issued by the NDC by the ministry of electricity has shown that total number of PGUs is 371-unit [2].

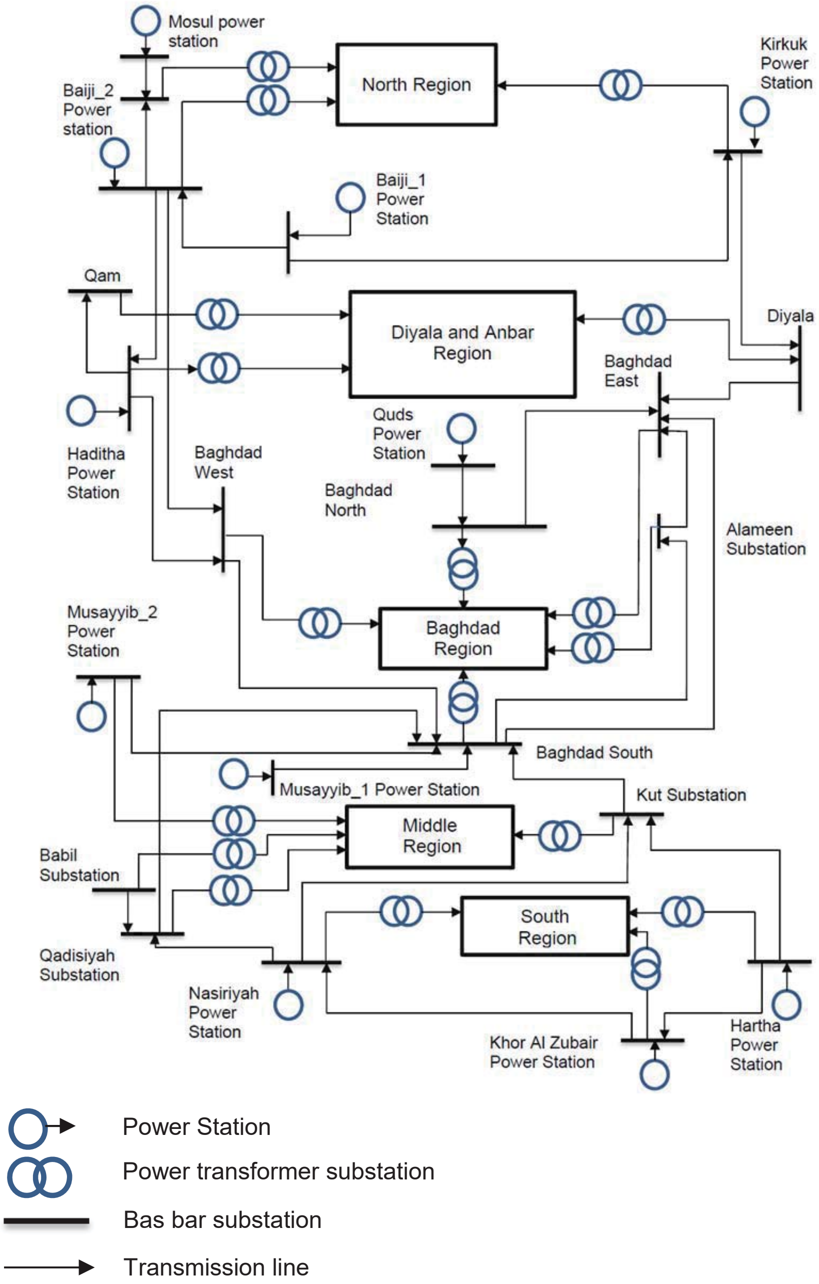


Figure 1.1. A single line diagram of Iraq power generating system.

These PGUs are classified as 342 TGUs (31 steam units, 194 gas units, 22 mobile diesel units and 95 fixed diesel units) and 29 hydropower generating units. Accordingly, 91.5% of the generated electricity comes from the TGUs.

Economic Dispatch (ED) of power (also termed as ED problem) is a fundamental tool in the PGS which plays a critical role in operation, planning and control of every power system. The primary purpose of ED investigation recognizes the optimum schedule of active output power of all committed PGUs to minimize the total fuel cost, while satisfying the power constraints imposed by PGSs [3]. Practically, the TGUs have several operating power constraints and limitations, e.g., power balance, transmission network loss (P_L), generation limits, ramp-rate limits (RRLs), prohibited operating zones (POZs), feasible operating zones (FOZs), and valve-point loading (VPL) effects [4].

The power balance constraint is an equality constraint and must be satisfied such that the total generated active output power equals to the sum of the load demand (P_D) and P_L. Furthermore, the online TGU uses RRLs that are an inequality power constraint which represent the rate at which the active output power level of a given TGU can be modified to satisfy the power balance. In such a case, the active output power cannot be adjusted instantaneously. Their corresponding RRLs restrict the operating range of all online TGUs.

Another inequality power constraint is POZs which stems from physical limitations of a TGU, e.g., the amplification of vibrations in a shaft bearing at specific operating regions. Because of the POZs, the TGU may not be able to work in specific operating zones. For instance, mechanical vibrations could cause cumulative metal fatigue in steam-turbine blades and lead to the premature turbine blade failure. The POZs create gaps in the fuel cost curve and thus introduce discontinuity in the fuel cost function [5]. Therefore, each TGU must be operated within the FOZs avoiding any existing POZ.

Another operating power constraint is VPL effects. These effects become prominent in the fuel cost function in the following case. Practically, a steam-turbine of the TGU has multiple valves that are used to control its active power outputs. When steam valves start to open and close simultaneously, this causes ripple-like effects. These ripples add to the fuel cost function. In such a case, the cost fuel function comprises non-linearity of a higher order [6]. The definitions of these power constraints are available in Papers A-H.

In this study, the ED of the power of online TGUs with various equality and inequality operating power constraints are addressed.

1.3 Current issues in economic dispatch problem

The current Iraq PGS is suffering from several concurrent challenges. One of these challenges is the ED of the power of large-scale TGUs with various practical operating power constraints.

A symposium was held in May 2013 in Baghdad under the sponsorship of ministry of higher education and scientific research and ministry of electricity, to evaluate the reality of the electricity sector. The team leader of the NDC emphasized as follows. "*Operation and management of large-scale TGUs to govern electrical energy to the consumer are big challenges for us. We are spending billions of dollars in each year to generate electricity. We need new algorithms for solving the ED problem that are effective and compatible with modern technologies. We see that the current computational techniques used by the NDC, e.g., traditional methods and some other optimization techniques, are inefficient. Therefore, the NDC today can no longer solely rely on the current traditional means of Iraqi power generating system planning*."

Besides the problem stated above, several other technical issues are summarized as follows.

1.3.1 Low operational effeciency

There exists a large gap between maximum actual generating power and installed generating capacity of the online TGUs. The Iraq PGS consists of 342 TGUs with total installed generating capacity (nameplate rating) equals to 24,646 MW. However, the maximum actual generating power by all TGUs is only about 8,746 MW, in the best cases [2].

Table 1.1 shows the operation of online TGUs during the period 2013-2016. One can see that despite an increase in maximum actual generating power from 2013 to 2016, the gap is still large. For example, in 2015, the total number of online TGUs was 265-TGU. The maximum actual generating power attained was about 7,551 MW against installed generating capacity of 20,903 MW. Here, the gap is 63.87%. However, in 2016, the maximum actual generating power has increased up to 8,746 MW with installed generating capacity of 20,087 MW. Here, the gap is 56.45%. It can be seen that the gap is reduced by 7.42% in 2016, but the gap still remains large.

Figure 1.2 gives more details about the gap between the installed generating capacity and maximum actual generating power of online TGUs of three types of the power stations, i.e., steam, gas and diesel power stations during the period 2013-2016. The gap must be within an acceptable range of 10-20% for such types of power stations. The main reason of this gap is estimating of the ED of active output power of large-scale TGUs is not optimum because of the current inefficient optimization techniques. This large gap causes low operational efficiency and thus, the PGS becomes non-economic.

Table 1.1. Installed generating capacity and maximum actual generating power of online TGUs during 2013-2016.

Year	Number of online TGUs (out of 342 TGUs)	Installed generating capacity (MW)	Maximum actual generating power (MW)	Gap (%)
2016	243	20,087	8,746	56.45
2015	265	20,903	7,551	63.87
2014	268	18,487	7,401	59.96
2013	277	17,272	6,126	64.53

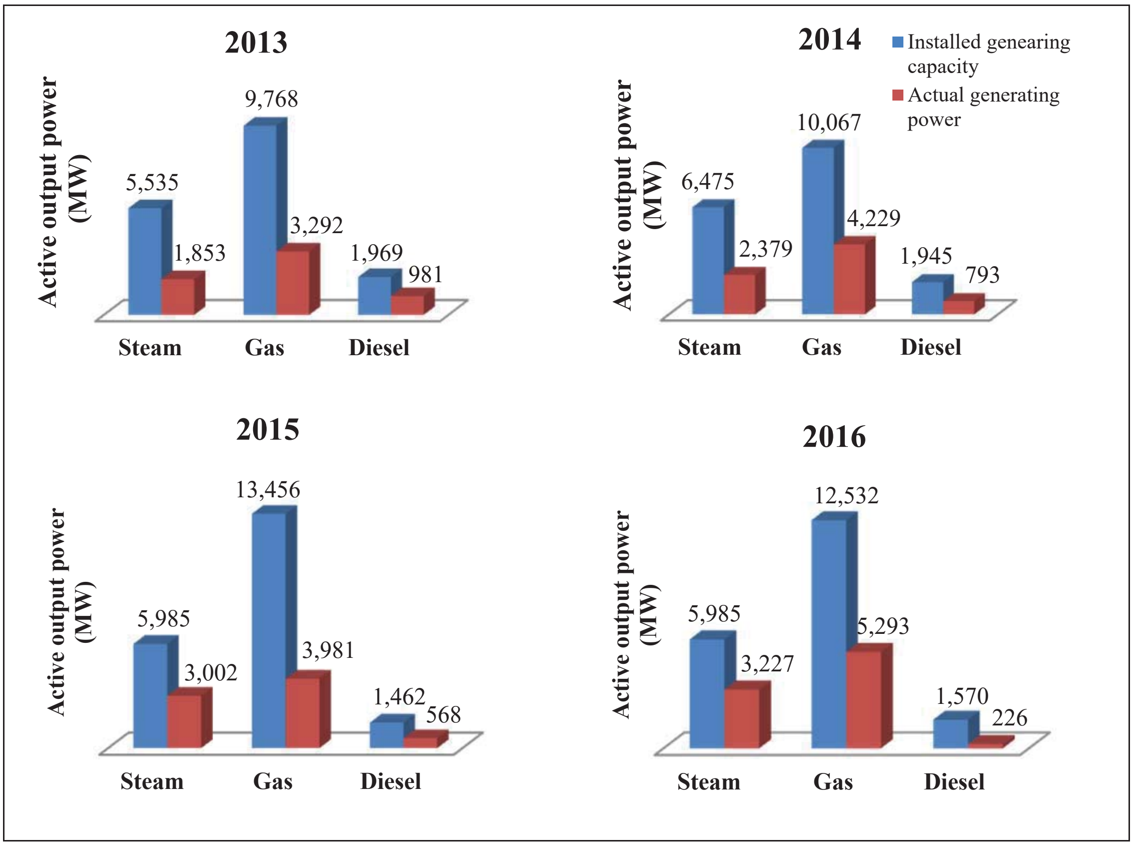


Figure 1.2. Installed generating capacity and maximum actual generating power of steam, gas and diesel power stations of Iraq PGS.

1.3.2 Unplanned increase of load demand

The load demand P_D has increased drastically due to a growing economy and a surge in consumer usages of the electrical appliances and electronic devices during the recent ten years. In addition, Iraqi government subsidizes the tariff of electricity. Despite the increase of supply electricity in the past two years (2015 and 2016), as shown in Figure 1.2, the current PGS is still unable to meet the P_D growth projections.

According to the statistical data of the NDC, the maximum generating power of the current PGS (TGUs, hydropower PGUs and the imported power from neighbouring countries) is at about 10,630 MW of the power currently required, i.e., maximum P_D =~ 17,000-18,000 MW, during Summer season [2]. This means that 37.47% to 40.90% of the load demand is not satisfied during Summer season. Therefore, the programmed load shedding has been used to prevent any possible shutdown. Furthermore, the unplanned P_D growth and improper system maintenance have led to cluster high loads (consumer) in the centre region.

1.3.3 Practical operating power constraints

Several practical operating power constraints need to be considered. Firstly, the inequality power constraints imposed on online TGUs are given by generation limits, RRLs ramp rate POZs, FOZs, VPL effects. Secondly, the equality power constraints imposed by the power grid are the P_L, mismatch in P_L ($P_{L,mismatch}$) and power balance. Under these practical operating power constraints, the objective function, i.e., fuel cost function, becomes non-convex, non-smooth and discontinuous. When the PGS consists of a large number of TGUs and it is with these power constraints, the optimization of such a complex problem becomes very hard.

1.3.4 High level of expenses in term of fuel cost

The total fuel cost of electricity production, in fossil fuel thermal power plants, becomes very high due to the usage of large-scale TGUs. For example, the MWh production in 2016 using thermal power plants only was 76,613,972 MWh with an average cost of \$100.00/MWh [2]. This cost is high compared with the average cost of 1

MWh production in other PGSs in the Middle East. For example, in UAE, the average cost is approximately \$70.00/MWh [7].

1.3.5 High level of emissions

Due to the usage of large-scale TGUs based on fossil fuel, they release a significant amount of the harmful pollutants, such as, carbon oxide (CO_2), sulfur dioxide (SO_2) and nitrogen oxides (NO_x) that cause significant and long term damages to the environment. For example, the study in [8] has shown that the average of CO_2 emission from online TGUs in Iraq was at about 1,190 g/kWh in 2015. Whereas, the standard and acceptable level of CO_2 emission by online TGUs is at about 453 g/kWh [9]. Therefore, the optimum dispatch of power from these online TGUs is required to reduce the amount of harmful pollutants.

1.3.6 Use of large number of TGUs

One of the important issues in PGS operation is finding an optimum solution to the practical ED problem. Efficient scheduling of the committed online TGUs results in significant cost savings. This scheduling becomes complicated when more and more TGUs need to be introduced into the PGS to meet the P_D while reducing the total fuel cost. Besides, optimum scheduling of all online TGUs while considering the practical operating power constraints further complicates the ED problem.

1.4 Motivation and research questions

The current issues discussed and reported in Section 1.3 give us the evidence and motivation in the re-considering reality of optimization of the ED problem in the current Iraq PGS. The issues associated with the explanations in Section 1.3 can be eliminated by finding an appropriate answer to the following questions:

Question #1: *Is the ED problem of large-scale TGUs in a power generating system with several power constraints a critical issue*?

Question #2: *What is the most effective technique/algorithm to solve such a complex ED problem*?

The answer to the above questions leads to analysis and formulation of the practical ED problem of large-scale TGUs under several practical operating power constraints and then finding novel algorithm(s) to solve such a complex problem. The answer to these questions is provided in this thesis.

1.5 Limitations of data availability

The NDC is facing a significant challenge in operation and management of large-scale TGUs with practical operating power constraints, since there is no use of an appropriate optimization methodology which would allow solving such a complex problem. However, due to recent developments in communication technologies and high-performance computing machines [10], the need to design or develop new optimization computational techniques for the ED problem in the PGSs becomes imperative.

An important limitation that is being faced in this study is that, I am not authorized to use data of the Iraq PGS, for security reasons. Therefore, due to availability of the technical data of other power systems that are similar to Iraq PGS, two real power systems, e.g., Taiwan [11] and South Korea [12] power systems are considered in this thesis. They are complex and each one is a large-scale TGU power system. In addition, small-scale and medium-scale power systems are also considered in this study, in order to evaluate the performance of the proposed algorithms.

1.6 List of the author's publications as part of thesis

The following eight Papers appended in Appendix-1 are carefully selected out of ten Papers and considered as main part of this thesis. The Papers A-D have been published in top-tier peer-reviewed international conferences. The Paper C was awarded a certificate of merit by the IEEE Victorian Section, Australia. The Papers E-G have been published in reputed peer-reviewed international journals, whereas, Paper H is under review in an international journal.

Paper A. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1-7.

Paper B. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of power under power grid constraints," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Hong Kong, 2015, pp. 14-19.

Paper C. **L. T. Al-Bahrani**, J. C. Patra, and R. Kowalczyk, "Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, 2016, pp. 660-667.

Paper D. **L. T. Al-Bahrani**, J. C. Patra, and R. Kowalczyk, "Multi-gradient PSO Algorithm for economic dispatch of thermal generating units in smart Grid," *in Proceedings of IEEE PES Innovative Smart Grid Technologies 2016 Asian Conference (ISGT'2016 Asia)*, Melbourne, Australia, 2016, pp. 258-263.

Paper E. **L. T. Al-Bahrani** and J. C. Patra, "A novel Orthogonal PSO algorithm based on orthogonal diagonalization," *Swarm and Evolutionary Computation*, vol. xxx, pp. 1-23, 2017. In press.

Paper F. **L. T. Al-Bahrani** and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid," *Applied Soft Computing,* vol. 58, pp. 401-426, 2017.

Paper G. **L. T. Al-Bahrani** and J. C. Patra, "Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid," *Energy*, vol. 147, pp. 1070-1091, 2018.

Paper H. **L. T. Al-Bahrani** and J. C. Patra, "Multi-gradient PSO algorithm with enhanced exploration and exploitation," *Applied Soft Computing*. Under review.

1.7 Contribution

As discussed in Sections 1.3 to 1.5, there is needed to develop powerful and effective optimization techniques to solve the ED problem for large-scale PGS with TGUs and power grid constraints. This thesis introduces two novel algorithms as stated below.

1.7.1 Orthogonal PSO

A novel orthogonal particle swarm optimization (OPSO) algorithm is proposed and applied to solve several complex unimodal and multimodal functions including ED problem. The *m* particles inside a swarm, i.e., possible solutions, are divided into two groups. An active group of best *d* particles and another is a passive group of (*m* – *d*) particles. The target of forming these two groups is to boost the diversity of the *m* particles inside the swarm. In every iteration, the active group particles subject to an orthogonal diagonalization process and are updated in which that their position vectors are orthogonally diagonalized. However, the passive group particles are not updated as their contribution in finding the correct direction is not important.

1.7.2 Multi-gradient PSO

Another novel multi-gradient particle swarm optimization (MG-PSO) algorithm is proposed and applied to solve several complex unimodal and multimodal functions including the ED problem and to reduce execution time that is an issue in the OPSO algorithm when solving high-dimension functions. Two phases used in the MG-PSO algorithm are called *Exploration* phase and *Exploitation* phase. The *m* particles in the *Exploration* phase are named *Explorers*. They undergo multiple episodes. In each episode, the *m* *Explorers* use a different negative gradient to explore a new neighbourhood. The *m* particles in the *Exploitation* phase are named *Exploiters*. They use only one negative gradient that is less than that of the *Exploration* phase, to exploit the best neighbourhood. This diversity in negative gradients gives a balance between global search and local search of the *m* particles.

Explanation of the learning strategy of the both OPSO and MG-PSO algorithms are presented in Papers A-H in Appendix-1. Furthermore, performance analysis of the two algorithms is carried out by considering the generation limits, VPL effects, P_L, $P_{L,mismatch}$, power balance, RRLs, and POZs as additional power constraints in solving the ED problem. In such cases, the fuel cost function is restricted by these power constraints and becomes non-linear, non-convex, multimodal and discontinuous.

The contribution of this research work stems from proposing two new algorithms which resulted in the eight high quality research articles. The salient features of the four international journal papers and four peer-reviewed international conference papers are given in Table 1.2. Here, in this study, the proposed OPSO and MG-PSO algorithms are used to solve the complex ED problem of small-scale to large-scale PGS.

Table 1.2. Contributions of the eight Papers, Paper A to Paper H, in this thesis.

Paper ID	Abbreviated	Contribution
Paper A	IEEE-IJCNN 2015	• Analysis and formulation the ED problem of small-scale PGS (6-TGU). • A novel algorithm called orthogonal particle swarm optimization (OPSO) algorithm was proposed. • The OPSO algorithm was evaluated and tested using 6 TGUs with RRLs and POZs constraints. • The OPSO algorithm was compared with several PSO variants and several other optimization methods. • Superior performance of the OPSO algorithm compared to several competing algorithms has been shown in terms of minimum, maximum and mean costs as well as standard deviation. • The OPSO algorithm was succeeded to improve the global PSO (GPSO) algorithm in terms of convergence rate, consistency and robustness.
Paper B	IEEE-SMC 2015	• Analysis and formulation the ED problem of medium-scale PGS (15-TGU). • The OPSO algorithm was proposed to solve the ED problem of 15 TGUs. • The OPSO algorithm was evaluated and tested using 15 TGUs with several equality and inequality constraints. • The OPSO algorithm was compared with several PSO variants and several other optimization methods. • The OPSO algorithm has provided better results in solving the total fuel cost of 15 TGUs and their power constraints. • Superior performance of the OPSO algorithm compared to several competing algorithms has been shown in terms of minimum, maximum and mean costs as well as standard deviation. • The OPSO algorithm was succeeded to improve the GPSO algorithm in terms of convergence rate, consistency and robustness.
Paper C	IEEE-IJCNN 2016	• Analysis and formulation the ED problem of large-scale Taiwan PGS (40-TGU). • The OPSO algorithm was proposed to solve the ED problem of Taiwan PGS. • The OPSO algorithm was evaluated and tested using Taiwan PGS with several equality and inequality constraints. • The OPSO algorithm was compared with several PSO variants and several other optimization methods.

		• The OPSO algorithm has provided better results compared to several competing algorithms in solving the total fuel cost of 40 TGUs with their power constraints. • The OPSO algorithm was succeeded to improve the GPSO algorithm in terms of convergence rate, consistency and robustness.
Paper D	IEEE (ISGT-Asia) 2016	• Analysis and formulation the ED problem of small-scale (6-TGU) and medium-scale (15-TGU) PGSs. • A novel algorithm called multi-gradient PSO (MG-PSO) algorithm was proposed. • The MG-PSO algorithm was evaluated and tested using 6 and 15 TGUs. • The MG-PSO algorithm has provided better results compared to several competing algorithms in solving the total fuel cost of 6 and 15 TGUs and their power constraints. • The MG-PSO algorithm was succeeded to improve the GPSO algorithm in terms of convergence rate, consistency and robustness.
Paper E	Swarm and Evolutionary Computation 2017	• The OPSO algorithm was proposed to improve the performance by overcoming the demerits of GPSO algorithm. • The OPSO algorithm was evaluated and tested using 30 unimodal and multimodal benchmark functions. • Superior performance of the OPSO algorithm compared with GPSO algorithm and several other optimization techniques has been shown in terms of convergence rate, accuracy, consistency, robustness and reliability. • The OPSO algorithm was found to be successful in achieving an optimum solution in all the 30 benchmark functions.
Paper F	Applied Soft Computing 2017	• The OPSO algorithm was proposed to solve the ED problem by taking three (small, medium and large) PGSs with several power constraints. • Mathematical analysis and theoretical justification of the OPSO algorithm was provided. • The OPSO algorithm was also applied for ten shifted and rotated benchmark functions. • Superior performance of the OPSO algorithm over the GPSO algorithm and several existing optimization techniques has been shown in terms of several performance measures. • Statistical significance of the OPSO algorithm has been shown using unpaired t-test against several contending algorithms including top-ranked CEC 2015 algorithms. • The OPSO algorithm has proved to be a robust and highly efficient algorithm which is capable of solving unimodal and multimodal functions including non-convex ED problem.
Paper G	Energy 2018	• The MG-PSO algorithm was proposed to solve the fuel cost function of medium-scale and large-scale PGSs with several power constraints. • Mathematical analysis and theoretical justification of the MG-PSO algorithm was provided. • The effectiveness of the MG-PSO algorithm was demonstrated using four real medium-scale and large-scale PGSs. • Superior performance of the MG-PSO algorithm has been shown over several PSO variants and several existing optimization techniques in terms of several performance measures. • The statistical t-test was carried out to demonstrate the effectiveness of the MG-PSO algorithm. • The MG-PSO algorithm has proved to be a robust and highly efficient algorithm which is capable of solving non-convex and multimodal ED problem.
Paper H	Applied Soft Computing 2018	• The MG-PSO algorithm was proposed to solve unimodal and multimodal complex problems. • Mathematical analysis and theoretical justification of the MG-PSO

		algorithm was provided. • The effectiveness of the MG-PSO algorithm was verified using ten selected shifted and rotated benchmark functions with dimensions of 30 and 100. • Superior performance of the MG-PSO algorithm has been shown over several PSO variants and several existing optimization techniques in terms of several performance measures. • Statistical significance of the MG-PSO algorithm has been shown using unpaired t-test against several contending algorithms including top-ranked CEC 2015 algorithms.

1.8 Outline of the thesis

This thesis is based on a combination of eight Papers and is integrated with five Chapters including the Introduction Chapter. The outline of rest of the Chapters presented in this thesis is organized as follows.

Chapter 2 provides the literature review for this thesis. In this Chapter, the investigations and research outcomes reported by other researchers are provided. Firstly, an overview of this Chapter is presented. Then, a brief explanation of the ED problem is given. After that, a review of the popular optimization techniques used for solving the ED problem with several TGU and PGS operating power constraints is presented. The classical optimization techniques, intelligent optimization techniques, i.e., evolutionary computation techniques (ECTs) and hybrid optimization techniques are addressed in this Chapter. Following that, several other improved ECTs are presented for solving such a complex problem. After that, several optimization techniques for ED problem of large-scale TGUs are discussed. Subsequently, some important observations are concluded from this literature review are presented in this Chapter. Finally, a summary of this Chapter is presented.

Chapter 3 provides the research methodology used in this study. An overview of this Chapter is presented. Then, research design of this thesis is given in this Chapter. Following this, introduction to optimization is presented. After that, a summary of the methods (algorithms developed) used in this study is presented. Thereafter, a comparison between the proposed OPSO and MG-PSO algorithms and original PSO variants in terms of several critical parameters is presented. Then, a comparison between the two proposed algorithms OPSO and MG-PSO algorithm, is provided. Finally, summary of this Chapter is presented.

Chapter 4 provides a summary of each Paper of the eight Papers is concisely provided. The contribution and the research methodology in each Paper are also presented. Finally, a summary of this Chapter is provided.

Chapter 5 provides a discussion of this study. Firstly, main investigations of this study are presented. Then, the answer of the research questions is provided. After that, outcomes of this study are discussed. Then, the significant contribution of this study to the knowledge is presented. Subsequently, limitations and recommendations future study are given. Finally, conclusion of this study is presented in this chapter.

Appendices: The eight Papers, Paper A to Paper H are attached in Appendix-1. The signed authorship indication form of each Paper is given in Appendix-2. Publisher permission for each published Paper is presented in the Appendix-3.

1.9 Chapter summary

One of the biggest challenges that is being faced by the PGS in Iraq is the ED of power of large-scale TGUs and solving practical operating power constraints. Because of use of inefficient optimization techniques to solve such a complex problem, billions of dollars are being wasted each year for the PGS. In addition, non-optimum scheduling of large-scale TGUs causes the PGS to be non-economic, unstable and unreliable. In order to solve the complex ED problem, two novel algorithms, i.e., OPSO and MG-PSO algorithms are proposed. In addition, this Chapter provided the details of contribution of each Paper of the eight Papers and the outline of this thesis.

Chapter 2: Literature Review

2.1 Chapter overview

The methodologies proposed by other researchers and related literature on economic dispatch (ED) problem are addressed in this Chapter. Chapter 2 aims to establish a framework of the research topic through a comprehensive review of a large number of the recent studies reported by other researchers, to address the economic dispatch (ED) problem. Section 2.2 briefly describes the ED problem. Several optimization techniques have been applied to solve the ED problem in the recent years. In general, these can be classified into three main categories: classical optimization techniques, evolutionary computation techniques (ECTs) and hybrid optimization techniques. Details of each category and their merits and demerits are presented in Sections 2.3, 2.4 and 2.5. Section 2.6 describes several studies that have been used to improve the performance of the ECTs to solve the ED problem of large-scale TGUs with several power constraints. In addition, some other optimization techniques using different methodologies are presented in Section 2.7. Some important observations are presented in Section 2.8. Finally, a summary of this Chapter is provided in Section 2.9.

2.2 Economic dispatch problem

Solving ED problem of the online TGUs with different operating power constraints is an essential part in operation and management of the PGSs. It aims to determine and assign the active output power of each online TGU for a given time interval to meet a specific load demand, P_D with minimum operating fuel cost subject to required equality and inequality power constraints. The theoretical justification and analysis of the ED problem, as well as mathematical formulation of the fuel cost function and its constraints, are available in the eight Papers appended in Appendix-1.

Ideally, the fuel cost of a TGU is characterised by a convex and smooth function [13]-[14]. The fuel cost function of each TGU is related to the active output power quadratically. Thus, it can be expressed by a quadratic function for solving of the ED problem. However, in practice, with large-scale TGUs operating under VPL effects and several other power constraints, e.g., RRLs, and POZs, the fuel cost function becomes non-convex, non-smooth and discontinuous with non-linear characteristics. In such the case, the cost function of the ED problem is represented by a multimodal objective function.

Many optimization techniques have been proposed to solve the ED problem. In general, the optimization techniques used for the ED problem can be classified into three main categories. They are classical optimization techniques, evolutionary computation techniques and hybrid optimization techniques.

2.3 Classical optimization techniques

The classical optimization techniques are beneficial to obtain the optimum solution of the problems that involve continuous and differentiable objective function. Such kind of optimization techniques can achieve the maximum or minimum solution for the unconstrained and constrained continuous objective function. The objective function and its constraints of these techniques are used to guide the optimization process [15]. This means that they require complete information of the objective function and its dependence on the nature of each variable of the objective function. The classical optimization techniques are widely used to solve power system operation problems including ED problem. Some of the favourite classical optimization techniques used in solving the ED problem are described below.

2.3.1 Linear programming

The linear programming (LP) method is widely used in science and engineering. It is being applied to problems of power systems, such as reactive power calculations, power flow and ED of active output power of PGSs [16]. The LP method is also successfully applied to economic growth prediction, design of diets, conservation of resources, transportation systems [17]. The LP method used for optimizing a linear objective function, is a maximization or minimization problem, subject to a number of linear constraints [18]-[19]. Thus, the objective function and its constraints have linear characteristics.

A maximization or minimization task is applied when solving the objective function. The structure of this method depends on the linear equation characteristics as follows.

$$\begin{aligned} &\text{Objective function} \quad && f(x) = c_1 x_1 + \cdots + c_d x_d \\ &\text{Subject to} && a_{1,1} x_1 + \cdots + a_{1,d} x_d \geq b_1 \\ & && \vdots \\ & && a_{n,1} x_1 + \cdots + a_{n,d} x_d \geq b_n \end{aligned} \tag{1}$$

where c_i, $a_{i,j}$ and b_{i} (i = 1, 2, …, n) and (j = 1, 2, …, d) are real numbers, which form the input to the objective function. The $f(x)$ to be minimized or maximized is the objective function and together with its constraints named the LP method [20]. The d is the dimension of $f(x)$. Each constraint can be interpreted as a half-space dividing the d-dimensional search space in two, and the intersection of all the half-spaces is named the feasible region, which is the set of all points satisfying all the constraints as shown in Figure 2.1. If the feasible region is empty or the solution vector x ($x = x_1, x_2, \ldots, x_d$) outside the feasible region, the LP method itself is infeasible as shown in Figure 2.1(a). If the solution vector x inside the feasible region then the LP method is feasible as shown in Figure 2.1(b). The solution may be bounded in the direction of the $f(x)$. In such the case, the solution vector x lies within the boundary of a search space, as shown in Figure 2.1(c). The solution may be bounded and valid within an intersection of two lines, in this case, there is a unique solution as shown in Figure 2.1(d). The solution is not always unique in LP method and the constraints can cause any possible solutions to be invalid.

The LP method is simple in implementation. It is most suitable for solving the problems that have linear characteristics. In addition, the convergence characteristics of this method are fast [21]-[22]. However, when the fuel cost function of the ED problem is non-convex, non-smooth and discontinuous with non-linear characteristics, the LP method may be unable to solve such a complex problem. In addition, the LP method is inaccurate to evaluate the equality and inequality power constraints, such as P_L and POZs constraints [15]. Furthermore, the LP method is inefficient to handle the problems that have high-dimensions because of the curse of dimensionality [23].

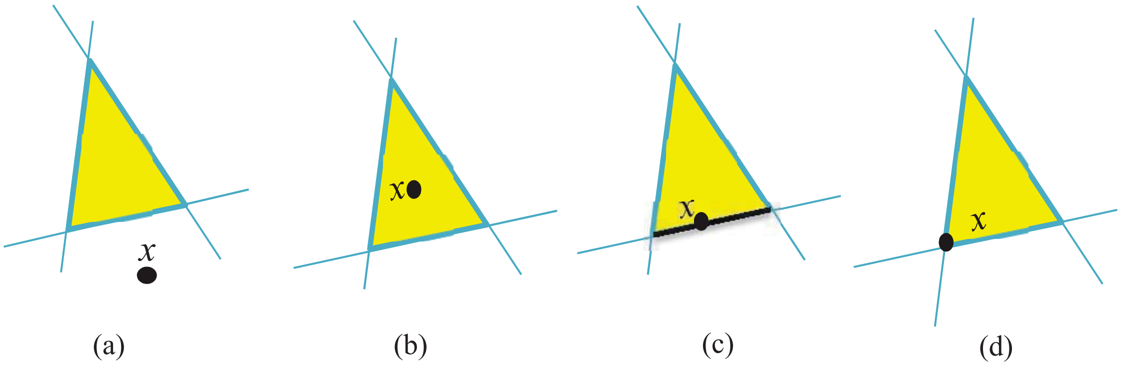


Figure 2.1. Possible solutions in linear programming method.

2.3.2 Mixed integer programming

The mixed integer programming (MIP) method is widely used in the field of optimization in power systems and communications. For example, unit commitment of TGU power systems and power flow analysis [24]-[25]. The MIP method is an optimization method where some or all the variables are restricted by integer values at the optimum solution [26]-[27]. The MIP method is formed as follows.

$$\begin{aligned}
&\text{Objective function} \quad f(x) = x^T Q\, x + q^T x \\
&\text{Subject to} \quad A_{eq}\, x = b_{eq} \quad \text{(linear equality constraints)} \qquad (2) \\
&\qquad A\, x \leq b \quad \text{(linear inequality constraints)} \\
&\qquad x^T Q_i\, x + q_i^{\,T} x \leq b_i \quad \text{(quadratic constraints)} \\
&\qquad l \leq x \leq u \quad \text{(bound constraints),}
\end{aligned}$$

where x, q, q_i, l, and u are vectors of d dimension. The number of equality and inequality constraints is denoted by M. Some or all of the solution vector values, x, are integer. The A and A_{eq} are $(M\times d)$ matrices, Q and Q_i are $(d\times d)$ matrices and the b_{eq}, b and b_i are vectors of equality, inequality and quadratic constraints, respectively.

Use of integer values expands the scope of useful optimization problems. However, the integer values make an optimization problem to be non-convex [27]-[28], and thus far more difficult to solve under high-dimensional search space optimization problems due to the complex combination of the integer variables. Also, many combinations of specific integer values for the variables must be tested, and each combination requires the solution of the optimization problem where the number of combinations may rise exponentially with the size of the optimization problem [29]. Furthermore, in most the real-world problems including ED problem, the optimum solution may be shifted or/and rotated at the origin. In such the case, tuning the integer values of the variables becomes hard. Moreover, the curve of the fuel cost function has some discontinuities due to the POZs imposed on several TGUs. The fuel cost function in such case becomes discontinuous and multimodal. Thus, solving such a complex problem by the MIP method may be very hard.

2.3.3 Sequential quadratic programming

Sequential quadratic programming (SQP) [30] is widely used in solving the practical optimization problems, e.g., image and signal processing, data analysis, reactive power optimization and ED problem [31]. The SQP is used for handling the non-linear objective function and its equality and inequality constraints. The SQP method closely simulates Newton's method for constrained optimization just as is done for the unconstrained optimization problems [32]. The non-linear optimization problem with both equality and inequality constraints can be written as a quadratic function. The quadratic function is then solved at each iteration. In each iteration, the Lagrangian function is used to update the objective function and its constraints. Then, the quasi-Newton approach is applied to generate a quadratic programming sub-problem whose solution is used to form a search direction as guide for a search procedure [33]. The procedure of the SQP method can be described as follows.

$$\begin{array}{ll} \text{Objective function} & f(x) = \Delta x^T \nabla f(x) + \frac{1}{2}\Delta x^T L\,\Delta x \\ \text{Subject to} & h_i(x) + \nabla h_i(x)^T \nabla x = 0\,,\ i = 1, 2, \ldots, k \\ & g_i(x) + \nabla g_i(x)^T \nabla x \le 0\,,\ i = 1, 2, \ldots, k+1, \end{array} \tag{3}$$

where k is the number of constraints, x is a position vector, Δx is a change in x in each iteration, $\nabla f(x)$ is a gradient of x, and L is Lagrangian function. The h_i and g_i are equality and inequality constraints, respectively. In each iteration, the SQP method approximates the $f(x)$ to a quadratic form and linearizes the constraints. The quadratic function is then solved to get Δx. The value of x is updated with Δx. Again, the $f(x)$ is approximated by a quadratic function, and its constraints are linearized with the new value of x. This procedure is repeated in each iteration until there is no further improvement in the objective function. In this method, a region around x has to be evaluated by Δx where a quadratic approximation of the function holds. The region is adjusted so that $f(x + \Delta x) < f(x)$.

The SQP method is widely used for the ED problem. Because, it can handle quadratic fuel cost function with non-linear characteristics [34]. However, as the fuel cost function to be minimized is multimodal and discontinuous due to presence of the POZs, the SQP method violates the POZs during the search process on a global optimum. In addition, the SQP method incorporates several derivatives, which may be evaluated in advance of iterating to a solution. Thus the SQP becomes quite slow for large-scale TGUs.

The details of some popular classical optimization techniques reported above reveal they are not effective in solving the ED problem for a large-scale highly constrained, non-linear, non-convex, and discontinuous optimization problem with multiple local optima. In general, many studies reveal that the classical optimization techniques are inefficient to solve such a complex problem [15], [23], [35]-[36] due to their major drawbacks as summarized in Table 2.1.

Table 2.1. Disadvantages of the classical optimization techniques

1. The convergence characteristics are sensitive to the initial values of vector x.
2. The objective function needs to be monotonically increasing in nature.
3. These techniques are primarily affected by the shape of the objective function.
4. These techniques are trapped into local minima under high-dimensions with several constraints.
5. These techniques are inefficient for handling the ED problem with a discontinuous search space.

2.4 Intelligent techniques: evolutionary computation techniques

Evolutionary computation techniques (ECTs) are a population-based optimization process. They are random search techniques inspired by the natural selection and survival of the fittest in the biological world [37]. The algorithm in these techniques comprises searching a population of candidate solutions. Each iterative step of an algorithm requires a competitive selection that eliminates poor candidate solutions. The candidate solutions with high fitness are recombined with other candidate solutions by swapping individuals of the approximate solution with those of the others. Some categories of the ECTs such as genetic algorithm (GA) and differential evolution (DE) use mutations in their learning strategies [35]-[38]. Mutations are applied to candidate solutions by making a small change to a single individual of the solution vector containing the decision variables. Recombination and mutation are used to yield new candidate solutions that are biased towards regions of the search space where good solutions have already been seen.

Some other categories of the ECTs that involve colonies of bees and ants, fish schools, birds flock, animal herds are called swarm intelligence [39]. The swarm intelligence mimics natural and artificial systems composed of many individuals that coordinate using decentralised control and self-organisation [40]. Here, the algorithm focuses on the collective behaviours that result from the local interactions of the individuals or candidate solutions with each other and with the environment where these individuals stay. The fundamental feature of the swarm intelligence is its ability to act in a coordinated way without the presence of an external controller or a coordinator. In spite of the lack of individuals in charge of the swarm or group, the swarm as a whole does show intelligent behaviour. This is the result of interactions of spatially located neighbouring individuals using guide rules.

The ECTs have been used successfully in many different fields of science and engineering including the ED problem. In the recent years, the ECTs have been widely used for solving real-world ED problem. Many researches have confirmed that the ECTs are suitable for solving the non-convex ED problem [41]-[75]. A brief description of several different types of the notable original ECTs for solving the ED problem is given below.

2.4.1 Genetic algorithm

Genetic algorithm (GA) is a search method used to find a possible best solution for an optimization problem. The GA belongs to the ECTs that inspired by evolutionary biology [41]. There are three key genetic operators of the GA, selection, crossover and mutation. The selection operator (reproduction) provides a driving force for the test system to evolve toward the desired states. The selection operator is essentially used for the intensive exploitation. The crossover operator (recombination) is mainly used for mixing within a sub-space. The crossover operator helps the individuals to exploit and enhance the algorithm convergence. Then, the mutation operator provides a main mechanism for a global search process, and it can be generalized as a randomization technique.

The GA generates the solutions of an optimization problem using the three operators (selection, crossover and mutation) with *i*th individual or candidate solution ($i = 1, 2, \ldots, m$) in a population (X). The X is denoted by a binary string of zeros (0s) and ones (1s) or sometimes using other forms of encodings as mentioned in [42]. The search process or the evolution process in GA starts from a population of individuals generated randomly within a d-dimensional search space and continues for iterations (generations). In each iteration, the fitness of each individual (candidate solution) is evaluated, and multiple individuals are randomly selected from the current population according to their fitness. Then, these are modified by the crossover and mutation operators to form a new population, which is then used in the next iteration of the evolution.

The fitness function f_i, $i = 1, 2, \ldots, m$, is defined to evaluate the fitness of *i*th individual and it is associated with the objective function of the problem. The fitness value of an individual should be positive. For a minimization task, when the objective function value becomes small, its fitness value is large. The fitness value of the individual is used to determine the probability (p_r) with which the individual is selected into the new population. This GA procedure is known as roulette wheel selection [42]. Some other alternatives of selection operators are also used, such as tournament selection. Subsequently, the search process terminates when either a maximum number of iterations have been achieved or when a satisfactory fitness level has been reached. The procedure of the GA is described in the following steps.

Procedure of GA [43]

Step #1: Encode the m individuals of the population (p) for the problem, generate the initial population $p(0)$.

Step #2: Evaluate the fitness of each individual $X_i(t)$, i =1, 2, …, m, in $p(t)$, t = 1, 2, …, N_{iter}, by f_i = fitness ($X_i(t)$).

Step #3: If the termination condition is achived, then the algorithm terminates, otherwise, determine the selection probability (p_r) of each individual by

$$p_{r,i} = \frac{f_i}{\sum_{j=1}^{m} f_j}, i = 1, 2, \ldots, m \tag{4}$$

Step #4: Using roulette wheel selection method, select multiple individuals from $p(t)$ with the above probability distribution to form a population as follows.

$$p^{(1)}(t+1) = \{X_i(t),\ i = 1, 2, \ldots, m\} \tag{5}$$

Step #5: Perform crossover operator on population $p^{(1)}(t+1)$ to form a population $p^{(2)}(t+1)$.

Step #6: Mutate a single element (called a gene) of an individual with probability $p_{r,m}$ to form a population $p^{(3)}(t+1)$.

Step #7: Set $t = t+1$, $p^{(1)}(t+1) = p^{(3)}(t+1)$ and return to Step #2.

In practice, the GA may often converge well and in many cases the global optimality can be achieved. The selection or survival of the fittest provides a good mechanism to select the best solution. This means that the individuals are able to enhance the convergence of the algorithm. However, when solving a complex problem such as the ED problem of large-scale TGUs with several power constraints, e.g., 40 TGUs or more, the mutation in GA may make the individuals far away from the global optimum while at the same time slowing down the convergence. Thus, the GA may fall into local minima for a such complex problem [13], [44].

2.4.2 Differential evolution

Differential evolution (DE) algorithm belongs to the class of ECTs. The DE algorithm was developed by R. Storn and K. Price in 1996 and 1997 [45]-[46]. It is a technique of mathematical optimization of multi-dimensional problems to find the global minimum (solution). It is fairly fast and reasonably robust. The DE algorithm is a population-based stochastic function minimizer and has become one of the most popular techniques used by the researchers [36]-[47] for solving the real-world problems including the ED problem. The DE can be explained as follows [48]-[49]. The learning strategy of DE algorithm depends on creating trial parameter vectors by adding the differential weight between two population vectors X_p and X_q to a third one X_r that makes the structure completely self-organizing. The solution vectors X_i, $i = 1, 2, \ldots, m$, are used for the next iteration (t+1) to optimize the objective function $f(x)$. The mutation in the DE algorithm [48] are written by

$$X_i = X_r + F\,(X_p - X_q), \qquad i = 1, 2, \ldots, m \tag{6}$$

where F is the differential weight and control parameter used to control the size of the disturbance in the mutation operator during the search process and to improve the convergence of the algorithm. The value of F is chosen using trial and error method within the range of [0,2] [48]. The X_r, X_p and X_q are three position vectors and they are generated by the random permutation. In addition, the DE has a crossover operator C_r that is controlled by a crossover probability $C_r \in [0,1]$. The actual crossover operator can be executed using binomial or exponential functions [49]. For the minimization task, the minimum objective value can be expressed by

$$X_i(t+1) = \begin{cases} U_i(t+1) & \text{if } \ f(U_i(t+1) \leq f(X_i(t)), \\ X_i(t) & \text{otherwise.} \end{cases} \tag{7}$$

One big challenge in the DE algorithm is that the search process is impaired during the search of the optimum solution due to the fast descending diversity of the m individuals. In addition, because of the diversity of the individuals descends faster during the search process, the DE is fast in convergence. Thus, the m individuals leads to a higher probability of reaching a local minimum [36], [50].

2.4.3 Particle swarm optimization

Particle swarm optimization (PSO) algorithm is another important algorithm developed by Kennedy and Eberhart in 1995 [51]. It was motivated by social behaviours of some animals such as birds flock or fish schooling. The PSO algorithm is a population-based search optimization algorithm with an initial population of random solutions. The possible solutions are usually called particles [51]. These particles fly in a d-dimensional search space by following their own experiences and the current optimum particles. The search for an optimum solution is obtained by updating the particles in each iteration.

The PSO algorithm begins with an initial population of random solutions. It searches for an optimum solution by updating the particles by changing the position of each particle in each iteration. It uses a velocity vector based on the social behaviour of the particles inside a swarm to update the current position of each particle flying in a *d*-dimensional search space. Each particle in a swarm adjusts its search direction using its personal experience, $G_{pers,i}$, i = 1, 2, …, m, and the best experience, G_{best} of the whole swarm through linear summation [52]. The velocity and position vectors of the *i*th particle, V_i and X_i, respectively, are updated as follows.

$$\begin{aligned} &V_i(t+1) = V_i(t) + c_1\, r_1\, [G_{pers,i}(t) - X_i(t)] + c_2\, r_2\, [G_{best}\,(t) - X_i(t)] \\ &X_i(t+1) = X_i(t) + V_i(t+1), \end{aligned} \tag{8}$$

where r_1 and r_2 are two random vectors with a range of [0,1]. The parameters $c_1$and c_2 are acceleration constants, which is typically taken as, $c_1 = c_2 = 2.0$ [53].

We can see from (8) that the new position $X_i(t+1)$ is generated by a pattern-search mutation, whereas selection is implicitly made by using the current global best solution, G_{best} found so far, as well as using the personal solution of *i*th particles, $G_{pers,i}$, i =1, 2, …, m. However, the role of individual best is not entirely clear, though the current global best seems very important for selection, as is shown in the accelerated PSO algorithm in [51] and [54]. Therefore, the PSO algorithm consists of mainly mutation and selection, but there is no crossover. This means that the PSO algorithm can have high mobility in particles with a high degree of an exploration.

The PSO algorithm has been successfully applied in different applications in science and engineering, such as optimum power flow, reactive power calculation, image processing and solving the ED problem [55]-[56]. It is now one of the most widely used algorithms in the optimization [4], [57]-[58]. The PSO algorithm is easy to implement and is fast in convergence. Besides, it has a few parameters to adjust. However, the use of G_{best} in PSO algorithm seems strongly selective. Its advantage is that it helps to speed up the convergence by drawing the optimum solution toward the current best G_{best}, however, it may lead to premature convergence by falling into a local minimum.

2.4.4 Ant and bee algorithms

Ant colony optimization (ACO) algorithm was developed by M. Dorigo [59]-[60]. It mimics the foraging behaviour of ants. The ACO algorithm uses pheromone as a chemical messenger and the pheromone concentration as an indicator of the quality solutions to a problem of interest. The candidate solutions are related to the pheromone concentration, which leading the individuals to routes and paths marked by the higher pheromone concentrations as better solutions. The ants are able to find shortest routes through marking their paths based on their behaviour with the pheromone concentration in finding an optimum solution. The shortest path is the route with the most pheromone concentration marks which the ants will use to carry their food back home.

In ACO algorithm, the random route generation is primarily mutation. Subsequently, pheromone-based selection gives a mechanism for selecting shorter routes. No explicit crossover operator available in ACO algorithm. However, the mutation operator is not as simple an action as flipping digits in the GA. The new solutions are essentially generated by fitness-proportional mutation [61]. The probability, p_r of ants in a grid problem at a particular node i to choose the route or path from node i to node j is given by

$$p_{r,ij} = \frac{\varphi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i,j=1}^{d} \varphi_{ij}^{\alpha} d_{ij}^{\beta}} \tag{9}$$

where $\alpha > 0$ and $\beta > 0$ are the influence parameters, φ_{ij} is the pheromone concentration on the route or path between the node i and the node j , and d_{ij} is the desirability of the same route. The selection is related to some a priori knowledge about the route or path, such as the distance s_{ij} is often used so that $d_{ij} \propto 1/s_{ij}$ [54].

Recent studies have shown that the ACO algorithm is highly effective in solving several real-world problems including the ED problem [61-63]. However, theoretical analysis of the ACO algorithm is difficult, sequences of random decisions are not independent, and probability distribution changes by iteration search are experimental instead of theoretical. In addition, although the execution time to reach convergence is uncertain, however, the convergence is guaranteed [63]. Besides, under high-dimensional multimodal ED problem, drawing the routes by the ants through the search process for the optimum solution becomes complex. In such the case, the ants are easy to fall into a local minimum [39].

In the bee colony optimization (BCO) algorithm, the bees in the colony are divided into three groups. They are the employed bees or forager bees, onlooker bees or observer bees, and scouts [64]-[65]. The employed bees and scout bees are initialized randomly and both are mainly used in mutation process. Selection operator is related to the honey or objective function. In the BCO algorithm, no explicit crossover operator is carried out. The detailed procedure of the BCO algorithm is given by the following steps:

Procedure of BCO algorithm

Step #1: Initialize the population of solutions X_{ij}, i = 1, 2, …, N_{bees}, where N_{bees} is number of onlooker bees and equals to the number of employed bees, j = 1, 2, …, d, where d is dimension of the problem.

Step #2: Evaluate the population at $t = 0$.

Step #3: Produce new solutions $X_{ij}(t+1)$, for the employed bees by using (10) and evaluate them.

$$X_{ij}(t+1) = X_{ij}(t) + \varphi_{ij}\,(X_{ij}(t) - X_{kj}(t)) \tag{10}$$

where $k \in \{1, 2,..., N_{bees}\}$ and $j \in \{1, 2,..., d\}$ are randomly chosen indexes. The φ_{ij} is a random number with a rage of [-1,1].

Step #4: Apply the greedy selection process [66].

Step #5: Calculate the probability values for the solutions $X_{ij}(t+1)$ by using (11)

$$p_{r,i} = \frac{f_i}{\sum_{i=1}^{N_{bees}} f_i} \tag{11}$$

where f_i is the fitness value of the solution i which is proportional to the nectar amount of the food source.

Step #6: Produce the new solutions $X_{ij}(t+1)$ for the onlookers from the solutions $X_{ij}(t)$ selected depending on $p_{r,i}$ and evaluate them.

Step #7: Apply the greedy selection process [66].

Step #8: Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution $X_{ij}(t)$ using (12)

$$X_{ij} = X_{j,min} + \text{rand}\,(0,1)\,(X_{j,max} - X_{j,min}) \tag{12}$$

where $X_{j,min}$ and $X_{j,max}$ are the limits for the abandonment.

Step #9: Memorize the best solution achieved so far.

Step #10: $t = t + 1$

Step #11: Go to Step #3 until $t = N_{iter}$, where N_{iter} is total number of iterations or sometimes called total number of cycles [67]-[68].

Both the ACO and BCO algorithms use only mutation and fitness-related selection operators, and they can have good global search ability to explore the search space relatively effectively. However, the convergence may be slow because they lack crossover operator. In addition, the local search ability of the individuals is comparatively low [69]. This may explain that the ACO and BCO algorithms can perform well for some optimization problems with low dimensional search space. However, when solving a complex problem, such as the ED problem with a high-dimensional search space, obtaing of the global optimum using the ACO or BCO algorithms may become hard.

2.4.5 Cuckoo search algorithm

Cuckoo search (CS) algorithm was developed in 2009 [70]. It is a nature-inspired algorithm. The CS algorithm is based on the brood parasitism of some cuckoo species. It is enhanced by the so-called Lévy flights rather than by simple isotropic random walks [71]. The recent studies show that the CS is potentially more efficient than the PSO algorithm and GA [72-74]. The CS algorithm uses a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter p_a . The local random walk can be expressed by

$$X_i(t+1) = X_i(t) + \alpha\, s \otimes H(p_a - \varepsilon) \otimes (X_j(t) - X_k(t)), \tag{13}$$

where $X_j(t)$ and $X_k(t)$ are position vectors with different solutions selected randomly by random permutation, H is a Heaviside function, ε is a random number drawn from a uniform distribution, and s is the step size. The symbol ⊗ denotes the entry-wise product.

On the other hand, the global random walk is accomplished by using Lévy flights [71], as flows.

$$X_i(t+1) = X_i(t) + \alpha\, L(s, \lambda), \tag{14}$$

where,

$$L(s,\lambda) = \frac{\lambda\,\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \times \frac{1}{s^{1+\lambda}}, \quad (s \le s_0 \le 0) \tag{15}$$

where, the step size scaling factor $\alpha > 0$, which should be related to the scales of the problem of interest.

The CS algorithm has some drawbacks: it may be particular for a specific class of the optimization problems but may not be effective in other real-world problems, such as the ED problem with high-dimensional search space because it may get stuck into a local optimum [75].

2.4.6 Bat algorithm

The bat algorithm (BA) was developed in 2010 [76]. It is inspired by the echolocation behaviour of bats. The BA uses frequency tuning to obtain the global optimum. Each bat is associated with a velocity vector $V_i(t)$ and a position vector $X_i(t)$, at iteration t , in a d-dimensional search space. Among all the bats in a swarm, there exists a current best solution denoted by X_{best}. The $X_i(t)$ and $V_i(t)$ are updating as follows.

$$R_i = R_{min} + (R_{max} - R_{min})\,\beta \tag{16}$$

$$V_i(t+1) = V_i(t) + (X_i(t) - X_{best})\,R_i \tag{17}$$

$$X_i(t+1) = V_i(t+1) + X_i(t) \tag{18}$$

where R_{min} and R_{max} are range of the search space of a problem, and $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. The loudness and pulse emission rates are regulated as follows.

$$A_i(t+1) = \alpha\, A_i(t) \tag{19}$$

$$r_i(t+1) = r_i(t)\,(1 - \exp(-\gamma\, t), \tag{20}$$

where $0 < \alpha < 1$ and $\gamma > 0$ are constants.

In the BA, the frequency tuning essentially acts as mutation operator, whereas selection operator pressure is relatively constant via the use of the current best solution X_{best} found so far. There is no explicit crossover operator. However, mutation operator varies due to the variations in loudness and pulse emission. The BA is simple, flexible and easy to implement. A wide range of problems including highly non-linear problems have been solved efficiently [77]. Also, the variations in loudness and pulse emission rates provide a mechanism for automatic control and auto-zooming ability so that exploitation becomes intensive as the search approaches the global optimality. In addition, the BA is fast in convergence. However, no mathematical analysis is used to link the parameters with convergence rates in this algorithm.

The studies carried out by a large number of other researchers in this Section reveal that the original ECTs have several merits and demerits for solving the complex high-dimensional search space problems, such as the ED problem. These merits and demerits are summarized in Tables 2.2 and 2.3, respectively.

Table 2.2. Merits of the ECTs.

1- The ECTs have the adaptability to change and ability to generate good enough solutions quickly.

2- These are not affected by the shape of the objective function.

3- The ECTs use information of the objective function directly through the search processes.

4- The ECTs can deal with non-smooth, non-continuous, and non-differentiable objective function.

5- The ECTs use stochastic transition rules, rather than deterministic rules, to select the population (individuals) in each iteration.

6- The ECTs can search a complicated and uncertain search space to find a global optimum.

7- The ECTs can deal with complex problems that cannot be solved by the classical optimization techniques.

8- The ECTs are easy to apply due to their simple mathematical structure and easy to combine with other techniques to create hybrid systems adding the strengths of every other technique.

Table 2.3. Demerits of the ECTs.

1. The ECTs may reach premature convergence and non-optimum local solution when solving the objective function with non-linear and discontinuous characteristics.
2. The ECTs may have weakness in either exploration or exploitation of the individuals during the search process.
3. The ECTs may lose the balance between global search and local search during the search process when solving the complex problems with high-dimensional search space.

2.5 Hybrid optimization techniques

The hybrid optimization technique is an integration of two or more optimization methods to provide complementary learning, searching and reasoning methods to combine domain knowledge and empirical data to develop flexible computing tools and then solving the real-world complex problems [78]. The aim of this integration is to overcome the limitations of the individual optimization techniques, e.g., classical optimization techniques and original ECTs through hybridization. The ECTs are robust and powerful global optimization techniques for solving many real-world problems. However, they are sometimes poor in terms of the convergence performance. Besides, most the original ECTs lack the balance between local search and global search during the search for a global optimum. Therefore, the hybrid algorithms are designed to yield better performance than the individual algorithms. Most studies achieved in the recent years show that hybrid optimization techniques include one or two original ECTs in their learning strategies [79]-[78]. Some of the major hybrid algorithms used in solving the ED problem are given below.

2.5.1 Hybridization between GA and another ECT

The GA is used with different types of ECTs for solving several practical problems including the ED problem. Some of these GA-based hybrid algorithms are a hybrid GA and bacterial foraging approach [79]. The original bacterial foraging (BF) algorithm suffers from poor convergence characteristics when solving the ED problem with a number of power constraints, e.g., VPL effects, RRLs and P_L. To overcome this drawback, integration between the GA and BF algorithm is made to obtain a hybrid GA with BF (HGBF) algorithm. In a combination of GA and SQP method [80], the GA is used as main optimizer and the SQP method is combined to fine tune in the solution of the GA run. In addition, non-uniform mutation operator and simplex crossover operator are achieved through this hybridization. The GA is combined with an immune algorithm to produce a hybrid algorithm, is called hybrid immune-genetic algorithm (HIGA) [81]. The HIGA algorithm is used to solve multimodal and non-convex ED problem. These algorithms are successfully used in solving the ED problem, however, they are applied only for small-scale and medium-scale PGSs.

2.5.2 Hybridization between DE and another ECT

The DE has been combined with other types of ECTs for solving ED problem of large-scale TGUs under several non-linear power constraints. The DE algorithm is combined with greedy randomized adaptive search procedure (GRASP) to produce DE-GRASP algorithm [12]. This algorithm is used to improve the global searching capability and to prevent from falling into local minima. The DE-GRASP algorithm is evaluated using medium-scale to large-scale PGSs. A combination of the DE algorithm and harmony search (HS) algorithm is made to create a hybrid algorithm and is called DHS algorithm [82]. Here, in order to enhance the exploitation ability of harmony search, the pitch adjustment operation is operated with the different mutation operations. In addition, to enhance the exploration ability of evolution search, both the memory consideration and the pitch adjustment are made. This hybrid algorithm is evaluated using small to large PGSs. Another hybrid algorithm is called DE-PSO-DE (DPD) is built by paralleling the DE algorithm and the PSO algorithm [83]. Here, the population is divided into three groups. The DE algorithm enhances the inferior and superior groups, whereas the PSO algorithm is used to enhance the mid-group. This combination improves the solution quality of each individual algorithm. However, such a hybrid algorithm consumes long execution time due to more computational processes.

2.5.3 Hybridization between PSO and another ECT

The PSO algorithm has been combined with some other types of the ECTs. In the fuzzy based hybrid PSO-DE (FBHPSO-DE) algorithm [3], the combination of the DE and PSO algorithms is made to enhance the ability of the population in local search and global search processes. A fuzzy decision making strategy is applied to find and sort the Pareto-optimal solutions. The FBHPSO-DE algorithm is evaluated using small-scale to large-scale PGSs to solve the ED problem with different power constraints. The PSO algorithm is modified using gravitational search algorithm (GSA) to construct a hybrid algorithm called PSOGSA [84]. Here, the local search ability of the PSO algorithm is improved by using the GSA. In addition, a fuzzy logic is used to control the ability to search for the global optimum and to increase the performance of the PSOGSA. This algorithm is evaluated using small-scale to large-scale PGSs. The co-swarm shrinking hypersphere PSO (CSHPSO) algorithm is obtained by hybridizing the shrinking hypersphere PSO algorithm with the DE algorithm [85]. Here, the swarm is divided into two sub swarms in such a way that the first swarm uses the SHPSO algorithm, whereas the second swarm uses the DE algorithm. Thus, this procedure enhances the exploration and exploitation process to obtain a global optimum.

Table 2.4 gives a summary of some hybrid algorithms used in the recent years to solve the non-convex and non-smooth ED problem of large-scale TGUs. We can see from Table 2.4 that these hybrid algorithms are successfully used in solving the ED problem of large-scale TGUs but with only a few power constraints. This means that addition of more constraints requires more time for the algorithm to find a global optimum. In addition, these algorithms are found to generate a high-quality solution, robustness and consistency for such a complex problem.

In addition to these merits of the hybrid optimization techniques in improving performance and solution of the original ECTs, they have several critical demerits as shown in Table 2.5.

Table 2.4. Some of the hybrid algorithms used in the recent years for solving the ED problem.

Sl. No.	Algorithm	Year	PGS		Fuel cost	Power constraints						
			# of TGUs	P_D (MW)		RR limits	Generation limits	P_L	P_L mismatch	Power balance	POZs	VPL effects
1	Fuzzy based hybrid PSO-DE (FBHPSO-DE) [3]	2017	40	10,500	√	-	-	√	-	-	√	√
			160	43,200	√	-	-	√	-	-	√	√
2	DE with greedy randomized adaptive search (DE-GRAS) [12]	2017	40	10,500	√	-	√	-	-	-	-	√
			140	49,342	√	√	√	-	-	-	√	√
3	DE-PSO-DE (DPD) [83]	2016	40	10,500	√	-	√	-	-	-	-	√
4	PSO with gravitational search algorithm (PSOGSA) [84]	2015	40	10,500	√	-	√	-	-	-	-	√
5	Co-swarm shrinking hypersphere PSO CSHPSO [85]	2014	40	10,500	√	-	√	-	-	-	-	√
6	DE with harmony search (DHS) [82]	2013	40	10,500	√	-	√	-	-	-	-	√

The symbol (√) represents that the corresponding constraint has been considered. The symbol (-) represents that the corresponding constraint has not been considered.

Table 2.5. Demerits of the hybrid optimization techniques.

1. The hybrid optimization techniques are often consume a long execution time because of the structure of these techniques is complicated. During this time of the search process, the algorithm may fall into a local optimum.
2. The parallelization between two algorithms is a key of these techniques.
3. The appropriate integration of the combined algorithms may be difficult to achieve.
4. Choosing suitable parameters values is difficult.
5. The hybrid optimization techniques suffer from slow convergence to an optimum value and require a large number of iterations in high-dimensional search space problems.
6. The balance between exploration and exploitation processes of the combined algorithms is hard to obtain.

2.6 Enhanced evolutionary computation techniques

In spite of the advantages of the original ECTs to solve the ED problem, they are often prone to get trapped into local optima due to the loss of balance between global search and local search of the individuals in large-scale TGUs with several non-linear operating power constraints [13], [44]. To enhance the global search and local search performance of the original ECTs to solve such a complex problem, many improved variants have been developed in the recent years. Among them, extensive studies have been made on the improved PSOs [11], [55], [57] [86]-[88], [91], [94] and DEs [89]-[90], [92]-[93], because of their popularity. These are summarized in Table 2.6.

Table 2.6 summarizes some of the improved PSO and DE algorithms used in the recent years to solve the non-convex and non-smooth ED problem. These algorithms improve the performance of the PSO and DE algorithms in terms of the high-quality solution, convergence rate, robustness and consistency. Also, they successfully applied to solve the ED problem of large-scale PGSs. However, they address few numbers of power constraints. For example, in Table 2.6, there is no algorithm solves the P_L and $P_{L,mismatch}$ constraints and satisfies the equality constraint due to the power balance of large-scale TGUs. This is due to that the data of these practical power systems may be not available or due to the complexity of these power systems. The optimization techniques mentioned in Table 2.6 are arranged based on date of the publication.

Table 2.6. Some of the improved PSO and DE algorithms used in the recent years for solving the ED problem.

Sl. No.	Algorithm	Year	PGS		Fuel Cost	Power constraints						
			# of TGUs	P_D (MW)		RR limits	Generation limits	P_L	$P_{L,mismatch}$	Power balance	POZs	VPL effects
1	Dynamic PSO with escaping prey (DPSOEP) [87]	2017	40	10,500	√	√	√	-	-	-	-	√
2	Improved random drift PSO (IRDPSO) [57]	2017	40	10,500	√	√	√	-	-	-	√	√
			140	49,342	√	-	-	-	-	-	-	√
3	Synergic predator-prey optimization (SPPO) [88]	2016	140	49,342	√	√	√	-	-	-	√	-
4	Colonial competitive DE (CCDE) [89]	2016	40	10,500	√	-	-	-	-	-	-	√
			140	49,342	√	√	√	-	-	-	√	√
5	Evolutionary DE (E-DE) [90]	2016	100	Not available	√	-	-	-	-	-	-	-
			150	Not available	√	-	-	-	-	-	-	-

6	Modified PSO (MPSO) [91]	2015	40	10,500	√	√	√	-	-	-	-	-
			140	49,342	√	√	√	-	-	-	√	√
7	Random drift PSO (RDPSO) [11]	2014	40	8,550	√	-	-	-	-	-	-	-
8	Improved DE (IDE) [92]	2014	40	10,500	√	-	-	-	-	-	-	√
			140	49,342	√	√	√	-	-	-	√	√
9	Shuffled DE (SDE) [93]	2013	40	10,500	√	-	-	-	-	-	-	√
10	Theta PSO (θ-PSO) [86]	2013	40	10,500	√	-	√	-	-	-	-	√
11	Iteration PSO with time varying acceleration coefficients (IPSO-TVAC) [94]	2012	40	10,500	√	√	√	-	-	-	-	-
12	Improved PSO (IPSO) [55]	2010	40	10,500	√	-	-	-	-	-	-	√
			140	49,342	√	√	√	-	-	-	√	√

The symbol (√) represents that the corresponding constraint has been considered. The symbol (-) represents that the corresponding constraint has not been considered.

2.7 Other optimization techniques used for the ED problem

Despite the success of the ECTs in solving the non-convex, non-smooth ED problem of large-scale TGUs with several operating power constraints, some other types of the optimization techniques use different learning strategies to address such a complex problem. List of some notable such optimization techniques are shown in Table 2.7. The optimization techniques mentioned in Table 2.7 are arranged based on date of the publication.

Table 2.7 summarizes some other optimization techniques used in the recent years to solve non-convex and non-smooth ED problem of large-scale TGUs. In addition, Table 2.7 indicates the current research trends of using different methodologies to solve such a complex problem. One can see that, despite these techniques reported in Table 2.7 were solved the ED problem of large-scale PGSs, they did not solve for all equality and inequality operating power constraints imposed by the TGUs and PGS. For example, the non-convex and non-smooth fuel cost function was evaluated by TPMIP algorithm [95] under RRLs, generation limits, P_L, P_L mismatch, power balance, POZs and VPL effects. However, the equality constraints, i.e., $P_{L,mismatch}$, and power balance, are not taking into consideration.

Table 2.7. Some notable optimization techniques used in the recent years for solving the ED problem.

Sl. No.	Algorithm	Year	PGS		Fuel cost	Power constraints						
			# of TGUs	P_D (MW)		RRLs	Generation limits	P_L	$P_{L,mismatch}$	Power balance	POZs	VPL effects
1	Parallel hurricane optimization algorithm (PHOA) [96]	2018	40	4,242	√	-	-	√	-	-	-	-
2	Oppositional grey wolf optimization (OGWO) [97]	2017	40	10,500	√	-	-	-	-	-	-	√
			140	10,500	√	-	-	-	-	-	-	√
			160	43,200	√	-	-	-	-	-	-	√
3	Lighting flash algorithm (LFA) [98]	2017	40	10,800	√	-	-	-	-	-	-	√
			80	21,600	√	-	-	-	-	-	-	√
			160	43,200	√	-	-	-	-	-	-	√
			320	172,800	√	-	-	-	-	-	-	√
4	Parallel augmented Lagrangian relaxation (PALR) [99]	2017	323		√	√	√	-	-	-	-	-
5	Hybrid grey wolf optimizer (HGWO) [100]	2016	40	10,500	√	-	√	√	-	-	-	√
			80	21,000	√	-	√	√	-	-	-	√
6	Modified symbiotic organisms search (MSOS) [101]	2016	40	10,500	√	-	√	√	-	-	√	√
			80	21,000	√	-	√	√	-	-	√	√
			160	31,500	√	-	√	√	-	-	-	√
			360	42,000	√	-	√	√	-	-	-	√
7	Crisscross optimization (CSO) [102]	2016	40	10,800	√	-	-	-	-	-	-	√
			80	21,600	√	-	-	-	-	-	-	√
			160	43,200	√	-	-	-	-	-	-	√
			320	86,400	√	-	-	-	-	-	-	√
			640	172,800	√	-	-	-	-	-	-	√
8	Modified social spider algorithm (MSSA) [103]	2016	40	10,500	√	-	-	-	-	-	-	√
			80	21,000	√	-	-	-	-	-	-	√
			140	49,342	√	-	-	-	-	-	√	√
9	Tournament-based harmony search (THS) [104]	2016	40	21,000	√	-	-	-	-	-	-	√
			80	21,000	√	-	-	-	-	-	-	√
10	Theta-modified bat algorithm (θ-MBA) [105]	2016	40	21,000	√	-	-	-	-	-	√	√
11	Chaotic bat algorithm (CBA) [77]	2016	40	10,500	√	-	-	-	-	-	-	√
			160	43,200	√	-	-	-	-	-	-	√
12	Flower pollination algorithm (FPA) [106]	2016	40	10,500	√	-	-	-	-	-	-	√
13	Two-phase mixed integer programming (TPMIP) [95]	2016	40	10,500	√	-	-	-	-	-	-	√
			40 MW	7,000	√	√	√	√	-	-	√	√
			140	49,342	√	√	√	√	-	-	√	√
14	Opposition-based greedy heuristic search (OGHS) [107]	2016	40	10,500	√	-	-	-	-	-	-	√
			52	7200	√	-	√	-	-	-	-	√
			52	10,800	√	-	√	-	-	-	-	√
			140	49,342	√	-	-	-	-	-	√	√
15	Oppositional invasive weed algorithm (OIWO) [108]	2015	40	10,500	√	-	-	√	-	-	-	√
			40	10,500	√	-	-	-	-	-	-	√
			110	15,000	√	-	-	-	-	-	-	-
			140	49,342	√	-	-	-	-	-	√	√
16	Fully decentralized approach (FDA) [109]	2015	40	10,500	√	-	-	-	-	-	-	√

17	Mixed integer quadratic programming (MIQP) [110]	2014	40	7,000	√	√	√	√	-	-	√	-
			40	10,500	√	-	-	-	-	-	-	√

The symbol (√) represents that the corresponding constraint has been considered. The symbol (-) represents that the corresponding constraint has not been considered.

2.8 Some important observations

Several observations have been made based on the review of the current literature, as stated below.

2.8.1 Observation #1

The first research question that has been mentioned in Section 1.4 is "*Is ED problem of large-scale TGUs in a power generating system with several power constraints a critical issue*?"

To answer this question, the discussion of the current issues of the ED problem in Section 1.3 and outcomes of the literature review in this Chapter reveal that the ED problem is in deed a critical issue. The ED problem becomes significant, especially in a large-scale power system, when there is a need of optimal scheduling of TGU outputs with various power constraints with respect to the predicted load demand, P_D. In order to economize the operating fuel cost, optimal dispatch of power generation from all the online TGUs while satisfying highly nonlinear power constraints to satisfy the power demand is in deed a critical issue. This issue need to be addressed so as to make the PGS economical, stable and reliable.

2.8.2 Observation #2

The comprehensive literature review reveals that the ECTs are continuously being developed and improved year by year as shown in Tables 2.4 and 2.6, to deal with ever larger PGSs with an increasing number of the power constraints. Thus, the need to design a new approach or develop new optimization techniques instead of the inefficient current optimization methods becomes imperative.

2.8.3 Observation #3

Although a large number of the ECTs-based algorithms have been proposed and applied for optimization of the non-convex and non-smooth fuel cost function, the issue of scalability has not been addressed sufficiently as shown in Tables 2.4 and 2.6. In other words, the number of ECTs which can actually be applied for the ED problem of large-scale TGUs with several power constraints remains low. Thus, this is a gap in this field.

2.8.4 Observation #4

This comprehensive literature review reveals that the PSO algorithm is widely used in this field compared with the other ECTs as shown in Tables 2.4 and 2.6. Besides, several studies in this literature review confirm that the performance of the original PSO algorithm were clearly improved by several newly proposed PSO-based algorithms [11], [55], [57], [86]-[88], [91], [94].

2.9 Chapter summary

The literature review presented in this Chapter provided a comprehensive study of the optimization techniques used in the recent years to solve an important real-world problem in operation and management of large-scale TGUs to govern electrical energy to the consumer, i.e., the ED problem. Different categories of the optimization techniques were used for such a complex problem. Among them, the ECTs are widely used and updated year by year. However, the number of ECTs which can actually be applied for the ED problem of large-scale TGUs with several power constraints remains low and this is a gap in this field.

The PSO algorithm is one popular type of the ECTs and has been widely used for the ED problem. However, the recent studies reported in literature review confirm that the original PSO algorithm is inefficient to solve the ED problem of large-scale TGUs with several power constraints.

The comprehensive study in this literature review tells us that the ED problem is a critical issue and the need to design new approaches to solve such a complex problem is necessary.

Chapter 3: Research Methodology

3.1 Chapter overview

This Chapter provides the research methodology-based on the research design to develop new algorithms and to answer the research questions outlined in Section 1.4. Summary of the research design is provided in Section 3.2. Then, algorithms developed are explained in Section 3.3. Following that, the comparison between the proposed algorithms and original PSO variants in terms of several critical parameters is presented in Sections 3.4 to 3.6. Finally, a summary of this Chapter is presented in Section 3.7.

3.2 Research design

In order to address the two research questions underpinning this thesis, the procedure in Figure 3.1 shows the research design carried out in this thesis. The research design provides the methodology of this thesis.

The first research question that mentioned in Section 1.4 was investigated in Section 2.8.1.

To investigate the second research question that mentioned in Section 1.4 is "*What is the most effective technique/algorithm to solve such a complex ED problem*?"

First of all, an overview of the learning strategy of global particle swarm optimization (GPSO) algorithm and critical drawbacks that make it unable to solve such a complex problem is presented. Then, the proposed orthogonal PSO (OPSO) and multi-gradient (MG-PSO) algorithms based on the research design plan shown in Figure 3.1 are explained. The details of OPSO and MG-PSO algorithms and their applications are available in the eight Papers appended in Appendix-1.

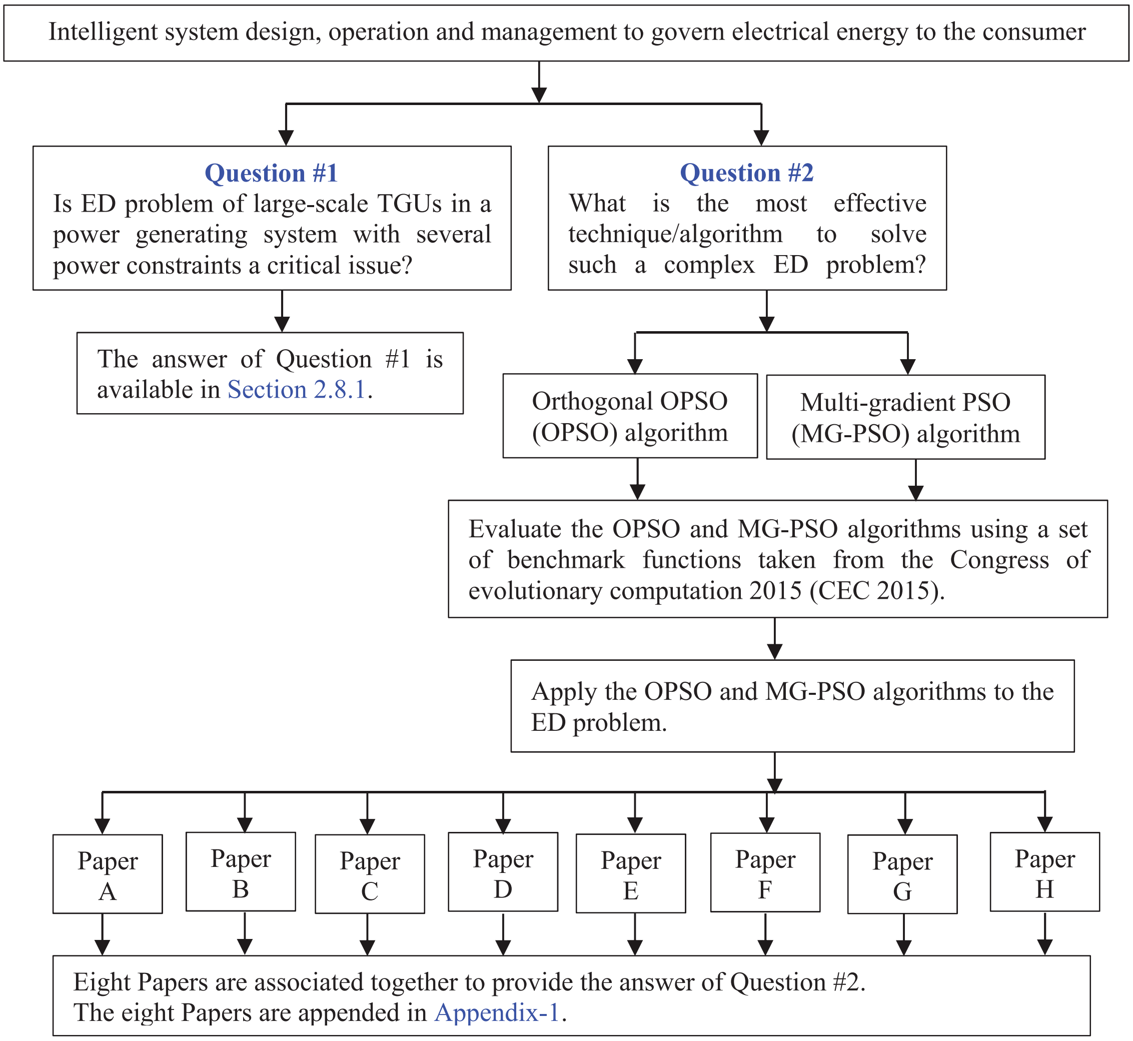

Intelligent system design, operation and management to govern electrical energy to the consumer
Question #1
Is ED problem of large-scale TGUs in a power generating system with several power constraints a critical issue?
Question #2
What is the most effective technique/algorithm to solve such a complex ED problem?
The answer of Question #1 is available in Section 2.8.1.
Orthogonal OPSO (OPSO) algorithm
Multi-gradient PSO (MG-PSO) algorithm
Evaluate the OPSO and MG-PSO algorithms using a set of benchmark functions taken from the Congress of evolutionary computation 2015 (CEC 2015).
Apply the OPSO and MG-PSO algorithms to the ED problem.
Paper A
Paper B
Paper C
Paper D
Paper E
Paper F
Paper G
Paper H
Eight Papers are associated together to provide the answer of Question #2.
The eight Papers are appended in Appendix-1.

Figure 3.1. Research design of this thesis.

3.3 Algorithms developed

Here, one version of the original PSO variants used in this thesis is the GPSO algorithm. Demerits that make the GPSO algorithm unable to solve the practical ED problem of large-scale TGUs are discussed. In addition, two proposed novel algorithms, i.e., orthogonal PSO and multi-gradient PSO algorithms used in this thesis are investigated and their performances are evaluated.

3.3.1 Global particle swarm optimization

As discussed in Section 2.4.3, the original PSO algorithm [51] is a population-based optimization technique. It emulates behaviours of some animals such as flock of birds or schools of fish. The population is called a swarm, and the individuals or possible solutions are called particles. The original PSO algorithm operates on a randomly created swarm of particles in a multi-dimensional search space. It searches for an optimum solution by updating the velocity and position of the particles according to a guiding rule. Each particle learns from its personal experience and its neighbourhood's experience. That is, each particle flying in the space, searches for an optimum solution by adjusting its flying trajectory according to its own experience and its neighbourhood's experience. Depending on the selection of neighbourhood formation, the original PSO algorithm is classified into two versions: global PSO (GPSO) algorithm, and local PSO (LPSO) algorithm. Without loss the generality, this thesis studies performance of the GPSO algorithm. The merits and demerits of GPSO algorithm compared to LPSO algorithm are stated as follows.

i. The GPSO algorithm is simple in structure and easy to implement compared to LPSO algorithm [111].

ii. The GPSO algorithm has faster convergence rate that the LPSO algorithm [111].

iii. The particles in GPSO algorithm are prone to fall into local minima [112].

iv. The GPSO algorithm uses only one neighbourhood topology structure (fully connected network) for the particles inside a swarm. Whereas, in the LPSO algorithm, different neighbourhood topologies are used for the particles, e.g., ring structure, pyramid structure, and Neumann structure [113-115].

The learning strategy of GPSO algorithm is explained as follows. This algorithm depends on the distribution of the particles in a swarm, i.e., neighbourhood topology structure of the *m* particles, as shown in Figure 3.2. One can see from Figure 3.2 that the *m* particles are neighbours of each other. Also, they are attracted to the best particle in a swarm to form a fully connected network.

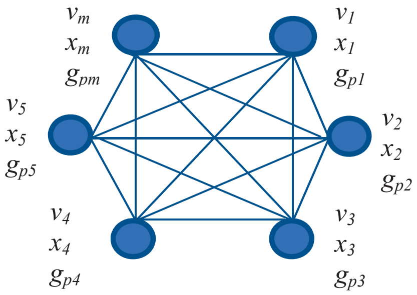


Figure 3.2. Graphical representation of the neighborhood topology (fully connected network) for the *m* particles of GPSO algorithm.

The learning strategy of GPSO algorithm is as follows. Firstly, Each particle i ($i = 1, 2, \ldots, m$) in a swarm flying in a d-dimensional search space adjusts its flying path based on two guides, its own experience, $G_{pers,i}$, and its neighbourhood's best experience, G_{best}. Secondly, when pursuing a global optimum, each particle learns from its own historical experience and its neighbourhood's historical experience. Then, a particle while choosing the neighbourhood's best experience uses the best experience of the whole swarm (m particles) as its neighbour's best experience. Since the position of each particle in a swarm is affected by the best-fit particle. Thus, this version is named, global PSO [51]. The following steps explain the learning strategy of the GPSO algorithm.

Procedure of the GPSO algorithm

Let us consider the m particles ($m > 1$) in a swarm refer to a swarm population. They are searching for a global optimum, i.e., minimum solution, of an objective function $f(x)$ in a d-dimensional search space. Total number of iterations is denoted by N_{iter}. The purpose is to minimize the given objective function $f(x)$. Each particle, i ($i = 1, 2, ..., m$), has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i and are denoted by

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{21}$$

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{22}$$

```
Step #1: Initialization: Iteration, $t = 0$.
    for $i = 1, 2, \ldots, m$
        Initialize $V_i$ and $X_i$ randomly with a defined range of $d$-dimensional search
        space and denote these by $V_i(0)$ and $X_i(0)$, respectively.
        Initialize the personal position vector ($G_{pers}$) of particle $i$, $i = 1, 2, ..., m$,
        $G_{pers,i}(0)$ as follows.
        $G_{pers,i}(0) = [g_{pi,1}, g_{pi,2}, \ldots, g_{pi,d}] = X_i(0)$          (23)
        Evaluate the $f(x)$ using $X_i(0)$.
    end $i$ loop
    Determine the global best position vector $G_{best}(0)$. It is the best position vector
    among all the $m$ personal position vectors inside a swarm. The $G_{best}(0)$ is
    denoted by:
    $G_{best}(0) = [g_{b,1}, g_{b,2}, \ldots, g_{b,d}]$          (24)
Step #2 Update
    for $t = 1, 2, \ldots, N_{iter}$
        for $i = 1, 2, \ldots, m$
```

```
                Update $V_i$ and $X_i$ as follows.
                $V_i(t) = V_i(t-1) + c_1 r_1(t)\,[G_{pers,i}(t-1) - X_i(t-1)] + c_2 r_2(t)\,[G_{best}(t-1) - X_i(t-1)]$   (25)
                $X_i(t) = X_i(t-1) + V_i(t)$   (26)
                where $c_1$ and $c_2$ are real and positive coefficients, called acceleration constants which are commonly set to 2.0 [51]. The $r_1(t)$ and $r_2(t)$ are two randomly generated values with a uniform distribution in the range of [0,1].
                Evaluate $f(x)$ for particle $i$ using $X_i(t)$.
                Update $G_{pers,i}(t)$ as follows.
                $G_{pers,i}(t) = \begin{cases} X_i(t) & if \quad f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & Otherwise \end{cases}$   (27)
                Obtain $f(G_{pers,i}(t))$
            end $i$ loop
        Obtain $f(G_{best}(t))$ as follows.
        $f(G_{best}(t)) = min\{f(G_{pers,i}(t))\}$   (28)
        Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
    end $t$ loop
Step #3: End of iteration: $t = N_{iter}$
    Optimum solution = $G_{best}(N_{iter})$ and optimum value = $f(G_{best}(N_{iter}))$   (29)
```

A flowchart of the GPSO algorithm is shown in Figure 3.3.

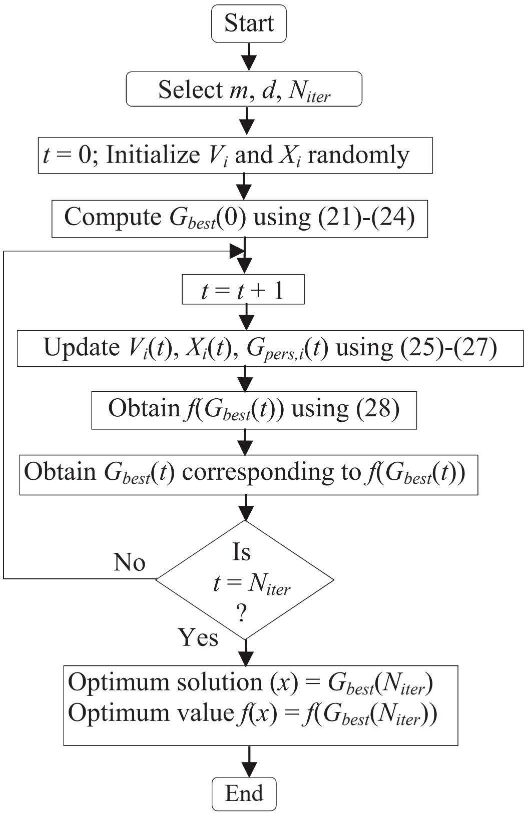

Start
Select m, d, N_{iter}
$t = 0$; Initialize V_i and X_i randomly
Compute $G_{best}(0)$ using (21)-(24)
$t = t + 1$
Update $V_i(t)$, $X_i(t)$, $G_{pers,i}(t)$ using (25)-(27)
Obtain $f(G_{best}(t))$ using (28)
Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
No
Is $t = N_{iter}$?
Yes
Optimum solution $(x) = G_{best}(N_{iter})$
Optimum value $f(x) = f(G_{best}(N_{iter}))$
End

Figure 3.3. Flowchart of the GPSO algorithm.

3.3.2 Global particle swarm optimization with inertia weight

One significant operator used in the GPSO algorithm is the inertia weight (w) [116]. The w uses to accelerate the convergence speed of ith particle (i = 1, 2, …, m) in a swarm to achieve better convergence through a balance between global search and local search particles. It is suggested that the value of w to change from 0.9 at the beginning search to 0.4 at the end of the search [117-119].

It has been shown that the GPSO-w algorithm has performed well on smooth and convex unimodal problems [4], [78], [120]-[121]. However, under high-dimensional complex unimodal and multimodal problems, the GPSO-w algorithm may suffer from the curse of dimensionality and may not perform well [112]. The following steps explain the learning strategy of the GPSO-w algorithm.

Procedure of the GPSO-*w* algorithm

In GPSO-w algorithm, the m particles (m > 1) are searching for a global minimum, i.e., searching for an optimum solution of an objective function $f(x)$ in a d-dimensional space. The objective is to minimize the given $f(x)$. Each particle i (i = 1, 2, ..., m), has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i , as given by (21) and (22).

Step #1: Same as Step #1 in Procedure of the GPSO algorithm.

Step #2: **Update**

```
**for** $t = 1, 2, \ldots, N_{iter}$
    **for** $i = 1, 2, \ldots, m$
        Determine inertia weight, $w(t)$ as given below.
```

$$w(t) = -\frac{0.5}{N_{iter}} t + 0.9 \tag{30}$$

Update V_i and X_i as follows.

$$V_i(t) = w(t)\, V_i(t-1) + c_1 r_1(t)\, [G_{pers,i}(t-1) - X_i(t-1)] + c_2 r_2(t)\, [G_{best}(t-1) - X_i(t-1)] \tag{31}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{32}$$

Evaluate $f(x)$ for particle i using $X_i(t)$.

Update $G_{pers,i}(t)$ as follows.

```
            $G_{pers,i}(t) = \begin{cases} X_i(t) & \text{if} \quad f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & \text{Otherwise} \end{cases}$        (33)
            Obtain $f(G_{pers,i}(t))$
        end $i$ loop
        Obtain $f(G_{best}(t))$ as follows.
        $f(G_{best}(t))= min\{f(G_{pers,i}(t))\}$        (34)
        Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
    end $t$ loop
```

Step #3: End of iteration: $t = N_{iter}$

Optimum solution = $G_{best}(N_{iter})$ and optimum value = $f(G_{best}(N_{iter}))$ (35)

A flowchart of the GPSO-*w* algorithm is shown in Figure 3.4.

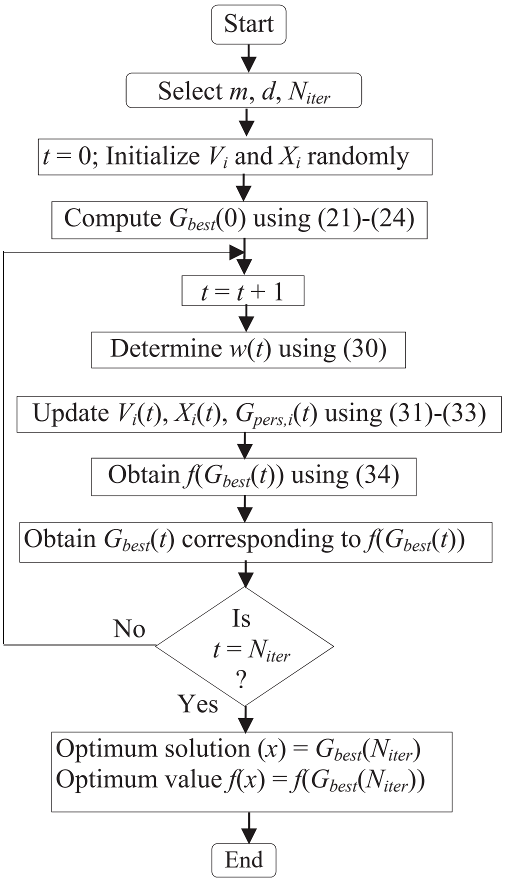

Start
Select m, d, N_{iter}
t = 0; Initialize V_i and X_i randomly
Compute $G_{best}(0)$ using (21)-(24)
$t = t + 1$
Determine $w(t)$ using (30)
Update $V_i(t)$, $X_i(t)$, $G_{pers,i}(t)$ using (31)-(33)
Obtain $f(G_{best}(t))$ using (34)
Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
Is $t = N_{iter}$?
No
Yes
Optimum solution (x) = $G_{best}(N_{iter})$
Optimum value $f(x) = f(G_{best}(N_{iter}))$
End

Figure 3.4. Flowchart of the GPSO-*w* algorithm.

3.3.3 Demerits of the PSO variants

The PSO variants, i.e., GPSO and GPSO-*w* algorithms achive satisfactory performance in some optimization problems [4], [78], [120]-[121]. However, they are inefficient in solving complex real-world practical ED problem of large-scale TGUs with practical operating power constraints as shown in Papers A-H (Appendix-1). The main reasons that make the performance of the original PSO variants unsatisfactory are explained below.

3.3.3.1 *Oscillation or zigzagging phenomenon*

In both GPSO and GPSO-*w* algorithms, the *i*th particle ($i = 1, 2, \ldots, m$) updates its flying velocity V_i and position X_i according to its personal position $G_{pers,i}$ and its neighbourhood best position G_{best}. Thus, search direction of the *i*th particle is updated based on these two guides $G_{pers,i}$ and G_{best} through a simple way, i.e., through a linear summation. This strategy can cause a phenomenon called "oscillation or zigzagging" [111]. The "oscillation or zigzagging" phenomenon is likely to be caused by the movement of $G_{pers,i}$ and G_{best} through a linear summation, as shown in (25) and (31).

To explain this phenomenon, let us consider that *i*th X_i position vector, is lying between $G_{pers,i}$ and G_{best}, as shown in Figure 3.5. At first, when $|G_{pers,i} - X_i| < |G_{best} - X_i|$, as shown in Figure 3.6(a), the X_i will move towards G_{best} because of its larger pull. Whereas, when $|G_{pers,i} - X_i| > |G_{best} - X_i|$, as shown in Figure 3.6(b), the distance between $G_{pers,i}$ and X_i will increase and then the X_i flying toward $G_{pers,i}$. Subsequently, the oscillation or zigzagging would occur. This phenomenon causes inefficiency to the search ability of the *m* particles and delays in convergence.

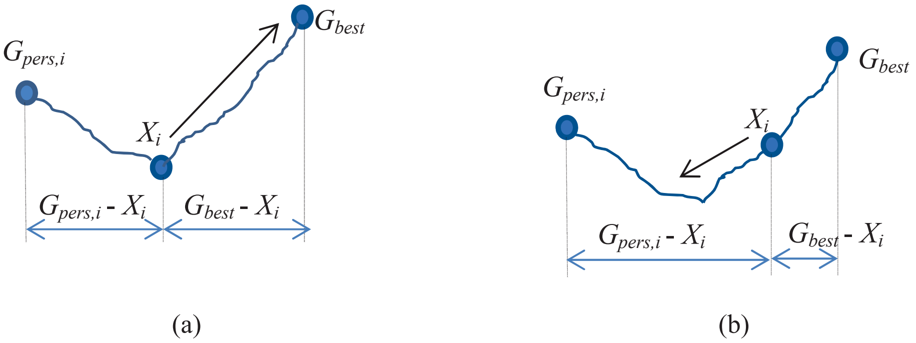


Figure 3.5. Oscillation phenomenon in both GPSO and GPSO-*w* algorithms. (a) The X_i flying toward $G_{best.}$ (b) The X_i flying toward $G_{pers,i}$.

3.3.3.2 *Imbalance between exploration and exploitation search*

In GPSO and GPSO-w algorithms, the balance between global search, i.e., *Exploration*, and local search, i.e., *Exploitation*, is maintained when solving high-dimensions complex problems [111]. In case of the GPSO algorithm, the m particles have weak local search ability (*Exploitation*). Also, the m particles are affected by the G_{best} particle. Furthermore, the m particles need to accelerate the convergence speed to achieve better balance, as shown in Paper E. In case of the GPSO-w algorithm, the inertia weight w is used to improve the balance between *Exploration* and *Exploitation*. However, the GPSO-w algorithm is more prone to encounter premature convergence when solving the ED problem of large-scale PGSs with practical power constraints as shown in Papers F and H.

In both GPSO and GPSO-w algorithms, if the G_{best} falls into a local optimum, then it would mislead the other particles in a swarm to move towards that point. This means that other promising search areas might be missed. Also, the m particles in both algorithms essentially follow a trajectory defined by V_i ($i = 1, 2, \ldots, m$) with two guides G_{pers} and G_{best} (25) and (31). Such type of search restricts the search domain of ith particle and may weaken the *Exploration* ability of the m particles, particularly, at the later stage of the search process [112]. Thus, these drawbacks make the performance of the GPSO and GPSO-w algorithms inefficient for solving the complex real-world problems, e.g., the ED problem of large-scale TGUs with power constraints.

3.3.4 The proposed orthogonal particle swarm optimization algorithm

The GPSO and GPSO-w algorithms have several drawbacks as mentioned in Section 3.3.3. These demerits make the original PSO variants inefficient to solve the complex real-world problems including the practical ED problem. Here, the first novel algorithm named orthogonal PSO (OPSO) algorithm is proposed with a new learning strategy to improve the performance of GPSO algorithm and to solve the practical ED problem of TGUs with several power constraints. The details of the proposed OPSO algorithm and explanation of the orthogonal diagonalization process are provided below.

3.3.4.1 *Orthogonal diagonalization process*

The proposed OPSO algorithm is based on an orthogonal diagonalization (OD) process. In this process, a diagonal matrix, D is obtained by multiplication of three matrices (39). Subsequently, it is applied in updating of the velocity and position vectors of d best particles in a swarm. The updating is carried out in such way that the ith velocity and position vectors are affected by only the diagonal element, d_{ii} ($i = 1, 2, \ldots, d$) of matrix D where d is dimension of the search space. This process enhances the convergence and provides a better solution as shown in Papers A, B, C, E and F.

The matrix diagonalization is the process of converting a square matrix, B of size ($d\times d$), into a diagonal matrix, D of size ($d\times d$), as shown below [122]:

$$B = Q\,D\,Q^{-1} \tag{36}$$

where Q is a matrix of size ($d\times d$) composed of eigenvectors of matrix B and the diagonal elements of matrix D comprises the corresponding eigenvalues. The matrix Q is an invertible because it contains linearly independent vectors. When matrix B is symmetric, the (36) may be written as

$$B = C\,D\,C^{-1} \tag{37}$$

in which the columns of matrix C are orthonormal to each other. Therefore (37) can be rewritten as

$$D = C^{-1}\,B\,C \tag{38}$$

Since matrix C is an orthonormal matrix, (38) can be written as

$$D = C^{\mathrm{T}}\,B\,C \tag{39}$$

Equation (39) is called the OD process. The process of OD is shown in Figure 3.6.

Line #1: Let B be a real symmetric matrix of size ($d\times d$).

Line #2: Apply Gram-Schmidt orthogonalization on matrix B to obtain d orthonormal vectors.

Line #3: Construct orthonormal matrix C using these vectors.

Line #4: Obtain the diagonal matrix D using (39).

Figure 3.6. The orthogonal diagonalization process.

3.3.4.2 *OPSO learning algorithm*

In this thesis, the proposed OPSO algorithm is used to improve the learning strategy of the GPSO algorithm and in the same time to solve the complex problems including ED problem. The objective of the OPSO algorithm is to minimize the given *d*-dimensional objective function *f*(*x*).

The OPSO algorithm provides a new topology structure in a swarm population. Consider a swarm population with *m* particles, each particle with a dimension of *d* ($m > d$). In each iteration, the *m* particles are divided into two groups based on the OD process, as follows, an active group that involves of best personal experiences of *d* particles and one passive group which comprises of the personal experiences of remaining ($m - d$) particles. The ideas of the active group particles are honoured by updating their respective velocity and position vectors. Whereas, the opinion of the passive group particles are ignored because their guidance may be erratic or insignificant, and therefore, their velocity and position vectors are not updated. However, the contributions of all the *m* particles in both groups are taken into account while determining the best experience of the swarm. In each iteration, the matrix *B* is obtained from *d* best particles of the active group, and then, orthonormal matrix *C* and diagonal matrix *D* are computed using the OD process (39). The steps included in the proposed OPSO algorithm are given below.

Procedure of the OPSO algorithm

Let *f*(*x*) is the objective function to be optimized, and N_{iter} is the number of iterations.

Initialization: Iteration, $t = 0$

Step #1: Randomly initialize the velocity vector $V_i(0)$ and position vector $X_i(0)$ for each particle *i*, ($i = 1, 2, ..., m$).

Step #2: Evaluate the objective function *f*(*x*) by using position vector $X_\mathrm{i}(0)$.

Step #3: Determine the personal position vectors $G_{pers,i}(0)$ by using

$$G_{pers,i}(0) = [g_{pi,1}, g_{pi,2}, \ldots, g_{pi,d}] = X_i(0) \tag{40}$$

Update: Iteration, $t = 1, 2, ..., N_{iter.}$

Step #4: Arrange the *m* personal position vectors $G_{pers,i}$ in an ascending order based on their *f*(*x*) values. The corresponding top *d* particles constitute active group particles.

Step #5: Construct matrix A of size (m×d) such that each row in this matrix occupies one of the m personal position vectors in the same ordered sequence as in **Step 4**.

Step #6: Using pseudocode given in Figure 3.7, convert matrix A to a symmetric matrix B of size (d×d), such that matrix B is a real symmetric matrix of dimension (d×d).

Step #7: Apply the OD process shown in Figure 3.6 on matrix B to obtain a diagonal matrix D of size d×d. Let D_i (i = 1, 2, ..., d) represent the ith row of matrix D.

Step 8: Update the position and velocity vectors of the d particles of the active group, i = 1, 2, .., d, as follows.

$$V_i(t) = V_i(t-1) + c\, r(t)\, [D_i(t) - X_i(t-1)] \tag{41}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{42}$$

where c is an acceleration coefficient and is chosen by trial and error method in the range [2,2.5] and $r(t)$ is a random value within the range of [0,1].

Step #9: Determine the $G_{pers,i}(t)$ from the m particles (i = 1, 2, ..., m), as follows.

$$G_{pers,i}(t) = \begin{cases} X_i(t) & if\;\; f(X_i(t)) \le f(G_i(t-1)) \\ G_{pers,i}(t-1) & Otherwise \end{cases} \tag{43}$$

```
for $i$ = 1, 2, …, $m$
    Evaluate $f(G_{pers,i}(t))$
end $i$ loop
```

Step #10: Determine the global best position $G_{best}(t)$, as follows.

Select $G_{best}(t)$ corresponding to minimum $\{f(G_{pers,i}(t))\}$, i = 1, 2, …, m.

Evaluate $f(x)$ to determine the global best position, $G_{best}(t)$.

$$G_{best}(t) = min\{G_{pers,i}(t)\} \tag{44}$$

End of iterations, $t = N_{iter}$.

Step #11: The $G_{best}(N_{iter})$ as computed in **Step 10** provides the optimum solution and optimum value is $f(G_{best}(N_{iter}))$.

Applications of the OPSO algorithm in solving different complex objective functions and more details about it and its performance are available in Papers A, B, C, E and F. A flowchart of the OPSO algorithm is shown in Figure 3.8.

```
Procedure for converting a matrix $A(m×d)$ to a symmetric matrix $B$ $(d×d)$.

for $i = 1, 2, ..., d$
    $B(1, i) = A(1, i)$
    $B(i, 1) = A(1, i)$
end for
for $k = 2, 3, ..., d$
    for $i = 2, 3, ..., d$
        $B(k, i) = A(k, i)$
        $B1(k, i) = B(k, i)$
        $B(i, k) = B1(k, i)$
    end for
end for
```

Figure 3.7. Pseudocode for converting matrix $A(m×d)$ to a symmetric matrix $B(d×d)$.

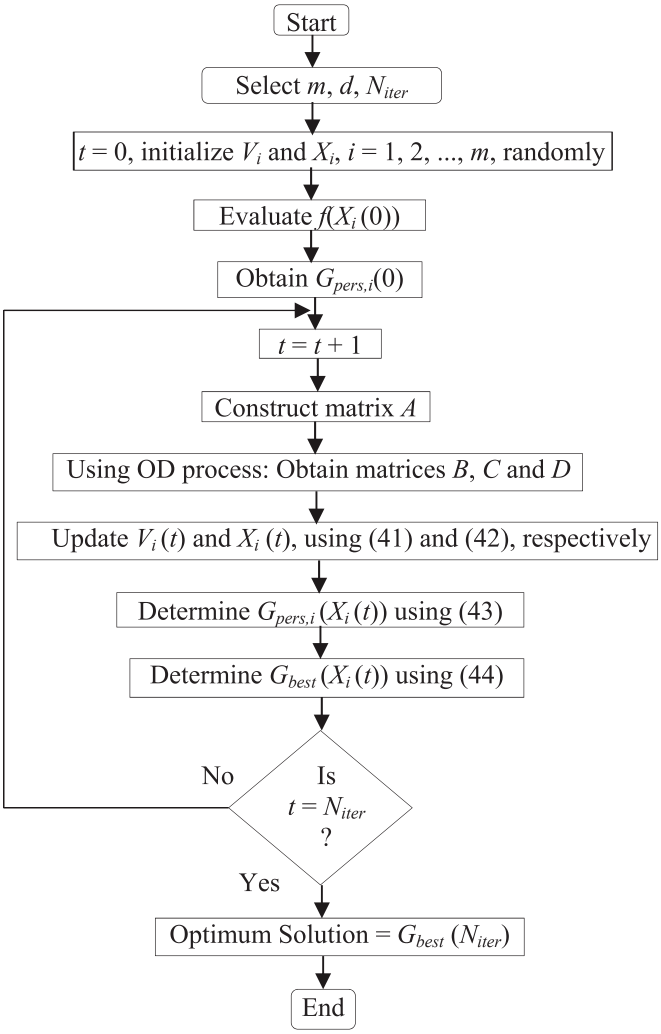

Start
Select m, d, N_{iter}
$t = 0$, initialize V_i and X_i, $i = 1, 2, ..., m$, randomly
Evaluate $f(X_i(0))$
Obtain $G_{pers,i}(0)$
$t = t + 1$
Construct matrix A
Using OD process: Obtain matrices B, C and D
Update $V_i(t)$ and $X_i(t)$, using (41) and (42), respectively
Determine $G_{pers,i}(X_i(t))$ using (43)
Determine $G_{best}(X_i(t))$ using (44)
No
Is $t = N_{iter}$?
Yes
Optimum Solution = $G_{best}(N_{iter})$
End

Figure 3.8. Flowchart of the OPSO algorithm.

3.3.5 The proposed multi-gradient particle swarm optimization algorithm

As discussed in Section 3.3.3, the GPSO and GPSO-*w* algorithms have several drawbacks. These drawbacks make these both algorithms are inefficient for solving the complex real-world problems including the ED problem. Here, the second novel algorithm named multi-gradient PSO (MG-PSO) algorithm is proposed to improve the performance of the GPSO-*w* algorithm by overcoming the drawbacks. The MG-PSO algorithm is used to solve the practical ED problem of TGUs operating with several practical power constraints. Here, the details of the second proposed MG-PSO algorithm and explanation of its mechanism are provided.

3.3.5.1 *Learning strategy*

The mechanism of the MG-PSO algorithm depends on the following considerations. Let us consider a swarm population with m particles, where $m > 1$, flying in a d-dimensional space searching for a global optimum, i.e., optimum solution. Two fundamental phases, "*Exploration* and *Exploitation*" are applied by the m particles. In *Exploration* phase, a particle is named *Explorer*. In each episode, the *m Explorers* use a different negative gradient to explore the new neighbourhood in the d-dimensional search space. The *m Explorers* boost the global search ability of the MG-PSO algorithm by using several episodes. The *m Explorers* aim to obtain a new neighbourhood within the d-dimensional search space in each episode and to obtain the best neighbourhood among episodes.

In each episode, the *m Explorers* using a different negative gradient and then they obtain best position vector following its neighbourhood in a d-dimensional search space. Its neighbourhood is obtained by taking "Floor" and "Ceil" of each element of the best position vector. These operations create a new search space, i.e., best neighbourhood, within a d-dimensional search space that will be used in the *Exploitation* phase.

In *Exploitation* phase, a particle is named *Exploiter*. The *m Exploiters* use only one negative gradient which is less than that of the *Exploration* phase. The *m Exploiters* boost the local search ability of the MG-PSO algorithm. The purpose of this phase is to gain an optimum position by exploiting the *Exploiters* in the best neighbourhood obtained from the *Exploration* phase.

3.3.5.2 *The MG-PSO algorithm*

In the second proposed MG-PSO algorithm, a number of negative gradients, i.e., N_{grad}, are used by the swarm population while searching for an optimum solution. In *Exploration* phase, $N_{grad} - 1$ negative gradients are used whereas one negative gradient is used in *Exploitation* phase. In each episode, the inertia weight (w) follows one negative gradient, as iteration increases.

The number of iterations in MG-PSO algorithm is N_{iter}. The number of iterations in *Exploration* phase is given by

$$N_{iter,xplore} = \gamma \times N_{iter} \tag{45}$$

where γ is a real and positive number in a range [0,1]. The number of iterations in the *Exploitation* phase is given by:

$$N_{iter,xploit} = (1-\gamma) \times N_{iter} \tag{46}$$

The initial and final values of the w for kth negative gradient (k = 1, 2, …, N_{grad}) are denoted by $w_{ini,k}$ and $w_{fin,k}$, respectively. These values are positive and real numbers within a range [0,1] and $w_{ini,k} > w_{fin,k}$. The kth negative gradient (k = 1, 2, …, $N_{grad} - 1$) in *Exploration* phase is given by:

$$grad_k = \frac{w_{fin,k} - w_{ini,k}}{N_{iter,xplre}} \tag{47}$$

In *Exploitation* phase, the negative gradient is given by:

$$grad_{N_{grad}} = \frac{w_{fin,N_{grad}} - w_{ini,N_{grad}}}{N_{iter,xploit}} \tag{48}$$

The N_{grad} gradients are selected such that (49) is satisfied.

$$\left|grad_1\right| > \left|grad_2\right| \cdots > \left|grad_{N_{grad}}\right| \tag{49}$$

The w for kth negative gradient (k = 1, 2, …, N_{grad}) at iteration t is given by:

$$w_k(t) = grad_k \times t + w_{ini,k} \tag{50}$$

3.3.5.3 *Procedure of MG-PSO algorithm*

Here, the detailed steps explaining the procedure of MG-PSO algorithm are provided.

Procedure of the MG-PSO algorithm

```
            Begin MG-PSO Algorithm
            Let $f(x)$ be the function to be minimized.
            Choose $N_{iter}$, $N_{grad}$, $w_{ini,k}$, $w_{fin,k}$, $k = 1, 2, \ldots, N_{grad.}$
            Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (45) and (46), respectively.
            Initialization: Iteration, $t = 0$.
Step #1:    Obtain $G_{best}(0)$ using (21)-(24).
Step #2:    Begin Exploration phase
              for $k = 1, 2, ..., N_{grad} - 1$ (begin of episode $k$).
                Determine $grad_k$ using (47).
                for $t = 1, 2, \ldots, N_{iter,xplore}$
                  Determine $w_k(t)$ using (50).
                  for $i = 1, 2, \ldots, m$
                    Update the particle's velocity and position vectors as follows.
                    $V_i^k(t) = w_k(t) V_i^k(t-1) + c_1 r_1(t)[G_{pers,i}^k(t-1) - X_i^k(t-1)] + c_2 r_2(t)[G_{best}^k(t-1) - X_i^k(t-1)]$    (51)
                    $X_i^k(t) = X_i^k(t-1) + V_i^k(t)$    (52)
                    Evaluate the particle's performance by substituting (52) in $f(x)$.
                    Update $G_{pers,i}$ as follows.
                    $G_{pers,i}^k(t) = \begin{cases} X_i^k(t) & if \quad f(X_i^k(t)) \le f(G_{pers,i}^k(t-1)) \\ G_{pers,i}^k(t-1) & Otherwise \end{cases}$    (53)
                    Obtain $f(G_{pers,i}^k(t))$
                  end $i$ loop
                  Obtain $f(G_{best}^k(t))$ as follows.
                  $f(G_{best}^k(t)) = min\{ f(G_{pers,i}^k(t)) \}$    (54)
                  Obtain $G_{best}^k(t)$ corresponding to $f(G_{best}^k(t))$
                end $t$ loop
```

```
                    Obtain $G_{best}^{k}(N_{iter,xplore})$ and $f(G_{best}^{k}(N_{iter,xplore}))$
                  end $k$ loop (end of episode $k$)
                  for $k = 1, 2, \ldots, N_{grad}-1$
                    Obtain $f(G_{best}^{k}(N_{iter,xplore}))$
                  end $k$ loop
                $f(G_{best,xplore}) = min\{\, f(G_{best}^{k}(N_{iter,xplore}))\,\}$                    (55)
                Obtain $BEST(G_{best,xplore})$ corresponding to $f(G_{best,xplore})$
                Obtain new search space (neighbourhood) by taking “Floor” and “Ceil” of
                each element of $BEST(G_{best,xplore})$
                End Exploration phase

                Begin Exploitation phase
                Use the new search space
Step #3:        Initialization: Iteration, $t = 1$
                  for $i = 1, 2, \ldots, m$
                    $V_i(1) = V_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$          (56)
                    $X_i(1) = X_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$          (57)
                    $G_{pers,i}(1) = G_{pers,i}(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$   (58)
                  end $i$ loop
                  $G_{best,xploit}(1) = BEST(G_{best,xplore})$                                       (59)
                  Determine $grad_{N_{grad}}$ using (48)
Step #4:          Update
                  for $t = 2, 3, \ldots, N_{iter,xploit}$
                    Determine $w_k(t)$ using (50)
                      for $i = 1, 2, \ldots, m$
                        Update the particle’s velocity and position vectors as follows.
                        $V_i(t) = w_{N_{grad}}(t)\, V_i(t-1) + c_1 r_1(t)\,[G_{pers,i}(t-1) - X_i(t-1)]$
                                    $+\, c_2 r_2(t)\,[G_{best,xploit}(t-1) - X_i(t-1)]$    (60)
                        $X_i(t) = X_i(t-1) + V_i(t)$                                           (61)
                        Evaluate the particle’s performance by substituting (61) in $f(x)$
                        Update $G_{pers,i}(t)$ as follows.
```

```
            $G_{pers,i}(t) = \begin{cases} X_i(t) & if\ \ f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & Otherwise \end{cases}$      (62)
            Obtain $f(G_{pers,i}(t))$
        end $i$ loop
        Obtain $f(G_{best,xolit}(t))$ as follows.
        $f(G_{best,xolit}(t)) = min\{ f(G_{pers,i}(t))\}$      (63)
        Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
    end $t$ loop
    Optimum solution = $G_{best,xploit}(N_{iter,xploit})$
    Optimum value = $f(G_{best,xploit}(N_{iter,xploit}))$
    End of Exploration phase
End of MG-PSO algorithm
```

More details about the learning strategy of the MG-PSO algorithm with its performance and applications are available in Papers D, G and H. A flowchart of the MG-PSO algorithm is shown in Figure 3.9.

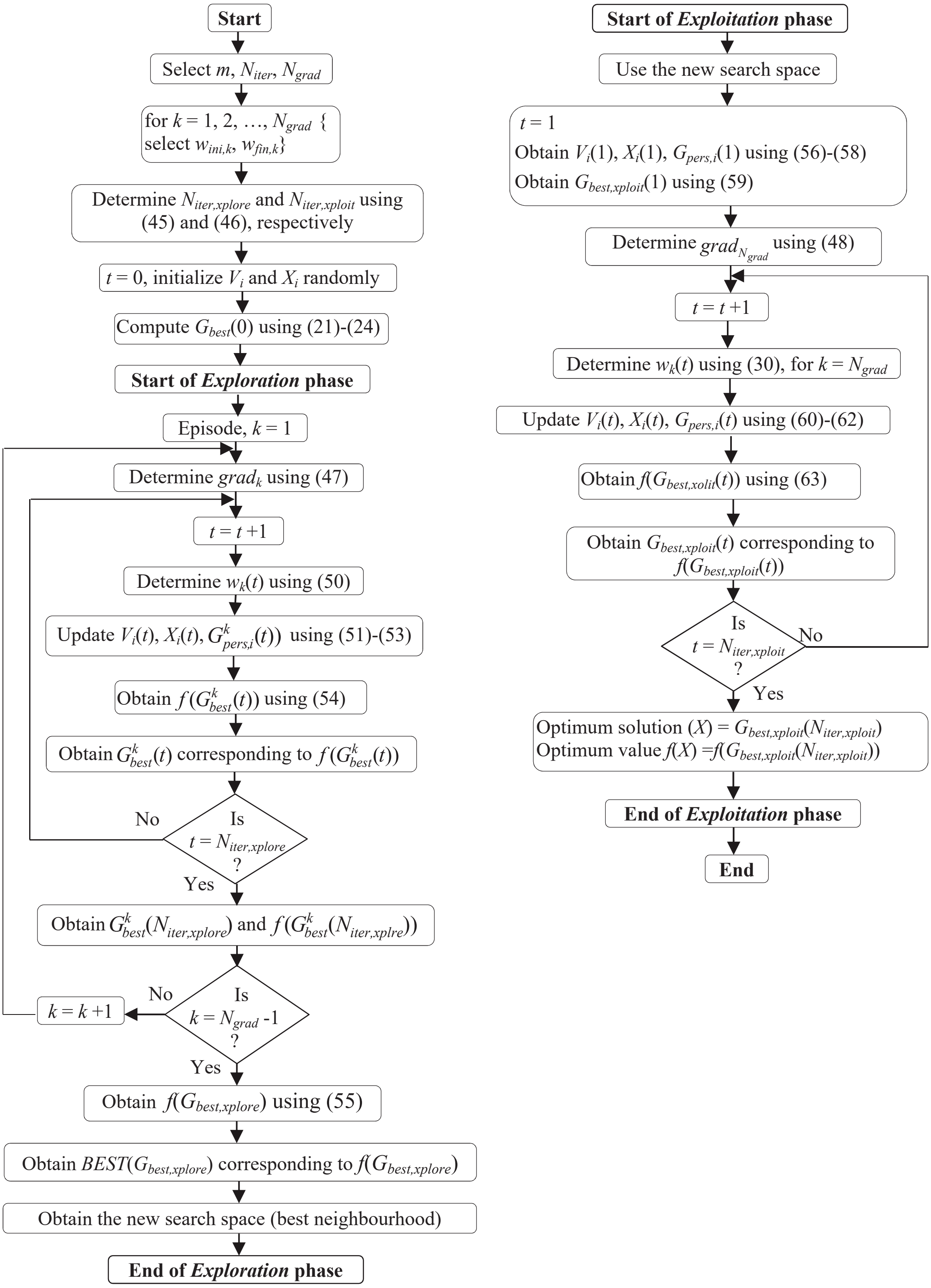

Start
Select m, N_{iter}, N_{grad}
for k = 1, 2, …, N_{grad} { select $w_{ini,k}$, $w_{fin,k}$}
Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (45) and (46), respectively
t = 0, initialize V_i and X_i randomly
Compute $G_{best}(0)$ using (21)-(24)
Start of *Exploration* phase
Episode, k = 1
Determine $grad_k$ using (47)
$t = t + 1$
Determine $w_k(t)$ using (50)
Update $V_i(t)$, $X_i(t)$, $G^k_{pers,i}(t)$) using (51)-(53)
Obtain $f(G^k_{best}(t))$ using (54)
Obtain $G^k_{best}(t)$ corresponding to $f(G^k_{best}(t))$
Is $t = N_{iter,xplore}$?
No
Yes
Obtain $G^k_{best}(N_{iter,xplore})$ and $f(G^k_{best}(N_{iter,xplre}))$
Is $k = N_{grad}$ -1 ?
No
$k = k + 1$
Yes
Obtain $f(G_{best,xplore})$ using (55)
Obtain $BEST(G_{best,xplore})$ corresponding to $f(G_{best,xplore})$
Obtain the new search space (best neighbourhood)
End of *Exploration* phase
Start of *Exploitation* phase
Use the new search space
t = 1
Obtain $V_i(1)$, $X_i(1)$, $G_{pers,i}(1)$ using (56)-(58)
Obtain $G_{best,xploit}(1)$ using (59)
Determine $grad_{N_{grad}}$ using (48)
$t = t + 1$
Determine $w_k(t)$ using (30), for $k = N_{grad}$
Update $V_i(t)$, $X_i(t)$, $G_{pers,i}(t)$ using (60)-(62)
Obtain $f(G_{best,xolit}(t))$ using (63)
Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
Is $t = N_{iter,xploit}$?
No
Yes
Optimum solution $(X) = G_{best,xploit}(N_{iter,xploit})$
Optimum value $f(X) = f(G_{best,xploit}(N_{iter,xploit}))$
End of *Exploitation* phase
End

Figure 3.9. Flowchart of the MG-PSO algorithm.

3.4 Comparison between OPSO and GPSO algorithms

A comparison between the proposed OPSO and GPSO algorithms in terms of several critical parameters is provided in Table 3.1. These parameters are important roles in determining the accuracy, consistency and reliability of the OPSO and GPSO algorithms in solving complex problems including ED problem.

Table 3.1. Comparison between the proposed OPSO and GPSO algorithms.

Parameters	OPSO algorithm	GPSO algorithm
Swarm Size	m = number of particles d = number of dimensions $m > d,$ as shown in Paper E.	m = number of particles d = number of dimensions $m > 1$, as shown in Paper E.
Neighbourhood topology structure	Orthogonal vectors are applied by the active group particles through using the OD process, as shown in Paper E.	The m particles use a fully connected network, as shown in Paper F.
Guidance	Only one guide is used by the OPSO algorithm, i.e., $D_i(t)$, as shown in Papers A, B, C, E and F.	In case of GPSO algorithm, two guides, i.e., $G_{pers,i}$ and G_{best} are used, as shown in Paper E.
Computational Complexity	When m becomes large, this increases the computational complexity in each iteration. Because the OPSO uses three matrices in its learning strategy, as shown in Papers E and F.	The GPSO algorithm is simple in sturcture. The computational complexity is less restrictive than that in the OPSO algorithm, as shown in Papers E and F.
Performance	The OPSO is a high performance in solving several unimodal and multimodal benchmark functions including the ED problem with high dimensional search space, as shown in Papers E and F.	The performance of GPSO algorithm is deteriorated in solving several unimodal and multimodal benchmark functions including the ED problem with high dimensional search space, as shown in Papers E and F.
Exploration and Exploitation Processes	Due to the use of the OD process, the orthogonal vectors of active group particles are capable of making a balance between local search and global search of d best particles, as shown in Papers E and F.	Due to the conflict between the two guides G_{pers} and G_{best}, the GPSO algorithm is weak in the exploration processes, as shown in Papers E and F.
Algorithm Execution Time	The OPSO algorithm consumes more execution time than that in the GPSO algorithm, as shown in Papers A, B, C, E and F.	The GPSO algorithm is fast in convergence and consumes short execution time, as shown in Papers A, B, C, E and F.

3.5 Comparison between MG-PSO and GPSO-w algorithms

A comparison between the proposed MG-PSO and GPSO-*w* algorithms in terms of several critical parameters is provided in Table 3.2. These parameters are important roles in determining the accuracy, consistency and reliability of the OPSO and GPSO-*w* algorithms in solving complex problems including ED problem.

Table 3.2. Comparison between the proposed MG-PSO and GPSO-*w* algorithms.

Parameters	MG-PSO algorithm	GPSO-*w* algorithm
Swarm Size	*m* = number of particles *d* = number of dimensions $m > 1$, $m =\sim 20$, as shown in Paper G and H.	*m* = number of particles *d* = number of dimensions $m > 1$, as shown in Paper E.
Neighborhood Topology Structure	the *m* particles use multiple episodes with different negative gradients as topology structure, as shown in Papers G and H.	The *m* particles use a fully connected network, as shown in Paper G and H.
Guidance	*Exploration* phase and *Exploitation* phase are used as guidance in the MG-PSO algorithm, as shown in Papers G and H.	In case of GPSO-*w* algorithm, two guides, i.e., $G_{pers,i}$ and G_{best} are used, as shown in Paper G and H.
Computational Complexity	The MG-PSO algorithm gives rise to more computations in case of using large number of episodes in the *Exploration* phase, as shown in Papers G and H.	The GPSO algorithm is simple in structure. The computational complexity is less restrictive than that in the MG-PSO algorithm, as shown in as shown in Papers G and H.
Performance	The MG-PSO algorithm is a high performance in solving several unimodal and multimodal benchmark functions including the ED problem with high dimensional search space, as shown in Papers G and H.	The performance of GPSO-*w* algorithm is deteriorated in solving several unimodal and multimodal benchmark functions including the ED problem with high dimensional search space, as shown in Papers E and F.
Exploration and Exploitation Processes	The combination between *Exploration* phase and *Exploitation* phase provides a balance in the exploration and exploitation processes, as shown in Paper G.	Due to the conflict between the two guides G_{pers} and G_{best}, the GPSO-*w* algorithm is weak in the exploration processes, as shown in Papers G and H.
Algorithm Execution Time	The MG-PSO algorithm consumes more execution time than that in the GPSO-*w* algorithm, as shown in Papers G and H.	The GPSO-*w* algorithm is fast in convergence and consumes short execution time, as shown in Papers G and H.

3.6 Comparison between OPSO and MG-PSO algorithms

A comparison between the two proposed algorithms OPSO and MG-PSO in terms of several critical parameters is provided in Table 3.3. Both algorithms have been successfully applied in solving several complex benchmark functions including ED problem of large-scale TGUs as shown in the Papers A-H, several merits and demerits between OPSO and MG-PASO algorithms are presented in Table 3.3.

Table 3.3. Comparison between the two proposed OPSO and MG-PSO algorithms.

Parameters	OPSO algorithm	MG-PSO algorithm
Swarm Size	m = number of particles. d = number of dimensions. $m > d$, as shown in Paper E.	m = number of particles. d = number of dimensions. $m > 1$, $m =\sim 20$, as shown in Paper G and H.
Neighborhood Topology Structure	Orthogonal vectors are applied by the active group particles through using the OD process, as shown in Paper E.	the m particles use multiple episodes with different negative gradients as topology structure, as shown in Papers G and H.
Guidance	Only one guide is used by the OPSO algorithm, i.e., $D_i(t)$, as shown in Papers A, B, C, E and F.	*Exploration* phase and *Exploitation* phase are used as guidance in the MG-PSO algorithm, as shown in Papers G and H.
Computational Complexity	When m becomes large, this increases the computational complexity in each iteration. Because the OPSO uses three matrices in its structure, as shown in Papers E and F.	The MG-PSO algorithm gives rise to more computations in case of using large number of episodes in the *Exploration* phase, as shown in Papers G and H.
Performance	The OPSO is a high performance in solving several unimodal and multimodal benchmark functions including the ED problem with high dimensional search space, as shown in Papers E and F.	The MG-PSO algorithm is a high performance in solving several unimodal and multimodal benchmark functions including ED problem with high dimensional search space, as shown in Papers G and H.
Exploration and Exploitation Processes	Due to the use of the OD process, the orthogonal vectors of active group particles are capable of making a balance between local search and global search of d best particles, as shown in Papers E and F.	The combination between *Exploration* phase and *Exploitation* phase provides a balance in the exploration and exploitation processes, as shown in Papers G and H.
Algorithm Execution Time	The OPSO algorithm consumes more execution time than that in the MG-PSO algorithm, as shown in Paper G and H.	The MG-PSO algorithm consumes less execution time than that in the OPSO algorithm, as shown in Papers G and H.

3.7 Chapter summary

In this Chapter, the research methodology based on the research design of this thesis was demonstrated. In the research design, the question #1 that mentioned in Section 1.4 has been investigated. In addition, the question #2 has been investigated using two proposed algorithms, i.e., orthogonal PSO (OPSO) and multi-gradient PSO (MG-PSO) algorithms. The original PSO variants, i.e., global PSO (GPSO) and global PSO with inertia weight (GPSO-*w*) were also studied. Performance comparison among these algorithms in terms of several critical parameters was also carried out.

Chapter 4: Summary of Papers

4.1 Chapter overview

In this Chapter, a brief summary of each of the eight Papers is presented. The purpose of this Chapter is to give sufficient details of each associated Paper appended in this thesis, the Papers A-H. At the end, a summary of this Chapter is provided. The sequence of the Papers A-H has been organized based on their date of publication.

4.2 Summary of Papers A-H

Here, a summary of the eight appended Papers A-H is provided. The salient features and important results are highlighted.

4.2.1 Summary of Paper A

L. T. Al-Bahrani and J. C. Patra, "Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications," in Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-7.

In Paper A, few operating power constraints are examined, e.g., generation limits, ramp rate limits (RRLs), prohibited operating zones (POZs) and power balance. When these power constraints are imposed, the fuel cost function becomes non-convex and non-smooth. In such a case, the ED problem becomes a multimodal problem.

In Paper A, a novel algorithm called orthogonal particle swarm optimization (OPSO) algorithm was proposed in 2015 to solve such a complex problem. The OPSO algorithm depends on the formation of orthogonal particle vectors that are found in the *d*-dimensional searching search. The *d* best particles construct a new guide and fly more steadily toward the optimum solution. This is accomplished by determining the promising movements of the *d* best particles in subsequent iterations based on orthogonality. Due to the use of orthogonal vectors in updating the velocity and position vectors, the OPSO algorithm provides substantial improvement in performance over PSO algorithm (the PSO algorithm in Paper A is GPSO algorithm).

Let us consider a 2-dimensional function given by $f(x, y) = x^2+y^2+3$. The objective is to find values x and y, so that the value of the $f(x, y)$ is minimized. With a range of [-100, 100] for x and y, the $f(x, y)$ is plotted in Figure 4.1. Note that the function $f(x, y)$ is minimized to 3 when $x = 0$ and $y = 0$. Both the OPSO and PSO algorithms have ten particles (m =10) in the swarm and use ten iterations (N_{iter} = 10). Figure 4.2 shows that the best particle in PSO algorithm moves step forward and step backward between 3rd iteration (-44.3, 31.6) and 4th iteration (16, -10.2) causing oscillations. Whereas, in the OPSO algorithm, the best particle moves steadily from the initial position to the solution in the 10th iteration.

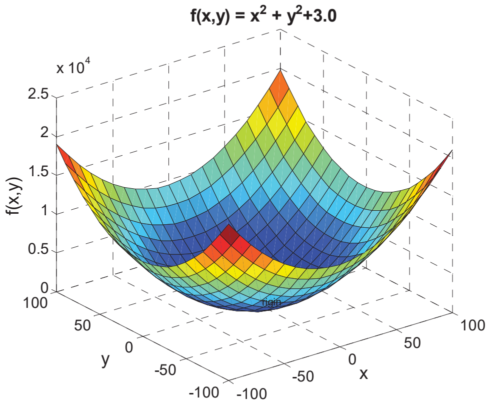


Figure 4.1. Plot of 2-dimentional function $f(x,y)$.

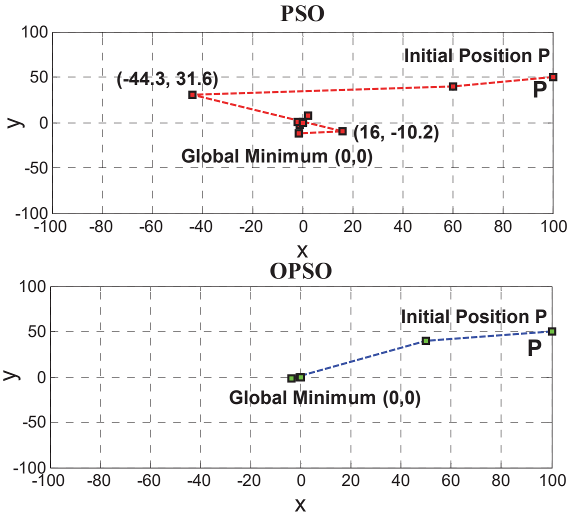


Figure 4.2. Movement of the best particle in PSO and OPSO algorithms at different iterations.

The OPSO and PSO algorithms are evaluated and tested using small PGS, i.e., an IEEE bus with 6 TGUs with generation limits, RRLs, POZs power constraints. The performance comparison results shown in Paper A between the OPSO algorithm and several other optimization techniques reveal that OPSO provides better performance in solving the ED problem of small-scale PGS in terms of several performance measures, e.g., minimum, maximum and mean costs and standard deviation. In addition, the OPSO algorithm is capable of solving the inequality power constraints and satisfying the equality power constraint. The results shown in Paper A indicate that the OPSO algorithm is a promising tool for solving non-convex, multimodal fuel cost function for the small-scale PGS.

4.2.2 Summary of Paper B

L. T. Al-Bahrani and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of power under power grid constraints," in Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC), Hong Kong, 2015, pp. 14-19.

In Paper B, the OPSO algorithm is proposed for ED of the generated power in medium-scale PGS (15 TGUs). Here, the equality and inequality power constraints and the power balance response against the mismatch between load demand and total power outputs of TGUs involve non-linear characteristics and non-smooth cost functions. The OPSO algorithm uses orthogonal vectors (OVs) for the *d* best particles in a *d*-dimensional search space. The OVs are generated and updated in each iteration. They are used to guide the *d* best particles to fly in one direction toward a global minimum. Also, instead of creating and updating two guides in PSO (the PSO algorithm in Paper B is GPSO algorithm), the *d* best particles update their position and its velocity according to OVs. This means that only one guide is used to update the velocity and position vectors. Thus, the OPSO algorithm has succeeded in eliminating the oscillation phenomenon occurring in the PSO algorithm.

Figure 4.3 shows the convergence characteristics of OPSO and PSO algorithms of medium-scale PGS (15 TGUs). In Figure 4.3(A), the OPSO algorithm is better convergence to reach optimum solution. The OPSO algorithm settled at about 370 iterations. Whereas, the PSO algorithm settled after 580 iterations. Figure 4.3(B) shows the distribution of total cost of OPSO and PSO algorithms over 100 runs for 15 TGUs, showing that the OPSO algorithm has a small deviation compared to the PSO algorithm. This indicates that the performance of the PSO algorithm is improved by the orthogonality.

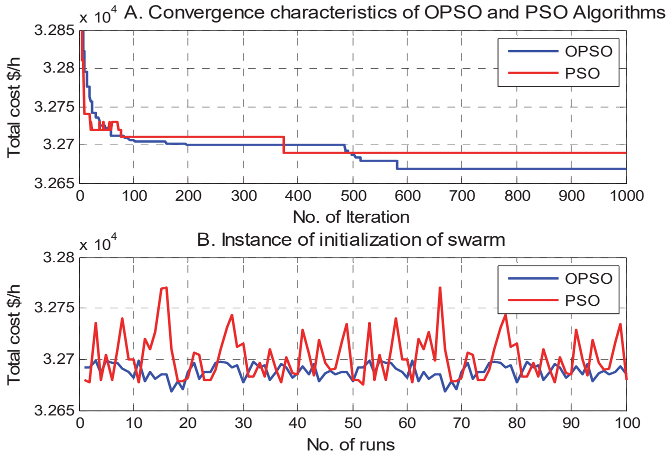


Figure 4.3. Convergence characteristics of OPSO and PSO algorithms of medium-scale PGS (15 TGUs).

The OPSO and PSO algorithms are evaluated using medium-scale PGS (IEEE 15 TGUs). In this PGS, P_D is 2,630 MW and the P_L is taken into account. In addition, there are 4 TGUs that have 11 prohibited zones. The generation limits and RRLs of 15 TGUs are also considered. Thus, more dimensions and power constraints are imposed on fuel cost function.

The performance of the OPSO algorithm is also compared with several other optimization techniques including PSO variants. These results reveal that OPSO algorithm provides better performance in solving the fuel cost function in terms minimum, maximum and mean costs and standard deviation. In addition, the OPSO algorithm is able to solve the equality and inequality power constraints and able to avoid all POZs. Furthermore, the OPSO algorithm is able to reduce the P_L. Moreover, the OPSO algorithm succeeded to improve the learning strategy of the PSO algorithm, in terms of consistency, robustness and convergence.

4.2.3 Summary of paper C

L. T. Al-Bahrani, J. C. Patra, and R. Kowalczyk, “Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints," in Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, Canada, 2016, pp. 660-667.

In Paper C, the OPSO algorithm is proposed for the ED of large-scale TGUs in smart power gird (SPG). Practically, the characteristics of TGUs are non-linear and the generation system becomes more and more complex when these large-scale TGUs are subjected to RRL and POZ constraints. In such case, the cost function becomes non-smooth, non-convex and discontinuous. Moreover, in large-scale TGUs, the high dimensions used in the ED problem become a big challenge in order to find a global minimum and to avoid being trapped into local minima. In this Paper, the proposed OPSO algorithm has the ability to solve such complex ED problem with equality and inequality power constraints and considering P_L, RRLs and POZs.

The OPSO algorithm applies the OD process and orthogonality to the d best particles in the swarm. It makes d best particles (out of total m particles, $m > d$) that have the possible solutions by constructing OVs in the d-dimensional search space. These OVs are generated and updated in each iteration and are utilized to guide the d best particles to fly in one direction toward a global minimum. The remaining ($m - d$) particles are not updated. This leads the search process primarily to concentrate on using best d best particles in a swarm.

The OPSO algorithm is evaluated and tested on the Taiwan power system. It is complex power system and consists of 40 mixed-generating units, e.g., coal-fired, gas-fired, diesel generating units and nuclear generating units. The load demand P_D is 8,550 MW. There are total 46 POZs in 25 TGUs. The transmission network loss P_L is taken into account. In addition, the generation limits and RRLs of 40 TGUs are also considered. Thus, more dimensions and power constraints are imposed on the fuel cost function.

The performance of the OPSO algorithm is compared with several other optimization techniques including original PSO (the PSO algorithm in Paper C is GPSO algorithm). The results in Paper C shows that the OPSO algorithm is achieving the best minimum, maximum and mean costs and the lowest standard deviation while comparing it with the PSO algorithm and other optimization techniques. In addition, the OPSO algorithm solves the inequality constraints in terms of generating limits, RRLs and POZs and avoids the POZs. Thus, the optimization power generation schedule fits to P_D =8,550 MW while satisfying all power constraints. In addition, the OPSO algorithm achieves on the lowest value of transmission network loss P_L compared to the other competitive algorithms. This means that the equality power balance constraint has been satisfied by the OPSO algorithm. Whereas, the PSO algorithm is unable to solve the equality power constraint of Taiwan power system. Thus, the OPSO algorithm significantly improves the PSO algorithm in terms of high solution quality, robustness and convergence rate.

Figure 4.4 shows the convergence characteristics of OPSO and PSO algorithms. Figure 4.4(A) shows essential average of the mean cost over 25 independent runs. The OPSO algorithm has better convergence characteristics than the PSO algorithm. Figure 4.4(B) shows the distribution of minimum costs over 25 independent runs. It shows that OPSO algorithm is more stable than the PSO algorithm in getting the optimum solution.

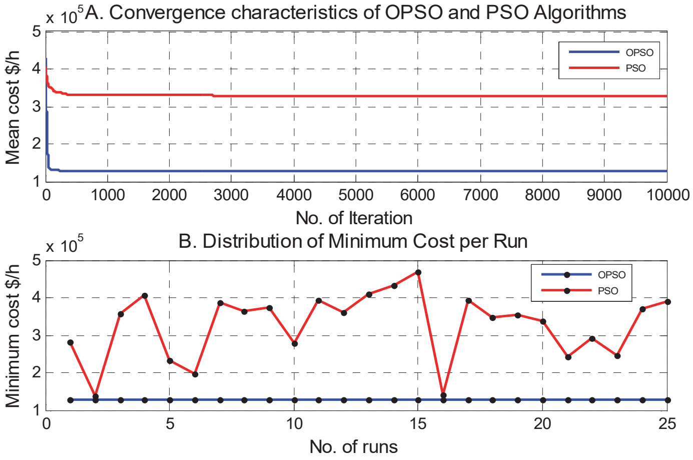


Figure 4.4. Convergence characteristics of OPSO and original PSO algorithms of Taipower system.

4.2.4 Summary of Paper D

L. T. Al-Bahrani, J. C. Patra, and R. Kowalczyk, “Multi-gradient PSO Algorithm for economic dispatch of thermal generating units in smart Grid,” in Proceedings of IEEE PES Innovative Smart Grid Technologies 2016 Asian Conference (ISGT’2016 Asia), Melbourne, Australia, 2016, pp. 258-263.

In paper D, another novel algorithm called multi-gradient particle swarm optimization (MG-PSO) is proposed for solving non-convex and non-smooth ED of TGUs with several operating power constraints, e.g., RRLs and POZs. In MG-PSO algorithm, different negative gradients are used. These negative gradients are used as guides for the *m* particles in the search for a global optimum. The key of the MG-PSO algorithm is the diversity in using negative gradients. Due to this diversity, the *m* particles cover a large search space area, as much as possible. The velocity vectors of the *m* particles are significantly affected by only one negative gradient named, the best negative gradient among all used negative gradients. This makes the *m* particles adjust their positions and improve their direction according to that the best negative gradient.

The performance of the MG-PSO algorithm has been verified on a small-scale PGS (6 TGUs) and a medium-scale PGS (15 TGUs). The proposed MG-PSO algorithm provides good quality and promising results in solving the ED problem. The MG-PSO algorithm gives better results in terms of fitness values, e.g., minimum, maximum and mean costs and has lowest standard deviation while comparing with PSO (the PSO algorithm in Paper D is GPSO algorithm) algorithm and other optimization techniques for both PGSs.

In terms of inequality and quality power constraints, the MG-PSO algorithm is able to solve the inequality constraints imposed on small-scale and medium-scale PGSs by avoiding the 12 POZs of 6 TGUs and 11 POZs of 15 TGUs. The MG-PSO algorithm operates within the RRLs of each TGU and it is able to solve the P_L for both PGSs. In addition, the MG-PSO algorithm has zero mismatch condition in solving power balance constraint for the 6 and 15 TGUs.

The MG-PSO algorithm significantly improves the PSO algorithm in terms of high solution quality, robustness and convergence rate for small-scale and medium-scale PGSs. Figure 4.5 shows the convergence characteristics of MG-PSO and PSO algorithms for small-scale PGS. Figure 4.5(A) shows average of the mean cost over 25 independent runs. The MG-PSO algorithm has better convergence characteristics than the PSO algorithm. Figure 4.5(B) shows the distribution of minimum costs over 25 independent runs. It can be seen that the MG-PSO algorithm is more stable in achieving the optimum solution than the PSO algorithm in solving the ED of small-scale PGS.

The convergence characteristics of MG-PSO and PSO algorithms are shown in Figure 4.6. Figure 4.6(A) shows average of the mean cost over 25 independent runs. The MG-PSO algorithm is better than the PSO algorithm in terms of convergence rate. The distribution of minimum costs over 25 independent runs shown in Figure 4.6(B). It can be seen that the MG-PSO algorithm is more stable in achieving the optimum solution than the PSO algorithm in solving the ED of medium-scale PGS.

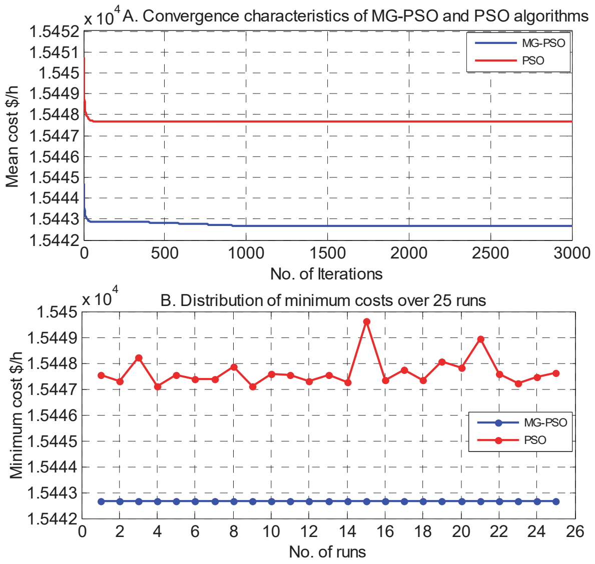


Figure 4.5. Convergence characteristics of MG-PSO and PSO algorithms for small-scale PGS (6 TGUs).

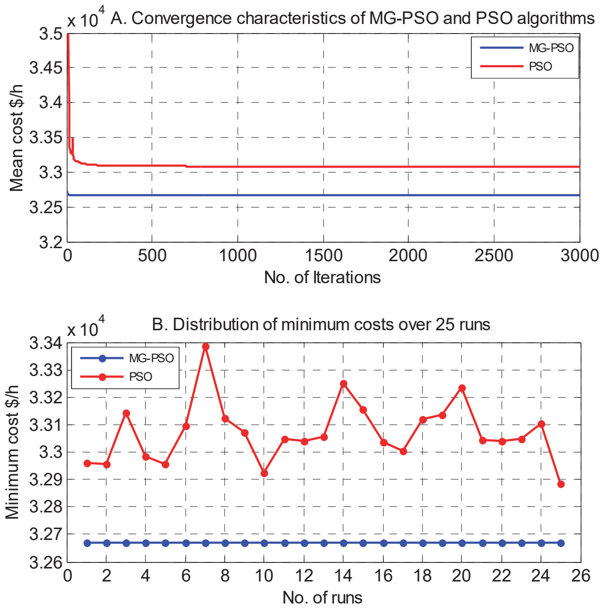


Figure 4.6. Convergence characteristics of MG-PSO and PSO algorithms for medium-scale PGS (15 TGUs).

4.2.5 Summary of Paper E

L. T. Al-Bahrani and J. C. Patra, “A novel Orthogonal PSO algorithm based on orthogonal diagonalization”, Swarm and Evolutionary Computation, vol. xxx, pp. 1-23, 2017. In press.

In Paper E, more details and mathematical justification of the OPSO algorithm are investigated. One of the prominent demerits of the GPSO algorithm is zigzagging of the direction of search that leads to premature convergence by falling into a local minimum. In this paper, the OPSO algorithm is proposed not only to overcome the associated problems in the GPSO algorithm but also achieves better performance.

The OPSO algorithm consists of a swarm with *m* particles that looks for the global optimum solution in a *d*-dimensional search space with $m > d$. In OPSO algorithm, the *m* particles in a swarm are divided into two groups: one active group of best personal experience of *d* particles and another passive group of personal experience of remaining $(m - d)$ particles. The aim of constructing two groups is to enhance the diversity of the particles in a swarm. The *d* active group particles in each iteration undergo an OD process. They are updated in such way that their position vectors are orthogonally diagonalized. The passive group particles are not updated as their contribution in finding correct direction is not significant.

The ideas of the active group particles are honoured by updating their respective velocity and position vectors. Whereas, the ideas of the passive group particles are ignored because their guidance may be insignificant or erratic, and thus, their velocity and position vectors are not updated. However, the contributions of the swarm in both groups are considered while determining the best experience. In OPSO algorithm, the particles are updated using only one guide, thus avoiding the conflict between the two guides that happens in the GPSO algorithm and leads the best *d* particles towards the optimum solution.

In Paper E, the mechanism of OPSO algorithm is explained through an example of a 2-dimensional shifted function, $f(x,y) = (x - 2)^2 + (y + 3)^2 + 9$, plotted in Figure 4.7. From visual inspection it can be seen that the *x* and *y* are shifted from the origin by

(2.0,-3.0). The optimum solution of the given function equals to 9 at (*x*,*y*) = (2.0,-3.0). The purpose of the OPSO algorithm is to find the values *x* and y such that the *f*(*x*,*y*) is minimized.

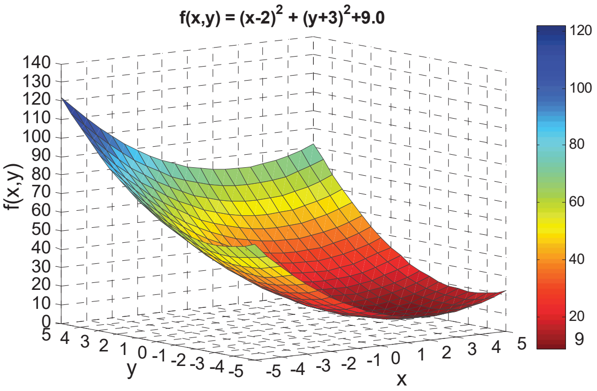


Figure 4.7. The landscape of *f*(*x*,*y*). The minimum value of the function *f*(*x*,*y*) is 9.0 at *x* = 2.0 and *y* = -3.0.

The OPSO algorithm was carried out with 6 particles (m = 6), d = 2 and N_{iter} = 200, where d is number of dimensions and N_{iter} is number of iterations. The values of position vectors (X_i, i = 1, 2, ..., 6), personal vectors ($G_{pers,i}$, i = 1, 2, ..., 6) and the diagonal vectors (D_i, i = 1, 2) for different iterations are shown in Figure 4.8. In each iteration, the six particles are divided into one active group of two best particles and a passive group of four particles. According to the OD process, $G_{pers,1}$ and $G_{pers,2}$ are assigned to active group and ($G_{pers,3}$, ..., $G_{pers,6}$) are assigned to passive group. In each iteration, the velocity and position vectors of only the active group are updated. As seen from Figure 4.8, as iteration increases, the OD process causes $[X]_{\text{active\_group}}$ = $[D]_{\text{active\_group}}$, and causing X to be a diagonal matrix. At the end of iteration, the best G_{pers} gives the optimum solution, yielding G_{best} = (2.0, -3.0).

In order to have a geometric interpretation of the OPSO algorithm, the movement of six position vectors and the two orthogonal vectors are shown in Figure 4.9. Here, X_1 and X_2 denote the position vectors of the active group and D_1 and D_2 denote the 2 orthogonal vectors. It can be seen that during early iterations, the position vectors X_1 and X_2 move from random positions toward the orthogonal vectors D_1 and D_2. Finally, as the algorithm iterates further, the X_1 and X_2 coincide with D_1 and D_2.

$$\boldsymbol{t = 1}$$

$$D = \begin{bmatrix} \mathbf{-33.4275} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{9.9113} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{55.4176} & \mathbf{-31.9043} & -80.94 & -23.13 & -6.20 & 79.95 \\ \mathbf{38.2285} & \mathbf{-4.3210} & 23.47 & 23.95 & -1.65 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{-6.2957} & \mathbf{10.2996} & -23.13 & 55.41 & -80.94 & 85.23 \\ \mathbf{-1.6169} & \mathbf{23.4768} & 23.95 & 38.22 & 23.47 & -40.30 \end{bmatrix}$$

$$\boldsymbol{t = 80}$$

$$D = \begin{bmatrix} \mathbf{-3.7383} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.3275} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-28.0374} & \mathbf{6.7003} & 43.83 & -92.54 & 96.11 & -99.44 \\ \mathbf{-14.4267} & \mathbf{16.8004} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0074} & \mathbf{1.4628} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-2.7485} & \mathbf{-2.7543} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$$\boldsymbol{t = 140}$$

$$D = \begin{bmatrix} \mathbf{-4.4026} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4027} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4068} & \mathbf{0.0001} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{-0.0002} & \mathbf{3.4019} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{1.9953} & \mathbf{1.9962} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0007} & \mathbf{-2.9972} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$$\boldsymbol{t = 200}$$

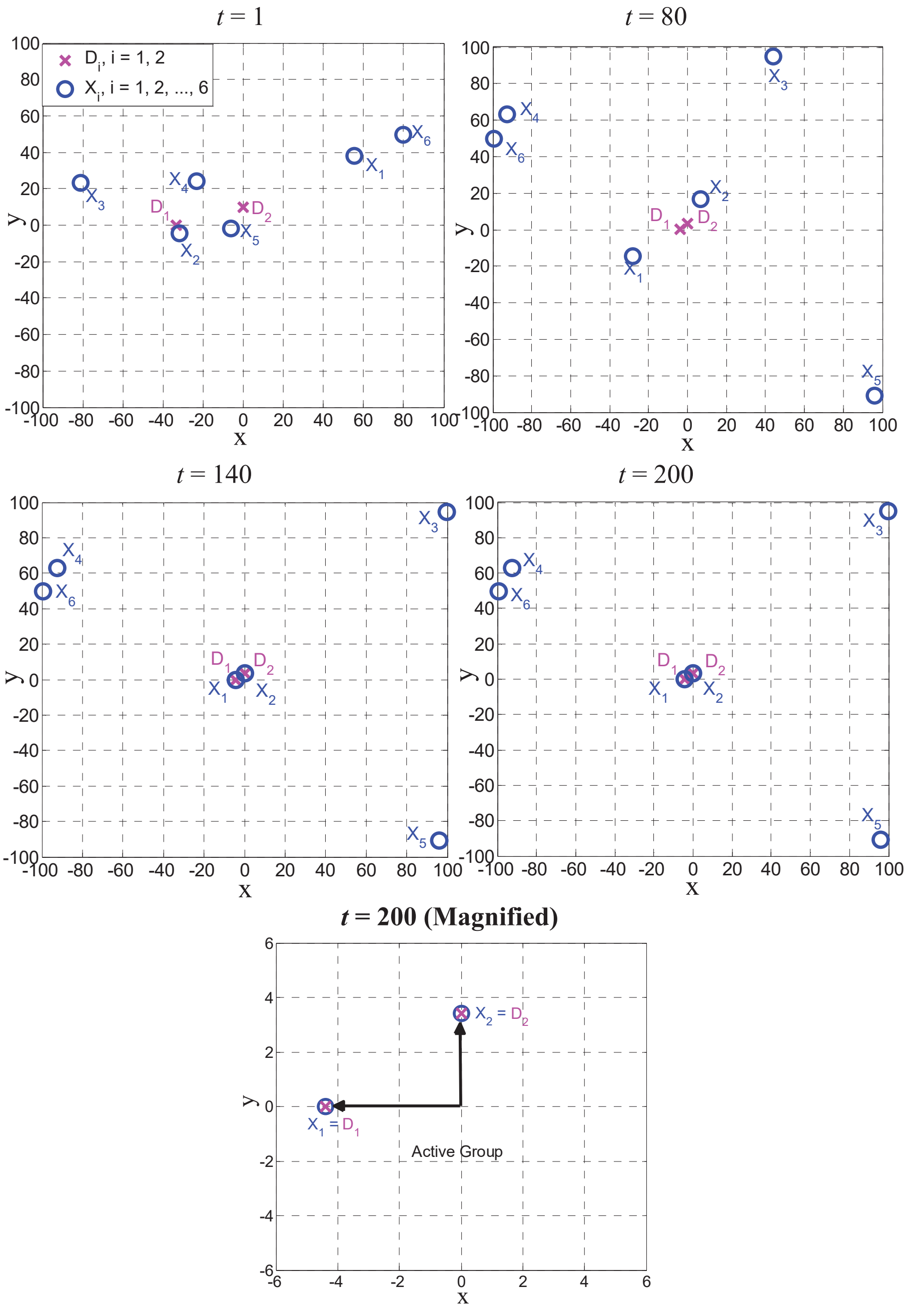
$$D = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4051} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{0.0000} & \mathbf{3.4051} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0000} & \mathbf{2.0000} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0000} & \mathbf{-3.0000} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$$\underbrace{\boldsymbol{\mathit{Active\ \ Group}}} \qquad \underbrace{\mathit{Passive\ \ Group}}$$

Figure 4.8. Numerical example showing convergence of the OPSO algorithm.



$t = 1$

$t = 80$

$t = 140$

$t = 200$

t = 200 (Magnified)

Figure 4.9. Movement of six position vectors (X_1, X_2, …, X_6) and two diagonal vectors, D_1 and D_2 in a 2-dimensional search space ($m = 6$, $d = 2$). The active group consists of X_1 and X_2. At $t = 200$, X_1 and X_2 coincide with D_1 and D_2.

Thirty benchmark functions taken from the Congress on evolutionary computation CEC 2005 [123] CEC 2008 [124] and CEC 2013 [125] are used in Paper E. All the thirty benchmark functions are minimization tasks and are divided into three groups based on their significant physical properties and shapes. The first group involves nine unimodal benchmark functions. The second group includes eleven multimodal benchmark functions and the third group includes ten shifted, rotated and shifted rotated functions.

In order to measure the accuracy, consistency and robustness of OPSO and GPSO algorithms, they were evaluated using the thirty benchmark functions given in Paper E. In terms of convergence characteristics, the OPSO algorithm achieves much better convergence than GPSO algorithm in all thirty benchmark functions. In addition, in terms of fitness values, i.e., best fitness value (BFV), worst fitness value (WFV), mean fitness value (MFV) and standard deviation (σ), in GPSO algorithm, the three fitness values BFV, WFV and MFV differ substantially from their optimum values for all the thirty benchmark functions. Whereas, in OPSO algorithm, the three fitness values are the same or very close to their optimum values for all the thirty benchmark functions. The σ remains 0 or close to 0 in OPSO algorithm, indicating high consistency and reliability. The results shown in Paper E give evidence that the OPSO algorithm is more accurate, stable and robust compared to the GPSO algorithm.

In Paper E, the sensitivity analysis of the proposed OPSO algorithm with variation of swarm size *m* is also studied. Nine selected benchmark functions with $d = 30$ dimensions are tested. With the swarm population, $m = 32$, it has 30 particles in active group and 2 particles in the passive group. The performance of OPSO algorithm improves substantially compared to $m = 30$ (i.e., the number of particles in the passive group equals to zero). Based on the observations in Paper E, as a thumb rule, one may select the swarm population size between 10-30% more than the dimension of the search space.

In addition, by taking thirty unimodal, multimodal, shifted, rotated, and shifted rotated benchmark functions of dimension 30 and 100, it is shown that the OPSO algorithm outperforms the GPSO algorithm and several recently reported ECTs in terms of convergence, accuracy, consistency and reliability.

4.2.6 Summary of Paper F

L. T. Al-Bahrani and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid," Applied Soft Computing, vol. 58, pp. 401-426, 2017.

In Paper F, the OPSO algorithm is proposed to solve the ED problem of small, medium and large-scale PGSs with several practical TGUs and PGS constraints and to improve the performance by overcoming the drawback of GPSO algorithm. In addition, the OPSO algorithm is also applied to solve shifted and rotated unimodal and multimodal benchmark functions with 30 dimensions taken form CEC 2015.

In Paper F, different power constraints imposed on the fuel cost function are power balance, P_L, mismatch in P_L ($P_{L,mismatch}$), generation limits, RRLs, POZs, and feasible operation zones (FOZs). To explain a set of inequality constraints imposed on the TGU, this paper provides an illustrative example to show the lower and upper generation limits and FOZs generated due to presence RRLs and POZs of a TGU. The specifications of TGU_2 taken from [11] are given below.

P_2^0 = 170 MW; $P_{2,min}$ = 50 MW; $P_{2,max}$ = 200 MW; UR_2 = 50 MW; DR_2 = 90 MW. The TGU_2 has two POZs are: POZ_1 = [90,110] and POZ_2 = [140,160].

From (16) in Paper F, the new lower and upper limits of TGU_2 based on RRLs are:

$$P_{2,low} = 80 \text{ MW and } P_{2,high} = 200 \text{ MW},$$

and there are three FOZs are:

$$FOZ_1: 80 \le P_2 \le 90$$
$$FOZ_2: 110 \le P_2 \le 140$$
$$FOZ_3: 160 \le P_2 \le 200$$

Figure 4.9 shows that TGU_2 has minimum and maximum operation limits (OL) given by 50 MW and 200 MW, respectively. However, due to presence OF up-ramp and down-ramp limits, the TGU_2 operates in a new lower and higher OLs given by $P_{2,low}$ = 80 MM and $P_{2,high}$ = 200 MW. Also, the three FOZs are given by: FOZ_1 = [80,90] MW, FOZ_2 = [110,160] MW and FOZ_3 = [160,200] MW shown in white color, and two POZs are given by: POZ_1 = [90,110] MW and POZ_2 = [140,160] MW shown in dark color in Figure 4.9. The intermittent zone ([50,80] MW) is out of OL of the TGU_2.

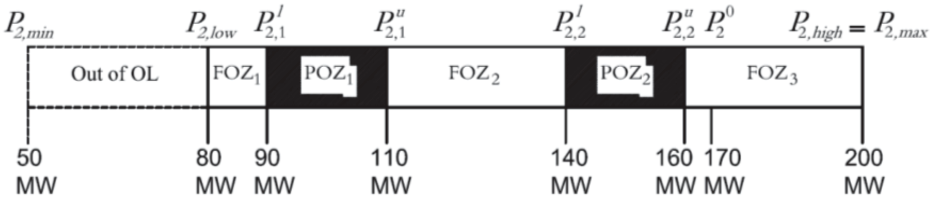


Figure 4.10. Lower and upper generation limits, POZs and FOZs for TGU_2.

To study the accuracy, stability and robustness of OPSO and GPSO algorithms for the ED problem, several performance measures are considered in this paper. The OPSO and GPSO algorithms are evaluated using a small-scale (6 TGUs), a medium-scale (15 TGUs) and a large-scale (40 TGUs) PGS.

In small-scale PGS (6 TGUs), the generation limits, power balance, P_L, $P_{L,mismatch}$, generation limits, RRLs, POZs, and FOZs are considered. The performance of the OPSO algorithm is compared with GPSO algorithm and several other competitive algorithms reported. In terms of fitness values, the performance of OPSO algorithm is compared with 21 ECTs and GPSO algorithm. The OPSO algorithm provides best result in terms of lowest mean fuel cost and lowest σ over 100 independent runs. This indicates that the OPSO algorithm provides stable and accurate solution.

In terms of range (R), the OPSO algorithm provides the second best result among 22 competitive algorithms. In term of AET, the OPSO achieves the third best in terms of AET. Thus, the overall performance of the OPSO algorithm is far superior than the other 22 ECTs. In terms of convergence characteristics, the OPSO shows faster in convergence compared to the GPSO algorithm. This indicates that OPSO algorithm is more consistent and stable than GPSO algorithm. In terms of inequality and equality constraints, the OPSO and GPSO algorithms avoid the 12 POZs of 6 TGUs and are within RRLs and generation limits of each TGU. This indicates that both algorithms are able to satisfy the inequality constraints of small PGS. In addition, The OPSO algorithm provides zero mismatch, i.e., $P_{L,mismatch} = 0$, indicating that the power balance constraint is satisfied.

In the medium-scale PGS (15 TGUs), the generation limits, power balance, P_L, $P_{L,mismatch}$, generation limits, RRLs, POZs, and FOZs, are considered. The performance of OPSO algorithm is compared with GPSO algorithm and other existing 17 ECTs. The OPSO algorithm achieves the best results in terms of mean fuel cost, σ and R. These results indicate that the OPSO algorithm provides consistent, stable and optimum results.

However, in term of AET, OPSO is the second best; the GPSO algorithm being the best among the 19 ECTs. In terms of convergence characteristics, the OPSO algorithm is more consistent, stable and reliable than the GPSO algorithm. In terms of inequality and equality constraints, both the OPSO and GPSO algorithms are able to avoid the eleven POZs of four TGUs and are within generation limits and RRL constraints, thus both algorithms are able to satisfy the inequality constraints. In addition, the OPSO satisfies the zero mismatch condition, i.e., $P_{L,mismatch} = 0$, thus satisfying of the power balance constraint for the medium-scale PGS.

The third PGS is a large-scale and taken from Taipower system [11], it consists of 40 TGUs with P_D = 8,550 MW. It contains 46 POZs distributed among 25 TGUs and the RRLs are imposed on all the 40 TGUs as shown in Figure 4.10. The *B*-loss coefficients are considered and they are generated randomly as is done in [126]. Unfortunately, this PGS is tested by only a few authors under RRLs, POZs and P_L constraints. This may be due to unavailability of *B*-loss coefficients or due to its high dimensions with a large number of power constraints. The generation limits, power balance, P_L, $P_{L,mismatch}$, generation limits, RRLs, POZs and FOZs are considered. The performance of the OPSO algorithm is compared with GPSO algorithm and several other competitive algorithms. The OPSO algorithm provides the best result in terms of mean fuel cost and σ over 100 independent runs. This means that the OPSO algorithm provides the most optimum and consistent results. The range *R* of OPSO algorithm is the lowest among the 15 ECTs, thus indicating that OPSO algorithm gives solution with the lowest dispersion. The AET of OPSO (69 sec), due to its computational complexity, is found to be higher than the GPSO (47 sec). These results indicate that among the 15 ECTs, the OPSO algorithm is the most robust, stable, and is able to provide most optimum solution.

In terms of the convergence characteristics, the GPSO algorithm is unable to solve the ED problem with such a high dimension and under such large number of constraints. Whereas, the OPSO algorithm is capable of providing consistent and reliable optimum solution and gives high accuracy in solving such this complex problem. In terms of inequality and quality constraints, the GPSO algorithm violates RRLs. This means that GPSO algorithm fails in solving 40 TGUs power system indicating that GPSO algorithm is unable to solve large-scale ED problem. Whereas, the OPSO algorithm is able to avoid the 46 POZs of 25 TGUs and remain within RRLs. In addition, the power balance constraint is achieved by the OPSO algorithm and the $P_{L,mismatch}$ is more close to 0.0 than the other competitive algorithms.

The ED of power with various power constraints makes the objective function, i.e., cost function becomes multimodal function and it has local optima. In order to provide a fair comparison and demonstrate the goodness of the proposed OPSO algorithm, ten selected shifted and rotated functions from CEC benchmark functions 2015 [127] are considered in Paper F.

The performance of the OPSO algorithm is compared with GPSO algorithm and several other competitive algorithms in terms of several performance measures. In terms of fitness values, i.e., BFV, WFV and MFV, in OPSO algorithm, these values are the same to their optimum values for all the ten functions. Whereas, in case of the GPSO algorithm, the BFV, WFV and MFV differ substantially from their optimum values. In terms of σ, it remains close to 0.0 in OPSO algorithm, indicating high consistency and reliability. Thus, the OPSO algorithm is more accurate, stable and robust compared to the GPSO algorithm. In terms of the AET, the OPSO algorithm reaches "Threshold Error" within a specific AET. However, GPSO algorithm cannot reach "Threshold Error", indicating that GPSO is unable to solve these ten shifted and rotated CEC 2015 benchmark functions.

In addition, the OPSO algorithm is compared with several existing ECTs with extensive simulation studies. The proposed OPSO algorithm has shown evidence of superior performance compared to several existing ECTs in providing reliable, consistent and optimum solution. The OPSO algorithm is also found to be statistically significant against several ECTs including top-ranked CEC 2015 algorithms.

Thus, the OPSO algorithm is able to achieve superior performance in terms of convergence, consistency and accuracy compared to GPSO algorithm and several competitive ECTs.

4.2.7 Summary of Paper G

L. T. Al-Bahrani and J. C. Patra, "Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid", Energy, vol. 147, pp. 1070-1091, 2018.

In Paper G, the MG-PSO algorithm is proposed to solve the challenging of the ED problem with non-linear, multimodal and discontinuous fuel cost function. In MG-PSO algorithm, two phases, called *Exploration* and *Exploitation*, are used. In the *Exploration* phase, the *m* particles are called *Explorers* and undergo multiple episodes. The *m Explorers* use a different negative gradient to explore new neighbourhood in each episode. Whereas, in the *Exploitation* phase, the *m* particles are named *Exploiters*. The *m Exploiters* use only one negative gradient that is less than that of the *Exploration* phase, to exploit a best neighborhood. This diversity in negative gradients provides a balance between global search and local search.

In Paper G, the MG-PSO algorithm is applied to solve the ED problem of four PGSs, considering more power constraints including valve-point loading (VPL) effects. In addition, the mathematical analysis and theoretical justification of MG-PSO algorithm is provided.

The PGS-1 is a medium-scale system and consists of 13 TGUs. Here, the VPL effects and power generation limits are considered. However, the RRLs, POZs and P_L are not considered. The performance of the MG-PSO algorithm is compared with GPSO-*w* algorithm and several other competitive algorithms. The performance of the proposed MG-PSO algorithm is compared with other 9 existing ECTs. The MG-PSO algorithm provides the best result in terms of mean fuel cost over 25 independent runs. However, it is the second best algorithm in terms of σ. This indicates that the MG-PSO algorithm provides most optimum and consistent results. In addition, the range *R* of MG-PSO algorithm is the second best, thus indicating that MG-PSO algorithm provides solution with low dispersion. These results indicate that among the 9 ECTs, the MG-PSO algorithm is stable, robust and is able to provide optimum solution. In terms of convergence characteristics, The MG-PSO algorithm settles at about 150 iterations and achieves mean fuel cost of about \$17,956/h. However, the GPSO-*w* algorithm takes more than 500 iterations to converge, and settles with a non-optimum mean fuel cost of about \$18,326/h. Thus, the MG-PSO algorithm is capable of providing consistent and reliable solution.

The PGS-2 is also a medium-scale PGS with 15 TGUs. It has 11 POZs in 4 TGUs. The generation limits and RRLs are applied to each TGU. In addition, power balance, the P_L, $P_{L,mismatch}$, generation limits, RRLs, POZs, and FOZs are considered. The performance of the MG-PSO algorithm is compared with GPSO-*w* algorithm and several other 19 ECTs competitive algorithms in terms of several performance measures. The MG-PSO algorithm achieves the best positions in terms of σ and *R* and the second best position result in terms of mean fuel cost. However, in terms of AET, the MG-PSO algorithm is the fourth best. In terms of inequality and equality constraints, both the MG-PSO and GPSO-*w* algorithms are able to avoid the 11 POZs imposed to 4 TGUs and are remain within RRLs imposed on each TGU. In addition, the MG-PSO algorithm is able to satisfy the zero mismatch condition, i.e., $P_{L,mismatch} = 0$, thus satisfying the power balance constraint. These results indicate that the MG-PSO algorithm provides consistent, stable and robust performance.

The PGS-3 is a large-scale PGS taken from Taipower system [11]. It consists of 40 TGUs with 46 POZs distributed among 25 TGUs. The generation limits and RRLs are imposed on all the 40 TGUs. In addition, the power balance, P_L, $P_{L,mismatch}$, POZs, and FOZs, are considered. The performance of MG-PSO algorithm is compared with GPSO-*w* and other 15 existing ECTs. The MG-PSO provides the best result in terms of mean fuel and σ over 25 independent runs. This indicates that the MG-PSO algorithm provides the most optimum and consistent results. In addition, the range *R* of MG-PSO algorithm is the lowest among the 16 ECTs, thus indicating that MG-PSO algorithm provides solution with lowest dispersion. In terms of AET, the MG-PSO algorithm is the third best. The GPSO-*w* algorithm is not able to provide an accurate solution.

In terms of inequality and equality constraints, the GPSO-*w* algorithm violates the RRLs of three TGUs. This means that GPSO-*w* algorithm fails in solving PGS-3. However, the MG-PSO algorithm avoids all the 46 POZs of 25 TGUs and remains within RRLs. In addition, the MG-PSO algorithm is able to satisfy the zero mismatch condition, i.e., $P_{L,mismatch} = 0$, thus satisfying the power balance constraint. These results indicate that among the 16 ECTs, the MG-PSO algorithm is the most stable, robust and is able to provide most optimum solution.

The PGS-4 is a very large-scale PGS taken from Korean PGS [12]. It is a complex system with 140 TGUs each having generation limits and RRLs. In addition, the cost functions of 12 TGUs have VPL effects and 4 TGUs have 11 POZs. The performance of MG-PSO algorithm is compared with two other algorithms as well as GPSO-*w*. The MG-PSO algorithm is efficient in obtaining the best result in terms of mean fuel cost over 25 independent runs. In addition, in terms of σ, the performance of the MG-PSO algorithm is the second best. This shows that the MG-PSO algorithm provides optimum and consistent results. Also, the range *R* of MG-PSO algorithm is the second lowest among the 4 ECTs, thus indicating that it provides solution with low dispersion. In terms of AET, the MG-PSO algorithm shows the second best performance.

In terms of inequality constraints, it is seen that the GPSO-*w* algorithm violets RRLs in 11 POZs. This indicates that GPSO-*w* algorithm is unable to solve ED problem of very large-scale TGUs. Whereas, the MG-PSO algorithm avoids all the 11 POZs imposed on 4 TGUs and remains working within the RRLs of each TGU and overcomes the VPL effects imposed on 12 TGUs. These results indicate that among the 4 ECTs, the MG-PSO algorithm is stable and robust and is able to provide optimum solution.

Table 4.1 shows the comparison in terms of AET between the two proposed algorithms, OPSO and MG-PSO algorithms, for solving the ED problem of medium-scale and large-scale PGSs. We have seen that both the proposed OPSO and MG-PSO algorithms achieve superior results for the ED problem in terms of several performance measures compared with the PSO variants and several competitive algorithms. However when solving large-scale TGUs, e.g., 40 TGUs, the OPSO algorithm requires more time than MG-PSO algorithm to obtain the global optimum because of the condition of $m > d$. However, this demerit is not prominent when solving small-scale and medium-scale PGSs.

Table 4.1. Comparison of the AET between OPSO and MG-PSO algorithms for medium and large PGSs.

Sl. No.	PGS	OPSO					MG-PSO				
		d	m	Mean Cost ($/h)	σ ($/h)	AET (sec)	d	m	Mean Cost ($/h)	σ ($/h)	AET (sec)
1	Medium-scale (15 TGUs)	15	18	32,666.92	0.1394	4.37	15	20	32,677.96	0.0348	9.20
2	Large-scale (40 TGUs)	40	45	127,349.83	302.35	69.32	40	20	126,625.02	20.27	29.38

In Paper G, statistical tests were also carried out to demonstrate the effectiveness of the proposed MG-PSO algorithm. The MG-PSO algorithm has proved to be a powerful and highly effective algorithm that is capable of solving multimodal, discontinuous and non-convex functions.

4.2.8 Summary of Paper H

L. T. Al-Bahrani and J. C. Patra, "Multi-gradient PSO algorithm with enhanced exploration and exploitation," Applied Soft Computing. Under review.

In Paper H, the impact of the VPL effects on the fuel cost function is demonstrated. Let us consider two TGUs, TGU_1 and TGU_2, with a set of parameters as shown in Table 4.2 [44]. The TGU_1 and TGU_2 are steam powered turbo generators with multiple valves. Practically, the valves of steam-turbine control the steam entering through separate nozzle groups. Each nozzle group provides best efficiency when it is operating at maximum active output power. Thus, when increasing the active output power, the valves of seam-turbine are opened and closed in sequence in order to achieve high efficiency for a given output power. Then, it causes ripple-like effects and subsequently the characteristics of the fuel cost function become non-linear.

Figure 4.10 shows the total fuel cost of TGU_1 and TGU_2 under VPL effects. Multiple local minima are caused by the sinusoidal function imposed on the fuel cost of each TGU. Thus, the MG-PSO algorithm is proposed to solve such a complex problem.

Table 4.2. Parameters of TGU_1 and TGU_2.

TGU_j	a_j \$/h	b_j \$/MWh	c_j \$/MW^{2}h	e_j \$/h	f_j (MW^{-1})	$P_{j,min}$ (MW)
1	958.29	21.60	0.00043	450	0.041	150
2	1,313.60	21.05	0.00063	600	0.036	135

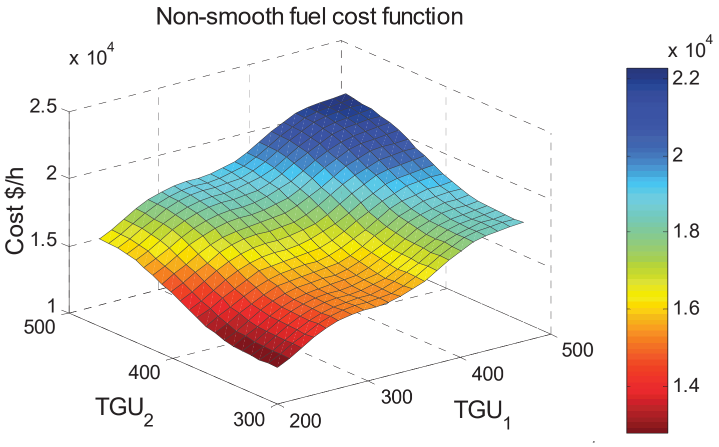


Figure 4.11. Total fuel cost function of TGU₁ and TGU₂ under VPL effects.

In addition, the MG-PSO algorithm is used to solve ten selected shifted and rotated unimodal and multimodal benchmark functions with dimensions of 30 and 100 taken from CEC 2015. The mechanism of the MG-PSO algorithm depends on two phases called *Exploration* and *Exploitation*. In the *Exploration* phase, the *m* particles are named *Explorers* and undergo multiple episodes. The *Explorers* use a different negative gradient in each episode to explore new neighbourhood whereas in the *Exploitation* phase, the *m* particles are named *Exploiters* and they use one negative gradient that is less than that of the *Exploration* phase, to exploit a best neighborhood. This diversity in negative gradients gives a balance between global search and local search of the *Explorers* and *Exploiters*.

Performance of the proposed MG-PSO and GPSO-*w* algorithms are obtained in terms of fitness values, convergence rate, accuracy, consistency, robustness and reliability. Ten shifted and rotated unimodal and multimodal benchmark functions are considered. Each benchmark function is tested with $d = 30$ and $d = 100$ dimensions. The comparison is achieved between the MG-PSO and GPSO-*w* algorithms in terms of BFV, WFV, MFV, σ and AET.

In case of GPSO-*w* algorithm, the BFV, WFV and MFV differ substantially from their optimum values for all the ten functions with $d = 30$ and $d = 100$. However, in MG-PSO algorithm, the three fitness values are the same as their optimum values for all the ten benchmark functions. In terms of the σ, it remains close to 0.0 in MG-PSO algorithm, indicating its high consistency and reliability. In terms of the AET, the

MG-PSO algorithm reaches “Accepted Error” within a specific AET. However, the GPSO-*w* algorithm could not reach “Accepted Error”, indicating that GPSO-*w* is unable to solve these ten shifted and rotated benchmark functions. Whereas, MG-PSO algorithm successfully achieves the optimum solution for all the ten benchmark functions with $d = 30$ and $d = 100$. These results prove that the MG-PSO algorithm is more accurate, stable and robust compared to the GPSO-*w* algorithm.

The MG-PSO algorithm is also applied to solve the ED of South Korea power generating system with 140 TGUs [12]. Superior performance by the MG-PSO algorithm over the GPSO-*w* algorithm and several existing optimization techniques has been shown in terms of fitness value, convergence rate, and consistency. In addition, by using unpaired t-test, the statistical significance of the MG-PSO algorithm is found out against several contending algorithms including top-ranked CEC 2015 algorithms.

4.3 Chapter summary

This Chapter provides an executive summary of each of the eight Papers. Contributions of these eight Papers were investigated in this Chapter. The salient features and important results of each of the eight Papers were highlighted.

Chapter 5: Discussions and Future Directions

5.1 Chapter overview

This thesis ends with a critical discussion, linking the characteristics of evaluating the proposed two novel algorithms, orthogonal PSO (OPSO) and multi-gradient PSO (MG-PSO) algorithms. The two algorithms have been applied to solve the ED problem of small-scale to large-scale TGU power generating systems and a set of benchmark functions taken from the Congress of evolutionary computation 2015 (CEC 2015). In Section 5.2, discussion on the main investigations is presented. Then, addressing the answer of the research questions is discussed in Section 5.3. After that, outcomes of this study are presented in Section 5.4. Significant contribution to knowledge from this study is given in Section 5.5. Subsequently, limitations and recommendations for future study are provided in Section 5.6. Then, the conclusion is presented in Section 5.7. Finally, Chapter summary is given in Section 5.8.

5.2 Main investigations

The economic dispatch (ED) of active output power of large-scale TGU power generating systems with various practical power constraints is a challenge in operation and management of the power generating systems. The aims of this thesis have been achieved by completing the eight Papers to address the following points.

The first point of investigation is the current issues faced by Iraq NDC to solve the complex ED problem as reported in Section 1.3.

The second point investigated is the review of the existing literature for solving non-convex ED problem of large-scale TGUs with different power constraints. The comprehensive literature review reveals that the ECTs are continuously being developed and improved year by year and being compared with other optimization techniques to deal with even larger PGSs with an increasing number of practical operating power constraints.

The third point of investigation is identifying the gap in this field of research. Recently, a large number of ECT-based algorithms have been proposed and applied for the optimization of non-convex and non-smooth ED problem. However, the issue of scalability has not been addressed sufficiently. In other words, the number of ECTs techniques which can be applied to large-scale ED problem with or more than 40 TGUs is small.

The next point of investigation was to address the potential difficulties of the ED problem. These have been thoroughly analysed in the Papers A-H that are the backbone of this thesis. Especially, analysis of the active output power of each TGU with different operating power constraints was emphasized. Two novel algorithms, OPSO and MG-PSO, are proposed and their performance was evaluated.

The performance of the proposed OPSO and MG-PSO algorithms is assessed on small-scale to large-scale ED problems with several operating power constraints. The power constraints are generation limits, VPL effects, POZs, FOZs, RRLs, P_L, $P_{L,mismatch}$, and power balance. It is shown that the OPSO and MG-PSO algorithms are robust, efficient, and have a high performance when applied to complex, large-scale practical ED problems.

5.3 Addressing the answer to the research questions

The importance of the ED problem as a highly complex optimization problem in operation and management of TGUs power generating system was investigated through the answer to question #1. One can see the significance of solving ED problem from the results shown in Papers F and G in testing four small-scale to large-scale PGSs (PGS-1, PGS-2, PGS-3 and PGS-4) in terms of minimum, maximum and mean fuel costs. The significance has been demonstrated by optimally allocating the required load demand among the online TGUs such that the operating fuel cost is minimized. Thus, this study gives an approach to determine the most efficient, low-cost, stable and reliable operation of the PGSs.

The answer to question #2 has been investigated by proposing two novel algorithms, OPSO and MG-PSO algorithms, for solving the ED problem of large-scale TGUs power generating system with several operating power constraints. The PSO variants, GPSO and GPSO-*w* algorithms, mentioned in this study are unable to solve such a complex problems, because of loss of balance between global search and local search. In addition, the performance of GPSO and GPSO-*w* algorithms deteriorates for high-dimensional optimization problems, especially in presence of non-linear operating constraints. Both the OPSO and MG-PSO algorithms were proposed in this thesis to alleviate the shortcomings in GPSO and GPSO-*w* algorithms and for solving the unimodal and multimodal complex problems including the ED problem, as shown in Papers A-H.

In addition, several operating power constraints, e.g., generation limits, RRLs, POZs, power balance, FOZs, P_L and $P_{L,mismatch}$, are taken into account in computing the fuel cost function. All these power constraints were solved through a set of equality and inequality constraints imposed on fuel cost function for the ED problem of small-scale to large-scale PGSs. Formulation the non-convex fuel cost function with VPL effects was also achieved. The mathematical analysis and theoretical justification for the ED problem were reported in Papers A-H.

A set of unimodal, multimodal, shifted, rotated and shifted and rotated benchmark functions taken from the Congress of evolutionary computation 2015 (CEC 2015) were addressed with 30 and 100 dimensions. In addition, using unpaired t-test, the statistical significance of the OPSO and MG-PSO algorithms have been found against several contending algorithms including top-ranked CEC 2015 algorithms.

The OPSO algorithm was found to be superior than several contending of ECTs in solving large-scale TGUs in terms of minimum, maximum and mean fuel costs. However, the OPSO algorithm consumes excessive execution time because it uses the OD process in *d* best particles. Therefore, the number of particles is about 10-30% more than the dimension of search space. This leads to more execution time in each iteration. Therefore, the MG-PSO algorithm was proposed to overcome this demerit of the OPSO algorithm.

5.4 Outcomes of this study

This study is focused on applying the OPSO and MG-PSO algorithms to solve ED problem. In addition, these two algorithms were successfully applied to solve a set of selected complex unimodal and multimodal benchmark functions taken from the CEC 2015. The main outcomes of this study are summarized below.

5.4.1 Orthogonal PSO algorithm

The OPSO algorithm is applied for solving the ED problem of small-scale to large-scale TGU power systems with various practical power constraints. In small-scale PGS (6 TGUs), the OPSO algorithm provides the best results in terms of mean cost value (F_{mean}) and has lowest standard deviation (σ) compared with 21 ECTs as well as the GPSO algorithm, as shown in Table 3 (Paper F). In terms of inequality and equality operating power constraints, both the OPSO and GPSO algorithms are able to avoid the 12 POZs in 6 TGUs and are within RRLs, as shown in Paper F. In terms of power balance constraint, the OPSO algorithm provides zero mismatch, i.e., $P_{L,mismatch} = 0$, indicating that the power balance constraint is satisfied.

For a medium-scale PGS (15 TGUs), the OPSO algorithm provides the best results in terms of F_{mean} and σ among 19 existing ECTs. In terms of average execution time (AET), the OPSO is the second best among the 19 ECTs. In terms of the convergence characteristics, the OPSO algorithm is more consistent, stable and reliable than the GPSO algorithm, as shown in Figures 13 and 14 (Paper F). In terms of inequality power constraints, the OPSO and GPSO algorithms are able to avoid the 11 POZs in 4 TGUs and are within RRL power constraints, as shown in Table 9 (Paper F). Thus, both algorithms are able to satisfy the inequality constraints of medium-scale PGS. In terms of the power balance constraints, the OPSO algorithm satisfies the zero mismatch condition, i.e., $P_{L,mismatch} = 0$.

For a large-scale PGS (Taiwan PGS with 40 TGUs), the OPSO algorithm provides the best result in terms of F_{mean} and σ, as shown in Table 12 (Paper F). The AET of the OPSO algorithm is 69 sec. Due to its computational complexity, it is found to be higher than the GPSO (47 sec). In terms of convergence characteristics, the GPSO algorithm is unable to solve the ED problem with such a high dimension (40 TGUs) with such a large number of constraints. In terms of inequality power constraints, the GPSO algorithm violates the RRLs, as shown in Table 13 (Paper F). However, the OPSO algorithm avoids the 46 POZs in 25 TGUs and remains within RRLs. The GPSO is out of comparison in terms of power balance constraint, because it failed in solving large-scale PGS. However, the OPSO algorithm satisfies the zero mismatch condition, i.e., $P_{L,mismatch} = 0$.

The proposed OPSO algorithm is also applied on a set of unimodal, multimodal, shifted, rotated, and shifted and rotated benchmark functions with dimensions of 30 and 100, as shown in Papers E and F. Performance comparison between the OPSO and GPSO algorithms in terms of best fitness value (BFV), worst fitness value (WFV), mean fitness value (MFV), mean fitness error value (MFEV), σ and AET are provided in Table 16 (Paper F) and Table 4 (Paper E).

In GPSO algorithm, the three fitness values BFV, WFV and MFV differ substantially from their optimum values. However, in OPSO algorithm, the three fitness values are the same to their optimum values. The MFEV of GPSO algorithm is so far from "Threshold Error" However, in the OPSO algorithm, the MFEV is smaller than "Threshold Error". In terms of the σ, the OPSO algorithm remains close to 0.0. In terms of the AET, the OPSO algorithm reaches "Threshold Error" within a specific AET. However, the GPSO algorithm cannot reach "Threshold Error", which indicating that the GPSO is unable to solve unimodal, multimodal, shifted, rotated, and shifted and rotated benchmark functions complex benchmark functions with 30 and 100 dimensions.

In addition, the performance of the proposed OPSO algorithm is compared with few ECTs recently reported by other authors as shown in Papers E and F including the three top-ranked algorithms in the CEC 2015. Superior performance of the proposed OPSO algorithm has been shown compared to several existing ECTs in providing reliable, consistent and optimum solution for CEC 2015 benchmark functions. In addition, The OPSO algorithm was found to be statistically significant against several ECTs including top-ranked CEC 2015 algorithms.

5.4.2 Multi-gradient PSO algorithm

The MG-PSO algorithm are applied for solving the ED problem of small-scale to large-scale TGU power generating systems with various practical power constraints. In small-scale PGS (6 TGUs), the MG-PSO algorithm provides the best results in terms of minimum, maximum and mean cost and has lowest standard deviation (σ) compared with 14 ECTs as shown in Table I (Paper D). In terms of inequality and equality operating power constraints, both the MG-PSO and GPSO algorithms are able to avoid the 12 POZs in 6 TGUs and are within RRLs, as shown in Paper D. In terms of power balance constraint, it can be seen that the MG-PSO algorithm provides zero mismatch, i.e., $P_{L,mismatch} = 0$, indicating that the power balance constraint is satisfied.

For a medium-scale PGS (13 TGUs), the effects of VPL and generation limits in each of the 13 TGUs are taken into account. Table 4 in Paper G shows, the MG-PSO algorithm provides the best result in terms of F_{mean}. However, it is the second best in terms of σ. In terms of the convergence characteristics, the MG-PSO algorithm is capable of providing consistent and reliable solution. However, the GPSO-*w* algorithm was rather far from optimum solution due to VPL effects, as shown in Figures 5 and 6 (Paper G).

For another medium-scale PGS (15 TGUs), the P_L, RRLs, POZs, generation limits are taken into account for solving the non-convex ED problem. The results in Table 9 (Paper G) show that the MG-PSO algorithm achieves the best positions in terms of σ and the second best position result in terms of F_{mean}. The best F_{mean} is achieved by the OPSO algorithm. However, in terms of AET, the MG-PSO algorithm was the fourth best (9.2088 sec). However, the OPSO algorithm is faster in convergence (4.377 sec) and is better performance than the MG-PSO algorithm in solving the ED of 15 TGUs. In terms of inequality power constraints, the MG-PSO, OPSO and GPSO-*w* algorithms are able to avoid the 11 POZs in 4 TGUs and are within RRL power constraints, as shown in Table 10 (Paper G). Thus, the MG-PSO, OPSO and GPSO algorithms are able to satisfy the inequality constraints of medium-scale power system (15 TGUs). In terms of power balance constraints, the MG-PSO and OPSO algorithms satisfy the zero mismatch condition, i.e., $P_{L,mismatch} = 0$, as shown in Table 11 (Paper G).

For large-scale PGS (Taiwan PGS with 40 TGUs), the generation limits, RRLs, P_L, $P_{L,mismatch}$, power balance, and POZs are taken into account. The MG-PSO algorithm provides the best result in terms of F_{mean} and σ as shown in Table 14 (Paper G). This indicates that the MG-PSO algorithm provides the most optimum and consistent results. In terms of AET, the MG-PSO is faster in convergence (29.38 sec) that the OPSO algorithm (69.32 sec). However, the GPSO-*w* algorithm is unable to provide an accurate solution. The GPSO-*w* algorithm was unable to solve the ED problem with such a high dimensional search space and under large number of power constraints, shown in Table 14 (Paper G).

In terms of inequality constraints, the GPSO-*w* algorithm failed in solving 40 TGUs system. It is unable to solve large-scale ED problem. However, the MG-PSO algorithm is able to avoid all the 46 POZs in 25 TGUs and remains within RRLs. In addition, the power balance constraint is solved by the MG-PSO algorithm, it is more close to 0.0 than the OPSO algorithm, as shown in Table 16 (Paper G). However, the GPSO-*w* is out of the comparison, because it failed in solving the ED of Taiwan PGS.

For a very large-scale PGS, it is South Korea PGS with 140 TGUs. Each TGU has RRLs and the fuel cost functions of 12 TGUs have VPL effects and 4 TGUs have 11 POZs. The GPSO-*w* was unable to solve the 140 TGUs power system with such a high dimension (d = 140) and under such a large number of power constraints, as shown in Table 18 (Paper G). Early convergence of the GPSO-*w* algorithm indicates that it has trapped into one local minimum. This indicates that the GPSO-*w* algorithm is unable to solve the ED problem. However, the MG-PSO algorithm is efficient in obtaining the best result in terms of F_{mean}. In addition, in terms of σ, the performance of the MG-PSO algorithm is the second best. Thus, the MG-PSO algorithm is stable and robust and is able to provide optimum solution of such a complex PGS.

In terms of inequality power constraints, the GPSO-*w* algorithm violates the RRLs in 11 TGUs, as shown in Table 19 (Paper G). The GPSO-*w* algorithm failed in solving the ED of 140 TGUs. Whereas, the MG-PSO algorithm avoids the 11 POZs in 4 TGUs and remains within the RRLs. In addition, the MG-PSO algorithm is able to solve the non-convex fuel cost function due to the effects of VPL in 12 TGUs.

The MG-PSO algorithm is also applied on ten selected shifted and rotated benchmark functions with dimensions of 30 and 100 taken from CEC 2015. Performance comparison between the MG-PSO and GPSO-*w* algorithms in terms of BFV, WFV, MFV, MFEV, σ and AET are shown in Tables 5 and 6 (Paper H). In case of GPSO-*w* algorithm, the BFV, WFV and MFV differ substantially from their optimum values for all the ten functions with $d = 30$ and $d = 100$. However, in MG-PSO algorithm, the BFV, WFV and MFV are the same as their optimum values for all the ten functions. The MFEV of GPSO-*w* algorithm is far from the "Accepted Error". Whereas, in MG-PSO algorithm, the MFEV is smaller than "Accepted Error", the MFEV = 0.0 for all the ten functions. In terms of the σ, the MG-PSO algorithm remains close to 0.0. In terms of the AET, the MG-PSO algorithm reaches "Accepted Error" within a specific AET. However, the GPSO-*w* algorithm is unable to reach "Accepted Error". Thus, the GPSO-*w* is unable to solve the selected ten shifted and rotated with 30 and 100 dimensions.

Based on the sensitivity analysis of the MG-PSO algorithm against a swarm population size, the appropriate value of *m* is 20 is selected in solving different objective functions. This means that the MG-PSO algorithm is less affected by swarm population size, as shown Papers G and H. In contrast, the OPSO algorithm is largely affected by increasing the dimension of the problems.

Statistical tests are carried out to demonstrate the effectiveness of the proposed MG-PSO algorithm. Thus, it has proved to be a powerful and highly effective algorithm that is capable of solving multimodal, discontinuous and non-convex functions.

5.5 Significant contribution to knowledge

The eight Papers, Papers A-H appended in Appendix-1 represent original and distinguished contribution to knowledge in the field of operation and management of the power generating systems. These eight papers have focused on the development of two novel algorithms, i.e., OPSO and MG-PSO algorithms, to solve unimodal and multimodal problems including the ED problem. Developing new computation techniques such as OPSO and MG-PSO algorithms for solving the ED problem is necessary with the rapid technological evolution in smart power grid. This study will help the electrical engineers in the dispatch center to manage and govern the electrical energy to the consumer.

5.6 Limitations and recommendations for future study

5.6.1 Limitations

An important limitation is due the security reasons, the Iraq PGS data is not available. Therefore, technical data of other power systems that are similar to Iraq PGS, e.g., Taiwan [11] and South Korea [12] power systems, as well as small and medium power systems were considered in this study.

Another important limitation that was being faced in this study is that, obtaining the complete technical data for a real-world PGS with its practical power constraints are difficult. For example, in Taiwan power system, the *B*-loss coefficients are not available. Therefore, they are generated randomly as is done in [126].

Finally, another important limitation that was being faced in this study is that, the number of other ECTs applied to large-scale real-world ED problems remains low. This means that the performance comparison of the proposed OPSO and MG-PSO algorithms is restricted by a limited number of competitive ECTs.

5.6.2 Recommendations

In this study, the OPSO and MG-PSO algorithms have been proved to be highly effective optimization techniques to solve the non-convex ED problem. Hence, the recommendation of this study is to apply these two novel algorithms, i.e., OPSO and MG-PSO algorithms, in solving the ED problem of Iraq PGS.

In a smart power gird environment, the dispatch of active output power of different energy resources at minimum operational cost has been a significant part of research. Recently, with increasing interests in renewable energy resources, the SPG comprises different types of power stations, e.g., solar, wind, thermal, geothermal, gas, hydro, nuclear power stations. In addition, a large number of the power constraints are imposed. Thus, finding an optimum solution to such a SPG has become an essential issue. The future work need to be directed towards identifying such a real-world problem which can be solved using the two novel algorithms, i.e. OPSO and MG-PSO algorithms.

5.7 Conclusion

Economic dispatch (ED) is one of the most essential and challenging task in a PGS. It is a highly complex optimization problem in operation and management of the power generating systems. The significance of the ED problem has been investigated this thesis.

Two novel algorithms, i.e., orthogonal PSO (OPSO) and multi-gradient PSO (MG-PSO) algorithms have been proposed to solve such a complex problem. With extensive simulation studies, performance of both the algorithms was compared with PSO variants and several existing competitive algorithms. Their superior performances are demonstrated in terms of mean, maximum and minimum costs, convergence rate, accuracy and consistency when solving small-scale to large-scale power systems. In addition, the OPSO and MG-PSO algorithms were applied to a set of complex unimodal and multimodal benchmark functions with 30 and 100 dimensions. Both algorithms outperformed several existing ECTs including top-ranked CEC 2015 algorithms.

The sensitivity analysis and statistical tests were carried out to demonstrate the effectiveness of the OPSO and MG-PSO algorithms. Thus, the both algorithms proved to be powerful and highly effective algorithms that are capable of solving several complex unimodal and multimodal functions including the ED problem.

References

[1] M. H. Yasen, "Analyse the problems of the Iraqi power system," *Advances in Engergy Engineering,* vol. 4, pp. 11-14, 2016.

[2] Ministry of Electricity/Iraq (2017). *Annual statistic report.* Available: http://www.moelc.gov.iq.

[3] E. Naderi, A. Azizivahed, H. Narimani, M. Fathi, and M. R. Narimani, "A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm," *Applied Soft Computing,* vol. 61, pp. 1186-1206, 2017.

[4] Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, "Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization," *Applied Soft Computing,* vol. 59, pp. 229-242, 2017.

[5] T. Niknam, M. R. Narimani, and M. Jabbari, "Dynamic optimal power flow using hybrid particle swarm optimization and simulated annealing," *International Transactions on Electrical Energy Systems,* vol. 23, pp. 975-1001, 2013.

[6] Y. Labbi, D. B. Attous, H. A. Gabbar, B. Mahdad, and A. Zidan, "A new rooted tree optimization algorithm for economic dispatch with valve-point effect," *International Journal of Electrical Power & Energy Systems,* vol. 79, pp. 298-311, 2016.

[7] Export.gov (2016). *Export Arab Emirates-energy/power.* Available: https://www.export.gov/article?id=United-Arab-Emirates-Energy-Power.

[8] S. Mohammed and G. Mansoori, "A unique view on carbon dioxide emissions around the world," *Global Journal of Earth Science and Engineering,* vol. 4, pp. 8-17, 2017.

[9] EPRI, "Can future coal power plants meet CO_2 emission standards without carbon capture and storage," *Electric Power Research Institute*, 2015.

[10] Y. Xu, W. Zhang, and W. Liu, "Distributed dynamic programming-based approach for economic dispatch in smart grids," *IEEE Transactions on Industrial Informatics,* vol. 11, pp. 166-175, 2015.

[11] J. Sun, V. Palade, X. J. Wu, W. Fang, and Z. Wang, "Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization," *IEEE Transactions on Industrial Informatics,* vol. 10, pp. 222-232, 2014.

[12] J. Neto, G. Reynoso-Meza, T. Ruppel, V. Mariani, and L. Coelho, "Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution," *International Journal of Electrical Power & Energy Systems,* vol. 84, pp. 13-24, 2017.

[13] P. Chen and H. Chan, "Large-scale economic dispatch by genetic algorithm," *IEEE Transactions on Power Systems,* vol. 10, pp. 1919-1926, 1995.

[14] A. Hoke, A. Brissette, S. Chandler, A. Pratt, and D. Maksimović, "Look-ahead economic dispatch of microgrids with energy storage, using linear programming," in *2013 1st IEEE Conference on Technologies for Sustainability (SusTech)*, 2013, pp. 154-161.

[15] F. Mahdi, P. Vasant, V. Kallimani, J. Watada, P. Fai, and M. Abdullah-Al-Wadud, "A holistic review on optimization strategies for combined economic emission dispatch problem," *Renewable and Sustainable Energy Reviews,* 2017.

[16] J. Delson and S. Shahidehpour, "Linear programming applications to power system economics, planning and operations," *IEEE Transactions on Power Systems,* vol. 7, pp. 1155-1163, 1992.

[17] P. Sharma and N. Hasteer, "Analysis of linear sequential and extreme programming development methodology for a gaming application," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, India, 2016, pp. 1916-1920.

[18] Z. Wu, J. Ding, Q. Wu, Z. Jing, and J. Zheng, "Reserve constrained dynamic economic dispatch with valve-point effect: A two-stage mixed integer linear programming approach," *Journal of Power and Energy Systems,* vol. 3, pp. 203-211, 2017.

[19] D. Reich, "A linear programming approach for linear programs with probabilistic constraints," *European Journal of Operational Research,* vol. 230, pp. 487-494, 2013.

[20] R. Jonker, *Optimization techniques: An introduction*. New York: Springer, 1981.

[21] B. Lazzerini and F. Pistolesi, "A linear programming-driven MCDM approach for multi-objective economic dispatch in smart grids," in *Proceedings of 2015 Intelligent Systems Conference*, London, UK, 2015, pp. 475-484.

[22] N. Nabona and L. Freris, "Optimisation of economic dispatch through quadratic and linear programming," in *Proceedings of Institution of Electrical Engineers,* IET Journals and Magazine, vol. 120, 1973, pp. 574-580,

[23] T. Panigrahi, A. Sahoo, and A. Behera, "A review on application of various heuristic techniques to combined economic and emission dispatch in a modern power system scenario," *Energy Procedia,* vol. 138, pp. 458-463, 2017.

[24] A. Viswanath, L. Goel, and W. Peng, "Mixed integer programming formulation techniques and applications to Unit Commitment problem," in *Proceedings of 10th International Power & Energy Conference (IPEC)*, Ho Chi, Minh, City, Vietnam, 2012, pp. 25-30.

[25] J. Muckstadt and R. Wilson, "An Application of mixed-integer programming duality to scheduling thermal generating systems," *IEEE Transactions on Power Apparatus and Systems,* vol. 87, pp. 1968-1978, 1968.

[26] M. Wang, H. Gooi, S. Chen, and S. Lu, "A Mixed integer quadratic programming for dynamic economic dispatch with valve point effect," *IEEE Transactions on Power Systems,* vol. 29, pp. 2097-2106, 2014.

[27] F. Alfarhan and A. Alsohaily, "Self-organizing wireless network parameter optimization through mixed integer programming," in *Proceedings of IEEE International Conference on Communications (ICC)*, Paris, France, 2017, pp. 1-6.

[28] Y. Zhao and S. Liu, "Global optimization algorithm for mixed integer quadratically constrained quadratic program," *Journal of Computational and Applied Mathematics,* vol. 319, pp. 159-169, 2017.

[29] E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling, "MIP: theory and practice-closing the gap," in *Proceedings of System Modelling and Optimization*, Boston, USA, 2000, pp. 19-49.

[30] M. Basu, "Hybridization of bee colony optimization and sequential quadratic programming for dynamic economic dispatch," *International Journal of Electrical Power & Energy Systems,* vol. 44, pp. 591-596, 2013.

[31] C. Guo, Y. Bai, and J. Jian, "An improved sequential quadratic programming algorithm for solving general nonlinear programming problems," *Journal of Mathematical Analysis and Applications,* vol. 409, pp. 777-789, 2014.

[32] P. Boggs and J. Tolle, "Sequential quadratic programming," *Acta Numerica,* vol. 4, pp. 1-51, 2008.

[33] R. Arora, *Optimization: algorithms and applications*. Boca Raton, Florida: CRC Press, 2015.

[34] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Transactions on Power Systems,* vol. 17, pp. 411-416, 2002.

[35] M. Nazari-Heris, B. Mohammadi-Ivatloo, and G. B. Gharehpetian, "A comprehensive review of heuristic optimization algorithms for optimal combined heat and power dispatch from economic and environmental perspectives," *Renewable and Sustainable Energy Reviews,* vol. 81, pp. 2128-2143, 2018.

[36] L. Jebaraj, C. Venkatesan, I. Soubache, and C. Rajan, "Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review," *Renewable and Sustainable Energy Reviews,* vol. 77, pp. 1206-1220, 2017.

[37] R. Reynolds, "*Evolutionary computation: Towards a new philosophy of machine intelligence*. New York, IEEE Press, 1995.

[38] B. Qu, Y. Zhu, Y. Jiao, M. Wu, P. Suganthan, and J. Liang, "A survey on multi-objective evolutionary algorithms for the solution of the environmental/ economic dispatch problems," *Swarm and Evolutionary Computation,* vol. 38, pp. 1-11, 2018.

[39] B. Mohan and R. Baskaran, "A survey: ant colony optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications,* vol. 39, pp. 4618-4627, 2012.

[40] K. Chaturvedi, M. Pandit, and L. Srivastava, "Self-Organizing hierarchical particle swarm optimization for nonconvex economic dispatch," *IEEE Transactions on Power Systems,* vol. 23, pp. 1079-1087, 2008.

[41] M. Nemati, M. Braun, and S. Tenbohlen, "Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming," *Applied Energy,* 2017.

[42] D. Fogel, *Evolutionary computation toward a new philosophy of machine intelligence*, 3rd ed. Hoboken: John Wiley & Sons, 2006.

[43] J. Sun, *Particle swarm optimisation classical and quantum perspectives*. Boca Raton, Florida, USA: CRC Press 2012.

[44] A. Meng, H. Hu, H. Yin, X. Peng, and Z. Guo, "Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects," *Energy,* vol. 93, pp. 2175-2190, 2015.

[45] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of Fuzzy Information conference*, Berkeley, USA, 1996, pp. 519-523.

[46] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *International Journal of global Optimization,* vol. 11, pp. 341-359, 1997.

[47] D. Pedroso, M. Bonyadi, and M. Gallagher, "Parallel evolutionary algorithm for single and multi-objective optimisation: differential evolution and constraints handling," *Applied Soft Computing,* vol. 61, pp. 995-1012, 2017.

[48] L. Lakshminarasimman and S. Subramanian, "Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution," *IEE Generation, Transmission and Distribution,* vol. 153, pp. 693-700, 2006.

[49] A. Qing, *Differential evolution: fundamentals and applications in electrical engineering*. Singapore: IEEE Press, 2009.

[50] A. Meng, Y. Chen, H. Yin, and S. Chen, "Crisscross optimization algorithm and its application," *Knowledge-Based Systems,* vol. 67, pp. 218-229, 2014.

[51] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks, 1995*, pp. 1942-1948.

[52] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995,* pp. 39-43.

[53] J. Kennedy, "The behavior of particles," in *Proceedings of 7th International Conference on Evolutionary Programming,* San Diego, California, USA, 1998, pp. 579-589.

[54] X. Yang, *Nature-inspired optimization algorithms*: Elsevier Science, 2014.

[55] J. Park, Y. Jeong, J. Shin, and K. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Transactions on Power Systems,* vol. 25, pp. 156-166, 2010.

[56] M. Arumugam and M. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing,* vol. 8, pp. 324-336, 2008.

[57] W. Elsayed, Y. Hegazy, M. El-bages, and F. Bendary, "Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem," *IEEE Transactions on Industrial Informatics,* vol. 13, pp. 1017-1026, 2017.

[58] P. Das, H. Behera, and B. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm and Evolutionary Computation,* vol. 28, pp. 14-28, 2016.

[59] M. Dorigo, G. Caro, and L. Gambardella, "Ant algorithms for discrete optimization," *Artificial life,* vol. 5, p. 137, 1999.

[60] M. Dorigo, *Ant colony optimization*. USA: MIT Press Computing & Engineering Collection Cambridge, 2004.

[61] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, "Ant colony optimisation for economic dispatch problem with non-smooth cost functions," *International Journal of Electrical Power & Energy Systems,* vol. 32, pp. 478-487, 2010.

[62] J. Zhou, C. Wang, Y. Li, P. Wang, C. Li, P. Lu*, et al.*, "A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security," *Applied Mathematical Modelling,* vol. 45, pp. 684-704, 2017.

[63] A. Mousa, "Hybrid ant optimization system for multiobjective economic emission load dispatch problem under fuzziness," *Swarm and Evolutionary Computation,* vol. 18, pp. 11-21, 2014.

[64] S. Nakrani and C. Tovey, "On honey bees and dynamic server allocation in internet hosting centers," *Adaptive Behavior,* vol. 12, pp. 223-240, 2004.

[65] M. Basu, "Bee colony optimization for combined heat and power economic dispatch," *Expert Systems With Applications,* vol. 38, pp. 13527-13531, 2011.

[66] C. Cerrone, R. Cerulli, and B. Golden, "Carousel greedy: A generalized greedy algorithm with applications in optimization," *Computers & Operations Research,* vol. 85, pp. 97-112, 2017.

[67] M. Dell'Orco, M. Marinelli, and M. Altieri, "Solving the gate assignment problem through the fuzzy bee colony optimization," *Transportation Research Emerging Technologies,* vol. 80, pp. 424-438, 2017.

[68] H. Shayeghi and A. Ghasemi, "A modified artificial bee colony based on chaos theory for solving non-convex emission/economic dispatch," *Energy Conversion and Management,* vol. 79, pp. 344-354, 2014.

[69] A. Rajasekhar, N. Lynn, S. Das, and P. Suganthan, "Computing with the collective intelligence of honey bees–A survey," *Swarm and Evolutionary Computation,* vol. 32, pp. 25-48, 2017.

[70] X. Yang and S. Deb, "*Cuckoo Search via Levy Flights*," in *Proceedings of World Congress on Natural & Biologically Inspired Computing*, Coimbatore, India, 2010, pp. 210-214.

[71] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *Journal of Computational Physics,* vol. 226, pp. 1830-1844, 2007.

[72] A. Gandomi, X. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers and Mathematics with Applications,* vol. 63, p. 191, 2012.

[73] X. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers and Operations Research,* vol. 40, p. 1616-1624, 2013.

[74] A. Gandomi, X. Yang, and A. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *An International Journal for Simulation-Based Engineering,* vol. 29, pp. 17-35, 2013.

[75] H. Soneji and R. Sanghvi, "*Towards the improvement of cuckoo search algorithm*," in *Proceedings of World Congress on Information and Communication Technologies*, Tivandrun, India, 2012, pp. 878-883.

[76] X. Yang, "A New Metaheuristic Bat-Inspired Algorithm," in *Proceedings of Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Berlin, Germany, 2010, pp. 65-74.

[77] B. Adarsh, T. Raghunathan, T. Jayabarathi, and X. Yang, "Economic dispatch using chaotic bat algorithm," *Energy,* vol. 96, pp. 666-675, 2016.

[78] S. Chakraborty, T. Senjyu, A. Yona, A. Saber, and T. Funabashi, "Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation," *IET Generation, Transmission & Distribution,* vol. 5, pp. 1042-1052, 2011.

[79] E. Elattar, "A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem," *International Journal of Electrical Power & Energy Systems,* vol. 69, pp. 18-26, 2015.

[80] D. He, F. Wang, and Z. Mao, "Hybrid genetic algorithm for economic dispatch with valve-point effect," *Electric Power Systems Research,* vol. 78, pp. 626-633, 2008.

[81] B. Mohammadi-Ivatloo, A. Rabiee, and A. Soroudi, "Nonconvex dynamic economic power dispatch problems solution using hybrid immune-genetic algorithm," *IEEE Systems Journal,* vol. 7, pp. 777-785, 2013.

[82] L. Wang and L. Li, "An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems," *International Journal of Electrical Power & Energy Systems,* vol. 44, pp. 832-843, 2013.

[83] R. Parouha and K. Das, "DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints," *Expert Systems with Applications,* vol. 63, pp. 295-309, 2016.

[84] S. Duman, N. Yorukeren, and I. Altas, "A novel modified hybrid PSOGSA based on fuzzy logic for non-convex economic dispatch problem with valve-point effect," *International Journal of Electrical Power & Energy Systems,* vol. 64, pp. 121-135, 2015.

[85] A. Yadav and K. Deep, "An efficient co-swarm particle swarm optimization for non-linear constrained optimization," *Journal of Computational Science,* vol. 5, pp. 258-268, 2014.

[86] V. Hosseinnezhad and E. Babaei, "Economic load dispatch using θ-PSO," *International Journal of Electrical Power & Energy Systems,* vol. 49, pp. 160-169, 2013.

[87] J. Chen, J. Zheng, P. Wu, L. Zhang, and Q. Wu, "Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems," *Expert Systems with Applications,* vol. 86, pp. 208-223, 2017.

[88] N. Singh, J. Dhillon, and D. Kothari, "Synergic predator-prey optimization for economic thermal power dispatch problem," *Applied Soft Computing,* vol. 43, pp. 298-311, 2016.

[89] M. Ghasemi, M. Taghizadeh, S. Ghavidel, and A. Abbasian, "Colonial competitive differential evolution: An experimental study for optimal economic load dispatch," *Applied Soft Computing,* vol. 40, pp. 342-363, 2016.

[90] M. Zaman, S. Elsayed, T. Ray, and R. Sarker, "Evolutionary algorithms for dynamic economic dispatch problems," *IEEE Transactions on Power Systems,* vol. 31, pp. 1486-1495, 2016.

[91] M. Basu, "Modified particle swarm optimization for nonconvex economic dispatch problems," *International Journal of Electrical Power & Energy Systems,* vol. 69, pp. 304-312, 2015.

[92] M. Basu, "Improved differential evolution for economic dispatch," *International Journal of Electrical Power & Energy Systems,* vol. 63, pp. 855-861, 2014.

[93] A. Srinivasa Reddy and K. Vaisakh, "Shuffled differential evolution for economic dispatch with valve point loading effects," *International Journal of Electrical Power & Energy Systems,* vol. 46, pp. 342-352, 2013.

[94] B. Mohammadi-Ivatloo, A. Rabiee, A. Soroudi, and M. Ehsan, "Iteration PSO with time varying acceleration coefficients for solving non-convex economic dispatch problems," *International Journal of Electrical Power & Energy Systems,* vol. 42, pp. 508-516, 2012.

[95] Z. Wu, J. Ding, Q. Wu, Z. Jing, and X. Zhou, "Two-phase mixed integer programming for non-convex economic dispatch problem with spinning reserve constraints," *Electric Power Systems Research,* vol. 140, pp. 653-662, 2016.

[96] R. Rizk-Allah, R. El-Sehiemy, and G. Wang, "A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution," *Applied Soft Computing,* vol. 63, pp. 206-222, 2018.

[97] M. Pradhan, P. K. Roy, and T. Pal, "Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system," *Ain Shams Engineering Journal,* 2017. In press.

[98] M. Kheshti, X. Kang, Z. Bie, Z. Jiao, and X. Wang, "An effective lightning flash algorithm solution to large scale non-convex economic dispatch with valve-point and multiple fuel options on generation units," *Energy,* vol. 129, pp. 1-15, 2017.

[99] T. Ding and Z. Bie, "Parallel augmented Lagrangian Relaxation for dynamic economic dispatch using diagonal quadratic approximation method," *IEEE Transactions on Power Systems,* vol. 32, pp. 1115-1126, 2017.

[100] T. Jayabarathi, T. Raghunathan, B. Adarsh, and P. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy,* vol. 111, pp. 630-641, 2016.

[101] D. Secui, "A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point effects," *Energy,* vol. 113, pp. 366-384, 2016.

[102] A. Meng, J. Li, and H. Yin, "An efficient crisscross optimization solution to large-scale non-convex economic load dispatch with multiple fuel types and valve-point effects," *Energy,* vol. 113, pp. 1147-1161, 2016.

[103] W. Elsayed, Y. Hegazy, F. Bendary, and M. El-bages, "Modified social spider algorithm for solving the economic dispatch problem," *Engineering Science and Technology, an International Journal,* vol. 19, pp. 1672-1681, 2016.

[104] M. Al-Betar, M. Awadallah, A. Khader, and A. Bolaji, "Tournament-based harmony search algorithm for non-convex economic load dispatch problem," *Applied Soft Computing,* vol. 47, pp. 449-459, 2016.

[105] A. Kavousi-Fard and A. Khosravi, "An intelligent θ-modified bat algorithm to solve the non-convex economic dispatch problem considering practical constraints," *International Journal of Electrical Power & Energy Systems,* vol. 82, pp. 189-196, 2016.

[106] A. Abdelaziz, E. Ali, and S. Abd-Elazim, "Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems," *Energy,* vol. 101, pp. 506-518, 2016.

[107] M. Singh and J. Dhillon, "Multiobjective thermal power dispatch using opposition-based greedy heuristic search," *International Journal of Electrical Power & Energy Systems,* vol. 82, pp. 339-353, 2016.

[108] A. Barisal and R. Prusty, "Large scale economic dispatch of power systems using oppositional invasive weed optimization," *Applied Soft Computing,* vol. 29, pp. 122-137, 2015.

[109] W. Elsayed and E. El-Saadany, "A fully decentralized approach for solving the economic dispatch problem," *IEEE Transactions on Power Systems,* vol. 30, pp. 2179-2189, 2015.

[110] M. Wang, H. Gooi, S. Chen, and S. Lu, "A mixed integer quadratic programming for dynamic economic dispatch with valve point effect," *IEEE Transactions on Power Systems,* vol. 29, pp. 2097-2106, 2014.

[111] Z. Zhan, J. Zhang, Y. Li, and Y. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 15, pp. 832-847, 2011.

[112] Z. Ren, A. Zhang, C. Wen, and Z. Feng, "A scatter learning particle sSwarm optimization algorithm for multimodal problems," *IEEE Transactions on Cybernetics,* vol. 44, pp. 1127-1140, 2014.

[113] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation*, Washington, USA, 1999, pp. 1931-1938.

[114] N. Awad, M. Ali, and R. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 1098-1105.

[115] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part C,* vol. 36, pp. 515-519, 2006.

[116] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation,* Anchorage, USA, 1998, pp. 69-73.

[117] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation,* La Jolla, USA, 2000, pp. 84-88 .

[118] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation,* Washington, USA, pp. 1945-1950.

[119] J. Liu, Y. Mei, and X. Li, "An analysis of the inertia weight parameter for binary particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 20, pp. 666-681, 2016.

[120] T. Victoire and A. Jeyakumar, "Discussion of particle swarm optimization to solving the economic dispatch considering the generator constraints", *IEEE Transactions on Power Systems,* vol. 19, pp. 2121-2122, 2004.

[121] G. Zwe-Lee, "Closure to discussion of particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Transactions on Power Systems,* vol. 19, pp. 2122-2123, 2004.

[122] S. I. Grossman, *Elementary linear algebra*, 5th ed.: Saunders College Pub., 1994.

[123] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger*, et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report 2005, IIT Kanpur, India,* 2005. Available: https://www.lri.fr/~hansen/Tech-Report-May-30-05.pdf

[124] K. Tang, X. Yáo, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen*, et al.*, "Benchmark functions for the CEC'2008 special session and competition on large scale global optimization," *Nature Inspired Computation and Applications Laboratory, USTC, China,* pp. 153-177, 2007. Available: http://www.al-roomi.org/multimedia/CEC_Database/CEC2008/CEC2008_TechnicalReport.pdf

[125] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization," *Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore,* vol. 201212, 2013. Available: http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf

[126] T. Adhinarayanan and M. Sydulu, "Efficient Lambda logic based optimisation procedure to solve the large scale generator constrained economic dispatch problem," *Journal of Electrical Engineering and Technology,* vol. 4, pp. 301-309, 2009.

[127] J. Liang, B. Qu, P. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization," *Technical Report 201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore,* 2014. Available: http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2015/CEC2015.htm

Appendices

Appendix-1

Here, the Papers A-H are appended as follows.

Paper A: IEEE-IJCNN 2015

Paper B: IEEE-SMC 2015

Paper C: IEEE-IJCNN 2016

Paper D: IEEE-ISGT-Asia 2016

Paper E: Swarm and Evolutionary Computation 2017

Paper F: Applied Soft Computing 2017

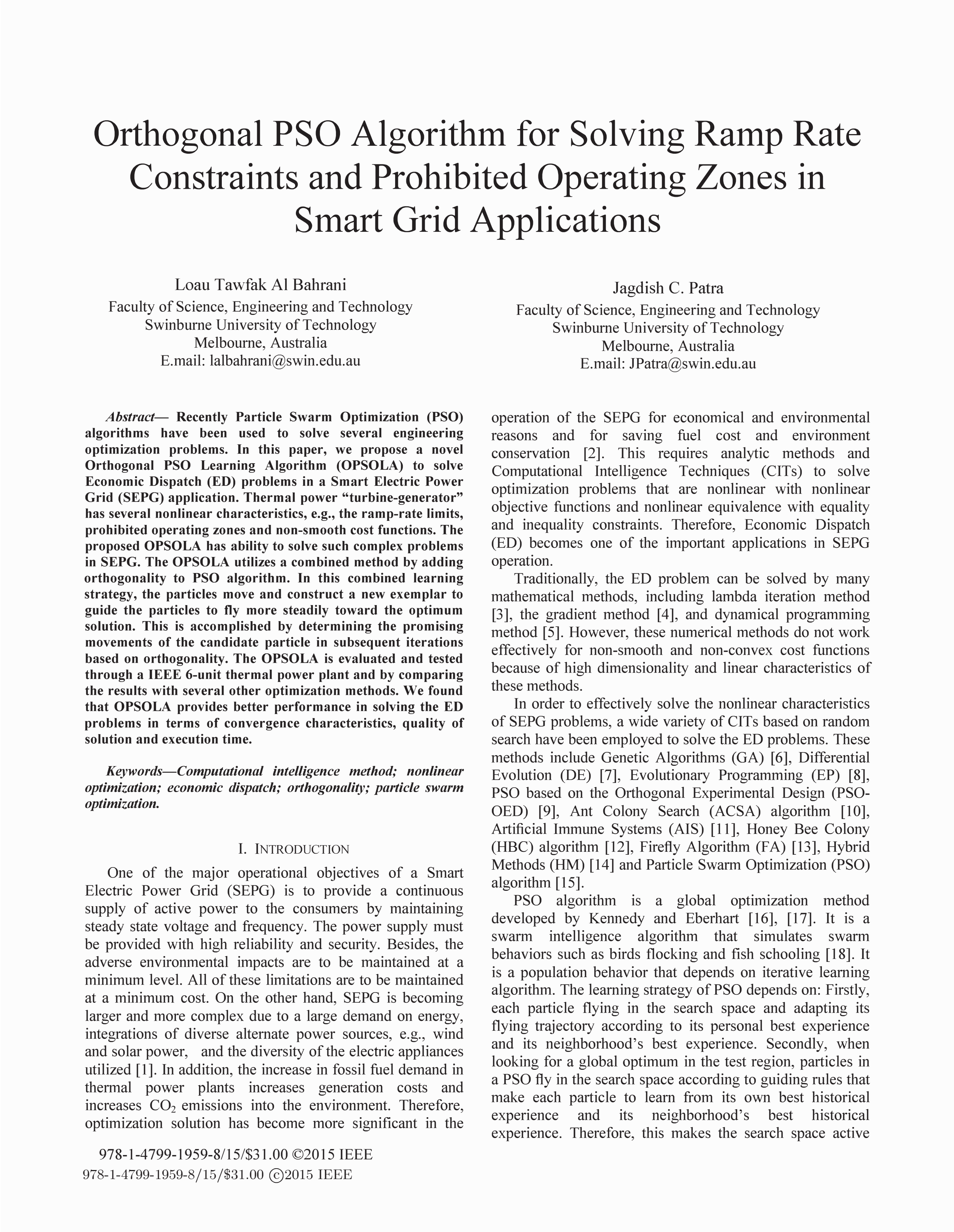
Paper G: Energy 2018

Paper H: Applied Soft Computing 2018 (Under Review)

Paper A: IEEE-IJCNN 2015

Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid application

Published in:	International Joint Conference on Neural Network (IJCNN) 2015
Date of Conference:	12-17 July 2015
Date Added to IEEE *Xplore*:	01 October 2015
Electronic ISBN:	978-1-4799-1960-4
DVD ISBN:	978-1-4799-1959-8
Print on Demand (PoD) ISBN:	978-1-4799-1961-1
Electronic ISSN:	2161-4407
Print ISSN:	2161-4393
INSPEC Accession Number:	15503877
DOI:	**10.1109/IJCNN.2015.7280429**
Publisher:	IEEE
Conference Location:	Killarney, Ireland



Orthogonal PSO Algorithm for Solving Ramp Rate Constraints and Prohibited Operating Zones in Smart Grid Applications

Loau Tawfak Al Bahrani
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia
E.mail: lalbahrani@swin.edu.au

Jagdish C. Patra
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia
E.mail: JPatra@swin.edu.au

***Abstract*— Recently Particle Swarm Optimization (PSO) algorithms have been used to solve several engineering optimization problems. In this paper, we propose a novel Orthogonal PSO Learning Algorithm (OPSOLA) to solve Economic Dispatch (ED) problems in a Smart Electric Power Grid (SEPG) application. Thermal power "turbine-generator" has several nonlinear characteristics, e.g., the ramp-rate limits, prohibited operating zones and non-smooth cost functions. The proposed OPSOLA has ability to solve such complex problems in SEPG. The OPSOLA utilizes a combined method by adding orthogonality to PSO algorithm. In this combined learning strategy, the particles move and construct a new exemplar to guide the particles to fly more steadily toward the optimum solution. This is accomplished by determining the promising movements of the candidate particle in subsequent iterations based on orthogonality. The OPSOLA is evaluated and tested through a IEEE 6-unit thermal power plant and by comparing the results with several other optimization methods. We found that OPSOLA provides better performance in solving the ED problems in terms of convergence characteristics, quality of solution and execution time.**

Keywords—Computational intelligence method; nonlinear optimization; economic dispatch; orthogonality; particle swarm optimization.

I. Introduction

One of the major operational objectives of a Smart Electric Power Grid (SEPG) is to provide a continuous supply of active power to the consumers by maintaining steady state voltage and frequency. The power supply must be provided with high reliability and security. Besides, the adverse environmental impacts are to be maintained at a minimum level. All of these limitations are to be maintained at a minimum cost. On the other hand, SEPG is becoming larger and more complex due to a large demand on energy, integrations of diverse alternate power sources, e.g., wind and solar power, and the diversity of the electric appliances utilized [1]. In addition, the increase in fossil fuel demand in thermal power plants increases generation costs and increases CO_2 emissions into the environment. Therefore, optimization solution has become more significant in the operation of the SEPG for economical and environmental reasons and for saving fuel cost and environment conservation [2]. This requires analytic methods and Computational Intelligence Techniques (CITs) to solve optimization problems that are nonlinear with nonlinear objective functions and nonlinear equivalence with equality and inequality constraints. Therefore, Economic Dispatch (ED) becomes one of the important applications in SEPG operation.

Traditionally, the ED problem can be solved by many mathematical methods, including lambda iteration method [3], the gradient method [4], and dynamical programming method [5]. However, these numerical methods do not work effectively for non-smooth and non-convex cost functions because of high dimensionality and linear characteristics of these methods.

In order to effectively solve the nonlinear characteristics of SEPG problems, a wide variety of CITs based on random search have been employed to solve the ED problems. These methods include Genetic Algorithms (GA) [6], Differential Evolution (DE) [7], Evolutionary Programming (EP) [8], PSO based on the Orthogonal Experimental Design (PSO-OED) [9], Ant Colony Search (ACSA) algorithm [10], Artificial Immune Systems (AIS) [11], Honey Bee Colony (HBC) algorithm [12], Firefly Algorithm (FA) [13], Hybrid Methods (HM) [14] and Particle Swarm Optimization (PSO) algorithm [15].

PSO algorithm is a global optimization method developed by Kennedy and Eberhart [16], [17]. It is a swarm intelligence algorithm that simulates swarm behaviors such as birds flocking and fish schooling [18]. It is a population behavior that depends on iterative learning algorithm. The learning strategy of PSO depends on: Firstly, each particle flying in the search space and adapting its flying trajectory according to its personal best experience and its neighborhood's best experience. Secondly, when looking for a global optimum in the test region, particles in a PSO fly in the search space according to guiding rules that make each particle to learn from its own best historical experience and its neighborhood's best historical experience. Therefore, this makes the search space active

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

and efficient. A particle while choosing the neighborhood's best historical experience, either uses the best historical experience of the whole swarm as its neighborhood's best historical experience or uses the best historical experience of the particle in its neighborhood which is determined by some topological structure, e.g., the ring structure and the pyramid structure [9]. In both cases, the learning strategy of a particle's best experience and its neighborhood's best experience is obtained through a linear combination. However, this linear operation will negatively reflect to the performance of learning strategy of PSO due to oscillation phenomenon which causes inefficiency to the search ability of the algorithm and delays in convergence.

Some Evolutionary Algorithms (EAs) attempted to overcome the drawback of the PSO algorithm by overtaking the oscillations. One of these methods is PSO-OED [9]. The PSO-OED uses the effect of several factors simultaneously and the best combination of factor levels is found by conducting several tests. However, the main drawback of OED is that it holds only when no or weak interaction of factors exists. It means *m* particles have Linear Combination Property (LCP) among vectors in order to have optimal solution. This limitation of OED will make PSO search effective on "unimodal" or simple problems. On the other hand, it is very vulnerable on "complex multimodal" problems [20].

Our proposed OPSOLA differs from PSO-OED in the application of a new learning strategy. This strategy is based on formation of orthogonal particle vectors that are found in the *d*-dimensional searching area and it does not depend on LCP. We used the orthogonal property as a guide to improve the PSO performance by improving particle's position and velocity. This improvement is due to the fact that the particle's movements will be in clear directions, through the movements of particles in orthogonal displacements from one level to another level based on the subsequent iterations. Therefore, the time duration that the particles move toward the optimal solution becomes shorter by following orthogonal particles that have best historical experience. We have applied OPSOLA to solve nonlinear characteristics of the generator with the power constraints to solve the ED problem. With extensive experimental results, we have shown superior performance of OPSOLA compared to several competing algorithms.

II. Problem Formulation

In this Section we introduce several constraints and nonlinearities used in the ED problem.

A. *Objective Cost Function*

In the ED problem, the optimal solution is to determine the active output power of each online generator and to determine the lowest total fuel cost of all generators over a period of time, while maintaining various nonlinear constraints [21]. The cost objective function is given by

$$\text{Minimize } F_{cost} = \sum_{j=1}^{N_{gen}} F(P_j) \qquad (1)$$

where $F(P_j)$ is the cost function of the *j*th generating unit in (\$/h), P_j is the active output power of the *j*th generating unit in (MW), and N_{gen} is the number of online generators of the power system. The cost function of each generating unit is related to the real power delivered into the system, and its performace is given by a quadratic function [22]

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \qquad (2)$$

where a_j, b_j, and c_j are the cost coefficients of *j*th generating unit.

B. *Constraints of the Power System in the SEPG*

Differnet power system constraints used this work are explained below.

Constraint # 1: Power Equilibrium Constraint

The constraint for total power balance is expressed by

$$\sum_{j=1}^{N_{gen}} P_j = P_D + P_L \qquad (3)$$

The constraint of the balance equation can be stated as the total system power generation equals to the load demand (P_D) in (MW) of the system plus the transmission network loss (P_L) in (MW). P_L is expressed of the active output power of the generators that can be represented using *B*-coefficients. P_L is usually approximated by Kron's loss formula [23] which is given by

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k + \sum_{j=1}^{N_{gen}} P_j B_{j0} + B_{00} \qquad (4)$$

where *j*, *k* = 1, 2, ..., N_{gen} are the indices of the generating units , and B_{jk}, B_{j0}, B_{00} are known as the loss coefficients or *B*-Coefficients [22].

Constraint # 2: Generation Limits

The generation limit is given by

$$P_j \min < P_j < P_j \max, \quad (j = 1, 2, ..., N_{gen}) \qquad (5)$$

This requires that the power generation of each unit varies between its minimum $P_j\text{min}$ and its maximum $P_j$max production limits.

Constraint # 3: Ramp Rate Limits

In real operation, the operating range of all online generating units is restricted by their ramp rate limits due to the physical limitation of the generating thermal units [24]. The inequality constraints due to ramp limits are given by

i) If power generation increases, then

$$P_j - P_j^0 \le UR_j \qquad (6)$$

ii) If power generation decreases, then

$$P_j^0 - P_j \le DR_j \qquad (7)$$

where P_j^0 in (MW) is the previous active output power, UR_j in (MW/h) is the up-ramp limit of the *j*th generator; and DR_j in (MW/h) is the down-ramp limit of the *j*th generator.

Constraint # 4: Porhibited Operating Zones

Due to the steam valve operations or vibration in a shaft bearings, the system contains some prohibited operating zones. Due to these constraints, discontinuities are produced in cost curves corresponding to the prohibited operating zones. Therefore, all the online generators in the system must avoid prohibited operating zones. Consequently, considering the constraint #2 in (5), the feasible operating zones of the *j*th generating thermal units are given by [22]:

$$\begin{aligned} &P_j \min \le P_j \le P_{j,1}^l \\ &P_{j,k-1}^u \le P_j \le P_{j,k}^l \ , \quad k = 2, 3, ..., N_{pz,j} \\ &P_{N_{pz,j}}^u \le P_j \le P_j \max \end{aligned} \tag{8}$$

where, $P_{j,k}^l$ and $P_{j,k}^u$ are the lower and upper bounds of the *k*th prohibited zone of the *j*th generating unit, and $N_{pz,j}$ is the number of prohibited zones of the *j*th generating unit.

Furthermore, in addition to the above constraints that face the power system, combining (6) and (7) with (5) results in

$$P_j^{low} \le P_j \le P_j^{high} \tag{9}$$

$$P_j^{low} = \max\left(P_j \min, (P_j^0 - DR_j)\right) \tag{10}$$

$$P_j^{high} = \min\left(P_j \max, (P_j^0 + UR_j)\right) \tag{11}$$

where P_j^{low} and P_j^{high} are the new lower and upper limits of unit *j*, respectively. Incorporating these constraints we obtain the final set of constraints as given below:

$$\begin{aligned} &P_j^{low} \le P_j \le P_{j,1}^l, \\ &P_{j,k-1}^u \le P_j \le P_{j,k}^l \ , \quad k = 2, 3, ..., N_{pz,j}, \\ &P_{N_{pz,j}}^u \le P_j \le P_j^{high} \end{aligned} \tag{12}$$

III. Orthogonal PSO Learning Algorithm

Here we briefly introduce the PSO algorithm and explain the proposed OPSOLA.

A. The PSO Algorithm

Let us consider a swarm population with *m* particles searching for solution in a *d*-dimensional space. Thus, the number of particles in the swarm, $N_{particle} = m$. Each particle of the swarm has one *d*-dimensional position vector and one *d*-dimensional velocity vector. The position vector represents a possible solution. The position vector X_i and the velocity vector V_i of the *i*th (i = 1, 2, …, m) particle are given by

$$X_i = [x_{i1},\ x_{i2},\ \ldots,\ x_{id}] \tag{13}$$

$$V_i = [v_{i1}, v_{i2},\ \ldots,\ v_{id}] \tag{14}$$

In addition, each particle *i* will keep its personal historical best position vector, $H_{i,pers}$ given by

$$H_{i,pers} = [h_{pi,1},\ h_{pi,2},\ \ldots,\ h_{pi,d}] \tag{15}$$

One of the important frameworks for solving PSO is the ring structure [9]. The behavior of the neighborhood of a particle *i* includes *i-1, i,* and *i+1* in a ring structure. The best position among all the particles in the *i*th particle's neighborhood is given by

$$H_{i,neigh} = [h_{ni,1},\ h_{ni,2},\ \ldots,\ h_{ni,d}] \tag{16}$$

The vectors V_i and X_i are initialized randomly. These are updated in each iteration through the guidance of $H_{i,pers}$ and $H_{i,neigh}$ according to (17) and (18).

$$V_i = V_i + c_1 r_{1i} (H_{i,pers} - X_i) + c_2 r_{2i} (H_{i,neigh} - X_i) \tag{17}$$

$$X_i = X_i + V_i \tag{18}$$

where c_1 and c_2 are coefficients of particle acceleration, usually positive values, and are chosen experimentally from the range [1, 2]. The r_{1i} and r_{2i} are two randomly generated values within range [0, 1]. In each iteration, considering all *m* particles, the best personal position vector and the best neighbor position vector are recorded. Finally, at the end of iterations, the optimal solution is given by the best neighbor position vector.

B. The Proposed OPSOLA

In PSO, each particle updates its velocity and position vectors according to its personal best position and its neighborhood's best position. However, this learning strategy can cause "Oscillation" phenomenon [9]. The oscillations occur due to linear combination of the personal effect and the neighborhood effect. Subsequently, this leads to slow convergence and non-optimal solution, especially for problems involving high *d*-dimensional space. In order to overcome these limitations, OPSOLA has been developed.

The OPSOLA differs from the PSO algorithm and PSO-OED. We incorporate a new strategy called "*Orthogonal Particle Formation*" (OPF) in the *d*-dimensional search space. In a swarm population with *m* particles, each particle *i* has a position vector X_i and a velocity vector V_i, as given in (13) and (14), respectively. The OPSOLA updates the position and velocity vectors in each iteration using the following steps.

***Step* 1:** For each particle, evaluate the cost using the position vector in the cost function (1). Select the best particle that has the minimum cost. Let us denote it's position vector as X_{best}. Similarly, obtain the next *d*-1 best position vectors.

***Step* 2:** Construct a matrix *A* with *m*×1 rows and *d* columns. The first row of $A = X_{best}$. Substitute the velocity vectors V_i (i = 1, 2, ..., m) in second to *m*+1 rows of *A* matrix. Convert the matrix A to matrix B, such that B is a symmetrical matrix with size $d\times d$, using the pseudocode given below:

```
Pseudocode: convert to symmetrical matrix
for $i = 1 : d$
    $B(1,i) = A(1,i)$
    $B(i,1) = A(1,i)$
end for
for $k = 2 : d$
    for $i = 2 : d$
        $B(k,i) = A(k,i)$
        $C(k,i) = B(k,i)$
        $B(i,k) = C(k,i)$
    end for
end for
```

***Step* 3:** Convert B to an orthogonal matrix C of dimension $d\times d$, using Gram–Schmidt orthogonalization method.

***Step* 4:** Obtain a diagonal matrix D of dimension $d\times d$ using (19)

$$D = CBC^T \tag{19}$$

Let D_k denotes the kth row of the matrix D. Now let us obtain a d number of orthogonal d-dimensional vectors, H_{ok} as given in (20)

$$H_{ok} = D_k \tag{20}$$

***Step* 5:** Update the position and velocity vectors of d number of particles out of m particles. Thus, the d best position and velocity vectors as obtained in Step 1 are updated as follows

$$V_k = V_k + c\,r_t\,(H_{ok} - X_k) \tag{21}$$

$$X_k = X_k + V_k \tag{22}$$

where c is an acceleration factor and set to 2 and r_t is a random number within range [0, 1] and $k = 1, 2, \ldots, d.$

At the end of iteration, the position vector X_{best} as computed in Step 1 provides the optimal solution. Instead of using the neighbor position vector and the personal position vector as done in PSO algorithm in linear summation (17), in OPSOLA we update the position and velocity vectors by constructing a new orthogonal vector H_{ok}. This means that only one orthogonal vector is used to control movement of each particle. Thus, the particles move in one direction based on the orthogonal vector H_{ok}.

A flowchart of the OPSOLA with major computational steps is shown in Fig. 1.

C. *A Simple Example*

To explain PSO and OPSOLA algorithms, a simple example is illustrated. Consider a 2-dimentional function given by $f(x, y) = x^2+y^2+3$. The objective is to find values x and y so the value of the function $f(x, y)$ is minimized. With a range of [-100 100], for x and y, the function is plotted in Fig. 2. Note that the function minimized to $f = 3$ when $x = 0$ and $y = 0$. Both the PSO and OPSOLA have 10 particles in the swarm and uses10 iterations. Figure 3 shows that the PSO algorithm moves forward and backward between 3rd iteration (-44.3, 31.6) and 4th iteration (16, -10.2) causing oscillations. Whereas, in the OPSOLA algorithm, the particles move steadily from the initial position to the solution in the 10th iteration.

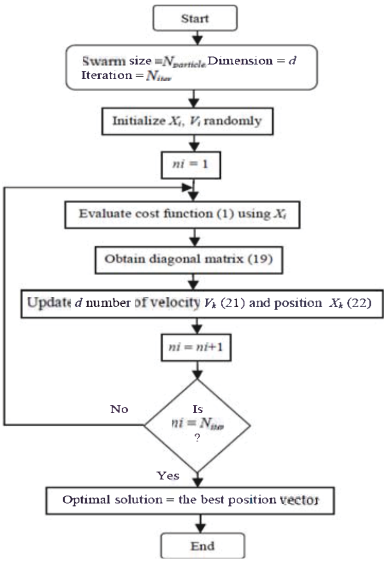

Start
Swarm size =$N_{particle}$ Dimension = d
Iteration = N_{iter}
Initialize X_i, V_i randomly
ni = 1
Evaluate cost function (1) using X_i
Obtain diagonal matrix (19)
Update d number of velocity V_k (21) and position X_k (22)
ni = ni+1
No
Is $ni = N_{iter}$?
Yes
Optimal solution = the best position vector
End

Fig. 1. Flowchart of the proposed OPSOLA.

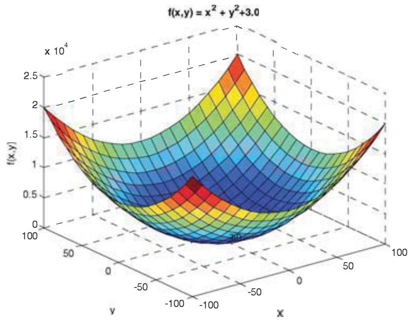


Fig. 2. Plot of function $f(x, y)$.

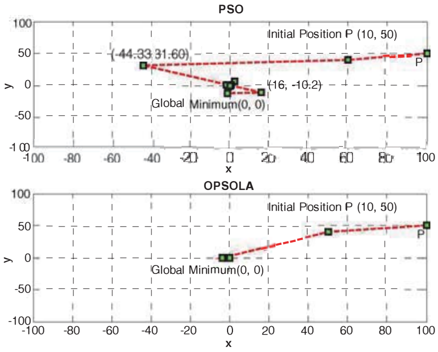


Fig. 3. Movement of one particle with iteration.

IV. Application of OPSOLA to ED Problem

Here we describe the simulation results carried out on the 6-unit generator power system with several constraints.

A. The Power System Specification

In order to show performance of the OPSOLA to solve the ED problem, a practical 6-unit power system was used. The single line diagram is shown in Fig. 4 [22]. This system consists of six thermal power units, 26-bus, and 46-transmission line. The generation parameters are given in Table I [24]. The load demand is 1263 MW. The *B*-loss coefficient matrix is listed Table II. B_{jk}, B_{j0}, B_{00} are taken into account with calculations. The power system is a small-scale one and generating units 1, 2, 3, 4, 5 and 6 have a total of 12 prohibited zones. Thus, there are 13 inequality constraints as described by (12) for the ED problem.

B. Experimental Results

Several evolutionary optimization methods that have been tested on the 6-unit power system by other authors are listed in Table III. The different methods are: ACSA [10], BCO [12], FA [13], Random Drift Particle Swarm Optimization (RDPSO) [22], PSO with inertia weight [23], Binary-Coded Algorithm GA [24], DE [25], Artificial Immune System (AIS) [26], Standard PSO (SPSO) with constriction and inertia weight [27], Hybrid Gradient Descent PSO (HGPSO) [27], Hybrid PSO with Mutation (HPSOM) [27], Hybrid PSO with Wavelet Mutation (HPSOWM) [27], Hopfield Neural Network (NN) [28], Chaotic PSO (CPSO) [29], and Anti-Predatory PSO (APSO) [30].

The results in Table III show the minimum, maximum and mean cost of generation. In addition, the standard deviation (σ) of the mean cost function for each algorithm is also listed. Two experiments were simulated. First experiment consists of a swarm with $N_{particle}$ = 100, N_{run} = 100 and N_{iter} = 200. The second experiment consists of a swarm with $N_{particle}$ = 20, N_{run} = 100 and N_{iter} = 1000. For the same category of the evolutionary algorithms, the best results were achieved by RDPSO algorithm with minimum, maximum and mean cost, while, CPSO algorithm provided the best performance in σ calculation. Table IV lists the performance of the proposed OPSOLA in terms of the minimum, maximum and mean cost of generation. The results show that OPSOLA provides the best in terms of minimum, maximum and mean costs among the tested evolutionary techniques and has the lowest σ of the mean cost function.

Figure 5 shows the convergence characteristics of OPSOLA on the ED problem of 6-unit power system. The algorithm was tested using a swarm population with $N_{particle}$ = 20, N_{run} = 10 and 20 and N_{iter} = 50. We found the execution time of the OPSOLA for 10 runs and 20 runs as 2.30 and 8.51 seconds, respectively. The OPSOLA settles at about 20 iterations for both runs. OPSOLA provides a steady performance without deviation and oscillation. Figure 5(b) shows the steady state performance of OPSOLA at different instances of runs showing evidence of stable and robust optimal solution.

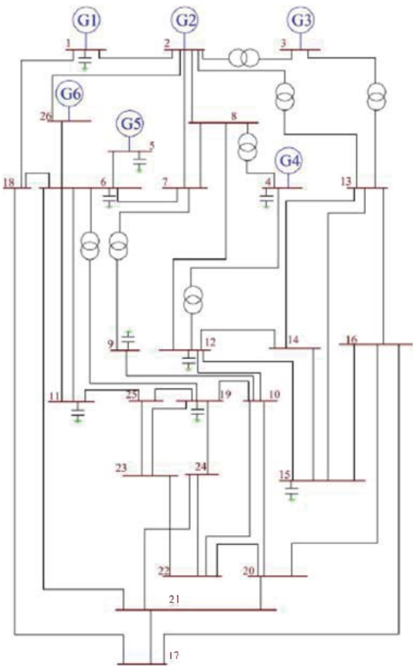


Fig. 4. Single line diagram of IEEE 6-unit thermal power system.

TABLE I. GENERATING UNIT RAMP RATE AND PROHIBITED ZONES

Unit	P_j^0 MW	$P_j min$ MW	$P_j max$ MW	a_i $	b_i $/MW	c_i $/MW^2	UR_j MW/h	DR_j MW/h	Prohibited Zones
1	440	100	500	240	7.0	0.0070	80	120	[210, 240] [350, 380]
2	170	50	200	200	10.0	0.0095	50	90	[90, 110] [140, 160]
3	200	80	300	220	8.5	0.0090	65	100	[150, 170] [210, 240]
4	150	50	150	200	11.0	0.0090	50	90	[80, 90] [110, 120]
5	190	50	220	220	10.5	0.0080	50	90	[90, 110] [140, 150]
6	1100	50	120	190	12.0	0.0075	50	90	[75, 85] [100, 105]

TABLE II. *B*-LOSS COEFFICIENTS OF THE 6-UNIT SYSTEM

B_{jk}	1	2	3	4	5	6
1	0.0017	0.0012	0.0007	-0.0001	-0.0005	-0.0002
2	0.0012	0.0014	0.0009	0.0001	-0.0006	-0.0001
3	0.0007	0.0009	0.0031	0.0000	-0.0010	-0.0006
4	-0.0001	0.0001	0.0000	-0.0024	0.0006	-0.0008
5	-0.0005	-0.0006	-0.0001	-0.0006	0.0129	-0.0002
6	-0.0002	-0.0001	0.0006	-0.0008	0.0002	-0.0150
B_{0j}	-0.0004	-0.0001	0.0007	0.0001	0.0002	-0.0007
B_{00}	0.0560					

Since each particle moves by orthogonal steps in each iteration, the OPSOLA provides a strong convergence and steady performance. In order to explain the behavior of OPSOLA, we divide Figure 5(a) into three regions.

TABLE III. PERFORMANCE OF DIFFERENT ALGORITHMS

Algorithm	Minimum Cost ($/h)	Maximum Cost ($/h)	Mean Cost ($/h)	σ
NN [28]	15485.90	15485.90	15485.90	0.00
$N_{particle}$ = 100, N_{run} =*100*, N_{iter} = 200				
ACSA [10]	15445.30	15511.52	15459.51	12.02
BCO [12]	15444.58	15482.39	15457.94	8.48
FA [13]	15445.94	15501.39	15461.30	9.33
RDPSO [22]	**15442.75**	**15455.29**	**15445.02**	**2.28**
PSO [23]	15444.77	15483.97	15466.56	7.91
GA [24]	15445.50	15491.50	15465.20	9.73
DE [25]	15444.94	15472.06	15450.13	6.98
AIS [26]	15446.32	15481.27	15456.66	7.39
SPSO [27]	15443.01	15490.26	15452.47	9.53
HGPSO [27]	15447.10	15497.03	15462.61	10.64
HPSOM [27]	15443.62	15497.86	15449.26	6.27
HPSOWM[27]	15442.82	15502.63	15455.62	15.88
CPSO [29]	15442.98	15466.39	15449.12	5.80
APSO [30]	15444.51	15538.60	15473.31	12.90
$N_{particle}$ = 20, N_{run} =*100*, N_{iter} = 1000				
ACSA [10]	15445.30	15511.52	15462.51	12.02
BCO [12]	15446.37	15503.29	15461.84	11.66
FA [13]	15447.19	15512.00	15466.32	12.55
RDPSO [22]	**15442.78**	**15484.86**	**15453.72**	**13.50**
PSO [23]	15443.83	15529.00	15459.96	13.37
GA [24]	15446.47	15493.20	15465.99	10.70
DE [25]	15442.98	15489.89	15455.25	13.74
AIS [26]	15443.16	15481.06	15458.08	8.58
SPSP [27]	15442.91	15597.45	15477.96	33.71
HGPSO [27]	15445.40	15494.03	15464.39	11.56
HPSOM [27]	15443.17	15533.17	15464.84	22.84
HPSOWM[27]	15442.99	15596.02	15466.33	27.32
CPSO [29]	15443.12	15460.48	15454.17	8.05
APSO [30]	15444.89	15493.37	15459.40	11.03

TABLE IV. PERFORMANCE OF THE PROPOSED OPSOLA

Algorithm	Minimum Cost ($/h)	Maximum Cost ($/h)	Mean Cost ($/h)	σ
OPSOLA $N_{particle}$ = 20 N_{run} = 10, N_{iter}=50	15303.00	15367.00	15324.00	0
OPSOLA $N_{particle}$ = 20 N_{run} = 20, N_{iter}=50	15306.00	15510.00	15359.00	0

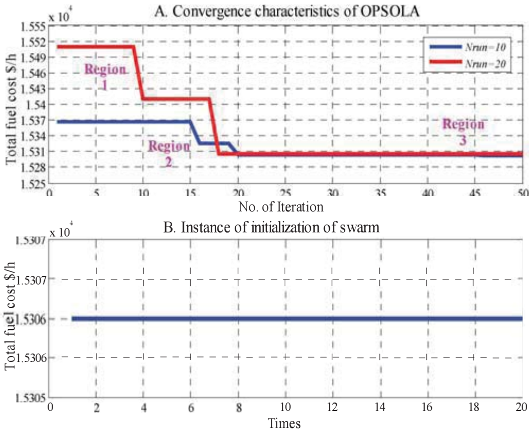


Fig. 5. Convergence characteristics of OPSOLA for 6-unit power system.

Region **#1:** The swarm searches in *d*-dimensional test space on direction to be used as guidance in order to acquire experience and this occur after several iterations.

Region **#2:** The swarm moves from previous position to a new position with high velocity in one direction (swarm moves either forward or backward based on orthogonality).

Region **#3:** The swarm particles receive a better guidance to move directly towards the target with shortest time.

Table V lists the solution vector P_j (j = 1, 2, ..., 6) corresponding to the best solution of OPSOLA with $N_{particle}$ = 20 and N_{iter} = 50. This solution satisfies the equality constraint in (3). The load demand P_D is 1263 MW. The transmission loss P_L is obtained by subtracting P_D from the total output. Thus, P_L = 8.56 MW. The total output power of 1271.56 MW satisfies the constraint for power equilibrium and inequality constraints in (5)-(7) and (12).

Table VI shows the performance comparison between OPSOLA and the best algorithm from the Table III, i.e., the RDPSO. The OPSOLA with $N_{particle}$ = 20, N_{run} = 100 and N_{iter} = 1000 was applied to the 6-unit power system. Simulation results demonstrated that OPSOLA was more effective in finding better solutions than that obtained from RDPSO. It can be seen that OPSOLA achieved a saving of 94.72 $/h, i.e., 0.61% improvement in mean cost over RDPSO. With respect to the network loss, OPSOLA achieved an improvement of 30.69 percent over RDPSO algorithm. The standard deviation of the mean cost can be considered as a measure of stability of the algorithm. Over 100 runs, in OPSOLA the σ of the mean cost remained zero, where as in RDPSO, σ was found to be 13.50. These results show that OPSOLA provides a stable and better solution.

TABLE V. POWER DELIVERED BY EACH GENERATOR

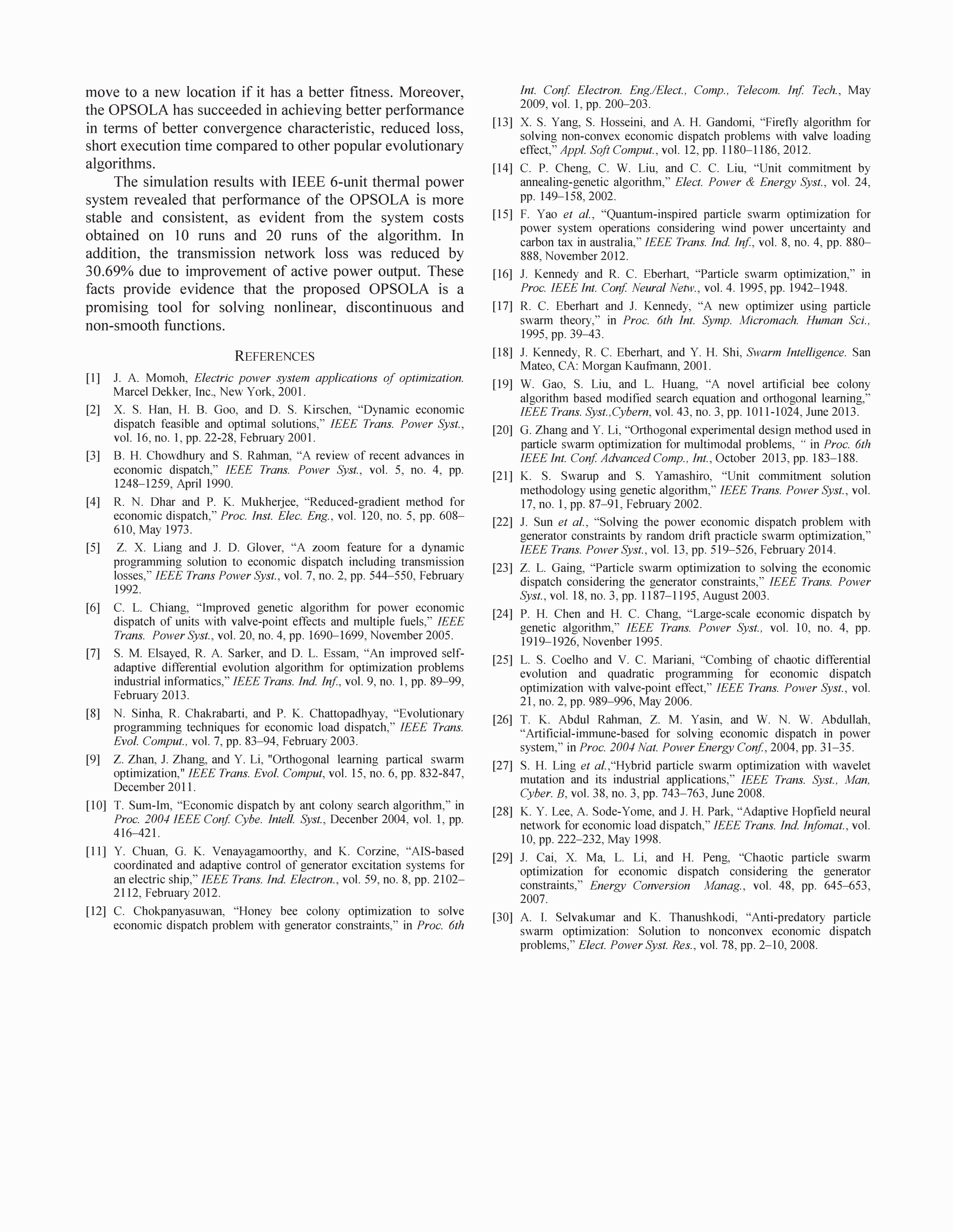
Algorithm	Optimum output power (MW)						Total Output Power (MW)
	P_1	P_2	P_3	P_4	P_5	P_6	
OPSOLA	461.19	186.36	244.93	129.98	158.44	90.66	1271.56

TABLE VI. PERFORMANCE COMPARISION BETWEEN TWO ALGORITHMS

Algorithm	Mean Cost ($/h)	P_L (MW)	σ
OPSOLA	15359.00	8.56	0
RDPSO [22]	15453.72	12.35	13.50

V. CONCLUSION

In this paper, we proposed a novel orthogonal PSO learning algorithm to solve ED problem. The orthogonality of the particle's movement is utilized to improve PSO performance. It uses as a guide to move the particle vectors from one level to another towards the solution point. We propose an updating strategy in which the particle will only



move to a new location if it has a better fitness. Moreover, the OPSOLA has succeeded in achieving better performance in terms of better convergence characteristic, reduced loss, short execution time compared to other popular evolutionary algorithms.

The simulation results with IEEE 6-unit thermal power system revealed that performance of the OPSOLA is more stable and consistent, as evident from the system costs obtained on 10 runs and 20 runs of the algorithm. In addition, the transmission network loss was reduced by 30.69% due to improvement of active power output. These facts provide evidence that the proposed OPSOLA is a promising tool for solving nonlinear, discontinuous and non-smooth functions.

REFERENCES

[1] J. A. Momoh, *Electric power system applications of optimization.* Marcel Dekker, Inc., New York, 2001.

[2] X. S. Han, H. B. Goo, and D. S. Kirschen, “Dynamic economic dispatch feasible and optimal solutions,” *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 22-28, February 2001.

[3] B. H. Chowdhury and S. Rahman, “A review of recent advances in economic dispatch,” *IEEE Trans. Power Syst.*, vol. 5, no. 4, pp. 1248–1259, April 1990.

[4] R. N. Dhar and P. K. Mukherjee, “Reduced-gradient method for economic dispatch,” *Proc. Inst. Elec. Eng.*, vol. 120, no. 5, pp. 608–610, May 1973.

[5] Z. X. Liang and J. D. Glover, “A zoom feature for a dynamic programming solution to economic dispatch including transmission losses,” *IEEE Trans Power Syst.*, vol. 7, no. 2, pp. 544–550, February 1992.

[6] C. L. Chiang, “Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1690–1699, November 2005.

[7] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “An improved self-adaptive differential evolution algorithm for optimization problems industrial informatics,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 89–99, February 2013.

[8] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, “Evolutionary programming techniques for economic load dispatch,” *IEEE Trans. Evol. Comput.*, vol. 7, pp. 83–94, February 2003.

[9] Z. Zhan, J. Zhang, and Y. Li, "Orthogonal learning partical swarm optimization," *IEEE Trans. Evol. Comput*, vol. 15, no. 6, pp. 832-847, December 2011.

[10] T. Sum-Im, “Economic dispatch by ant colony search algorithm,” in *Proc. 2004 IEEE Conf. Cybe. Intell. Syst.*, Decenber 2004, vol. 1, pp. 416–421.

[11] Y. Chuan, G. K. Venayagamoorthy, and K. Corzine, “AIS-based coordinated and adaptive control of generator excitation systems for an electric ship,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 2102–2112, February 2012.

[12] C. Chokpanyasuwan, “Honey bee colony optimization to solve economic dispatch problem with generator constraints,” in *Proc. 6th Int. Conf. Electron. Eng./Elect., Comp., Telecom. Inf. Tech.*, May 2009, vol. 1, pp. 200–203.

[13] X. S. Yang, S. Hosseini, and A. H. Gandomi, “Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect,” *Appl. Soft Comput.*, vol. 12, pp. 1180–1186, 2012.

[14] C. P. Cheng, C. W. Liu, and C. C. Liu, “Unit commitment by annealing-genetic algorithm,” *Elect. Power & Energy Syst.*, vol. 24, pp. 149–158, 2002.

[15] F. Yao *et al.*, “Quantum-inspired particle swarm optimization for power system operations considering wind power uncertainty and carbon tax in australia,” *IEEE Trans. Ind. Inf.*, vol. 8, no. 4, pp. 880–888, November 2012.

[16] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. 1995, pp. 1942–1948.

[17] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proc. 6th Int. Symp. Micromach. Human Sci.*, 1995, pp. 39–43.

[18] J. Kennedy, R. C. Eberhart, and Y. H. Shi, *Swarm Intelligence.* San Mateo, CA: Morgan Kaufmann, 2001.

[19] W. Gao, S. Liu, and L. Huang, “A novel artificial bee colony algorithm based modified search equation and orthogonal learning,” *IEEE Trans. Syst.,Cybern*, vol. 43, no. 3, pp. 1011-1024, June 2013.

[20] G. Zhang and Y. Li, “Orthogonal experimental design method used in particle swarm optimization for multimodal problems, “ in *Proc. 6th IEEE Int. Conf. Advanced Comp., Int.*, October 2013, pp. 183–188.

[21] K. S. Swarup and S. Yamashiro, “Unit commitment solution methodology using genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 87–91, February 2002.

[22] J. Sun *et al.*, “Solving the power economic dispatch problem with generator constraints by random drift practicle swarm optimization,” *IEEE Trans. Power Syst.*, vol. 13, pp. 519–526, February 2014.

[23] Z. L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, August 2003.

[24] P. H. Chen and H. C. Chang, “Large-scale economic dispatch by genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1919–1926, Novenber 1995.

[25] L. S. Coelho and V. C. Mariani, “Combing of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect,” *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 989–996, May 2006.

[26] T. K. Abdul Rahman, Z. M. Yasin, and W. N. W. Abdullah, “Artificial-immune-based for solving economic dispatch in power system,” in *Proc. 2004 Nat. Power Energy Conf.*, 2004, pp. 31–35.

[27] S. H. Ling *et al.*,“Hybrid particle swarm optimization with wavelet mutation and its industrial applications,” *IEEE Trans. Syst., Man, Cyber. B*, vol. 38, no. 3, pp. 743–763, June 2008.

[28] K. Y. Lee, A. Sode-Yome, and J. H. Park, “Adaptive Hopfield neural network for economic load dispatch,” *IEEE Trans. Ind. Infomat.*, vol. 10, pp. 222–232, May 1998.

[29] J. Cai, X. Ma, L. Li, and H. Peng, “Chaotic particle swarm optimization for economic dispatch considering the generator constraints,” *Energy Conversion Manag.*, vol. 48, pp. 645–653, 2007.

[30] A. I. Selvakumar and K. Thanushkodi, “Anti-predatory particle swarm optimization: Solution to nonconvex economic dispatch problems,” *Elect. Power Syst. Res.*, vol. 78, pp. 2–10, 2008.

Paper B: IEEE-SMC 2015

Orthogonal PSO algorithm for economic dispatch of power under power grid constraints

Published in:	Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on System, Man, and Cybernetics
Date of Conference:	9-12 Oct. 2015
Date Added to IEEE *Xplore*:	14 January 2016
Electronic ISBN:	978-1-4799-8697-2
USB ISBN:	978-1-4799-8696-5
INSPEC Accession Number:	15718931
DOI:	**10.1109/SMC.2015.16**
Publisher:	IEEE
Conference Location:	Kowloon, China

2015 IEEE International Conference on Systems, Man, and Cybernetics

Orthogonal PSO Algorithm for Economic Dispatch Of Power Under Power Grid Constraints

Loau Tawfak Al Bahrani
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia
E-mail: lalbahrani@swin.edu.au

Jagdish C. Patra
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia
E-mail: JPatra@swin.edu.au

***Abstract*—We propose an enhanced Particle Swarm Optimization (PSO) algorithm named Orthogonal PSO (OPSO) algorithm, for Economic Dispatch (ED) of the generated power in a smart grid environment. The equality and inequality constraints, and power balance response against mismatch between load demand and total power outputs of generating units involve nonlinear characteristics and non-smooth cost functions. The proposed OPSO algorithm has the ability to solve such complex problems of power systems including ED. The OPSO algorithm applies Orthogonal Vectors (OVs) in the *d*-dimensional search space. The *d* particles that have possible solutions move in the *d*-dimensional search space to form OVs. These OVs are generated and updated in each iteration and they used to guide those particles to fly in one direction toward global minimum. The OPSO algorithm is evaluated and tested through 15 generating units and its performance is compared with several other optimization methods. We found that OPSO algorithm provides better results in solving the total cost, and power constraints. Furthermore, the OPSO algorithm is succeeded to improve the PSO algorithm in terms of high solution quality, robustness and convergence.**

Keywords-economic dispatch; linearity; orthogonality; particle swarm optimization; power system constraints.

I. Introduction

Several optimization techniques have been used to improve the effectiveness and efficiency of electric power systems. Optimization techniques help energy companies that deal with these issues by enabling them to make better and faster decisions. Economic Dispatch (ED) is one of the most important problems for energy companies. For example, dividing the total load demand among available generators with as minimum cost as possible gives important tools about the behaviour of power system during estimated short-term.

Traditionally, the ED problem can be solved by various mathematical methods, including the lambda-iteration method [1], the gradient method [2], and the dynamical programming method [3]. However, these numerical methods do not work effectively for non-smooth and non-convex functions because of high "*dimensionality*". In order to successfully solve the nonlinear functions, a wide variety of population-based random search have been used to solve the ED problems. These methods include Genetic Algorithm (GA) [4], Particle Swarm Optimization (PSO) algorithm [5], Differential Evolution (DE) algorithm [6], [7], Evolutionary Programming (EP) algorithm [8], Neural Networks (NN) [9], Ant Colony Search (ACS) algorithm [10], Artificial Immune Systems (AIS) [11], PSO based on the Orthogonal Experimental Design (OED) [12], [13], [14], Honey Bee Colony (HBC) algorithm [15], Firefly Algorithm (FA) [16], to name only a few.

One of those evolutionary algorithms is PSO algorithm. It is a global optimization method developed by Kennedy and Eberhart [17], [18]. It is a swarm intelligence algorithm that simulates the behavior of some animal species, such as birds flocking and fish schooling [19]. The population strategy depends on the iterative learning algorithm. The learning strategy of PSO depends on: Firstly, each particle flying in the search space and adapting its flying trajectory according to its personal best experience and its neighborhood's best experience. Secondly, when looking for a global optimum in the test area, particles fly in the search space according to guiding rules that make each particle learns from its own best historical experience and its neighborhood's best historical experience. A particle while choosing the neighborhood's best historical experience, either uses the best historical experience of the whole swarm as its neighborhood's best historical experience or uses the best historical experience of the particle in its neighborhood which is determined by some topological structure, e.g., the ring structure [12]. In both cases, the learning strategy of a particle's best experience and its neighborhood's best experience is obtained through a linear summation. However, this linearity will negatively reflect on the performance of the PSO algorithm due to oscillation phenomenon which causes inefficiency to the search ability of the algorithm and delays convergence.

Some evolutionary algorithms tried to overcome the drawback of the PSO algorithm through overtaking the oscillation caused by the combination of the personal effect and the neighbourhood effect in the velocity equation. One of these methods is PSO-OED [13]. OED is orthogonal learning design, and sometimes called orthogonal design. It is usually used to study the effect of several factors simultaneously and the best combination of factor levels can be found in several tests. However, the main effect of OED holds only when no or weak interaction of factors exists. It means *m* particles have "Linear Combination Property" (LCP) in order to obtain solution. Sometimes, some particles loss this property, therefore the

978-1-4799-8697-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SMC.2015.16

ability to find the solution becomes difficult. This limitation of OED will make PSO search effective on "unimodal". On the other hand, it is very vulnerable on "complex multimodal" problems [14].

In this paper, we propose a novel Orthogonal PSO (OPSO) algorithm. The proposed OPSO algorithm differes PSO-OED in several aspects. The OPSO algorithm depends on forming Orthogonal Vectors (OVs) that are basically found in the *d*-dimensional searching area. Subsequently, LCP is constantly achieved. The OPSO algorithm is a stochastic optimization technique. The OVs are utilized to improve the PSO performance through improving the particles (that have the possible solutions) positions and their velocities. This process makes those particles have ability to move in clear direction from one level to another based on the following iteration. In other word, OPSO algorithm has ability to control on the particles chaotic sequences through using orthogonality. Therefore, the process of exploration and exploitation by best particles to the search on global minimum is clearly advanced.

We applied OPSO algorithm to solve nonlinear characteristics of the generator and describe the power constraints of the ED problems. With extensive experimental results, we have shown the better performance of OPSO algorithm compared to several competing algorithms in solving ED problems.

II. Problem Formulation

Here, we explain the cost function and the constraints involved in this study.

A. Cost

For efficient management of the power system, it is required to achieve optimal combination of output power generated that minimize the total generation cost while satisfying the load demand, and practical operation constraints of generators. The objective function of the cost is given by

$$\text{Minimize } F_{cost} = \sum_{j=1}^{N_{gen}} F(P_j) \qquad (1)$$

where $F(P_j)$ is the cost function of the *j*th generating unit in (\$/h), P_j is the active power output of the *j*th generating unit in (MW), and N_{gen} is the number of available generators. The cost function of each generating unit is related to the real power delivered into the system, and is typically modeled by a smooth quadratic function [5].

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \qquad (2)$$

where a_j, b_j, and c_j are the cost coefficients of the *j*th generating unit.

B. Constraints of the Power System Under Test

In this work, we consider the following constraints:

Constraint # 1: *Power Balance Constraint*: The constraint of power balance can be stated as the total system generation equals to the total load demand (P_D) in MW plus the transmission network loss (P_L) in MW. This constraint is expressed by

$$\sum_{j=1}^{N_{gen}} P_j = P_D + P_L \qquad (3)$$

P_L represents to the losses in terms of the active output power of the generators that can be represented using *B*-coefficients. P_L is approximated by Kron's loss formula [5], given by

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k + \sum_{j=1}^{N_{gen}} P_j B_{j0} + B_{00} \qquad (4)$$

where *j*, *k* = 1, 2, ..., N_{gen} indicate to the indices of the generating units, and B_{jk}, B_{j0}, B_{00} are known as the loss coefficients or *B*-coefficients [20].

Constraint # 2: *Generation Limits*: The generation limit is given by

$$P_{j,min} < P_j < P_{j,max} \ , \quad (j = 1, 2, ..., N_{gen}) \qquad (5)$$

which requires that the power generation of each unit remains within its minimum $P_{j,min}$ and its maximum $P_{j,max}$ production limits, which are directly related to the design of the generator.

Constraint # 3: *Ramp Rate Limits*: In the real operating process of the generating units, the operating range of all units is restricted by their ramp rate limits. According to [20], the inequality constraints due to the ramp limits are:

i) If power generation increases

$$P_j - P_j^0 \le UR_j \qquad (6)$$

ii) If power generation decreases

$$P_j^0 - P_j \le DR_j \qquad (7)$$

where P_j^0 in (MW) is the previous active output power, UR_j in (MW/h) is the up-ramp limit of the *j*th generator; and DR_j in (MW/h) is the down-ramp limit of the *j*th generator.

By substituting (6) and (7) in (5), we obtain

$$max\left\{P_{j,min}, \left(P_j^0 - DR_j\right)\right\} \le P_j \le min\left\{P_{j,max}, \left(P_j^0 + UR_j\right)\right\} \qquad (8)$$

Let us consider,

$$P_{j,low} = max\left\{P_{j,min}, \left(P_j^0 - DR_j\right)\right\}, \text{ and} \qquad (9)$$

$$P_{j,high} = min\left\{P_{j,max}, \left(P_j^0 + UR_j\right)\right\} \qquad (10)$$

where $P_{j,low}$ and $P_{j,high}$ are the new lower and upper limits of unit *j*, respectively. Ramp rate is a function of resource size. The generators must be able to increase/decrease the power output at a specific rate per minute. For example, depending on the generating capacity of generators the *UR* and *DR* range from 25-50 MW, the performance ramp rate will be 2-3 MW/min [21].

Constraint # 4: *Prohibited Operating Zones:* Due to the steam valve operation or vibration in a shaft bearing, the system contains some prohibited operating zones. Due to this, constraints and discontinuities are produced in cost curves corresponding to the prohibited operating zones. Therefore, P_j (j = 1, 2, ..., N_{gen}) of power system must avoid the prohibited operating zones. Consequently, considering the constraint #2 in (5), the feasible operating zones of the jth generating unit are given by [20].

$$\begin{aligned} & P_{j,min} \le P_j \le P^{l}_{j,1} \\ & P^{u}_{j,k-1} \le P_j \le P^{l}_{j,k} \ , \quad k = 2, 3, ..., N_{pz,j} \qquad (11) \\ & P^{u}_{j,N_{pz,j}} \le P_j \le P_{j,max} \end{aligned}$$

where $P^{l}_{j,1}$ and $P^{u}_{j,k}$ are the lower and upper bounds of the kth prohibited zone of the jth generating unit, and $N_{pz,j}$ is the number of prohibited zones of the jth generating unit.

Incorporating these constraints in (8) and (9) and (10), we obtain the final set of constraints as given below

$$\begin{aligned} & P_{j,low} \le P_j \le P^{l}_{j,1}, \\ & P^{u}_{j,k-1} \le P_j \le P^{l}_{j,k} \ , \quad k = 2, 3, ..., N_{pz,j}, \qquad (12) \\ & P^{u}_{j,N_{pz,j}} \le P_j \le P_{j,high} \end{aligned}$$

III. Orthogonal PSO Learning Algorithm

Here, we briefly describe the PSO algorithm and explain our proposed OPSO algorithm:

A. The PSO Algorithm

Let us consider a swarm with m particles searching for solution in a d-dimensional search space. Thus, the size of swarm, $N_{particle} = m$. Each particle of the swarm has one d-dimensional position vector and one d-dimensional velocity vector. A position vector represents a possible solution. The position vector X_i and the velocity vector V_i of the ith (i = 1, 2, …, m) particle are given by

$$X_i = [x_{i,1},\ x_{i,2},\ \ldots,\ x_{i,d}\ ] \qquad (13)$$

$$V_i = [v_{i,1},\ v_{i,2},\ \ldots,\ v_{i,d}\ ] \qquad (14)$$

Each particle i will keep its personal historical best position vector, $H_{i,pers}$, as it progresses through iterations.

$$H_{i,pers} = [h_{pi,1},\ h_{pi,2},\ \ldots,\ h_{pi,d}] \qquad (15)$$

One of the important frameworks for solving PSO algorithm is the ring structure [12]. The behavior of the neighborhood of a particle i includes i-1, i, and i+1 in a ring structure. The best position among all the particles in the ith particle's neighborhood is given by

$$H_{i,neigh} = [h_{ni,1},\ h_{ni,2},\ \ldots,\ h_{ni,d}] \qquad (16)$$

The vectors V_i and X_i are initialized randomly. These are updated iteration by iteration through the guidance of $H_{i,pers}$ and $H_{i,neigh}$ according to (17) and (18).

$$V_i = V_i + c_1 r_{1i} (H_{i,pers} - X_i) + c_2 r_{2i} (H_{i,neigh} - X_i) \quad (17)$$

$$X_i = X_i + V_i \qquad (18)$$

where c_1 and c_2 are coefficients of particle acceleration, usually positive values, and are chosen experimentally from the range [1, 2]. The r_{1i} and r_{2i} are two randomly generated values within range [0, 1]. In each iteration, considering all m particles, the best personal position vector and the best neighbor position vector are recorded. Finally, at the end of iterations, the optimal solution is given by the best neighbor position vector.

B. The Proposed OPSO Algorithm

In PSO algorithm, each particle updates its velocity and position vectors according to its personal best position and its neighborhood's best position. However, this learning strategy can cause "Oscillation" phenomenon [12]. This oscillation becomes more prominent with high d-dimensional search space. The oscillation occurs due to linear combination of the personal effect and the neighborhood effect. This makes the particle in a confused state to decide where it goes. So, the search ability of the PSO algorithm becomes inefficient and delays the convergence speed. In order to overcome this limitation, OPSO algorithm is proposed to improve PSO algorithm.

The OPSO algorithm provides a new topology inside the swarm called "Orthogonal Particle Formation" (OPF). In a swarm population with ($m \ge d$), each particle i has a position vector X_i and a velocity vector V_i, as given in (13) and (14), respectively. The OPSO algorithm updates the position and velocity vectors in each iteration using the following steps:

***Step* 1:** For each particle i, (i = 1, 2, …, m) evaluate the cost using the position vector in the cost function (1). Select the particle that has the minimum cost. Let us denote its position vector as X_{best}. Similarly, obtain the next m 1 best position vectors. Arrange these m best vectors in ascending order of costs.

***Step* 2:** Construct a matrix A of size $m \times d$ such that each row occupies one of the m best position vectors in the same ordered sequence. Then, convert the matrix A to matrix B, such that matrix B is a symmetric matrix with size $d \times d$ using the pseudocode given in Fig. 1.

```
Procedure: Convert to a symmetric matrix
    for i=1:d
            B(1,i) = A(1,i)
            B(i,1) = A(1,i)
    end for
    for k = 2:d
        for i = 2:d
                B(k,i) = A(k,i)
                C(k,i) = B(k,i)
                B(i,k) = C(k,i)
        end for
    end for
```

Fig. 1. Pseudocode for converting matrix [A] to a symmetric matrix [B].

***Step* 3:** Convert matrix B to an orthogonal matrix C of dimension $d\times d$, using Gram–Schmidt orthogonalization method.

***Step* 4:** Obtain a diagonal matrix D of dimension $d\times d$ using (19)

$$D = CBC^{T} \tag{19}$$

Let D_k denotes the kth row of the matrix D.

***Step* 5:** Update the position and velocity vectors of d number of particles out of m particles. Thus, the d best position and velocity vectors as obtained in *Step* 1 are updated as follows

$$V_k = V_k + c\, r_k\,(D_k - X_k) \tag{20}$$

$$X_k = X_k + V_k \tag{21}$$

where c is an acceleration factor and set to 2 and r_k is a random number within range [0, 1] and $k = 1, 2, \ldots, d$.

At the end of iterations, the position vector X_{best} as computed in *Step* 1 provides the optimal solution.

Instead of using the neighbor position vector and the personal position vector as done in PSO algorithm in linear combination (17), in OPSO algorithm we update the position and velocity vectors by constructing a new OV, D_k. This means that only one OV is used to control on the movement of each particle. Thus, the particles move in one direction based on the OV. A flowchart of the OPSO algorithm with major computational steps is shown in Fig. 2.

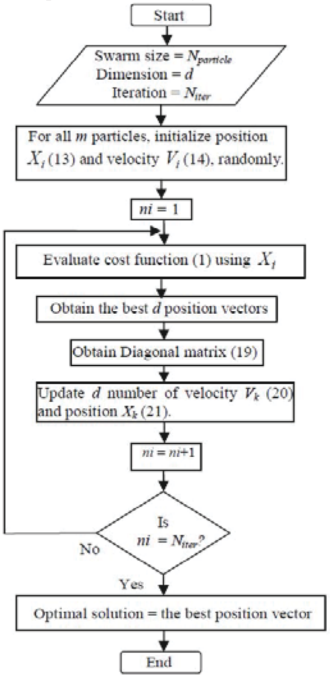

Start
Swarm size = $N_{particle}$
Dimension = d
Iteration = N_{iter}
For all m particles, initialize position X_i (13) and velocity V_i (14), randomly.
ni = 1
Evaluate cost function (1) using X_i
Obtain the best d position vectors
Obtain Diagonal matrix (19)
Update d number of velocity V_k (20) and position X_k (21).
$ni = ni$+1
Is $ni = N_{iter}$?
No
Yes
Optimal solution = the best position vector
End

Fig. 2. Flowchart of the proposed OPSO algorithm.

C. *A Simple Example*

To explain PSO and OPSO algorithms, a simple example is illustrated. Consider a 2-dimentional function given by $f(x, y) = x^2+y^2+3$. The objective is to find values x and y so the value of the function $f(x, y)$ is minimized. With a range of [-100 100], for x and y, the function is plotted in Fig. 3. Note that the function minimized to $f = 3$ when $x = 0$ and $y = 0$. Both the PSO and OPSO algorithms have 10 particles in the swarm and uses 10 iterations. Fig. 4 shows that the PSO algorithm moves forward and backward between 3rd iteration (-44.3, 31.6) and 4th iteration (16, -10.2) causing oscillations. In the OPSO algorithm, the particles move steadily from the initial position to the solution in the 10th iteration.

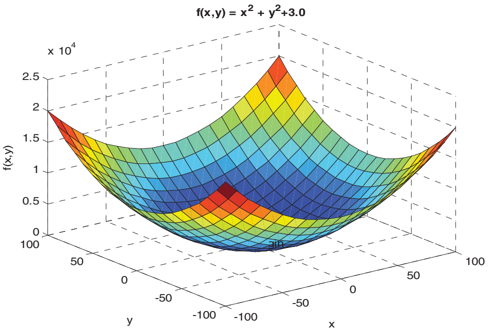


Fig. 3. Plot of function $f(x, y)$.

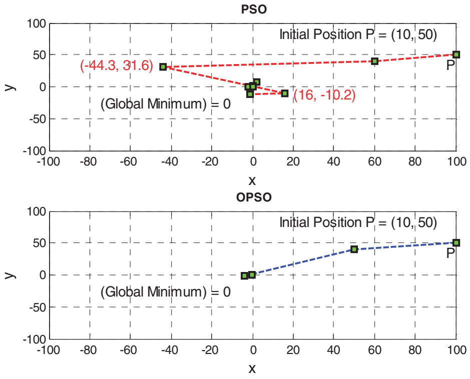


Fig. 4. Movement of one particle with iteration.

IV. Application of OPSO Algorithm to ED Problem

Here we describe the simulation results carried out on the 15-unit generator power system with several constraints.

A. *The Power System Specification*

In order to exhibit performance of the OPSO algorithm to solve optimal dispatch of generation, a practical IEEE 15-unit power system was used. The generation parameters are given in Table I [20]. The load demand is 2630 MW. The *B*-loss coefficients matrix is not listed because of space limitation, however, B_{jk}, B_{j0}, B_{00} are taken into account with calculations. The power system is a medium-scale one and the generating units 2, 5, 6, and 12 have a total of 11 prohibited zones. Thus, there are 12 inequality constraints as described by (12).

TABLE I. GENERATING UNIT RAMP AND PROHIBITED ZONES

Unit	P_j^0 MW	P_jmin MW	P_jmax MW	a_i $	b_i $/MW	c_i $/MW^2	UR_j MW/h	DR_j MW/h	Prohibited Zones
1	400	150	455	671	10.1	299	80	120	-
2	300	150	455	574	10.2	183	80	120	[185, 225] [305, 335] [450, 450]
3	105	20	130	374	8.8	1126	130	130	-
4	100	20	130	374	8.8	1126	100	130	-
5	90	150	470	461	10.4	205	80	120	[180, 200] [305, 335] [390, 420]
6	400	135	460	630	10.1	301	80	120	[230, 225] [365, 395] [430, 455]
7	350	135	465	548	9.8	364	80	120	-
8	95	60	300	227	11.2	338	65	100	-
9	105	25	162	173	11.2	807	60	100	-
10	110	25	160	175	10.7	1203	60	100	-
11	60	20	80	186	10.2	3586	80	80	-
12	40	20	80	230	9.9	5513	80	80	[30, 40] [55, 65]
13	30	25	85	225	13.1	371	80	80	-
14	20	15	55	309	12.1	1929	55	55	-
15	20	15	55	323	12.4	4447	55	55	-

B. *Experimental Results*

Six evolutionary computation techniques used for the 15-unit power system by other authors are listed in Table II, for example, ACSA [10], AIS [11], HBC [15], FA [16], Random Drift Particle Swarm Optimization (RDPSO) [20], and the Differential Evolution (DE) [22]. The results in Table II represent the minimum, maximum and mean cost. In addition, Standard Deviation σ of the minimum cost is also listed. The best results are for RDPSO algorithm with minimum, maximum and mean cost. While, AIS algorithm is the best in σ calculation.

TABLE II. RESULTS OF 15-UNIT SYSTEM OVER 100 RUNS

Algorithm	**Min. Cost** $/h	**Max. Cost** $/h	**Mean Cost** $/h	**σ**
ACSA [10]	32863.17	33256.28	33120.02	86.16
AIS [11]	32895.91	33132.01	33017.65	**58.12**
HBC [15]	32789.23	33301.49	33030.86	69.79
FA [16]	32898.01	33310.72	33116.90	96.38
RDPSO[20]	**32652.33**	**32959.79**	**32744.58**	82.47
DE [22]	32718.82	33213.31	32966.43	110.32

In addition to our proposed OPSO algorithm, we tested 15-unit power system by PSO algorithm. The test was carried out on the same parameters, $c_1 = c_2 = 2$, $N_{particle} = 20$, $d = 15$, $N_{run} = 100$, each independent run has $N_{iter} = 1000$. Table III lists the statistical results of the total costs and σ were obtained by PSO and OPSO algorithm for the ED problem. The result shows that OPSO algorithm is the best in mean cost and has lowest σ. This means that PSO algorithm was clearly improved through orthogonality. Furthermore, the OPSO algorithm is the best in mean cost and has lowest σ among other evolutionary techniques as given in Table II. The OPSO algorithm has the best performance and robustness (i.e., the stability in performance in terms of mean cost and σ) than its competitors in 15-unit power system.

TABLE III. RESULTS OF PSO AND OPSO ALGORITHMS OF 15-UNIT OVERV 100 RUNS

Algorithm	**Min. Cost** $/h	**Max. Cost** $/h	**Mean Cost** $/h	**σ**
PSO	32675.00	32770.00	32705.00	22.36
OPSO	32669.00	32699.00	32688.00	7.21

Fig. 5 visualizes the best convergence characteristics of OPSO and PSO algorithms over 100 runs on the ED problem of 15-unit power system, showing that the OPSO algorithm has better convergence properties than the PSO algorithm.

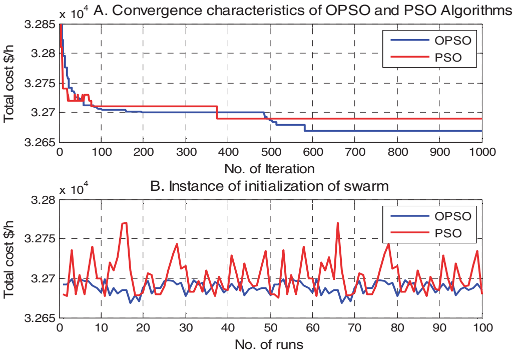


Fig. 5. Convergence characteristics of OPSO and PSO algorithms of 15-unit power system.

Table IV lists the solution vector P_j (j = 1, 2, ..., 15) corresponding to the best solution for PSO and OPSO algorithms applied to the 15-unit power system with $N_{run} = 100$, $N_{iter} = 1000$, and $N_{particle} = 20$. In addition, the inequality constraints (12) have been solved and all jth generating units have avoided the prohibited operating zones.

TABLE IV. OPTIMIZED POWER DELIVERED BY EACH GENERATOR USING PSO AND OPSOALGORITHMS

	Generator Power MW				
Gen. #	1	2	3	4	5
PSO	455.00	380.00	129.30	129.20	170.00
OPSO	454.48	380.00	129.30	129.20	150.00
Gen. #	6	7	8	9	10
PSO	428.00	430.00	60.00	162.00	156.87
OPSO	428.00	430.00	93.25	149.93	107.23
Gen. #	11	12	13	14	15
PSO	20.00	80.00	25.00	15.00	15.00
OPSO	70.87	75.50	25.00	15.00	15.00
Total output power: PSO = 2655.37 MW, OPSO = 2652.76 MW					

Table V shows the mean cost, transmission network loss P_L and σ for the best three methods tested on the IEEE 15-unit power system with $N_{run} = 100$, each run has $N_{iter} = 1000$, and $N_{particle} = 20$. To study the robustness of the OPSO algorithm compared with RDPSO and PSO alogarithms as shown in Table V, the OPSO algorithm has satisfied the lowest value of σ, and it obtained the best mean cost. It is stable in performance. In addition, by comparing the results of OPSO algorithm with RDPSO and PSO algorithms, the results show that the mean cost was reduced by 0.17% (saving $56/h) and by 0.05% (saving $17/h), respectively. Moreover, P_L was decreased by 11.42% (2.60 MW) and 11.59% (2.64 MW), respectively. In addition, the equality constraint in (3) has been satisfied. Total system generation equals to P_D plus P_L as given by:

OPSO algorithm: 2652.76 = 2630 + 22.76
PSO algorithm: 2655.37 Ŏ2630 + 25.40

TABLE V. COMPARISION AMONG ALGORITHMS RESUTS

Algorithm	**Mean Cost** $/h	$\boldsymbol{P_L}$ MW	**σ**
RDPSO [20]	32744.58	25.36	82.47
PSO	32705.00	25.40	22.36
OPSO	32688.00	22.76	7.21

V. CONCLUSION

In this paper, we proposed a novel orthogonal PSO algorithm to solve ED problems. The orthogonality was utilized by the particles that have the possible solutions in the swarm using *d* orthogonal vectors (OVs). These OVs help *d* particles to improve their directions at each iteration. In addition, instead of creating and updating the guidance $H_{i,pers}$ and $H_{i,neigh}$ in PSO algorithm, *d* particles in the swarm updates its position and its velocity according to OVs. This means that only one guidance D_k has been used in the updated velocity (20). In this way, these particles fly steadily and clearly to find the global minimum. Subsequently, the OPSO algorithm has succeeded in eliminating the generated oscillation by the particle's movement forward and backward in PSO algorithm.

The OPSO algorithm has been tested on IEEE 15-unit power system. The results showed that OPSO algorithm has a better performance and clear superiority in mean cost and standard deviation calculations compared with other competition algorithms. The OPSO algorithm has shown more stability and robustness. Consequently, the OPSO algorithm has completely succeeded to improve the performance of the PSO algorithm.

We have shown that OPSO algorithm has ability to solve the equality and inequality constraints in the power system and avoids all prohibited operating zones. Furthermore, the OPSO algorithm was clearly reduced transmission network loss.

REFERENCES

[1] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," IEEE Trans Power Syst., vol. 5, no. 4, pp. 1248–1259, April 1990.

[2] R. N. Dhar and P. K. Mukherjee, "Reduced-gradient method for economic dispatch," Proc. Inst. Elec. Eng., vol. 120, no. 5, pp. 608–610, May 1973.

[3] R. R. Shoultsa, R. K. Chakravartya, and R. Lowtherb, "Quasi-static economic dispatch using dynamic programming with an improved zoom feature," Elec. Power Syst. Res., vol. 39, no. 3, pp. 215–222, 1996.

[4] A. Bhattacharya and P. K. Chattopadhyay, "Biogeography-based optimization for different economic load dispatch problems," IEEE Trans. Power Syst., vol. 25, no. 2, pp. 1064-1077, May 2010.

[5] Z.-L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," IEEE Trans. Power Syst., vol. 18, no. 3, pp. 1187–1195, August 2003.

[6] M. Vanitha and K. Thanushkodi, "Solution to economic dispatch problem by differential evolution algorithm considering linear equality and inequality constrains," Int. J. Res. Rev. Elec. Comp. Eng., vol. 1, no. 1, pp. 21–27, March 2011.

[7] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems industrial informatics," IEEE Trans. Ind. Inf., vol. 9, no. 1, pp. 89–99, February 2013.

[8] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," IEEE Trans. Evol. Comput., vol. 7, pp. 83–94, February 2003.

[9] C. Yang, G. Deconinck, and W. Gui, "An optimal power-dispatching control system for the electrochemical process of zinc based on backpropagation and Hopfield neural networks," IEEE Trans. Ind. Electron., vol. 50, no. 5, pp. 953–961, May 2003.

[10] T. Sum-Im, "Economic dispatch by ant colony search algorithm," IEEE Conf. Cyber. Intell. Syst., vol. 1, pp. 416–421. December 2004.

[11] Y. Chuan, G. K. Venayagamoorthy, and K. Corzine, "AIS-based coordinated and adaptive control of generator excitation systems for an electric ship," IEEE Trans. Ind. Electron., vol. 59, no. 8, pp. 3102–2112, February 2012.

[12] Z. Zhan, J. Zhang and Y. Li, "Orthogonal learning partical swarm optimization," IEEE Trans. Evol. Comput., vol. 15, no. 6, pp. 832-847, December 2011.

[13] W. Gao, S. Liu and L. Huang, "A novel artificial bee colony algorithm based modified search equation and orthogonal learning," IEEE Trans. Cybern, Syst., vol. 43, no. 3, pp. 1011-1024, June 2013.

[14] G. Zhang and Y. Li, "Orthogonal experimental design method used in particle swarm optimization for multimodal problems, " in Proc. 6th IEEE Int. Conf. Advanced Comp. Int., October 2013, pp. 183–188.

[15] C. Chokpanyasuwan, "Honey bee colony optimization to solve economic dispatch problem with generator constraints," in Proc. IEEE 6th Int. Conf. Electron. Eng./Elect., Comp., Telecom. Inf. Tech., May 2009, vol. 1, pp. 200–203.

[16] X.-S. Yang, S. Hosseini, and A. H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," Appl. Soft Comput., vol. 12, pp. 1180–1186, 2012.

[17] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., vol. 4. 1995, pp. 1942–1948.

[18] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Proc. 6th Int. Symp. Micromach. Human Sci., 1995, pp. 39–43.

[19] J. Kennedy, R. C. Eberhart, and Y. H. Shi, Swarm Intelligence. San Mateo, CA: Morgan Kaufmann, 2001.

[20] J. Sun *et al.*, "Solving the power economic dispatch problem with generator constraints by random drift practicle swarm optimization," IEEE Trans. Power Syst., vol. 13, pp. 519–526, February 2014.

[21] Hawaiian Electric Company. (2013, February, 22). Invitation for low cost renewable energy projects on Oahu through request for waiver from competitive bidding [Online]. Available: http : // www .hawaiianelectric .com.

[22] L. S. Coelho and V. C. Mariani, "Combing of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," IEEE Trans. Power Syst., vol. 21, no. 2, pp. 989–996, May 2006.

Paper C: IEEE-IJCNN 2016

Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints

Published in: International Joint Conference on Neural Network (IJCNN) 2015

Date of Conference: 24-29 July 2016

Date Added to IEEE *Xplore*: 03 November 2016

Electronic ISBN: 978-1-5090-0620-5

USB ISBN: 978-1-5090-0619-9

Print on Demand(PoD) ISBN: 978-1-5090-0621-2

Electronic ISSN: 2161-4407

INSPEC Accession Number: 16446890

DOI: 10.1109/IJCNN.2016.7727263

Publisher: IEEE

Conference Location: Vancouver, BC, Canada

Orthogonal PSO Algorithm for Optimal Dispatch of Power of Large-Scale Thermal Generating Units in Smart Power Grid Under Power Grid Constraints

Loau Tawfak Al Bahrani[1], Jagdish C. Patra[1], and Ryszard Kowalczyk[1,2]
[1]Faculty of Science, Engineering and Technology
Swinburne University of Technology, Melbourne, Australia
[2]Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
E-mail: lalbahrani@swin.edu.au; JPatra@swin.edu.au; rkowalczyk@swin.edu.au

***Abstract*—We propose a novel approach called, an orthogonal particle swarm optimization (OPSO) algorithm, for economic dispatch (ED) of thermal generating units (TGUs) in smart electric power gird (SEPG) environment. The characteristics of TGUs are nonlinear and the generation system becomes more and more complicated when these TGUs are subjected to ramp rate constraints and prohibited operating zones. In such case, the cost functions become non-smooth and non-convex due to the discontinuities in the cost curves. Moreover, for large-scale TGUs, the high dimensions used in ED problem become a big challenge to find global minimum and to avoid falling into local minima. The proposed OPSO algorithm has the ability to solve such complex problems including ED. The OPSO algorithm applies an orthogonal diagonalization process. It makes *d* particles (out of total *m* particles, *m* ≥ *d*) that have the possible solutions by constructing orthogonal vectors in the *d*-dimensional search space. These orthogonal vectors are generated and updated in each iteration and are utilized to guide the *d* particles to fly in one direction toward global minimum. The OPSO algorithm is evaluated and tested through 40 TGUs and its performance is compared with several other optimization methods. We found that the OPSO algorithm provides better results in term of cost under power grid constraints. Furthermore, we have shown that the OPSO algorithm significantly improves the PSO algorithm in terms of high solution quality, robustness and convergence.**

Keywords—Orthogonal particle swarm optimization, orthogonal diagonalization process, economic dispatch, ramp rate limits, prohibited operating zones, thermal generating units.

I. Introduction

The cost of power generation, particularly in fossil fuel plants, is very high, and economic dispatch (ED) of power helps in saving a significant amount of revenue [1]. Therefore, ED is one of the important functions in smart electric power grid (SEPG) operation and control [2], aims to minimize total system generation costs and schedule committed generators to meet the load demand while satisfying power grid constraints. Traditionally, large-scale ED problem can be solved by many traditional optimization methods (TOMs), including lambda iteration method [3]. The gradient method [4], and dynamical programming method [5]. However, these numerical methods do not work effectively because the cost functions have discontinuities and higher order nonlinearities which are due to prohibited operating zones and ramp rate limits of generators [6]. In addition, high dimensionality associated with the ED problem. Because of these issues, TOMs fail to model these discontinuities and usually result in inaccurate dispatches causing loss of revenue. The ED of power with nonlinearities translates into a complicated optimization problem with complex and nonlinear characteristics, and multiple minima. In order to solve these problems, a wide variety of evolutionary computation techniques (ECTs) based on random search have been proposed for large-scale ED problems in smart electric power grid (SEPG). Some of the ECTs include genetic algorithm (GA) [7], differential evolution (DE) algorithm [8], particle swarm optimization (PSO) algorithm [9][10], ant colony search (ACSA) algorithm [11], artificial immune systems (AIS) [12], honey bee colony (HBC) algorithm [13], firefly algorithm (FA) [14]. These methods impose a few or no restrictions on the shape of cost functions, however, they are often prone to get trapped into local optima when applied to large-scale ED with multiple prohibited operating zones.

To enhance the global search ability of solving large-scale ED problem, several ECTs have been developed in the recent years. Among them, extensive researches focus on the improved PSO algorithms because of its popularity. These improved ECTs have been proved to be more effective to solve large-scale ED problem. For example, chaotic PSO (CPSO) method [15] combines PSO with adaptive inertia weight factor (AIWF) and chaotic local search (CLS) based on logistic and tent equations to solve ED problems with generator constraints. The anti-predatory PSO (APSO) [16] applies anti-predatory nature, which helps the swarm to escape from predators. The anti-predatory is modeled and embedded in the original PSO algorithm to form APSO

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

model. The hybrid PSO with wavelet-theory-based mutation (HPSOWM) operation [17] is used to enhance PSO algorithm in exploration and searching for a better solution. Random drift particle swarm optimization (RDPSO) [6] is inspired by free electron model in metal conductors placed in an external electric field. RDPSO method uses set of evolution equations to enhance the PSO global search ability. However, [6] and [15]-[17] have not used prohibited operating zones corresponding to each TGU in their computations so that there are less inequality constraints for the ED problem.

In order to make a fair comparison, some other ECTs are used in this paper. They include lambda logic (ʎ-logic) [18] and a mixed-integer quadratically constrained quadratic programming (MIQCQP) [19]. These two methods are used to solve large-scale ED problem. They take into account the prohibited operating zones and ramp rate limits as well as transmission network loss.

Recently, orthogonal particle swarm optimization (OPSO) algorithm was successfully used in solving ED problem under power grid constraints in SEPG of small and medium power systems [20][21].

In this paper, we propose a novel algorithm to improve the performance of PSO algorithm, named orthogonal PSO (OPSO) to obtain superior global search ability in solving ED problem. The OPSO algorithm is also applied to solve equality and inequality constraints, considering the ramp rate limits and prohibited operating zones and transmission network loss. The OPSO algorithm consists of *m* particles in a swarm in *d*-dimensional search space ($m \geq d$). The OPSO algorithm goes thought an orthogonal diagonalization process in which we obtain a *d* number of orthogonal guidance vectors from *m* number of current position vectors. These guidance vectors are used to update the velocity and position vectors of only *d* selected particles, whereas, the remaining ($m - d$) particles are not updated. This leads the search process primarily to concentrate on using best *d* particles in a swarm. We have shown that the OPSO algorithm is able to solve large-scale ED problems under power grid constraints quite effectively.

Rest of the paper is organized as follows. We explain the problems formulation in Section II. Explanation on our suggested OPSO algorithm is provided in Section III. In Section IV, we present the application of OPSO algorithm to ED problem. Finally, conclusion of this study is given in Section V.

II. Problem Formulation

Here, we explain the cost function and the power constraints involved in this study.

A. Objective Cost Function

The cost function of ED problem is formulated as follows. Firstly, determine the real output power of each on-line TGU. Secondly, determine the lowest total fuel cost of all generators over a period of time, while maintaining various nonlinear constraints [19]. The cost objective function is given by

$$\text{Minimize} \quad F_{cos\,t} = \sum_{j=1}^{N_{gen}} F_j(P_j) \tag{1}$$

where $F(P_j)$ is the cost function of the *j*th TGU in (\$/h), P_j is the real output power of the *j*th TGU in (MW), and N_{gen} is the number of online TGUs of the power system. The cost function of each TGU is related to the real power delivered into the system, and its performace is given by a quadratic function [16]

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{2}$$

where a_j, b_j, and c_j are the cost coefficients of *j*th TGU.

B. Power Constraints in SEPG

Differentt power constraints used in this work are explained below.

Constraint #1: *Power Balance Constraint*: The constraint for total power balance is expressed by

$$\sum_{j=1}^{N_{gen}} P_j = P_D + P_L \tag{3}$$

The constraint of the balance equation can be stated as the total system power generation equals to the load demand (P_D) in (MW) of the system plus the transmission network loss (P_L) in (MW). P_L is a function of real output power. In addition, P_L is usually approximated by Kron's loss formula [6] which is given by

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k + \sum_{j=1}^{N_{gen}} P_j B_{j0} + B_{00} \tag{4}$$

where j, k = 1, 2, ..., N_{gen} are the indices of the generating units , and B_{jk}, B_{j0}, B_{00} are known as the loss coefficients or *B*-coefficients.

Constraint #2: *Generation Limits*: The generation limit is given by

$$P_{j,min} < P_j < P_{j,max} \qquad j = 1, 2, ..., N_{gen} \tag{5}$$

This requires that the power generation of each unit varies between its minimum $P_{j,min}$ and its maximum P_j *max* generation limits.

Constraint #3: *Ramp Rate Limits*: In real operation, the operating range of all on-line TGUs is restricted by their ramp rate limits due to the physical limitation of TGUs [6]. The inequality constraints due to ramp limits are given by

i) If power generation increases, then

$$P_j - P_j^0 \leq UR_j \tag{6}$$

ii) If power generation decreases, then

$$P_j^0 - P_j \leq DR_j \tag{7}$$

where P_j^0 in (MW) is the previous active output power, UR_j in (MW) is the up-ramp limit of the *j*th generator; and DR_j in (MW) is the down-ramp limit of the *j*th generator. By substituting (6) and (7) in (5), we obtain

$$max\left\{P_{j,min},\left(P_j^0-DR_j\right)\right\}\le P_j\le min\left\{P_{j,max},\left(P_j^0+UR_j\right)\right\} \quad (8)$$

Let us consider,

$$P_{j,low}=max\left\{P_{j,min},\left(P_j^0-DR_j\right)\right\}, \text{ and} \quad (9)$$

$$P_{j,high}=min\left\{P_{j,max},\left(P_j^0+UR_j\right)\right\} \quad (10)$$

where $P_{j,low}$ and $P_{j,high}$ are the new lower and higher limits of unit *j*, respectively. Ramp rate is a function of resource size. The generators must be able to increase/decrease the power output at a specific rate per minute. For example, TGU has 50 MW installed capacity, the specified ramp rate range is 3 MW per minute. Then, the 50 MW TGU has 3 MW per minute ramp rate [22].

Constraint #4: *Porhibited Operating Zones* (POZs): Due to steam valve operation or vibration in a shaft bearing of TGU, the generation system contains some prohibited operating zones (POZs) [6]. Due to these constraints, discontinuities are produced in cost curves corresponding to the POZs. In this case, it is difficult to determine the shape of the cost curve under POZs through actual performance testing. Therfore, the best solution for the on-line TGUs in the power system must avoid POZs. Consequently, considering the constraint #2 in (5), the feasible operating zones of the *j*th generating thermal units are given by

$$\begin{gathered} P_{j,min}\le P_j\le P_{j,1}^l \\ P_{j,k-1}^u\le P_j\le P_{j,k}^l \quad k=2,3,...,N_{pz,j} \\ P_{j,N_{pz,j}}^u\le P_j\le P_{j,max} \end{gathered} \quad (11)$$

where $P_{j,k}^l$ and $P_{j,k}^u$ are the lower and upper bound of the *k*th POZs of the *j*th generating unit, and $N_{pz,j}$ is the number of prohibited zones of the *j* th generating unit.

Incorporating these constraints in (8), (9) and (10), we obtain the final set of power constraints as given below

$$\begin{gathered} P_{j,low}\le P_j\le P_{j,1}^l, \\ P_{j,k-1}^u\le P_j\le P_{j,k}^l \quad k=2,3,...,N_{pz,j}, \\ P_{j,N_{pz,j}}^u\le P_j\le P_{j,high} \end{gathered} \quad (12)$$

III. Orthogonal PSO Learning Algorithm

Here we briefly introduce the PSO algorithm and explain the proposed OPSO algorithm.

A. The PSO *Algorithm*

The PSO algorithm is a global optimization method. The particles inside swarm represent to the possible solutions in the dimensional search space. The population strategy depends on the iterative learning algorithm. The learning strategy of PSO algorithm is as follows. Firstly, each particle flying in the search space adjusts its flying trajectory according to two guides, its personal experience and its neighborhood's best experience. Secondly, when looking for a global optimum, each particle learns from its own historical experience and its neighborhood's historical experience. In this method, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. This method is called "global practical swam optimization" [23], because the position of each particle is influenced by the best-fit particle in the entire swarm. The following steps explain the learning strategy of the PSO algorithm.

***Step* 1:** Let us consider a swarm population with *m* particles ($N_{particle}=m$) searching for a solution in *d*-dimensional space, where $m>1$. The objective of the PSO algorithm is to minimize the objective function $f(x)$.

***Step* 2**: Each particle *i* ($i=1,2,...,m$) in the swarm has one *d*-dimensional velocity vector V_i and one *d*-dimensional position vector X_i are given by

$$V_i=[v_{i1},v_{i2},\ldots,v_{id}] \quad (13)$$

$$X_i=[x_{i1},x_{i2},\ldots,x_{id}] \quad (14)$$

***Step* 3:** For each particle *i*, evaluate the objective function $f(x)$ using the position vector X_i.

***Step* 4:** Determine the personal position vector, $G_{i,pers}$. The $G_{i,pers}$ is a personal position vector of particle *i* that is obtained by a solution of the objective function $f(x)$ when evaluating particle *i*. The $G_{i,pers}$ is given by

$$G_{i,pers}=[g_{pi,1},g_{pi,2},\ldots,g_{pi,d}] \quad (15)$$

***Step* 5:** Determine the global best position vector, G_{best}. The G_{best} is a best particle's position vector among all personal positions vectors of whole swarm. The G_{best} is obtained by a solution that corresponds to lowest value of the *m* evaluated objective functions. The G_{best} is given by

$$G_{best}=[g_{b,1},g_{b,2},\ldots,g_{b,d}] \quad (16)$$

***Step* 6:** Consider the total number of iterations, N_{iter}. In iteration, *t*, ($t=1,2,...,N_{iter}$), a particle's velocity and position vectors are updated as follows

$$\begin{aligned} V_i(t)=V_i(t-1)&+c_1r_{1i}(G_{i,pers}(t-1)-X_i(t-1)) \\ &+c_2r_{2i}(G_{best}(t-1)-X_i(t-1)) \end{aligned} \quad (17)$$

$$X_i(t)=X_i(t-1)+V_i(t) \quad (18)$$

where c_1 and c_2 are coefficients whose values are chosen experimentally from [0, 2.5]. The r_{1i} and r_{2i} are two randomly generated values with range [0, 1].

***Step* 7:** Each particle *i* is evaluated using the objective function $f(x)$ and using the position vector $X_i(t)$ (18).

***Step* 8:** In every iteration, the $G_{i,pers}$ and G_{best} are updated according to (19) and (20).

$$G_{i,pers}(t)=\left\{\begin{matrix} G_{i,pers}(t-1) \ if \ f(X_i(t))>f(G_{i,pers}(t-1)) \\ X_i(t) \quad\quad if \ f(X_i(t))\le f(G_{i,pers}(t-1)) \end{matrix}\right\} \quad (19)$$

$$G_{best}(t)=min\{G_{i,pers}(t)\} \quad (20)$$

***Step* 9:** Finally, at the end of iterations, the optimal solution of $f(x)$ is given by the global best position vector, G_{best} (t) (20).

B. The Proposed OPSO *Algorithm*

In PSO algorithm, each particle updates its velocity and position vectors according to its personal best position and its neighborhood's best position. However, this learning strategy can cause "Oscillation" phenomenon [23]. The oscillation becomes more salient with high d-dimensional search space. The oscillation is due to linear combination of the personal effect and the neighborhood effect. Subsequently, the particle becomes in a confused state variable to decide which direction to go. So, the search ability of the original PSO algorithm becomes inefficient. In this case, the particles suffer from local optimum trapping due to premature convergence and insufficient capability to find extreme points and lack of efficient mechanism to treat the constraints. In order to overcome this limitation, we propose OPSO algorithm to improve PSO algorithm.

The OPSO algorithm gives a new topology inside the swarm called "orthogonal particle formation" (OPF). In a swarm population with m particles ($m \geq d$), each particle i has a position vector X_i and a velocity vector V_i, as given in (13) and (14), respectively. The OPSO algorithm updates the position and velocity vectors in each iteration using the following steps:

***Step* 1:** For each particle, evaluate the cost using the position vector in the cost function (1). Select the best particle that has the minimum cost. Let us denote its position vector as X_{best}. Similarly, obtain the next m–1 best position vectors. Arrange these m best vectors in ascending order of costs.

***Step* 2:** Construct a matrix A of dimension m×d such that each row occupies one of the m best position vectors in the same ordered sequence. Then, convert the matrix A to matrix B, such that matrix B is a symmetric matrix of dimension d×d using the pseudocode given in Fig. 1.

```
Procedure: Convert to a symmetric matrix
for $i$ = 1: $d$
    $B(1, i) = A(1, i)$
    $B(i, 1) = A(1, i)$
end for
for $k$ = 2: $d$
    for $i$ = 2: $d$
        $B(k, i) = A(k, i)$
        $B_1(k, i) = B(k, i)$
        $B(i, k) = B_1(k, i)$
    end for
end for
```

Fig. 1. Pseudocode for converting matrix A to symmetric matrix B

***Step* 3:** Convert matrix B to an orthogonal matrix C of dimension d×d, using Gram–Schmidt orthogonalization method [24].

***Step* 4:** Apply orthogonal diagonalization (OD) process [24] which has the following properties.

1) *If matrix B is a real, symmetric matrix of dimension d×d, then matrix B is diagonalizable.*
2) *Matrix B is orthogonally diagonalizable if there exists an orthonormal matrix C such that D is orthogonal diagonal symmetric matrix.*

***Step* 5:** Obtain an orthogonal diagonal matrix D of dimension d×d using (21)

$$D = CBC^T \tag{21}$$

Let D_i denotes the ith row of the matrix D, (i = 1, 2, ..., d).

***Step* 6:** Update the position and velocity vectors of the selected particles. Thus, the d best position and velocity vectors as obtained in *Step* 1 are updated as follows.

$$V_i(t) = V_i(t-1) + c\,r_i(D_i(t-1) - X_i(t-1)) \tag{22}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{23}$$

where c is an acceleration coefficient which is chosen experimentally from [0, 2.5], r_i is a random number within range [0, 1], i = 1, 2, …, d, and t is a current iteration, t = 1, 2, ..., N_{iter}, N_{iter} is number of iterations. Because of using only one guide D_i in the particle's velocity vector V_i, the d particles move in one direction in successive iterations toward the target. This means that OVs are used to control movement of the d particles that have the possible solutions. This will avoid the oscillation phenomenon and trapping into local minima.

At end of N_{iter}, the position vector X_{best} as computed in *Step* 1 provides the optimal solution. Instead of using two guides ($G_{i,pers}$ and G_{best}) in the PSO algorithm (17), the OPSO algorithm uses only one guide and has less number of control parameters. A flowchart of the OPSO algorithm with major computational steps is shown in Fig. 2.

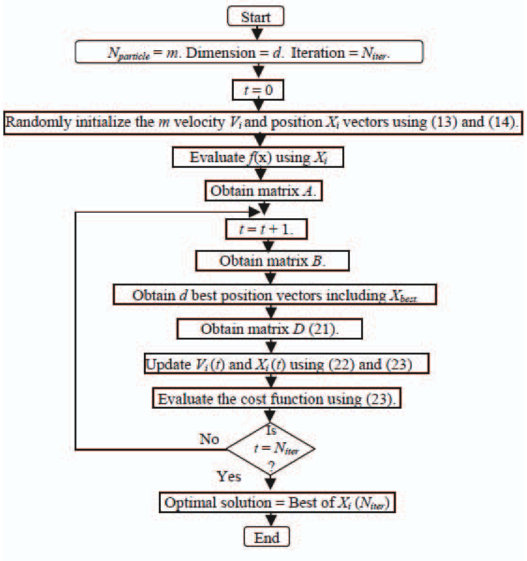

Start
$N_{particle} = m$. Dimension = d. Iteration = N_{iter}.
$t = 0$
Randomly initialize the m velocity V_i and position X_i vectors using (13) and (14).
Evaluate $f(x)$ using X_i
Obtain matrix A.
$t = t + 1$.
Obtain matrix B.
Obtain d best position vectors including X_{best}
Obtain matrix D (21).
Update $V_i(t)$ and $X_i(t)$ using (22) and (23)
Evaluate the cost function using (23).
No
Is $t = N_{iter}$?
Yes
Optimal solution = Best of X_i (N_{iter})
End

Fig. 2. Flowchart of OPSO algorithm

C. A Simple Example

To explain performance of the PSO algorithm and the proposed OPSO algorithms, a simple example is illustrated. Let us consider a 2-dimentional function given by $f(x, y) = x^2+y^2+3$. The objective function $f(x, y)$ is to find values x and y so the value of the $f(x, y)$ is minimized. With a range of [-100 100], for x and y, the function is plotted in Fig. 3. Note that the function is minimized to 3 when $x = 0$ and $y = 0$. Both the PSO and OPSO algorithms use same parameters as follow.

1) 10 particles in the swarm and using $N_{iter} = 10$.
2) $c_1 = c_2 = c = 2.0$.

Figure 4 shows that the best particle in the original PSO algorithm moves forward and backward between 3rd iteration (-44.3, 31.6) and 4th iteration (16, -10.2) causing oscillations. In the OPSO algorithm, the best particle moves steadily from the initial position to the solution in the 10th iteration.

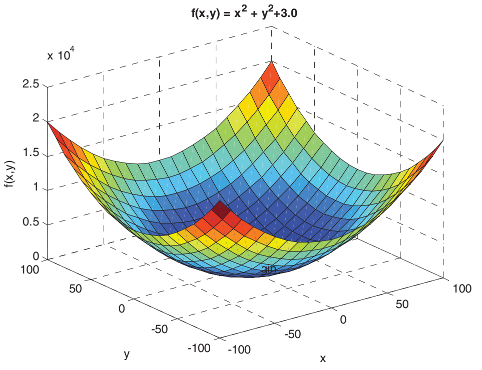


Fig. 3. Plot of function $f(x, y)$.

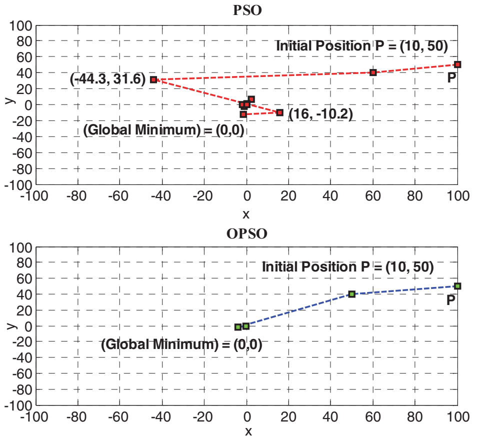


Fig. 4. Movement of best particle with iteration.

IV. Application of OPSO Algorith to ED Problem

Here we describe the simulation results carried out on a 40 TGUs power system with several constraints.

A. The Power System Specification

In order to show performance of the OPSO algorithm to solve large-scale ED problem, a realistic power system in Taiwan, named Taipower system is considered. It is complex and consists of 40 mixed-generating units, coal-fired, gas-fired, gas-turbine with complex cycle, diesel generating units and nuclear generating units [6]. The load demand of this system is 8,550 MW. There are total 46 prohibited zones for 25 TGUs are shown in Table I. This means, there are 47 inequality constraints according to (12). In normal steady-state operation, the B-coefficients with 100 MVA base is considered. The B-coefficients are generated randomly as is done in [18]. The B-coefficients (B_{jk}) of dimension 40×40 are listed in Appendix. The B_{j0} and B_{00} are neglected in our calculation.

TABLE I. TGU Capacity with Coefficients and Ramp Rate Limits and Prohibited Zones of Taipower System

Unit	P_j^0 MW	$P_{j,min}$ MW	$P_{j,max}$ MW	a_i \$/h	b_i \$/MWh	c_i \$/MW²h	UR_j MW	DR_j MW	POZs MW
1	50	40	80	170.77	8.336	0.03073	35	60	
2	60	60	120	309.54	7.0706	0.02028	40	70	[80,85]
3	150	80	190	369.03	8.1817	0.00942	50	90	[82,88]
4	24	24	42	135.48	6.9467	0.08482	42	42	
5	42	26	42	135.19	6.5595	0.09693	42	42	
6	75	68	140	222.33	8.0543	0.01142	40	75	
7	100	110	300	287.71	8.0323	0.00357	65	100	[155,162][221,235]
8	152	135	300	391.98	6.9990	0.00492	65	100	
9	200	135	300	455.76	6.6020	0.00573	65	100	[235,246]
10	100	130	300	722.82	12.908	0.00605	65	100	[200,211]
11	300	94	375	635.20	12.986	0.00515	55	95	[213,220]
12	300	94	375	654.69	12.796	0.00569	55	95	[213,220]
13	150	125	500	913.40	12.501	0.00421	80	120	[201,211][290,310] [413,425]
14	200	125	500	1760.4	8.8412	0.00752	80	120	[205,217][306,318] [409,420]
15	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290] [402,412]
16	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290] [402,412]
17	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290] [402,412]
18	400	220	500	647.85	7.9691	0.00313	70	110	[307,321][407,421]
19	400	220	500	649.69	7.9550	0.00313	70	110	[301,310][421,431]
20	398	242	500	647.83	7.9691	0.00313	70	110	[340,351][421,431]
21	398	242	500	647.81	7.9691	0.00313	70	110	[340,351][421,431]
22	390	254	550	785.96	6.6313	0.00298	70	110	[306,320][440,445]
23	390	254	550	785.96	6.6313	0.00298	70	110	[306,320][440,445]
24	390	254	550	794.53	6.6311	0.00284	70	110	[370,390][495,502]
25	390	254	550	794.53	6.6311	0.00284	70	110	[370,390][495,502]
26	390	254	550	801.32	7.1032	0.00277	70	110	[380,410][501,520]
27	390	254	550	801.32	7.1032	0.00277	70	110	[380,410][501,520]
28	20	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
29	20	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
30	30	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
31	30	20	70	1207.8	13.052	0.25098	70	70	
32	40	20	70	810.79	21.887	0.16766	70	70	
33	40	20	70	1247.7	10.244	0.2635	70	70	
34	25	20	70	1219.2	8.3707	0.30575	70	70	
35	25	18	60	641.43	26.258	0.18362	60	60	
36	20	18	60	1112.8	9.6956	0.32563	60	60	
37	20	20	60	1044.4	7.1633	0.33722	60	60	
38	25	25	60	832.24	16.339	0.23915	60	60	
39	25	25	60	832.24	16.339	0.23915	60	60	
40	25	25	60	1035.2	16.339	0.23915	60	60	

B. Experimental Results

In [6], several optimization methods, for example, GA, DE, ACSA, AIS, HBC, FA, CPSO, APSO, HPSOWM and RDPSO have been tested on Taipower system are listed in Table II. These methods were not taken into account the POZs corresponding to each TGU in their computations so that there are less inequality constraints for the ED problem. In addition, due to lack of *B* coefficients data of Taipower system, the P_L was not considered by these methods in [6].

The two authors [18] and [19] also tested Taipower system but with the POZs and P_L and are listed in Table II.

The results in Table II represent the minimum, maximum and mean cost. In addition, standard deviation (σ) is also listed. The best result in terms of minimum cost is from [19] MIQCQP. However, RDPSO algorithm [6] is the best in terms of mean, maximum costs and σ. NA represents that the results are not available in corresponding reference.

In addition to our proposed OPSO algorithm to solve ED problem under power grid constraints, we tested Taipower system by PSO algorithm. The test was carried out on the same parameters, $c_1 = c_2 = 2$, $N_{particle} = 100$, $d = 40$, $N_{run} = 25$, and $N_{iter} = 10{,}000$ in each independent run.

TABLE II. PERFORMANCE OF 12 ECTS METHODS

Algorithm	Min.Cost ($/h)	Max. Cost ($/h)	Mean Cost ($/h)	σ
GA [6]	133,435.69	136,274.97	135,012.4	729.35
DE [6]	129,915.56	137,042.94	130,600.2	1335.43
ACSA [6]	131,167.34	134,923.62	132,844.71	741.08
AIS [6]	130,133,92	132,703.18	131,482.27	561.79
HBC [6]	130,337.72	132,999.88	131,733.94	589.80
FA [6]	130,948.84	134,997.92	133,511.45	747.36
CPSO [6]	129,638.45	134,184.26	130,812.04	651.06
APSO [6]	130,861.52	130,044.63	132,587.84	675.03
HPSOWM [6]	129,717.35	132,303.59	130,858.67	591.76
RDPSO [6]	128,864.45	**131,129.0**	**129,482.0**	**568.93**
λ-logic [18]	129,777.53	NA	NA	NA
MIQCQP [19]	**128,395.29**	NA	NA	NA

The results in Table III shows, OPSO algorithm is the best in minimum, maximum and mean cost and has lowest σ while comparing it with PSO algorithm and other optimization methods listed in Table II. The OPSO algorithm gives superior result in minimum cost when comparing it with MIQCQP method. The minimum cost is reduced by 1.42%, i.e., a saving of $1,827.23/h. Moreover, by comparing mean cost of OPSO algorithm with mean cost computed by RDPSO algorithm, the mean cost is reduced by 1.85%, i.e., saving of $2,400.4/h. Furthermore, it has the lowest σ compared to all other 12 optimization methods as well as PSO algorithm. This gives evidence that the OPSO algorithm is more stable and more robustness.

TABLE III. RESULTS OF PSO AND OPSO ALGORITHMS OF TAIPOWER SYSTEM OVER 25 RUNS

Algorithm	Min. Cost $/h	Max. Cost $/h	Mean Cost $/h	σ
PSO	138,298.67	468,520.0	327,806.2	87,771.26
OPSO	**126,567.97**	**127,583.90**	**127,081.60**	**267.246**

Figure 5 shows the convergence characteristics of OPSO and PSO algorithms. Fig. 5 (a) shows essential average of the mean cost over 25 independent runs. The OPSO algorithm has better convergence properties than the PSO algorithm. Fig. 5 (b) shows the distribution of minimum costs over 25 independent runs. It shows that OPSO algorithm is more stable than the PSO algorithm in getting the optimal solution.

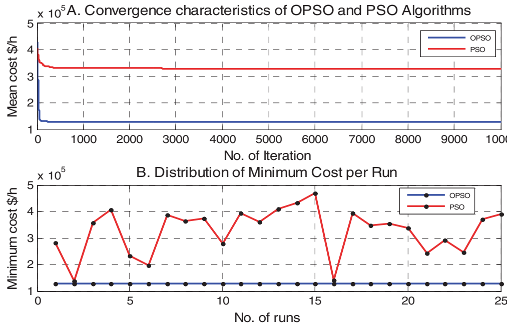


Fig. 5. Convergence characteristics of OPSO and original PSO algorithms of Taipower system

The above results provide evidence that the performance of OPSO algorithm substantially improved over the PSO algorithm.

The reasons that made PSO algorithm inefficient in global search ability can be summarized as follow:

1) The original PSO algorithm that was discussed in Section III, all particles learn from $G_{i,pers}$ and G_{best} in updating their velocities and positions. The guide G_{best} moves fast, and this makes original PSO algorithm inefficient to find new positions.
2) The G_{best} restricts whole swarm and leads the swarm to the permature convergence.
3) The fluctuation between $G_{i,pers}$ and G_{best} makes the PSO algorithm lacks synchronization. Then, the G_{best} movement is oscillatory and out of control.
4) Early convergence of PSO algorithm in high-dimensional objective function (e.g., d = 40) has made PSO algorithm leading to non-optimal solution.

Table IV lists the solution vector P_j (j = 1, 2, ..., 40) corresponding to the best solution for PSO and OPSO algorithms applied to the 40 TGUs Taipower system. Note that the load demand is 8,550 MW. Both PSO and OPSO algorithms solve the 47 inequality constraints, also both algorithms avoid the POZs. However, the OPSO algorithm is able to achieve a better solution, i.e., optimization power generation schedule that fits to the load demand while satisfying all constraints.

Table V shows the comparison between OPSO algorithm and RDPSO [6], λ-logic [18] methods, and PSO algorithm in terms of transmission loss P_L. The P_L of OPSO and PSO algorithms are calculated by using (4). The results shown in Table V reveal that the OPSO algorithm obtains on lowest value of P_L compared among other three algorithms.

TABLE IV. OPTIMIZED POWER DELIVERED BY EACH TGU USING PSO AND OPSO ALGORITHMS

Generator Output Power (MW)		
Generator #	PSO	OPSO
1	60.00	79.76
2	96.84	100.00
3	186.17	190.00
4	39.67	40.09
5	28.26	37.18
6	68.37	115.00
7	153.29	153.42
8	203.28	217.00
9	264.86	265.00
10	130.40	161.97
11	350.78	349.05
12	354.45	355.00
13	228.68	221.63
14	276.34	280.00
15	266.84	270.00
16	269.99	270.00
17	265.85	270.00
18	469.96	470.00
19	467.04	470.00
20	466.63	468.00
21	439.92	468.00
22	459.97	460.00
23	459.82	460.00
24	407.88	459.99
25	444.47	460.00
26	456.58	460.00
27	452.75	460.00
28	99.09	51.45
29	75.77	26.29
30	86.60	22.23
31	64.37	61.90
32	63.01	58.08
33	59.61	46.12
34	48.55	41.76
35	57.22	40.00
36	59.89	40.00
37	42.91	40.00
38	55.32	51.81
39	59.44	57.42
40	59.11	40.00
Total Output Real Power	8,600	**8588.15**

TABLE V. COMPARISION OF TRANSMISSION NETWORK LOSS RESULTS

Algorithm	P_L MW
RDPSO [6]	73.54
λ-logic [18]	87.40
PSO	38.20
OPSO	**38.18**

The equality power constraint in (3) has been satisfied. Total system power generation equals to P_D plus P_L as given by

$$\text{OPSO algorithm:} \quad 8{,}588.15 = 8{,}550 + 38.18$$

$$\text{PSO algorithm:} \quad 8600.00 \neq 8{,}550 + 38.20$$

The PSO is unable to solve the equality power constraint given in (3) of Taipower system. However, OPSO algorithm solves the equality power constraint.

The OPSO algorithm overcomed all problems that made PSO algorithm inefficient to get acceptable results. In addition, the OPSO algorithm appeared superior in solving a large-scale economic dispatch problem under power grid constraints compared with other ECTs mentioned in literature.

V. CONCLUSION

In this paper, we proposed a novel approach, called orthogonal PSO algorithm to solve economic dispatch of power of a large-scale thermal generating units in smart electric power grid environment. In the proposed OPSO algorithm, the orthogonal diagonalization process and orthogonality are applied to *d* particles that have the possible solutions in the swarm. The *d* orthogonal vectors are used as to guide *d* particles to improve their directions in each iteration.

The OPSO algorithm was tested on the Taipower system, a large-scale 40-unit power system. The results showed that OPSO algorithm has a better performance in terms of the minimum, maximum and mean cost compared to several competitive algorithms. The OPSO algorithm also showed better stability and robustness.

We have shown that OPSO algorithm is able to solve the equality and inequality power constraints imposed on the Taipower system and also able to avoid all prohibited operating zones.

REFERENCES

[1] K. T. Chaturvedi, M. Pandit, and L. Srivastava, "Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch," *IEEE Trans. Power Syst*., vol. 23, no. 3, pp. 1079–1087, August 2008.

[2] S. S. Reddy, P. R. Bijwe, and A. R. Abhyankar, "Real-time economic dispatch considering renewable power generation variability and uncertainty over scheduling period," *IEEE Syst., Journal,* vol. 9, no. 4, pp. 1440–1451, December 2008.

[3] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE Trans. Power Syst*., vol. 5, no. 4, pp. 1248–1259, April 1990.

[4] R. N. Dhar and P. K. Mukherjee, "Reduced-gradient method for economic dispatch," *Proc. Inst. Elec. Eng*., vol. 120, no. 5, pp. 608–610, May 1973.

[5] Z. X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses," *IEEE Trans Power Syst*., vol. 7, no. 2, pp. 544–550, February 1992.

[6] J. Sun *et al*., "Solving the power economic dispatch problem with generator constraints by random drift practicle swarm optimization," *IEEE Trans. Power Syst*., vol. 13, no. 1, pp. 519–526, February 2014.

[7] P.-H. Chen and H.-C. Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Trans. Power Syst*., vol. 10, no. 4, pp. 1919–1926, November 1995.

[8] M. F. Zaman *et al*., "Evolutionary algorithms for dynamic economic dispatch problems," *IEEE Trans. Power Syst*., vol. 31, no. 2, pp. 1486–1495, March 2016.

[9] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw*., vol. 4. December 1995, pp. 1942–1948.

[10] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromach. Human Sci*., 1995, pp. 39–43.

[11] T. Sum-Im, "Economic dispatch by ant colony search algorithm," in *Proc. IEEE Conf. Cybe. Intell. Syst*., December 2004, vol. 1, pp. 416–421.

[12] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial immune system for units with valve-point effect," *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 4, pp. 868–874, 2011.

[13] T. Niknam and F. Golestaneh, "Enhanced bee swarm optimization algorithm for dynamic economic dispatch," *IEEE Syst. J.*, vol. 7, no. 4, pp. 754–762, 2013.

[14] T. Niknam, R. Azizipanah-Abarghooee, and A. Roosta, "Reserve constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm," *IEEE Syst. J.*, vol. 6, no. 4, pp. 635–646, 2012.

[15] J. Cai, X. Ma, L. Li, and H. Peng, "Chaotic particle swarm optimization for economic dispatch considering the generator constraints," *Energy Conversion Manag*., vol. 48, pp. 645–653, 2007.

[16] A. I. Selvakumar and K. Thanushkodi, "Anti-predatory particle swarm optimization: Solution to nonconvex economic dispatch problems," *Elect. Power Syst. Res*., vol. 78, pp. 2–10, 2008.

[17] S. H. Ling *et al*.,"Hybrid particle swarm optimization with wavelet mutation and its industrial applications," *IEEE Trans. Syst., Man, Cyber. B*, vol. 38, no. 3, pp. 743–763, June 2008.

[18] T. Adhinarayanan and M. Sydulu, "Efficient lambda logic based optimization procedure to solve the large scale generator constrained economic dispatch problem," *J. Eng. Technol*., vol. 4, no. 3, pp. 301–309, 2009.

[19] T. Dind *et al*., "A bi-level branch and bound method for economic dispatch with disjiont prohibited zones considering network losses," *IEEE Trans Power Syst*., vol. 30, no. 6, pp. 2841–2855, November 2015.

[20] L. T. Al Bahrani and J. C. Patra, "Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications," in *Proc. IJCNN Conf. Neural Netw*., 2015, pp. 1–7.

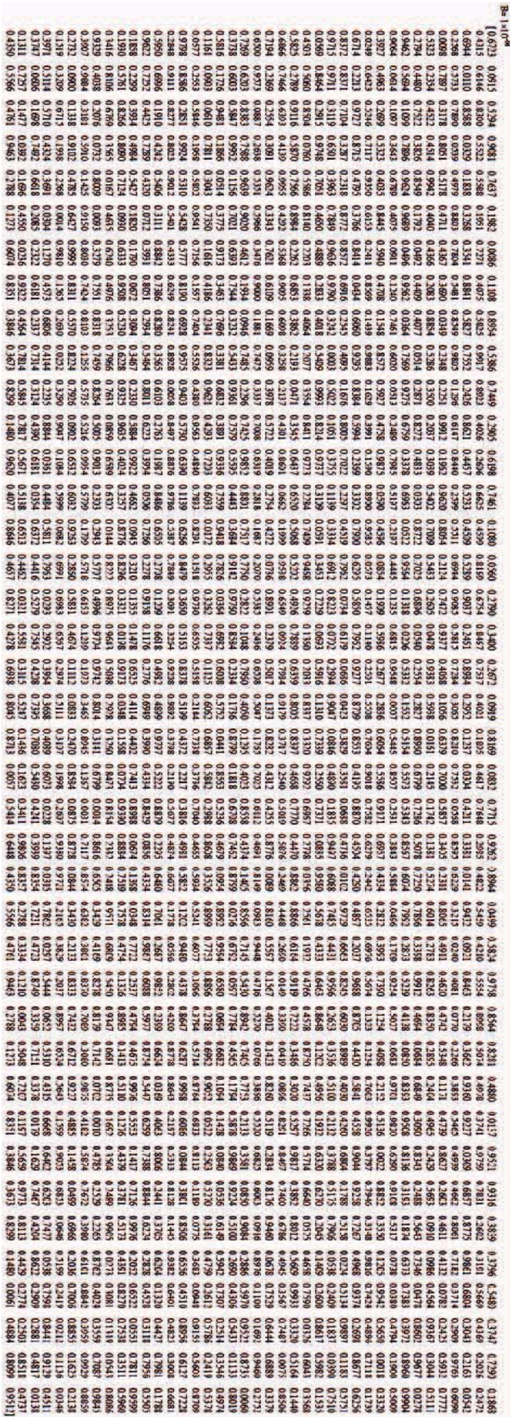
[21] L. T. Al Bahrani and J. C. Patra, "Orthogonal PSO algorithm for economic dispatch of power under power grid constraints, " in *Proc. IEEE Int. Conf. on Syst., Man, and Cybern.,* (*SMC*) 2015, pp. 14–19.

[22] Hawaiian Electric Company. (2013, February, 22). Invitation for low cost renewable energy projects on Oahu through request for waiver from competitive bidding [Online]. Available: http://www .hawaiianelectric.com.

[23] Z. H. Zhan *et al*., "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput*., vol. 15, no. 6, pp. 832–847, Dec. 2011.

[24] S. A. Dianat and E.S. Saber, Advanced linear algebra for engineers with MATLAB. New York, USA: CRC Press, 2009.

APPENDIX



Paper D: IEEE (ISGT-Asia) 2016

Multi-gradient PSO algorithm for economic dispatch of thermal generating units in smart grid

Published in:	Innovative Smart Grid Technologies - Asia (ISGT-Asia), 2016 IEEE
Date of Conference:	28 Nov.-1 Dec. 2016
Date Added to IEEE *Xplore*:	26 December 2016
Electronic ISBN:	978-1-5090-4303-3
USB ISBN:	978-1-5090-4302-6
Print on Demand(PoD) ISBN:	978-1-5090-5228-8
Electronic ISSN:	2378-8542
INSPEC Accession Number:	16555309
DOI:	**10.1109/ISGT-Asia.2016.7796395**
Publisher:	IEEE
Conference Location:	Melbourne, VIC, Australia

2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)
Melbourne, Australia, Nov 28 - Dec 1, 2016

Multi-Gradient PSO Algorithm for Economic Dispatch of Thermal Generating Units in Smart Grid

Loau Tawfak Al Bahrani[1], Jagdish C. Patra[1], and Ryszard Kowalczyk[1,2]
[1]Faculty of Science, Engineering and Technology
Swinburne University of Technology, Melbourne, Australia
[2]Warsaw School of Information Technology, Warsaw, Poland
E-mail: lalbahrani@swin.edu.au; JPatra@swin.edu.au; rkowalczyk@swin.edu.au

***Abstract*—We propose a novel algorithm called, multi-gradient particle swarm optimization (MG-PSO), for solving economic dispatch (ED) problem of thermal generating units (TGUs) under smart power grid constraints. The curve of cost function of TGUs becomes non-convex when these are subjected to ramp rate limits and prohibited operating zones. The proposed MG-PSO algorithm is able to solve such complex problem. In MG-PSO algorithm, different negative gradients are used. These negative gradients are used as guides for *m* particles in the search of global minima. The diversity in negative gradients is a key of the MG-PSO algorithm. Due to this diversity, the *m* particles cover largest search area as much as possible. The velocity vectors of the *m* particles are significantly affected by only one negative gradient called, the best negative gradient among all used negative gradients. This makes the *m* particles adjust their positions and improve their direction according to the best negative gradient. The performance of the MG-PSO algorithm has been verified on 6 and 15 TGUs test systems. The proposed MG-PSO algorithm gives good quality and promising results in solving the ED problem. In addition, the MG-PSO algorithm produces better results in terms of fitness values when compared with PSO algorithm and other optimization techniques.**

Keywords—Multi-gradient PSO algorithm, thermal generating units, smart power grid, power constraints, economic dispatch.

I. Introduction

Solving the economic dispatch (ED) problem helps in making significant savings in smart power grid (SPG). The aim of ED is to minimize the total generation cost of on-line thermal generating units (TGUs), while satisfying SPG constraints. The practical formulation of ED problem involves a non-convex cost function due to the ramp rate limits (RRLs) and prohibited operating zones (POZs). These power constraints result in the cost curve of TGU with discontinuities and high order nonlinearities. Therefore, the exact formation of the cost function under SPG constraints gives correct details about the production cost and scheduling TGUs to meet the load demand.

In order to treat a non-convex problem with the cost function of TGU, a wide variety of the evolutionary computation techniques (ECTs) based on random search have been proposed over the last few decades. Some of ECTs include genetic algorithm (GA) [1], evolutionary algorithm (EA) [2], particle swarm optimization (PSO) algorithm [3], [4], ant colony search (ACS) algorithm [5], artificial immune system (AIS) [6], honey bee colony (HBC) algorithm [7], and firefly algorithm (FA) [8]. These techniques impose a few or no restrictions on the shape of a cost function. However, they are often prone to get trapped into local optima when applied to multiple prohibited zones.

To enhance the global search ability to solve ED problem under multiple power constraints, several ECTs have been developed in the last decade including PSO based algorithms. For example, orthogonal PSO (OPSO) algorithm has been proposed to solve ED problem under different power constraints [9], [10]. A fully decentralized approach (DE) uses three stages, one to achieve consensus among agents and the second and third stages are used for solving ED problem [11]. The chaotic PSO (CPSO) method combines PSO with an adaptive inertia weight factor and chaotic local search to solve ED problem [12]. The anti-predatory PSO (APSO) applies anti-predatory behavior, which guides the swarm to escape from the predators [13]. The hybrid PSO wavelet mutation (HPSOWM) uses the wavelet-theory-based mutation to enhance PSO algorithm in exploration and searching for a better solution [14]. However, hybrid methods are often time-consuming due to the complex algorithm structure and finding an appropriate integration of hybrid algorithm is difficult. The random drift PSO (RDPSO) is inspired by a free electron model in the metal conductors placed in an external electric field [15]. The RDPSO uses a set of evolution equations to enhance the PSO global search ability. The simulated annealing PSO (SA-PSO) algorithm uses probabilistic jumping to prevent obtaining infeasible solution [16]. A mixed-integer quadratically constrained quadratic programming (MIQCQP) uses a bi-level branch and bound method to solve ED problem [17]. The modified PSO (MPSO) has been used for a nonconvex ED problem [18].

Some improved PSO variants in the literature (e.g., CPSO [12] and MPSO [18]) use another approach, called the inertia weight factor and time-varying inertia weight factors, respectively, as a controller on the velocity vector of each particle. The objective of both factors is the control on the impact of the previous velocity of the *m* particles on the current iteration. Therefore, different equations have been used to describe the weight factor. In this paper, we propose a novel algorithm called multi-gradient PSO (MG-PSO) to solve ED problem, considering the generation limits, RRLs, the POZs and transmission network loss (P_L) in SPG environment.

The MG-PSO algorithm uses several negative gradients. The particle's velocity is clearly affected according to best

978-1-5090-4303-3/16/$31.00 ©2016 Crown

negative gradient among all used negative gradients. We have shown that MG-PSO algorithm is able to solve ED problem under TGU constraints quite effectively.

The rest of this paper is organized as follows. We present the problem formulation in Section II. An explanation of MG-PSO algorithm is provided in Section III. In Section IV, we present the application of MG-PSO algorithm to ED problem. Finally, the conclusion of this study is given in Section V.

II. Problem Formulation

Here, we explain the cost function and the power constraints imposed on SPG involved in this study.

A. Objective Cost Function

The objective of an ED problem is to find the optimal allocation of output real power of on-line TGUs over a period of time in order to minimize the total generation cost while satisfying the equality and inequality power constraints [15]. The cost function can be stated mathematically as

$$\text{Minimize} \quad F_{cos\,t} = \sum_{j=1}^{N_{gen}} F(P_j) \tag{1}$$

where $F(P_j)$ is the cost function of jth TGU in \$/h, P_j is the output real power of jth TGU in MW, and N_{gen} is the number of on-line TGUs. The cost function of each TGU is related to the output real power delivered into SPG and specified by a quadratic function [13] as follows:

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{2}$$

where a_j, b_j, and c_j are the cost coefficients of jth TGU.

B. Power Constraints in SPG

Different power constraints imposed on TGUs in SPG used in the literature are explained below.

1) Power Balance Constraint: The equality constraint of the power balance can be stated as the total power generation equals to the load demand (P_D) in MW plus the transmission network loss (P_L) in MW. This is expressed by

$$\sum_{j=1}^{N_{gen}} P_j - P_D - P_L = 0 \tag{3}$$

The P_L is a function of the output real power of TGU and is given by [19]

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k \tag{4}$$

where $B_{jk,}$ are known as the loss coefficients or B-coefficients.

2) Generation Limits: The generation limits of each TGU is given by

$$P_{j,min} < P_j < P_{j,max} \qquad j = 1, 2, ..., N_{gen} \tag{5}$$

This requires that the power generation of each TGU remains between its minimum $P_{j,min}$ and its maximum $P_{j,max}$ limits.

3) Ramp Rate Limits Constraint: The operating range of all on-line TGUs is restricted by their ramp rate limits (RRLs) due to the physical limitation of TGUs [13]. In addition, TGUs cannot change their output power immediately. A change in TGU output power from one specific interval to the next cannot exceed a specified limit, as follows:

- If power generation increases, then

$$P_j - P_j^0 \leq UR_j \tag{6}$$

- If power generation decreases, then

$$P_j^0 - P_j \leq DR_j \tag{7}$$

where P_j^0 is the TGU output power at the previous interval and P_j is the TGU output power at current interval. The UR_j and DR_j are the up-ramp and down-ramp limits of unit j, respectively, in MW/h. By substituting (6) and (7) in (5), we obtain

$$max\left\{P_{j,min}, \left(P_j^0 - DR_j\right)\right\} \leq P_j \leq min\left\{P_{j,max}, \left(P_j^0 + UR_j\right)\right\} \tag{8}$$

Let us assume that,

$$P_{j,low} = max\left\{P_{j,min}, \left(P_j^0 - DR_j\right)\right\}, \text{ and} \tag{9}$$

$$P_{j,high} = min\left\{P_{j,max}, \left(P_j^0 + UR_j\right)\right\} \tag{10}$$

where $P_{j,low}$ and $P_{j,high}$ are the new lower and higher limits of unit j, respectively.

4) Prohibited Operating Zone Constraint: The physical limitations due to the steam valve operation or vibration in a shaft bearing of TGU may result in the generation units operating within prohibited zones (POZs) [16]. Due to presence of POZs, discontinuities are produced in the cost curve corresponding to POZs. In this case, it is difficult to determine the shape of the cost curve under POZs through actual performance testing. Therefore, the best solution is, the TGU that contains POZs avoids these prohibited zones. By using (5) mentioned in constraint number 2, the feasible operating zones of the jth TGU are given by

$$\begin{gathered} P_{j,min} \leq P_j \leq P_{j,1}^l \\ P_{j,k-1}^u \leq P_j \leq P_{j,k}^l \qquad k = 2, 3, ..., N_{pz,j} \\ P_{j,N_{pz,j}}^u \leq P_j \leq P_{j,max} \end{gathered} \tag{11}$$

where $P_{j,k}^l$ and $P_{j,k}^u$ are the lower and upper bound of the kth POZs of the jth unit, and $N_{pz,j}$ is the number of prohibited zones of the jth unit. Incorporating these power constraints in (8), (9) and (10), we get the final set of constraints as follows:

$$\begin{gathered} P_{j,low} \leq P_j \leq P_{j,1}^l, \\ P_{j,k-1}^u \leq P_j \leq P_{j,k}^l \qquad k = 2, 3, ..., N_{pz,j}, \\ P_{j,N_{pz,j}}^u \leq P_j \leq P_{j,high} \end{gathered} \tag{12}$$

III. Multi-Gradient PSO Algorithm

Here we briefly introduce the PSO algorithm and explain the proposed MG-PSO algorithm.

A. The PSO Algorithm

The PSO algorithm is a global optimization technique. The population (swarm) is distributed randomly and using iterative approach to reach global optimum. The particles inside swarm refer to the possible solutions in multi-dimensional search space. The PSO algorithm depends on; firstly, each particle flying in the search area adjusts its flying trajectory according to two guides, its personal experience and its neighborhood's best experience. Secondly, when seeking a global solution, each particle learns from its own historical experience and its neighborhood's historical experience. In such a case, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. This PSO algorithm is named, global PSO [3], [4], because the position of each particle is affected by the best-fit particle in the entire swarm. The following steps explain the learning strategy of the PSO algorithm mentioned and described in [4] that is the original PSO.

***Step* 1:** Let us consider a swarm population with m particles ($N_{particle} = m$) searching for a solution in d-dimensional space, where $m > 1$. The objective of the PSO algorithm is to minimize an objective function $F(P_j)$.

***Step* 2**: Each particle i (i = 1, 2, ..., m) in the swarm has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i are given by

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{13}$$

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{14}$$

***Step* 3:** For each particle i, evaluate the objective function $F(P_j)$ using the position vector X_i.

***Step* 4:** The $G_{i,pers}$ is a personal position vector of particle i that is obtained by evaluating the objective function $F(P_j)$. The $G_{i,pers}$ is given by

$$G_{i,pers} = [g_{pi,1}, g_{pi,2}, \ldots, g_{pi,d}] \tag{15}$$

***Step* 5:** Determine the global best position vector, G_{best}. The G_{best} is a best particle's position vector among all personal positions vectors of whole swarm. The G_{best} is obtained by a solution that corresponds to lowest value of the m evaluated objective functions. The G_{best} is given by

$$G_{best} = [g_{b,1}, g_{b,2}, \ldots, g_{b,d}] \tag{16}$$

***Step* 6:** Consider the total number of iterations, N_{iter}. In iteration t, t = 1, 2, ..., N_{iter}, a particle's velocity and position vectors are updated as follows:

$$\begin{aligned} V_i(t) = V_i(t-1) &+ c_1 r_{1i}(G_{i,pers}(t-1) - X_i(t-1)) \\ &+ c_2 r_{2i}(G_{best}(t-1) - X_i(t-1)) \end{aligned} \tag{17}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{18}$$

where c_1 and c_2 are coefficients whose values are chosen experimentally from [0, 2.5]. The r_{1i} and r_{2i} are two randomly generated values within the range [0, 1].

***Step* 7:** Each particle i is evaluated using the objective function $f(x)$ and using the position vector $X_i(t)$ (18).

***Step* 8:** In every iteration, the $G_{i,pers}$ and G_{best} are updated according to (19) and (20).

$$G_{i,pers}(t) = \begin{Bmatrix} G_{i,pers}(t-1) \; if \;\; f(X_i(t)) > f(G_{i,pers}(t-1)) \\ X_i(t) \qquad\quad if \;\; f(X_i(t)) \leq f(G_{i,pers}(t-1)) \end{Bmatrix} \tag{19}$$

$$G_{best}(t) = min\{G_{i,pers}(t)\} \tag{20}$$

***Step* 9:** Finally, at the end of iteration, the optimal solution of $F(P_j)$ is given by the global best position vector, $G_{best}(t)$ (20).

B. The Proposed MG-PSO *Algorithm*

We propose an algorithm, called the multi-gradient PSO (MG-PSO). The mechanism of MG-PSO algorithm depends on the following considerations.

Consider m particles descend at a particular negative gradient at a position X after they were flying in the space in searching for food. However, the food may be few or not found in position X. Therefore, they decide to change their direction to another gradient that has a steeper negative straight line within another position (e.g., position Y). The position Y may better than the position X. Then, after several times of different negative gradients, the m particles obtain a best position that corresponds a best negative gradient among all used negative gradients. This diversity in gradients (multiple gradients) generates steeper and less steep slopes. In such a case, the m particles have gained ability to coverage larger search space area. Subsequently, the m particles are guaranteed find the food.

Let us consider $grad_i$, i = 1, 2, ..., N_{grad} are negative gradients. In each $grad_i$ we introduce two variables, the first variable called, *time* and denoted by (t). The t represents the iteration (t = 1, 2, ..., N_{iter}). The second variable called *velocity decay factor* (*vdf*). The *vdf* decreases progressively with increase in t. The change in t is Δt and the change in *vdf* is Δvdf. The negative gradients $grad_i$ are given by

$$grad_i = \frac{\Delta vdf(i)}{\Delta t} \quad i = 1, 2, ..., N_{grad} \tag{21}$$

where N_{grad} is number of negative gradients. The *vdf* at t, is given by

$$vdf(t) = vdf_{initial} \times (1 - \frac{t}{N_{iter}}) + vdf_{final} \times (\frac{t}{N_{iter}}) \tag{22}$$

where $vdf_{initial}$ and vdf_{final} are real and positive numbers within a range [0, 1] and $vdf_{initial} > vdf_{final}$.

The following steps explain the learning strategy of MG-PSO algorithm.

***Steps* 1-5**: Same as PSO algorithm as in Section IIIA.

***Step* 6**:

For i = 1, 2, ..., N_{grad}

Choose a set of $vdf_{initial}$ and vdf_{final} for each gradient $grad_i$, i = 1, 2, …, N_{grad}.

Choose number of iterations N_{iter}, t = 1, 2, …, N_{iter}.

Determine $grad_i$, i = 1, 2, …, N_{grad}, using (21).

For each iteration, update the particle's velocity and position vectors as follows:

$$\begin{aligned} V_i(t) = vdf(t)\; V_i(t-1) &+ c_1 r_{1i}(G_{i,pers}(t-1) - X_i(t-1)) \\ &+ c_2 r_{2i}(G_{best}(t-1) - X_i(t-1)) \end{aligned} \tag{23}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{24}$$

where c_1 and c_2 are coefficients whose values are chosen by trail and error method from [0, 2.5]. However, the best values of c_1 and c_2 depend mainly on the experimental test. The r_{1i} and r_{2i} are two randomly generated values with range [0, 1]. **Evaluate** the particle's performance by substituting (24) in the objective function $F(P_j)$.

Determine $G_{i,pers}(t)$, $G_{best}(t)$ using (19) and (20), respectively.

End For

Repeat for all negative gradients.
End For

***Step* 7:** Select the negative gradient with the chosen $vdf_{initial}$ and vdf_{final} that gives the best fitness value.

It may be seen from (17) and (23) that the basic difference between PSO and MG-PSO algorithms lies on the update procedure of V_i. In PSO algorithm while updating, the previous value of V_i is added with the guidance obtained from its own personal experience and the global experience. Whereas, in the case of MG-PSO algorithm, the *vdf* of the best negative gradient is multiplied with the previous value of V_i while adding with the personal experience and the global experience. The multiplication of *vdf* to the velocity diminishes the contribution of previous velocity while updating and thus improves the performance of PSO algorithm. The *vdf* is introduced in the MG-PSO algorithm in order to better control the *m* particles' ability to exploration and exploitation.

IV. Application of MG-PSO Algorithm to ED Problem

Here we describe the simulation results carried out on two power systems with several SPG constraints.

A. Test Case 1: *Power system* with 6 TGUs (PS-1)

1) Details of PS-1

The PS-1 consists of 6 TGUs, 26 buses and 46 transmission lines [15]. This system is a small-scale with six dimensions for its ED problem. There are 12 POZs of 6 TGUs, which yield 13 inequality constraints according to (12). The data of PS-1, TGU capacity and coefficients, RRLs and POZs of TGU and *B*-coefficients were listed in [15]. At steady-state operation, the maximum load demand is 1,263 MW. The computations are achieved with 100 MVA base capacity.

2) Comparison in terms of fitness values

In [15], the 10 ECTs (e.g., GA, DEA, ACSA, AIS, HBC, FA, CPSO, APSO, HPSOWM and RDPSO) that have been tested on PS-1 are listed in Table I. In addition, the fitness values of [11], [16] and [17] are also presented in Table I. In addition to our proposed MG-PSO algorithm to solve ED problem under SPG constraints, we tested PS-1 by PSO algorithm. Therefore, the total ECTs that have been tested by PS-1 are 15.

The PS-1 is a small-scale and it is easy for MG-PSO algorithm to obtain the global optimum. Thus, we select only two negative gradients $N_{grad} = 2$ with a set of initial and final of *vdf* that corresponds to $grad_i$, $i = 1, 2$. The set parameters of MG-PSO algorithm are ([$grad_1 = -0.09$, $vdf_{initial} = 1$, $vdf_{final} = 0.1$], [$grad_2 = -0.07$, $vdf_{initial} = 1$, $vdf_{final} = 0.3$]. The other parameters were used out on the same with PSO algorithm, $c_1 = c_2 = 2$, $N_{particle} = 20$, $d = 6$, $N_{run} = 25$, and $N_{iter} = 3{,}000$ in each independent run. The results in Table I show that the MG-PSO algorithm provides the best result in terms of the minimum, maximum and mean costs and has lowest standard deviation (σ) when compared with PSO algorithm and other ECTs. This gives evidence that the MG-PSO algorithm is more stable and robust.

TABLE I. Cost Performance of The Fifteen ECTs for PS-1

Algorithm	**Min.Cost** ($/h)	**Max. Cost** ($/h)	**Mean Cost** ($/h)	**σ**
DE [11]	15,449.58	15,449.65	15,449.61	NA
GA [15]	15,445.59	15,491.47	15,465.17	9.73
DEA [15]	15,444.94	15,472.06	15,450.13	6.98
ACSA [15]	15,445.30	15,511.52	15,459.51	12.02
AIS [15]	15,446.32	15,481.27	15,456.66	7.39
HBC [15]	15,444.58	14,482.39	15,457.94	8.48
FA [15]	15,445.94	15,501.39	15,461.30	9.33
CPSO [15]	15,442.98	15,466.39	15,449.12	5.8
APSO [15]	15,445.51	15,538.60	15,473.31	12.90
HPSOWM [15]	15,442.82	15,502.63	15,455.62	15.88
RDPSO [15]	15,442.75	15,455.29	15,445.02	2.28
SA-PSO [16]	15,447.00	15,455.00	15,447.00	2.52
MIQCQP [17]	15,443.07	NA	NA	NA
PSO	15,447.09	15,449.60	15,447.65	0.56
MG-PSO	**15,442.65**	**15,442.65**	**15,442.65**	**0.00**

3) Convergence characteristics of MG-PSO *and* PSO *algorithms*

Figure 1 shows the convergence characteristics of MG-PSO and PSO algorithms. Fig. 1 (a) shows average of the mean cost over 25 independent runs. The MG-PSO algorithm has better convergence properties than the PSO algorithm. Fig. 1 (b) shows the distribution of minimum costs over 25 independent runs. It shows that MG-PSO algorithm is more stable than the PSO algorithm in obtaining the global solution.

4) Comparison in terms of inequality and equality constraints

Table II lists the solution vector, P_j (j = 1, 2, ..., 6) corresponding to the best solution for MG-PSO and PSO algorithms. Both MG-PSO and PSO algorithms were able to solve the 13 power inequality constraints (12), also both algorithms avoid the 12 POZs of 6 TGUs. In addition, the MG-PSO and PSO algorithm operate within RRLs of each TGU.

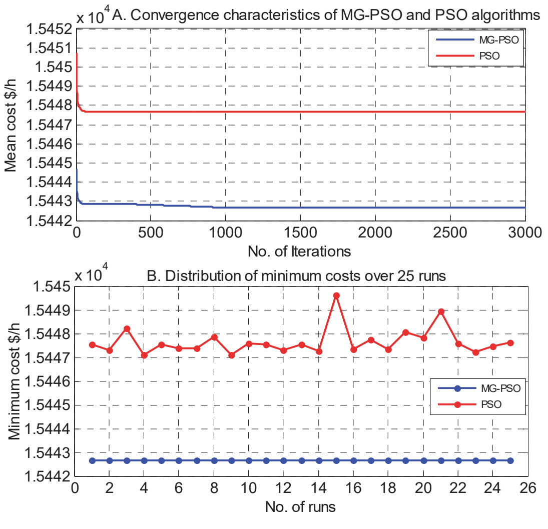


Fig. 1. Convergence characteristics of MG-PSO and PSO algorithms for PS-1.

TABLE II. Optimized Power Dispatch by Each TGU Using MG-PSO and PSO Algorithms for PS-1

Algorithm	Optimum output power (MW)						Total Output Power (MW)
	P_1	P_2	P_3	P_4	P_5	P_6	
PSO	441.95	169.39	258.53	138.46	161.00	106.04	1,275.3999
MG-PSO	447.06	173.18	262.92	139.05	165.57	86.61	1,275.4157

Table III shows the comparison in terms of power balance constraint among MG-PSO algorithm and several ECTs. The load demand P_D is 1,263 MW. Total output power generated as shown in Table II and the P_L is computed using (4) and then these values are substituted in (3) to determine any mismatch from zero. The results presented in Table III show that MG-PSO and MIQCQP [17] algorithms have satisfied zero mismatch in solving power balance constraint. However, other ECTs listed in Table III, DE [11] and DRPSO [15], SA-PSO [16] and PSO techniques have obtained on mismatch closer to zero.

B. Test Case 2: Power system with 15 TGUs PS-2

1) Details of PS-2

The PS-2 is a medium-scale power system with 15 TGUs whose characteristics and the data are taken from [20]. The maximum load demand of the PS-2 at steady–state operation is 2,630 MW. The dimension of this ED problem is d =15. The PS-2 has a total of 11 POZs of 4 TGUs. Thus, there are 12 inequality power constraints (12) for this ED problem. Compared to PS-1, the ED problem of this system is relatively harder to be optimized.

2) Comparison in terms of fitness values

Eight ECTs as well as our proposed MG-PSO and PSO algorithms are listed in Table IV. These 10 optimization techniques are tested on PS-2. We choose 3 negative gradients N_{grad} = 3 with a set of initial and final of vdf that corresponds to $grad_i$, i =1, 2, 3. The set parameters of the MG-PSO algorithm are ([$grad_1$ = -0.09, $vdf_{initial}$ = 1, vdf_{final} = 0.1], [$grad_2$ = -0.07, $vdf_{initial}$ = 1, vdf_{final} = 0.3], [$grad_3$ = -0.05, $vdf_{initial}$ = 1, vdf_{final} = 0.5]), respectively. The other parameters of MG-PSO and PSO algorithms are same as in Section IVA2.

The results presented in Table IV show that the MG-PSO algorithm achieves the best result in terms of the minimum, maximum and mean cost and has lowest σ = 0.15 compared with PSO algorithm and other eight ECTs. This indicates that the MG-PSO algorithm is more stable and robust. NA represents that the results are not available in corresponding reference.

TABLE III. Power Balance Constraint Results of Six ECTs for PS-1

Algorithm	Total P_j (MW)	P_D (MW)	P_L (MW)	Mismatch (MW)
DE [11]	1,275.9300	1,263	12.9500	-0.0200
DRPSO [15]	1,275.3565	1,263	12.3598	-0.0033
SA-PSO [16]	1,275.7000	1,263	12.7330	-0.0330
MIQCQP [17]	**1,275.4400**	**1,263**	**12.4400**	**0.0000**
PSO	1,275.3999	1,263	12.4000	-0.0001
MG-PSO	**1,275.4157**	**1,263**	**12.4157**	**0.0000**

TABLE IV. Cost Performance of The Ten ECTs for PS-2

Algorithm	Min. Cost ($/h)	Max. Cost ($/h)	Mean Cost ($/h)	σ
OPSO [10]	32,669.00	32,699.00	32,688.00	7.21
ACSA [15]	32,863.17	33,256.28	33,120.02	86.16
AIS [15]	32,895.91	33,132.01	33,017.65	58.12
HBC [15]	32,789.23	33,301.49	33,030.86	69.79
FA [15]	32,898.01	33,310.72	33,116.90	96.38
RDPSO [15]	32,652.33	32,959.79	32,744.58	82.47
DEA [15]	32,718.82	33,213.31	32,966.43	110.32
SA-PSO [16]	32,708.00	32,789.00	32,732.00	NA
PSO	32,885.20	33,386.20	33,075.12	110.76
MG-PSO	**32,668.69**	**32,669.32**	**32,668.98**	**0.15**

3) Convergence characteristics of MG-PSO *and* PSO *algorithms*

The convergence characteristics of MG-PSO and PSO algorithms are shown in Fig. 2. Figure 2 (a) shows average of the mean cost over 25 independent runs. The MG-PSO algorithm is better than PSO algorithm in terms of convergence properties.

The distribution of minimum costs over 25 independent runs shown in Fig. 2 (b) shows that MG-PSO algorithm is more stable in obtaining the optimum solution than the PSO algorithm.

4) Comparison in terms of inequality and equality constraints

Table V presents the best solution vector P_j (j = 1, 2, ..., 15) obtained by MG-PSO and PSO algorithms for PS-2. Both MG-PSO and PSO algorithms solve the 12 power inequality constraints in (12) by avoiding the 11 POZs of 4 TGUs. In addition, MG-PSO and PSO algorithms work within RRLs of 15 TGUs.

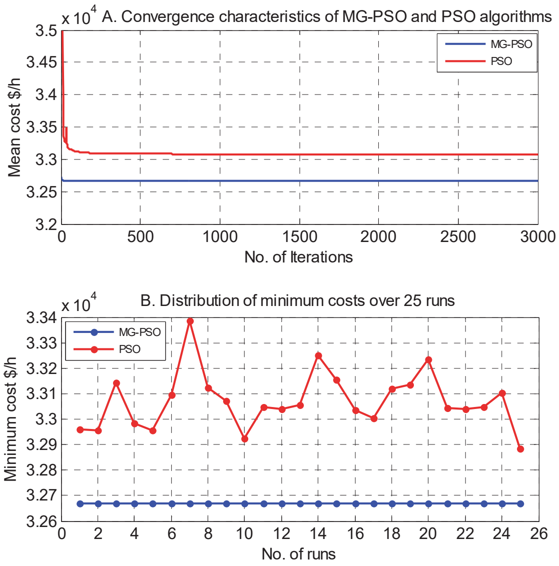


Fig. 2. Convergence characteristics of MG-PSO and PSO algorithms for PS-2.

TABLE V. OPTIMIZED POWER DISPATCH BY EACH TGU USING MG-PSO AND PSO ALGORITHMS FOR PS-2

Optimum output power (MW)					
Gen. #	P_1	P_2	P_3	P_4	P_5
PSO	455.0000	364.1374	129.2110	129.3094	168.8872
MG-PSO	455.0000	380.0000	129.9551	129.9649	170.0000
Gen. #	P_6	P_7	P_8	P_9	P_{10}
PSO	456.8272	426.5248	63.8214	107.4216	51.4106
MG-PSO	457.3041	430.0000	71.6674	58.3692	160.00
Gen. #	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
PSO	68.9185	78.2871	79.1707	30.8404	46.5101
MG-PSO	80.0000	80.0000	25.0000	15.0000	15.0000
Total output power: MG-PSO = 2,657.2606 MW, PSO = 2,656.2771 MW					

Table VI shows the comparison among MG-PSO, OPSO [10], DRPSO [15], SA-PSO [16] and PSO optimization techniques in terms of power balance constraint. The load demand P_D in this case, 2,630 MW. The P_L of MG-PSO and PSO algorithms are computed by (4). The power balance results of both algorithms are obtained by (3). The power balance results shown in Table VI appear that MG-PSO and OPSO [10] algorithms have satisfied zero mismatch within four places after decimal point. However, other ECTs, RDPSO [15], SA-PSO [16] and PSO algorithm obtained on mismatch are -0.0046, -0.0080, and 0.0304, respectively.

TABLE VI. POWER BALANCE CONSTRAINT RESULTS OF FIVE ECTS FOR PS-2.

Algorithm	Total P_j (MW)	P_D (MW)	P_L (MW)	Mismatch (MW)
OPSO [10]	**2,652.7600**	**2,630**	**22.7600**	**0.0000**
RDPSO [15]	2,655.3650	2,630	25.3696	-0.0046
SA-PSO [16]	2,660.9000	2,630	30.9080	-0.0080
PSO	2,656.2771	2,630	26.2467	0.0304
MG-PSO	**2,657.2606**	**2,630**	**27.2606**	**0.0000**

The proposed MG-PSO algorithm overcomed all problems that made PSO algorithm inefficient to get acceptable results. In addition, the MG-PSO algorithm appeared superior in solving the economic dispatch problem under SPG constraints of small- and medium-scale power systems in SPG compared with other ECTs mentioned in the literature.

V. CONCLUSION

In this paper, the multi-gradient PSO algorithm has been proposed. It has been able to effectively solve 6- and 15-thermal generating unit (TGU) of economic dispatch (ED) of power considering the multiple smart power grid (SPG) constraints.

We have shown that the MG-PSO algorithm was able to solve the equality and inequality constraints including the transmission network loss of two power systems, and avoiding all prohibited operating zones and operating within ramp rate limits.

It is evident from the minimum power dispatch results that the MG-PSO algorithm has a better performance in terms of the minimum, maximum and mean costs compared to several competitive algorithms including PSO algorithm.

The MG-PSO algorithm also showed better stability and robustness.

REFERENCES

[1] P-.H. Chen and H.-C. Chang, “Large-scale economic dispatch by genetic algorithm,” IEEE Trans. Power Syst., vol. 10, no. 4, pp. 1919–1926, November 1995.

[2] H. Yang, P. Yang, and C. Hang, “Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions,” IEEE Trans. Power Syst., vol. 11, no. 1, pp. 112–118, February 1996.

[3] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., vol. 4. December 1995, pp. 1942–1948.

[4] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in Proc. 6th Int. Symp. Micromach. Human Sci., 1995, pp. 39–43.

[5] I. Ciornei and E. Kyriakides, “A GA-API solution for the economic dispatch of generation in power system operation,” IEEE Trans. Power Syst., vol. 27, no. 1, pp. 233–242, February 2012.

[6] S. Hemamalini and S. Simon, “Dynamic economic dispatch using artificial immune system for units with valve-point effect,” Int. J. Electr. Power Energy Syst., vol. 33, no. 4, pp. 868–874, 2011.

[7] T. Niknam and F. Golestaneh, “Enhanced bee swarm optimization algorithm for dynamic economic dispatch,” IEEE Syst. J., vol. 7, no. 4, pp. 754–762, 2013.

[8] T. Niknam, R. A-Abarghooee, and A. Roosta, “Reserve constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm,” IEEE Syst. Journal, vol. 6, no. 4, pp. 635–646, 2012.

[9] L. T. Al Bahrani and J. C. Patra, “Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications,” in Proc. IJCNN Conf. Neural Netw., 2015, pp. 1–7.

[10] L. T. Al Bahrani and J. C. Patra, “Orthogonal PSO algorithm for economic dispatch of power under power grid constraints,” in Proc. IEEE Int. Conf. on Syst., Man, and Cybern., (SMC) 2015, pp. 14–19.

[11] W. T. Elsayed and E. F. El-Saadany “A fully decentralized approach for solving the economic dispatch problem,” IEEE Trans. Power Syst., vol. 30, no. 4, pp. 2179–2189, July 2015.

[12] J. Cai, X. Ma, L. Li, and H. Peng, “Chaotic particle swarm optimization for economic dispatch considering the generator constraints,” Energy Conversion Manag., vol. 48, pp. 645–653, 2007.

[13] A. Selvakumar and K. Thanushkodi, “Anti-predatory particle swarm optimization: Solution to nonconvex economic dispatch problems,” Elect. Power Syst. Research, vol. 78, pp. 2–10, 2008.

[14] S. Ling et al., “Hybrid particle swarm optimization with wavelet mutation and its industrial applications,” IEEE Trans. Syst., Man, Cyber. Part B, vol. 38, no. 3, pp. 743–763, June 2008.

[15] J. Sun, V. Palade, X.-J. Wu, W. Fang, and Z. Wang, “Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization,”IEEE Trans. Power Syst., vol. 13, no. 1, pp. 519–526, February 2014.

[16] C. Kuo, “A novel coding scheme for practical economic dispatch by modified particle swarm optimization,” IEEE Trans. Power Syst., vol. 23, no. 4, pp. 1825–1835, November 2008.

[17] T. Dind, R. Bo, F, Li, and H. Sun, “A bi-level branch and bound method for economic dispatch with disjoint prohibited zones considering network losses,” IEEE Trans. Power Syst., vol. 30, no. 6, pp. 2841–2855, November 2015.

[18] M. Basu, “Modified particle swarm optimization for nonconvex economic dispatch problems,” Elect. Power and Energy Syst., vol. 69, pp. 304–312, 2015.

[19] R. Naresh, J. Dubey, and J. Sharma, “Two-phase neural network based modelling framework of constrained economic load dispatch”, *IEE Proc.* Gen. Trans. and Dist., vol. 151, no. 3, pp. 373 – 378, May 2004.

[20] Z.-L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” IEEE Trans. Power Syst., vol. 18, no. 3, pp. 1187–1195, August 2003.

Paper E: Swarm and Evolutionary Computation 2017

A novel orthogonal PSO algorithm based on orthogonal diagonalization

Article reference:	SWEVO334
Journal:	Swarm and Evolutionary Computation
Received at Editorial Office:	20 Sep 2016
Article revised:	22 Sep 2017
Article accepted for publication:	5 Dec 2017
DOI:	10.1016/j.swevo.2017.12.004
Volume/Issue:	Will appear soon

ARTICLE IN PRESS

Swarm and Evolutionary Computation xxx (2017) 1–23

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

A novel orthogonal PSO algorithm based on orthogonal diagonalization

Loau Tawfak Al-Bahrani *, Jagdish C. Patra

Swinburne University of Technology, Melbourne, Australia

ARTICLE INFO

Keywords:
Particle swarm optimization
Orthogonal diagonalization
Orthogonal PSO
Active and passive groups

ABSTRACT

One of the major drawbacks of the global particle swarm optimization (GPSO) algorithm is zigzagging of the direction of search that leads to premature convergence by falling into local minima. In this paper, a new algorithm named orthogonal PSO (OPSO) algorithm is proposed that not only alleviates the associated problems in GPSO algorithm but also achieves better performance. In OPSO algorithm, the m particles of the swarm are divided into two groups: one active group of best personal experience of d particles and a passive group of personal experience of remaining $(m-d)$ particles. The purpose of creating two groups is to enhance the diversity in the swarm's population. In each iteration, the d active group particles undergo an orthogonal diagonalization process and are updated in such way that their position vectors are orthogonally diagonalized. The passive group particles are not updated as their contribution in finding correct direction is not significant. In the proposed algorithm, the particles are updated using only one guide, thus avoiding the conflict between the two guides that occurs in the GPSO algorithm. We tested the OPSO algorithm with thirty unimodal and multimodal high-dimensional benchmark functions and compared its performance with GPSO and several competing evolutionary techniques. With extensive simulated experiments, we have shown superiority of the proposed algorithm in terms of convergence, accuracy, consistency, robustness and reliability over other algorithms. The proposed algorithm is found to be successful in achieving optimal solution in all the thirty benchmark functions.

1. Introduction

In the recent years, several evolutionary computation techniques (ECTs) have been proposed to solve complex optimization problems. Particle swarm optimization (PSO) algorithm is one of the ECTs that was proposed by observing social behaviours among animal herding, fishes, birds and even humans. The PSO algorithm is a population based technique that emulates such behaviours. The individual members of the swarm are called “*particles*”. Each particle moves with an adaptable velocity within a multi-dimensional search space and keeps the best position it ever encountered in its memory. In 1995, Kennedy and Eberhart proposed two variants of the PSO algorithm, named global PSO (GPSO) and local PSO (LPSO) algorithms based on neighborhood topology of the particles [1,2]. In GPSO algorithm, each particle's neighborhood includes all social neighbors in the swarm. In other words, the topology of the particles in the swarm represents to a fully connected network in which the particles are attracted to the best solution found by any member of the swarm. In contrast, in LPSO algorithm, a particle uses the best historical experience of the particle in its neighborhood that is defined by a topological structure, e.g., the ring structure, the von Neumann structure, or the pyramid structure [3,4].

In GPSO and LPSO algorithms, the particles of the swarm are initialized randomly and then they search for global optimum by updating their position and velocity in each iteration. Each particle uses its personal experience and its neighborhood's best experience as two guides through a linear summation. They have been empirically demonstrated to perform well in many continuous domain optimization problems. In addition, both have a few adjustable parameters and easy to implement [4]. The GPSO algorithm is faster in convergence than LPSO algorithm. Whereas, the LPSO algorithm has lower susceptibility of the solution to be trapped into local minima. However, the main drawback of GPSO and LPSO algorithms is falling into local minima while solving complex problems with nonlinear multi-dimensional objective functions [5].

The reasons for poor performance of the PSO algorithm can be summarized as follows. Firstly, the learning mechanism of PSO algorithm depends on the fact that each particle in the swarm adjusts its search trajectory according to its personal experience and its neighbor's experiences. Therefore, each particle in a swarm obtains two possible

* Corresponding author.
E-mail addresses: lalbahrani@swin.edu.au (L.T. Al-Bahrani), JPatra@swin.edu.au (J.C. Patra).

https://doi.org/10.1016/j.swevo.2017.12.004
Received 20 September 2016; Received in revised form 22 September 2017; Accepted 5 December 2017
Available online xxxx

Please cite this article in press as: L.T. Al-Bahrani, J.C. Patra, A novel orthogonal PSO algorithm based on orthogonal diagonalization, Swarm and Evolutionary Computation (2017), https://doi.org/10.1016/j.swevo.2017.12.004

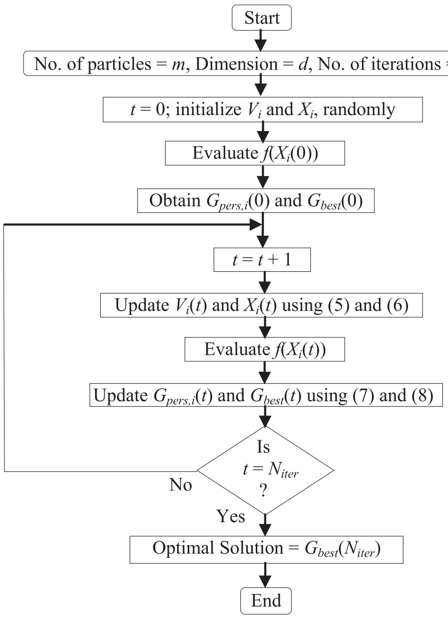

Start
No. of particles = m, Dimension = d, No. of iterations = N_{iter}
t = 0; initialize V_i and X_i, randomly
Evaluate $f(X_i(0))$
Obtain $G_{pers,i}(0)$ and $G_{best}(0)$
$t = t + 1$
Update $V_i(t)$ and $X_i(t)$ using (5) and (6)
Evaluate $f(X_i(t))$
Update $G_{pers,i}(t)$ and $G_{best}(t)$ using (7) and (8)
Is $t = N_{iter}$?
No
Yes
Optimal Solution = $G_{best}(N_{iter})$
End

Fig. 1. Flowchart of the GPSO algorithm.

solutions, one from its personal experience and the other from its neighborhood's experience and then sums them together. The problem here is not only existence of the summation, but also to maintain a coherent decision between these two guides. The two guides may have a large difference or may even have opposite directions at the early stage of search that may lead the particles to pull to a local trapping and may lead to early premature convergence or remain in opposite directions until final search stage. Thus, the particles might be still remaining far away from the global optimum. In addition, these two guides and their linear summation may cause a phenomenon called "oscillation" or "zigzagging" [6]. This phenomenon becomes more prominent with high-dimensional search space.

Only a few parameters have been used in PSO algorithm to give a potential advantage and to enhance their performance. Among user parameters of PSO algorithm, several strategies of inertia weight, e.g., constant inertia weight [7], time-varying inertia weight [8,9], and adaptive inertia weight [10–12] have been proposed. However, when the problem, i.e., objective function has multiple local minima and has a high-dimensional search space, the PSO algorithm with these few parameters becomes inefficient [6].

In order to improve the performance of PSO algorithm, several ECTs have been proposed by different researchers. These ECTs are based on different swarm topologies and mechanisms for updating velocity and position vectors. Some of the ECTs based on swarm topologies are given here. For example, fully informed PSO (FIPSO) algorithm [4] uses the neighbors in the swarm to influence on the particle's velocity under different topologies (e.g., ring, four clusters, pyramid, and square topologies) and then the information of the entire neighborhood is used to guide the particles for finding the best solution. The comprehensive learning PSO (CLPSO) algorithm tries to encourage the particle to learn from different topologies on different dimensions to maintain diversity of the swarm to discourage premature convergence [13]. A self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-T-VAC) algorithm [14] improves the performance of PSO algorithm. The aging leader challenger PSO (ALC-PSO) algorithm uses the leader of a swarm with a growing age and a lifespan to allow the other particles to challenge the leadership when the leader becomes aged [15]. The extraordinariness PSO (EPSO) uses the movement of the particles called extraordinary motion as a topology [16]. The cellular learning automata bare bones PSO technique uses probability distributions, e.g., Gaussian distribution as a topology [17]. The binary PSO (BPSO) algorithm uses a transfer function to map a continuous search space to a discrete search space by dividing the transfer functions into two families, i.e., S-shaped and V-shaped [18].

Some of the ECTs are based on the trade-off between exploration and exploitation. For example, mixed swarm cooperative PSO (MCPSO) algorithm is used to efficiently handle the trade-off between the global and local search in PSO algorithm by dividing the particles into two groups, one for exploration and the other for exploitation [19]. The heterogeneous comprehensive learning PSO (HCLPSO) algorithm is used to enhance exploration and exploitation of the particles by using comprehensive learning (CL) strategy [20]. The stability-based adaptive inertia weight (SAIW) algorithm uses an adaptive approach to determine the inertia weight for each particle based on its performance and distance from its best position to satisfy the stability of a swarm [21]. A pattern search PSO (*pk*PSO-G) algorithm makes the particles explore and exploit the promising global areas and solutions with clustering on the Euclidean spatial neighborhood structure [22]. The crisscross search particle swarm optimization (CSPSO) uses two search operators, i.e., horizontal crossover and vertical crossover and these two operators are used for global convergence and swarm stability [23]. The multiple learning PSO with space transformation perturbation (MLPSO-STP) allows each particle to learn from the average information on the personal historical best experience of all particles and from the information on best positions that are randomly chosen according to personal experience specification [24].

In the multi-function GPSO algorithm, its learning mechanism is based on the effect of population density on the search ability of PSO algorithm which is saturated when the population density exceeds a certain limit [25]. The two-swarm cooperative PSO algorithm uses two swarms, i.e., a master and a slave, for accelerating the convergence and for keeping swarm's diversity invariant [26]. The competitive and cooperative PSO with information sharing mechanism (CCPSO-ISM) algorithm allows each particle to share its best search information by using the ISM, so that all other particles in the swarm can take advantage of the shared information [27]. A directionally driven self-regulating PSO algorithm uses two strategies for the swarm; a directional update strategy and a rotational invariant strategy [28]. In the learning mechanism of a dynamic tournament topology PSO (DTT-PSO) algorithm, each particle is guided by several better solutions, chosen stochastically from the swarm [29]. An improved velocity

```
L1: Let $B$ be a real symmetric matrix of size $d×d$.
L2: Apply Gram-Schmidt orthogonalization on matrix $B$ to obtain $d$ orthonormal vectors.
L3: Construct orthonormal matrix $C$ using these vectors.
L4: Obtain the diagonal matrix $D$ using (12).
```

Fig. 2. The orthogonal diagonalization process.

ARTICLE IN PRESS

```
Procedure: Convert matrix $A(m×d)$ to
        a symmetric matrix $B(d×d)$.
for $i = 1 : d$
    $B(1, i) = A(1, i)$
    $B(i, 1) = A(1, i)$
end for
for $k = 2 : d$
    for $i = 2 : d$
        $B(k, i) = A(k, i)$
        $B1(k, i) = B(k, i)$
        $B(i, k) = B1(k, i)$
    end for
end for
```

Fig. 3. Pseudocode for converting matrix *A* to a symmetric matrix *B*.

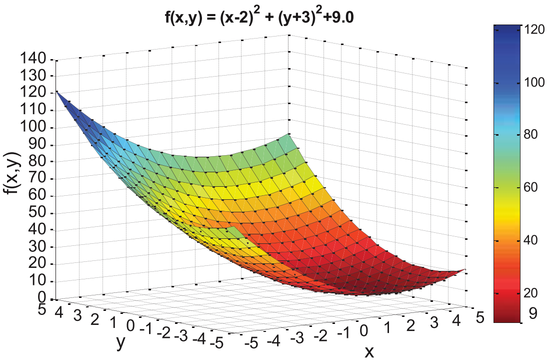


Fig. 5. The landscape of $f(x,y)$. The minimum value of the function is 9.0 at $x = 2.0$ and $y = -3.0$.

bounded Boolean PSO algorithm uses a minimum velocity parameter that makes it more effective in solving feature selection problem [30]. The termite spatial correlation PSO algorithm modifies the velocity equation in the PSO algorithm based on a termite motion mechanism and using the mutation strategy to avoid stagnation for the particles in the swarm [31]. A territorial PSO (TPSO) algorithm uses a collision operator and adaptively varying (territories) to prevent the particles from premature convergence and encourage them to explore new neighbourhoods based on a hybrid self-social metric that leads to improvement in exploration capability [32]. A competitive swarm optimizer (CSO) algorithm was proposed for large-scale optimization [33]. Here, a pairwise competition strategy is used in which the loser particle updates its own position by learning from the winner. The CCPSO2 algorithm [34] has been proposed for large-scale optimization problems in which Cauchy and Gaussian distribution are used to update the positions of the particles. In Ref. [35], the PSO algorithm was improved by social learning PSO (SL-PSO) algorithm. Here, each particle learns from any one of the better particles in the current swarm. In addition, a dimension-dependent parameter control method was used to mitigate the burden of parameter settings.

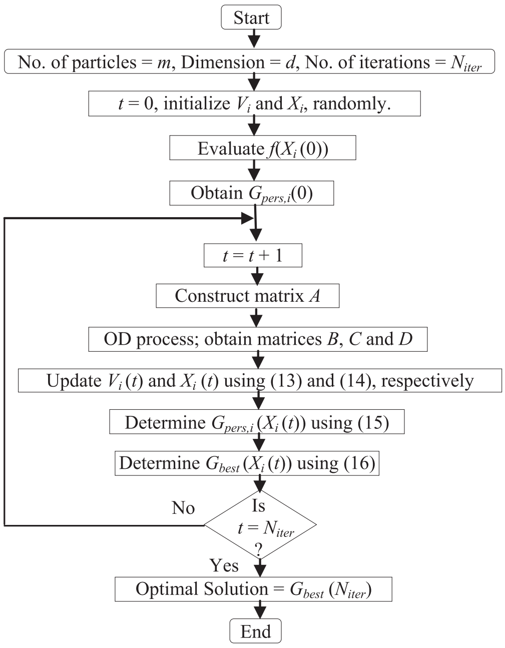

Start
No. of particles = m, Dimension = d, No. of iterations = N_iter
t = 0, initialize V_i and X_i, randomly.
Evaluate f(X_i (0))
Obtain G_pers,i(0)
t = t + 1
Construct matrix A
OD process; obtain matrices B, C and D
Update V_i (t) and X_i (t) using (13) and (14), respectively
Determine G_pers,i (X_i (t)) using (15)
Determine G_best (X_i (t)) using (16)
Is t = N_iter ?
No
Yes
Optimal Solution = G_best (N_iter)
End

Fig. 4. Flowchart of the OPSO algorithm.

$\boldsymbol{t = 1}$

$$D = \begin{bmatrix} \mathbf{-33.4275} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{9.9113} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{55.4176} & \mathbf{-31.9043} & -80.94 & -23.13 & -6.20 & 79.95 \\ \mathbf{38.2285} & \mathbf{-4.3210} & 23.47 & 23.95 & -1.65 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{-6.2957} & \mathbf{10.2996} & -23.13 & 55.41 & -80.94 & 85.23 \\ \mathbf{-1.6169} & \mathbf{23.4768} & 23.95 & 38.22 & 23.47 & -40.30 \end{bmatrix}$$

$\boldsymbol{t = 80}$

$$D = \begin{bmatrix} \mathbf{-3.7383} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.3275} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-28.0374} & \mathbf{6.7003} & 43.83 & -92.54 & 96.11 & -99.44 \\ \mathbf{-14.4267} & \mathbf{16.8004} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0074} & \mathbf{1.4628} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-2.7485} & \mathbf{-2.7543} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$\boldsymbol{t = 140}$

$$D = \begin{bmatrix} \mathbf{-4.4026} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4027} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4068} & \mathbf{0.0001} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{-0.0002} & \mathbf{3.4019} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{1.9953} & \mathbf{1.9962} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0007} & \mathbf{-2.9972} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$\boldsymbol{t = 200}$

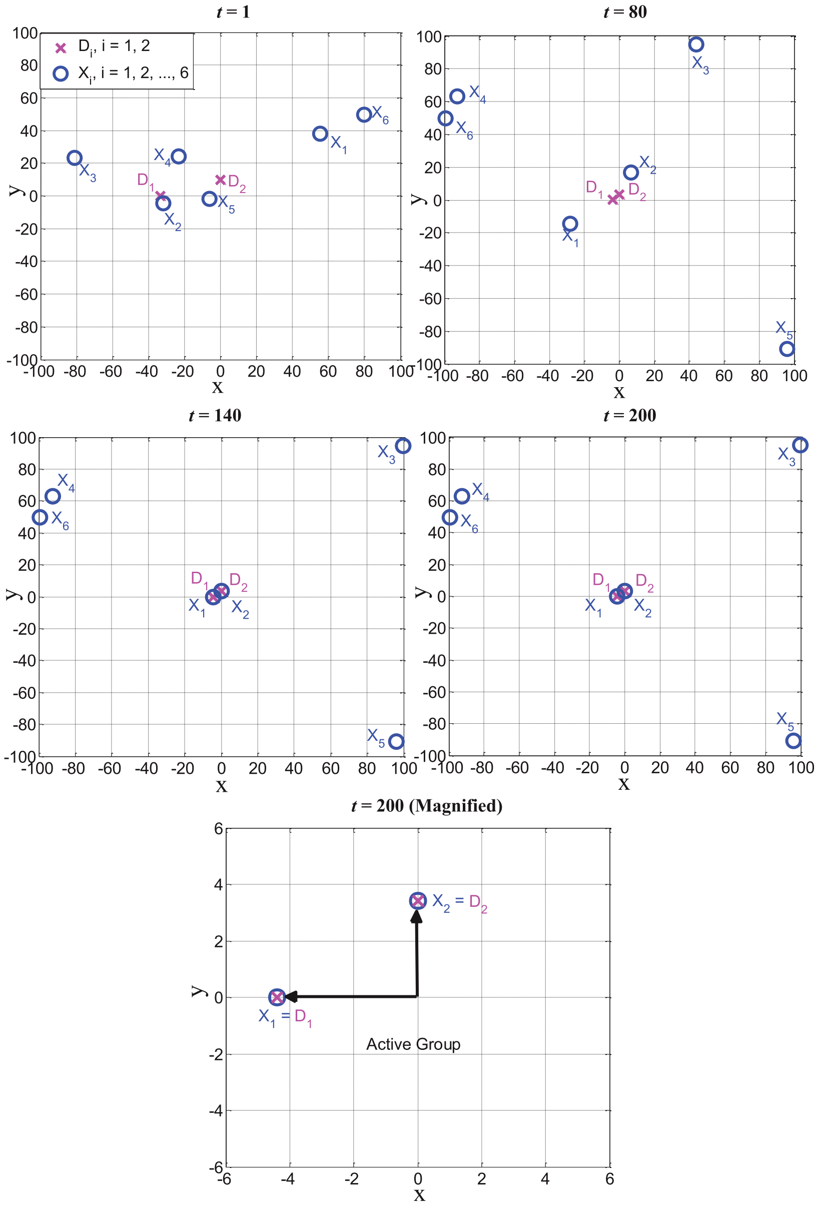
$$D = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4051} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{0.0000} & \mathbf{3.4051} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0000} & \mathbf{2.0000} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0000} & \mathbf{-3.0000} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$$\underbrace{\textbf{\textit{Active Group}}}\quad \underbrace{\textit{Passive Group}}$$

Fig. 6. Numerical example showing convergence of the OPSO algorithm.


t = 1

t = 80

t = 140

t = 200

t = 200 (Magnified)

Fig. 7. Movement of six position vectors (X_1, X_2, …, X_6) in a 2-dimensional search space ($m = 6$, $d = 2$), and two diagonal vectors, D_1 and D_2. The active group consists of X_1 and X_2. At $t = 200$, X_1 and X_2 coincide with D_1 and D_2.

Some other ECTs are based on hybrid systems. For example, the multi-swarm PSO (MsPSO) algorithm develops its searching strategy by using new parameters that are created by Takagi-Sugeno fuzzy system [36]. A dynamic feed-forward neural network is used for predictive control in which adaptive parameters are adjusted using Gaussian PSO algorithm [37]. The fuzzy PSO with cross-mutated (FPSOCM) algorithm uses fuzzy logic system that is based the knowledge of swarm behaviour to determine the inertia weight of PSO algorithm and the cross-mutation operator [38]. However, in such hybrid systems an appropriate integration between different systems may be hard to determine under complex problems.

Another group of ECTs are based on orthogonal experimental design (OED). These are, for example, two orthogonal learning PSO (OLPSO) algorithms, one for local (OLPSO-L) and another for global (OLPSO-G)

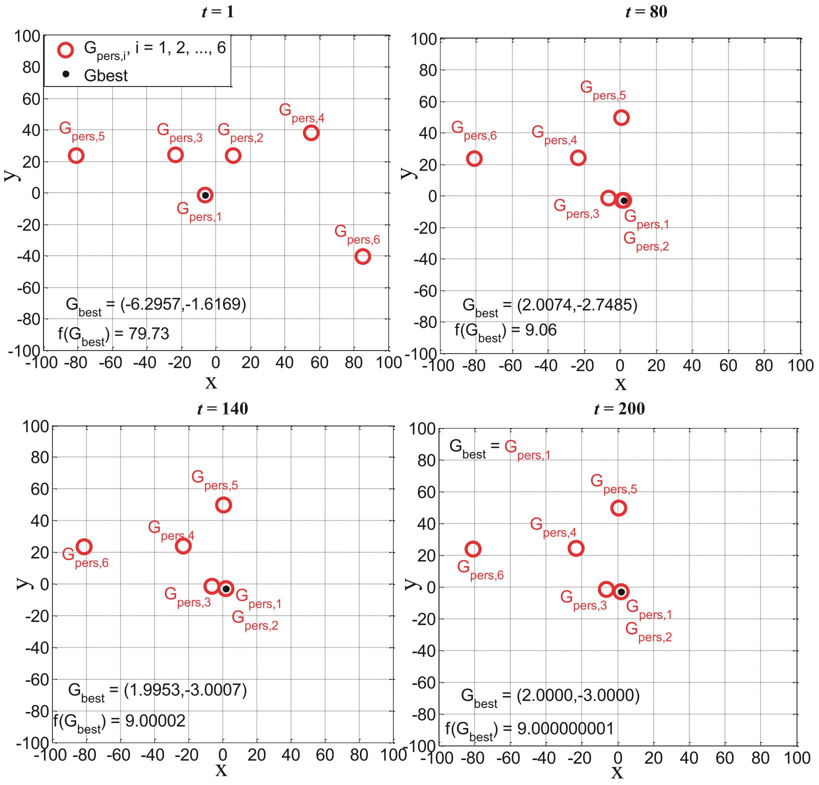


Fig. 8. Movement of personal position vectors of six particles $G_{pers,i}$, $i = 1, 2, \ldots, 6$. The active group of particles corresponds to $G_{pers,1}$ and $G_{pers,2}$. The G_{best} corresponds to $G_{pers,1}$. At $t = 200$, the G_{best} coincides with global position (2,−3).

optimization [39], orthogonal global-best-guided artificial bee colony (OGABC) algorithm [40], and an orthogonal genetic algorithm with quantization (OGAQ) [41]. The OED works on a predefined table of an orthogonal array of N factors with Q levels per factor. It allows the inputs to interact among themselves such that the output process can be optimized. Then, a set of possible solutions is obtained to achieve global optimum. However, the drawbacks of the OED-based algorithms are: firstly, it holds only when no or weak interaction among the factors exists; secondly, the table that contains variable designs is complicated; and thirdly, the orthogonality may not be possible to achieve in complex problems.

Some other ECTs based on different architecture are also available in the literature. They include differential evolution using neighborhood-based mutation (DEGL/SAW) algorithm [42], modified Gaussian bare-bones differential evolution (MGBDE) algorithm [43], estimation of distribution algorithm with local (EDAL) search [44], Latin squares based on evolutionary algorithm (LEA) [45], biogeography-based optimization (BBO) [46], and p-best adaptive fast evolutionary programming (p-best AFEP) [47].

In this paper, we propose a novel algorithm named orthogonal PSO (OPSO) algorithm with a new learning mechanism to improve the performance by overcoming the drawbacks of GPSO algorithm. The OPSO algorithm consists of a swarm with m particles that looks for the global optimal solution in a d-dimensional search space ($m > d$). The swarm population is divided into two groups: an active group of best personal experience of d particles and a passive group of personal experience of remaining ($m – d$) particles. The purpose of creating two groups is to enhance the diversity in the swarm's population. The position vectors associated with the m particles undergo an orthogonal diagonalization (OD) process in which the d orthogonal guidance vectors in the active group are obtained. In each iteration, using only one guide, the velocity and position vectors of only the active group particles are updated and the remaining ($m – d$) particles are left unchanged. This avoids the conflicting situation of the GPSO algorithm and leads the best d particles towards the optimal solution in a multi-dimensional search space. We applied the OPSO algorithm to several unimodal and multimodal benchmark functions and have shown that the OPSO algorithm is able to achieve superior performance in terms of convergence, consistency and accuracy compared to GPSO and several competitive ECTs. In our recent works, the effectiveness of the proposed OPSO algorithm has been shown for optimal power dispatch in smart power grid applications [48–50]. Our proposed OPSO algorithm is completely different from Ref. [39–41]. In OPSO algorithm, the position vectors are orthogonalized that gives rise to faster convergence and better solution. Whereases, in Ref. [39–41], the algorithm is based on OED in which the updating of the velocity and position vectors are done by using a predefined orthogonal array.

The rest of the paper is organized as follows. We briefly explain the learning strategy of the GPSO algorithm in Section 2. Details of our proposed OPSO algorithm are provided in Section 3. In Section 4, we present performance comparison between OPSO and GPSO algorithms by taking thirty unimodal, multimodal, shifted, rotated and shifted rotated benchmark functions. Performance comparison between the OPSO

Table 1
Thirty benchmark objective functions used in the study.

f	Name	Function	Search Space	Optimum (x)	Minimum $f(x)$	Accept Value
f_1	Sphere [39]	$f_1\ (x) = \sum_{i=1}^{d} x_i^2$	$[-100,100]^{30}$	$[0]^{30}$	0.0	1×10^{-6}
f_2	Elliptic [40]	$f_2(x)\ = \sum_{i=1}^{d} [(10^6)]^{\left(\frac{i-1}{d-1}\right)} x_i^2$	$[-100,100]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_3	Sum Squares [40]	$f_3\ (x) = \sum_{i=1}^{d} ix_i^2$	$[-10,10]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_4	Sum Powers [40]	$f_4\ (x) = \sum_{i=1}^{d} \lvert x_i \rvert^{(i+1)}$	$[-1,1]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_5	Schwefel's P2.22 [40]	$f_5\ (x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	$[-100,100]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_6	Step [40]	$f_6\ (x) = \sum_{i=1}^{d} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]^{30}$	$-\frac{1}{2} \le x < \frac{1}{2}$	0.0	1×10^{-8}
f_7	Quadric [37]	$f_7\ (x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	$[-100,100]^{30}$	$[0]^{30}$	0.0	1×10^{-6}
f_8	Noise [40]	$f_8\ (x) = \sum_{i=1}^{d} i(x_i)^4 + random[0,\ 1)$	$[-1.28,1.28]^{30}$	$[0]^{30}$	0.0	1×10^{-2}
f_9	Hyper Ellipsoid [37]	$f_9\ (x) = \sum_{i=1}^{d} \sum_{j=1}^{i} (x_j)^2$	$[-100,100]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{10}	Rosenbrock [40]	$f_{10}\ (x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10,\ 10]^{30}$	$[1]^{30}$	0.0	1×10^{2}
f_{11}	Griewank [39]	$f_{11}\ (x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{12}	Levy [39]	$f_{12}\ (x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10\sin^2(\pi w_i + 1)]$ $+ (w_d - 1)^2 [1 + \sin^2(2\pi\omega_d)], w_i = 1 + \frac{x_i - 1}{4}$	$[-10,\ 10]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{13}	Himmelblau [40]	$f_{13}\ (x) = \frac{1}{d} \sum_{i=1}^{d} (x_i^4 - 16x_i + 5x_i)$	$[-5,\ 5]^{100}$	NA	−78.3323	−78.0
f_{14}	Alpine [40]	$f_{14}\ (x) = \sum_{i=1}^{d} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	$[-10,\ 10]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{15}	Weierstrass [40]	$f_{15}(x) = \sum_{i=1}^{d} [\sum_{k=1}^{k_{\max}} a^k \cos(2\pi b^k (x_i + 0.5)) - d\sum_{k=1}^{k_{\max}} a^k \cos(\pi b^k)]$ $a = 0.5,\ b = 3,\ k_{\max} = 20$	$[-0.5,\ 0.5]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{16}	NCRastrigin [40]	$f_{16}\ (x) = [y_i^2 - 10\cos(2\pi y_i) + 10] y_i = \left\{ \begin{array}{cc} x_i & \lvert x_i \rvert < \frac{1}{2} \\ \frac{round(2x_i)}{2} & \lvert x_i \rvert \ge \frac{1}{2} \end{array} \right\}$	$[-5.12,5.12]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{17}	Michalewics [40]	$f_{17}\ (x) = -\sum_{i=1}^{d} \sin(xi) \sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0,\ \pi]^{100}$	NA	−99.2784	−95.0
f_{18}	Schwefel [40]	$f_{18}\ (x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500,500]^{30}$	$[420.9687]^{30}$	0.0	2×10^{03}
f_{19}	Rastrigin [40]	$f_{19}\ (x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12,5.12]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{20}	Ackley [40]	$f_{20}\ (x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right)$ $-\exp\ \frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)\Bigg) + a + \exp(1), a = 20, b = 0.2, c = 2\pi$	$[-32,\ 32]^{30}$	$[0]^{30}$	0.0	1×10^{-8}
f_{21}	Shifted Sphere [55]	$f_{21}(x) = \sum_{i=1}^{d} Z_i^2 - 450,$ $Z = x - O,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-100,100]^{30}$	O	–450	–350
f_{22}	Shifted Schwefel problem 1.2 [55]	$f_{22}(x) = \sum_{i=1}^{d} \sum_{j=1}^{i} (Z_j)^2 - 450,$ $Z = x - O,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-100,100]^{30}$	O	–450	–350
f_{23}	Shifted Rosenbrock [55]	$f_{23}(x) = \sum_{i=1}^{d-1} [100(Z_{i+1} - Z_i^2)^2 + (Z_i - 1)^2] + 390,$ $Z = x - O + 1,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-100,100]^{30}$	O	390	490
f_{24}	Shifted Rastrigin [55]	$f_{24}(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)] - 330,$ $Z = x - O,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-5,5]^{30}$	O	–330	−230
f_{25}	Shifted Ackley [55]	$f_{25}(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} Z_i^2}\right)$ $-\exp\ \frac{1}{d}\sum_{i=1}^{d} \cos(cZ_i)\Bigg) + a + \exp(1) - 140, a = 20, b = 0.2, c = 2\pi$ $Z = x - O,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-32,32]^{30}$	O	–140	−135
f_{26}	Shifted Griewank [55]	$f_{26}\ (x) = \frac{1}{4000} \sum_{i=1}^{d} Z_i^2 - \prod_{i=1}^{d} \cos\left(\frac{Z_i}{\sqrt{i}}\right) + 1 - 180$ $Z = x - O,\ x = [x_1, x_2, \ldots, x_d],\ O = [o_1, o_2, \ldots, o_d]$	$[-600,600]^{30}$	O	–180	−170

(*continued on next page*)

Table 1 (*continued*)

f	Name	Function	Search Space	Optimum (x)	Minimum $f(x)$	Accept Value
f_{27}	Rotated Rosenbrock [56]	$f_{27}(x) = \sum_{i=1}^{d-1}[100(Z_{i+1} - Z_i^2)^2 + (Z_i - 1)^2] - 900,$ $Z = M\left(\frac{2.048\times(x-O)}{100}\right) + 1, x = [x_1, x_2, \ldots, x_d],$ $O = [o_1, o_2, \ldots, o_d]$	$[-100,100]^{30}$	O	–900	−800
f_{28}	Shifted Rotated High Conditioned Elliptic [53]	$f_{28}(x) = \sum_{i=1}^{d}[(10^6)]^{\left(\frac{i-1}{d-1}\right)} x_i^2 - 450,$ $Z = (x - O) \times M, x = [x_1, x_2, \ldots, x_d], O = [o_1, o_2, \ldots, o_d]$	$[-100,100]^{30}$	O	–450	–350
f_{29}	Shifted Rotated Rastrigin [53]	$f_{29}\ (x) = 10d + \sum_{i=1}^{d}[Z_i^2 - 10\cos(2\pi Z_i)],$ $Z = (x - O) \times M, x = [x_1, x_2, \ldots, x_d], O = [o_1, o_2, \ldots, o_d]$	$[-5,5]^{30}$	O	−330	−300
f_{30}	Shifted Rotated Griewank [53]	$f_{26}\ (x) = \frac{1}{4000}\sum_{i=1}^{d} Z_i^2 - \prod_{i=1}^{d}\cos\left(\frac{Z_i}{\sqrt{i}}\right) + 1 - 180$ $Z = (x - O) \times M, x = [x_1, x_2, \ldots, x_d], O = [o_1, o_2, \ldots, o_d]$	$[-600,600]^{30}$	O	–180	−180

algorithm and several ECTs reported by other authors is provided in Section 5. Finally, conclusion of this study is given in Section 6.

2. The GPSO algorithm

Consider the GPSO algorithm given in Refs. [1] and [2] as a fundamental technique of PSO algorithm. The learning mechanism of GPSO algorithm depends on the distribution of the particles (possible solutions) in the swarm and their updating procedure. Firstly, each particle flying in a d-dimensional search area adjusts its flying trajectory according to two guides; its personal experience ($G_{pers,i}$) and its neighborhood's best experience (G_{best}). Secondly, when seeking the global optimum, each particle learns from its own historical experience and its neighborhood's historical experience. In such a case, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. Since the position of each particle is affected by the best-fit particle in the entire swarm, this technique is named as global PSO [1]- [2]. The following steps explain the mechanism of the GPSO algorithm.

Let us consider a swarm population with m particles searching for an optimal solution in a d-dimensional search space. Each particle i ($i = 1, 2, \ldots, m$) has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i. The objective of the GPSO algorithm is to minimize the given objective function $f(x)$.

Initialization: Iteration, $t = 0$.

Step 1: For each particle, i ($i = 1, 2, \ldots, m$), the velocity and position vectors are randomly initialized and are denoted by

$$V_i(0) = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{1}$$

$$X_i(0) = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{2}$$

Step 2: For each particle i, evaluate the objective function $f(x)$ using the position vector X_i (0).

Step 3: Initialize the personal position vector of particle i, $G_{pers,i}$ (0) as follows:

$$G_{pers,i}(0) = X_i(0) \tag{3}$$

Step 4: Determine the global best position vector, G_{best} (0). It is the best position vector among

All personal positions vectors of the swarm. The G_{best} (0) is given by

$$G_{best}(0) = \left[g_{b,1}, g_{b,2}, \ldots, g_{b,d}\right] \tag{4}$$

Update: Iteration, $t = 1, 2, \ldots, N_{iter}$, the total number of iterations $= N_{iter}$.

Step 5: In iteration t, the particle's velocity and position vectors are updated as follows:

$$\begin{aligned} Vi(t) = Vi(t-1) + c_1 r_1(t)\big(G_{pers,i}(t-1) - Xi(t-1)\big) \\ + c_2 r_2(t)(G_{best}(t-1) - Xi(t-1)) \end{aligned} \tag{5}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{6}$$

where c_1 and c_2 are two positive coefficients, called acceleration constants, which are commonly set to 2.0 as default values [7]. The $r_1(t)$ and $r_2(t)$ are two randomly generated values with uniform distribution in the range of [0,1] [51].

Step 6: For each particle i, the $f(x)$ is evaluated using the position vector X_i (t).

Step 7: The $G_{pers,i}$ and G_{best} are updated as follows:

$$G_{pers,i}(t) = \begin{cases} X_i(t) & \mathit{if}\ f(X_i(t)) \le f\big(G_{pers,i}(t-1)\big) \\ G_{pers,i}(t-1) & \mathit{Otherwise} \end{cases} \tag{7}$$

Evaluate f ($G_{pers,i}$ (t)), $i = 1, 2, \ldots, m.$

Select $G_{best}(t)$ corresponding to minimum {f ($G_{pers,i}$ (t))}

Evaluate $f(x)$ to determine the global best position, $G_{best}(t)$

$$G_{best}(t) = \min\{G_{pers,i}(t)\} \tag{8}$$

Step 8: End of iterations, $t = N_{iter}$.

The global best position vector G_{best} (N_{iter}) becomes the global optimal solution and the f (G_{best} (N_{iter})) gives the optimal value of the objective function.

A flowchart of the GPSO algorithm is shown in Fig. 1.

3. The OPSO algorithm

Here, the details of the proposed OPSO algorithm and explanation of the OD process are provided.

3.1. Orthogonal diagonalization process

The OPSO algorithm is based on OD process. This process basically converts multiplication of three matrices to obtain a diagonal matrix, D, which is used in updating of the velocity and position vectors of the swarm particles. The updating is carried out in such way that the ith velocity and position vectors are affected by only the diagonal element, d_{ii} of matrix D. This process enhances the convergence and provides a better solution as shown in Observation 4 below.

The matrix diagonalization is the process of converting a square matrix, B of size ($d\times d$), into a diagonal matrix, D of size ($d\times d$), as shown below [52].

$$B = QDQ^{-1} \tag{9}$$

where Q is a matrix of size ($d\times d$) composed of eigenvectors of B and the diagonal elements of D contains the corresponding eigenvalues. The Q is an invertible matrix because it contains linearly independent vectors. When B is symmetric, the (9) may be written as

$$B = CDC^{-1} \tag{10}$$

in which the columns of matrix C are orthonormal to each other. The (10) can be rewritten as

$$D = C^{-1}BC \tag{11}$$

Since matrix C is an orthonormal matrix, the (11) can be written as

$$D \ = \ C^{\mathrm{T}}BC \tag{12}$$

Equation (12) is called the OD process. The process of OD is shown in Fig. 2.

3.2. OPSO learning algorithm

In this paper, the OPSO) algorithm is proposed to improve the learning strategy of the GPSO algorithm. The objective of the OPSO algorithm is to minimize the given d-dimensional objective function $f(x)$. The OPSO algorithm provides a new topology in the swarm population. Consider a swarm population with m particles, each with a dimension of d ($m > d$). In each iteration, the m particles are divided into two groups based on OD process as follows: an active group that consists of best personal experiences of d particles and one passive group which consists of the personal experiences of remaining ($m - d$) particles. The opinions of the active group particles are honoured by updating their respective velocity and position vectors. Whereas, the opinions of the passive group particles are ignored because their guidance may be insignificant or erratic, and therefore, their velocity and position vectors are not updated. However, the contributions of all the m particles in both groups are considered while determining the best experience. In each iteration, the OD process (12) is applied to obtain the matrix B from d best particles of the active group and thereafter, orthonormal matrix C and diagonal matrix D are computed. The steps involved in OPSO algorithm are given below. Let $f(x)$ be the objective function to be optimized and N_{iter} be the number of iterations.

Initialization: Iteration, $t = 0$.

Step 1: For each particle i, ($i = 1, 2, \ldots, m$), randomly initialize the velocity $V_i(0)$ and position X_i (0) vectors.
Step 2: Evaluate the objective function $f(x)$ using position vector X_i (0).
Step 3: Determine the personal position vectors, $G_{pers,i}$ (0) using (3).

Update, Update: Iteration, t = 1, 2, …, Niter.

Step 4: Arrange the m personal position vectors in an ascending order based on their $f(x)$ values. The corresponding top d particles constitute the active group particles.
Step 5: Construct matrix A of size ($m \times d$) such that each row occupies one of the m personal position vectors in the same ordered sequence as in step 4.
Step 6: Using pseudocode given in Fig. 3, convert matrix A to a symmetric matrix B of size ($d \times d$), such that B is a real symmetric matrix of dimension ($d \times d$).
Step 7: Apply the OD process shown in Fig. 2 on matrix B to obtain a diagonal matrix D of size $d \times d$. Let D_i denote the ith row of matrix D, where $i = 1, 2, \ldots, d$.
Step 8: Update the position and velocity vectors of the d particles of the active group, $i = 1, 2, \ldots, d$, as follows.

$$V_i(t) = V_i(t-1) + cr(t)[D_i(t) - X_i(t-1)] \tag{13}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{14}$$

where c is an acceleration coefficient and is chosen by trial and error method in the range [2,2.5] and $r(t)$ is a random value within the range [0,1].

Step 9: Determine the $G_{pers,i}(t)$ from the m particles ($i = 1, 2, \ldots, m$), as follows.

$$G_{pers,i}(t) = \begin{cases} X_i(t) & \mathit{if}\ f(X_i(t)) \le f(G_i(t-1)) \\ G_{pers,i}(t-1) & \mathit{Otherwise} \end{cases} \tag{15}$$

Evaluate f ($G_{pers,i}$ (t)), $i = 1, 2, \ldots, m$.

Step 10: Determine the global best position $G_{best}(t)$, as follows.

Select $G_{best}(t)$ corresponding to minimum {f ($G_{pers,i}$ (t))}, $i = 1, 2, \ldots, m$.

Evaluate $f(x)$ to determine the global best position, $G_{best}(t)$

$$G_{best}(t) = \min\{G_{pers,i}(t)\} \tag{16}$$

Step 11: End of iterations, t = N_{iter}.

The G_{best} (N_{iter}) as computed in step 10 provides the optimal solution. A flowchart of the OPSO algorithm is shown in Fig. 4.

Observation 1. One of the important observations of the OPSO algorithm is as follows. Sine matrix D is a diagonal matrix, its d rows or d columns are orthogonal vectors. These d vectors are used to diminish the contribution of $X_i(t–1)$ while updating $V_i(t)$, for $i = 1, 2, \ldots, d$. As $t\to\infty$, assume that the algorithm has converged. In such case, (14) can be written as:

$$\lim_{t\to\infty} X_i(t) = X_i(t-1) \tag{17}$$

This implies that $\lim\limits_{t\to\infty} V_i(t) = 0$. Therefore, (13) can be written as:

$$\lim_{t\to\infty} V_i(t) = V_i(t-1) = 0 \tag{18}$$

This implies that

$$\lim_{t\to\infty} cr(t)[D_i(t) - \ X_i(t-1)] = 0 \tag{19}$$

Since c $r(t)$ is constant,

$$\lim_{t\to\infty} X_i(t-1) = D_i(t) \tag{20}$$

From (20) it is evident that $\lim\limits_{t\to\infty} X_i(t)$ becomes diagonal and equals to D_i when iteration becomes large and the algorithm has converged.

Considering the d position vectors, (20) can be written in matrix form as:

$$[X(t)]_{active\_group} = [D(t)]_{active\_group} \tag{21}$$

This means that at $t \rightarrow \infty$, the d position vectors of the active group particles are equal to the d orthogonal vectors of the matrix D. Thus, the OPSO algorithm reaches convergence and leads to optimal solution.

Observation 2. In case of GPSO algorithm (5), two guides, $G_{pers,i}$ and G_{best}, are used to update the velocity vector $V_i(t)$. These two guides may conflict each other which leads to zigzag behaviour of the algorithm, that in turn causes trapping into local minima. Whereas, in the OPSO algorithm, since only one guide, $D_i(t)$ is used while updating of the velocity vector (13), the zigzagging behaviour is eliminated.

Observation 3. Due to orthogonalization of the position vectors, as iteration progresses, once the optimal solution is achieved, the solution remains the same in the subsequent iterations until the end of total number of iterations. This fact provides consistency to the proposed algorithm.

Observation 4. From (13), the velocity vector of each particle of active group at ith iteration can be rewritten as follows.

$$\begin{array}{lllllll} V_1(t) & = & V_1(t-1) & + & cr(t)[D_1(t) & - & X_1(t-1)] \\ V_2(t) & = & V_2(t-1) & + & cr(t)[D_2(t) & - & X_2(t-1)] \\ \vdots & & \vdots & & \vdots & & \vdots \\ V_d(t) & = & V_d(t-1) & + & cr(t)[D_d(t) & - & X_d(t-1)] \end{array} \tag{22}$$

$$\text{where } \begin{array}{llllll} D_1(t) = & [d_{11}, & 0, & 0, & \cdots, & 0] \\ D_2(t) = & [0, & d_{22}, & 0, & \cdots, & 0] \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ D_d(t) = & [0, & 0, & 0, & & d_{dd}] \end{array} \tag{23}$$

It can be seen from (22) and (23) that the position vector X_i, $i = 1, 2, \ldots, d$, of active group is affected only by the corresponding orthogonal vector D_i, $i = 1, 2, \ldots, d$. Thus, while updating, each V_i is perturbed only in the ith dimension of the d-dimensional search space. Due to this, the OPSO algorithm gives faster convergence and better solution.

Observation 5. As seen from the sensitivity analysis (Section 4.3), when $m \gg d$, the algorithm gives rise to more computations, but does not provides any better solution. Whereas, when $m = d$, there is no existence of passive group and therefore we do not see any advantages of diversity and the solution may not yield the best. Considering these two extremes, a reasonable value of m is about 10–30% more than d.

An Illustrative Example: In order to explain the mechanism of OPSO algorithm, Fig. 5 illustrates an example of a 2-dimensional shifted function, $f(x,y) = (x-2)^2 + (y+3)^2 + 9$. From visual inspection, it can be seen that the x and y are shifted from the origin (0,0) by (2.0,-3.0). The optimum solution of the given function equals to 9.0 at $(x,y) = (2.0,-3.0)$. The aim of the algorithm is to find the values x and y such that the $f(x,y)$ is minimized.

The OPSO algorithm program was implemented using MATLAB software in a personal computer with Intel (R) core (TM) 2 Duo CPU T6570 @ 2.1 GHz, 4 GB RAM and 64-bit Windows 7 operating system. The OPSO algorithm was executed with $m = 6$, $d = 2$ and $N_{iter} = 200$. The values of position vectors (X_i, $i = 1, 2, \ldots, 6$), the diagonal vectors (D_i, $i = 1, 2$) and personal vectors ($G_{pers,i}$, $i = 1, 2, \ldots, 6$) for different iterations are shown in Fig. 6. In each iteration, the six particles are divided into one active group of two best particles and a passive group of four particles. According to the OD process, $G_{pers,1}$ and $G_{pers,2}$ are assigned as active group and ($G_{pers,3}$, …, $G_{pers,6}$) are assigned as passive group. In each iteration, the velocity and position vectors of only the active group are updated. As seen from Fig. 6, as iteration increases, the OD process causes $[X]_{\text{active\_group}} = [D]_{\text{active\_group}}$, thus satisfying (21) and causing X to be a diagonal matrix. At the end of iteration, the best G_{pers}, provides the optimal solution, yielding $G_{best} = (2.0,-3.0)$.

In order to have geometric interpretation of the learning strategy of the OPSO algorithm, the movement of six position vectors and the two orthogonal vectors are shown in Fig. 7. Here, X_1 and X_2 represent the position vectors of the active group and D_1 and D_2 represent the two orthogonal vectors. It can be seen that during early iterations, the position vectors X_1 and X_2 move from random positions toward the orthogonal vectors D_1 and D_2. Finally, as the algorithm iterates further, the X_1 and X_2 coincide with D_1 and D_2.

In Fig. 8, the movement of personal vectors, $G_{pers,i}$ ($i = 1, 2, \ldots, 6$) of the six particles ($i = 1$ and 2 correspond to the active group) with increase in iteration is shown. Here, G_{best} corresponds to $G_{pers,1}$. It can be seen that the G_{best} moves from some random position to the optimal solution (2,-3) as the algorithm converges when iteration becomes large.

4. Experimental results and performance comparison

Here, we describe thirty benchmark functions and investigate performance of the OPSO and GPSO algorithms and a few competitive ECTs.

4.1. Benchmark functions

Thirty benchmark functions listed in Table 1 are used in this study. These benchmark functions are widely used in performance comparison of global optimization algorithms. All the thirty benchmark functions are minimization tasks and are divided into three groups based on their significant physical properties and shapes.

The first group involves nine unimodal benchmark functions $f_1 - f_9$ [39,40]. There is only one mode (global optimum) in its geometric distribution. The global optimum solution G_{best} is at the center of the search space. Therefore, the convergence rate of the search algorithm is important in finding global optimum. The nine unimodal benchmark functions are f_1 (Sphere), f_2 (Elliptic), f_3 (Sum Squares), f_4 (Sum Powers), f_5 (Schwefel's P.2.22), f_6 (Step a non-continuous), f_7 (Quadric), f_8 (Noise), and f_9 (Rotated Hyper Ellipsoid).

The second group includes eleven multimodal benchmark functions $f_{10} - f_{20}$. Finding G_{best} is more challenging since these are more difficult to optimize because of the number of local minima. In multimodal functions, the number of local minima increases as the problem dimension increases [6], [39]. Therefore, the search algorithm should be able to obtain a good solution and not be trapped in a local minimum. The eleven multimodal functions are f_{10} (Rosenbrock), f_{11} (Griewank), f_{12} (Levy), f_{13} (Himmelblau), f_{14} (Apline), f_{15} (Weierstrass), f_{16} Non-continuous Rastrigin (NCRastrigin), f_{17} (Michalewics), f_{18} (Schwefel), f_{19} (Rastrigin), and f_{20} (Ackley). The function f_{10} is unimodal in 2- or 3-dimensional search space. However, it may have local minima under high-dimensional cases (30-dimension) [6] [14], [44].

The optimum solution, x is at the origin of the search domain in the first and second groups, except f_{10}. Some algorithms simply converge to the center of the search domain that happens to be the optimum (x). Hence, these benchmark functions are not enough to test effectiveness of an optimization algorithm. Therefore, to avoid this drawback, a third group with ten benchmark functions [53–56] are used and are shown in Table 1.

The third group includes ten shifted, rotated and shifted rotated functions. In the shifted functions, the global optimum solution x as shown in Table 1 is not lying at the center of the search domain. The optimum solution x is shifted to a new position vector [53–55], i.e., shifted global optimum, $O = [o_1, o_2, \ldots, o_d]$, where d is dimension of the benchmark function. In the rotated functions, the rotation does not affect the shape of the function but increases the function complexity in finding global optimum. An orthogonal (rotation) matrix M is applied to obtain the rotation [56]. The matrix M is generated from a standard normally distributed entries using Gram-Schmidt orthogonalization process. In the shifted rotated functions, in addition to the shift in function's coordinates, the optimum x is rotated based on M. The third group is taken from the CEC 2005 [53,54], CEC 2008 [55] and CEC 2013 [56] special session on real parameter optimization. The ten benchmark functions are: f_{21} (Shifted Sphere), f_{22} (Shifted Schwefel's Problem 1.2), f_{23} (Shifted

Table 2
Values of m and Niter used in GPSO and OPSO algorithms.

Algorithm	f_1		f_2		f_3		f_4		f_5		f_6		f_7		f_8		f_9		f_{10}	
	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}
GPSO	20	10,000	25	8000	25	8000	20	10,000	20	10,000	20	10,000	16	12,500	8	25,000	20	10,000	20	10,000
OPSO	32	6250	32	6250	32	6250	32	6250	32	6250	32	6250	32	6250	32	6250	32	6250	32	6250

Algorithm	f_{11}		f_{12}		f_{13}		f_{14}		f_{15}		f_{16}		f_{17}		f_{18}		f_{19}		f_{20}	
	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}
GPSO	10	20,000	25	8000	25	8000	16	12,500	16	12,500	20	10,000	25	8000	32	6250	25	8000	25	8000
OPSO	32	6250	32	6250	110	1818	32	6250	32	6250	32	6250	110	1818	32	2000	32	6250	32	6250

Algorithm	f_{21}		f_{22}		f_{23}		f_{24}		f_{25}		f_{26}		f_{27}		f_{28}		f_{29}		f_{30}	
	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}	m	N_{iter}
GPSO	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000
OPSO	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000	40	5000

Rosenbrock), f_{24} (Shifted Rastrigin), f_{25} (Shifted Ackley), f_{26} (Shifted Griewank), f_{27} (Rotated Rosenbrock), f_{28} (Shifted rotated high coordinated Elliptic), f_{29} (Shifted rotated Rastrigin), and f_{30} (Shifted rotated Griewank).

The range and dimension of the search space of each benchmark function is given in column 4 of Table 1. All the functions are tested with 30-dimension, except f_{13} and f_{17} which are tested with 100-dimension. The "optimum $x = G_{best}$" is available in column 5, and minimum value of each function, "minimum $f(x) = f(G_{best})$" is given in column 6. The column 7 is "Accept Value", i.e., the accepted value of each function $f(x)$ under test. If the optimized value found by OPSO or GPSO algorithm falls between the "Accept Value" and "minimum $f(x)$", the solution of that function is judged to be successful, in other words, the algorithm passes the test. The symbol "NA" given in Table 1 denotes that the results are not available in the corresponding reference.

4.2. Performance measures and experimental setup

In order to evaluate performance of an algorithm in terms of accuracy, consistency and reliability, several performance measures are defined. Let m be the number of particles in the swarm, and d be the dimension of the search space. An algorithm is executed N_{iter} iterations over N_{run} runs.

1. Number of Function Evaluations (NFE): The NFE is used as a measure of computational complexity of an algorithm. The NFE is the number of times the objective function $f(x)$ is evaluated in one run of the algorithm and is given by

$$\text{NFE} = m \times Niter \tag{24}$$

2. Best Fitness Value (BFV): The BFV is defined as the minimum optimized $f(x)$ value obtained from N_{run} independent runs.
3. Worst Fitness Value (WFV): The WFV is defined as the maximum optimized $f(x)$ value obtained from N_{run} independent runs.
4. Mean Fitness Value (MFV): The MFV is defined as the average of the N_{run} BFVs.
5. Standard deviation (σ): The σ is the standard deviation of the N_{run} BFVs.
6. Success Rate (SR): At the end of one run, an algorithm is successful if the obtained optimized $f(x)$ value falls between the "Accept Value" and "Minimum $f(x)$". The SR is used as a measure of reliability of an algorithm [6] [39],- [40]. The SR in percentage is given by

$$\text{SR} = \frac{\text{Number of successful runs}}{N_{run}} \times 100 \tag{25}$$

7. Reliability Rate (RR): The RR of an algorithm over all the thirty benchmark functions is defined as

$$\text{RR} = \frac{1}{30}\sum_{i=1}^{30} \text{SR}_i \tag{26}$$

where SR_i is the success rate of the benchmark function $f_i(x)$, $i = 1, 2, \ldots, 30$.

8. Average execution time (AET): It is the execution time of an algorithm until it reaches to "Accept Value", averaged over N_{run} independent runs.

In order to measure the accuracy, consistency and robustness

Table 3
Sensitivity analysis of the OPSO algorithm with variation of swarm size.

f	Min. $f(x)$	Fitness	$m = 30$	AET (sec)	$m = 32$	AET (sec)	$m = 34$	AET (sec)	$m = 36$	AET (sec)	$m = 38$	AET (sec)	$m = 40$	AET (sec)
f_2	0	BFV	0.0	24.1	0.0	26.4	0.0	28.3	0.0	30.1	0.0	32.8	0.0	33.4
		WFV	2.45×10^{-27}		0.0		0.0		0.0		0.0		0.0	
		MFV	1.98×10^{-27}		0.0		0.0		0.0		0.0		0.0	
f_3	0	BFV	0.0	21.1	0.0	22.4	0.0	25.2	0.0	25.5	0.0	27.1	0.0	27.8
		WFV	4.67×10^{-34}		0.0		0.0		0.0		0.0		0.0	
		MFV	3.56×10^{-34}		0.0		0.0		0.0		0.0		0.0	
f_{12}	0	BFV	1.82×10^{-19}	31.4	1.76×10^{-32}	37.4	1.49×10^{-32}	39.3	1.49×10^{-32}	42.9	1.49×10^{-32}	45.3	1.49×10^{-32}	46.8
		WFV	2.49×10^{-19}		3.57×10^{-32}		1.49×10^{-32}		1.49×10^{-32}		1.49×10^{-32}		1.49×10^{-32}	
		MFV	1.97×10^{-19}		2.06×10^{-32}		1.49×10^{-32}		1.49×10^{-32}		1.49×10^{-32}		1.49×10^{-32}	
f_{20}	0	BFV	3.23×10^{-174}	22.5	2.09×10^{-278}	25.0	0.0	26.8	0.0	27.9	0.0	28.3	0.0	29.8
		WFV	3.10×10^{-15}		1.13×10^{-277}		0.0		0.0		0.0		0.0	
		MFV	1.04×10^{-15}		1.78×10^{-278}		0.0		0.0		0.0		0.0	
f_{21}	−450	BFV	−450.00	18.3	−450.00	19.1	−450.00	20.1	−450.00	21.1	−450.00	21.5	−450.00	22.5
		WFV	−449.98		−450.00		−450.00		−450.00		−450.00		−450.00	
		MFV	−449.99		−450.00		−450.00		−450.00		−450.00		−450.00	
f_{23}	390	BFV	390.94	49.8	390.10	53.0	390.00	57.4	390.00	59.4	390.00	63.3	390.00	66.3
		WFV	393.99		390.87		390.18		390.18		390.18		390.18	
		MFV	392.04		390.27		390.09		390.09		390.09		390.09	
f_{27}	−900	BFV	−899.99	18.9	−900.00	20.1	−900.00	21.5	−900.00	22.6	−900.00	24.1	−900.00	25.1
		WFV	−899.54		−899.99		−899.99		−899.99		−899.99		−899.99	
		MFV	−898.23		−899.99		−899.99		−899.99		−899.99		−899.99	
f_{29}	−330	BFV	−330.93	28.2	−330.10	29.4	−330.00	32.0	−330.00	34.3	−330.00	36.3	−330.00	37.2
		WFV	−322.34		−331.92		−330.75		−330.75		−330.75		−330.75	
		MFV	328.67		−330.57		−330.26		−330.26		−330.26		−330.26	
f_{30}	−180	BFV	−180.67	27.2	−180.00	29.1	−180.00	30.5	−180.00	32.4	−180.00	34.8	−180.00	36.3
		WFV	−181.45		−180.00		−180.00		−180.00		−180.00		−180.00	
		MFV	180.5		−180.00		−180.00		−180.00		−180.00		−180.00	

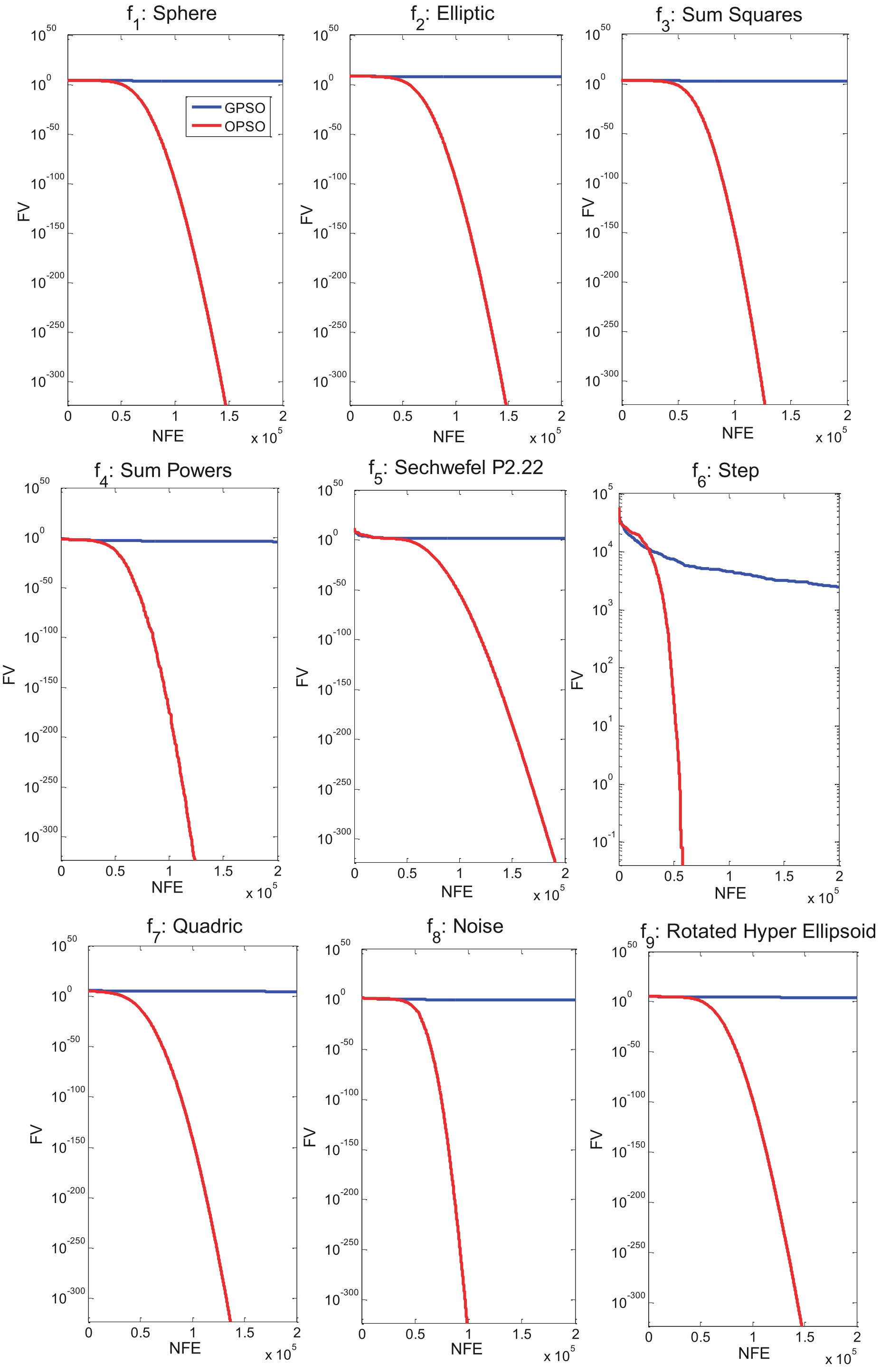


Fig. 9. Comparison of convergence characteristics between OPSO and GPSO algorithms for $f_1 - f_9$.

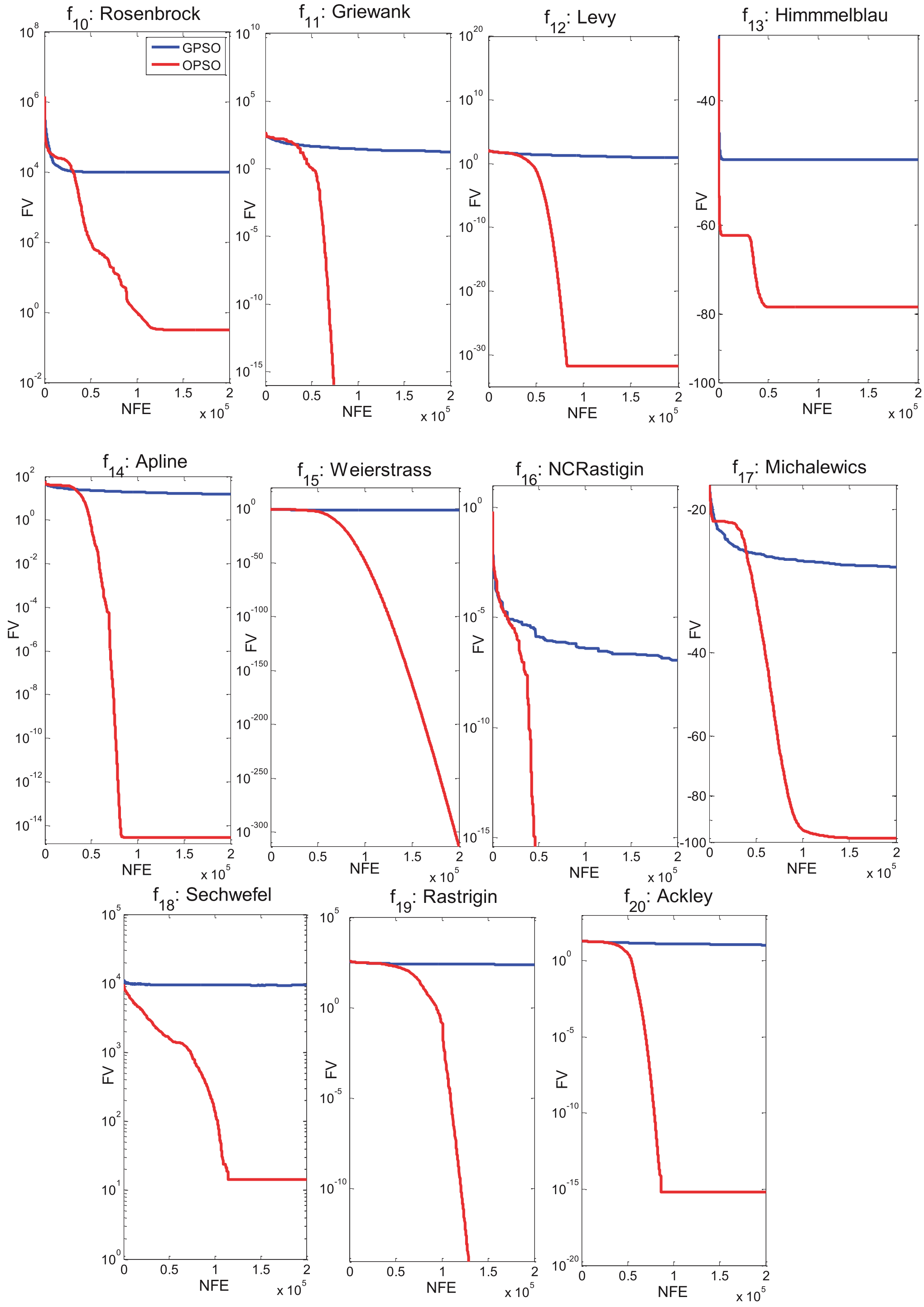


Fig. 10. Comparison of convergence characteristics between OPSO and GPSO algorithms for $f_{10} - f_{20}$.

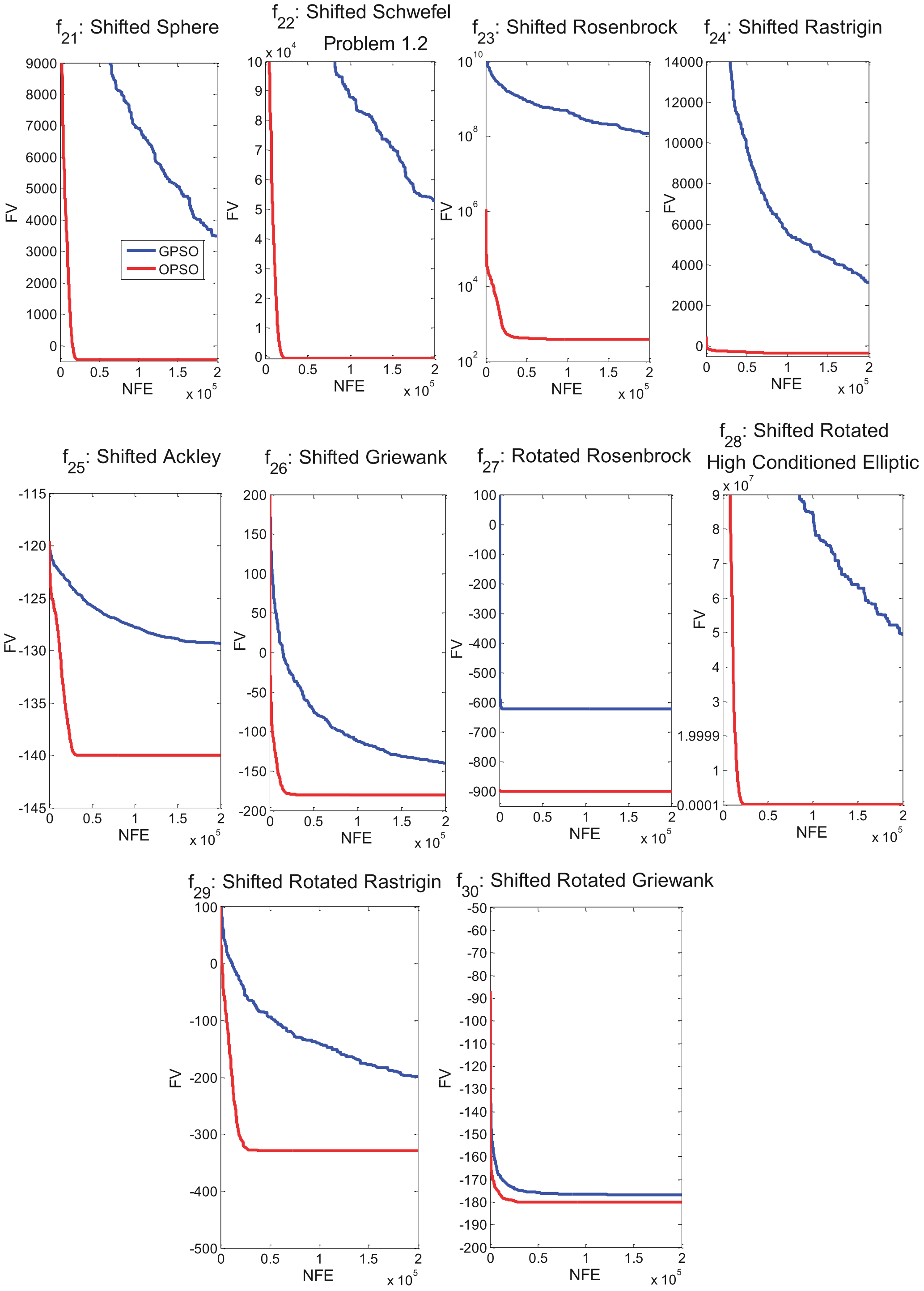


Fig. 11. Comparison of convergence characteristics between OPSO and GPSO algorithms for $f_{21} - f_{30}$.

Table 4
Performance comparison between GPSO and OPSO algorithm on thirty benchmark functions.

f	Minimum $f(x)$	Accept Value	Fitness	GPSO	OPSO
f_1	0.0	1×10^{-06}	BFV	1.2486×10^{03}	**0.0**
			WFV	4.1798×10^{03}	**0.0**
			MFV	2.4550×10^{03}	**0.0**
			σ	5.7333×10^{02}	**0.0**
			AET (sec)	–	**15.5355**
f_2	0.0	1×10^{-08}	BFV	2.0131×10^{07}	**0.0**
			WFV	6.9372×10^{07}	**0.0**
			MFV	3.5123×10^{07}	**0.0**
			σ	1.1286×10^{07}	**0.0**
			AET (sec)	–	**24.7634**
f_3	0.0	1×10^{-08}	BFV	1.8157×10^{02}	**0.0**
			WFV	6.8391×10^{02}	**0.0**
			MFV	4.0027×10^{02}	**0.0**
			σ	1.0706×10^{02}	**0.0**
			AET (sec)	–	**16.3259**
f_4	0.0	1×10^{-08}	BFV	2.3191×10^{-06}	**0.0**
			WFV	3.9232×10^{-04}	**0.0**
			MFV	8.6259×10^{-05}	**0.0**
			σ	1.0228×10^{-04}	**0.0**
			AET (sec)	–	**15.6650**
f_5	0.0	1×10^{-08}	BFV	1.8218×10^{01}	**0.0**
			WFV	3.3622×10^{01}	**9.8813×10^{-324}**
			MFV	2.5419×10^{01}	**0.0**
			σ	3.5500×10^{0}	**0.0**
			AET (sec)	–	**17.7892**
f_6	0.0	1×10^{-08}	BFV	2.4679×10^{03}	**0.0**
			WFV	3.6900×10^{03}	**0.0**
			MFV	2.4679×10^{03}	**0.0**
			σ	5.7707×10^{02}	**0.0**
			AET (sec)	–	**14.0659**
f_7	0.0	1×10^{-06}	BFV	1.9811×10^{04}	**0.0**
			WFV	5.2134×10^{04}	**0.0**
			MFV	3.2116×10^{04}	**0.0**
			σ	9.4586×10^{03}	**0.0**
			AET (sec)	–	**26.2586**
f_8	0.0	1×10^{-02}	BFV	1.0849×10^{-02}	**0.0**
			WFV	5.6201×10^{-01}	**0.0**
			MFV	1.2788×10^{-01}	**0.0**
			σ	1.2006×10^{-01}	**0.0**
			AET (sec)	–	**8.9479**
f_9	0.0	1×10^{-08}	BFV	7.8459×10^{03}	**0.0**
			WFV	2.2750×10^{04}	**0.0**
			MFV	1.5896×10^{04}	**0.0**
			σ	3.8854×10^{03}	**0.0**
			AET (sec)	–	**25.8315**
f_{10}	0.0	1×10^{02}	BFV	6.8740×10^{03}	**5.7699×10^{-05}**
			WFV	1.1945×10^{04}	**5.1926×10^{0}**
			MFV	9.8909×10^{03}	**3.1611×10^{-01}**
			σ	1.3836×10^{03}	**1.0618×10^{0}**
			AET (sec)	–	**11.4709**
f_{11}	0.0	1×10^{-08}	BFV	7.2437×10^{0}	**0.0**
			WFV	3.0307×10^{01}	**0.0**
			MFV	1.7175×10^{01}	**0.0**
			σ	6.3452×10^{0}	**0.0**
			AET (sec)	–	**21.8874**
f_{12}	0.0	1×10^{-08}	BFV	4.7802×10^{0}	**1.4998×10^{-32}**
			WFV	1.3640×10^{1}	**1.4998×10^{-32}**
			MFV	8.3570×10^{0}	**1.4998×10^{-32}**
			σ	2.8145×10^{0}	**8.3800×10^{-48}**
			AET (sec)	–	**44.7980**
f_{13}	-7.833223×10^{01}	-7.8×10^{01}	BFV	-5.4403×10^{01}	**-7.83323×10^{01}**
			WFV	-4.3101×10^{01}	**-7.83323×10^{01}**
			MFV	-4.8420×10^{01}	**-7.83323×10^{01}**
			σ	3.2065×10^{0}	**1.3605×10^{-14}**
			AET (sec)	–	**23.8133**
f_{14}	0.0	1×10^{-08}	BFV	1.3035×10^{01}	**0.0**
			WFV	1.8438×10^{01}	**1.3101×10^{-14}**
			MFV	1.5513×10^{01}	**2.8333×10^{-15}**
			σ	1.5921×10^{0}	**3.7142×10^{-15}**
					21.4088
f_{15}	0.0	1×10^{-08}	BFV	4.1622×10^{-02}	**0.0**
			WFV	7.4146×10^{-02}	**3.0038×10^{-312}**
			MFV	5.4681×10^{-02}	**1.2027×10^{-313}**
			σ	9.5018×10^{-03}	**0.0**

(*continued on next page*)

Table 4 (*continued*)

f	Minimum *f*(*x*)	Accept Value	Fitness	GPSO	OPSO
			AET (sec)	–	**18.9627**
f_{16}	0.0	1×10^{-08}	BFV	8.9136×10^{-10}	**0.0**
			WFV	5.8023×10^{-07}	**0.0**
			MFV	1.1718×10^{-07}	**0.0**
			σ	1.4790×10^{-07}	**0.0**
			AET (sec)	–	**33.7595**
f_{17}	-9.92784×10^{01}	-9.5×10^{01}	BFV	-2.9595×10^{01}	**-9.85149×10^{01}**
			WFV	-2.4683×10^{01}	**-9.72067×10^{01}**
			MFV	-2.6475×10^{01}	**-9.80995×10^{01}**
			σ	1.2014×10^{-01}	**3.8892×10^{-01}**
			AET (sec)	–	**41.2021**
f_{18}	0.0	2×10^{03}	BFV	9.4000×10^{03}	**3.8183×10^{-04}**
			WFV	1.0081×10^{04}	**1.1844×10^{02}**
			MFV	9.5014×10^{03}	**1.4213×10^{01}**
			σ	3.4563×10^{09}	**3.9282×10^{01}**
			AET (sec)	–	**32.5160**
f_{19}	0.0	1×10^{-08}	BFV	2.0729×10^{02}	**0.0**
			WFV	2.6763×10^{02}	**0.0**
			MFV	2.4074×10^{02}	**0.0**
			σ	1.3470×10^{0}	**0.0**
			AET (sec)	–	**30.6209**
f_{20}	0.0	1×10^{-08}	BFV	8.4121×10^{0}	**4.4409×10^{-15}**
			WFV	1.4451×10^{01}	**7.9936×10^{-15}**
			MFV	1.0876×10^{01}	**6.5725×10^{-15}**
			σ	1.4402×10^{0}	**1.7764×10^{-15}**
			AET (sec)	–	**28.3525**
f_{21}	–450	–350	BFV	1.3606×10^{03}	**-4.5000×10^{02}**
			WFV	6.3523×10^{03}	**-4.5000×10^{02}**
			MFV	3.4897×10^{03}	**-4.5000×10^{02}**
			σ	1.1566×10^{03}	**1.5825×10^{-13}**
			AET (sec)	–	**8.9907**
f_{22}	–450	–350	BFV	3.3409×10^{04}	**-4.5000×10^{02}**
			WFV	9.2368×10^{04}	**-4.5000×10^{02}**
			MFV	5.3040×10^{04}	**-4.5000×10^{02}**
			σ	1.5313×10^{04}	**1.4950×10^{-13}**
			AET (sec)	–	**17.2141**
f_{23}	390	490	BFV	2.8616×10^{07}	**3.9000×10^{02}**
			WFV	3.0831×10^{08}	**3.9018×10^{02}**
			MFV	1.2195×10^{08}	**3.9009×10^{02}**
			σ	7.3326×10^{07}	**1.1040×10^{0}**
			AET (sec)	–	**12.3678**
f_{24}	–330	−230	BFV	1.5114×10^{03}	$-3.3000\times$**10^{02}**
			WFV	5.6216×10^{03}	$-3.3000\times$**10^{02}**
			MFV	3.1415×10^{03}	$-3.3000\times$**10^{02}**
			σ	1.0315×10^{03}	$4.3415\times$**10^{-14}**
			AET (sec)	–	**28.1717**
f_{25}	–140	−135	BFV	-1.3118×10^{02}	**-1.4000×10^{02}**
			WFV	-1.2755×10^{02}	**-1.4000×10^{02}**
			MFV	-1.2935×10^{02}	**-1.4000×10^{02}**
			σ	8.7805×10^{-01}	**4.7488×10^{-14}**
			AET (sec)	–	**8.9032**
f_{26}	–180	−170	BFV	-1.5831×10^{02}	**-1.8000×10^{02}**
			WFV	-1.1322×10^{02}	**-1.7995×10^{02}**
			MFV	-1.3991×10^{02}	**-1.7999×10^{02}**
			σ	1.0342×10^{01}	**1.2766×10^{-02}**
			AET (sec)	–	**9.8174**
f_{27}	–900	−800	BFV	-7.3860×10^{02}	**-9.0000×10^{02}**
			WFV	-4.8226×10^{02}	**-9.0000×10^{02}**
			MFV	-6.2256×10^{02}	**-9.0000×10^{02}**
			σ	7.6227×10^{01}	**1.6450×10^{-04}**
			AET (sec)	–	**2.1427**
f_{28}	–450	–350	BFV	1.3391×10^{07}	**-4.5000×10^{02}**
			WFV	9.7058×10^{07}	**-4.5000×10^{02}**
			MFV	4.9441×10^{07}	**-4.5000×10^{02}**
			σ	1.9915×10^{07}	**4.7488×10^{-13}**
			AET (sec)	–	**9.9062**
f_{29}	−330	−300	BFV	-2.6583×10^{02}	**-3.3000×10^{02}**
			WFV	-1.0334×10^{02}	**-3.3075×10^{02}**
			MFV	-1.9788×10^{02}	**-3.3026×10^{02}**
			σ	3.8162×10^{01}	**2.1549×10^{00}**
			AET (sec)	–	**26.2184**
f_{30}	–180	−180	BFV	-1.7848×10^{02}	**-1.8000×10^{02}**
			WFV	-1.7555×10^{02}	**-1.8000×10^{02}**
			MFV	-1.7675×10^{02}	**-1.8000×10^{02}**
			σ	6.9078×10^{02}	**2.5945×10^{-14}**
			AET (sec)	–	**26.6115**

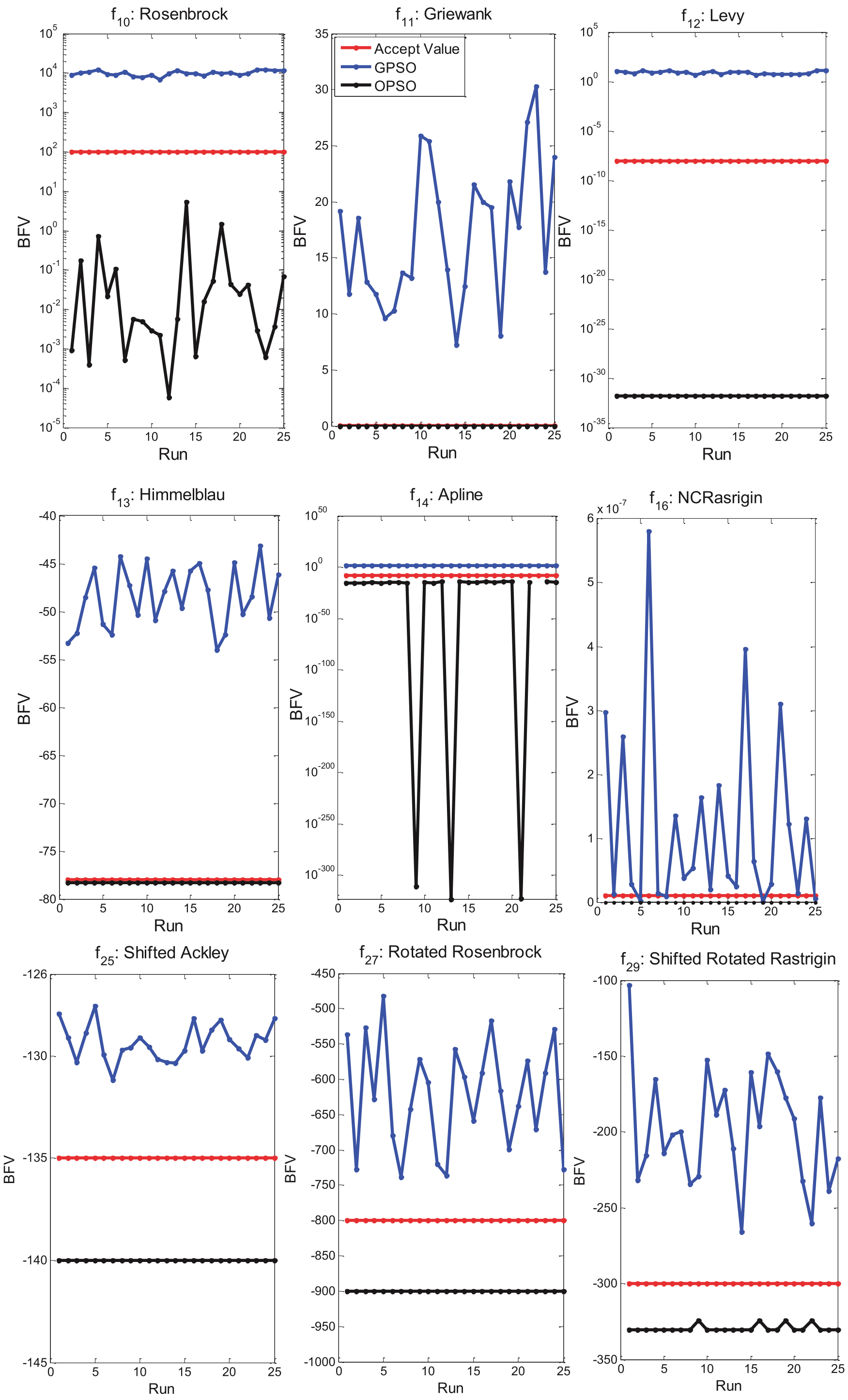


Fig. 12. The BFVs obtained at different runs by the OPSO and GPSO algorithms for $f_{10} - f_{14}$, f_{16}, f_{25}, f_{27} and f_{29}.

Table 5
Performance comparison between OPSO algorithm and 15 ECTs using 4 unimodal functions with $d = 30$.

ECTs	Performance measure	f_1	f_5	f_6	f_8
OPSO	MFV	**0.0**	**0.0**	**0.0**	**0.0**
(proposed)	σ	**0.0**	**0.0**	**0.0**	**0.0**
	NFE	6.00×10^{04}	6.60×10^{04}	5.83×10^{04}	4.04×10^{04}
	AET (sec)	15.5355	17.7892	14.0659	**8.9479**
FIPSO	MFV	2.42×10^{-13}	2.76×10^{-08}	**0.0**	4.24×10^{-03}
[4]	σ	1.73×10^{-13}	9.04×10^{-09}	**0.0**	1.28×10^{-04}
	NFE	1.18×10^{05}	1.65×10^{05}	–	9.10×10^{04}
CLPSO	MFV	4.46×10^{-14}	2.51×10^{-08}	**0.0**	5.85×10^{-03}
[13]	σ	1.73×10^{-14}	5.84×10^{0}	**0.0**	1.11×10^{-03}
	NFE	1.39×10^{05}	1.72×10^{05}	–	1.33×10^{05}
HPSO-TVAC	MFV	2.83×10^{-33}	9.03×10^{-20}	**0.0**	9.82×10^{-02}
[14]	σ	3.19×10^{-33}	9.58×10^{-20}	**0.0**	3.26×10^{-02}
	NFE	6.39×10^{04}	7.89×10^{04}	–	–
ALC-PSO	MFV	1.66×10^{-161}	1.61×10^{-90}	**0.0**	–
[15]	σ	8.20×10^{-161}	4.14×10^{-90}	**0.0**	–
	NFE	7.49×10^{04}	8.18×10^{04}	1.31×10^{04}	–
EPSO	MFV	1.662×10^{-74}	1.90×10^{-47}	**0.0**	2.58×10^{-04}
[16]	σ	2.76×10^{-74}	2.15×10^{-47}	**0.0**	1.87×10^{-04}
SAIW	MFV	3.83×10^{-147}	–	1.40×10^{0}	–
[20]	σ	7.85×10^{-147}	–	1.34×10^{0}	–
*Pk*PSO-G	MFV	1.27×10^{-34}	5.02×10^{-20}	**0.0**	5.13×10^{-02}
[22]	σ	2.61×10^{-34}	6.91×10^{-20}	**0.0**	2.84×10^{-02}
	NFE	1.40×10^{05}	–	–	–
MLPSO-STP	MFV	2.37×10^{-15}	7.94×10^{-84}	**0.0**	2.97×10^{-04}
[24]	σ	**0.0**	2.67×10^{-83}	**0.0**	9.68×10^{-05}
	NFE	9.11×10^{04}	9.51×10^{04}	7.51×10^{04}	7.94×10^{04}
CCPSO-ISM	MFV	3.03×10^{-52}	–	**0**	6.71×10^{-03}
[27]	σ	9.55×10^{-52}	–	**0**	1.71×10^{-03}
	NFE	3.502×10^{03}	–	2.023×10^{04}	133,985
	AET (sec)	**0.27**	–	**0.23**	**1.12**
FPSOCM	MFV	2.882×10^{-05}	1.92×10^{-06}	1.60×10^{-01}	4.35×10^{-03}
[38]	σ	3.108×10^{-05}	2.48×10^{-06}	4.22×10^{-01}	8.48×10^{-04}
	AET (sec)	**9.140**	**8.759**	**6.560**	9.344
OLPSO-G	MFV	4.12×10^{-54}	9.85×10^{-30}	**0.0**	1.16×10^{-02}
[39]	σ	6.34×10^{-54}	1.01×10^{-29}	**0.0**	4.10×10^{-03}
	NFE	8.92×10^{04}	1.01×10^{05}	–	1.50×10^{05}
OLPSO-L	MFV	1.11×10^{-38}	7.67×10^{-22}	**0.0**	1.64×10^{-02}
[39]	σ	1.28×10^{-38}	5.63×10^{-22}	**0.0**	3.25×10^{-03}
	NFE	9.83×10^{04}	1.14×10^{05}	–	1.86×10^{05}
OGABC	MFV	4.69×10^{-38}	5.33×10^{-20}	**0.0**	5.09×10^{-03}
[40]	σ	5.27×10^{-38}	3.13×10^{-20}	**0.0**	2.07×10^{-03}
	NFE	$\mathbf{2.91 \times 10^{04}}$	4.64×10^{04}	$\mathbf{1.02 \times 10^{04}}$	–
DEGL/SAW	MFV	8.74×10^{-37}	4.93×10^{-36}	9.56×10^{-48}	1.05×10^{-07}
[42]	σ	3.82×10^{-35}	3.92×10^{-34}	2.73×10^{-45}	2.33×10^{-06}
	NFE	3.88×10^{04}	$\mathbf{4.45 \times 10^{04}}$	4.63×10^{04}	$\mathbf{2.71 \times 10^{04}}$
MGBDE	MFV	8.79×10^{-68}	8.50×10^{-41}	**0.0**	2.14×10^{-03}
[43]	σ	5.21×10^{-69}	3.38×10^{-41}	**0.0**	1.08×10^{-03}
Minimum $f(x)$		0.0	0.0	0.0	0.0
Accept Value $f(x)$		1.00×10^{-06}	1.00×10^{-08}	1.00×10^{-08}	1.00×10^{-02}

of each algorithm, the OPSO and GPSO algorithms were evaluated using the thirty unimodal and multimodal functions given in Table 1. Both OPSO and GPSO algorithms are run with maximum NFE = 200,000. The acceleration coefficients values of c_1 and c_2 in GPSO and c in OPSO algorithm are set at 2.0 and 2.05, respectively, using trial and error method. The parameters $r(t)$, $r_1(t)$and $r_2(t)$ are chosen randomly. Since $m > d$ in the OPSO algorithm, the number of particles (m) in OPSO algorithm is different from GPSO algorithm. The shifted global optimum vector O is randomly distributed in the range $[-80,80]^d$ for all functions except f_{24} and f_{29} in which the range is $[-4,4]^d$. The orthogonal (rotation) matrix M is generated using Gram-Schmidt orthogonalization process. NFE is taken as 200,000. The value of N_{iter} is obtained from (24), once NFE and m are selected. The values of m and N_{iter} used in OPSO and GPSO algorithms are given in Table 2.

4.3. Sensitivity analysis of swarm size on OPSO algorithm

In order to study the sensitivity analysis of the proposed OPSO algorithm with variation of swarm size m, nine 30-dimensional benchmarks functions, f_2, f_3, f_{12}, f_{20}, f_{21}, f_{23}, f_{27}, f_{29} and f_{30} are tested with $N_{iter} = 5000$ and $N_{\text{run}} = 25$. In addition, the AET is obtained with $t = 5000$ over 25 independent runs. Table 3 shows sensitivity analysis of the OPSO algorithm with variation of swarm size m, in terms of BFV, WFV, MFV and AET. When the swarm population, $m = 32$, it has 30 particles in active group and 2 particles in the passive group. It can be seen that with $m = 32$, the performance of OPSO algorithm improves substantially compared to $m = 30$ (i.e., the number of particles in the passive group equals to zero). Therefore, $m = 32$ has been selected for the set of 9 unimodal (f_1-f_9) and 11 multimodal (f_{10}-f_{20}) benchmark functions, as shown in Table 2. However, in case of shifted, rotated and shifted rotated benchmark functions, the performance of OPSO algorithm substantially improves as m increases from 32 to 40. Therefore, as shown in Table 2, for the ten such functions (f_{21}-f_{30}), $m = 40$ has been selected which makes the number of passive group particles to be 10. Based on these observations, as a thumb rule, are may select the swarm population size between 10 and 30% more than the dimension of the search space.

Table 6
Performance comparison between OPSO algorithm and other 16 ECTs on 7 multimodal functions with $d = 30$.

ECTs	Performance measure	f_{10}	f_{11}	f_{15}	f_{16}	f_{18}	f_{19}	f_{20}
OPSO	MFV	3.16×10^{-01}	**0.0**	**1.20×10^{-313}**	**0.0**	1.42×10^{01}	**0.0**	6.57×10^{-15}
(proposed)	σ	1.06×10^{00}	**0.0**	**0.0**	**0.0**	3.92×10^{01}	**0.0**	1.77×10^{-15}
	NFE	5.03×10^{04}	6.70×10^{04}	6.54×10^{04}	**3.81×10^{04}**	**3.97×10^{04}**	1.16×10^{05}	7.57×10^{04}
	AET (sec)	11.4709	21.8874	18.9627	33.7595	32.5160	30.6209	28.3525
FIPSO	MFV	2.51×10^{01}	9.01×10^{-12}	–	7.01×10^{01}	9.93×10^{02}	6.51×10^{01}	2.33×10^{-07}
[4]	σ	5.10×10^{-01}	1.84×10^{-11}	–	1.47×10^{01}	5.09×10^{02}	1.33×10^{01}	7.19×10^{-08}
	NFE	4.84×10^{04}	1.33×10^{05}	–	–	1.33×10^{05}	7.94×10^{04}	1.83×10^{05}
CLPSO	MFV	2.10×10^{01}	3.14×10^{-01}	3.45×10^{-07}	4.36×10^{-1}	1.27×10^{-1}	4.85×10^{-1}	**0.0**
[13]	σ	2.98×10^{0}	4.64×10^{-01}	1.94×10^{-07}	2.44×10^{-1}	8.79×10^{-1}	3.63×10^{-1}	**0.0**
	NFE	1.08×10^{05}	1.67×10^{05}	–	–	6.54×10^{04}	4.40×10^{04}	1.90×10^{05}
HPSO-TVAC	MFV	2.39×10^{01}	9.75×10^{-03}	–	1.03×10^{01}	1.59×10^{03}	9.43×10^{0}	7.29×10^{-14}
[14]	σ	2.65×10^{01}	8.33×10^{-03}	–	8.24×10^{0}	3.26×10^{02}	3.48×10^{0}	3.00×10^{-14}
	NFE	5.06×10^{04}	6.69×10^{04}	–	–	5.66×10^{04}	**6.09×10^{03}**	1.02×10^{05}
ALC-PSO	MFV	7.61×10^{0}	1.22×10^{-02}	–	1.25×10^{-11}	2.10×10^{01}	2.52×10^{-14}	1.14×10^{-14}
[15]	σ	6.65×10^{0}	**0.0**	–	6.75×10^{-11}	5.41×10^{01}	1.37×10^{-14}	2.94×10^{-15}
	NFE	6.04×10^{04}	**1.01×10^{04}**	–	5.89×10^{04}	4.66×10^{04}	7.42×10^{04}	**5.89×10^{04}**
MCPSO	MFV	6.12×10^{0}	1.91×10^{-14}	–	–	1.32×10^{-03}	7.08×10^{-06}	6.38×10^{-12}
[16]	σ	1.09×10^{01}	3.26×10^{-14}	–	–	2.16×10^{-03}	3.46×10^{-05}	5.09×10^{-11}
SAIW	MFV	9.88×10^{0}	22.0×10^{-04}	–	31.8×10^{0}	-6.37×10^{03}	29.84×10^{0}	22.71×10^{0}
[21]	σ	19.96×10^{0}	47.0×10^{-04}	–	10.30×10^{0}	0.74×10^{03}	10.54×10^{0}	3.74×10^{-15}
*pk*PSO-G	MFV	8.14×10^{01}	1.19×10^{-01}	–	1.65×10^{01}	-1.10×10^{04}	**0.0**	4.09×10^{-14}
[22]	σ	4.82×10^{01}	1.37×10^{-01}	–	2.46×10^{01}	138.96	**0.0**	9.23×10^{-14}
	NFE	1.40×10^{05}	–	–		–	1.58×10^{05}	1.45×10^{05}
MLPSO-STP	MFV	2.52×10^{01}	5.75×10^{-04}	**0.0**	9.60×10^{-01}	–	**0.0**	4.03×10^{-15}
[24]	σ	1.84×10^{-01}	2.21×10^{-03}	**0.0**	3.14×10^{0}	–	**0.0**	1.23×10^{-15}
	NFE	39.0×10^{04}	9.30×10^{04}	8.80×10^{04}	8.90×10^{04}	–	7.50×10^{04}	8.38×10^{04}
CCPSO-ISM	MFV	**7×10^{-02}**	6.84×10^{-14}	–	–	–	**0**	1.40×10^{-14}
[27]	σ	19×10^{-02}	1.69×10^{-13}	–	–	–	**0**	1.60×10^{-14}
	NFE	**3.22×10^{04}**	4.06×10^{04}	–	–	–	1.78×10^{04}	4.89×10^{04}
	AET (sec)	**0.27**	**0.49**	–	–	–	**0.22**	**0.56**
MsPSO	MFV	2.89×10^{01}	**0.0**	–	–	8.63×10^{-53}	**0.0**	8.88×10^{-16}
[36]	σ	1.44×10^{-02}	**0.0**	–	–	2.65×10^{-52}	**0.0**	2.01×10^{-32}
Gaussian PSO	MFV	4.25×10^{0}	7.42×10^{-04}	5.41×10^{-03}	5.00×10^{0}	3.91×10^{02}	1.74×10^{00}	4.18×10^{-05}
[37]	σ	7.60×10^{-01}	1.50×10^{-03}	6.3×10^{-03}	4.1×10^{-04}	8.9×10^{-06}	2.00×10^{-02}	2.50×10^{-05}
	NFE	1.26×10^{05}	1.37×10^{05}	–	–	1.11×10^{05}	7.52×10^{04}	1.52×10^{05}
FPSOCM	MFV	6.09×10^{-01}	2.23×10^{-04}	–	–	–	8.26×10^{00}	9.01×10^{-04}
[38]	σ	1.85×10^{-01}	2.84×10^{-05}	–	–	–	2.78×10^{00}	6.95×10^{-04}
	AET (sec)	8.070	7.875	–	–	–	7.641	13.703
OLPSO-G	MFV	2.15×10^{01}	4.83×10^{-03}	–	–	3.84×10^{02}	1.07×10^{0}	2.15×10^{01}
[39]	σ	2.99×10^{01}	8.63×10^{-03}	–	–	2.17×10^{02}	9.92×10^{-1}	2.99×10^{01}
	NFE	7.87×10^{04}	9.33×10^{04}	–	–	4.05×10^{04}	3.77×10^{04}	7.87×10^{04}
OLPSO-L	MFV	1.26×10^{01}	**0.0**	–	–	3.82×10^{-04}	**0.0**	4.14×10^{-15}
[39]	σ	1.40×10^{0}	**0.0**	–	–	**0.0**	**0.0**	**0.0**
	NFE	9.22×10^{04}	10.72×10^{04}	–	–	5.14×10^{04}	4.36×10^{04}	12.65×10^{04}
OGABC	MFV	1.10×10^{0}	**0.0**	**0.0**	1.18×10^{-09}	**0.0**	**0.0**	1.93×10^{-14}
[40]	σ	1.74×10^{0}	**0.0**	**0.0**	5.30×10^{-09}	**0.0**	**0.0**	4.07×10^{-15}
	NFE	3.61×10^{04}	3.50×10^{04}	**5.44×10^{04}**	4.10×10^{04}	4.30×10^{04}	3.94×10^{04}	5.89×10^{04}
MGBDE	MFV	1.69×10^{0}	**0.0**	–	–	-1.30×10^{-02}	3.98×10^{0}	7.69×10^{-15}
[43]	σ	2.24×10^{0}	**0.0**	–	–	1.09×10^{-12}	2.98×10^{0}	**0.0**
	NFE	3.92×10^{04}	3.64×10^{04}	–	–	6.80×10^{04}	2.78×10^{04}	7.46×10^{04}
Minimum *f*(*x*)		0.0	0.0	0.0	0.0	0.0	0.0	0.0
Accept Value *f*(*x*)		1.00×10^{02}	1.00×10^{-08}	1.00×10^{-08}	1.00×10^{-08}	2.00×10^{03}	1.00×10^{-08}	1.00×10^{-08}

4.4. Convergence characteristics

Fig. 9 shows the convergence characteristics of OPSO and GPSO algorithms for the nine unimodal functions f_1 – f_9. The comparison is obtained in terms of fitness value (FV) averaged over N_{run} times at each NFE. It can be seen that, in case of OPSO algorithm the FV reduces to a small value ($\sim 10^{-300}$ or lower) as NFE reaches about 1.5×10^{05}. Whereas, in case of GPSO algorithm, the FV fails to converge and remains above the “Accept Value”, which is indicating failure of the algorithm. The convergence characteristics of the eleven multimodal benchmark functions f_{10} – f_{20} shown in Fig. 10, indicate successful convergence of the OPSO algorithm and failure of the GPSO algorithm. Fig. 11 shows convergence characteristics of the ten shifted, rotated and shifted rotated benchmark functions f_{21} – f_{30}. It can be seen that the OPSO algorithm achieves much better convergence than GPSO algorithm in all the ten benchmark functions.

4.5. Comparison in terms of fitness values

Performance comparison between OPSO and GPSO algorithms in terms of BFV, WFV, MFV and σ are shown in Table 4. It can be seen that in case of GPSO algorithm, the three fitness values BFV, WFV and MFV differ substantially from their optimal values for all the thirty benchmark functions. Whereas, in OPSO algorithm, the three fitness values are the same or very close to their optimum values for all the thirty functions. The standard deviation σ remains 0 or close to 0 in OPSO algorithm, indicating high consistency and reliability. In terms of average execution time AET, the OPSO algorithm reaches the “Accept Value” within a specific AET as shown in Table 4. However, GPSO algorithm can not reach the “Accept Value”, therefore, it is out of the comparison. The results shown in Table 4 give evidence that the OPSO algorithm is more accurate, stable and robust compared to the GPSO algorithm.

In order to highlight the superior performance of the OPSO algorithm

Table 7
Performance comparison between OPSO algorithm and 4 ECTs on 2 multimodal functions with $d = 100$.

ECTs	Performance measure	f_{13}	f_{17}
OPSO	MFV	**-7.833233×10^{01}**	**-9.809959×10^{01}**
(proposed)	σ	**1.36×10^{-14}**	**3.88×10^{-01}**
	NFE	4.69×10^{04}	1.03×10^{05}
OGABC	MFV	-7.833230×10^{01}	-9.51915×10^{01}
[40]	σ	8.60×10^{-11}	6.15×10^{-01}
	NFE	**2.52×10^{04}**	**9.41×10^{04}**
OGAQ	MFV	-7.83×10^{01}	-9.283×10^{01}
[41]	NFE	1.34×10^{05}	1.34×10^{05}
EDAL	MFV	-7.83×10^{01}	-9.43757×10^{01}
[44]	NFE	1.14×10^{05}	1.14×10^{05}
LEA	MFV	-7.831×10^{01}	-9.301×10^{01}
[45]	NFE	1.30×10^{05}	1.30×10^{05}
Minimum $f(x)$		-7.833223×10^{01}	-9.92784×10^{01}
Accept Value $f(x)$		-7.80×10^{01}	9.50×10^{01}

over the GPSO algorithm, in Fig. 12, we provide the BFVs obtained in different runs for nine selected benchmark functions. The BFVs obtained by GPSO algorithm are unacceptable as these are much above the respective "Accept Value". In contrast, the OPSO algorithm was successful as the BFVs in these nine functions remain much below the respective "Accept Values". Similar observations were also made for the remaining functions.

4.6. Success rate and reliability rate

The performance comparison between the OPSO and the GPSO algorithms is carried out with $N_{run} = 25$ (independent runs) in terms of SR and RR. The GPSO algorithm fails in all the benchmark functions except in f_{16} in which it was successful only four times out of twenty five runs. However, the OPSO algorithm was successful in all the thirty benchmark functions giving rise to an SR of 100%. The RR of GPSO and OPSO algorithms are thus found to be 0.53% and 100%, respectively.

5. Comparison between the proposed OPSO algorithm and other ECTs

Here, we evaluate performance of the proposed OPSO algorithm by comparing it with few ECTs recently reported by other authors [4] [13–16], [18–24] [27], [29], [32–47]. The comparison includes four unimodal functions, f_1, f_5, f_6 and f_8, nine multimodal functions f_{10}, f_{11}, f_{13} and $f_{15} - f_{20}$, ten shifted, rotated and shifted-rotated benchmark functions, $f_{21} - f_{30}$.

5.1. Unimodal functions: d = 30

Performance comparison between the proposed OPSO algorithm and recently reported fifteen other ECTs for four unimodal functions f_1, f_5, f_6 and f_8 is shown in Table 5. Three performance measures MFV, σ and NFE are used for this comparison. In terms of MFV and σ, the proposed OPSO algorithm provides superior performance compared to the other ECTs. In terms of NFE, it is slightly inferior to [40] and [42]. In terms of AET the OPSO algorithm is slower than CCPSO-ISM algorithm [27] on functions f_1, f_6 and f_8.

5.2. Multimodal functions d = 30

Seven multimodal functions with 30-dimension are considered. The global minimum is more difficult to achieve in these multimodal functions. The performance comparison between the OPSO algorithm and 16 ECTs for 30-dimensional multimodal functions, f_{10}, f_{11}, f_{15}, f_{16} and $f_{18} - f_{20}$ are shown in Table 6. The OPSO algorithm shows superior performance compared with other ECTs in terms of MFV and σ in case of f_{11}, f_{15}, f_{16} and f_{19}. It also shows better performance in terms of NFE in case of f_{16} and f_{18}. In terms of other performance measures,

Table 8
Performance comparison between OPSO algorithm and other 9 ECTs on 6 shifted functions with $d = 30$.

ECTs	Performance measure	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}
OPSO	MFV	**-4.50×10^{02}**	**-4.50×10^{02}**	**3.90×10^{02}**	**-3.30×10^{02}**	**-1.40×10^{02}**	**-1.80×10^{02}**
(proposed)	σ	**1.58×10^{-13}**	**1.49×10^{-13}**	**1.10×10^{00}**	**4.34×10^{-14}**	**4.74×10^{-14}**	**1.27×10^{-12}**
	NFE	**1.88×10^{04}**	**2.34×10^{04}**	**2.99×10^{04}**	**1.49×10^{04}**	**1.84×10^{04}**	**1.42×10^{04}**
	AET (sec)	**8.9907**	**17.2141**	**12.3678**	**28.1717**	**8.9032**	**9.8174**
BPSO	MFV	-4.33×10^{02}	-4.29×10^{02}	4.01×10^{02}	-3.09×10^{02}	-	-
[18]	σ	3.98×10^{01}	4.40×10^{01}	1.16×10^{01}	4.3×10^{-03}	-	-
MCPSO	MFV	2.55×10^{03}	3.63×10^{01}	4.35×10^{07}	6.87×10^{02}	7.11×10^{00}	1.79×10^{01}
[19]	σ	3.43×10^{02}	1.97×10^{00}	6.945×10^{06}	2.03×10^{01}	1.00×10^{01}	1.66×10^{01}
	NFE	5×10^{04}	5×10^{04}	5×10^{04}	5×10^{04}	5×10^{04}	5×10^{04}
HCLPSO	MFV	-	1.70×10^{-06}	2.39×10^{00}	-	-	-
[20]	σ	-	1.71×10^{-06}	4.27×10^{00}	-	-	-
CSPSO	MFV	-	2.52×10^{-26}	3.99×10^{00}	-	-	-
[23]	σ	-	8.49×10^{-12}	2.74×10^{00}	-	-	-
	NFE		-	1.00×10^{05}	-	-	-
DTT-PSO	MFV	-	-	6.48×10^{01}	1.11×10^{02}	-	-
[29]	σ	-	-	7.28×10^{01}	1.65×10^{01}	-	-
	NFE	-	-	1.60×10^{05}	1.60×10^{05}	-	-
TPSO	MFV	4.47×10^{01}	2.67×10^{02}	1.30×10^{05}	4.26×10^{01}	-	-
[32]	σ	1.41×10^{02}	1.49×10^{02}	4.08×10^{05}	9.48×10^{00}	-	-
	NFE	5.00×10^{04}	5.00×10^{04}	5.00×10^{04}	5.00×10^{04}	-	-
FPSOCM	MFV	1.038×10^{-05}	1.99×10^{-01}	1.65×10^{-01}	5.58×10^{01}	-	-
[38]	σ	2.50×10^{-05}	1.24×10^{-01}	3.29×10^{-01}	4.0×10^{01}	-	-
	AET (sec)	24.46	43.52	12.672	40.045	-	-
BBO	MFV	1.7×10^{00}	0.8×10^{00}	1.1×10^{00}	12.4×10^{00}	5.1×10^{00}	–
[46]							
p-best AFEP	MFV	3.01×10^{-08}	4.056×10^{-04}	2.08×10^{02}	9.99×10^{-01}	2.04×10^{01}	–
[47]	σ	1.62×10^{-08}	2.91×10^{-03}	1.67×10^{03}	2.62×10^{-01}	1.02×10^{01}	–
	NFE	3.00×10^{05}	3.00×10^{05}	3.00×10^{05}	3.00×10^{05}	3.00×10^{05}	–
Minimum $f(x)$		-4.50×10^{02}	-4.50×10^{02}	3.90×10^{02}	-3.30×10^{02}	-1.40×10^{02}	-1.80×10^{02}
Accept Value $f(x)$		-3.50×10^{02}	-3.50×10^{02}	4.90×10^{02}	-2.30×10^{02}	-1.35×10^{02}	-1.70×10^{02}

Table 9
Performance comparison between OPSO algorithm and other 7 ECTs on 4 rotated and shifted rotated functions with $d = 30$.

ECTs	Performance measure	f_{27}	f_{28}	f_{29}	f_{30}
OPSO	MFV	**-90.0×10^{02}**	**-4.50×10^{02}**	**-3.29×10^{02}**	**-1.80×10^{02}**
(proposed)	σ	**1.64×10^{-14}**	**4.73×10^{-13}**	2.15×10^{00}	**2.59×10^{-14}**
	NFE	**5.20×10^{02}**	**3.20×10^{04}**	**1.90×10^{04}**	**3.20×10^{04}**
	AET (sec)	**2.1427**	**9.9062**	**26.2184**	**26.6115**
BPSO	MFV	-	-4.37×10^{02}	-3.09×10^{02}	-1.59×10^{02}
[18]	σ	-	236×10^{01}	4.1×10^{-03}	3.30×10^{01}
HCLPSO	MFV	-	2.61×10^{05}	5.60×10^{01}	2.00×10^{-02}
[20]	σ	-	6.42×10^{05}	1.29×10^{01}	2.00×10^{-02}
CSPSO	MFV	-	3.64×10^{04}	-	-
[23]	σ	-	3.22×10^{04}	-	-
	NFE	-	-	-	-
DTT-PSO	MFV	-	-	8.47×10^{01}	1.10×10^{-02}
[29]	σ	-	-	1.30×10^{01}	4.85×10^{-03}
	NFE	-	-	1.60×10^{05}	1.60×10^{05}
TPSO	MFV	-	5.23×10^{06}	5.78×10^{01}	1.40×10^{01}
[32]	σ	-	1.85×10^{06}	2.21×10^{01}	1.79×10^{01}
	NFE	-	5.0010^{04}	5.0010^{04}	5.0010^{04}
BBO	MFV	-	0.3×10^{00}	-	9.9×10^{00}
[46]					
p-best AFEP	MFV	-	5.07×10^{05}	3.28×10^{01}	2.34×10^{-03}
[47]	σ	-	1.36×10^{05}	2.00×10^{01}	5.34×10^{-04}
	NFE	-	3.00×10^{05}	3.00×10^{05}	3.00×10^{05}
Minimum $f(x)$		-9.00×10^{02}	-4.50×10^{02}	-3.30×10^{02}	-1.80×10^{02}
Accept Value $f(x)$		-8.00×10^{02}	-3.50×10^{02}	-3.00×10^{02}	-1.80×10^{02}

Table 10
Performance comparison between OPSO algorithm and other 3 ECTs on 6 shifted functions with $d = 100$.

ECTs	Performance measure	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}
OPSO	MFV	**-4.50×10^{02}**	**-4.50×10^{02}**	**3.90×10^{02}**	**-3.30×10^{02}**	**-1.40×10^{02}**	**-1.80×10^{02}**
(proposed)	σ	3.33×10^{-13}	**4.11×10^{-13}**	**6.60×10^{01}**	**2.41×10^{-13}**	8.03×10^{-14}	2.84×10^{-14}
CSO	MFV	**-4.50×10^{02}**	-4.16×10^{02}	7.80×10^{02}	-2.74×10^{02}	1.39×10^{02}	**-1.80×10^{02}**
[33]	σ	**1.10×10^{-28}**	5.38×10^{00}	5.53×10^{02}	7.48×10^{00}	1.52×10^{-01}	**0.0**
CCPSO2	MFV	**-4.50×10^{02}**	-4.43×10^{02}	8.13×10^{02}	-3.29×10^{02}	**-1.40×10^{02}**	**-1.80×10^{02}**
[34]	σ	3.23×10^{-14}	7.83×10^{00}	8.65×10^{02}	1.99×10^{-01}	3.06×10^{-14}	4.88×10^{-03}
SL-PSO	MFV	**-4.50×10^{02}**	**-4.50×10^{02}**	9.64×10^{02}	2.55×10^{02}	**-1.40×10^{02}**	**-1.80×10^{02}**
[35]	σ	3.50×10^{-28}	4.97×10^{-06}	1.76×10^{02}	1.21×10^{01}	**-1.40×10^{-15}**	**0.0**
Minimum $f(x)$		-4.50×10^{02}	-4.50×10^{02}	3.90×10^{02}	-3.30×10^{02}	-1.40×10^{02}	-1.80×10^{02}
Maximum NFE		500,000					

performance of the OPSO algorithm is comparable to other ECTs, in case of other functions.

5.3. Multimodal functions $d = 100$

Out of thirty functions considered in this study, f_{13} and f_{17} are two multimodal functions with dimension of 100. These two functions are more complicated, have several local minima and their number change dramatically with increase in dimension [40]. In Table 7, performance comparison between OPSO algorithm and other four available ECTs is provided. The proposed OPSO algorithm performs better than the other four ECTs in terms of MFV and σ for both functions. However, its performance in terms of NFE is slightly inferior to other ECTs.

5.4. Shifted, rotated and shifted rotated functions $d = 30$

Ten shifted, rotated and shifted rotated benchmark functions with 30-dimension are considered. The global minimum is more difficult to achieve in these functions, because, the solution vector is shifted from the origin by the vector O, rotated by matrix M or shifted and rotated by O and M. The performance comparison between the OPSO algorithm and 9 ECTs for 30-dimensional shifted functions, $f_{21} - f_{26}$ are shown in Table 8. The OPSO algorithm shows superior performance compared with other 9 ECTs in terms of four fitness values, MFV, σ, NFE and AET. The performance of the OPSO algorithm is the best among the 9 ECTs. Table 9 shows the comparison between OPSO algorithm and 7 other ECTs for 4 rotated and shifted rotated functions $f_{27} - f_{30}$. The OPSO algorithm is found to perform the best in terms of MFV, σ and NFE for these four functions.

5.5. Shifted functions $d = 100$

Six shifted benchmark functions, i.e., $f_{21} - f_{26}$ with $d = 100$ are selected from CEC 2008 [55]. The performance comparison between the OPSO algorithm and three ECTs (CSO [33], CCPSO2 [34] and SL-PSO [35]) is shown in Table 10. As recommended in Ref. [52], the comparison has been implemented with $d = 100$ and $N_{run} = 25$. For each independent run, the NFE $= d \times 5000 = 500{,}000$. The other parameters used in OPSO algorithm were $m = 110$, $N_{iter} = 4545$ based on (24) and $c = 2.05$. It can be seen that for f_{23} and f_{24}, the OPSO algorithm outperforms the other three ECTs in terms of MFV and σ. For f_{21} and f_{25}, the OPSO algorithm shares the three ECTs with similar results in terms of MFV and σ. In case of f_{22}, performance of the OPSO algorithm is similar to that of the SL-PSO algorithm. Finally, for f_{26}, the OPSO algorithm performs similar to other three ECTs in terms of MFV. However, in terms of σ, its performance is better than CCPSO2 only.

From the above observations, it is evident that the proposed OPSO algorithm gives better performance compared to that of recently reported ECTs for solving unimodal, multimodal and functions in terms of convergence, accuracy, stability and reliability.

6. Conclusion

A novel orthogonal PSO (OPSO) algorithm was proposed whose performance is found to be superior than the GPSO algorithm. While searching for optimal solution in a d-dimensional space, the OPSO algorithm with m particles ($m > d$) divides the swarm population into two groups. The first group, named active group, consists of d particles that have the best personal experiences. The remaining ($m - d$) particles constitute the passive group. The two groups introduce diversity in swarm population. Using an orthogonal diagonalization process the position vectors of only the active group are updated. When the convergence is attained, the d position vectors constitute a diagonal (orthogonal) matrix. Due to the improvement of the updating procedure, the OPSO algorithm avoids the zigzagging phenomenon of GPSO algorithm. From the sensitivity analysis, we infer that, the swarm population size need to be selected at about 10–30% higher than the search dimension, d. By taking thirty unimodal, multimodal, shifted, rotated, and shifted rotated benchmark functions of dimension 30 and 100, we have shown that the OPSO algorithm outperforms the GPSO algorithm and several recently reported ECTs in terms of convergence, accuracy, consistency and reliability.

References

[1] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks vol. 4, 1995, pp. 1942–1948.

[2] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.

[3] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the Congress on Evolutionary Computation, 2002, pp. 1671–1676, 2.

[4] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.

[5] T. Niknam, H.D. Mojarrad, M. Nayeripour, A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch, Energy 35 (2010) 1764–1778.

[6] Z. Ren, A. Zhang, C. Wen, Z. Feng, A scatter learning particle swarm optimization algorithm for multimodal problems, IEEE Trans. Cybern. 44 (7) (2014) 1127–1140.

[7] Y.H. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, 1998, pp. 69–73.

[8] R.C. Eberhart, Y.H. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of the IEEE International Conference on Evolutionary Computation, 2000, pp. 84–88.

[9] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Comput. Oper. Res. 33 (3) (2006) 859–871.

[10] K. Lei, Y. Qiu, Y. He, A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization, in: Proceedings of the International Symposium on Systems and Control in Aerospace and Astronautic (ISSCAA), 2006, pp. 977–980.

[11] X. Yang, J. Yuan, J. Yuan, H. Mao, A modified particle swarm optimizer with dynamic adaptation, Appl. Math. Comput. 189 (2) (2007) 1205–1213.

[12] M.S. Arumugam, M.V. Rao, On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems, Appl. Soft Comput. 8 (1) (2008) 324–336.

[13] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baska, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[14] A. Ratnaweera, S. Halgamuge, H. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.

[15] W.N. Chen, J. Zhang, Y. Lin, N. Chen, Z.H. Zhan, H.S. Chung, Y. Li, Y.H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2013) 241–258.

[16] T.T. Arani, N.A. Sadollah, J.H. Kim, A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems, J. Comput. Sci. 13 (2016) 68–82.

[17] R. Vafashoar, M.R. Meybodi, Multi swarm bare bones particle swarm optimization with distribution adaption, Appl. Soft Comput. 47 (2016) 534–552.

[18] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.

[19] J. Jie, J. Zang, H. Zheng, B. Hou, Formalized model and analysis of mixed swarm based cooperative particle swarm optimization, Neurocomputing 174 (2016) 542–552.

[20] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.

[21] M. Taherkhani, R. Safabakhsh, A novel stability-based adaptive inertia weight for particle swarm optimization, Appl. Soft Comput. 38 (2016) 281–295.

[22] X. Zhao, W. Lin, J. Hao, X. Zuo, J. Yuan, Clustering and pattern search for enhancing particle swarm optimization with Euclidean spatial neighborhood search, Neurocomputing 171 (2016) 966–981.

[23] A. Meng, Z. Li, H. Yin, S. Hao, Chen, Z. Guo, Accelerating particle swarm optimization using crisscross search, Inf. Sci. 329 (2016) 52–72.

[24] K. Yu, X. Wang, Z. Wang, Multiple learning particle swarm optimization with space transformation perturbation and its application in ethylene cracking furnace optimization, Knowledge-Based Syst. 96 (2016) 156–170.

[25] Z.-H. Ruan, Y. Yuan, Qi-X. Chen, C.-X. Zhang, Y. Shuai, H.-P. Tan, A new multi-function global particle swarm optimization, Appl. Soft Comput. 49 (2016) 279–291.

[26] S. Sun, J. Li, A two-swarm cooperative particle warms optimization, Swarm Evol. Comput. 15 (2014) 1–18.

[27] Y. Li, Z.-H. Zhan, S. Lin, J. Zhang, X. Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, Inf. Sci. 293 (2015) 370–382.

[28] M.R. Tanweer, R. Audity, S. Suresh, N. Sundararajan, N. Srikanth, Directionally Driven self-regulating particle swarm optimization algorithm, Swarm Evol. Comput. 28 (2016) 98–116.

[29] L. Wang, B. Yang, J. Orchard, Particle swarm optimization using dynamic tournament topology, Swarm Evol. Comput. 48 (2016) 584–596.

[30] S. Gunasundari, S. Janakiraman, S. Meenambal, Velocity Bounded Boolean Particle Swarm Optimization for improved feature selection in liver and kidney disease diagnosis, Expert Syst. Appl. 56 (2016) 28–47.

[31] A. Sharma, R. Kumar, B.K. Panigrahi, S. Das, Termite spatial correlation based particle swarm optimization for unconstrained optimization, Swarm Evol. Comput. 33 (2017) 93–107.

[32] B.O. Arani, P. Mirzabeygi, M.S. Panahi, An improved PSO algorithm with a territorial diversity preserving scheme and enhanced exploration-exploitation balance, Swarm Evol. Comput. 11 (2013) 1–15.

[33] R. Chen, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. 45 (2) (2015) 191–204.

[34] X. Li, Cooperatively coevolving particle swarm for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (2012) 210–224.

[35] R. Cheng, Y. Jin, A Social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015) 43–60.

[36] N.J. Cheung, X. Ding, H. Shen, OptiFel: a convergent heterogeneous particle swarm optimization algorithm for Takagi-Sugeno fuzzy modeling, IEEE Trans. Fuzzy Syst. 22 (4) (2014) 919–933.

[37] M. Han, J. Fan, J. Wang, A Dynamic feed forward neural network based on Gaussian particle swarm optimization and its application for predictive control, IEEE Trans. Neural Netw. 22 (9) (2011) 1457–1468.

[38] S.H. Ling, K.Y. Chan, F.H. Leung, F. Jiang, H. Nguyen, Quality and robustness improvement for real world industrial systems using a fuzzy particle swarm optimization, Eng. Appl. Artif. Intell. 47 (2016) 68–80.

[39] Z.H. Zhan, J. Zhang, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (6) (2011) 832–847.

[40] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024.

[41] Y.W. Leung, Y. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization, IEEE Trans. Evol. Comput. 5 (1) (2011) 41–53.

[42] S. Das, A. Abraham, U. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.

[43] H. Wang, S. Rahamayan, H. Sun, M.H. Omran, Gaussian bare-bones differential evolution, IEEE Trans. Cybern. 43 (2) (2013) 634–647.

[44] Q. Zhang, J. Sun, E. Tsang, J. Ford, Hybrid estimation of distribution algorithm for global optimization, Eng. Comput. 21 (1) (2004) 91–107.

[45] Y.P. Wang, C.Y. Dang, An evolutionary algorithm for global optimization based on level-set evolution and Latin squares, IEEE Trans. Evol. Comput. 11 (5) (2007) 579–595.

[46] D. Simon, A. Shah, C. Scheidegger, Distributed learning with biogeography-based optimization: Markov modeling and robot control, Swarm Evol. Comput. 10 (2016) 12–24.

[47] S. Das, R. Mallipeddi, D. Maity, Adaptive evolutionary programming with *p*-best mutation strategy, Swarm Evol. Comput. 9 (2013) 58–68.

[48] L.T. Al Bahrani, J.C. Patra, Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–7.

[49] L.T. Al Bahrani, J.C. Patra, Orthogonal PSO algorithm for economic dispatch of power under power grid constraints, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2015, pp. 14–19.

[50] L.T. Al Bahrani, J.C. Patra, R. Kowalczyk, Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints, in: Proceedings of International Joint Conference on Neural Networks (IJCNN), 2016, pp. 660–667.

[51] D. Bratton, J. Kennedy, Defining a standard for particle swarm optimization, in: Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), 2007, pp. 120–127.

[52] S.I. Grossman, Elementary Linear Algebra, Saunders College Publishing, USA, 1994.

[53] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC'2005 Special Session on Real Parameter Optimization, Nanyang Technological University, Tech. Rep, 2005. Available online in, http://www.ntu.edu.sg/home/epnsugan/index_files/cec05/Tech-Report-May-30–05.pdf.

[54] B.Y. Qu, J.J. Liang, Z.Y. Wang, Q. Chen, P.N. Suganthan, Novel benchmark functions for continuous multimodal optimization with comparative results, Swarm Evol. Comput. 26 (2016) 23–34.

[55] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.P. Chen, C.M. Chen, Z. Yang, Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization, Tech. Rep, 2008. Available online in, http://www.al-roomi.org/multimedia/CEC_Database/CEC2008/CEC2008_Technical.Report.pdf.

[56] J. Liang, Y. Qu, P. Suganthan, A. Hernández Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-parameter Optimization, Technical Report, Zhengzhou University and Nanyang Technological University, 2012.

Paper F: Applied Soft Computing 2017

Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid

Article reference:	ASOC4192
Journal:	Applied Soft Computing Journal
Received at Editorial Office:	15 Nov 2016
Article revised:	21 Apr 2017
Article accepted for publication:	26 Apr 2017
DOI:	10.1016/j.asoc.2017.04.059

Applied Soft Computing 58 (2017) 401–426

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid

Loau Tawfak Al Bahrani*, Jagdish Chandra Patra

Swinburne University of Technology, Melbourne, Australia

ARTICLE INFO

Article history:
Received 15 November 2016
Received in revised form 21 April 2017
Accepted 26 April 2017
Available online 2 May 2017

Keywords:
Orthogonal particle swarm optimization
Orthogonal diagonalization
Economic dispatch
Ramp rate limits
Prohibited operating zones

ABSTRACT

Particle swarm optimization (PSO) algorithm has been successfully applied to solve various optimization problems in science and engineering. One such popular one is called global PSO (GPSO) algorithm. One of major drawback of GPSO algorithm is the phenomenon of "zigzagging", that leads to premature convergence by falling into local minima. In addition, the performance of GPSO algorithm deteriorates for high-dimensional problems, especially in presence of nonlinear constraints. In this paper we propose a novel algorithm called, orthogonal PSO (OPSO) that alleviates the shortcomings of the GPSO algorithm. In OPSO algorithm, the *m* particles of the swarm are divided into two groups: active group and passive group. The *d* particles of the active group undergo an orthogonal diagonalization process and are updated in such way that their position vectors become orthogonally diagonalized. In the OPSO algorithm, the particles are updated using only one guide, thus avoiding the conflict between the two guides that occurs in the GPSO algorithm. We applied the OPSO algorithm for solving economic dispatch (ED) problem by taking three power systems under several power constraints imposed by thermal generating units (TGUs) and smart power grid (SPG), for example, ramp rate limits, and prohibited operating zones. In addition, the OPSO algorithm is also applied for ten selected shifted and rotated CEC 2015 benchmark functions. With extensive simulation studies, we have shown superior performance of OPSO algorithm over GPSO algorithm and several existing evolutional computation techniques in terms of several performance measures, e.g., minimum cost, convergence rate, consistency, and stability. In addition, using unpaired *t*-Test, we have shown the statistical significance of the OPSO algorithm against several contending algorithms including top-ranked CEC 2015 algorithms.

1. Introduction

Economic dispatch (ED) of power is one of the fundamental problems in smart power grid (SPG) operations. Its objective is to allocate the load demand among committed thermal generating units (TGUs) in most economical manner, while satisfying various practical power constraints imposed by the TGUs and SPG. In order to make SPG more efficient, flexible, adaptive and reliable, these constraints need to be integrated with smart meters and sensors, advanced communication technology and high-performance computing machines [1]. Therefore, there is a need to develop new computational techniques to solve ED problem that is compatible with rapid technological evolution in SPG.

Traditionally, the cost function of a TGU is approximately represented by a quadratic or a piecewise quadratic function [2,3]. However, due to several power constraints, e.g., ramp rate limits (RRLs) and prohibited operating zones (POZs), the cost function of an on-line TGU becomes non-convex and non-smooth with multiple modes, i.e., multimodal objective function. Because, the operating range of on-line TGUs is restricted by their RRLs and discontinuities in the cost curve due to their POZs.

In the past decades, many optimization techniques including traditional methods have been adopted in order to find the optimum power dispatch and the rate of optimum product for each on-line TGU. Some of the traditional methods include linear programming [4], quadratic programming [5], Lagrange relaxation [6], Lambda iterative method [7], and dynamic programming [8]. These methods offer certain advantages, for example, they only need to run once and do not have any problem specific parameters to specify. However, the traditional methods are able to solve the ED problem only when the cost function is piecewise linear and monotonically increasing.

To overcome the above mentioned deficiencies, several evolutionary computation techniques (ECTs) have been proposed to tackle the non-convex ED problems. Some of the popular ECTs

* Corresponding author.
E-mail addresses: lalbahrani@swin.edu.au (L.T. Al Bahrani), JPatra@swin.edu.au (J.C. Patra).

http://dx.doi.org/10.1016/j.asoc.2017.04.059
1568-4946/© 2017 Elsevier B.V. All rights reserved.

include genetic algorithm (GA) [9], simulated annealing [10], artificial immune system (AIS) [11], artificial bee colony [12], evolutionary programming [13], differential evolution (DE) [14], harmony search [15]. Another popular ECT is the particle swarm optimization (PSO) algorithm proposed by Kennedy and Eberhart [16,17]. The PSO algorithm can be divided into types, one for local PSO (LPSO) and the other for global PSO (GPSO) which is more popular. Some of the advantages of application of GPSO algorithm in solving ED problem are: firstly, it imposes a few or no restrictions on the shape of the cost function; secondly, it has a few parameters and is easy to execute. However, the GPSO algorithm is suitable only for solving optimization problems that have a continuous domain and it is prone to get trapped into local minima when applied to multimodal functions.

The GPSO algorithm mimics the behaviour of swarm population of some animal species, such as, birds or fish flocking. The GPSO algorithm operates by initializing the particles (with possible solutions) of the swarm randomly and searching for global optima by updating the position and the velocity of each particle iteratively. While updating, each particle in a swarm uses its own personal experience and the best experience of the swarm as two guides through linear summation. Thus, a particle that has best experience among the personal experiences of all particles is considered to be a best solution. The reasons for poor performance of the GPSO algorithm can be summarized as follows. Firstly, the learning strategy of the GPSO algorithm depends on the fact that each particle in a swarm adjusts its search trajectory according to its own personal experience and its neighbors' experiences. Therefore, each particle in a swarm obtains two possible solutions, one from its personal experience and the other from its neighborhood's experience and then sums them together. The problem here is not only existence of the summation, but also the presence of two guides. These two guides may have a large difference or may even be opposite directions at the early search step, which may lead the particles to pull to local minima trapping and may lead to early convergence. Often, they remain in opposite directions until final search stage. Secondly, these two guides and their linear summation may cause a phenomenon called "Oscillation or zigzagging" [18–20]. This phenomenon becomes more prominent with high dimensional search space in which the particles remain in a confused state and are unable to control their search trajectories.

Several ECTs have been proposed in recent years to boost the global search ability for solving different types and forms of unimodal and multimodal objective functions. Some of these ECTs have been focused on solving non-convex multimodal cost function and improving the performance of the GPSO and LPSO algorithms in order to obtain a global optimum and to prevent the falling into local minima. For example, self-organizing hierarchical PSO (SOH-PSO) algorithm [3], modified PSO algorithm [21], PSO with modified stochastic acceleration factors (PSO-MSAF) [22], an improved PSO algorithm [23], hybrid gradient PSO (HGPSO) [24], hybrid PSO with mutation (HPSOM) algorithm [24], hybrid PSO with wavelet mutation (HPSOWM) algorithm [24], random drift PSO (RDPSO) [25], simulated annealing PSO (SA-PSO) algorithm [26], new PSO local random search (NPSO-LRS) algorithm [27], anti-predatory PSO (APSO) [28], chaotic PSO (CPSO) algorithm [29], and quantum mechanism PSO (QMPSO) algorithm [30] have been proposed. In [25], eleven ECTs, i.e., GA, DE, ant colony search algorithm (ACSA), bee colony optimization (BCO), AIS, firefly algorithm (FA), APSO, HGPSO, HPSOM, HPSOWM, RDPSO have been applied to the ED problem with three different power systems.

Other ECTs have been applied on solving other types of unimodal and multimodal objective functions, i.e., shifted and rotated CEC 2015 benchmark objective functions [31,32]. For example, local Lipschits underestimate differential evolution (LLUDE) [33], strategy adaptation differential evolution (SaDE) [33], JADE: adaptive differential evolution [33], composite differential evolution (CoDE) [33], self-adaptive binary variant differential evolution (SbaDE) [34], directionally driven self-regulating PSO (DD-SRPSO) [35], extraordinariness PSO (EPSO) [36], shrinking hypersphere PSO with gravitational search algorithm (SHPSO-GSA) [37] have been proposed.

In order to make a fair comparison, some other ECTs are also used in the literature. They include fully decentralized approach (FDA) [38], biogeography-based optimization (BBO) algorithm [39], genetic algorithm with API (GA-API) algorithm [40], mixed-integer quadratically constrained quadratic programming (MIQCQP) [41], enhanced gradient simplified swarm optimization algorithm (EGSSOA) [42], Lambda logic (λ-logic) [43], self-tuning hybrid differential evolution [44], two-phase neural network [45], synergic predator-prey optimization (SPPO) algorithm [46], and chaotic teaching-learning-based optimization algorithm [47], self optimization based adaptive DE with linear population, L-SHADE and eigenvector-based crossover and successful-parent-selecting, SPS-L-SHADE-EIG [48], DE with success-based parameter adaption (DEsPA) algorithm [49], mean-variance mapping optimization (MVMO) algorithm [50,51], and tuned covariance matrix evolution strategy (TunedCMAES) [52].

In this paper, we propose a novel algorithm called orthogonal PSO (OPSO) algorithm with a new learning strategy to solve non-convex ED problem for TGUs under several practical TGUs and power grid constraints and to improve the performance by overcoming the drawbacks of GPSO algorithm. The OPSO algorithm consists of a swarm with m particles that looks for optimal solution in a d-dimensional search space ($m \geq d$). The swarm population is divided into two groups: an active group of best personal experience of d particles and another passive group of personal experience of remaining (m – d) particles. The position vectors associated with the m particles undergo an orthogonal diagonalization (OD) process in which the d orthogonal guidance vectors in the active group are obtained. In each iteration, using only one guide, the velocity and position vectors of only the active group particles and the remaining (m – d) particles are left unchanged. This avoids the conflicting situation of the GPSO algorithm and leads the best d particles towards the optimal solution in multi-dimensional search space. We applied the OPSO algorithm to small, medium and large TGUs power system and ten selected shifted and rotated CEC 2015 benchmark functions. We have shown that the OPSO algorithm is able to achieve superior performance in terms of convergence, stability and accuracy compared to GPSO algorithm and several competitive ECTs. In the recent works, the effectiveness of the proposed OPSO algorithm has been shown for finding optimum power dispatch in smart power grid applications [53–55].

Recently, the ECTs in [18–20] have been proposed under the name, "orthogonal", i.e., orthogonal learning PSO (OLPSO) algorithms, one for local and the other for global optimization [18], orthogonal global-best-guided artificial bee colony algorithm [19], and orthogonal genetic algorithm with quantization [20]. They are using a different approach called, orthogonal experimental design (OED). The OED allows the inputs interact among them such that the output process can be optimized. The OED works on a predefined table of an orthogonal array of N factors with Q levels per factor. The OED is applied to obtain a set of possible solutions to achieve the optimal solution. However, the drawbacks through applying OED are: firstly, it holds only when no or weak interaction among the factors exists; secondly, the table that contains variable designs is complicated; thirdly, the orthogonality may not be possible achieve in the complex problems.

The rest of the paper is organized as follows. Section 2 describes the non-convex ED problem under various power constraints. Explanation of the learning strategy of the GPSO algorithm is presented in Section 3. Details of the proposed OPSO algorithm are provided in Section 4. In Section 5, the application of the OPSO algorithm to ED problem for three power systems is presented. In Section 6, the application of the OPSO algorithm to ten selected CEC 2015 benchmark functions is presented. Finally, conclusion of this study is given in Section 7.

2. Problem formulation

Here, we explain the cost function and various practical power constraints involved in this study.

2.1. Cost objective function

The main purpose of ED problem is to estimate the optimum arrangement of on-line TGUs generation in order to minimize the entire generation fuel cost subjected to the on-line TGUs and power grid constraints. The fuel cost function of each on-line TGU is characterized by quadratic function [25] and given by:

$$F(P_j) = a_j + b_jP_j + c_jP_j^2 \tag{1}$$

where $F(P_j)$ is the fuel cost function of jth TGU in ($/h), P_j is the output active power of jth TGU in (MW), a_j, b_j, and c_j are fuel cost coefficients of jth TGU. The total fuel cost of the on-line TGUs is given by

$$\text{Minimize}F_{\cos t} = \sum_{j=1}^{N_{gen}} F(P_j) \tag{2}$$

where N_{gen} is the number of committed on-line TGUs. F_{cost} is the function to be minimized.

2.2. Power constraints

Different practical power constraints imposed on on-line TGUs and by SPG used in the literature are explained below.

2.2.1. Power balance constraint

The total output power of committed on-line TGUs should be able to satisfy load demand and transmission network loss. The power balance constraint is given as

$$\sum_{j=1}^{N_{gen}} P_j = P_D + P_L \tag{3}$$

where P_D is load demand in (MW) and P_L is transmission network loss in (MW).

2.2.2. Transmission network loss

The transmission network loss $\mathrm{P_L}$ is a critical constraint of the ED problem. Not only is it desired that the power loss incurred in the system be minimized along with the total fuel cost, but the system must also generate enough power to satisfy the load demand as well as to compensate for the $\mathrm{P_L}$. The $\mathrm{P_L}$ is given by [38,56]:

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k \tag{4}$$

where B_{jk}, are known as the loss coefficients or B-coefficients [56,57].

2.2.3. Transmission network loss mismatch

The total power generated is obtained using (3). From (3), the $\mathrm{P_L}$ is obtained and let's call $\mathrm{P_{L1}}$ as follows

$$P_{L1} = \sum_{j=1}^{N_{gen}} P_j - P_D \tag{5}$$

Besides, the P_L is also computed using (4) and let's denote this time by P_{L2} as follows:

$$P_{L2} = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k \tag{6}$$

Then, the difference between two, P_{L2} and P_{L1}, is called transmission network loss mismatch ($P_{L,mismatch}$) as follows:

$$P_{L,mismatch} = P_{L2} - P_{L1} \tag{7}$$

The significance of the $P_{L,mismatch}$ can be expressed as follows:

- The effectiveness and the accuracy of the algorithm in computing optimal P_j can be determined.
- One can determine whether or not the power balance constraint (3) is satisfied.

When $P_{L,mismatch} = 0$, then $P_{L2} = P_{L1}$. In such case, the (5) can be written as

$$P_{L2} = \sum_{j=1}^{N_{gen}} P_j - P_D \tag{8}$$

Subsequently, the (3) is satisfied.

2.2.4. Generation limits

Each TGU has a specified range within which its operation is stable. Therefore, it is desired that the TGUs be run within this range in order to maintain system stability

The generation limits of the jth TGU is given by

$$P_{j,min} \le P_j \le P_{j,max} \quad \mathrm{j} = 1, 2, \ldots, N_{gen} \tag{9}$$

In other words, the power generation of each TGU must remain between its minimum $P_{j,min}$ and its maximum $P_{j,max}$ limits.

2.2.5. Ramp rate limits

The operating range of all on-line TGUs is restricted by their ramp rate limits RRLs due to the physical limitation of TGUs [58,59]. For any sudden change in the load demand, TGUs increase or decrease their generation in order to satisfy the power balance constraint (3). However, the TGUs can change their generation only at a certain rate determined by the up-ramping and down-ramping rate. If a TGU is operating at a specific point, then its point of operation can be changed only up to a certain rate determined by the ramp rate. Therefore, a change in TGU output power from one specific interval to the next cannot exceed a specified limit.

If power generation need to increase, then per unit time rate of increase $P_j - {P_j}^0$ must satisfy

$$P_j - {P_j}^0 \le UR_j \tag{10}$$

If power generation need to decrease, then per unit time rate of decrease ${P_j}^0 - P_j$ must satisfy

$${P_j}^0 - P_j \le DR_j \tag{11}$$

where ${P_j}^0$ is the TGU output active power at the previous time interval and P_j is the TGU output power at current time interval. The UR_j

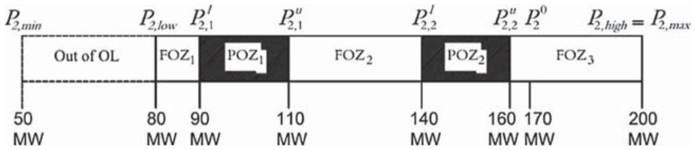


Fig. 1. Lower and upper generation limits, POZs and FOZs for TGU$_2$.

and DR_j are the up-ramp and down-ramp limits of TGU j, respectively, in (MW/h). By substituting (10) and (11) in (9), the (12) is obtained.

$$max\{P_{j,min}, (P_j{}^0 - DR_j)\} \le P_j \le min\{P_{j,max}, (P_j{}^0 + UR_j)\} \tag{12}$$

Let us assume that,

$$P_{j,low} = max\{P_{j,min}, (P_j{}^0 - DR_j)\}, \text{ and} \tag{13}$$

$$P_{j,high} = min\{P_{j,max}, (P_j{}^0 + UR_j)\} \tag{14}$$

where $P_{j,low}$ and $P_{j,high}$ are the new lower and higher limits of unit j, respectively.

2.2.6. Prohibited operating zones

The physical limitations due to the steam valve operation or vibration in a shaft bearing of TGU may result in the generation units operating within prohibited operating zones (POZs) [60]. The POZs make the cost function discontinuous in nature. In such case, it is difficult to determine the shape of the cost curve under POZs through actual performance testing. In addition, if the TGU operates within the POZs range in the SPG then it may lead to loss of stability. Therefore, these regions are usually avoided during generation. By using (9) mentioned in constraint 2.2.4, the feasible operating zones FOZs of the jth TGU are given by

$$\begin{aligned} & P_{j,\min} \le P_j \le P^l_{j,1} \\ & \quad P^u_{j,k-1} \le P_j \le P^l_{j,k},\ k = 2, 3, \ldots, N_{j,PZ} \\ & \quad P^u_{j,N_{j,PZ}} \le P_j \le P_{j,\max} \end{aligned} \tag{15}$$

where$P^l_{j,k}$and $P^u_{j,k}$are the lower and upper bound of the kth POZs of the jth unit, and $N_{j,PZ}$ is number of POZs of the jth TGU. Incorporating these power constraints in (12)–(15), we get the final set of inequality power constraints imposed on TGU in SPG are as follows:

$$\begin{aligned} & P_{j,low} \le P_j \le P^l_{j,1} \\ & \quad P^u_{j,k-1} \le P_j \le P^l_{j,k},\ k = 2, 3, \ldots, N_{jPZ} \\ & \quad P^u_{j,N_{j,PZ}} \le P_j \le P_{j,high} \end{aligned} \tag{16}$$

2.2.6.1. Observation. Eq. (16) gives the final set of the inequality power constraints imposed on jth TGU in terms of new lower and upper generation limits with RRLs and FOZs. In addition, the (16) avoids all POZ$_S$ imposed on jth TGU. Thus, all TGUs will have operation limits (OLs) satisfying all power constraints.

2.2.6.2. An illustrative example. In order to illustrate new lower and upper generation limits and FOZs generated due to presence RRLs and POZs of jth TGU, an example of specifications of TGU$_2$ per one hour generation is given below [25]:

$P_2{}^0 = 170$ MW; $P_{2,min} = 50$ MW; $P_{2,max} = 200$ MW; $UR_2 = 50$ MW; $DR_2 = 90$ MW. The TGU$_2$ has two POZs are: POZ$_1$ = [90,110] and POZ$_2$ = [140,160].

From (16), the new lower and upper limits of TGU$_2$ based on RRLs are:

$P_{2,low} = 80$ MW and $P_{2,high} = 200$ MW,
and there are three FOZs are:

$$\text{FOZ}_1: 80 \le P_2 \le 90$$

$$\text{FOZ}_2: 110 \le P_2 \le 140$$

$$\text{FOZ}_3: 160 \le P_2 \le 200$$

Fig. 1 shows that TGU$_2$ has minimum and maximum OLs given by 50 MW and 200 MW, respectively. However, due to presence RRLs (up-ramp and down-ramp limits) power constraint, TGU$_2$ operates within new lower and higher OLs given by $P_{2,low} = 80$ MM and $P_{2,high} = 200$ MW. In addition, three FOZs are: FOZ$_1$ = [80,90] MW, FOZ$_2$ = [110,160] MW, FOZ$_3$ = [160,200] MW in white color, and two POZs are: POZ$_1$ = [90,110] MW, POZ$_2$ = [140,160] MW in dark color are shown in Fig. 1. The intermittent zone ([50,80] MW) is out of OL of the TGU$_2$.

3. The GPSO algorithm

The mechanism of GPSO algorithm depends on distribution of the particles (possible solutions) in a swarm. Firstly, each particle flying in the multi-dimensional search area adjusts its flying trajectory according to two guides, its personal experience ($G_{pers,i}$) and its neighborhood's best experience (G_{best}). Secondly, when seeking a global solution, each particle learns from its own historical experience and its neighborhood's historical experience. In such a case, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. Since the position of each particle is affected by the best-fit particle in the entire swarm, this technique is named, global PSO [16,17,61–64]. Only a few parameters have been used in GPSO algorithm later to give potential advantage and to enhance its performance. Among user parameters of GPSO algorithm, several strategies of inertia weight have been used. For example, constant inertia weight [61], time-varying inertia weight [62] and constriction factor for balancing global and local searches [63]. However, when the cost function is within high-dimensional search space and restricted by many POZs, this makes the cost function has multiple local minima. Then, finding global optimum becomes more difficult with these few parameters, because these few parameters become an ineffective. Therefore, the original GPSO algorithm is considered as a fundamental technique of PSO algorithm. The following steps explain the mechanism of the GPSO algorithm.

Let us consider a swarm population with m particles searching for a solution in d-dimensional search space, where $m > 1$. The objective of the GPSO algorithm is to minimize the given objective function $f(x)$. Each particle i ($i = 1, 2, \ldots, m$) is associated with two d-dimensional vectors; a velocity vector V_i and a position vector X_i.

Initialization: Iteration, $t = 0$.

Step 1: Initialize the velocity and position vectors randomly for particle i, ($i = 1, 2, \ldots, m$).

$$V_i(0) = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{17}$$

$$X_i(0) = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{18}$$

Step 2: For each particle i, evaluate the objective function $f(x)$ using the position vector X_i (0).

Step 3: Initialize the personal position vector of particle i, $G_{pers,i}(0)$ as follows:

$$G_{pers,i}(0) = [g_{pi,1}, g_{pi,2}, \ldots, g_{pi,d}] = X_i(0) \tag{19}$$

Step 4: Determine the global best position vector, $G_{best}(0)$. It is the best position vector among all personal positions vectors in the swarm. The $G_{best}(0)$ is given by

$$G_{best}(0) = [g_{b,1}, g_{b,2}, \ldots, g_{b,d}] \tag{20}$$

Update: Iteration, $t = 1, 2, \ldots, N_{iter}$, where N_{iter} is the total number of iterations.

Step 5: In iteration t, the particle's velocity and position vectors are updated as follows:

$$\begin{aligned} V_i(t) &= V_i(t-1) + c_1 r_1(t)(G_{pers,i}(t-1) - X_i(t-1)) \\ &\quad + c_2 r_2(t)(G_{best}(t-1) - X_i(t-1)) \end{aligned} \tag{21}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{22}$$

where c_1 and c_2 are two positive acceleration coefficients whose values are chosen by using trail and error. The range of c_1 and c_2 are of [2.00,2.25]. They are adapting controlled based on the evolutionary states [64–66]. However, in most cases, $c_1 = c_2 = 2.00$. The $r_1(t)$ and $r_2(t)$ are two randomly generated values within the range (0,1).

Step 6: Each particle i, the objective function $f(x)$ is evaluated using the position vector X_i (t).

Step 7: The $G_{pers,i}$ and G_{best} are updated as follows:

$$G_{pers,i}(t) = \left\{ \begin{array}{ll} X_i(t) & iff(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & \text{Otherwise} \end{array} \right\} \tag{23}$$

$$G_{best}(t) = min\{G_{pers,i}(t)\} \tag{24}$$

End of iteration, $t = N_{iter}$

Step 8: The global best position vector $G_{best}(t)$ becomes the optimal solution and the $f(G_{best}(t))$ gives the optimal value of the objective function. A flowchart of the GPSO algorithm is shown in Fig. 2.

4. The OPSO algorithm

Here, the details of the proposed OPSO algorithm and explanation of the diagonalization process are provided.

4.1. Orthogonal diagonalization process

The matrix diagonalization is the process of converting a square matrix, B of size $(d \times d)$, into a diagonal matrix, D of size $(d \times d)$, as shown below [67].

$$B = QDQ^{-1} \tag{25}$$

where Q is a matrix of size $(d \times d)$ composed of eigenvectors of B and the diagonal elements of D contains the corresponding eigenvalues. The Q is an invertible matrix because it contains linearly independent vectors. When B is symmetric, the (25) may be written as

$$B = CDC^{-1} \tag{26}$$

in which the columns of matrix C are orthonormal to each other. The (26) can be rewritten as

$$D = C^{-1}BC \tag{27}$$

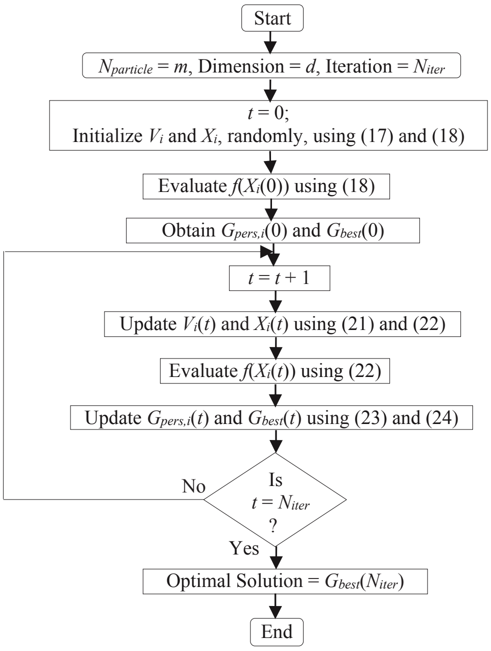

Start
$N_{particle} = m$, Dimension = d, Iteration = N_{iter}
$t = 0$;
Initialize V_i and X_i, randomly, using (17) and (18)
Evaluate $f(X_i(0))$ using (18)
Obtain $G_{pers,i}(0)$ and $G_{best}(0)$
$t = t + 1$
Update $V_i(t)$ and $X_i(t)$ using (21) and (22)
Evaluate $f(X_i(t))$ using (22)
Update $G_{pers,i}(t)$ and $G_{best}(t)$ using (23) and (24)
No
Is $t = N_{iter}$?
Yes
Optimal Solution = $G_{best}(N_{iter})$
End

Fig. 2. Flowchart of the GPSO algorithm.

Since matrix C is an orthonormal matrix, the (27) can be written as

$$D = C^{T}BC \tag{28}$$

Eq. (28) is called orthogonal diagonalization (OD). The process of OD is shown in Fig. 3.

4.2. OPSO learning algorithm

In this paper, an orthogonal PSO (OPSO) algorithm is proposed to improve the performance of the GPSO algorithm. The objective of the OPSO algorithm is to minimize the given d-dimensional objective function $f(x)$. Consider a swarm population with m particles, each with a dimension of d $(d \le m)$. The OPSO algorithm provides a new topology to the swarm population. In each iteration, the swarm population of m particles are divided into two groups: an active group that consists of best personal experience of d particles and one passive group which consists of the personal experience of rest $(m - d)$ particles. The opinion of the active group particles are honoured by updating their respective velocity and position vectors. The opinion of the passive group particles are ignored because their guidance may be erratic, and therefore, their velocity and position vectors are not updated. In each iteration, the OD process (28) is applied. The matrix B is obtained from the d best particles of the active group and thereafter, orthonormal matrix C and diagonal matrix D are computed using (28). The steps involved in OPSO algorithm are given below.

Let a d-dimensional objective function $f(x)$ need to be optimized.

Initialization: Iteration, $t = 0$.

Step 1: Randomly initialize the velocity $V_i(0)$ and position $X_i(0)$ vectors for particle i, $(i = 1, 2, \ldots, m)$.

Step 2: Determine the personal position vectors, $G_{pers,i}(0)$ using (19).

Step 3: Evaluate the objective function $f(x)$ using position vector $X_i(0)$.

```
L1: Let $B$ be a real symmetric matrix of size $d×d$.
L2: Apply Gram-Schmidt orthogonalization on matrix $B$ to obtain $d$ orthonormal vectors [67].
L3: Construct orthonormal matrix $C$ using these vectors.
L4: Obtain the diagonal matrix $D$ using (28).
```

Fig. 3. The orthogonal diagonalization process.

```
Procedure: Convert matrix $A(m×d)$ to a symmetric matrix $B$ $(d×d)$.

for $i = 1 : d$
    $B(1, i) = A(1, i)$
    $B(i, 1) = A(1, i)$
end for
for $k = 2 : d$
    for $i = 2 : d$
        $B(k, i) = A(k, i)$
        $B1(k, i) = B(k, i)$
        $B(i, k) = B1(k, i)$
    end for
end for
```

Fig. 4. Pseudocode for converting matrix A to a symmetric matrix B.

OD process:

In iteration t, $t = 1, 2, \ldots, N_{iter}$, where N_{iter} is the total number of iterations.

Step 4: Arrange the m personal position vectors in an ascending order based on their $f(x)$ values.

Step 5: Construct matrix A of size $(m \times d)$ such that each row occupies one of the m personal position vectors in the same ordered sequence as in step 4.

Step 6: Using pseudocode given in Fig. 4, convert matrix A to a symmetric matrix B of size $(d \times d)$, such that B is a real symmetric matrix of dimension $(d \times d)$.

Step 7: Apply the OD process shown in Fig. 3 on matrix B to obtain the diagonal matrix D. Let D_i denote the ith row of matrix D, where $i = 1, 2, \ldots, d$.

Update:

Step 8: Update the position and velocity vectors of the active group particles, $i = 1, 2, ., d$, as follows.

$$V_i(t) = V_i(t-1) + cr(t)[D_i(t) - X_i(t-1)] \quad (29)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (30)$$

where c is an acceleration coefficient and is chosen by trial and error in the range [2.00,2.25] and $r(t)$ is a random number within range (0,1).

Step 9: Determine the $G_{pers,i}(t)$, $i = 1, 2, \ldots, m$, as follows.

$$G_{pers,i}(t) = \left\{ \begin{array}{ll} X_i(t) & if\, f(X_i(t)) \le f\left(G_{pers,i}(t-1)\right) \\ G_{pers,i}(t-1) & \text{Otherwise} \end{array} \right\} \quad (31)$$

Step 10: Determine the global best position, $G_{best}(t)$.

$$G_{best}(t) = min\{G_{pers,i}(t)\}, i = 1, 2, \ldots, m \quad (32)$$

End of iteration, $t = N_{iter}$

Step 11: The $G_{best}(N_{iter})$ as computed in step 10 provides the optimal solution. A flowchart of the OPSO algorithm is shown in Fig. 5.

4.2.1. Observation 1

One of the important observation of the OPSO algorithm is as follows. Sine D is a diagonal matrix, the d rows or columns of matrix D are orthogonal vectors. These d vectors are used to diminish the contribution of $X_i(t-1)$ while updating $V_i(t)$, $i = 1, 2, \ldots, d$. As $t \to \infty$, assume that the algorithm has converged. In such case, (30) can be written as:

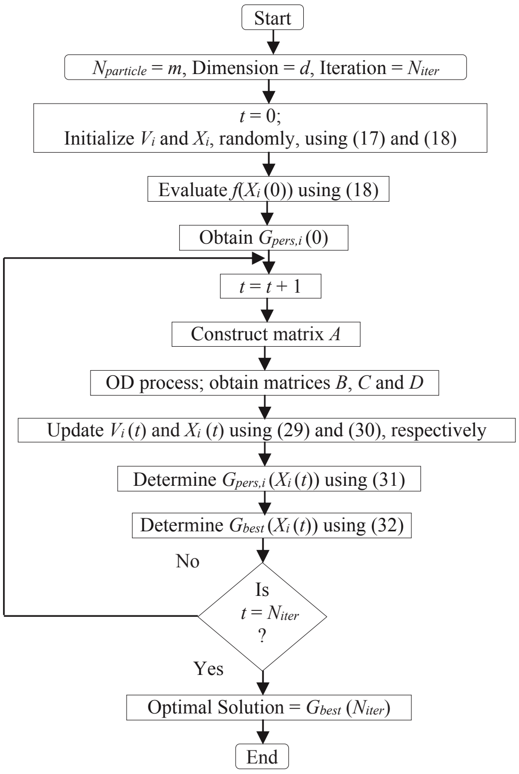

Start
$N_{particle} = m$, Dimension = d, Iteration = N_{iter}
$t = 0$;
Initialize V_i and X_i, randomly, using (17) and (18)
Evaluate $f(X_i(0))$ using (18)
Obtain $G_{pers,i}(0)$
$t = t + 1$
Construct matrix A
OD process; obtain matrices B, C and D
Update $V_i(t)$ and $X_i(t)$ using (29) and (30), respectively
Determine $G_{pers,i}(X_i(t))$ using (31)
Determine $G_{best}(X_i(t))$ using (32)
No
Is $t = N_{iter}$?
Yes
Optimal Solution = $G_{best}(N_{iter})$
End

Fig. 5. Flowchart of the OPSO algorithm.

$$\lim_{t\to\infty} X_i(t) = X_i(t-1) \quad (33)$$

This implies that $\lim_{t\to\infty} V_i(t) = 0$. Then, (29) can be written as:

$$\lim_{t\to\infty} V_i(t) = V_i(t-1) = 0 \quad (34)$$

which implies that

$$\lim_{t\to\infty} c\, r(t)\, [D_i(t) - X_i(t-1)] = 0 \quad (35)$$

Since $c\, r(t)$ is constant,

$$\lim_{t\to\infty} X_i(t-1) = D_i(t) \quad (36)$$

From (36) it is evident that $\lim_{t\to\infty} X_i(t)$ becomes diagonal and equals to D_i when iteration becomes large and the algorithm has converged.

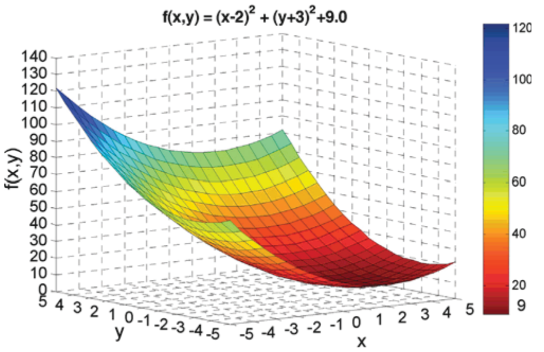


Fig. 6. Fitness landscape of f(x,y). The minimum value of the function is 9.0 at x = 2.0 and y = −3.0.

Considering the d positions vectors, (36) can be written in matrix form as:

$$[X(t)]_{\text{active\_group}} = [D(t)]_{\text{active\_group}} \tag{37}$$

This means, at $t \to \infty$, the d position vectors of the active group particles are equal to the d orthogonal vectors in matrix D. Thus, the OPSO algorithm reaches convergence and leads to optimal solution.

4.2.2. Observation 2

In case of GPSO algorithm (21), two guides, $G_{pers,i}$ and G_{best}, are used to update the velocity vector $V_i(t)$. These two guides may conflict each other which leads to zigzag behaviour of the algorithm, that in turn causes trapping into a local minima. Whereas, in case of OPSO algorithm (29), as only one guide, $D_i(t)$ is used in updating of the velocity vector, $V_i(t)$, such situation is eliminated.

4.2.3. Observation 3

Due to the orthogonalization of the position vectors, as iteration progresses, once the optimal solution is achieved, the solution remains the same in the subsequent iterations until the end of the total number of iterations. This fact provides stability to the proposed algorithm.

4.2.4. Observation 4

From (29), the velocity vector of each particle of active group can be rewritten as follows.

$$\begin{aligned} V_1(t) &= V_1(t-1) + cr(t)[D_1(t) - X_1(t-1)] \\ V_2(t) &= V_2(t-1) + cr(t)[D_2(t) - X_2(t-1)] \\ &\vdots \\ V_d(t) &= V_d(t-1) + cr(t)[D_d(t) - X_d(t-1)] \end{aligned} \tag{38}$$

where
$$\begin{aligned} D_1(t) &= [d_{11}, \quad 0, \quad 0, \cdots, 0] \\ D_2(t) &= [0, \quad d_{22}, 0, \cdots, 0] \\ &\vdots \quad \vdots \; \vdots \quad \vdots \; \vdots \\ D_d(t) &= [0, \quad 0, \quad 0, \; d_{dd}] \end{aligned} \tag{39}$$

It can be seen by cooperation (38) and (39), the position vector X_i, $i = 1, 2, \ldots, d$, of active group is affected by only one orthogonal vector D_i, $i = 1, 2, \ldots, d$. Thus, while updating, each V_i, $i = 1, 2, \ldots, d$, is perturbed only by d_{ii} in the d-dimensional search space. Due to this, the OPSO algorithm gives faster convergence and better solution.

$t = 1$

$$D = \begin{bmatrix} \mathbf{-33.4275} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{9.9113} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{55.4176} & \mathbf{-31.9043} & -80.94 & -23.13 & -6.20 & 79.95 \\ \mathbf{38.2285} & \mathbf{-4.3210} & 23.47 & 23.95 & -1.65 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{-6.2957} & \mathbf{10.2996} & -23.13 & 55.41 & -80.94 & 85.23 \\ \mathbf{-1.6169} & \mathbf{23.4768} & 23.95 & 38.22 & 23.47 & -40.30 \end{bmatrix}$$

$t = 80$

$$D = \begin{bmatrix} \mathbf{-3.7383} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.3275} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-28.0374} & \mathbf{6.7003} & 43.83 & -92.54 & 96.11 & -99.44 \\ \mathbf{-14.4267} & \mathbf{16.8004} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0074} & \mathbf{1.4628} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-2.7485} & \mathbf{-2.7543} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$t = 140$

$$D = \begin{bmatrix} \mathbf{-4.4026} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4027} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4068} & \mathbf{0.0001} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{-0.0002} & \mathbf{3.4019} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{1.9953} & \mathbf{1.9962} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0007} & \mathbf{-2.9972} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$t = 200$

$$D = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} \\ \mathbf{0.0000} & \mathbf{3.4051} \end{bmatrix}$$

$$X = \begin{bmatrix} \mathbf{-4.4051} & \mathbf{0.0000} & 99.98 & -92.54 & 96.11 & -99.44 \\ \mathbf{0.0000} & \mathbf{3.4051} & 94.80 & 63.09 & -90.95 & 49.64 \end{bmatrix}$$

$$G_{pers} = \begin{bmatrix} \mathbf{2.0000} & \mathbf{2.0000} & -6.20 & -23.13 & 0.81 & -80.94 \\ \mathbf{-3.0000} & \mathbf{-3.0000} & -1.65 & 23.95 & 49.64 & 23.47 \end{bmatrix}$$

$$\underbrace{\textbf{\textit{Active}}\quad\textbf{\textit{Group}}}\qquad\underbrace{\textit{Passive}\quad\textit{Group}}$$

Fig. 7. A numerical example showing convergence of the OPSO algorithm.

4.2.5. Observation 5

When $m = d$, there is no existence of passive group and therefore we do not see any advantages of diversity and the solution may not yield the best. When $m \gg d$, it gives rise to more computation, but does not provides any better solution, as seen from sensitivity analysis (Section 6.6). Considering these two extremes a reasonable value of m is could be about 10% more than d.

4.2.6. An illustrative example

In order to explain the mechanism of OPSO algorithm, Fig. 6 illustrates an example of a 2-dimensional shifted function, $f(x,y) = (x-2)^2 + (y+3)^2 + 9$. From visual inspection, it can be seen that the x and y are shifted from the origin [0,0] by [2.0,–3.0]. The optimum solution of the given function equals to 9.0 at [x,y] = [2.0,–3.0]. The aim of the algorithm is to find the values x and y such that the $f(x,y)$ is minimized.

The OPSO algorithm program is implemented using MATLAB software in a personal computer with the following specifications: Intel (R) core (TM) 2 Duo CPU T6570 @ 2.1 GHz. RAM of 4GB and

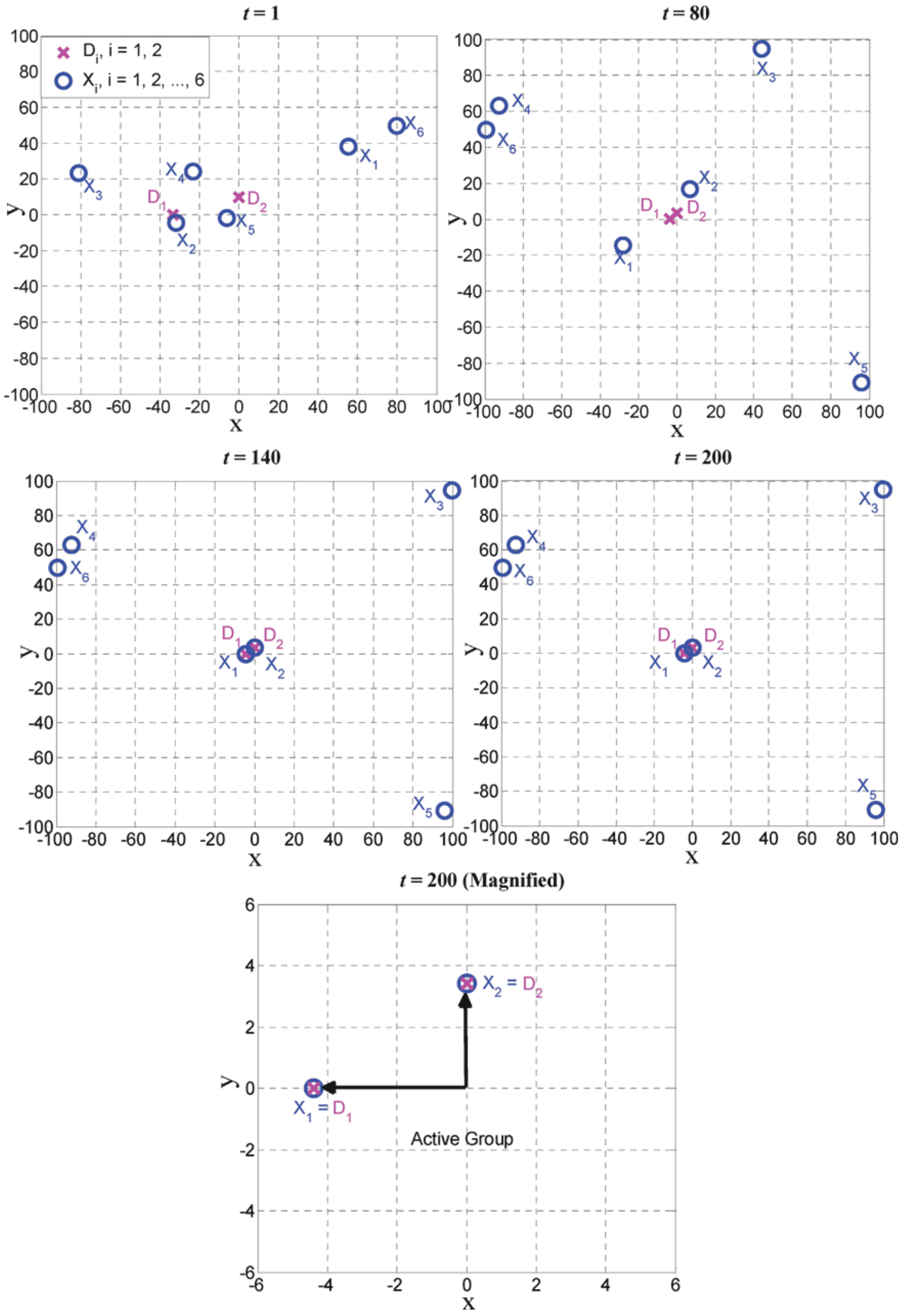


Fig. 8. Movement of positions X_i, $i = 1, 2, \ldots, 6$ of six particles based on orthogonal vectors D_1 and D_2, two active group particles (1 and 2) and four passive group particles (3–6) searching for global solution [2.0,–3.0].

Windows 7, 64-bit operating system. The OPSO algorithm is executed with $m = 6$ for $N_{iter} = 200$. The values of position vectors (X_i, $i = 1, 2, \ldots, 6$), the diagonal vectors (D_i, $i = 1, 2$) and personal vectors ($G_{pers,i}$, $i = 1, 2, \ldots, 6$) at iteration $t = 1$, 80, 140 and 200 are shown in Fig. 7. The six particles are divided into an active group of two best particles and a passive group of remaining four particles. Accord-

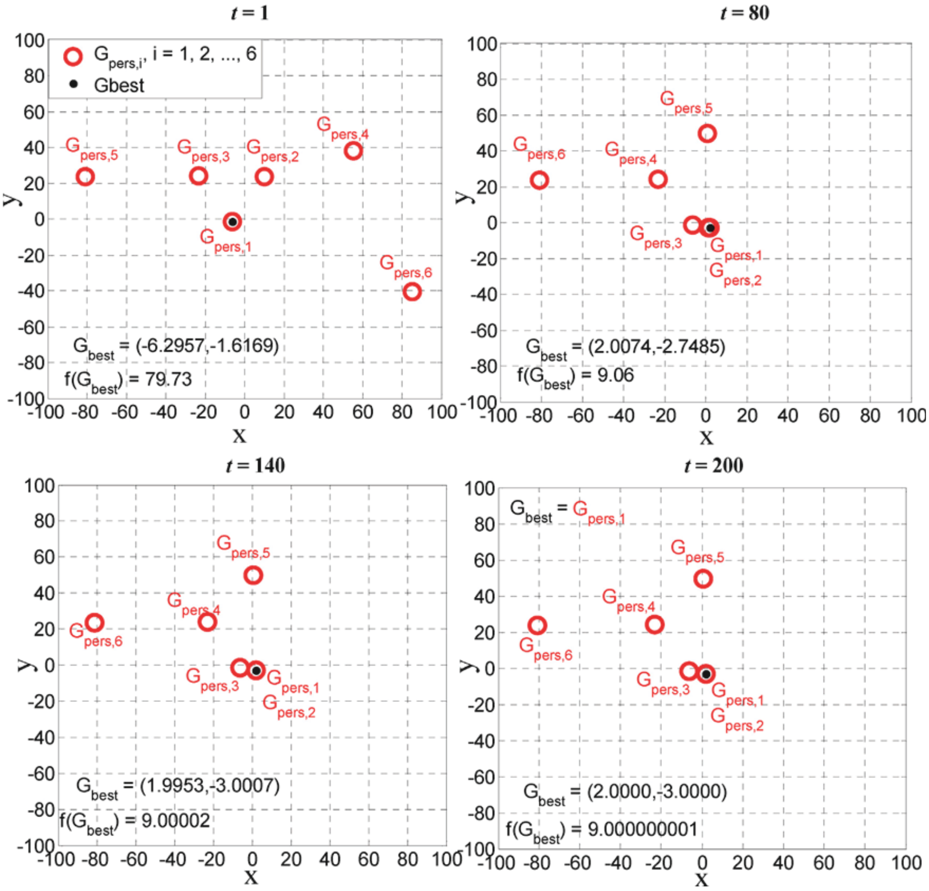


Fig. 9. Movement of the personal position vectors of six particles $G_{pers,i}$, i = 1, 2, . . ., 6 and global best position vector G_{best}, two active group particles (1 and 2) and four passive group particles (3, 4, 5 and 6) searching for global solution [2.0, −3.0].

ing to the OD process, $G_{pers,1}$ and $G_{pers,2}$ are assigned to the active group and ($G_{pers,3}$, . . ., $G_{pers,6}$) are assigned to the passive group. In each iteration, the velocity and position vectors of only the active group are updated. As seen from Fig. 7, as iteration reaches 200, the OD process causes $[X]_{\text{active\_group}} = [D]_{\text{active\_group}}$, satisfying (37) and causing X to be a diagonal matrix. At the end of iteration, the best G_{pers}, provides the optimal solution i.e., $G_{best} = [2.0, -3.0]$.

Fig. 8 geometrically explain the movement of the six particles of the swarm as iteration progress from 1 to 200. At first iteration, the X_i vectors are at random positions in search space range of ± 100. The vectors D_1 and D_2 are orthogonal to each other. As the iteration increases, the active group position vectors (X_1 and X_2) moves towards the orthogonal vectors (D_1 and D_2) and at the end of iteration, $X_1 = D_1$ and $X_2 = D_2$. This can be seen from bottom subfigure of Fig. 8 which is the magnified view at $t = 200$. This signifies achievements of orthogonalization of the active group position vectors. Fig. 9 shows the movement of the personal position vectors of six particles $G_{pers,i}$, $i = 1, 2, \ldots, 6$ and global best position vector G_{best}, two active group particles (1 and 2) and four passive group particles (3–6) searching for global solution [2.0, −3.0]. The best personal vector, i.e., G_{best} of active group gives the optimal solution as shown in Fig. 9.

5. Application of OPSO algorithm to ED problem

Here we describe the simulation results carried out on three power systems with several TGUs and SPG constraints.

5.1. Performance measures

To study the accuracy, stability and robustness of different algorithms, several fitness values as explained below are considered. Every algorithm is executed over N_{run} runs each with N_{iter} iterations.

1 Minimum fuel cost ($F_{\min}$): Defined as the minimum of the optimized F_{cost} values obtained from N_{run} independent runs.
2 Maximum fuel cost ($F_{\max}$): Defined as the maximum of the optimized F_{cost} values obtained from N_{run} independent runs.
3 Mean fuel cost (F_{mean}): Defined as the average of the optimized F_{cost} values obtained from N_{run} independent runs.
4 Standard deviation (σ): The σ is the standard deviation of the optimized F_{cost} values obtained from N_{run} independent runs.
5 Range (R): The range (R) is defined as the difference between $F_{\max}$ and $F_{\min}$.

Table 1
Specifications and power constraints for PS-1.

TGU	P_j^0 (MW)	$P_{j,min}$ (MW)	$P_{j,max}$ (MW)	a_i ($)	b_i ($/MW)	c_i ($/MW^2)	UR_j (MW/h)	DR_j (MW/h)	POZs (MW)
1	440	100	500	240	7.0	0.0070	80	120	[210,240] [350,380]
2	170	50	200	200	10.0	0.0095	50	90	[90,110] [140,160]
3	200	80	300	220	8.5	0.0090	65	100	[150,170] [210,240]
4	150	50	150	200	11.0	0.0090	50	90	[80,90] [110,120]
5	190	50	220	220	10.5	0.0080	50	90	[90,110] [140,150]
6	110	50	120	190	12.0	0.0075	50	90	[75,85] [100,105]

Table 2
B-loss coefficients of 6 TGUs of PS-1.

	B_{jk}	1	2	3	4	5	6
B_{jk}=1 × 10^{-06} MW^{-1}	*1*	17	12	7	−1	−5	−2
	2	12	14	9	1	−6	−1
	3	7	9	31	0	−10	−6
	4	−1	1	0	24	−6	−8
	5	−5	−6	−10	−6	129	−2
	6	−2	−1	−6	−8	−2	150

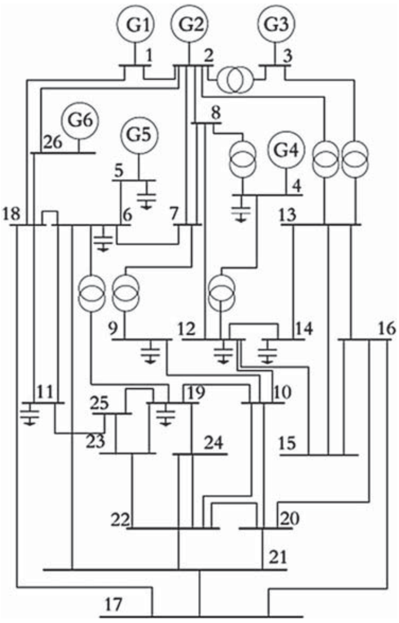


Fig. 10. Single line diagram of PS-1.

6 Average execution time (AET): It is the time consumed by an algorithm after convergence, averaged over N_{run} independent runs.

5.2. Test case 1: power system-1 (PS-1)

The PS-1, as shown in Fig. 10, is a small-scale power system with six TGUs (N_{gen} = 6) and 26 buses [25]. At steady state normal operation, the maximum load demand is given as P_D = 1263 MW. The specifications of PS-1, i.e., initial output power, generation limits, cost coefficients, RRLs and 12 POZs are given in Table 1 and the B-loss coefficients are given in Table 2.

5.2.1. Comparison in terms of fitness values

In [25], the authors have shown the superior performance of their proposed RDPSO algorithm over other 10 ECTs for the ED problem using PS-1. In addition, PS-1 was also used by other researchers for performance comparison, e.g., the SOH-PSO [3], SA-PSO [26], NPSO-LRS [27], CPSO [29], QMPSO [30], FDA [38], GA-API [40], MIQCQP [41], λ-logic [43], SPPO [46]. We compared the performance of our proposed OPSO algorithm with these 21 ECTs and GPSO algorithm. The comparison results of fitness values are shown in Table 3. The parameters used OPSO and GPSO algorithms are: m = 10, N_{run} = 100, N_{iter} = 1000, $c_1 = c_2 = c$ = 2.05. Here, "NA" stands for not available. One can observe the following from this Table. Firstly, the OPSO algorithm provides best result in terms of lowest F_{mean} and lowest σ. This shows that the OPSO algorithm provides stable and accurate solution. Secondly, in terms of range, R, OPSO provides the second best result, i.e., R = $0.3276/h (the best is from GA-API algorithm [40], R = $0.0300/h). This indicates that OPSO algorithm has second lowest dispersion of optimum F_{cost}. However, the $F_{\min}$ and $F_{\max}$ of OPSO algorithm are much better than $F_{\min}$ and $F_{\max}$ for GA-API algorithm [40]. In term of AET, the MIQCQP [41] is best one. The GPSO algorithm is second best but it provides worst results in terms of other performance measures. The OPSO achieves the third best in terms of AET. Thus, the overall performance of the OPSO algorithm is far superior than the other 22 ECTs.

5.2.2. Convergence characteristics of OPSO and GPSO algorithms

Fig. 11 shows the convergence characteristics of OPSO and GPSO algorithms for PS-1. It shows ensemble average F_{cost} values obtained from 100 independent runs at each iteration. It can be seen that OPSO algorithm settles approximately at 40 iterations and achieved F_{mean} = $15,443.5921/h whereas GPSO algorithm takes about 190 iterations to converge and achieved F_{mean} = $15,460.8461/h. This shows faster convergence of OPSO algorithm compared with the GPSO algorithm.

Fig. 12 shows the comparison of optimized F_{cost} at each run between the OPSO and GPSO algorithms. In case of OPSO algorithm, the optimized F_{cost} after each run remains more or less steady at about $15,443/h, whereas in GPSO algorithm, the optimized F_{cost} varies between $15,442.8326/h and $16,103.3400/h. This indicates that OPSO algorithm is more consistent and stable than GPSO algorithm.

5.2.3. Comparison in terms of inequality constraint

Table 4 lists the solution vector, P_j (j = 1, 2, . . ., 6) corresponding to the best solution for OPSO and GPSO algorithms. The results in Table 4 observe that OPSO and GPSO algorithms avoid the 12 POZs of 6 TGUs and are within RRLs (16). This indicates that both algorithms are able to satisfy the inequality constraints of PS-1.

Table 3
Comparison of cost performance between OPSO and other 22 ECTs for PS-1.

Sl.No.	Algorithm	F_{min} ($/h)	F_{max} ($/h)	F_{mean} ($/h)	σ ($/h)	*R* ($/h)	AET (sec)
1	SOH-PSO [3]	15,446.0200	15,609.6400	15,497.3500	NA	136.6200	8.8620
2	GA [25]	15,445.5961	15,491.4797	15,465.1757	9.7336	45.8836	NA
3	DE [25]	15,444.9466	15,472.0651	15,450.1339	6.9854	27.1185	NA
4	ACSA [25]	15,445.3052	15,511.5269	15,459.5170	12.0247	66.2217	NA
5	AIS [25]	15,446.3283	15,481.2766	15,456.6660	7.3954	34.9483	NA
6	FA [25]	15,445.9448	15,501.3958	15,461.3003	9.3385	55.4510	NA
7	BCO [25]	15,444.5837	15,482.3963	15,457.9441	8.4816	37.8126	NA
8	APSO [25]	15,445.5109	15,538.6016	15,473.3164	12.9048	93.0907	NA
9	HGPSO [25]	15,447.1055	15,497.0335	15,462.6151	10.6456	49.9280	NA
10	HPSOM [25]	15,443.6281	15,479.8640	15,449.2603	6.2745	36.2359	NA
11	HPSOWM [25]	15,442.8205	15,502.6333	15,455.6220	15.8867	59.8128	NA
12	RDPSO [25]	15,442.7575	15,455.2936	15,445.0245	2.2828	12.5361	NA
13	SA-PSO [26]	15,447.0000	15,455.0000	15,447.0000	2.5280	8.0000	7.5800
14	NPSO-LRS [27]	15,450.0000	15,452.0000	15,450.5000	NA	2.0000	NA
15	CPSO [29]	15,446.0000	15,490.0000	15,449.0000	NA	44.0000	8.1300
16	QMPSO [30]	15,457.3380	15,489.9270	15,472.1840	4.5270	32.5890	7.1000
17	FDA [38]	15,449.5826	15,449.6508	15,449.6171	NA	0.0682	3.6340
18	GA-API [40]	15,449.7800	15,449.8100	15,449.8500	NA	**0.0300**	NA
19	MIQCQP [41]	15,443.0700	NA	NA	NA	NA	**0.4500**
20	λ-logic [43]	15,449.7960	NA	NA	NA	NA	NA
21	SPPO [46]	15,450. 0000	NA	NA	NA	NA	NA
22	GPSO	15,442.8334	16,103.3400	15,458.4000	69.6797	660.5074	2.5211
23	**OPSO**	15,442.8720	15,443.1996	**15,443.9754**	**0.2116**	0.3276	2.6334

Bold values signifies the best results in the respective category.

Table 4
Optimized output power for each TGU obtained by OPSO and GPSO algorithms for PS-1.

Algorithm	Optimum output power (MW)						Total output power (MW)
	P_1	P_2	P_3	P_4	P_5	P_6	
GPSO	450.5975	172.5329	261.3800	140.0706	164.4095	86.3979	1,275.3884
OPSO	450.0003	174.1009	261.9800	140.0674	162.3809	86.8386	1,275.3681

Bold values signifies the best results in the respective category.

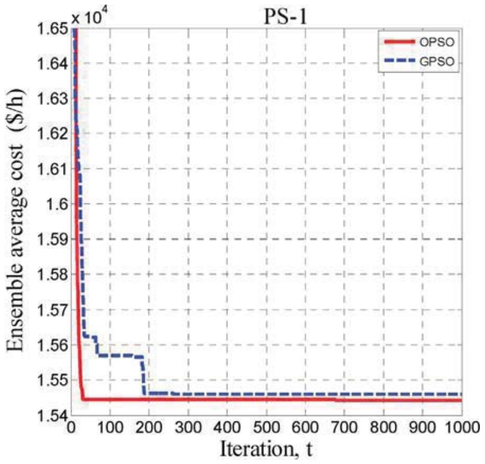


Fig. 11. Convergence characteristics of OPSO and GPSO algorithms for PS-1.

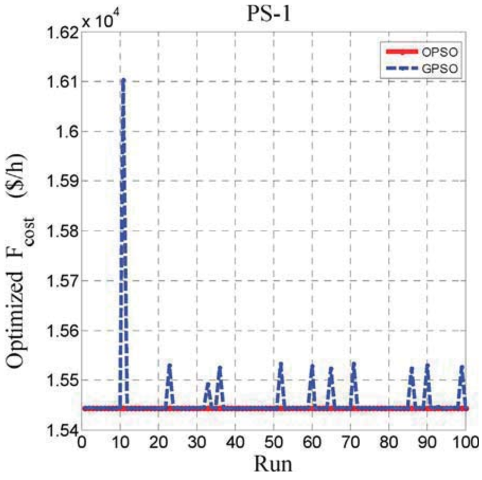


Fig. 12. Comparison of optimized cost per run between OPSO and GPSO algorithms for PS-1.

5.2.4. Comparison in terms of power balance constraint

Table 5 shows the results of power balance constraint based on $P_{L,mismatch}$ for 11 ECTs. The load demand of PS-1 is given as $P_D = 1263$ MW. Using the total optimum output power generated (from Table 4) and Eqs. (5)–(7), P_{L1}, P_{L2} and $P_{L,mismatch}$ are computed and the results presented in Table 5. It can be seen that OPSO as well as SOH-PSO [3], GA-API [40], MIQCQP [41], λ-logic [43] and SPPO [46] algorithms provide zero mismatch, i.e., $P_{L,mismatch} = 0$, indicating that the power balance constraint, that is Eq. (3) is satisfied.

5.3. Test case 2: power system-2 (PS-2)

The PS-2 is a medium-scale power system [29] with 15 TGUs ($N_{gen} = 15$) whose generation specifications and *B*-loss coefficients are shown in Tables 6 and 7, respectively. The maximum load demand at steady state normal operation is given as $P_D = 2630$ MW. The PS-2 has 11 POZs in 4 TGUs (2, 5, 6 and 12) and RRLs for each TGU.

Table 5
Comparison of power balance constraint among 11 ECTs for PS-1.

Sl. No.	Algorithm	Total P_j (MW)	P_D (MW)	P_{L1} (MW)	P_{L2} (MW)	$P_{L,\text{mismatch}}$ (MW)
1	SOH-PSO [3]	1,275.5500	1263	12.5500	12.5500	**0.0000**
2	DRPSO [25]	1,275.3565	1263	12.3565	12.3598	0.0033
3	NPSO-LRS [27]	1,275.9400	1263	12.9400	12.9361	0.0039
4	CPSO [29]	1,276.0000	1263	13.0000	12.9582	−0.0418
5	QMPSO [30]	1,275.3159	1263	12.3159	12.4058	0.0899
6	GA-API [40]	1,276.1300	1263	13.1300	13.1300	**0.0000**
7	MIQCQP [41]	1,275.4400	1263	12.4400	12.4400	**0.0000**
8	λ-logic [43]	1,275.9500	1263	12.9500	12.9500	**0.0000**
9	SPPO [46]	1,275.9700	1263	12.9700	12.9700	**0.0000**
10	GPSO	1,275.3884	1263	12.3884	12.3883	−0.0001
11	**OPSO**	1,275.3681	1263	12.3681	12.3681	**0.0000**

Bold values signifies the best results in the respective category.

Table 6
Specifications and power constraints for PS-2.

TGU	P_j^0 (MW)	$P_{j,}min$ (MW)	$P_{j,}max$ (MW)	a_i ($)	b_i ($/MW)	c_i ($/MW$^{2)}$	UR_j (MW/h)	DR_j (MW/h)	POZs (MW)
1	400	150	455	671	10.1	0.000299	80	120	–
2	300	150	455	574	10.2	0.000183	80	120	[185,225] [305,335,450,450]
3	105	20	130	374	8.8	0.001126	130	130	–
4	100	20	130	374	8.8	0.001126	100	130	–
5	90	150	470	461	10.4	0.000205	80	120	[180,200] [305,335] 390,420]
6	400	135	460	630	10.1	0.000301	80	120	[230,225] [365,395] [430,455]
7	350	135	465	548	9.8	0.000364	80	120	–
8	95	60	300	227	11.2	0.000338	65	100	–
9	105	25	162	173	11.2	0.000807	60	100	–
10	110	25	160	175	10.7	0.001203	60	100	–
11	60	20	80	186	10.2	0.003586	80	80	–
12	40	20	80	230	9.9	0.005513	80	80	[30,40] [55,65]
13	30	25	85	225	13.1	0.000371	80	80	–
14	30	15	55	309	12.1	0.001929	55	55	–
15	20	15	55	323	12.4	0.004447	55	55	–

Table 7
B-loss coefficients of 15 TGUs of PS-2.

$B_{jk}(\times 10^{-6}\ \text{MW}^{-1})$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	14	12	7	−1	−3	−1	−1	−1	−3	5	−3	−2	4	3	−1
2	12	15	13	0	−5	−2	0	1	−2	−4	−4	−0	4	10	−2
3	7	13	76	−1	−13	−9	−1	0	−8	−12	−17	−0	−26	111	−28
4	−1	0	−1	34	−7	−4	11	50	29	32	−11	−0	1	1	−26
5	−3	5	−13	−7	90	14	−3	−12	−10	−13	7	−2	−2	−24	−3
6	−1	−2	−9	−4	14	16	−0	−6	−5	−8	11	−1	−2	−17	3
7	−1	0	−1	11	−3	−0	15	17	15	9	−5	7	−0	−2	−8
8	−1	1	0	50	−12	−6	17	168	82	79	−23	−36	1	5	−78
9	−3	−2	−8	29	−10	−5	15	82	129	116	−21	−25	7	−12	−72
10	−5	−4	−12	32	−13	−8	9	79	116	200	−27	−34	9	−11	−88
11	−3	−4	−17	−11	7	11	−5	−23	−21	−27	140	1	4	−38	168
12	−2	−0	−0	−0	−2	−1	7	−36	−25	−34	1	54	−1	−4	28
13	4	4	−26	1	−2	−2	−0	1	7	9	4	−1	103	−101	28
14	3	10	111	1	−24	−17	−2	5	−12	−11	−38	−4	−101	578	−94
15	−1	−2	−28	−26	−3	3	−8	−78	−72	−88	168	28	28	−94	1283

5.3.1. Comparison in terms of fitness values

In [25], the RDPSO algorithm was tested with PS-2 and its superior performance compared to other 10 ECTs has been shown. In addition, the PSO-MSAF [22], SA-PSO [26], CPSO [29], EGSSOA [42], λ-logic [43], SPPO [46] algorithms have also been tested with PS-2. Here we compare performance of OPSO algorithm with GPSO and other existing 17 ECTs. The set of parameters used in GPSO and OPSO algorithms are: $m = 18$, $d = 15$, $N_{run} = 100$, $N_{iter} = 1000$ and $c = c_1 = c_2 = 2.05$. Comparison of fitness values between OPSO algorithm and other 18 existing ECTs are listed in Table 8. It can be seen that, the OPSO algorithm provides the best results in terms of F_{mean}, σ and R. These results indicate that the OPSO algorithm provides consistent, stable and optimal results. However, in term of AET, OPSO is the second best; the GPSO algorithm being the best among the 19 ECTs.

5.3.2. Convergence characteristics of OPSO and GPSO algorithms

Fig. 13 shows the convergence characteristics of OPSO and GPSO algorithms for PS-2. It shows ensemble average F_{cost} values at each iteration obtained from 100 independent runs. It can be seen that OPSO algorithm settles at about 60 iterations to achieve F_{mean} = $32,670/h whereas GPSO algorithm takes about 150 iterations to settle and achieved F_{mean} = $33,185/h.

Fig. 14 shows the distribution of optimized F_{cost} at each run. It shows that the optimized F_{cost} of OPSO remains steady at about $32,669/h, whereas in GPSO algorithm, the optimized F_{cost} varies over a wide range from $32,892/h and $33,851/h. This indicates that OPSO algorithm is more consistent, stable and reliable than the GPSO algorithm.

Table 8
Comparison of cost performance between OPSO and other 18 ECTs for PS-2.

Sl. No.	Algorithm	Min. Cost ($/h)	Max. Cost ($/h)	Mean Cost ($/h)	σ ($/h)	*R* ($/h)	AET (sec)
1	PSO-MSAF [22]	32,713.0900	32,798.2500	32,759.6400	NA	85.1600	19.1500
2	GA [25]	32,939.5208	33,231.6216	33,106.0019	100.1279	292.1008	NA
3	DE [25]	32,818.5792	33,116.9340	32,990.8673	61.5145	298.3548	NA
4	ACSA [25]	32,785.6031	33,185.2761	33,051.7711	77.8005	399.6730	NA
5	AIS [25]	32,895.9173	33,132.0191	33,017.6537	58.1230	236.1018	NA
6	FA [25]	32,901.6610	33,197.2718	33,081.0107	91.0111	295.6108	NA
7	BCO [25]	32,989.2341	33,301.4940	33,113.0149	69.7986	312.2599	NA
8	APSO [25]	32,687.9840	33,359.6609	32,948.0533	92.0040	671.6769	NA
9	HGPSO [25]	32,864.0501	33,280.2655	33,034.1894	63.9932	416.2154	NA
10	HPSOM [25]	32,697.2458	33,015.7284	32,819.5931	83.0907	318.4826	NA
11	HPSOWM [25]	32,696.9585	33,034.3413	32,805.7185	87.8689	337.3828	NA
12	RDPSO [25]	32,666.1818	32,934.3089	32,739.7165	56.7070	268.1271	NA
13	SA-PSO [26]	32,708.0000	32,789.0000	32,732.0000	18.0250	81.0000	12.7900
14	CPSO [29]	32,834.0000	33,318.0000	33,021.0000	NA	484.0000	13.1300
15	EGSSOA [42]	NA	NA	32,680.1038	NA	NA	NA
16	λ-logic [43]	32,713.9510	NA	NA	NA	NA	NA
17	SPPO [46]	32,713.2100	NA	NA	NA	NA	NA
18	GPSO	32,891.8329	33,850.9528	33,137.5549	197.3331	959.1199	**3.5901**
19	**OPSO**	32,668.4863	32,669.3005	**32,668.9205**	**0.1394**	**0.8142**	4.3777

Bold values signifies the best results in the respective category.

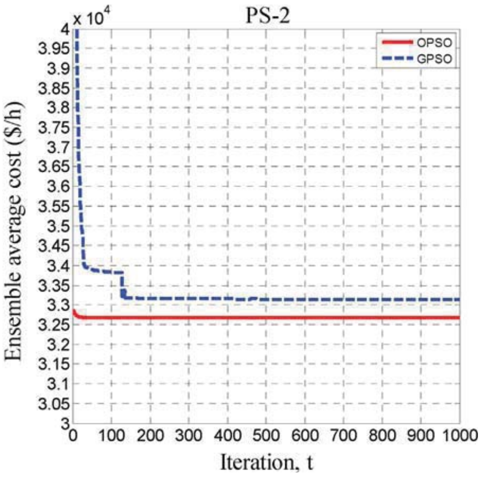


Fig. 13. Convergence characteristics of OPSO and GPSO algorithms for PS-2.

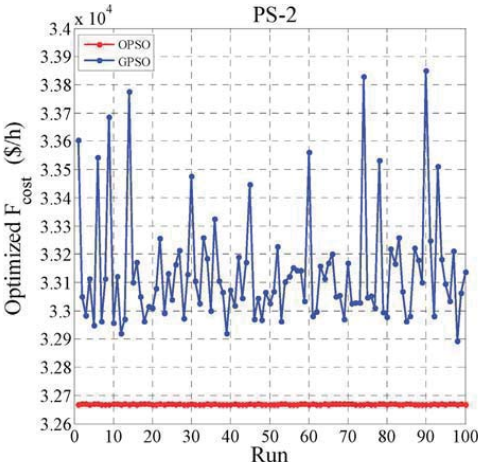


Fig. 14. Comparison of optimized cost per run between OPSO and GPSO algorithms for PS-2.

5.3.3. *Comparison in terms of inequality constraint*

Table 9 presents the solution vector, P_j (j = 1, 2, . . ., 15) corresponding to the best solution for OPSO and GPSO algorithms. It can be seen that both the OPSO and GPSO algorithms are able to avoid the 11 POZs of 4 TGUs and are within RRL constraints, thus both algorithms are able to satisfy the inequality constraints of PS-2.

5.3.4. *Comparison in terms of power balance constraint*

Table 10 shows results of power balance constraints for the OPSO and other 8 ECTs. The load demand of PS-2 is given as P_D = 2630 MW. Using the optimum output power generated given in Table 9 and Eqs. (5)–(7), P_{L1}, P_{L2} and $P_{L,mismatch}$, were determined. It can be seen that the OPSO algorithm as well as PSO-MSAF [22], EGSSOA [42] and λ-logic [43] algorithm are satisfying the zero mismatch condition, i.e., $P_{L,mismatch}$ = 0, thus satisfying (3).

5.4. *Test case 3: power system-3 (PS-3)*

The PS-3 is a large-scale power system taken from Taipower system [9], [25]. It consists of 40 mixed-generating units, coal-fired, gas-fired, gas-turbines with complex cycle, diesel generating

Table 9
Optimized output power for each TGU obtained by OPSO and GPSO algorithms for PS-2.

Optimum output power (MW)		
TGU	GPSO	**OPSO**
1	455.0000	455.0000
2	377.1693	380.0000
3	125.8555	130.0000
4	113.9563	129.9696
5	161.2816	170.0000
6	458.6809	457.3942
7	418.2693	430.0000
8	152.0496	71.6613
9	78.3982	58.2340
10	71.8736	160.0000
11	60.5822	80.0000
12	65.7008	80.0000
13	34.3022	25.0000
14	50.5299	15.0000
15	36.3461	15.0000
Total output power (MW)	2659. 9955	2,657.2591

Bold values signifies the best results in the respective category.

Table 10
Comparison of power balance constraint among 9 ECTs for PS-2.

Sl. No.	Algorithm	Total P_j MW	P_D MW	P_{L1} MW	P_{L2} MW	$P_{L,\text{mismatch}}$ MW
1	PSO-MSAF [22]	2,660.4900	2630	30.4900	30.4900	**0.0000**
2	DRPSO [25]	2,655.3650	2630	25.3650	25.3696	0.0460
3	SA-PSO [26]	2,660.9000	2630	30.9000	30.9080	0.0080
4	CPSO [29]	2,662.1000	2630	32.1000	32.1303	0.0303
5	EGSSOA [42]	2,657.0120	2630	27.0120	27.0120	**0.0000**
6	λ-logic [43]	2,659.9491	2630	29.9491	29.9491	**0.0000**
7	SPPO [46]	2,660.0000	2630	30.0000	31.4300	1.4300
8	GPSO	2,659.9955	2630	29.9950	29.9971	0.0016
9	**OPSO**	2,657.2591	2630	27.2591	27.2591	**0.0000**

Bold values signifies the best results in the respective category.

Table 11
Specifications and power constraints for PS-3.

TGU	P_j^0 (MW)	$P_{j,min}$ (MW)	$P_{j,max}$ (MW)	a_i (\$/h)	b_i (\$/MWh)	c_i (\$/MW^{2}h)	UR_j (MW/h)	DR_j (MW/h)	POZs (MW)
1	50	40	80	170.77	8.336	0.03073	35	60	–
2	60	60	120	309.54	7.0706	0.02028	40	70	[80,85]
3	150	80	190	369.03	8.1817	0.00942	50	90	[82,88]
4	24	24	42	135.48	6.9467	0.08482	42	42	–
5	42	26	42	135.19	6.5595	0.09693	42	42	–
6	75	68	140	222.33	8.0543	0.01142	40	75	–
7	100	110	300	287.71	8.0323	0.00357	65	100	[155,162] [221,235]
8	152	135	300	391.98	6.9990	0.00492	65	100	–
9	200	135	300	455.76	6.6020	0.00573	65	100	[235,246]
10	100	130	300	722.82	12.908	0.00605	65	100	[200,211]
11	300	94	375	635.20	12.986	0.00515	55	95	[213,220]
12	300	94	375	654.69	12.796	0.00569	55	95	[213,220]
13	150	125	500	913.40	12.501	0.00421	80	120	[201,211] [290,310] [413,425]
14	200	125	500	1760.4	8.8412	0.00752	80	120	[205,217,306,318,409,420]
15	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230] [277,290] [402,412]
16	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230] [277,290] [402,412]
17	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230] [277,290] [402,412]
18	400	220	500	647.85	7.9691	0.00313	70	110	[307,321] [407,421]
19	400	220	500	649.69	7.9550	0.00313	70	110	[301,310] [421,431]
20	398	242	500	647.83	7.9691	0.00313	70	110	[340,351] [421,431]
21	398	242	500	647.81	7.9691	0.00313	70	110	[340,351] [421,431]
22	390	254	550	785.96	6.6313	0.00298	70	110	[306,320] [440,445]
23	390	254	550	785.96	6.6313	0.00298	70	110	[306,320] [440,445]
24	390	254	550	794.53	6.6311	0.00284	70	110	[370,390] [495,502]
25	390	254	550	794.53	6.6311	0.00284	70	110	[370,390] [495,502]
26	390	254	550	801.32	7.1032	0.00277	70	110	[380,410] [501,520]
27	390	254	550	801.32	7.1032	0.00277	70	110	[380,410] [501,520]
28	20	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
29	20	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
30	30	10	150	1055.1	3.3353	0.52124	90	150	[102,113]
31	30	20	70	1207.8	13.052	0.25098	70	70	–
32	40	20	70	810.79	21.887	0.16766	70	70	–
33	40	20	70	1247.7	10.244	0.2635	70	70	–
34	25	20	70	1219.2	8.3707	0.30575	70	70	–
35	25	18	60	641.43	26.258	0.18362	60	60	–
36	20	18	60	1112.8	9.6956	0.32563	60	60	–
37	20	20	60	1044.4	7.1633	0.33722	60	60	–
38	25	25	60	832.24	16.339	0.23915	60	60	–
39	25	25	60	832.24	16.339	0.23915	60	60	–
40	25	25	60	1035.2	16.339	0.23915	60	60	–

units and nuclear generating units. The maximum load demand at steady state normal operation is given as $P_D = 8550$ MW. The PS-3 contains 46 POZs distributed among 25 TGUs and are shown in Table 11. The RRLs are imposed on all the 40 TGUs. The *B*-loss coefficients are considered and they are generated randomly as is done in [43]. The *B*-loss coefficients matrix of dimension 40 × 40 is listed in Appendix. Unfortunately, the PS-3 is tested by only a few authors under RRLs, POZs and P_L constraint. This may be due to unavailability of *B*-loss coefficients or due to its high dimension with a large number of power constraints.

5.4.1. Comparison in terms of fitness values

In [25], PS-3 has been tested with 11 ECTs and superior performance of RDPSO algorithm over other 10 ECTs has been shown. However, the 46 POZs, RRLs of each TGU and the P_L constraint have not been considered. Therefore, these results are less constrained. Considering all the POZs, RRLs and P_L constraint, the PS-3 has been tested by MIQCQP [41], λ-logic [43], the proposed OPSO and GPSO algorithms. Thus, the performance of OPSO algorithm and other existing 14 ECTs are compared. The set parameters used in GPSO and OPSO algorithms are: $m = 45$, $N_{run} = 100$, $N_{iter} = 10{,}000$, $d = 40$, and $c = c_1 = c_2 = 2.05$. The fitness values of the 15 ECTs are listed in Table 12. It can be seen that the OPSO algorithm provides the best result in terms of F_{mean} and σ over 100 independent runs. This indicated that the OPSO algorithm provides the most optimal and consistent results. In addition, the range *R* of OPSO algorithm is the lowest among the 15 ECTs, thus indicating that OPSO algorithm provides solution with the lowest dispersion. Since the AET values are not available for other ECTs, between OPSO and GPSO algorithms, the AET of OPSO (69 s), due to its computational com-

Table 12
Comparison of cost performance between OPSO and other 14 ECTs for PS-3.

Sl. No.	Algorithm	Min.Cost ($/h)	Max. Cost ($/h)	Mean Cost ($/h)	σ ($/h)	R ($/h)	AET (sec)
Without POZs, RRLs and P_L							
1	GA [25]	133,435.6906	136,274.9726	135,012.4985	729.3536	2,839.2820	NA
2	DE [25]	129,915.5635	137,042.9461	130,600.2269	1,335.4343	7,127.3826	NA
3	ACSA [25]	131,167.3417	134,923.6245	132,844.7110	741.0843	3,756.2828	NA
4	AIS [25]	130,133.9214	132,703.1884	131,482.2767	561.7950	2,569.2670	NA
5	FA [25]	130,948.8466	134,997.9243	133,511.4572	747.3692	4,049.0777	NA
6	BCO [25]	130,337.7290	132,999.8803	131,733.9439	589.8034	2,662.1513	NA
7	APSO [25]	130,861.5242	134,044.6303	132,587.8486	675.0344	3,183.1061	NA
8	HGPSO [25]	132,072.2495	135,528.3862	134,012.5706	684.4951	3,456.1367	NA
9	HPSOM [25]	129,177.4413	131,281.3077	130,234.1694	529.5827	2,103.8664	NA
10	HPSOWM [25]	129,717.3557	132,303.5999	130,858.6741	591.7691	2,586.2442	NA
11	RDPSO [25]	128,864.4525	131,129.0861	129,482.0970	568.9333	2.264.6336	NA
With POZs, RRLs and P_L							
12	MIQCQP [41]	128,395.2900	NA	NA	NA	NA	**13.34**
13	λ-logic [43]	129,777.5300	NA	NA	NA	NA	NA
14	GPSO	139,051.1893	515,712.5007	398,122.4784	95,189.4180	376,661.3114	47.32
15	**OPSO**	126,489.6228	127,916.1972	**127,349.8324**	**302.3502**	**1,426.5744**	69.32

Bold values signifies the best results in the respective category.

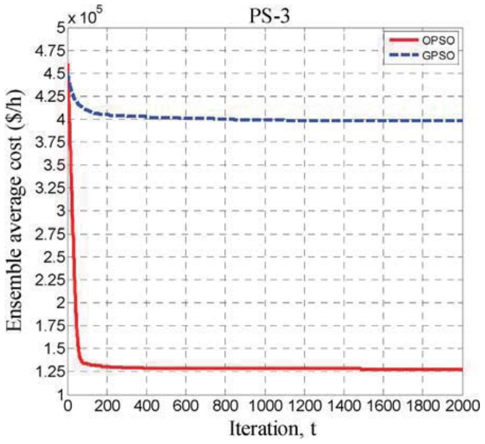


Fig. 15. Convergence characteristics of OPSO and GPSO algorithms for PS-3.

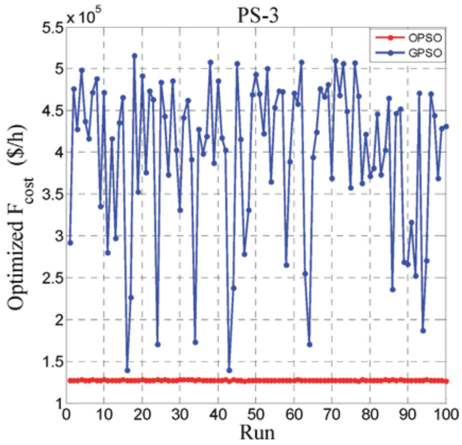


Fig. 16. Comparison of optimized cost per run between OPSO and GPSO algorithms for PS-3.

plexity, is found to be higher than the GPSO (47 s). These results indicate that among the 15 ECTs, the OPSO algorithm is the most stable, robust and is able to provide most optimal solution.

5.4.2. Convergence characteristics of OPSO and GPSO algorithms

Fig. 15 shows the convergence characteristics of OPSO and GPSO algorithms for PS-3. Note that Fig. 14 is drawn with N_{iter} = 2000, to give a better visualization. However, the OPSO and GPSO algorithms run with N_{iter} = 10,000. It shows ensemble average F_{cost} values at each iteration obtained from 100 independent runs. It can be seen that OPSO algorithm settles at about 1600 iterations and achieves F_{mean} of about $127,997/h. Whereas, the GPSO algorithm takes about 1000 iterations to converge, but settles at a local minimum with a non-optimal F_{mean} of about $398,709/h. This indicates that the GPSO algorithm is unable to solve ED problem with such a high dimension and under such large number of power constraints. In contrast, the OPSO algorithm gives high accuracy in solving such this complex problem.

Fig. 16 shows the variation of optimized F_{cost} over 100 independent runs achieved by the OPSO and GPSO algorithms. It shows that the optimized F_{cost} of OPSO varies between $126,489.6/h and $127,916.2/h, whereas in GPSO algorithm, it varies between $139,051.2/h and $515,712.5/h. This indicates that OPSO algorithm is capable of providing consistent and reliable optimal solution. Whereas, the GPSO algorithm is unable to provide optimal solution due to the high complexity of the problem.

5.4.3. Comparison in terms of inequality constraint

Table 13 presents solution vector, P_j (j = 1, 2, ..., 40) corresponding to the best solution obtained from the OPSO and GPSO algorithms. In case of GPSO algorithm, the TGU$_4$ violates RRLs (red color). The TGU$_4$ must operate within $P_{4,low}$ = 24 MW and $P_{4,high}$ = 42 MW (16). This means that GPSO algorithm fails in solving PS-3 indicating that GPSO algorithm is unable to solve large scale ED problem. Whereas, the OPSO algorithm avoids the 46 POZs of 25 TGUs and is within RRLs.

5.4.4. Comparison in terms of power balance constraint

Since all the data for other existing ECTs are not available for PS-3, we compare the performance between OPSO algorithm and λ-logic [43]. The GPSO is out of comparison, because it failed in solving PS-3. The load demand of PS-3 is given as P_D = 8550 MW. Using the total optimum output power generated (Table 13) and Eqs. (5)–(7), P_{L1}, P_{L2} and $P_{L,mismatch}$, are determined and are presented in Table 14. It can be seen that $P_{L,mismatch}$ of OPSO algorithm is more

Table 13
Optimized output power for each TGU obtained by OPSO and GPSO algorithms for PS-3.

Optimum output power (MW)		
TGU	GPSO	**OPSO**
1	70.6053	79.9959
2	87.4700	100.0000
3	177.0581	190.0000
4	3.5760	41.0390
5	30.3749	36.6804
6	114.6114	115.0000
7	151.8385	153.4231
8	216.0995	217.0000
9	264.3870	265.0000
10	164.1162	160.5088
11	344.3448	354.1711
12	345.5677	348.6364
13	228.4384	219.8558
14	263.4364	280.0000
15	268.9301	270.0000
16	232.0047	270.0000
17	268.6993	270.0000
18	469.5224	470.0000
19	464.1353	470.0000
20	467.6499	468.0000
21	466.5318	468.0000
22	447.0084	460.0000
23	458.6545	460.0000
24	440.7477	460.0000
25	446.9992	460.0000
26	457.2052	460.0000
27	459.8870	460.0000
28	99.6434	35.1200
29	99.9167	35.9365
30	99.2875	45.3153
31	23.7599	56.3309
32	63.3148	36.1663
33	65.5211	41.2330
34	20.1840	47.6815
35	40.1371	40.0000
36	57.9855	55.0525
37	59.5737	53.4175
38	56.4260	40.0487
39	43.1472	40.0000
40	49.1706	54.4607
Total output power (MW)	8,587.9672	8,588.0734

close to 0.0 than λ-logic [43], which indicates better performance of OPSO algorithm.

6. Application of OPSO algorithm to CEC 2015 benchmark functions

Economic dispatch of power under various power constraints makes the objective function, i.e., cost function becomes multimodal function and it has multiple local optimums as discussed in Section 1. In order to provide a fair comparison and demonstrate the goodness of the proposed OPSO algorithm, ten selected shifted and rotated functions from CEC benchmark functions 2015 are added. Here we describe these functions and investigate performance of the GPSO and OPSO algorithms along with a few competitive ECTs.

6.1. Benchmark functions

Ten benchmark functions listed in Table 15 are used in this study. These benchmark functions are taken from the congress on evolutionary computation (CEC) 2015 and are used in performance comparison of global optimization algorithms [31,32]. All ten benchmark functions are minimization tasks. In addition, these are shifted and rotated functions, the global optimum solution x as shown in Table 15 is not located in the center of the search domain. The optimum solution x is shifted to a new position vector, i.e., shifted global optimum, $O_{ji} = [o_{j1}, o_{j2}, .., o_{jd}]$, where $j = 1, 2, \ldots, 10$ and $i = 1, 2, \ldots, d$. the d is the dimension of the each benchmark function. As well as, all functions are rotated by rotation matrix, M_j, $j = 1, 2, \ldots, 10$. The rotation does not affect the shape of the function but increases the function complexity in finding global optimum. The M is $d \times d$ matrix. It is applied to obtain the rotation and is generated from standard normally distributed entries using Gram-Schmidt orthnormalization process [31,32].

The ten benchmark functions are divided into two groups based on their significant physical properties. The first group involves three unimodal benchmark functions f_1–f_3 [31,32]. These are f_1 (Shifted and Rotated High Conditioned Elliptic), f_2 (Shifted and Rotated Cigar) and f_3 (Shifted and Rotated Discus). The second group includes seven multimodal benchmark functions f_4–f_{10}. Finding global optimum solution G_{best} is more interesting since these benchmark functions are more difficult to optimize because of the number of local minima as well as they are shifted and rotated. In multimodal functions, the number of local minima increases as the problem dimension increases [6,43]. Therefore, the search algorithm should be able to obtain a good solution and not become trapped in a local minimum. The seven multimodal functions are f_4 (Shifted and Rotated Ackley), f_5 (Shifted and Rotated Weierstrass), f_6 (Shifted and Rotated Rastrigin), f_7 (Shifted and Rotated Katsuura), f_8 (Shifted and Rotated HappyCat), f_9 (Shifted and Rotated HGBat) and f_{10} (Shifted and Rotated Expanded Griewank plus Rosenbrock).

Table 15 shows the details of the ten selected CEC 2015 benchmark functions. The name and mathematical description of f_1–f_{10} are shown in columns 2 and 3, respectively. The "Threshold Error" value of each function is available in column 4. The "optimum x" in column 5 and the minimum value of each function, "minimum $f(x)$" is in column 6. The solution of each function is judged successful, when the algorithm reaches to a value smaller than "Threshold Error". In other words, the algorithm passes the test.

6.2. Performance measures and experimental setup

To study the accuracy, stability and reliability of different algorithms, nine performance measures as explained below are considered. Let m be the number of particles in the swarm. Each algorithm is run over N_{run} times for N_{iter} iterations.

1. Number of Function Evaluations (NFE): The NFE is used as a measure of computational complexity of the algorithms. The NFE is the number of times the objective function $f(x)$ is evaluated in one run of the algorithm and is given by

$$\mathrm{NFE} = m \times N_{iter} \qquad (40)$$

Table 14
Comparison of power balance constraint between 2 ECTs for PS-3.

Algorithm	Total P_j MW	P_D MW	P_{L1} MW	P_{L2} MW	$P_{L,\mathrm{mismatch}}$ MW
λ-logic [43]	8,637.3300	8550	87.3300	87.4037	0.0737
OPSO	8,588.0734	8550	38.0734	38.1121	**0.0387**

Bold values signifies the best results in the respective category.

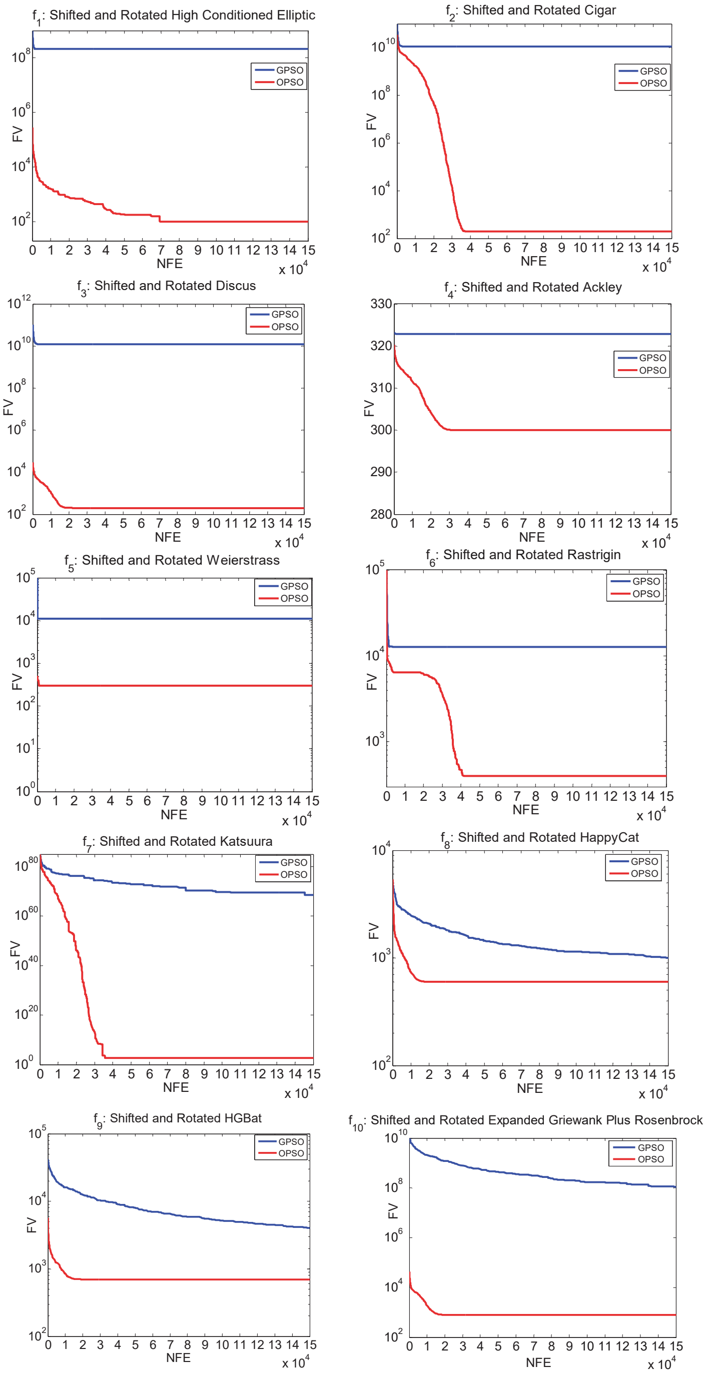


Fig. 17. Comparison of convergence characteristics between GPSO and OPSO algorithms for f_1–f_{10}.

Table 15
Ten benchmark objective functions used in the study.

f	Name	Function	Threshold Error	Optimum (x)	Minimum $f_i(x)$
f_1	Shifted and Rotated High	$f_1(x)=\sum_{i=1}^{d}\left(10^6\right)^{\left(\frac{i-1}{d-1}\right)}Z_{1,i}^2+F_1,$ $Z_1=M_1\times(x-O_{1i}),\ x=[x_1,x_2,\ldots,x_d],\ O_{1i}=[o_{11},o_{12},\ldots,o_{1d}]$	0.001	O_{1i}	$F_1=100$
f_2	Shifted and Rotated	$f_2(x)=Z_{2,1}^2+10^6\sum_{i=2}^{d}Z_{2,i}^2+F_2,$ $Z_2=M_2\times(x-O_{2i}),\ x=[x_1,x_2,\ldots,x_d],\ O_{2i}=[o_{21},o_{22},\ldots,o_{2d}]$	0.001	O_{2i}	$F_2=200$
f_3	Shifted and Rotated	$f_3(x)=10^6Z_{3,1}^2+\sum_{i=2}^{d}Z_{3,i}^2+F_3,$ $Z_3=M_3\times(x-O_{3i}),\ x=[x_1,x_2,\ldots,x_d],\ O_{3i}=[o_{31},o_{32},\ldots,o_{3d}]$	0.001	O_{3i}	$F_3=200$
f_4	Shifted and Rotated	$f_4(x)=-a\ \ \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}Z_{4,i}^2}\ \right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cZ_{4,i})\right)+$ $a+\exp\ (1)+F_4,\quad a=20,\quad b=0.2,\quad c=2\pi$ $Z_4=M_4\times(x-O_{4i}),\ x=[x_1,x_2,\ldots,x_d],\ O_{4i}=[o_{41},o_{42},\ldots,o_{4d}]$	0.001	O_{4i}	$F_4=300$
f_5	Shifted and Rotated	$f_5(x)=\sum_{i=1}^{d}\left[\sum_{k=1}^{k\max}a^k\cos(2\pi b^k(Z_{5,i}+0.5))\text{-}d\sum_{k=1}^{k\max}a^k\cos(\pi b^k)\right]+F_5$ $a=0.5,\ b=3,\ k_{max}=20,Z_5=M_5\times\left(\frac{0.5\times(x-O_{5i})}{100}\right),\ x=[x_1,x_2,\ldots,x_d]$ $O_{5i}=[o_{51},o_{52},\ldots,o_{5d}]$	0.001	O_{5i}	$F_5=300$
f_6	Shifted and Rotated	$f_6(x)=10\,d+\sum_{i=1}^{d}[Z_{6,i}^2-10\cos(2\pi Z_{6,i})]+F_6,$ $Z_6=M_6\times\left(\frac{5.12\times(x-O_{6i})}{100}\right),\ x=[x_1,x_2,\ldots,x_d],\ O_{6i}=[o_{61},o_{62},\ldots,o_{6d}]$	0.001	O_{6i}	$F_6=400$
f_7	Shifted and Rotated	$f_7(x)=\frac{10}{d^2}\prod_{i=1}^{d}\left[1+i\sum_{i=1}^{32}\frac{\lvert 2^i\times Z_{7,i}-round(2^i\times Z_{7,i})\rvert}{2^i}\right]^{\frac{10}{d^{1.2}}}-\frac{10}{d^2}+F_7$ $Z_7=M_7\times\left(\frac{5\times(x-O_{7i})}{100}\right),\ x=[x_1,x_2,\ldots,x_d],\ O_{7i}=[o_{71},o_{72},\ldots,o_{7d}]$	0.001	O_{7i}	$F_7=500$
f_8	Shifted and Rotated	$f_8(x)=\left[\sum_{i=1}^{d}Z_{8,i}^2-d\right]^{\frac{1}{4}}+\left(\frac{0.5\sum_{i=1}^{d}Z_{8,i}^2+\sum_{i=1}^{d}Z_{8,i})}{\mathrm{d}}\right)+0.5+F_8,$ $Z_8=M_8\times\left(\frac{5\times(x-O_{8i})}{100}\right),\ x=[x_1,x_2,\ldots,x_d],\ O_{8i}=[o_{81},o_{82},\ldots,o_{8d}]$	0.001	O_{8i}	$F_8=600$
f_9	Shifted and Rotated	$f_9(x)=\left[(\sum_{i=1}^{d}Z_{9,i}^2)^2-\left(\sum_{i=1}^{d}Z_{9,i}\right)^2\right]^{\frac{1}{2}}+\left(\frac{0.5\sum_{i=1}^{d}Z_{9,i}^2+\sum_{i=1}^{d}Z_{9,i}}{\mathrm{d}}\right)+0.5+F_9,$ $Z_9=M_9\times\left(\frac{5\times\left(x-O_{9i}\right)}{100}\right),\ x=[x_1,x_2,\ldots,x_d],\ O_{9i}=[o_{91},o_{92},\ldots,o_{9d}]$	0.001	O_{9i}	$F_9=700$
f_{10}	Shifted and Rotated	$f_g(x)=\frac{1}{4000}\sum_{i=1}^{d}Z_{10,i}^2-\prod_{i=1}^{d}\cos\left(\frac{Z_{10,i}}{\sqrt{i}}\right)+1$ $f_r(x)=\sum_{i=1}^{d-1}[100\ \ (\mathrm{Z}_{10,i+1}-Z_{10,i}^2)^2+(Z_{10,i}-1)^2]$ $f_{10}(x)=f_g(f_r(Z_{10,1},Z_{10,2})+f_g(f_r(Z_{10,2},Z_{10,3})+\cdots+$ $f_g(f_r(Z_{10,d-1},Z_{10,d})+f_g(f_r(Z_{10,d},Z_{10,1})+F_{10}$ $Z_{10}=M_{10}\times\left(\frac{5\times(x-O_{10i})}{100}\right)+1,\ x=[x_1,x_2,\ldots,x_d],$ $O_{10i}=[o_{101},o_{102},\ldots,o_{10d}]$	0.001	O_{10i}	$F_{10}=800$

2. Best Fitness Value (BFV): The BFV is defined as the minimum optimized $f(x)$ value obtained from N_{run} independent runs.

3. Worst Fitness Value (WFV): The WFV is defined as the maximum optimized $f(x)$ value obtained from N_{run} independent runs.

4. Mean Fitness Value (MFV): The MFV is defined as the average of the N_{run} BFVs.

5. Minimum Function Error Value (MFEV): The MFEV is defined as the difference between minimum f(x), i.e., column 6 in Table 15 and MFV. The MFEV is given by

$$MFEV\ =\ |minimum\,f\,(x)-\mathrm{MFV}| \tag{41}$$

6. Standard deviation (σ): The σ is the standard deviation of the N_{run} BFVs.

7. Success Rate (SR): An algorithm is successful if the MFEV of each function falls below the "Threshold Error". The SR is used as a

Table 16
Performance comparison between GPSO and OPSO algorithm on ten benchmark functions.

f	Minimum $f(x)$	Threshold Error	Fitness	GPSO	OPSO	f	Minimum $f(x)$	Threshold Error	Fitness	GPSO	OPSO
f_1	100	1×10^{-03}	BFV	6.7217×10^{07}	1.0000×10^{02}	f_6	400	1×10^{-03}	BFV	6.3726×10^{03}	4.0000×10^{02}
			WFV	3.5895×10^{08}	1.0000×10^{02}				WFV	1.9427×10^{04}	4.0000×10^{02}
			MFV	2.1017×10^{08}	1.0000×10^{02}				MFV	1.2650×10^{04}	4.0000×10^{02}
			MFEV	2.1007×10^{08}	0.0000				MFEV	1.264×10^{04}	0.0000
			σ	8.1122×10^{07}	5.9432×10^{-50}				σ	3.3683×10^{03}	3.7342×10^{-44}
			AET (sec)	–	172.4597				AET (sec)	–	210.0116
f_2	200	1×10^{-03}	BFV	6.0323×10^{09}	2.0000×10^{02}	f_7	500	1×10^{-03}	BFV	1.7048×10^{10}	5.0000×10^{02}
			WFV	1.5370×10^{10}	2.0000×10^{02}				WFV	2.8180×10^{69}	5.0000×10^{02}
			MFV	1.1061×10^{10}	2.0000×10^{02}				MFV	3.8636×10^{68}	5.0000×10^{02}
			MFEV	1.1051×10^{10}	0.0000				MFEV	3.8626×10^{68}	0.0000
			σ	2.7141×10^{09}	3.3248×10^{-54}				σ	8.8480×10^{68}	1.8626×10^{-43}
			AET (sec)	–	73.0369				AET (sec)	–	114.8182
f_3	200	1×10^{-03}	BFV	5.2660×10^{09}	2.0000×10^{02}	f_8	600	1×10^{-03}	BFV	8.6363×10^{02}	6.0000×10^{02}
			WFV	1.8594×10^{10}	2.0000×10^{02}				WFV	1.1724×10^{03}	6.0000×10^{02}
			MFV	1.2112×10^{10}	2.0000×10^{02}				MFV	9.9938×10^{02}	6.0000×10^{02}
			MFEV	1.2102×10^{10}	0.0000				MFEV	9.9928×10^{02}	0.0000
			σ	3.8578×10^{09}	3.4503×10^{-54}				σ	7.4942×10^{01}	1.5208×10^{-43}
			AET (sec)	–	35.8556				AET (sec)	–	121.8450
f_4	300	1×10^{-03}	BFV	3.2275×10^{02}	3.0000×10^{02}	f_9	700	1×10^{-03}	BFV	2.6696×10^{03}	7.0000×10^{02}
			WFV	3.2283×10^{02}	3.0000×10^{02}				WFV	6.4626×10^{03}	7.0000×10^{02}
			MFV	3.2289×10^{02}	3.0000×10^{02}				MFV	4.0527×10^{03}	7.0000×10^{02}
			MFEV	3.2279×10^{02}	0.0000				MFEV	4.0517×10^{03}	0.0000
			σ	3.3191×10^{-02}	6.1167×10^{-44}				σ	9.2835×10^{02}	3.0416×10^{-43}
			AET (sec)	–	37.4985				AET (sec)	–	122.2512
f_5	300	1×10^{-03}	BFV	1.1001×10^{02}	3.0000×10^{02}	f_{10}	800	1×10^{-03}	BFV	3.1925×10^{07}	8.0000×10^{02}
			WFV	1.1002×10^{02}	3.0000×10^{02}				WFV	2.6340×10^{08}	8.0000×10^{02}
			MFV	1.1001×10^{02}	3.0000×10^{02}				MFV	1.1174×10^{08}	8.0000×10^{02}
			MFEV	1.0991×10^{02}	0.0000				MFEV	1.1164×10^{08}	0.0000
			σ	4.7370×10^{-02}	1.0935×10^{-41}				σ	6.9107×10^{07}	2.2737×10^{-41}
			AET (sec)	–	230.8762				AET (sec)	–	139.2776

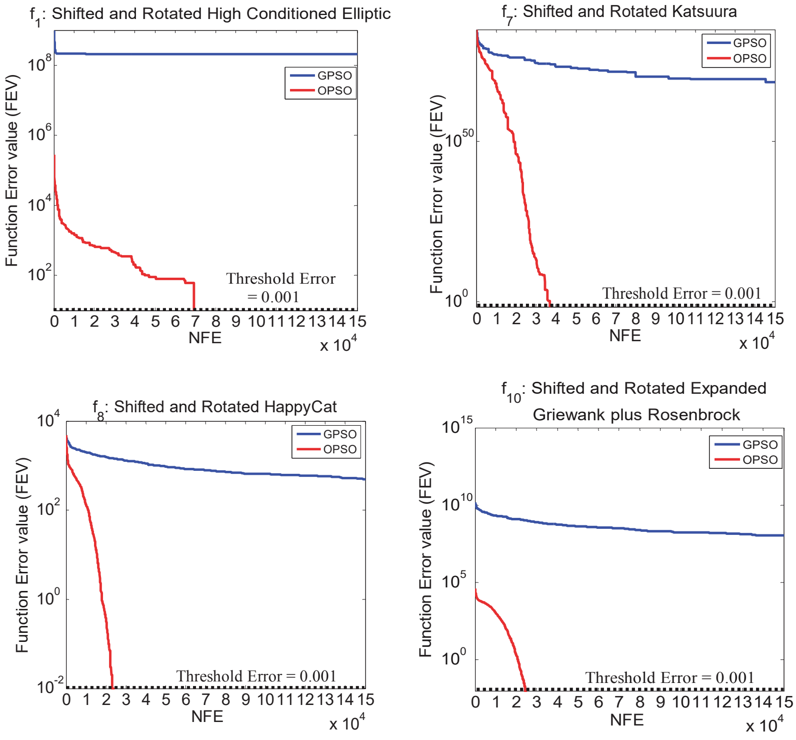


Fig. 18. The FEVs obtained at different runs at each NFE by the GPSO and OPSO algorithms for f_1, f_7–f_8 and f_{10}.

measure of reliability of the algorithm [31,32]. The SR in percentage is given by

$$SR = \frac{\text{Number of successful runs}}{N_{run}} \times 100 \quad (42)$$

8. Reliability Rate (RR): The RR of an algorithm over all the ten benchmark functions is defined

$$RR = \frac{1}{10} \sum_{i=1}^{10} SR_i \quad (43)$$

where SR_i is the success rate of the benchmark function $f_i(x)$, i = 1, 2, . . ., 10.

9. Average execution time (AET): It is the time consumed by an algorithm until it reaches to MFEV, averaged over N_{run} independent runs.

In order to measure the accuracy, stability and robustness of each algorithm, the GPSO and OPSO algorithms were evaluated using the ten unimodal and multimodal functions given in Table 15. Each function is tested with 30-dimension, d = 30. Based on the suggestion by the CEC 2015 [32], the optimization task has been carried out for N_{run} = 20 independent runs. The GPSO and OPSO algorithms are terminated when reaching the MFEV of each is smaller than 1.00×10^{-03}. The $N_{particle}$ in the OPSO and GPSO algorithm for f_1, f_2, . . ., f_{10} is, m = 40. The OPSO algorithm requires to be $m \geq d$. However this condition is not imposed in GPSO algorithm. The number of iterations N_{iter} is obtained from (40) once maximum NFE and m is decided. Thus, N_{iter} = 3750. Both GPSO and OPSO algorithms are run with maximum NFE = 150,000. The acceleration coefficients values of c_1 and c_2 in GPSO and c in OPSO algorithm are set at 2.00 and 2.05, respectively, using trial and error method. The parameters $r(t)$, $r_1(t)$and $r_2(t)$ are chosen randomly. In addition, the shifted global optimum vector O_{ji} for each function is randomly distributed in $[-80,80]^{30}$. In addition, an orthogonal (rotation) matrix M_j of each function is generated using Gram-Schmidt orthnormalization process.

6.3. Comparison in terms of fitness values

Performance comparison between GPSO and OPSO algorithms in terms of BFV, WFV, MFV, MFEV, σ and AET are shown in Table 16. It can be seen that in case of GPSO algorithm, the three fitness values BFV, WFV and MFV differ substantially from their optimal values for the ten functions. Whereas, in OPSO algorithm, the three fitness values are the same to their optimum values for all the ten functions. The MFEV of GPSO algorithm is so far from "Threshold Error" in f_1–f_{10}. However, in OPSO algorithm, the MFEV is smaller than "Threshold Error". It is achieved MFEV = 0.0 for the f_1–f_{10}. In terms of the standard deviation σ, it remains close to 0.0 in OPSO algorithm, indicating high stability and reliability of the OPSO algorithm. The results shown in Table 16, thus proves that OPSO algorithm is more accurate, stable and robust compared to the GPSO algorithm. In terms of the AET, the OPSO algorithm reaches "Threshold Error" within a specific AET as shown in Table 16. However, GPSO algorithm can't reach "Threshold Error", Which indicating that GPSO is unable to solve these ten shifted and rotated CEC 2015 benchmark functions under 30-dimension.

6.4. Success rate and reliability rate

The performance comparison between the GPSO and the OPSO algorithms with N_{run} = 20 (independent runs) in terms of SR and RR. The GPSO algorithm fails in ten CEC 2015 benchmark functions.

Table 17
Sensitivity analysis for the OPSO algorithm with increasing swarm's size with d = 30.

f	Minimum $f(x)$	Fitness	$m = 33$	AET (sec)	$m = 40$	AET (sec)	$m = 50$	AET (sec)	$m = 80$	AET (sec)	$m = 100$	AET (sec)
f_1	100	BFV	100.00	127.44	100.00	188.	100.00	313.28	100.00	390.09	100.00	430.70
		WFV	100.00		100.00	67	100.00		100.00		100.00	
		MFV	100.00		100.00		100.00		100.00		100.00	
f_7	500	BFV	500.00	55.26	500.00	124.57	500.00	182.50	500.00	232.89	500.00	282.63
		WFV	500.00		500.00		500.00		500.00		500.00	
		MFV	500.00		500.00		500.00		500.00		500.00	
f_8	600	BFV	600.00	65.22	600.00	131.45	600.00	183.07	600.00	246.92	600.00	317.89
		WFV	600.00		600.00		600.00		600.00		600.00	
		MFV	600.00		600.00		600.00		600.00		600.00	
f_{10}	800	BFV	800.00	74.13	800.00	106.63	800.00	191.14	800.00	234.64	800.00	294.27
		WFV	800.00		800.00		800.00		800.00		800.00	
		MFV	800.00		800.00		800.00		800.00		800.00	

Table 18
Performance comparison between OPSO algorithm and four ECTs using four CEC 2015-LBP [31] benchmark functions with d = 30.

ECTs	Performance measure	f_1	f_2	f_4	f_6
OPSO (proposed)	MFEV	**0.00**	**0.00**	**0.00**	**0.00**
	σ	**5.94×10^{-50}**	**3.32×10^{-54}**	**6.11×10^{-44}**	**3.73×10^{-44}**
LLUDE [33]	MFEV	5.93×10^{-01}	2.84×10^{-14}	2.03×10^{01}	2.59×10^{01}
	σ	2.47×10^{-01}	2.69×10^{-14}	2.33×10^{-02}	3.28×10^{00}
SaDE [33]	MFEV	1.78×10^{03}	2.38×10^{-11}	2.05×10^{01}	3.46×10^{01}
	σ	1.43×10^{03}	7.22×10^{-11}	5.99×10^{-02}	6.44×10^{00}
JADE [33]	MFEV	6.23×10^{00}	3.41×10^{-14}	2.03×10^{01}	2.61×10^{01}
	σ	1.55×10^{01}	1.17×10^{-14}	2.86×10^{-02}	3.39×10^{00}
CoDE [33]	MFEV	1.58×10^{04}	6.02×10^{-13}	2.00×10^{01}	2.97×10^{01}
	σ	1.16×10^{04}	9.88×10^{-13}	9.98×10^{-02}	1.08×10^{01}
SPS-L-SHADE-*E*IG	MFEV	**0.00**	**0.00**	2.00×10^{01}	1.03×10^{01}
[48] Rank #1 CEC 2015-LBP [31]	σ	**0.00**	**0.00**	7.29×10^{-05}	1.41×10^{01}
DEsPA	MFEV	**0.00**	**0.00**	2.01×10^{01}	9.71×10^{00}
[49] Rank #2 CEC 2015-LBP [31]	σ	**0.00**	**0.00**	4.36×10^{-02}	3.02×10^{00}
MVMO	MFEV	**0.00**	**0.00**	2.00×10^{01}	9.54×10^{00}
[50] Rank #3 CEC 2015-LBP [31]	σ	**0.00**	**0.00**	5.42×10^{-04}	3.53×10^{00}
Max.NFE		3.00×10^{05}	3.00×10^{05}	3.00×10^{05}	3.00×10^{05}

Bold values signifies the best results in the respective category.

However, the OPSO algorithm was successful in all these functions giving rise to SR of 100%. The RR of GPSO and OPSO algorithms are thus found to be 0.0% and 100%, respectively.

6.5. Convergence characteristics

Fig. 17 shows the convergence characteristics of GPSO and OPSO algorithms for ten shifted and rotated CEC 2015 benchmark functions f_1–f_{10}. The comparison is obtained in terms of fitness value (FV) averaged over N_{run} times at each NFE. It can be seen that, in case of OPSO algorithm the FV reduces to minimum value of f(x)) as NFE less than 4.0×10^{04} except f_1. Whereas, in case of GPSO algorithm, the FV fails to converge and remains above the "Threshold Error", which is indicating failure of the algorithm.

In order to highlight the superior performance of the OPSO algorithm over the GPSO algorithm, in Fig. 18 we provide the function error value (FEV) averaged over N_{run} times at each NFE for the four selected CEC 2015 benchmark functions, f_1, f_7–f_8 and f_{10}. The FEVs obtained by GPSO algorithm are much above "Threshold Error". In contrast, the OPSO algorithm was successful as the FEVs in these four benchmark functions remain below the "Threshold Error". Similar observations were made for the remaining benchmark functions.

The above mentioned observations provide the evidence of superior performance of the OPSO algorithm in terms of three fitness values, σ, convergence, SR and RR.

6.6. Sensitivity analysis of the OPSO algorithm against swarm's size

In order to study the sensitivity analysis of the proposed OPSO algorithm against the changing in the swarm population, m, four benchmarks functions f_1, f_7–f_8 and f_{10} are selected with a fixed number of iteration, i.e., $N_{iter} = 1000$. The test is carried out with $N_{\text{run}} = 20$. In addition, the AET is obtained at $t = 3750$ over 20 runs. Table 17 shows sensitivity analysis of the OPSO algorithm under effect of changing swarm's size m, from 40 to 100, on the performance of OPSO algorithm in terms of BFV, WFV, MFV and AET for f_1, f_7–f_8 and f_{10}. We can see from Table 17 that the AET increases as m increases. This gives rise to more computation. In addition, it does not yield any tangible change or better solution in BFV, WFV and MFV. This result leads to that when $m \gg d$, it gives rise to more computation, but does not provides any better solution. Considering these two extremes a reasonable value of m is could be about 10% more than d.

6.7. Comparison between the proposed OPSO algorithm and other ECTs

Here, we verify the performance of the proposed OPSO algorithm by comparing it with few ECTs recently reported by other authors [33–37,48–52].

6.7.1. Shifted and rotated benchmark functions f_1, f_2, f_4 and f_6

The three top-ranked algorithms in the CEC 2015 [31] learning based papers (LBP) are SPS-L-SHADE-*E*IG [48], DEsPA [49], and MVMO [50], in which the four shifted and rotated benchmark functions f_1, f_2, f_4 and f_6 are considered. Here, we compare the performance of the OPSO algorithm against the above three top-ranked algorithms and a few competitive algorithms from [33] for the four functions.

The comparison has been achieved with $N_{run} = 20$ and for each run the maximum NFE = 10,000 × d, i.e., 10,000 × 30 = 300,000, as given in [33]. The criterion used in this comparison depends on the values of maximum NFE and MFEV (41). When the algorithm

Table 19
Performance comparison between OPSO algorithm and four ECTs using six CEC 2015-EOP [32] benchmark functions with d = 30.

ECTs	Performance measure	f_3	f_5	f_7	f_8	f_9	f_{10}
OPSO (proposed)	MFEV	6.81×10^{-02}	**1.37×10^{00}**	**6.40×10^{-01}**	**1.49×10^{-02}**	**1.54×10^{-02}**	**1.29×10^{00}**
	σ	2.96×10^{-02}	**3.35×10^{-01}**	**4.00×10^{-01}**	**5.97×10^{-03}**	**7.45×10^{-03}**	6.74×10^{-02}
	AET (sec)	**2.32×10^{00}**	1.98×10^{01}	**4.51×10^{00}**	**1.47×10^{00}**	**1.43×10^{00}**	**1.60×10^{00}**
SbaDE [34]	MFEV	2.54×10^{09}	2.00×10^{01}	4.33×10^{02}	4.56×10^{04}	7.52×10^{01}	2.39×10^{07}
	σ	5.00×10^{09}	2.04×10^{02}	9.46×10^{01}	4.07×10^{04}	4.15×10^{03}	5.43×10^{07}
	AET (sec)	1.76×10^{01}	**1.65×10^{01}**	1.68×10^{01}	1.71×10^{01}	1.78×10^{01}	1.72×10^{01}
DD-SRPSO [35]	MFEV	2.59×10^{04}	2.24×10^{01}	2.71×10^{00}	5.61×10^{-01}	5.43×10^{-01}	3.32×10^{02}
	σ	1.05×10^{04}	2.12×0^{00}	7.58×10^{-01}	1.00×10^{-01}	2.03×10^{-01}	2.12×10^{02}
	AET (sec)	–	–	–	–	–	–
EPSO [36]	MFEV	6.37×10^{04}	3.38×10^{02}	5.04×10^{02}	6.02×10^{02}	7.21×10^{02}	1.27×10^{05}
	σ	–	–	–	–	–	–
	AET (sec)	–	–	–	–	–	–
SHPSO-GSA [37]	MFEV	6.96×10^{-01}	1.80×10^{01}	1.06×10^{00}	5.26×10^{-01}	1.36×10^{-01}	2.78×10^{00}
	σ	2.11×10^{01}	9.03×10^{-01}	5.55×10^{-01}	4.62×10^{-01}	3.11×10^{-01}	**1.04×10^{-04}**
	AET (sec)	–	–	–	–	–	–
MVMO [51] Rank #1 CEC 2015-EOP [32]	MFEV	**6.93×10^{-03}**	3.79×10^{01}	1.67×10^{01}	5.20×10^{-01}	4.39×10^{-01}	4.03×10^{02}
	σ	**3.24×10^{-04}**	3.85×10^{0}	5.04×10^{-01}	1.32×10^{-01}	9.93×10^{-02}	2.63×10^{02}
	AET (sec)	–	–	–	–	–	–
TunedCMAES [52] Rank #2 CEC 2015-EOP [32]	MFEV	1.17×10^{05}	3.21×10^{02}	5.05×10^{02}	6.00×10^{02}	7.00×10^{02}	8.22×10^{02}
	σ	2.19×10^{04}	5.06×10^{0}	5.91×10^{-01}	2.35×10^{-01}	2.86×10^{-01}	1.09×10^{-01}
	AET (sec)	–	–	–	–	–	–
Max.NFE		1.50×10^{03}	1.50×10^{03}	1.50×10^{03}	1.50×10^{03}	1.50×10^{03}	1.50×10^{03}

Bold values signifies the best results in the respective category.

Table 20
Statistical results of unpaired *t*-Test of OPSO algorithm against seven ECTs for CEC 2015-LBP [31].

			Competitive Algorithms						
Sl. No.	f	Statistical Results	LLUDE [33]	SaDE [33]	JADE [33]	CoDE [33]	SPS-L-SHADE-*E*IG [48] Rank #1 CEC 2015-LBP [31]	DEsPA [49] Rank #2 CEC 2015-LBP [31]	MVMO [50] Rank #3 CEC 2015-LPB [31]
1	f_1	t-value	$-\infty$	$-\infty$	**-1.53×10^{16}**	$-\infty$	0.0	0.0	0.0
		p-value	**0**	**0**	**7.99×10^{-297}**	**0**	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}
2	f_2	t-value	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		p-value	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}
3	f_4	t-value	**-2.49×10^{16}**	**-2.00×10^{16}**	**-2.49×10^{16}**	**-2.47×10^{16}**	**-2.47×10^{16}**	**-2.47×10^{16}**	**-2.47×10^{16}**
		p-value	**7.50×10^{-301}**	**4.74×10^{-299}**	**7.50×10^{-301}**	**9.05×10^{-301}**	**9.05×10^{-301}**	**9.05×10^{-301}**	**9.05×10^{-301}**
4	f_6	t-value	**-1.06×10^{16}**	**-2.12×10^{16}**	**-1.08×10^{16}**	$-\infty$	**-2.53×10^{16}**	**-1.19×10^{16}**	**-1.17×10^{16}**
		p-value	**8.51×10^{-294}**	**1.56×10^{-299}**	**7.35×10^{-294}**	**0**	**5.67×10^{-301}**	**9.13×10^{-295}**	**1.28×10^{-294}**
t = negative		t < 0	3	3	3	3	2	2	2
t = positive		t = ≥ 0	1	1	1	1	2	2	2
General Merit Over Contender			**2**	**2**	**2**	**2**	0	0	0

Bold values signifies the best results in the respective category.

Table 21
Statistical results of unpaired *t*-Test OPSO algorithm against six ECTs for CEC 2015-EOP [32].

			Competitive Algorithms					
Sl. No.	f	Statistical Results	SbaDE [34]	DD-SRPSO [35]	EPSO [36]	SHPSO-GSA [37]	MVMO [51] Rank #1 CEC 2015-EOP [32]	Tuned CMAES [52] Rank #2 CEC 2015-EOP [32]
1	f_3	t-value	**-3.80×10^{11}**	**-3.88×10^{06}**	**-9.54×10^{06}**	**-9.41×10^{01}**	1.33×10^{00}	**-1.75×10^{07}**
		p-value	**2.37×10^{-209}**	**1.63×10^{-114}**	**6.14×10^{-122}**	**7.89×10^{-27}**	9.80×10^{-02}	**5.90×10^{-127}**
2	f_5	t-value	**-2.49×10^{02}**	**-2.81×10^{02}**	**-4.49×10^{03}**	**-2.22×10^{02}**	**-4.87×10^{02}**	**-4.27×10^{03}**
		p-value	**7.69×10^{-35}**	**7.69×10^{-36}**	**1.00×10^{-58}**	**6.66×10^{-34}**	**2.13×10^{-40}**	**2.69×10^{-58}**
3	f_7	t-value	**-6.49×10^{04}**	**-3.96×10^{02}**	**-7.55×10^{04}**	**-1.48×10^{02}**	**-2.49×10^{03}**	**-7.56×10^{04}**
		p-value	**9.44×10^{-81}**	**1.12×10^{-38}**	**5.27×10^{-82}**	**1.35×10^{-30}**	**7.39×10^{-54}**	**5.07×10^{-82}**
4	f_8	t-value	**-3.40×10^{07}**	**-4.09×10^{02}**	**-4.50×10^{05}**	**-3.82×10^{02}**	**-3.78×10^{02}**	**-4.48×10^{05}**
		p-value	**1.93×10^{-132}**	**6.07×10^{-39}**	**9.85×10^{-97}**	**2.13×10^{-38}**	**2.67×10^{-38}**	**1.05×10^{-96}**
5	f_9	t-value	**-4.07×10^{04}**	**-2.85×10^{02}**	**-3.90×10^{05}**	**-1.01×10^{01}**	**-2.29×10^{02}**	**-3.79×10^{05}**
		p-value	**6.56×10^{-77}**	**5.50×10^{-36}**	**1.45×10^{-95}**	**8.08×10^{-24}**	**3.56×10^{-34}**	**2.55×10^{-95}**
6	f_{10}	t-value	**-1.59×10^{09}**	**-2.21×10^{04}**	**-8.48×10^{06}**	**-1.77×10^{02}**	**-2.69×10^{04}**	**-5.49×10^{04}**
		p-value	**3.46×10^{-164}**	**6.77×10^{-72}**	**5.71×10^{-121}**	**4.79×10^{-32}**	**1.70×10^{-73}**	**2.22×10^{-79}**
t = negative		t < 0	6	6	6	6	5	6
t = positive		t = ≥ 0	0	0	0	0	1	0
General Merit		Over Contender	**6**	**6**	**6**	**6**	**4**	**6**

Bold values signifies the best results in the respective category.

reaches NFE = 300,000, the MFEV is recorded as a better result. The algorithm obtains a best result when the MFEV is 0.0 or close to 0.0.

Table 18 presents the results of the MFEV and the corresponding σ obtained by the eight ECTs. We can find that the OPSO algorithm is significantly superior to LLUDE, SaDE, JADE and CoDE [33] in solving f_1,f_2,f_4 and f_6. While comparing with CEC 2015-LBP [31] algorithms, performance of the OPSO algorithm found similar to that of the three top-ranked algorithms for the functions f_1 and f_2. However, the OPSO algorithm outperforms the three top-ranked algorithms for the function f_4 and f_6.

6.7.2. *Shifted and rotated benchmark functions f_3, f_5, f_7–f_{10}*

The benchmark functions in CEC 2015 expensive optimization papers (EOP) [32] are highly competitive and require efficient optimization algorithms to provide fast solutions with a high accuracy. The two top-ranked algorithms are MVMO [51] and TunedCMAES [52], in which the six shifted and rotated benchmark functions f_3, f_5, f_7–f_{10} are considered.

Here, we compare the performance of the OPSO algorithm with that of the above two top-ranked algorithms and few other competitive algorithms from [34–37]. The comparison has been achieved with N_{run} = 20 and for each run the exact maximum NFE = 1500 as given in [32]. The dimension of each tested function is d = 30. The 50 particles have been used in the DD-SRPSO [35] and SHPSO-GSA [37] algorithms, whereas 60 particles are used for the EPSO [36] algorithm. In this experiment, The OPSO algorithm uses 50 particles. Thus, the N_{iter} = 30 based on (40).

Table 19 shows the MFEV, the corresponding σ and AET of the seven ECTs. Among the seven ECTs, the OPSO algorithm achieves the best MFEV performance for the five functions, f_5, f_7–f_{10}, whereas the MVMO [51] gives the best MFEV performance for the function f_3. In terms of σ, the performance of the OPSO algorithm is the best in case of the four functions f_5, f_7–f_9 and is the second best for the functions, f_3 and f_{10}. Thus, the performance of the OPSO algorithm found to be superior to the two CEC 2015-EOP [32] algorithms. In terms of AET, the OPSO algorithm performance is better than SabDE [34] algorithm for all the functions except f_5.

6.8. *Statistical significance of the proposed OPSO algorithm*

In order to determine the statistical significance of the proposed OPSO algorithm, we carried out three sets of unpaired one-tailed *t*-Test [68] with a significance level of α = 0.05. The results of the *t*-Test for CEC 2015-LBP [31] are shown in Table 20. Here, we provide the statistical results of the comparison between OPSO algorithm and seven competitive algorithms for f_1, f_2, f_4 and f_6. The comparison is made with a degree of freedom equals to 19. The OPSO algorithm is considered to be statistically significant against the contender algorithm when t-value < 0 and p-value less than 0.05. The general merit over contender is shown in the last row of Table 20. It is calculated as the difference between the number of times the OPSO algorithm is found to be statistically significant and statistically not significant among four tested functions. It can be seen that out of the seven algorithms, the OPSO algorithm is statistically significant against four algorithms, i.e., LLUDE, SaDE, JADE, and CoDE [33]. However, against the three top-ranked algorithms CEC 2015-LBP [31], SPS-L-SHADE-EIG [48], DEsPA [49], and MVMO [50], the OPSO algorithm is statistically significant for f_4 and f_6, whereas it is statistically not significant for f_1 and f_2.

Table 21 shows the statistical results of the comparison between OPSO algorithm and six competitive algorithms for the six functions, f_3, f_5, f_7–f_{10}. The comparison is made with a degree of freedom equals to 19. The general merit over contender is shown in the last row of Table 21. It can be seen that the OPSO algorithm is statistically significant against all the six algorithms.

Table 22
Statistical results of unpaired *t*-Test of OPSO algorithm against twenty four ECTs for PS-1, PS-2 and PS-3.

Sl. No.	Competitive	PS-1		PS-2		PS-3		t = negative t < 0	t = positive t ≥ 0	General Merit Over Contender
		t-value	p-value	t-value	p-value	t-value	p-value			
1	GA [25]	**-1.03×10^{03}**	**1.14×10^{-201}**	**-3.14×10^{04}**	**0.0**	-2.53×10^{02}	4.69×10^{-141}	3	0	**3**
2	DE [25]	**-3.27×10^{02}**	**4.33×10^{-152}**	**-2.31×10^{04}**	**0.0**	-1.07×10^{02}	2.48×10^{-104}	3	0	**3**
3	ACSA [25]	**-7.71×10^{02}**	**7.23×10^{-189}**	**-2.75×10^{04}**	**0.0**	-1.82×10^{02}	8.67×10^{-127}	3	0	**3**
4	AIS [25]	**-6.36×10^{02}**	**1.35×10^{-180}**	**-2.50×10^{04}**	**0.0**	-1.48×10^{02}	3.68×10^{-118}	3	0	**3**
5	FA [25]	**-8.55×10^{02}**	**2.51×10^{-193}**	**-2.96×10^{04}**	**0.0**	-2.04×10^{02}	1.06×10^{-131}	3	0	**3**
6	BCO [25]	**-6.96×10^{02}**	**1.64×10^{-184}**	**-3.19×10^{04}**	**0.0**	-1.45×10^{02}	4.08×10^{-117}	3	0	**3**
7	APSO [25]	**-1.42×10^{03}**	**3.25×10^{-215}**	**-2.00×10^{04}**	**0.0**	-1.73×10^{02}	9.77×10^{-125}	3	0	**3**
8	HGPSO [25]	**-9.17×10^{02}**	**2.43×10^{-196}**	**-2.62×10^{04}**	**0.0**	-2.20×10^{02}	4.70×10^{-135}	3	0	**3**
9	HPSOM [25]	**-2.86×10^{02}**	**2.63×10^{-146}**	**-1.08×10^{04}**	**1.64×10^{-302}**	-9.54×10^{01}	3.04×10^{-99}	3	0	**3**
10	HPSOWM [25]	**-5.86×10^{02}**	**3.88×10^{-177}**	**-9.84×10^{03}**	**2.34×10^{-298}**	-1.16×10^{02}	1.35×10^{-107}	3	0	**3**
11	RDPSO [25]	**-8.60×10^{02}**	**7.53×10^{-95}**	**-5.09×10^{03}**	**4.92×10^{-270}**	**-7.05×10^{01}**	**1.92×10^{-86}**	3	0	**3**
12	λ-logic [43]	**-3.11×10^{02}**	**6.62×10^{-150}**	**-3.24×10^{03}**	**1.39×10^{-250}**	**-8.02×10^{01}**	**6.29×10^{-92}**	3	0	**3**
13	GPSO	**-7.18×10^{02}**	**8.08×10^{-186}**	**-3.37×10^{04}**	**0.0**	-8.95×10^{03}	2.69×10^{-294}	3	0	**3**
14	CPSO [29]	**-2.74×10^{02}**	**2.01×10^{-144}**	**-2.53×10^{04}**	**0.0**	–	–	2	0	**2**
15	SPPO [46]	**-3.21×10^{02}**	**2.97×10^{-151}**	**-3.18×10^{03}**	**7.17×10^{-250}**	–	–	2	0	**2**
16	SA-PSO [26]	**-1.79×10^{02}**	**2.80×10^{-126}**	**-4.53×10^{03}**	**4.47×10^{-265}**	–	–	2	0	**2**
17	MIQCQP [41]	**6.12×10^{00}**	**1.82×10^{-08}**	–	–	**-3.45×10^{01}**	**4.49×10^{-57}**	2	0	**2**
18	MSAF [22]	–	–	**-6.52×10^{03}**	**1.06×10^{-280}**	–	–	1	0	**1**
19	EGSSOA [42]	–	–	**-8.04×10^{02}**	**1.07×10^{-190}**	–	–	1	0	**1**
20	SOH-PSO	**-2.55×10^{03}**	**1.94×10^{-240}**	–	–	–	–	1	0	**1**
21	NPSO-LRS [27]	**-3.44×10^{02}**	**2.66×10^{-154}**	–	–	–	–	1	0	**1**
22	QMPSO [30]	**-1.36×10^{03}**	**1.44×10^{-144}**	–	–	–	–	1	0	**1**
23	FDA [38]	**-3.02×10^{02}**	**1.02×10^{-148}**	–	–	–	–	1	0	**1**
24	GA-API [40]	**-3.14×10^{02}**	**2.70×10^{-150}**	–	–	–	–	1	0	**1**

Bold values signifies the best results in the respective category.

Table 22 shows the *t*-Test results for the three power systems, PS-1–PS-3. The one-tailed unpaired with α = 0.05 with a degree of freedom of 99 is performed against twenty four competitive algorithms. As seen from the data in the last column, the proposed OPSO algorithm is found to be statistically significant against the first thirteen competitive algorithms for three power systems.

Since the data are not available for PS-1, PS-2, or PS-3 for eleven algorithms with Sl.No. 14–24, the *t*-Test is carried out against one or two power systems. Again, from the data in the last column, one can observe that the OPSO algorithm is statistically significant against the eleven contending algorithms. These results give enough evidence that the proposed OPSO algorithm is statistically significant against the twenty four contending algorithms.

7. Conclusion

A novel optimization algorithm named orthogonal PSO algorithm is proposed to alleviate the problems associated with the global PSO algorithm. An orthogonal diagonalization process is carried out in the OPSO algorithm which aims to diagonalize the position vectors of the active group particles. In contrast to two guides as used in GPSO algorithm, the OPSO algorithm uses only one guide while updating of the position and velocity vectors. The OPSO algorithm is applied for solving economic dispatch problem of thermal generating units (TGUs) under various practical power constraints imposed by the smart grid and power systems. The OPSO algorithm is tested with three practical power systems and ten selected CEC 2015 benchmark functions of increasing complexity and its superiority over GPSO algorithm and several existing ECTs has been shown with extensive simulation studies. The proposed OPSO algorithm has shown evidence of superior performance compared to several existing ECTs in providing reliable, consistent and optimal solution for the economic dispatch problem and shifted rotated CEC 2015 benchmark functions. The OPSO algorithm is found to be statistically significant against several ECTs including top-ranked CEC 2015 algorithms.

Appendix A.

$$B = 1\times10^{-4}\begin{bmatrix}
0.6723 & 0.0915 & 0.5294 & 0.9081 & 0.7637 & 0.1982 & 0.0086 & 0.1108 & 0.8954 & 0.5386 & 0.7449 & 0.2905 & 0.6198 & 0.7461 & 0.1080 & 0.0560 & 0.2790 & 0.3400 & 0.2672 & 0.0919 & 0.8169 & 0.0832 & 0.7715 & 0.9262 & 0.8964 & 0.0499 & 0.5824 & 0.9758 & 0.8564 & 0.8281 & 0.4880 & 0.0137 & 0.9521 & 0.9316 & 0.3839 & 0.3796 & 0.5480 & 0.3747 & 0.7290 & 0.1868 \\
0.4315 & 0.6146 & 0.8300 & 0.5522 & 0.5588 & 0.1951 & 0.7271 & 0.4075 & 0.5825 & 0.9917 & 0.8923 & 0.4026 & 0.2606 & 0.6625 & 0.4599 & 0.8169 & 0.6754 & 0.8467 & 0.7537 & 0.4021 & 0.1895 & 0.4617 & 0.7648 & 0.2985 & 0.4822 & 0.5459 & 0.4210 & 0.5554 & 0.8998 & 0.5074 & 0.4978 & 0.3741 & 0.9759 & 0.7831 & 0.2602 & 0.3191 & 0.5669 & 0.4369 & 0.2026 & 0.2472 \\
0.6944 & 0.0110 & 0.8588 & 0.0329 & 0.1838 & 0.3268 & 0.3541 & 0.8841 & 0.5827 & 0.7552 & 0.2426 & 0.8621 & 0.4457 & 0.5233 & 0.4509 & 0.5289 & 0.9037 & 0.2461 & 0.8984 & 0.2952 & 0.1237 & 0.0304 & 0.4211 & 0.3381 & 0.0141 & 0.9432 & 0.0921 & 0.8463 & 0.2179 & 0.3662 & 0.9360 & 0.9227 & 0.0309 & 0.6857 & 0.8775 & 0.9861 & 0.6804 & 0.3043 & 0.2163 & 0.0542 \\
0.2568 & 0.5733 & 0.7890 & 0.0539 & 0.4979 & 0.8803 & 0.7804 & 0.5481 & 0.8549 & 0.9805 & 0.1296 & 0.6147 & 0.8440 & 0.2599 & 0.5511 & 0.6944 & 0.9085 & 0.5815 & 0.7284 & 0.3065 & 0.8210 & 0.7532 & 0.0568 & 0.8595 & 0.6229 & 0.3215 & 0.0240 & 0.4081 & 0.0770 & 0.2266 & 0.3893 & 0.5465 & 0.4939 & 0.4662 & 0.8061 & 0.7182 & 0.3714 & 0.2909 & 0.9763 & 0.6090 \\
0.0098 & 0.7897 & 0.3178 & 0.8051 & 0.5178 & 0.4711 & 0.4367 & 0.3690 & 0.0349 & 0.2348 & 0.2251 & 0.9912 & 0.1962 & 0.9620 & 0.8054 & 0.2124 & 0.7472 & 0.9377 & 0.4068 & 0.1056 & 0.6379 & 0.7000 & 0.5857 & 0.3405 & 0.2311 & 0.8065 & 0.4911 & 0.4620 & 0.4742 & 0.5348 & 0.1171 & 0.4739 & 0.8627 & 0.2603 & 0.4611 & 0.4132 & 0.0782 & 0.2425 & 0.5932 & 0.7772 \\
0.5323 & 0.2354 & 0.4522 & 0.4514 & 0.9942 & 0.4040 & 0.4366 & 0.2083 & 0.8854 & 0.5286 & 0.3500 & 0.2037 & 0.3039 & 0.5402 & 0.7009 & 0.5433 & 0.2605 & 0.0478 & 0.9383 & 0.5938 & 0.0161 & 0.2145 & 0.1742 & 0.1381 & 0.5274 & 0.6014 & 0.2783 & 0.8263 & 0.8350 & 0.2895 & 0.2404 & 0.4965 & 0.2429 & 0.5693 & 0.0910 & 0.0986 & 0.4564 & 0.9367 & 0.3044 & 0.5111 \\
0.2794 & 0.4480 & 0.7522 & 0.3826 & 0.8549 & 0.1792 & 0.0492 & 0.4409 & 0.4077 & 0.0514 & 0.2871 & 0.8272 & 0.4833 & 0.0303 & 0.8722 & 0.7025 & 0.6896 & 0.0540 & 0.2554 & 0.2827 & 0.8960 & 0.6799 & 0.7286 & 0.5078 & 0.7250 & 0.7896 & 0.3398 & 0.9912 & 0.4694 & 0.0684 & 0.6849 & 0.3090 & 0.8343 & 0.2488 & 0.5643 & 0.7346 & 0.0478 & 0.8602 & 0.9677 & 0.0278 \\
0.9462 & 0.5694 & 0.1099 & 0.7896 & 0.9624 & 0.9689 & 0.0496 & 0.9562 & 0.0364 & 0.7569 & 0.9275 & 0.6759 & 0.3378 & 0.6963 & 0.0522 & 0.9564 & 0.1318 & 0.0206 & 0.5332 & 0.1552 & 0.5154 & 0.5573 & 0.5343 & 0.8567 & 0.6074 & 0.7992 & 0.2873 & 0.5239 & 0.4138 & 0.0850 & 0.8393 & 0.9508 & 0.8136 & 0.3193 & 0.1874 & 0.6373 & 0.7383 & 0.3972 & 0.8960 & 0.9904 \\
0.9064 & 0.0614 & 0.1097 & 0.3643 & 0.6789 & 0.4075 & 0.0911 & 0.1240 & 0.7461 & 0.6020 & 0.0513 & 0.2489 & 0.7985 & 0.5197 & 0.2197 & 0.4445 & 0.1235 & 0.6815 & 0.9548 & 0.0007 & 0.5445 & 0.8507 & 0.2531 & 0.3843 & 0.5884 & 0.0496 & 0.1709 & 0.9254 & 0.5027 & 0.0683 & 0.9701 & 0.9820 & 0.6290 & 0.9108 & 0.5317 & 0.0738 & 0.0380 & 0.4794 & 0.1900 & 0.5009 \\
0.3927 & 0.4963 & 0.2699 & 0.5323 & 0.4035 & 0.8445 & 0.5940 & 0.4708 & 0.1548 & 0.8572 & 0.5927 & 0.4758 & 0.9875 & 0.0590 & 0.4596 & 0.0854 & 0.1909 & 0.5986 & 0.2677 & 0.2836 & 0.6064 & 0.5586 & 0.9171 & 0.6957 & 0.4334 & 0.2832 & 0.3993 & 0.7390 & 0.1254 & 0.4098 & 0.2152 & 0.5136 & 0.0022 & 0.8852 & 0.3550 & 0.1205 & 0.9542 & 0.5650 & 0.0018 & 0.3320 \\
0.0249 & 0.6423 & 0.5246 & 0.7117 & 0.9350 & 0.6153 & 0.2411 & 0.8569 & 0.1439 & 0.9883 & 0.1629 & 0.3991 & 0.1590 & 0.8900 & 0.9585 & 0.0573 & 0.1457 & 0.1140 & 0.2501 & 0.5508 & 0.7604 & 0.9018 & 0.7582 & 0.6279 & 0.2442 & 0.6535 & 0.6976 & 0.5674 & 0.1323 & 0.1234 & 0.7603 & 0.9926 & 0.3797 & 0.7946 & 0.3148 & 0.9816 & 0.7424 & 0.4896 & 0.7118 & 0.1739 \\
0.6714 & 0.2213 & 0.9727 & 0.8715 & 0.4795 & 0.3766 & 0.8414 & 0.0434 & 0.6060 & 0.9295 & 0.8384 & 0.5994 & 0.2369 & 0.3302 & 0.7900 & 0.6295 & 0.5850 & 0.7962 & 0.9277 & 0.8709 & 0.8553 & 0.4195 & 0.8870 & 0.4504 & 0.4290 & 0.4897 & 0.2037 & 0.9688 & 0.8705 & 0.4430 & 0.5841 & 0.4558 & 0.9044 & 0.9258 & 0.7267 & 0.4968 & 0.9374 & 0.2698 & 0.8677 & 0.6256 \\
0.8372 & 0.8371 & 0.7104 & 0.3287 & 0.2318 & 0.8772 & 0.8572 & 0.6916 & 0.2545 & 0.4095 & 0.1676 & 0.8005 & 0.7022 & 0.2297 & 0.4519 & 0.7962 & 0.0734 & 0.6179 & 0.0686 & 0.0423 & 0.3829 & 0.3581 & 0.0688 & 0.4736 & 0.0102 & 0.9729 & 0.6663 & 0.8245 & 0.6030 & 0.8989 & 0.4030 & 0.4260 & 0.6804 & 0.1788 & 0.5158 & 0.0224 & 0.5134 & 0.9897 & 0.1183 & 0.5751 \\
0.9715 & 0.9711 & 0.3119 & 0.6501 & 0.3963 & 0.7849 & 0.9636 & 0.9790 & 0.3242 & 0.0003 & 0.5022 & 0.1051 & 0.3755 & 0.1139 & 0.3334 & 0.6912 & 0.8223 & 0.0702 & 0.2994 & 0.9047 & 0.0846 & 0.4890 & 0.1835 & 0.9497 & 0.6088 & 0.7485 & 0.4431 & 0.9596 & 0.2653 & 0.3536 & 0.5100 & 0.2132 & 0.3788 & 0.5175 & 0.7906 & 0.0538 & 0.2409 & 0.1837 & 0.0390 & 0.7510 \\
0.0569 & 0.8464 & 0.2915 & 0.9748 & 0.7051 & 0.4650 & 0.4889 & 0.2833 & 0.4018 & 0.5409 & 0.9993 & 0.8214 & 0.9737 & 0.3109 & 0.0591 & 0.3453 & 0.7229 & 0.0693 & 0.5916 & 0.1310 & 0.7339 & 0.2560 & 0.7371 & 0.0835 & 0.9580 & 0.5678 & 0.4333 & 0.6463 & 0.8648 & 0.1202 & 0.4956 & 0.1932 & 0.6320 & 0.6270 & 0.2045 & 0.1409 & 0.2600 & 0.8617 & 0.5982 & 0.1535 \\
0.4503 & 0.5060 & 0.8504 & 0.0760 & 0.5586 & 0.8140 & 0.2203 & 0.1338 & 0.4064 & 0.2077 & 0.3554 & 0.8411 & 0.9723 & 0.2284 & 0.7409 & 0.9468 & 0.9259 & 0.1360 & 0.2033 & 0.8337 & 0.3320 & 0.9292 & 0.6967 & 0.2798 & 0.0356 & 0.2561 & 0.1932 & 0.4766 & 0.4578 & 0.8750 & 0.7437 & 0.7266 & 0.5714 & 0.6640 & 0.0525 & 0.4658 & 0.7590 & 0.0326 & 0.6043 & 0.3568 \\
0.5825 & 0.2789 & 0.9116 & 0.5870 & 0.7566 & 0.8984 & 0.2262 & 0.6853 & 0.3862 & 0.2193 & 0.0471 & 0.3545 & 0.6437 & 0.6520 & 0.5068 & 0.5202 & 0.4926 & 0.7889 & 0.6359 & 0.8005 & 0.8397 & 0.4668 & 0.7770 & 0.4470 & 0.8862 & 0.8866 & 0.6164 & 0.9119 & 0.7222 & 0.3486 & 0.3020 & 0.5297 & 0.9817 & 0.3892 & 0.8012 & 0.5609 & 0.9933 & 0.3320 & 0.5164 & 0.1440 \\
0.6866 & 0.7466 & 0.6393 & 0.4139 & 0.9955 & 0.4292 & 0.5368 & 0.9095 & 0.6098 & 0.3258 & 0.2137 & 0.4301 & 0.8601 & 0.0662 & 0.1999 & 0.9538 & 0.6549 & 0.0924 & 0.7984 & 0.9179 & 0.3717 & 0.2540 & 0.5019 & 0.5876 & 0.2469 & 0.4468 & 0.2690 & 0.0149 & 0.3390 & 0.0419 & 0.0896 & 0.8291 & 0.8497 & 0.7400 & 0.6786 & 0.4945 & 0.3567 & 0.7487 & 0.0075 & 0.8506 \\
0.7194 & 0.2369 & 0.2554 & 0.3091 & 0.9624 & 0.3343 & 0.7621 & 0.6109 & 0.1669 & 0.0959 & 0.3978 & 0.5722 & 0.4019 & 0.2754 & 0.4272 & 0.0736 & 0.8901 & 0.2379 & 0.5017 & 0.1373 & 0.8282 & 0.4312 & 0.4255 & 0.8776 & 0.0089 & 0.8160 & 0.5597 & 0.1567 & 0.4012 & 0.1423 & 0.8260 & 0.5119 & 0.2834 & 0.8176 & 0.9460 & 0.0678 & 0.7529 & 0.6444 & 0.6889 & 0.3379 \\
0.6500 & 0.9573 & 0.0887 & 0.2638 & 0.5351 & 0.5966 & 0.3476 & 0.9000 & 0.1881 & 0.7475 & 0.3337 & 0.7008 & 0.6319 & 0.2818 & 0.1687 & 0.2070 & 0.5385 & 0.2436 & 0.6508 & 0.5047 & 0.1765 & 0.7025 & 0.6112 & 0.4691 & 0.8149 & 0.0983 & 0.9448 & 0.4716 & 0.5270 & 0.0766 & 0.3896 & 0.5520 & 0.6825 & 0.6003 & 0.0916 & 0.8976 & 0.1100 & 0.1692 & 0.9460 & 0.2752 \\
0.7269 & 0.6203 & 0.8383 & 0.7588 & 0.9639 & 0.9020 & 0.4612 & 0.1934 & 0.0946 & 0.7485 & 0.2296 & 0.7425 & 0.9852 & 0.8801 & 0.7517 & 0.7750 & 0.2822 & 0.1048 & 0.7960 & 0.4050 & 0.1295 & 0.4023 & 0.8558 & 0.4374 & 0.1405 & 0.8596 & 0.7145 & 0.5430 & 0.8942 & 0.7405 & 0.7753 & 0.2133 & 0.3581 & 0.0850 & 0.9084 & 0.2886 & 0.5970 & 0.9522 & 0.8735 & 0.0060 \\
0.3738 & 0.6003 & 0.5847 & 0.9952 & 0.1156 & 0.7021 & 0.6393 & 0.7544 & 0.3232 & 0.5433 & 0.9361 & 0.7579 & 0.5595 & 0.4443 & 0.3684 & 0.9142 & 0.9760 & 0.8584 & 0.2334 & 0.1736 & 0.8799 & 0.1818 & 0.6708 & 0.7462 & 0.8799 & 0.0276 & 0.6792 & 0.0597 & 0.7784 & 0.4565 & 0.1794 & 0.5878 & 0.9869 & 0.9224 & 0.5100 & 0.2690 & 0.4306 & 0.5433 & 0.1133 & 0.8019 \\
0.5816 & 0.1726 & 0.9481 & 0.1866 & 0.0514 & 0.3775 & 0.9173 & 0.3463 & 0.7696 & 0.3381 & 0.6832 & 0.3891 & 0.9336 & 0.7559 & 0.9418 & 0.7826 & 0.0364 & 0.6982 & 0.6008 & 0.5752 & 0.0441 & 0.8563 & 0.5236 & 0.4679 & 0.0954 & 0.8992 & 0.9594 & 0.6580 & 0.0694 & 0.6682 & 0.1094 & 0.1428 & 0.0840 & 0.0536 & 0.6149 & 0.5942 & 0.7307 & 0.2514 & 0.3546 & 0.4974 \\
0.1161 & 0.0903 & 0.0610 & 0.7811 & 0.3043 & 0.7350 & 0.1616 & 0.4186 & 0.2341 & 0.8323 & 0.9621 & 0.4293 & 0.7203 & 0.6033 & 0.0172 & 0.2955 & 0.3262 & 0.7337 & 0.1125 & 0.6062 & 0.6867 & 0.5842 & 0.2988 & 0.8608 & 0.3526 & 0.8999 & 0.7753 & 0.8896 & 0.2788 & 0.6992 & 0.9052 & 0.0522 & 0.2503 & 0.5270 & 0.3161 & 0.4759 & 0.2612 & 0.5786 & 0.2419 & 0.5378 \\
0.0577 & 0.2553 & 0.5846 & 0.1958 & 0.5802 & 0.9541 & 0.7156 & 0.1557 & 0.7404 & 0.5526 & 0.4380 & 0.9563 & 0.4840 & 0.7833 & 0.8291 & 0.1518 & 0.9730 & 0.6505 & 0.5158 & 0.2144 & 0.7338 & 0.3736 & 0.7040 & 0.4665 & 0.5934 & 0.5241 & 0.6077 & 0.1096 & 0.3794 & 0.5714 & 0.8764 & 0.6833 & 0.8113 & 0.1189 & 0.0775 & 0.3683 & 0.0948 & 0.9155 & 0.5603 & 0.8709 \\
0.9798 & 0.8586 & 0.2851 & 0.9924 & 0.5310 & 0.5428 & 0.5777 & 0.8190 & 0.6928 & 0.9575 & 0.9403 & 0.5730 & 0.6390 & 0.1139 & 0.6266 & 0.8479 & 0.3650 & 0.5163 & 0.8378 & 0.5199 & 0.4372 & 0.2217 & 0.3816 & 0.4981 & 0.5852 & 0.1202 & 0.9480 & 0.4378 & 0.8647 & 0.6287 & 0.9998 & 0.6086 & 0.0844 & 0.3801 & 0.8506 & 0.6556 & 0.4510 & 0.8956 & 0.6127 & 0.7228 \\
0.2848 & 0.9111 & 0.8277 & 0.8023 & 0.9012 & 0.5401 & 0.4333 & 0.6249 & 0.8241 & 0.8928 & 0.0058 & 0.8497 & 0.8876 & 0.9786 & 0.5387 & 0.7849 & 0.3091 & 0.3264 & 0.9208 & 0.9892 & 0.3798 & 0.2190 & 0.5677 & 0.4874 & 0.6677 & 0.1778 & 0.0596 & 0.2802 & 0.4200 & 0.8778 & 0.8643 & 0.2197 & 0.5313 & 0.8128 & 0.1445 & 0.9382 & 0.6401 & 0.4825 & 0.3008 & 0.6681 \\
0.5950 & 0.6996 & 0.1910 & 0.4242 & 0.5406 & 0.3111 & 0.8842 & 0.7386 & 0.8280 & 0.3565 & 0.6103 & 0.2763 & 0.1987 & 0.8486 & 0.6505 & 0.2708 & 0.1209 & 0.6618 & 0.4982 & 0.4899 & 0.9797 & 0.5222 & 0.8879 & 0.2295 & 0.6480 & 0.7061 & 0.2687 & 0.9852 & 0.2399 & 0.6624 & 0.0369 & 0.4063 & 0.8006 & 0.2441 & 0.3705 & 0.6204 & 0.1320 & 0.4427 & 0.7981 & 0.1788 \\
0.9622 & 0.7252 & 0.4425 & 0.7289 & 0.4320 & 0.0712 & 0.3931 & 0.8051 & 0.2934 & 0.5464 & 0.8011 & 0.6223 & 0.3954 & 0.0506 & 0.7266 & 0.2278 & 0.9158 & 0.1176 & 0.2776 & 0.6949 & 0.3990 & 0.4334 & 0.8429 & 0.0856 & 0.4334 & 0.8314 & 0.9867 & 0.6088 & 0.5977 & 0.8754 & 0.5447 & 0.6299 & 0.7388 & 0.8844 & 0.6224 & 0.2828 & 0.4528 & 0.3118 & 0.7956 & 0.5505 \\
0.1858 & 0.2299 & 0.3934 & 0.4984 & 0.5427 & 0.1820 & 0.1790 & 0.0672 & 0.3094 & 0.3467 & 0.2330 & 0.5884 & 0.9922 & 0.4662 & 0.0945 & 0.3210 & 0.1355 & 0.1478 & 0.6525 & 0.4114 & 0.4402 & 0.7413 & 0.8988 & 0.0674 & 0.1398 & 0.0348 & 0.7722 & 0.2537 & 0.4794 & 0.4675 & 0.9976 & 0.5553 & 0.1417 & 0.7126 & 0.9976 & 0.2052 & 0.6522 & 0.0553 & 0.7811 & 0.9599 \\
0.1930 & 0.5761 & 0.8266 & 0.8090 & 0.7124 & 0.0930 & 0.6333 & 0.9508 & 0.5230 & 0.6228 & 0.9325 & 0.9635 & 0.4024 & 0.3257 & 0.8776 & 0.8296 & 0.3321 & 0.0198 & 0.9173 & 0.0348 & 0.1568 & 0.0704 & 0.9390 & 0.8884 & 0.7519 & 0.7578 & 0.4754 & 0.1326 & 0.8985 & 0.1413 & 0.5110 & 0.1276 & 0.4379 & 0.3781 & 0.5173 & 0.4391 & 0.8270 & 0.7538 & 0.3511 & 0.5960 \\
0.3416 & 0.8106 & 0.6769 & 0.3565 & 0.0167 & 0.4635 & 0.6240 & 0.4976 & 0.3253 & 0.7966 & 0.7633 & 0.0859 & 0.6589 & 0.6302 & 0.0144 & 0.8222 & 0.8975 & 0.9643 & 0.5098 & 0.2928 & 0.3260 & 0.8473 & 0.8154 & 0.2332 & 0.2418 & 0.9571 & 0.6809 & 0.5450 & 0.9347 & 0.0681 & 0.8735 & 0.1692 & 0.3504 & 0.2489 & 0.9905 & 0.0273 & 0.3081 & 0.1319 & 0.0543 & 0.8086 \\
0.9329 & 0.4038 & 0.2076 & 0.0732 & 0.8009 & 0.0093 & 0.3279 & 0.7551 & 0.8318 & 0.7459 & 0.8264 & 0.5005 & 0.9013 & 0.2303 & 0.2943 & 0.5707 & 0.4996 & 0.9704 & 0.9742 & 0.8014 & 0.3141 & 0.6799 & 0.0014 & 0.8616 & 0.6505 & 0.3429 & 0.4169 & 0.8278 & 0.8179 & 0.7142 & 0.0702 & 0.0010 & 0.4785 & 0.2529 & 0.2265 & 0.8762 & 0.4024 & 0.3559 & 0.7087 & 0.9845 \\
0.3907 & 0.9884 & 0.3181 & 0.5910 & 0.1425 & 0.9150 & 0.8030 & 0.7424 & 0.8103 & 0.1255 & 0.5735 & 0.5216 & 0.9954 & 0.5799 & 0.1799 & 0.5718 & 0.6153 & 0.1239 & 0.1973 & 0.3465 & 0.8945 & 0.1367 & 0.0031 & 0.7117 & 0.8574 & 0.6382 & 0.3801 & 0.8370 & 0.7089 & 0.3080 & 0.9875 & 0.4182 & 0.5874 & 0.7672 & 0.3980 & 0.6101 & 0.8842 & 0.3959 & 0.9929 & 0.8859 \\
0.2732 & 0.0900 & 0.1338 & 0.9102 & 0.4785 & 0.6427 & 0.9995 & 0.8311 & 0.5570 & 0.8224 & 0.7926 & 0.0902 & 0.6532 & 0.6032 & 0.9263 & 0.2860 & 0.5831 & 0.4674 & 0.1112 & 0.0833 & 0.2470 & 0.8584 & 0.0875 & 0.8728 & 0.0844 & 0.3430 & 0.2133 & 0.8333 & 0.7432 & 0.6712 & 0.9227 & 0.4885 & 0.1458 & 0.0499 & 0.6966 & 0.2036 & 0.7006 & 0.8855 & 0.1625 & 0.2138 \\
0.1519 & 0.3209 & 0.6715 & 0.1938 & 0.2568 & 0.0014 & 0.9810 & 0.1565 & 0.2630 & 0.0252 & 0.3290 & 0.9047 & 0.1084 & 0.5999 & 0.0682 & 0.6991 & 0.6983 & 0.6567 & 0.2974 & 0.5111 & 0.3107 & 0.1998 & 0.2607 & 0.9380 & 0.9721 & 0.2165 & 0.3829 & 0.2037 & 0.8997 & 0.6524 & 0.5643 & 0.1599 & 0.9053 & 0.6853 & 0.0646 & 0.5199 & 0.2419 & 0.0212 & 0.1136 & 0.0346 \\
0.3971 & 0.5114 & 0.5710 & 0.4324 & 0.3691 & 0.0304 & 0.1270 & 0.4573 & 0.6806 & 0.4144 & 0.2235 & 0.8844 & 0.0361 & 0.4484 & 0.5811 & 0.7963 & 0.0293 & 0.2902 & 0.3964 & 0.3668 & 0.4089 & 0.6073 & 0.0228 & 0.1397 & 0.0315 & 0.7862 & 0.0297 & 0.5444 & 0.0652 & 0.5310 & 0.4315 & 0.6668 & 0.6402 & 0.6203 & 0.7477 & 0.0538 & 0.7598 & 0.8441 & 0.9129 & 0.4511 \\
0.3747 & 0.0606 & 0.1698 & 0.7492 & 0.6618 & 0.2085 & 0.2322 & 0.6181 & 0.2337 & 0.7314 & 0.3124 & 0.4390 & 0.6181 & 0.0354 & 0.6372 & 0.4416 & 0.5279 & 0.7545 & 0.4208 & 0.7395 & 0.7080 & 0.5430 & 0.4241 & 0.3939 & 0.8354 & 0.7231 & 0.4723 & 0.8749 & 0.3359 & 0.7151 & 0.3378 & 0.0179 & 0.1629 & 0.7467 & 0.4204 & 0.8622 & 0.2909 & 0.2881 & 0.4817 & 0.0138 \\
0.1311 & 0.7257 & 0.1477 & 0.0392 & 0.1696 & 0.4550 & 0.0236 & 0.9322 & 0.4564 & 0.7814 & 0.5845 & 0.7817 & 0.5671 & 0.5138 & 0.6513 & 0.4462 & 0.0321 & 0.5581 & 0.3115 & 0.5247 & 0.1436 & 0.1623 & 0.3411 & 0.9806 & 0.8357 & 0.2788 & 0.3334 & 0.1210 & 0.0043 & 0.5048 & 0.7207 & 0.1197 & 0.5659 & 0.9773 & 0.8113 & 0.4429 & 0.2774 & 0.2503 & 0.8518 & 0.4737 \\
0.4350 & 0.5566 & 0.4761 & 0.9463 & 0.2788 & 0.1273 & 0.6074 & 0.8351 & 0.3846 & 0.3673 & 0.8299 & 0.1480 & 0.9620 & 0.4077 & 0.8646 & 0.4657 & 0.8271 & 0.4278 & 0.6938 & 0.8045 & 0.8713 & 0.0057 & 0.5414 & 0.6448 & 0.4350 & 0.5566 & 0.4761 & 0.9463 & 0.2788 & 0.1273 & 0.6074 & 0.8351 & 0.3846 & 0.3673 & 0.8299 & 0.1480 & 0.0061 & 0.4884 & 0.8099 & 0.9512
\end{bmatrix}$$

References

[1] Y. Xu, W. Zhang, W. Liu, Distributed dynamic programming-based approach for economic dispatch in smart grid, IEEE Trans. Ind. Inf. 11 (1) (2015) 166–175.

[2] Y. Zhang, D. Gong, N. Geng, X. Sun, Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects, Appl. Soft Comput. 18 (2014) 248–260.

[3] K.T. Chatuverdi, M. Pandit, L. Srivastava, Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch, IEEE Trans. Power Syst. 23 (2008) 1079–1087.

[4] F. Meng, X. Chen, Y. Zhang, Consistency-based linear programming models for generating the priority vector from interval fuzzy preference relations, Appl. Soft Comput. 41 (2016) 247–264.

[5] A. Liefooghe, S. Verel, J.-K. Hao, A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming, Appl. Soft Comput. 16 (2014) 10–19.

[6] V. Sharma, R. Jha, R. Naresh, Optimal multi-reservoir network control by augmented Lagrange Programming neural network, Appl. Soft Comput. 7 (2007) 783–790.

[7] A. Theerthamalai, S. Maheswarapu, An effective non-iterative "λ-logic based" algorithm for economic dispatch of generators with cubic fuel cost function, Int. J. Electrical Power Energy Syst. 32 (2010) 539–542.

[8] A. heydari, S.N. Balakrishnan, Global optimality of approximate dynamic programming and its use in non-convex function minimization, Appl. Soft Comput. 24 (2014) 291–303.

[9] P. Chen, H. Chang, Large-scale economic dispatch by genetic algorithm, IEEE Trans. Power Syst. 0 (4) (1995) 1919–1926.

[10] H. Zhang, D. Yue, X. Xie, S. Hu, S. Weng, Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch, Appl. Soft Comput. 34 (2015) 312–323.

[11] P. Savsani, R.L. Jhala, V. Savsani, Effect of hybridizing biogeography-based optimization (BBO) technique with artificial immune algorithm (AIA) and ant colony optimization (ACO), Appl. Soft Comput. 21 (2014) 542–553.

[12] D. Aydin, S. Özyön, Solution to non-convex economic dispatch problem with valve point effects by incremental artificial bee colony with local search, Appl. Soft Comput. 13 (2013) 2456–2466.

[13] V. Dieu, P. Schegner, Augmented Lagrange Hopfield network initialized by quadratic programming for economic dispatch with piecewise quadratic cost functions and prohibited zones, Appl. Soft Comput. 13 (2013) 292–301.

[14] A. Bhattacharya, P. Chattopadhaay, Solving economic dispatch load dispatch problems using hybrid differential evolution, Appl. Soft Comput. 11 (2011) 2526–2537.

[15] M. Al-Betar, M. Awadallah, A. Khader, A. Bolaji, Tournament-based harmony search algorithm for non-convex economic dispatch problem, Appl. Soft Comput. 47 (2016) 449–459.

[16] J. Kennedy, C. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks 4 (1995) 1942–1948.

[17] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science (1995) 39–43.

[18] Z. Zhan, J. Zhang, Y. Li, Y. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 6 (2011) 832–847.

[19] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024.

[20] Y. Leung, Y. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization, IEEE Trans. Evol. Comput. 5 (1) (2011) 41–53.

[21] M. Basu, Modified particle swarm optimization for nonconvex economic dispatch problems, Int. J. Electrical Power Energy Syst. 69 (2015) 304–312.

[22] P. Subbaraj, R. Rengaraj, S. Salivahanan, T. Senthilkumar, Parallel particle swarm optimization with modified stochastic acceleration factors for solving large scale economic dispatch problem, Int. J. Electrical Powe Energy Syst. 32 (2010) 1014–1023.

[23] J. Park, Y. Jeong, J. Shin, An Improved particle swarm optimization for nonconvex economic dispatch problems, IEEE Trans. Power Syst. 25 (1) (2010) 156–166.

[24] S.H. Ling, C. Iu, K. Chan, K. Lam, B. Yeung, F. Leung, Hybrid particle swarm optimization with wavelet mutation and its industrial applications, IEEE Trans. Syst. Man Cybern. 38 (3) (2008) 743–763.

[25] J. Sun, V. Palade, X. Wu, W. Fang, Z. Wang, Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization, IEEE Trans. Power Syst. 13 (1) (2014) 519–526.

[26] C. Kuo, A Novel coding scheme for practical economic dispatch by modified particle swarm approach, IEEE Trans. Power Syst. 23 (4) (2008) 1825–1835.

[27] I. Selvakumar, K. Thanushkodi, A new particle swarm optimization solution to nonconvex economic dispatch problems, IEEE Trans. Power Syst. 22 (1) (2007) 42–51.

[28] A. Selvakumar, K. Thanushkodi, Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems, Int. J. Electric Power Syst. Res. 78 (2008) 2–10.

[29] J. Cai, X. Ma, L. Li, H. Peng, Chaotic particle swarm optimization for economic dispatch considering the generator constraints, Energy Conversat. Manage. 48 (2007) 645–653.

[30] S. Chakraborty, T. Senjyu, A. Saber, A. Yona, T. Funabashi, A novel particle swarm optimization method based on quantum mechanics computation for thermal economic load dispatch problem, IEEJ Trans. Electr. Electron. Eng. 7 (2012) 461–470.

[31] J.J. Liang, B.Y. Qu, P.N. Suganthan, Q. Chen, Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-parameter Single Objective Optimization, Zhengzhou University/Nanyang Technological University, Zhengzhou, China/Singapore, 2014, Technical Report.

[32] Q. Chen, B. Liu, Q. Zhang, J.J. Liang, P.N. Suganthan, B.Y. Qu, Problem definition and evaluation criteria for CEC 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization, in: IEEE C. Evol. Comput, Sendai, Japan, 2014, Special session.

[33] X.-G. Zhou, G.-J. Zhan, X.-H. Hao, D.-W. Xu, L. Yu, Enhance differential evolution using local Lipschits underestimate strategy for computationally expensive optimization problems, Appl. Soft Comput. 48 (2016) 169–181.

[34] A. Banitalebi, M. Aziz, Z. Aziz, A self–adaptive binary differential evolution algorithm for large scale binary optimization problems, Inf. Sci. 367–368 (2016) 487–511.

[35] M. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self- regulating particle swarm optimization algorithm, Swarm Evol. Comput. 28 (2016) 98–116.

[36] T. Ngo, A. Sadollah, J. Kim, A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems, J. Comput. Sci. 13 (2016) 68–82.

[37] A. Yadav, K. Deep, J. Kim, A. Nagar, Gravitational swarm optimizer for global optimization, Swarm Evol. Comput. 31 (2016) 64–89.

[38] W. Elsayed, E. El-Saadany, A fully decentralized approach for solving the economic dispatch problem, IEEE Trans. Power Syst. 30 (4) (2015) 2179–2189.

[39] A. Bhattacharya, P. Chattopadhyay, Biogeography-based optimization for different economic load dispatch problems, IEEE Trans. Power Syst. 25 (2) (2010) 1064–1077.

[40] I. Ciornei, E. Kyriakide, A GA-API solution for the economic dispatch of generation in power system operation, IEEE Trans. Power Syst. 27 (1) (2012) 233–242.

[41] T. Ding, R. Bo, F. Li, H. Sun, A bi-level branch and bound method for economic dispatch with disjoint prohibited zones considering network losses, IEEE Trans. Power Syst. 30 (6) (2015) 2841–2855.

[42] R. Azizipanah-Abarghooee, T. Niknam, M. Gharibzadeh, F. Golestaneh, Robust, fast and optimal solution of practical economic dispatch by a new enhanced gradient-based simplified swarm optimisation algorithm, IET Gener. Transm. Distrib. 7 (6) (2013) 620–635.

[43] T. Adhinarayanan, M. Sydulu, Efficient Lambda logic based optimization procedure to solve the large scale generator constrained economic dispatch problem, Int. J. Electrical Eng. Technol. 4 (3) (2009) 301–309.

[44] S. Wang, J. Chiou, C. Liu, Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm, IET Gener. Transm. Distrib. 1 (5) (2007) 703–793.

[45] R. Naresh, J. Dubey, J. Sharma, Two-phase neural network based modelling framework of constrained economic load dispatch, IEE Gener. Transm. Distrib. 151 (3) (2004) 373, 378.

[46] N. Singh, J. Dhillon, D. Kothari, Synergic predator-prey optimization for economic thermal power dispatch problem, Appl. Soft Comput. 43 (2016) 298–311.

[47] X. He, Y. Rao, J. Huang, A novel algorithm for economic load dispatch of power systems, Neurocomputing 171 (2016) 1454–1461.

[48] S.M. Guo, J.S.H. Tsai, C.C. Yang, P.H. Hsu, A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CECbenchmark set, Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015 (2015) 1003–1010.

[49] N. Awad, M.Z. Ali, R.G. Reynolds, A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization, Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 2015 (2015) 1098–1105.

[50] J.L. Rueda, I. Erlich, Testing MVMO on learning-based real-parameter single objective benchmark optimization problems, Proceedings of The IEEE Congress on Evolutionary Computation (CEC), 2015 (2015) 1025–1032.

[51] J.L. Rueda, I. Erlich, MVMO for bound constrained single-objective computationally expensive numerical optimization, Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC) (2015) 1011–1017.

[52] V. Berthier, Experiments on the CECexpensive optimization testbed, Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015 (2015) 1059–1066.

[53] L.T. Al Bahrani, J.C. Patra, Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1–7.

[54] L.T. Al Bahrani, J.C. Patra, Orthogonal PSO algorithm for economic dispatch of power under power grid constraints, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kowloon, Hong Kong, 2015, pp. 14–19.

[55] L.T. Al Bahrani, J.C. Patra, R. Kowalczyk, Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generrating units in smart power grid under power grid constraints, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, Canada, 2016, pp. 660–667.

[56] K.K. Mandal, S. Mandal, B. Bhattacharya, N. Chakraborty, Non-convex emission constrained dispatch using a new self-adaptive particle swarm optimization technique, Appl. Soft Comput. 28 (2015) 188–195.

[57] N. Pandit, A. Tripathi, S. Tapaswi, M. Pandit, An improved bacterial foraging algorithm for combined static/dynamic environmental economic dispatch, Appl. Soft Comput. 12 (2012) 3500–3513.

[58] D. Datta, Unit commitment problem with ramp rate constraint using a binary-real-coded genetic algorithm, Appl. Soft Comput. 13 (2013) 3873–3883.

[59] A.K. Barisal, R.C. Prusty, Large scale economic dispatch of power systems using oppositional invasive weed optimization, Appl. Soft Comput. 29 (2015) 122–137.

[60] P. Roy, S. Bhui, C. Paul, Solution of economic load dispatch using hybrid chemical reaction optimization approach, Appl. Soft Comput. 24 (2014) 109–125.

[61] Y.H. Shi, R.C. Eberhart, A modified particle swarm optimizer, Proceedings of the IEEE International Conference on Evolutionary Computation (1998) 69–73.

[62] R.C. Eberhart, Y.H. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, Proceedings of the IEEE International Conference on Evolutionary Computation (2000) 84–88.

[63] D. Bratton, J. Kennedy, Defining a standard for particle swarm optimization, Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2007), 2007 (2007) 120–127.

[64] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Proceedings of Swarm Intelligence (2007) 33–57.

[65] Z.-H. Ruan, Y. Yuan, Q.-X. Chen, C.-X. Zhang, Y. Shuai, H.-P. Tan, A new multi-function global particle swarm optimization, Appl. Soft Comput. 49 (2016) 279–291.

[66] T. Niknam, H. Mojarrad, H. Meymand, Non-smooth economic dispatch computation by fuzzy and self adaptive particle swarm optimization, Appl. Soft Comput. 11 (2011) 2805–2817.

[67] S. Grossman, Elementary Linear Algebra, Saunders College Publishing, USA, 1994.

[68] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, State University, USA,Western Connecticut, 2000.

Paper G: Energy 2018

Multi-gradient PSO algorithm for optimization of multimodal discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power gird

Article reference:	EGY12001
Journal:	Energy
Received at Editorial Office:	2 Sep 2017
Article revised:	1 Dec 2017
Article accepted for publication:	11 Dec 2017
DOI:	10.1016/j.energy.2017.12.052

Energy 147 (2018) 1070–1091

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid

Loau Tawfak Al-Bahrani [a, b, *], Jagdish Chandra Patra [a]

[a] Swinburne University of Technology, Melbourne, Australia
[b] University of Diyala, Iraq

ARTICLE INFO

Article history:
Received 2 September 2017
Received in revised form
1 December 2017
Accepted 11 December 2017
Available online 14 December 2017

Keywords:
Multi-gradient particle swarm optimization
Exploration and exploitation phases
Economic dispatch
Power constraints
Prohibited operating zones
Valve-point loading effects

ABSTRACT

Optimization of fuel cost function of large-scale thermal generating units under several constraints in smart power grid is a challenging problem. Because of these constraints, the fuel cost function becomes multimodal, discontinuous and non-convex. Although the global particle swarm optimization with inertia weight (GPSO-*w*) algorithm is a popular optimization technique, it is not capable of solving such complex problems satisfactory. In this paper, a novel multi-gradient PSO (MG-PSO) algorithm is proposed to solve such a challenging problem. In MG-PSO algorithm, two phases, called *Exploration* phase and *Exploitation* phase, are used. In the *Exploration* phase, the *m* particles are called *Explorers* and undergo multiple episodes. In each episode, the *Explorers* use a different negative gradient to explore new neighbourhood whereas in the *Exploitation* phase, the *m* particles are called *Exploiters* and they use one negative gradient that is less than that of the *Exploration* phase, to exploit a best neighborhood. This diversity in negative gradients provides a balance between global search and local search. The effectiveness of the MG-PSO algorithm is demonstrated using four (medium and large) power generation systems. Superior performance of the MG-PSO algorithm over several PSO variants in terms of several performance measures has been shown.

1. Introduction

Economic dispatch (ED) problem is one of the fundamental issues in power generation systems (PGSs) of smart power grid (SPG). Its objective is to allocate the load demand among the committed thermal generating units (TGUs) in the most economical manner, while satisfying all operational and physical power constraints, e.g., ramp rate limits, prohibited operating zones and valve-point loading effects. Under these constraints, the fuel cost function becomes discontinuous and non-convex with multiple local minima [1].

Evolutionary computation techniques (ECTs), e.g., population-based algorithms, have been proposed and developed by several researches to solve real-world complex optimization problems including ED problem. In ECTs, finding the optimum solution of a problem is based on two phases, namely *Exploration* and *Exploitation* phases. In the *Exploration* phase, a global search exploring all over the search space as much as possible is carried out to find promising neighbourhood(s). Whereas, in the *Exploitation* phase, a local search exploiting the best neighbourhood to fine-tune the search space is carried out to obtain the optimum solution. The best performance of an ECT is achieved when an appropriate balance between these two phases is maintained [2]. Focusing more on *Exploration* will lead to excessive search time because of wastage of time in searching over inferior neighbourhoods, whereas focusing more on *Exploitation* will cause loss of diversity, thereby possibly getting stuck into a local optimum.

One of such popular ECTs called the global particle swarm optimization (GPSO) algorithm has been proposed to boost the global search and local search abilities and to make a good balance between the *Exploration* and *Exploitation*, for solving ED problem

* Corresponding author. Swinburne University of Technology, Melbourne, Australia.
E-mail addresses: lalbahrani@swin.edu.au (L.T. Al-Bahrani), JPatra@swin.edu.au (J. Chandra Patra).

https://doi.org/10.1016/j.energy.2017.12.052
0360-5442/© 2017 Elsevier Ltd. All rights reserved.

[3,4]. It is easy to implement and has performed well on many optimization problems. The GPSO algorithm has the ability to quickly converge to the optimum solution [5]. However, in this algorithm, all of its particles share the swarm's best experience, i.e., global best and this may lead the particles to cluster around the global best. If the global best is located near a local minimum, escaping from it becomes hard, because loss of balance between local search guide (personal experience of each particle) and global search guide (global best) [6–8]. Thus, the GPSO algorithm suffers diversity loss near a local minimum.

The GPSO with inertia weight (GPSO-*w*) was proposed in Ref. [9] to improve the performance of GPSO algorithm by controlling the convergence tendency of the particles with a negative gradient of the inertia weight, *w*. However, when the fuel cost function is high-dimensional, e.g., 40 TGUs or more, and restricted by many power constraints, it may have several local minima. In such case, finding a global optimal with *w* becomes hard, because of loss of balance between local search guide and global search guide still remains.

Recently, some of the notable ECTs that have been applied to solve ED problem under several power constraints imposed on TGU and PGS are orthogonal PSO (OPSO) algorithm [10], PSO with modified stochastic acceleration factors (PSO-MSAF) [11], chaotic sequence and crossover PSO (CCPSO) algorithm [12], hybrid PSO with mutation (HPSOM) algorithm [13], hybrid PSO with wavelet mutation (HPSOWM) algorithm [13], improved random drift PSO (IRDPSO) algorithm [14], self-tuning IRDPSO (ST-IRDPSO) algorithm [14], simulated annealing PSO (SA-PSO) algorithm [15], anti-predatory PSO (APSO) [16], and Chaotic PSO (CPSO) algorithm [17]. In Ref. [18], performance of the RDPSO algorithm was compared with ten ECTs, i.e., genetic algorithm (GA), differential evolution (DE), ant colony search algorithm (ACSA), bee colony optimization (BCO), artificial immune system (AIS), firefly algorithm (FA), and APSO, hybrid gradient PSO (HGPSO), HPSOM and HPSOWM algorithms.

Some other groups of ECTs are also used to compare with GPSO algorithm and its variants by several researchers [19–28] to solve ED problem of small, medium and large PGSs. They are mixed-integer quadratically constrained quadratic programming (MIQCQP) [19], Lambda logic (λ-logic) [20], synergic predator-prey optimization (SPPO) algorithm [21], modified symbiotic organisms search (MSOS) algorithm [22], fuzzy adaptive chaotic ant swarm optimization with sequential quadratic programming (FCASO-SQP) algorithm [23], chaotic bat algorithm (CBA) [24], greedy randomized adaptive search procedure (GRASP) algorithm [25], crisscross optimization algorithm [26], θ-modified bat algorithm (θ-MBA) [27], and root tree optimization (RTO) algorithm [28].

In the recent years, the gradient method is integrated and combined with few optimization techniques to create hybrid optimization techniques. This combination is used to achieve faster convergence without getting trapped into local minima. The gradient method helps particles to move faster toward optimum solution, whereas the optimization algorithm controls the movement of the particles from falling into local minimum. Some of the recently proposed such techniques are HGPSO algorithm [29], enhanced gradient simplified swarm optimization algorithm (EGSSOA) [30], and gradient-based Jaya algorithm [31]. Whereas, in the proposed MG-PSO algorithm, multiple negative gradients are used by *m* particles while searching for a global optimum. In addition, multiple gradients help to prevent the global best particle to fall in a local minimum.

In MG-PSO algorithm two phases are used, i.e., *Exploration* phase and *Exploitation* phase. In *Exploration* phase, a particle is called an *Explorer*. The *Explorers* operate in several episodes. In each episode, the *Explorers* use a different negative gradient to explore a new neighbourhood. *Explorers* enhance global search ability of the MG-PSO algorithm. At the end of *Exploration* phase, the *Explorers* provide a search boundary which becomes the new search space in the *Exploitation* phase. In the *Exploitation* phase, a particle is called an *Exploiter*. *Exploiters* use one negative gradient which is less than that of the *Exploration* phase to exploit the best neighborhood. The small negative gradient leads to small incremental change in the velocity and position vectors during updating process. This helps the particles to move steadily towards optimal solutions. Thus, *Exploiters* enhance local search ability of MG-PSO algorithm. This diversity in negative gradients helps the best particle from falling into a local minimum. The combination of two phases provides a balance between *Exploration* and *Exploitation* in search space.

In a recent work, the effectiveness of the proposed MG-PSO algorithm has been shown in solving ED problem of small and medium PGSs under a few power constraints in SPG applications [32]. Whereas, in the current study, the MG-PSO algorithm is applied to solve ED problem of four (medium and large) PGSs, considering more constraints including valve-point loading effects. In addition, the mathematical analysis and theoretical justification of MG-PSO algorithm is provided. With extensive simulated experiments, superior performance of the MG-PSO algorithm has been shown in terms of convergence, consistency and accuracy compared to GPSO-*w* algorithm and several competitive ECTs.

The rest of the paper is organized as follows. Section 2 describes the ED problem under various power constraints. Explanation of the GPSO-*w* algorithm is presented in Section 3. Details of the proposed MG-PSO algorithm are provided in Section 4. In Section 5, application of the MG-PSO algorithm to ED problem in four PGSs is presented. Finally, conclusion of this study is provided in Section 6.

2. Problem formulation

Here, we explain the fuel cost function and various practical power constraints involved in this study.

2.1. Fuel cost function

The aim of ED problem is to guess the optimum arrangement of power generation of online TGUs in order to minimize the entire generation fuel cost subjected to the online TGUs and PGS constraints in SPG. Considering N_{gen} committed online TGUs, the fuel cost function of *j*th online TGU $F(P_j)$, $j = 1, 2, \ldots, N_{gen}$, in ($/h) is characterized by a quadratic function given by Ref. [33]:

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{1}$$

where P_j is the output active power in (MW) at a current time interval, and a_j, b_j, and c_j are fuel cost coefficients. Under valve-point loading (VPL) effects, sinusoidal functions are added to the quadratic fuel cost function (1) [34]. This makes the fuel cost function non-smooth with multiple modes as follows:

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 + \left| e_j \times \sin\left(f_j \times \left(P_{j,\min} - P_j\right)\right) \right| \tag{2}$$

where e_j and f_j are the coefficients reflecting VPL effects and $P_{j,min}$ is the minimum output active power of *j*th TGU. The symbol || corresponds to absolute value. The total fuel cost, F_{cost}, considering all online TGUs is given by:

$$F_{\text{cost}} = \sum_{j=1}^{N_{gen}} F(P_j) \tag{3}$$

Here, F_{cost} is the function that needs to be minimized.

2.2. Power constraints

Different practical power constraints imposed on the online TGUs and by PGS in SPG used in literature are stated below:

2.2.1. Power balance constraint

The total output active power of the committed online TGUs should be able to satisfy load demand and transmission network loss. The power balance constraint is given as:

$$P_{total} = P_D + P_L \tag{4}$$

where $P_{total} = \sum_{j=1}^{N_{gen}} P_j$, P_D is load demand in (MW) and P_L is transmission network loss in (MW).

2.2.2. Transmission network loss

The transmission network loss P_L is a critical constraint of the ED problem. Not only is it desired that the power loss incurred in the system be minimized along with the total fuel cost, but also the power generation system must generate enough power to satisfy the P_D as well as to compensate for the P_L. The P_L is given by:

$$P_L = \sum_{j=1}^{N_{gen}} \sum_{k=1}^{N_{gen}} P_j B_{jk} P_k \tag{5}$$

where B_{jk} are known as the loss coefficients or B-coefficients [34].

2.2.3. Transmission network loss mismatch

The P_L obtained from (4) is denoted by P_{L1} and is given by:

$$\mathrm{P_{L1} = P_{total} - P_D} \tag{6}$$

Let P_{L2} be the transmission network loss obtained from (5). The transmission network loss mismatch, $P_{L,mismatch}$ is given by:

$$\mathrm{P_{L,mismatch} = P_{L2} - P_{L1}} \tag{7}$$

When $P_{L,mismatch} = 0$, it implies that optimum P_{total} is found and the power balance constraint (4) is achieved.

2.2.4. Generation limits

Each TGU has a specified range within which its operation is stable. Therefore, it is desired that all the TGUs be run within their operation range in order to maintain system stability. The generation limits of the jth TGU is given by:

$$P_{j,min} \le Pj \le P_{j,max} \quad j = 1, 2, \ldots, N_{gen} \tag{8}$$

2.2.5. Ramp rate limits

The operating range of on-line TGU is restricted by its ramp rate limits (RRLs) due to its physical limitations [35,36]. For any sudden change in the P_D, TGU increase or decrease its generation in order to satisfy system stability. However, the TGU can change its generation only at a certain rate determined by its up-ramp and down-ramp rate. If a TGU is operating at a specific point, then its point of operation can be changed only up to a certain rate determined by the ramp rate. Therefore, a change in TGU output active power from one specific interval to the next cannot exceed a specified limit.

If power generation need to increase, then per unit time rate of increase $P_j - P_j^0$ must satisfy:

$$\mathrm{P_j - P_j^0 \le UR_j} \tag{9}$$

If power generation need to decrease, then per unit time rate of decrease $P_j^0 - P_j$ must satisfy:

$$\mathrm{P_j^0 - P_j \le DR_j} \tag{10}$$

where P_j^0 is the TGU output active power at the previous time interval. The UR_j and DR_j are the up-ramp and down-ramp limits of TGU j in (MW/h), respectively. By substituting (9) and (10) in (8), we obtain the following constraints.

$$max\left\{P_{j,min},\ \left(P_j^0 - DR_j\right)\right\} \le P_j \le min\left\{P_{j,max},\ \left(P_j^0 - UR_j\right)\right\} \tag{11}$$

where

$$P_{j,low} = max\left\{P_{j,min},\ \left(P_j^0 - DR_j\right)\right\}, \tag{12}$$

$$P_{j,high} = min\left\{P_{j,max},\ \left(P_j^0 + UR_j\right)\right\}, \text{and} \tag{13}$$

$P_{j,low}$ and $P_{j,high}$ are the new lower and higher limits of jth TGU, respectively.

2.2.6. Prohibited operating zones

The physical limitations due to the steam valve operation or vibration in shaft bearing of TGU may result in the generation units operating within prohibited operating zones (POZs) [37]. The POZs make the fuel cost function discontinuous in nature. In such case, it is difficult to determine the shape of the cost curve under POZs through actual performance testing. In addition, if the TGU operates within the POZ range then it may lead to loss of the stability. Therefore, in this study, these regions are usually avoided during generation. By using (8), the feasible operating zones (FOZs) of the jth TGU are given by:

$$\begin{aligned} &P_{j,min} \le P_j \le P_{j,1}^{l} \\ &P_{j,k-1}^{u} \le P_j \le P_{j,k}^{l} \quad k = 2, 3, \ldots, \mathrm{N}_{j,PZ} \\ &P_{j,N_{j,PZ}}^{u} \le P_j \le P_{j,max} \end{aligned} \tag{14}$$

where $P_{j,k}^{l}$and $P_{j,k}^{u}$ are the lower and upper bound of the kth POZs of the jth TGU, and $N_{j,PZ}$ is number of POZs of the jth TGU. Incorporating these power constraints in (11)–(14), we get the final set of inequality power constraints imposed on TGU as given below.

$$\begin{aligned} &P_{j,low} \le P_j \le P_{j,1}^{l}, \\ &P_{j,k-1}^{u} \le P_j \le P_{j,k}^{l} \quad k = 2, 3, \ldots, \mathrm{N}_{j,PZ} \\ &P_{j,N_{j,PZ}}^{u} \le P_j \le P_{j,high} \end{aligned} \tag{15}$$

Equation (15) gives the final set of the inequality power constraints imposed on jth TGU in terms of new lower and upper generation limits with RRLs and FOZs and avoiding all POZ$_S$. Thus, all TGUs will have a set of operation limits that satisfies all the power constraints.

3. The GPSO-*w* algorithm

The optimization mechanism in a global particle swarm optimization with inertia weight, GPSO-*w* algorithm depends on the distribution of m particles in the swarm [9]. It is represented by a fully connected network, in which each particle has access to the information of the swarm population. Firstly, each particle flying in the multi-dimensional search space adjusts its flying trajectory based on two guides, its personal experience, $G_{pers,i}$ and its neighborhood's best experience, G_{best}. Secondly, when seeking an optimum, i.e., global solution, each particle learns from its own historical experience and its neighborhood's historical experience. In such a case, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. Therefore, the GPSO algorithm is named as global PSO, because the position of each particle is affected by the best-fit particle in the entire swarm. The following steps explain the mechanism of the GPSO-*w* algorithm.

Let us consider a swarm population with m particles ($m > 1$) searching for optimal solution (minimum) of an objective function $f(x)$ in a d-dimensional search space. Let total number of iterations is N_{iter}. The objective is to minimize the given $f(x)$. A particle, i ($i = 1, 2, \ldots, m$), has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i and are denoted by

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{16}$$

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{17}$$

```
Step 1: Initialization: Iteration, t = 0.
    for $i = 1, 2, \ldots, m$
    The $V_i$ and $X_i$ of $i$th particle are randomly initialized within a defined range of the search space and are denoted by $V_i(0)$ and $X_i(0)$, respectively.
    Initialize the personal position vector of particle $i$, $i = 1, 2, \ldots, m$, $G_{pers,i}(0)$ as follows:
```

$$G_{pers,i}(0) = X_i(0) \tag{18}$$

```
    Evaluate the $f(x)$ using $X_i(0)$.
    Determine the global best position vector, $G_{best}(0)$. It is the best position vector among all the $m$ personal position vectors. The $G_{best}(0)$ is denoted by
```

$$G_{best}(0) = [g_{b,1}, g_{b,2}, \ldots, g_{b,d}] \tag{19}$$

```
    end i loop

Step 2: Update:
    for $t = 1, 2, \ldots, N_{iter}$
        for $i = 1, 2, \ldots, m$
        Determine inertia weight, $w(t)$ as given below [38].
```

$$w(t) = -\frac{0.5}{N_{iter}} t + 0.9 \tag{20}$$

```
        Update $V_i$ and $X_i$ as follows:
```

$$\begin{aligned} V_i(t) &= w(t)\, V_i(t-1) + c_1 r_1(t)\left[G_{pers,i}(t-1) - X_i(t-1)\right] \\ &\quad + c_2 r_2(t)\left[G_{best}(t-1) - X_i(t-1)\right] \end{aligned} \tag{21}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{22}$$

```
        where $c_1$ and $c_2$ are positive coefficients, called acceleration constants which are commonly set to 2.0 [9]. The $r_1(t)$ and $r_2(t)$ are two randomly generated values with a uniform distribution in the range of [0,1].
        Evaluate $f(x)$ for particle $i$ using $X_i(t)$.
        Update $G_{pers,i}(t)$ as follows:
```

$$G_{pers,i}(t) = \begin{cases} X_i(t) & if \quad f(X_i(t)) \le f\left(G_{pers,i}(t-1)\right) \\ G_{pers,i}(t-1) & Otherwise \end{cases} \tag{23}$$

```
        end i loop
    Obtain $f(G_{best}(t))$ as follows:
    $f(G_{best}(t)) = min\{f(G_{pers,i}(t))\}$, $i = 1, 2, \ldots, m$
    Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
    end t loop

Step 3: End of iteration: $t = N_{iter}$
```

$$\text{Optimum solution} = G_{best}(N_{iter}) \text{ and optimum value} = f(G_{best}(N_{iter})) \tag{24}$$

4. Learning strategy and MG-PSO algorithm

Here, the details of the proposed MG-PSO algorithm and explanation of its learning strategy are provided.

4.1. Learning strategy

The learning strategy of MG-PSO algorithm depends on the following considerations. Consider a swarm population with m particles ($m > 1$) flying in a d-dimensional space searching for a solution, i.e., global optimum. Two fundamental phases, "*Exploration* and *Exploitation*" are used by the m particles. In *Exploration* phase, a particle is called *Explorer*. In each episode, the *Explorers,* i.e., m particles, use different negative gradient to explore new neighbourhood in a d-dimensional search space. *Explorers* enhance a global search ability of MG-PSO algorithm. The purpose of *Explorers* is

- To obtain new neighbourhoods within a d-dimensional search space.
- To obtain best neighborhood within a d-dimensional search space

In each episode, the *Explorers* obtain best position vector following its neighbourhood. Its neighborhood is obtained by taking "Floor" and "Ceil" of each element of the best position vector. These operations create a new neighborhood within d-dimensional search space that will be used in the *Exploitation* phase.

In *Exploitation* phase, a particle is called an *Exploiter*. The *Exploiters*, i.e., m particles, use one negative gradient which is less than that of the *Exploration* phase. The *Exploiters* enhance the local search ability of MG-PSO algorithm. The purpose of this phase is to obtain an optimal position by exploiting the *Exploiters* in the best neighborhood obtained from the *Exploration* phase.

4.2. The MG-PSO algorithm

In MG-PSO algorithm, N_{grad} number of negative gradients is used while the swarm population searches for an optimal solution. In *Exploration* phase, $N_{grad} - 1$ negative gradients are used and one negative gradient is used in *Exploitation* phase. In each episode, the inertia weight follows one negative gradient.

Let N_{iter} be number of iterations in MG-PSO algorithm. The number of iterations in *Exploration* phase is given by

$$N_{iter,xplore} = \gamma \times N_{iter} \tag{25}$$

where γ is a real and positive number in a range [0,1]. The number of iteration in *Exploration* phase is given by:

$$N_{iter,xploit} = (1–\gamma) \times N_{iter} \tag{26}$$

The initial and final values of the inertia weight for *k*th negative gradient ($k = 1, 2, \ldots, N_{grad}$) are denoted by $w_{ini,k}$ and $w_{fin,k}$, respectively. These values are real and positive numbers within a range [0,1] and $w_{ini,k} > w_{fin,k}$. The *k*th negative gradient ($k = 1, 2, \ldots, N_{grad} - 1$) in *Exploration* phase is given by:

$$grad_k = \frac{w_{fin,k} - w_{ini,k}}{N_{iter,xplre}} \tag{27}$$

In *Exploitation* phase, the negative gradient is given by:

$$grad_{N_{grad}} = \frac{w_{fin,N_{grad}} - w_{ini,N_{grad}}}{N_{iter,xploit}} \tag{28}$$

The N_{grad} gradients are selected such that (29) is satisfied.

$$|grad_1| > |grad_2| \cdots > \left| grad_{N_{grad}} \right| \tag{29}$$

The inertia weight for *k*th negative gradient ($k = 1, 2, \ldots, N_{grad}$) at iteration *t* is given by:

$$w_k(t) = grad_k \times t + w_{ini,k} \tag{30}$$

The flowchart of the MG-PSO algorithm is shown in Fig. 1 and the detailed steps explaining the MG-PSO algorithm are given in Appendix.

4.3. An illustrative example

In order to explain the mechanism of MG-PSO algorithm, an example of a 2-dimensional shifted function, $f(x,y) = (x - 2)^2 + (y + 3)^2 + 9$ is illustrated. It can be seen that *x* and *y* are shifted from the origin (0,0) by (2,–3). The optimum solution of the given function equals to 9 at $x = 2$ and $y = -3$. The aim of the MG-PSO algorithm is to find the values *x* and *y* for which the $f(x,y)$ is minimized. The MG-PSO algorithm was implemented using MATLAB in a personal computer with Intel (R) core (TM) 2 Duo CPU T6570 @ 2.1 GHz. RAM of 4GB and Windows 7, 64-bit operating system.

The MG-PSO algorithm was executed with $m = 6$, $N_{iter} = 60$, $\gamma = 0.4$, $N_{grad} = 4$. $N_{iter,xplore} = 24$ and $N_{iter,xploit} = 36$. The $w_{ini,k}$ and $w_{fin,k}$ are chosen within a range of [0,1] and are shown in Table 1. Selection of $grad_k$ ($k = 1, 2, \ldots, N_{grad}$) was done by trial and error method. The results of the *Exploration* phase with three episodes ($k = 1$, 2 and 3) are shown in Table 1. At the end of episodes, $f(G_{best,xplore})$ is found as 9.0566 that is corresponding to episode #1. The $BEST(G_{best,xplore})$ corresponding to episode #1 is found as (2.1625,-2.8262). The range of new search space (neighborhood) is obtained by taking “Floor” and “Ceil” of 2.1625, i.e. [2,3], and “Floor” and “Ceil” of −2.8262, i.e., [-3,-2]. Thus, the new search space is given by a range of *x* as [2,3] and a range of *y* as [-3,-2].

In *Exploitation* phase, the *Exploiters* navigate in the newly found search space, i.e. [2,3], [-3,-2], using one negative gradient ($k = 4$) that is less than that of *Exploration* phase. As shown in Table 1, at the end, the *Exploiters* obtain the optimum value $G_{best,xploit}(36)) = 9.0000$ and $G_{best,xploit}(36) = (2.0,-3.0)$ give the optimum value of $f(x,y)$ and the optimum solution of (x,y) as 9.0 and optimum solution as $(x,y) = (2.0, -3.0)$.

The movement of best particle $G^k_{best}(t)$ for three episodes, $k = 1$, 2 and 3 over 24 iterations are shown in Fig. 2(a), (b) and (c), respectively. The $G^k_{best}(t)$ follows its $grad_k$ in each episode.

This diversity in negative gradients makes the MG-PSO algorithm to obtain different solutions, i.e., $G^1_{best}(t)$, $G^2_{best}(t)$ and $G^3_{best}(t)$ which are close to optimum solution. This means that the global best particle prevents the swarm from falling into a local minimum.

Fig. 2(d), (e) and (f) illustrate variation of inertia weight with iteration for the episodes 1, 2 and 3, respectively. It can be seen that the inertia weights follow different negative gradients. The convergence characteristics, i.e., variation $f(G^k_{best}(t))$ with iteration are also shown in this figure. At the end of iteration the optimum values obtained from the three episodes are given by 9.0566, 11.3179 and 10.6136. The corresponding optimum solutions for the three episodes are given by (2.1625,-2.8262), (3.4993,-2.7352) and (1.3299,-1.9208), respectively (as shown in Fig. 2(a), (b) and (c) and Table 1).

Fig. 3(a) shows movement of the best particle $G_{best,xploit}(t)$ in *Exploitation* phase within the new search space range of *x* and *y* as [2,3] and [-3,-2], respectively. The variation of $f(G_{best,xploit})$ and inertia weight w_4 over 36 iterations are shown in Fig. 3(b). The $G_{best,xploit}$ gives the optimum solution $(x,y) = (2.0,-3.0)$ and optimum value of $f(x,y) = 9.0$.

4.4. Observations

Some of the important observations of the MG-PSO algorithm are as follows:

4.4.1. Observation 1

Due to use of a different negative gradient in each episode, in *Exploration* phase, the *Explorers* have ability to find a new neighbourhood within a *d*-dimensional search space. The global best particle is able to prevent the swarm from falling into a local minimum. In addition, the diversity in negative gradients enhances the local search ability of the *Exploiters* to obtain optimum solution, as shown in Figs. 2 and 3.

4.4.2. Observation 2

In case of GPSO-*w* algorithm (21), two guides, $G_{pers,i}$ and G_{best}, are used to update the velocity vector $V_i(t)$. This leads to loss of balance between global search and local search. However, in case of MG-PSO algorithm two phases are used. In the *Exploration* phase, using several episodes (each one with different gradient) a new search space (new neighborhood) is obtained. This search space is used in *Exploitation* phase to achieve the optimum solution. In this way a balance is maintained between the global and local search spaces.

4.4.3. Observation 3

The $w_{N_{grad}}(t)$ that follows negative gradient, $grad_{N_{grad}}$ in the *Exploitation* phase (Appendix) is used to diminish the contribution of $X_i(t-1)$ while updating $V_i(t)$, $i = 1, 2, \ldots, m$. As $t \to \infty$, assume that

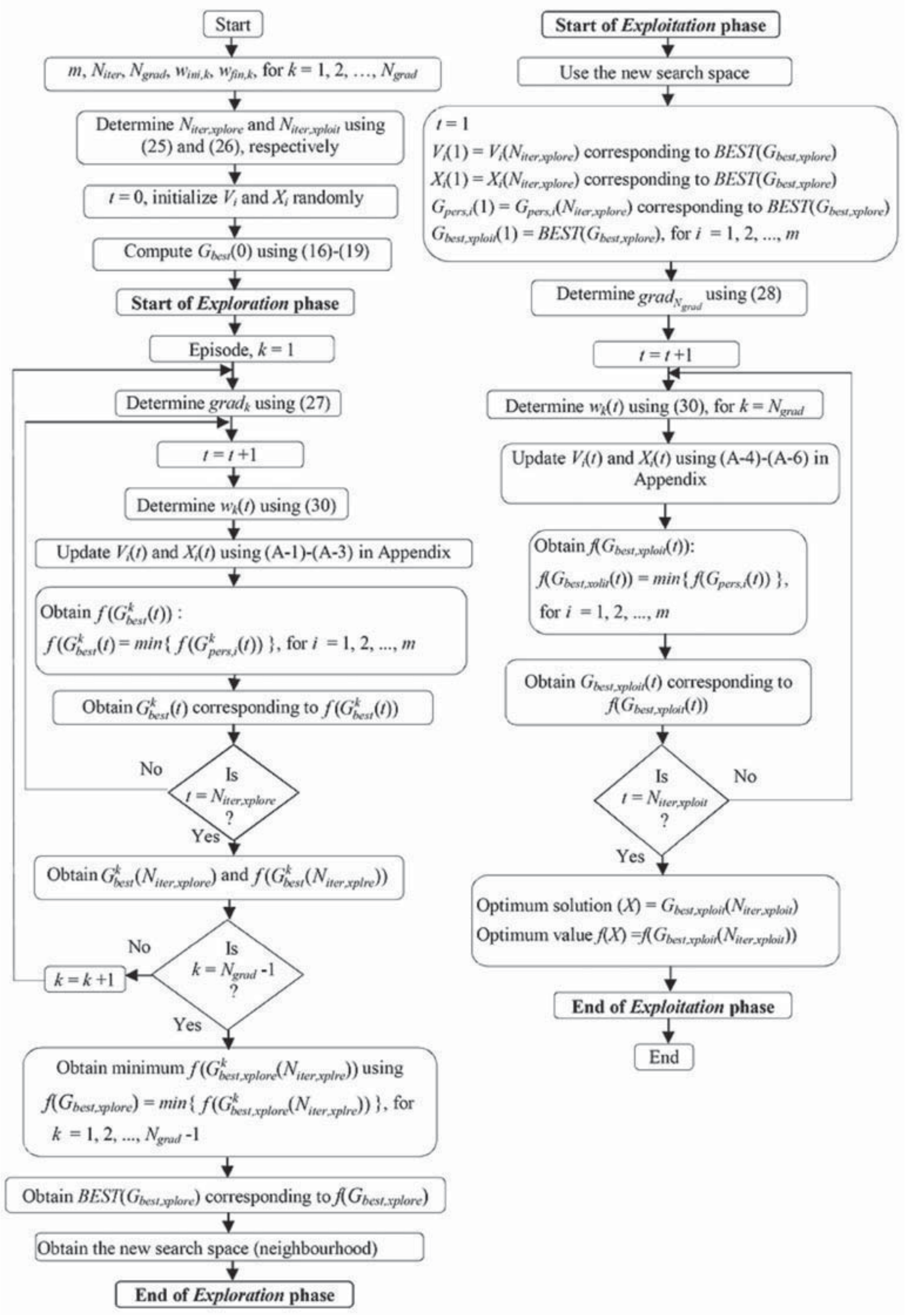

Start
m, N_{iter}, N_{grad}, $w_{ini,k}$, $w_{fin,k}$, for $k = 1, 2, \ldots, N_{grad}$
Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (25) and (26), respectively
$t = 0$, initialize V_i and X_i randomly
Compute $G_{best}(0)$ using (16)-(19)
Start of *Exploration* phase
Episode, $k = 1$
Determine $grad_k$ using (27)
$t = t + 1$
Determine $w_k(t)$ using (30)
Update $V_i(t)$ and $X_i(t)$ using (A-1)-(A-3) in Appendix
Obtain $f(G^k_{best}(t))$:
$f(G^k_{best}(t) = min\{ f(G^k_{pers,i}(t)) \}$, for $i = 1, 2, ..., m$
Obtain $G^k_{best}(t)$ corresponding to $f(G^k_{best}(t))$
No
Is $t = N_{iter,xplore}$?
Yes
Obtain $G^k_{best}(N_{iter,xplore})$ and $f(G^k_{best}(N_{iter,xplre}))$
No
Is $k = N_{grad}$ -1 ?
$k = k + 1$
Yes
Obtain minimum $f(G^k_{best,xplore}(N_{iter,xplre}))$ using
$f(G_{best,xplore}) = min\{ f(G^k_{best,xplore}(N_{iter,xplre})) \}$, for
$k = 1, 2, ..., N_{grad}$ -1
Obtain $BEST(G_{best,xplore})$ corresponding to $f(G_{best,xplore})$
Obtain the new search space (neighbourhood)
End of *Exploration* phase
Start of *Exploitation* phase
Use the new search space
$t = 1$
$V_i(1) = V_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$
$X_i(1) = X_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$
$G_{pers,i}(1) = G_{pers,i}(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$
$G_{best,xploit}(1) = BEST(G_{best,xplore})$, for $i = 1, 2, ..., m$
Determine $grad_{N_{grad}}$ using (28)
$t = t + 1$
Determine $w_k(t)$ using (30), for $k = N_{grad}$
Update $V_i(t)$ and $X_i(t)$ using (A-4)-(A-6) in Appendix
Obtain $f(G_{best,xploit}(t))$:
$f(G_{best,xolit}(t)) = min\{ f(G_{pers,i}(t)) \}$,
for $i = 1, 2, ..., m$
Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
Is $t = N_{iter,xploit}$?
No
Yes
Optimum solution $(X) = G_{best,xploit}(N_{iter,xploit})$
Optimum value $f(X) = f(G_{best,xploit}(N_{iter,xploit}))$
End of *Exploitation* phase
End

Fig. 1. Flowchart of MG-PSO algorithm.

Table 1
Set of parameters and performance of MG-PSO algorithm with $d = 2$, $m = 6$, $N_{iter} = 60$.

Set of parameters	*Exploration* phase			*Exploitation phase*
	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$w_{ini,k}$	**1.0**	0.90	0.80	0.50
$w_{fin,k}$	**0.1**	0.15	0.20	0.35
$grad_k$	**−0.0375**	−0.0313	−0.0250	−0.0042
G^k_{best}	**(2.1625, -2.8262)**	(3.4993, −2.7352	(1.3299, −1.9208)	**(2.0000,-3.0000)**
$f(G^k_{best})$	**9.0566**	11.3179	10.6136	**9.0000**

The bold numbers indicate the best solution of ith particle in a swarm during Exploration and Exploitation phases.

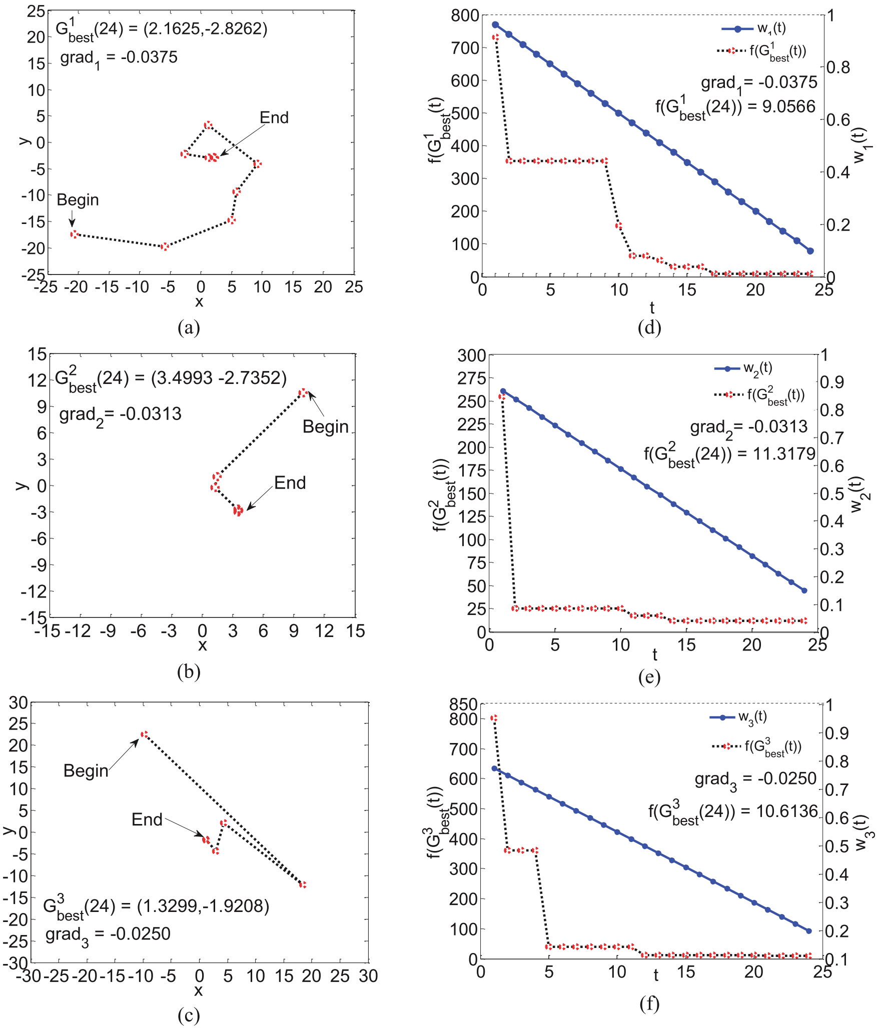
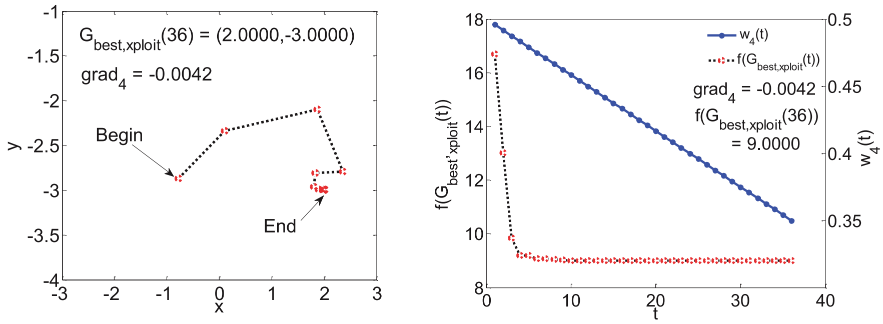


Fig. 2. Movement of best particle in *Exploration* phase over 24 iterations based on three different negative gradients in three episodes, (a), (b) and (c): movement of G^k_{best} over 24 iterations, (d), (e) and (f): the change of $f(G^k_{best}(t))$ and $w_k(t)$ with iteration t.



(a) (b)

Fig. 3. Movement of the best particle in *Exploitation* phase with one negative gradient, (a): movement of G_{best} over 36 iterations, (b): the change of $f(G_{best,xploit}(t))$ and $w_4(t)$ with iteration t.

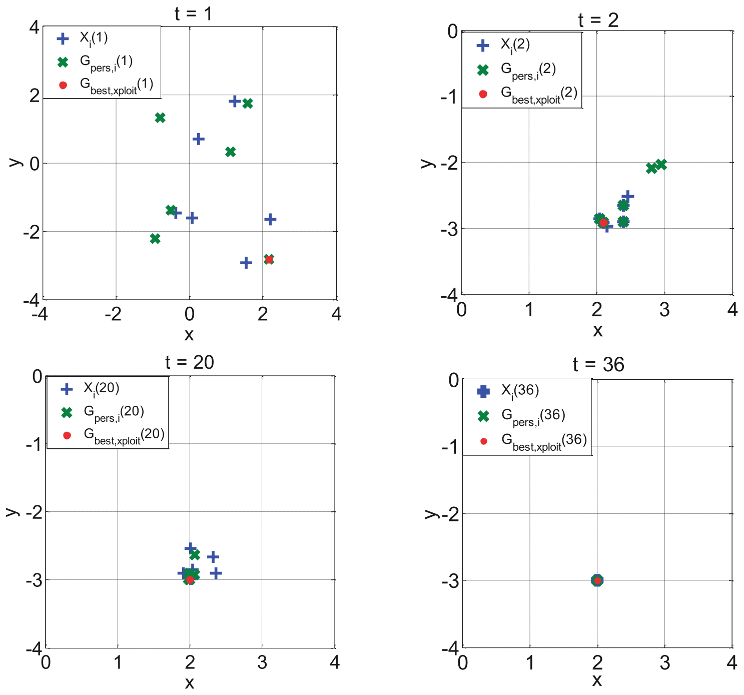


Fig. 4. Performance of MG-PSO algorithm in *Exploitation* phase showing movement of X_i, $G_{pers,i}$, (i = 1, 2, …, 6) and $G_{best,xploit}$ over 36 iterations to obtain a solution of (2.0,-3.0).

the algorithm has converged. In such case,

$$\begin{aligned} \mathit{lim}_{t\to\infty} X_i(t-1) &= G_{pers,i}(t-1) \\ \mathit{lim}_{t\to\infty} X_i(t-1) &= G_{best,xploit}(t-1) \end{aligned} \tag{31}$$

Then, Equation (A-4) becomes

$$\lim_{t\to\infty} V_i(t) = w_{N_{grad}}(t)\, V_i(t-1) = 0 \tag{32}$$

which implies that

$$\lim_{t\to\infty} X_i(t) = X_i(t-1) \tag{33}$$

Thus,

$$\lim_{t\to\infty} X_i(t) = \lim_{t\to\infty} G_{pers,i}(t) = \lim_{t\to\infty} G_{best,xploit}(t) \tag{34}$$

Thus, when iteration becomes large and the algorithm has converged, all the position vectors X_i and personal vectors $G_{pers,i}$, i = 1, 2, …, m, move towards the best position vector, $G_{best,xploit}$. Fig. 4 shows performance of MG-PSO algorithm in the *Exploitation* phase at different iterations. At t = 36, as the algorithm converges, it

Table 2
Specifications and power constraints for PGS-1.

TGU$_j$	a_j ($/h)	b_j ($/MWh)	c_j ($/MW^{2}h)	e_j ($/h)	f_j (rad/MW)	$P_{j,min}$ (MW)	$P_{j,max}$ (MW)
1	0.00028	8.10	550	300	0.035	0	680
2	0.00056	8.10	309	200	0.042	0	360
3	0.00056	8.10	307	200	0.042	0	360
4	0.00324	7.74	240	150	0.063	60	180
5	0.00324	7.74	240	150	0.063	60	180
6	0.00324	7.74	240	150	0.063	60	180
7	0.00324	7.74	240	150	0.063	60	180
8	0.00324	7.74	240	150	0.063	60	180
9	0.00324	7.74	240	150	0.063	60	180
10	0.00284	8.60	126	100	0.084	40	120
11	0.00284	8.60	126	100	0.084	40	120
12	0.00284	8.60	126	100	0.084	55	120
13	0.00284	8.60	126	100	0.084	55	120

Table 3
Set of parameters used in MG-PSO and GPSO-*w* algorithms for PGS-1.

Set of parameters	MG-PSO		GPSO-w
	Exploration phase	*Exploitation phase*	
	$k = 1$	$k = 2$	
γ	0.30	0.3	–
c_1, c_2	2.05	2.05	2.00
N_{iter}	150	350	500
$w_{ini,k}$	0.90	0.45	0.90
$w_{fin,k}$	0.10	0.20	0.40
$grad_k$	-5.33×10^{-03}	7.14×10^{-04}	-1.00×10^{-03}

can be seen that X_i, $G_{pers,i}$ ($i = 1, 2, \ldots, 6$) and $G_{best,xploit}$, all converge to the optimum solution (2.00,-3.00).

5. Application of MG-PSO algorithm to ED problem

Here we illustrate the simulation results carried out on four PGSs with several TGUs and SPG constraints.

5.1. Performance measures

To study the accuracy, consistency and robustness of different algorithms in solving ED problem, several fitness values as illustrated below are considered. Every algorithm is executed over N_{run} runs each with N_{iter} iterations.

1. Ensemble average cost (F_{cost}): At each iteration, it is the average cost value obtained from N_{run} independent runs.
2. Minimum fuel cost (F_{min}): Defined as the minimum of the optimized F_{cost} values obtained from N_{run} independent runs.
3. Maximum fuel cost (F_{max}): Defined as the maximum of the optimized F_{cost} values obtained from N_{run} independent runs.

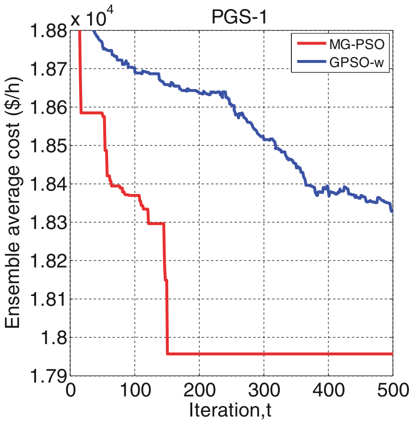


Fig. 5. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-1.

4. Mean fuel cost (F_{mean}): Defined as the average of the optimized F_{cost} values obtained from N_{run} independent runs.
5. Standard deviation (σ): The σ is the standard deviation of the optimized F_{cost} values obtained from N_{run} independent runs.
6. Range (R): The range (R) is defined as the difference between F_{max} and F_{min}.
7. Average execution time (AET): It is the time consumed by an algorithm after convergence, averaged over N_{run} independent runs.

5.2. Test case 1: power generation system-1 (PGS-1)

The PGS-1 is a medium-scale PGS [25], and consists of 13 TGUs.

Table 4
Comparison of cost performance between MG-PSO algorithm and other 8 ECTs for PGS-1.

Sl.No.	Algorithm	Min. Cost F_{min} ($/h)	Max. Cost F_{max} ($/h)	Mean Cost F_{mean} ($/h)	σ ($/h)	R ($/h)	AET (sec)
1	IRDPSO [14]	17,965.8480	NA	17,972.8090	0.8326	NA	2.26
2	ST-IRDPSO [14]	17,963.8300	NA	17,966.5700	3.3070	NA	2.28
3	MSOS [22]	17,963.8292	17,963.8292	17,963.8292	**6.8×10^{-12}**	**0.0**	**0.81**
4	FCASO-SQP [23]	17,964.0800	NA	18,001.9600	NA	NA	19.62
5	CBA [24]	17,963.8300	17,995.2256	17,965.4889	6.8730	31.3956	0.97
6	C-GRASP-SaDE [25]	17,960.3930	17,968.8680	17,966.1060	2.7010	8.4750	NA
7	RTO [28]	17,969.8024	18,204.6303	18,056.9358	NA	234.8279	NA
8	GPSO-*w*	18,047.1192	18,531.1387	18,326.4056	145.2735	484.0195	11.95
9	MG-PSO	17,955.8802	17,956.2793	**17,955.9948**	0.1085	0.3991	17.36

The bold numbers indicate the best solution found by corresponding algorithm.

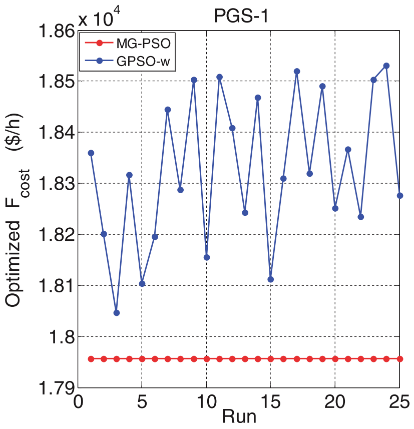


Fig. 6. Comparison of optimized cost per run between MG-PSO and GPSO-*w* algorithms for PGS-1.

Table 5
Optimized output power in (MW) for each TGU obtained by MG-PSO and GPSO-*w* algorithms for PGS-1.

TGU$_j$	P_j (MW)	
	GPSO-*w*	MG-PSO
1	449.7956	628.3002
2	149.5601	149.5959
3	224.7231	222.5866
4	109.8681	109.7003
5	110.1797	60.0017
6	160.0134	109.9501
7	109.8367	109.9506
8	60.0840	109.9501
9	160.1193	109.9506
10	77.4152	40.0028
11	40.2176	40.0013
12	92.4584	55.0071
13	55.7288	55.0025
Total output power (MW)	1800	1800

The maximum load demand at normal operation steady state is given as $P_D = 1800$ MW. Here, the VPL effects and power generation limits are taken into account. However, the RRLs, POZs and P_L are not considered. The specification of PGS-1 is shown in Table 2 [25].

5.2.1. Comparison in terms of fitness values

Recently, the PGS-1 has been tested by several ECTs. In Ref. [14], PGS-1 has been tested by 2 ECTs, i.e., IRDPSO and ST-IRDPSO. It has also been tested by MSOS [22], FCASO-SQP [23], CBA [24], continuous GRASP with self-adaptive differential evolution (C-GRASP-SaDE) [25] and RTO [28]. Here, we compare the performance of the proposed MG-PSO algorithm with other 8 existing ECTs. The set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Table 3. In addition, both are run with $m = 20$, $d = 13$ and $N_{run} = 25$. In MG-PSO algorithm, two negative gradients are selected ($N_{grad} = 2$), by using trial and error method, one for *Exploration* phase and another for *Exploitation* phase as shown in Table 3.

The fitness values of MG-PSO algorithm and other 8 ECTs are listed in Table 4. It can be seen that MG-PSO algorithm provides the best result in terms of F_{mean} over 25 independent runs. However, it is the second best in terms of σ. This indicates that the MG-PSO algorithm provides most optimum and consistent results. In addition, the range *R* of MG-PSO algorithm is close to the best one MSOS [22], thus indicating that MG-PSO algorithm provides solution with low dispersion. In terms of AET, the MG-PSO algorithm is the fifth best. These results indicate that among the 9 ECTs, the MG-PSO algorithm is stable, robust and is able to provide optimum solution.

5.2.2. Convergence characteristics of MG-PSO and GPSO-w algorithms

Fig. 5 shows the convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-1. It shows ensemble average F_{cost} values at each iteration obtained from 25 independent runs. It can be seen that MG-PSO algorithm settles at about 150 iterations and achieves F_{mean} of about $17,956/h. Whereas, the GPSO-*w* algorithm takes more than 500 iterations to converge, and settles at a local minimum with a non-optimum F_{mean} of about $18,326/h. This indicates that MG-PSO algorithm gives higher accuracy in solving this PGS, compared to GPSO-*w* algorithm.

Fig. 6 shows the variation of optimized F_{cost} over 25 independent runs achieved by the MG-PSO and GPSO-*w* algorithms. It shows that the optimized F_{cost} of MG-PSO algorithm varies between $17,955.88/h and $17,956.27/h whereas, in GPSO-*w* algorithm, it varies between $18,047.11/h and $18,531.13/h. This indicates that MG-PSO algorithm is capable of providing consistent and reliable solution. Whereas, the GPSO-*w* algorithm is rather far from optimum solution due to VPL effects.

Table 6
Specifications and power constraints for PGS-2.

TGU	P_j^0 (MW)	$P_{j,}min$ (MW)	$P_{j,}max$ (MW)	a_i ($/h)	b_i ($/MW)	c_i ($/MW$^{2)}$	UR_j (MW/h)	DR_j (MW/h)	POZs (MW)
1	400	150	455	671	10.10	0.000299	80	120	NC
2	300	150	455	574	10.20	0.000183	80	120	[185,225][305,335][420,450]
3	105	20	130	374	8.80	0.001126	130	130	NC
4	100	20	130	374	8.80	0.001126	130	130	NC
5	90	150	470	461	10.40	0.000205	80	120	[180,200][305,335][390,420]
6	400	135	460	630	10.10	0.000301	80	120	[230,255][365,395][430,455]
7	350	135	465	548	9.80	0.000364	80	120	NC
8	95	60	300	227	11.20	0.000338	65	100	NC
9	105	25	162	173	11.20	0.000807	60	100	NC
10	110	25	160	175	10.70	0.001203	60	100	NC
11	60	20	80	186	10.20	0.003586	80	80	NC
12	40	20	80	230	9.90	0.005513	80	80	[30,40][55,65]
13	30	25	85	225	13.10	0.000371	80	80	NC
14	30	15	55	309	12.10	0.001929	55	55	NC
15	20	15	55	323	12.40	0.004447	55	55	NC

NC: No constraints.

Table 7
B-loss coefficients of 15 TGUs of PGS-2.

B_{jk}	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
1	14	12	7	−1	−3	−1	−1	−1	−3	5	−3	−2	4	3	−1
2	12	15	13	0	−5	−2	0	1	−2	−4	−4	−0	4	10	−2
3	7	13	76	−1	−13	−9	−1	0	−8	−12	−17	−0	−26	111	−28
4	−1	0	−1	34	−7	−4	11	50	29	32	−11	−0	1	1	−26
5	−3	5	−13	−7	90	14	−3	−12	−10	−13	7	−2	−2	−24	−3
6	−1	−2	−9	−4	14	16	−0	−6	−5	−8	11	−1	−2	−17	3
7	−1	0	−1	11	−3	−0	15	17	15	9	−5	7	−0	−2	−8
8	−1	1	0	50	−12	−6	17	168	82	79	−23	−36	1	5	−78
9	−3	−2	−8	29	−10	−5	15	82	129	116	−21	−25	7	−12	−72
10	−5	−4	−12	32	−13	−8	9	79	116	200	−27	−34	9	−11	−88
11	−3	−4	−17	−11	7	11	−5	−23	−21	−27	140	1	4	−38	168
12	−2	−0	−0	−0	−2	−1	7	−36	−25	−34	1	54	−1	−4	28
13	4	4	−26	1	−2	−2	−0	1	7	9	4	−1	103	−101	28
14	3	10	111	1	−24	−17	−2	5	−12	−11	−38	−4	−101	578	−94
15	−1	−2	−28	−26	−3	3	−8	−78	−72	−88	168	28	28	−94	1283

$B_{jk} = 1 \times 0^{-06}\mathrm{MW}^{-1}$.

Table 8
Set of parameters used in MG-PSO and GPSO-*w* algorithms for PGS-2.

Set of parameters	MG-PSO		GPSO-w
	Exploration phase	*Exploitation phase*	
	$k = 1$	$k = 2$	
γ	0.40	0.40	–
c_1, c_2	2.05	2.05	2.00
N_{iter}	400	600	1000
$w_{ini,k}$	1.00	0.50	0.90
$w_{fin,k}$	0.10	0.10	0.40
$grad_k$	-2.25×10^{-03}	6.67×10^{-04}	-5.00×10^{-04}

5.2.3. Comparison in terms of inequality constraints

Table 5 presents the solution vector, P_j ($j = 1, 2, \ldots, 13$) corresponding to the best solution of MG-PSO and GPSO-*w* algorithms. Note that the $P_D = 1800$ MW. It can be seen that both the MG-PSO and GPSO-*w* algorithms are working within generation limits (8) and satisfying power balance constraint.

5.3. Test case 2: power generation system-2 (PGS-2)

The PGS-2 is a medium-scale PGS [18] with 15 TGUs ($N_{gen} = 15$). The TGU specification and *B*-loss coefficients are shown in Table 6 and Table 7, respectively. Maximum load demand at normal operation is given as $P_D = 2630$ MW. As seen from Table 6, the PGS-2 has

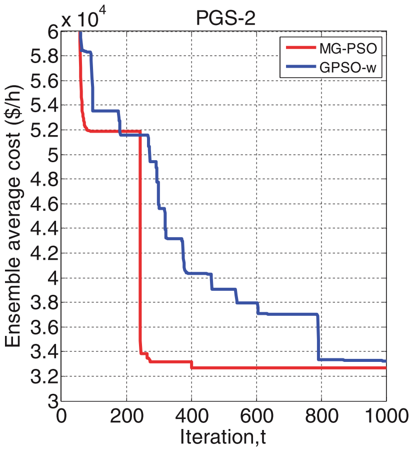


Fig. 7. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-2.

Table 9
Comparison of cost performance between MG-PSO algorithm and other 20 ECTs for PGS-2.

Sl.No.	Algorithm	Min. Cost ($/h)	Max. Cost ($/h)	Mean Cost ($/h)	σ ($/h)	*R* ($/h)	AET (sec)
1	OPSO [10]	32,668.4863	32,669.3005	**32,668.9205**	0.1394	0.8142	4.3777
2	PSO-MSAF [11]	32,713.0900	32,798.2500	32,759.6400	NA	85.1600	19.1500
3	SA-PSO [15]	32,708.0000	32,789.0000	32,732.0000	18.0250	81.0000	12.7900
4	CPSO [17]	32,834.0000	33,318.0000	33,021.0000	NA	484.0000	13.1300
5	GA [18]	32,939.5208	33,231.6216	33,106.0019	100.1279	292.1008	NA
6	DE [18]	32,818.5792	33,116.9340	32,990.8673	61.5145	298.3548	NA
7	ACSA [18]	32,785.6031	33,185.2761	33,051.7711	77.8005	399.6730	NA
8	AIS [18]	32,895.9173	33,132.0191	33,017.6537	58.1230	236.1018	NA
9	FA [18]	32,901.6610	33,197.2718	33,081.0107	91.0111	295.6108	NA
10	BCO [18]	32,989.2341	33,301.4940	33,113.0149	69.7986	312.2599	NA
11	APSO [18]	32,687.9840	33,359.6609	32,948.0533	92.0040	671.6769	NA
12	HGPSO [18]	32,864.0501	33,280.2655	33,034.1894	63.9932	416.2154	NA
13	HPSOM [18]	32,697.2458	33,015.7284	32,819.5931	83.0907	318.4826	NA
14	HPSOWM [18]	32,696.9585	33,034.3413	32,805.7185	87.8689	337.3828	NA
15	RDPSO [18]	32,666.1818	32,934.3089	32,739.7165	56.7070	268.1271	NA
16	λ-logic [20]	32,713.9510	NA	NA	NA	NA	NA
17	SPPO [21]	32,713.2100	NA	NA	NA	NA	NA
18	θ-MBA [27]	32,680.5956	32,693.2640	32,687.3305	NA	12.6684	**0.0983**
19	EGSSOA [30]	NA	NA	32,680.1038	NA	NA	NA
20	GPSO-*w*	33,118.2274	33,429.7840	33,246.5605	72.0523	311.5566	7.2378
21	MG-PSO	32,677.9098	32,678.0217	32,677.9666	**0.0348**	**0.1119**	9.2084

The bold numbers indicate the best solution found by corresponding algorithm.

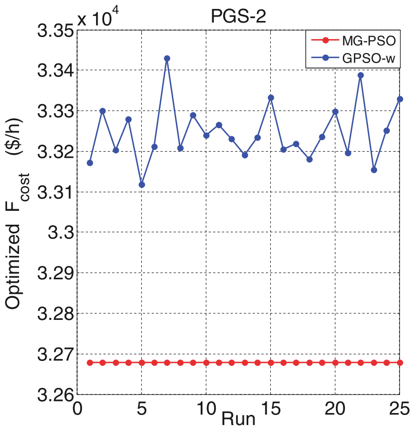


Fig. 8. Comparison of optimized cost per run between MG-PSO and GPSO-*w* algorithms for PGS-2.

Table 10
Optimized output power in (MW) for each TGU obtained by MG-PSO and GPSO-*w* algorithms for PGS-2.

TGU_j	P_j(MW)	
	GPSO-*w*	MG-PSO
1	455.0000	455.0000
2	303.3837	380.0000
3	128.0633	129.9965
4	129.6879	130.0000
5	169.8932	170.0000
6	363.6612	459.8764
7	429.8892	430.0000
8	148.3850	60.0045
9	161.6133	50.0179
10	135.3756	160.0000
11	79.7284	80.0000
12	20.0354	80.0000
13	32.9961	25.0000
14	54.6227	31.9996
15	54.6609	15.0000
Total output power (MW)	2666.9959	2656.8949

11 POZs in 4 TGUs and RRLs are applied in each TGU.

5.3.1. Comparison in terms of fitness values

In Ref. [18], the RDPSO algorithm was tested on PGS-2 and its performance was compared to other 10 ECTs. In addition, the OPSO [10], PSO-MSAF [11], SA-PSO [15], CPSO [17], λ-logic [20], SPPO [21], θ-MBA [27] and EGSSOA [30] algorithms have also been tested with PGS-2. Here, the performance of MG-PSO with GPSO-*w* algorithms and other existing 19 ECTs are compared. The set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Table 8. In addition, both are run with $m = 20$, $d = 15$ and $N_{run} = 25$. In MG-PSO algorithm, two negative gradients are selected ($N_{grad} = 2$), by using trial and error method, one for *Exploration* phase and another one for *Exploitation* phase as shown in Table 8.

The Comparison of fitness values between MG-PSO algorithm and other 20 existing ECTs are listed in Table 9. It can be seen that, the MG-PSO algorithm achieves the best positions in terms of σ and *R* and the second best position result in terms of F_{mean}. The best F_{mean} is achieved by OPSO algorithm [10]. However, in terms of AET, the MG-PSO algorithm is the fourth best. The NA denotes that the value was not available in corresponding paper. The bold values indicate the best solution found by corresponding algorithm. These results indicate that the MG-PSO algorithm provides consistent, stable and robust performance.

5.3.2. Convergence characteristics of MG-PSO and GPSO-w algorithms

Fig. 7 shows convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-2. It shows ensemble average F_{cost} values at each iteration obtained from 25 independent runs. It can be seen that MG-PSO algorithm settles at about 450 iterations to achieve $F_{mean} = \$32,678/h$ whereas GPSO-*w* algorithm takes about 800 iterations to settle and achieved $F_{mean} = \$33,000/h$.

Fig. 8 shows the distribution of optimized F_{cost} at each run. It shows that the optimized F_{cost} of MG-PSO algorithm remains steady at about $32,677/h, whereas in GPSO-*w* algorithm, the optimized F_{cost} varies over a wide range from $33,118/h to $33,429/h. This indicates that MG-PSO algorithm is more consistent, stable and reliable than the GPSO-*w* algorithm.

5.3.3. Comparison in terms of inequality constraints

Table 10 presents the solution vector, P_j ($j = 1, 2, \ldots, 15$) corresponding to the best solution for MG-PSO and GPSO-*w* algorithms. It can be seen that both the MG-PSO and GPSO-*w* algorithms are able to avoid the 11 POZs imposed on 4 TGUs and are remain within RRLs constraints imposed on each TGU. Thus, both algorithms are able to satisfy the power constraints (15). Note that the $P_D = 2630$ MW for PGS-2.

5.3.4. Comparison in terms of power balance constraint

Table 11 shows results of power balance constraint for the MG-PSO algorithm and other 9 ECTs. The load demand of PGS-2 is given by $P_D = 2630$ MW. Using the optimum output power generated as

Table 11
Comparison of power balance constraint among 10 ECTs for PGS-2.

Sl. No.	Algorithm	Total P_j (MW)	P_D (MW)	P_{L1} (MW)	P_{L2} (MW)	$P_{L,\text{mismatch}}$ (MW)
1	OPSO [10]	2657.2591	2630	27.2591	27.2591	**0.0000**
2	PSO-MSAF [11]	2660.4900	2630	30.4900	30.4900	**0.0000**
3	SA-PSO [15]	2660.9000	2630	30.9000	30.9080	0.0080
4	CPSO [17]	2662.1000	2630	32.1000	32.1303	0.0303
5	RDPSO [18]	2655.3650	2630	25.3650	25.3696	0.0460
6	λ-logic [20]	2659.9491	2630	29.9491	29.9491	**0.0000**
7	SPPO [21]	2660.0000	2630	30.0000	31.4300	1.4300
8	EGSSOA [30]	2657.0120	2630	27.0120	27.0120	**0.0000**
9	GPSO-*w*	2666.9959	2630	36.9959	36.9956	0.0003
10	**MG-PSO**	2656.8949	2630	26.8949	26.8949	**0.0000**

The bold numbers indicate the best solution found by corresponding algorithm.

Table 12
Specifications and power constraints for PGS-3.

TGU$_j$	P_j^0 (MW)	$P_{j,min}$ (MW)	$P_{j,max}$ (MW)	a_i ($/h)	b_i ($/MWh)	c_i ($/MW^{2}h)	UR_j (MW/h)	DR_j (MW/h)	POZs (MW)
1	50	40	80	170.77	8.3360	0.03073	35	60	NC
2	60	60	120	309.54	7.0706	0.02028	40	70	NC
3	150	80	190	369.03	8.1817	0.00942	50	90	[82,88]
4	24	24	42	135.48	6.9467	0.08482	42	42	NC
5	42	26	42	135.19	6.5595	0.09693	42	42	NC
6	75	68	140	222.33	8.0543	0.01142	40	75	NC
7	100	110	300	287.71	8.0323	0.00357	65	100	[155,162][221,235]
8	152	135	300	391.98	6.9990	0.00492	65	100	NC
9	200	135	300	455.76	6.6020	0.00573	65	100	[235,246]
10	100	130	300	722.82	12.9080	0.00605	65	100	[200,211]
11	300	94	375	635.20	12.9860	0.00515	55	95	[213,220]
12	300	94	375	654.69	12.7960	0.00569	55	95	[213,220]
13	150	125	500	913.40	12.5010	0.00421	80	120	[201,211][290,310][413,425]
14	200	125	500	1760.4	8.8412	0.00752	80	120	[205,217][306,318][409,420]
15	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290][402,412]
16	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290][402,412]
17	190	125	500	1728.3	9.1575	0.00708	80	120	[214,230][277,290][402,412]
18	400	220	500	647.85	7.9691	0.00313	70	110	[307,321][407,421]
19	400	220	500	649.69	7.9550	0.00313	70	110	[301,310][421,431]
20	398	242	500	647.83	7.9691	0.00313	70	110	[340,351][421,431]
21	398	242	500	647.81	7.9691	0.00313	70	110	[340,351][421,431]
22	390	254	550	785.96	6.6313	0.00298	70	110	[306,320][440,445]
23	390	254	550	785.96	6.6313	0.00298	70	110	[306,320][440,445]
24	390	254	550	794.53	6.6311	0.00284	70	110	[370,390][495,502]
25	390	254	550	794.53	6.6311	0.00284	70	110	[370,390][495,502]
26	390	254	550	801.32	7.1032	0.00277	70	110	[380,410][501,520]
27	390	254	550	801.32	7.1032	0.00277	70	110	[380,410][501,520]
28	20	10	150	1055.10	3.3353	0.52124	90	150	[102,113]
29	20	10	150	1055.10	3.3353	0.52124	90	150	[102,113]
30	30	10	150	1055.10	3.3353	0.52124	90	150	[102,113]
31	30	20	70	1207.80	13.0520	0.25098	70	70	NC
32	40	20	70	810.79	21.887	0.16766	70	70	NC
33	40	20	70	1247.70	10.2440	0.26350	70	70	NC
34	25	20	70	1219.20	8.3707	0.30575	70	70	NC
35	25	18	60	641.43	26.2580	0.18362	60	60	NC
36	20	18	60	1112.80	9.6956	0.32563	60	60	NC
37	20	20	60	1044.40	7.1633	0.33722	60	60	NC
38	25	25	60	832.24	16.3390	0.23915	60	60	NC
39	25	25	60	832.24	16.3390	0.23915	60	60	NC
40	25	25	60	1035.2	16.3390	0.23915	60	60	NC

NC: No constraints.

shown in Table 10 and Equations (5)–(7), P_{L1}, P_{L2} and $P_{L,mismatch}$, were determined. It can be seen that the proposed MG-PSO algorithm as well as OPSO [10], PSO-MSAF [11], λ-logic [20] and EGSSOA [30] algorithms are able to satisfy the zero mismatch condition, i.e., $P_{L,mismatch} = 0$, thus satisfying (4).

5.4. Test case 3: power generation system-3 (PGS-3)

The PGS-3 is a large-scale PGS taken from Taipower system [39]. The PGS-3 consists of 40 mixed-generating units, coal-fired, gas-fired, gas-turbines with complex cycle, diesel generating units and nuclear generating units. The maximum load demand at normal and steady-state operations and is given as $P_D = 8550$ MW. The PGS-3 contains 46 POZs distributed among 25-TGU and are shown in Table 12. The RRLs are imposed on all the 40 TGUs. The *B*-loss coefficients matrix of dimension 40 × 40 is taken from Ref. [10]. Unfortunately, the PGS-3 is tested by only a few authors with RRLs, POZs and P_L constraint. This may be due to unavailability of *B*-loss coefficients or due to its high dimension with a large number of power constraints.

Table 13
Set of parameters used in MG-PSO and GPSO-*w* algorithms for PGS-3.

Set of parameters	MG-PSO			GPSO-*w*
	Exploration phase		*Exploitation* phase	
	$k = 1$	$k = 2$	$k = 3$	
γ	0.30	0.30	0.30	–
c_1, c_2	2.05	2.05	2.05	2.00
N_{iter}	300	300	700	1000
$w_{ini,k}$	1.00	0.80	0.40	0.90
$w_{fin,k}$	0.20	0.20	0.15	0.40
$grad_k$	-2.67×10^{-03}	-2.00×10^{-03}	-3.57×10^{-04}	-5.00×10^{-04}

Table 14
Comparison of cost performance between MG-PSO and other 15 ECTs for PGS-3.

Sl. No.	Algorithm	Min.Cost F_{min} ($/h)	Max. Cost F_{max} ($/h)	Mean Cost F_{mean} ($/h)	σ ($/h)	R ($/h)	AET (sec)
Without POZs, RRLs and P_L							
1	GA [18]	133,435.6906	136,274.9726	135,012.4985	729.3536	2839.2820	NA
2	DE [18]	129,915.5635	137,042.9461	130,600.2269	1335.4343	7127.3826	NA
3	ACSA [18]	131,167.3417	134,923.6245	132,844.7110	741.0843	3756.2828	NA
4	AIS [18]	130,133.9214	132,703.1884	131,482.2767	561.7950	2569.2670	NA
5	FA [18]	130,948.8466	134,997.9243	133,511.4572	747.3692	4049.0777	NA
6	BCO [18]	130,337.7290	132,999.8803	131,733.9439	589.8034	2662.1513	NA
7	APSO [18]	130,861.5242	134,044.6303	132,587.8486	675.0344	3183.1061	NA
8	HGPSO [18]	132,072.2495	135,528.3862	134,012.5706	684.4951	3456.1367	NA
9	HPSOM [18]	129,177.4413	131,281.3077	130,234.1694	529.5827	2103.8664	NA
10	HPSOWM [18]	129,717.3557	132,303.5999	130,858.6741	591.7691	2586.2442	NA
11	RDPSO [18]	128,864.4525	131,129.0861	129,482.0970	568.9333	2264.6336	NA
With POZs, RRLs and P_L							
12	OPSO [10]	126,489.6228	127,916.1972	127,349.8324	302.3502	1426.5744	69.32
13	MIQCQP [19]	128,395.2900	NA	NA	NA	NA	**13.34**
14	λ-logic [20]	129,777.5300	NA	NA	NA	NA	NA
15	GPSO-*w*	136,185.0955	564,890.7051	396,596.5735	155,100.0223	424,705.6096	18.60
16	**MG-PSO**	126,561.5538	126,683.8917	**126,625.0260**	**20.2709**	**145.2457**	29.38

The bold numbers indicate the best solution found by corresponding algorithm.

5.4.1. Comparison in terms of fitness values

In Ref. [18], PGS-3 has been tested with 11 ECTs and superior performance of RDPSO algorithm over other 10 ECTs has been shown. However, the 46 POZs of 25-TGU, RRLs of each TGU and the P_L constraint have not been considered. Therefore, these results are less constrained. Considering all the POZs, RRLs and P_L, the PGS-3 has been tested by OPSO [10], MIQCQP [19], λ-logic [20], the proposed MG-PSO and GPSO-*w* algorithms. Thus, here, we are compare the performance of MG-PSO algorithm and other 15 existing ECTs. The set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Table 13. In addition, both are run with $m = 20$, $d = 40$ and $N_{run} = 25$. In MG-PSO algorithm, three negative gradients were selected ($N_{grad} = 3$) by trial and error method, two for *Exploration* phase and another one for *Exploitation* phase as shown in Table 13.

The fitness values of the 16 ECTs are listed in Table 14. It can be seen that the MG-PSO algorithm provides the best result in terms of F_{mean} and σ over 25 independent runs. This indicates that the MG-PSO algorithm provides the most optimum and consistent results. In addition, the range R of MG-PSO algorithm is the lowest among the 16 ECTs, thus indicating that MG-PSO algorithm provides solution with lowest dispersion. In terms of AET, the MG-PSO algorithm is the third best. The GPSO-*w* algorithm is not able to provide an accurate solution. These results indicate that among the 16 ECTs, the MG-PSO algorithm is the most stable, robust and is able to provide most optimum solution.

5.4.2. Convergence characteristics of MG-PSO and GPSO-*w* algorithms

Fig. 9 shows convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-3. It shows ensemble average F_{cost} values at each iteration obtained from 25 independent runs. It can be seen that MG-PSO algorithm settles at about 300 iterations and achieves F_{mean} of about $126,850/h. Whereas, the GPSO-*w* algorithm takes about 320 iterations to converge, and settles at a local minimum with a non-optimal F_{mean} of about $396,798/h, which is not acceptable solution. This indicates that the GPSO-*w* algorithm is unable to solve ED problem with such a high dimensional search space and under large number of power constraints. In contrast, the MG-PSO algorithm gives high accuracy in solving this complex

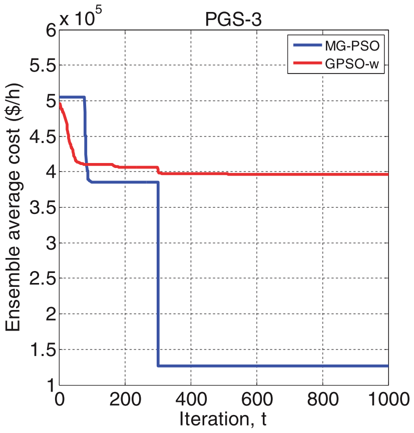


Fig. 9. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-3.

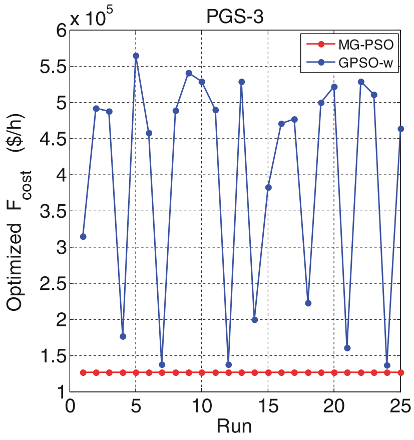
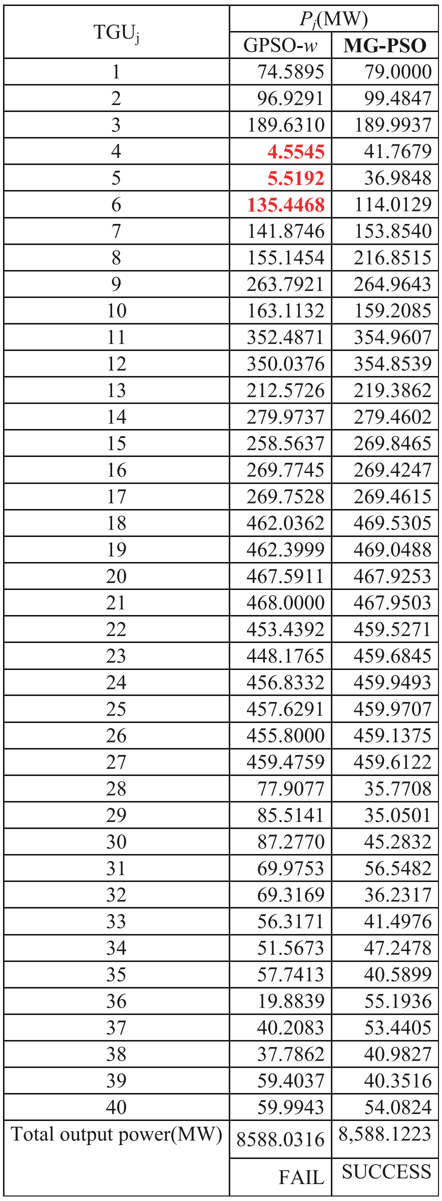


Fig. 10. Comparison of optimized cost per run between MG-PSO and GPSO-*w* algorithms for PGS-3.

Table 15
Optimized output power in (MW) for each TGU obtained by MG-PSO and GPSO-*w* algorithms for PGS-3.



TGU_j	P_j(MW)	
	GPSO-*w*	**MG-PSO**
1	74.5895	79.0000
2	96.9291	99.4847
3	189.6310	189.9937
4	**4.5545**	41.7679
5	**5.5192**	36.9848
6	**135.4468**	114.0129
7	141.8746	153.8540
8	155.1454	216.8515
9	263.7921	264.9643
10	163.1132	159.2085
11	352.4871	354.9607
12	350.0376	354.8539
13	212.5726	219.3862
14	279.9737	279.4602
15	258.5637	269.8465
16	269.7745	269.4247
17	269.7528	269.4615
18	462.0362	469.5305
19	462.3999	469.0488
20	467.5911	467.9253
21	468.0000	467.9503
22	453.4392	459.5271
23	448.1765	459.6845
24	456.8332	459.9493
25	457.6291	459.9707
26	455.8000	459.1375
27	459.4759	459.6122
28	77.9077	35.7708
29	85.5141	35.0501
30	87.2770	45.2832
31	69.9753	56.5482
32	69.3169	36.2317
33	56.3171	41.4976
34	51.5673	47.2478
35	57.7413	40.5899
36	19.8839	55.1936
37	40.2083	53.4405
38	37.7862	40.9827
39	59.4037	40.3516
40	59.9943	54.0824
Total output power(MW)	8588.0316	8,588.1223
	FAIL	SUCCESS

problem.

Fig. 10 shows the variation of optimized F_{cost} over 25 independent runs achieved by the MG-PSO and GPSO-*w* algorithms. It shows that the optimized F_{cost} of MG-PSO algorithm varies between \$126,561.5538/h to \$126,683.8917/h, whereas in GPSO-*w* algorithm, it varies between \$136,185.0955/h to \$564,890.7051/h. This indicates that MG-PSO algorithm is capable of providing consistent and reliable optimal solution. Whereas, the GPSO-*w* algorithm is unable to provide optimal solution due to high complexity of the problem.

5.4.3. Comparison in terms of inequality constraints

Table 15 presents the solution vector, P_j ($j = 1, 2, \ldots, 40$) corresponding to the best solution obtained from MG-PSO and GPSO-*w* algorithms. Note that the load demand, $P_D = 8850$ MW. In case of GPSO-*w* algorithm, based on (15), the TGU_4, TGU_5 and TGU_6 violate RRLs (red color). The TGU_4, TGU_5 and TGU_6 must operate within $P_{4,low} = 24$ MW to $P_{4,high} = 42$ MW, $P_{5,low} = 26$ MW to $P_{5,high} = 42$ MW, $P_{6,low} = 68$ MW to $P_{6,high} = 115$ MW, respectively. This means that GPSO-*w* algorithm fails in solving PGS-3 indicating that it is unable to solve large-scale ED problem. Whereas, the MG-PSO algorithm avoids all the 46 POZs of 25 TGUs and remains within RRLs.

5.4.4. Comparison in terms of power balance constraint

Since all the data for other existing ECTs are not available for PGS-3, we compare the performance between MG-PSO algorithm and OPSO [10], λ-logic [20]. The GPSO-*w* is out of comparison, because it failed in solving PGS-3. The load demand of PGS-3 is given as $P_D = 8550$ MW. Using the total optimum output power generated (Table 15) and Equations (5)–(7), P_{L1}, P_{L2} and $P_{L,mismatch}$, were determined and are presented in Table 16. It can be seen that $P_{L,mismatch}$ of MG-PSO algorithm is more close to 0.0 than OPSO [10] and λ-logic [20], which indicates better performance of MG-PSO algorithm.

5.5. Test case 4: power generation system-4 (PGS-4)

The PGS-4 is a very large-scale PGS taken from Korean PGS [25]. It is a complex with 140 TGUs each having RRLs. In addition, the cost functions of 12 TGUs have VPL effects and 4 TGUs have 11 POZs. The maximum load demand under steady-state and normal operations is 49,342 MW. The P_L of this PGS is neglected. The PGS-4 data are available in Ref. [25].

5.5.1. Comparison in terms of fitness values

The PGS-4 has already been tested with 2 existing ECTs, i.e., CCPSO [12] and C-GRASP-SaDE [25]. Here, the performance of MG-PSO algorithm is compared with these two algorithms and GPSO-*w*. The set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Table 17. In addition, both are run with $m = 20$, $d = 140$ and $N_{run} = 25$. In MG-PSO algorithm, four negative gradients were selected ($N_{grad} = 4$) by trial and error method, three for *Exploration* phase and another one for *Exploitation* phase as shown in Table 17.

The fitness values of the 4 ECTs are listed in Table 18. It can be seen that in GPSO-*w*, $F_{mean} = \$2,529,855.79$/h and $\sigma = \$358,126.35$/

Table 16
Comparison of power balance constraint among 3 ECTs for PGS-3.

Algorithm	Total P_j (MW)	P_D (MW)	P_{L1} (MW)	P_{L2} (MW)	$P_{L,mismatch}$ (MW)
OPSO [10]	8588.0734	8550	38.0734	38.1121	0.0387
λ-logic [20]	8637.3300	8550	87.3300	87.4037	0.0737
MG-PSO	8588.1223	8550	38.1223	38.1227	**0.0004**

The bold number indicates the best solution found by corresponding algorithm.

Table 17
Set of parameters used in MG-PSO and GPSO-*w* algorithms for PGS-4.

Set of parameters	MG-PSO				GPSO-*w*
	Exploration phase			*Exploitation* phase	
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	
γ	0.40	0.40	0.40	0.40	–
c_1, c_2	2.05	2.05	2.05	2.05	2.00
N_{iter}	1200	1200	1200	1800	3000
$w_{ini,k}$	0.80	0.80	0.80	0.35	0.90
$w_{fin,k}$	0.10	0.20	0.30	0.20	0.40
$grad_k$	-5.83×10^{-4}	-5.00×10^{-4}	-4.16×10^{-4}	-8.33×10^{-5}	-1.67×10^{-04}

Table 18
Comparison of cost performance between MG-PSO algorithm and other 3 ECTs for PGS-4.

Sl.No.	Algorithm	Min. Cost F_{min} (\$/h)	Max. Cost F_{max} (\$/h)	Mean Cost F_{mean} (\$/h)	σ (\$/h)	*R* (\$/h)	AET (sec)
1	CCPSO [12]	1,657,962.7300	1,657,962.7300	1,657,962.7300	**0.00**	**0.00**	150.00
2	C-GRASP-SaDE [25]	1,657,962.7268	1,658,583.5267	1,658,006.2712	NA	620.79	NA
3	GPSO-*w*	1,933,419.8873	3,366,473.6288	2,529,855.7978	358,126.35	1,433,053.74	**31.29**
4	MG-PSO	1,656,515.4715	1,656,917.3113	**1,656,667.4650**	8.01	401.83	48.37

The bold numbers indicate the best solution found by corresponding algorithm.

h. These results indicate that GPSO-*w* is unable to solve PGS-4. Whereas, the MG-PSO algorithm is efficient in obtaining the best result in terms of F_{mean} over 25 independent runs. In addition, in terms of σ, the performance of the MG-PSO algorithm is the second best. This shows that the MG-PSO algorithm provides optimum and consistent results. In addition, the range *R* of MG-PSO algorithm is the second lowest among the 4 ECTs, thus indicating that it provides solution with low dispersion. In terms of AET, the MG-PSO algorithm shows the second best performance. These results indicate that among the 4 ECTs, the MG-PSO algorithm is stable and robust and is able to provide optimum solution.

5.5.2. Convergence characteristics of MG-PSO and GPSO-*w* algorithms

Fig. 11 shows the convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-4. It shows ensemble average F_{cost} values at each iteration obtained from 25 independent runs. It can be seen that MG-PSO algorithm settles at about 1300 iterations and achieves F_{mean} of about \$1,656,700/h. Whereas, the GPSO-*w* algorithm settles at a non-optimal F_{mean} of about \$2,529,855.79/h since the beginning of its learning. Early convergence of the GPSO-*w* algorithm indicates that it has trapped into a local minimum of at about \$2,529,855.79/h. This indicates that the GPSO-*w* algorithm is unable to solve ED problem with such a high dimension ($d = 140$) and under such a large number of power constraints. Whereas, it is clear that for this complex PGS, the MG-PSO algorithm efficiently converges to the vicinity of the optimum solution with different power constraints imposed by SPG.

Fig. 12 shows the variation of optimized F_{cost} over 25 independent runs achieved by the MG-PSO and GPSO-*w* algorithms. It shows that the optimized F_{cost} of MG-PSO algorithm varies between \$1,656,515/h and \$1,656,917/h, whereas in GPSO-*w* algorithm, it varies between \$1,933,419.88/h and \$3,366,473.62/h. This indicates that MG-PSO algorithm is capable of providing consistent and reliable optimal solution. Whereas, the GPSO-*w* algorithm is unable

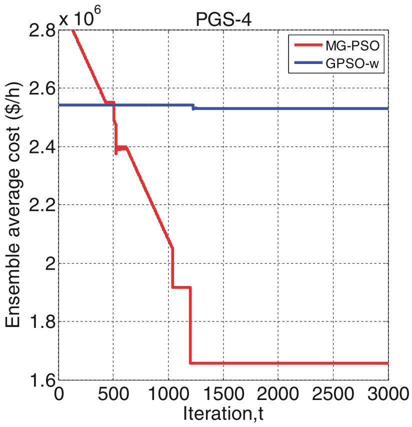


Fig. 11. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-4.

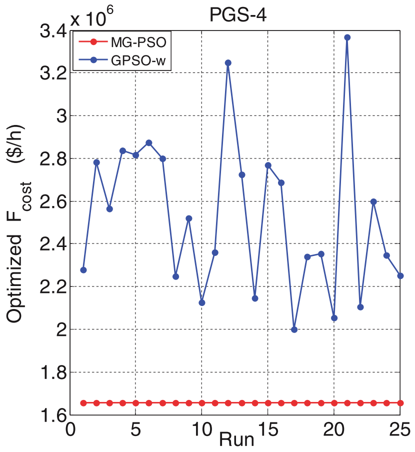


Fig. 12. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for PGS-4.

Table 19
Optimized output power in (MW) for each TGU obtained by GPSO-*w* and MG-PSO algorithms for PGS-4.

GPSO-*w*								MG-PSO							
TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j
1	119.0	39	611.1	77	513.1	115	368.2	1	118.9029	39	773.9550	77	389.9391	115	244.0759
2	150.8	40	707.6	78	336.4	116	286.4	2	163.9317	40	768.9389	78	330.8755	116	244.3029
3	167.7	41	4.7	79	196.7	117	348.2	3	189.8356	41	3.5417	79	530.7148	117	244.0177
4	137.8	42	12.1	80	272.4	118	160.9	4	189.8119	42	3.1421	80	530.9548	118	95.1031
5	183.4	43	170.4	81	525.4	119	119.8	5	189.8391	43	249.7563	81	541.6917	119	95.2248
6	129.3	44	185.4	82	90.8	120	189.3	6	189.9790	44	249.7986	82	56.1218	120	116.1326
7	320.7	45	222.9	83	156.1	121	300.7	7	489.9250	45	249.9315	83	115.4416	121	175.2108
8	285.0	46	197.2	84	236.1	122	11.0	8	489.9932	46	249.7574	84	115.3084	122	2.0596
9	392.6	47	178.2	85	241.9	123	17.6	9	495.8361	47	249.9664	85	115.4370	123	4.1350
10	266.4	48	191.0	86	239.3	124	63.7	10	495.9449	48	249.8982	86	207.4802	124	15.1293
11	441.0	49	224.3	87	211.7	125	16.2	11	495.8284	49	249.8841	87	207.0506	125	9.3245
12	329.6	50	234.8	88	295.0	126	20.6	12	495.8931	50	249.8927	88	175.2168	126	12.3679
13	351.5	51	365.2	89	193.6	127	13.3	13	505.9485	51	165.0090	89	175.2709	127	10.0243
14	467.3	52	484.4	90	239.3	128	230.0	14	508.9545	52	165.0718	90	175.3078	128	112.1335
15	424.4	53	383.5	91	324.6	129	19.0	15	505.9317	53	165.2701	91	175.3082	129	4.0748
16	377.6	54	457.9	**92**	**516.7**	130	34.9	16	504.8797	54	165.3485	92	575.3236	130	5.3571
17	309.7	55	269.9	**93**	**509.8**	131	6.2	17	505.9219	55	180.0918	93	547.2814	131	5.3220
18	360.9	56	380.7	**94**	**983.9**	132	86.9	18	505.9252	56	180.2213	94	836.2569	132	50.0785
19	369.5	57	278.5	95	813.1	133	6.8	19	504.9787	57	103.0974	95	837.2834	133	5.1698
20	348.6	58	521.0	96	634.2	134	71.9	20	504.8703	58	198.6536	96	681.8989	134	42.1825
21	263.6	59	230.9	97	712.4	135	65.6	21	504.9936	59	311.8011	97	719.9877	135	42.3435
22	352.5	60	216.3	98	683.5	**136**	**51.2**	22	504.9871	60	311.6648	98	717.6493	136	41.2530
23	470.7	61	425.2	99	705.5	137	23.1	23	504.8732	61	163.7961	99	719.6865	137	17.2795
24	272.5	62	298.6	100	891.5	138	17.8	24	504.8014	62	95.8232	100	963.8141	138	7.0127
25	505.1	63	478.2	101	945.7	139	7.6	25	536.8458	63	510.9918	101	957.9583	139	7.0810
26	507.6	64	470.4	102	862.4	140	34.6	26	536.9566	64	510.9026	102	946.5203	140	26.1599
27	481.	65	205.	**103**	**976.1**	P_D = 49,342MW		27	548.948	65	489.632	103	933.772	P_D = 49,342MW	

	4		1					8		8		7	
28	420.2	66	458.6	**104**	**984.6**	Total output power=49,231.1MW. FAIL	28	548.9923	66	252.6819	104	934.8718	Total output power=49,342MW. SUCCESS
29	324.9	67	413.8	**105**	**1,014.3**		29	500.9955	67	489.7562	105	876.4437	
30	463.6	68	293.6	**106**	**906.0**		30	498.9895	68	489.6741	106	880.1277	
31	430.2	69	175.3	**107**	**938.4**		31	505.9392	69	130.8168	107	873.4553	
32	285.0	70	297.0	108	874.2		32	505.8717	70	296.6273	108	877.2801	
33	473.8	71	337.9	**109**	**879.0**		33	505.9013	71	142.7573	109	871.2103	
34	455.5	72	411.5	**110**	**919.5**		34	505.8016	72	367.6518	110	864.6802	
35	461.1	73	345.3	**111**	**969.3**		35	499.9410	73	195.6819	111	882.3793	
36	293.9	74	473.3	112	165.3		36	499.8281	74	219.6960	112	94.3471	
37	164.8	75	483.2	113	193.4		37	240.9999	75	217.8117	113	94.2248	
38	216.7	76	523.9	114	113.6		38	240.9666	76	267.8831	114	94.2292	

to provide optimal solution due to the high complexity of the problem.

5.5.3. Comparison in terms of inequality constraints

Table 19 presents solution vector, P_j ($j = 1, 2, \ldots, 140$) corresponding to the best solution obtained from MG-PSO and GPSO-*w* algorithms. Note that $P_D = 49{,}342$ MW. In case of GPSO-*w* algorithm, 11 TGUs violate RRLs, as shown in red color. Table 20 shows the details of the 11 TGUs for which GPSO-*w* algorithm failed to satisfy RRLs. These 11 TGUs must operate within the range of RRLs based on (15). In addition, TGU #136 violates POZ [50–74] MW based on (15), as shown in blue color in Table 19. This means that GPSO-*w* algorithm fails to solve PGS-4. This indicates that GPSO-*w* algorithm is unable to solve ED problem of very large-scale TGUs under different power constraints. Whereas, the MG-PSO algorithm avoids the 11 POZs imposed on 4 TGUs and working within RRLs of each TGU and solving non-smooth cost function due to VPL effects imposed on 12 TGUs.

5.6. Statistical significance of the proposed MG-PSO algorithm

In order to determine the statistical significance of the proposed MG-PSO algorithm, three sets of unpaired one-tailed *t*-Test are carried out [40] with a significance level of $\alpha = 0.05$. The MG-PSO algorithm is considered to be statistically significant against the contender algorithm when t-value < 0 and p-value less than 0.05. The general merit over contender is shown in the last row of Table 21.

Table 21 shows the *t*-Test results for the four PGSs, PGS-1, PGS-2, PGS-3 and PGS-4. The one-tailed unpaired ($\alpha = 0.05$ with a degree of freedom of 24) is performed against 29 competitive algorithms. As seen from the data in the last column in Table 21, the proposed MG-PSO algorithm is found to be statistically significant against the GPSO-*w* algorithm for four PGSs.

Since data are not available for the 28 algorithms corresponding to the four PGSs with Sl.No., 2 to 29, the *t*-Test was carried out against one or two PGSs. Again, from the data in the last column, one can see that the MG-PSO algorithm is statistically significant against the 28 contending algorithms and neutral with the OPSO algorithm [10]. These results give enough evidence that the proposed MG-PSO algorithm is statistically significant against all the 29 contending algorithms.

6. Conclusion

A novel algorithm called multi-gradient PSO (MG-PSO) algorithm is proposed and applied to optimize total fuel cost of four medium and large PGSs under several practical constraints. In MG-PSO algorithm, several negative gradients are used by *m* particles while searching for a global optimum in two phases called

Table 20
List of 11 TGUs that violate RRLs based on output active power P_j obtained by GPSO-*w* for PGS-4.

TGU_j	92	93	94	103	104	105	106	107	109	110	111
$P_{j,low}$ (MW)	539.4	511.5	795.0	844.0	875.0	816.5	820.9	813.7	799.5	795.0	810.0
$P_{j,high}$ (MW)	575.4	547.5	836.8	934.0	935.0	876.5	880.9	873.7	871.7	864.8	882.0
P_j (MW)	516.7	509.8	983.9	976.1	984.6	1014.3	906.0	938.4	879.0	919.5	969.3

Table 21
Statistical results of unpaired *t*-Test of MG-PSO algorithm against 29 ECTs for PGS-1, PGS-2, PGS-3 and PGS-4.

Sl. No.	Competitive Algorithms	PGS-1		PGS-2		PGS-3		PGS-4		t = negative t < 0	t = positive t ≥ 0	General Merit Over Contender
		t-value	p-value	t-value	p-value	t-value	p-value	t-value	p-value			
1	GPSO-*w*	-170×10^{03}	1.60×10^{-86}	-6.19×10^{04}	5.74×10^{-100}	7.02×10^{03}	2.83×10^{-77}	-4.02×10^{04}	1.84×10^{-95}	4	0	4
2	IRDPSO [14]	-7.37×10^{02}	8.72×10^{-54}	-	-	-	-	–	–	1	0	1
3	ST-IRDPSO [14]	-4.61×10^{02}	6.87×10^{-49}	-	-	-	-	–	–	1	0	1
4	MSOS [22]	-3.22×10^{02}	3.56×10^{-45}	-	-	-	-	–	–	1	0	1
5	FCASO-SQP [23]	-2.07×10^{02}	1.46×10^{-64}	-	-	-	-	–	–	1	0	1
6	CBA [24]	-4.15×10^{02}	8.60×10^{-48}	-	-	-	-	–	–	1	0	1
7	RTO [28]	-1.70×10^{04}	1.60×10^{-86}	-	-	-	-	–	–	1	0	1
8	SA-PSO [15]	-	-	-7.76×10^{03}	2.58×10^{-78}	–	–	–	–	1	0	1
9	CPSO [17]	-	-	-4.92×10^{04}	1.40×10^{-99}	–	–	–	–	1	0	1
10	SPPO [26]	-	-	-5.06×10^{03}	7.35×10^{-74}	–	–	–	–	1	0	1
11	EGSSOA [30]	-	-	-3.06×10^{02}	1.20×10^{-44}	–	–	–	–	1	0	1
12	θ-MBA [27]	-	-	-1.34×10^{03}	4.80×10^{-60}	–	–	–	–	1	0	1
13	PSO-MSAF [11]	-	-	-1.17×10^{04}	1.27×10^{-82}	–	–	–	–	1	0	1
14	GA [18]	-	-	-6.14×10^{04}	6.93×10^{-100}	2.06×10^{03}	1.55×10^{-64}	–	–	2	0	2
15	DE [18]	-	-	-4.48×10^{04}	1.36×10^{-96}	-9.80×10^{02}	9.43×10^{-57}	–	–	2	0	2
16	ACSA [18]	-	-	-5.36×10^{04}	1.79×10^{-98}	-1.53×10^{03}	2.03×10^{-61}	–	–	2	0	2
17	AIS [18]	-	-	-2.50×10^{04}	1.78×10^{-97}	-1.28×10^{03}	1.38×10^{-59}	–	–	2	0	2
18	FA [18]	-	-	-5.78×10^{04}	2.93×10^{-99}	-1.69×10^{03}	1.76×10^{-62}	–	–	2	0	2
19	BCO [18]	-	-	-3.19×10^{04}	4.97×10^{-100}	-1.26×10^{03}	2.28×10^{-59}	–	–	2	0	2
20	APSO [18]	-	-	-3.87×10^{04}	4.38×10^{-94}	-1.47×10^{03}	5.60×10^{-61}	–	–	2	0	2
21	HGPSO [18]	-	-	-5.11×10^{04}	5.69×10^{-98}	-1.83×10^{03}	3.27×10^{-63}	–	–	2	0	2
22	HPSOM [18]	-	-	-2.03×10^{04}	2.33×10^{-88}	-8.90×10^{02}	9.58×10^{-56}	–	–	2	0	2
23	HPSOWM [18]	-	-	-1.83×10^{04}	2.78×10^{-87}	-1.04×10^{03}	2.08×10^{-57}	–	–	2	0	2
24	RDPSO [18]	-	-	-8.86×10^{03}	1.05×10^{-79}	-7.04×10^{02}	2.61×10^{-53}	–	–	2	0	2
25	λ-logic [20]	-	-	-5.16×10^{03}	4.46×10^{-74}	-7.77×10^{02}	2.46×10^{-54}	–	–	2	0	2
26	OPSO [10]	-	-	1.29×10^{03}	1.09×10^{01}	-1.78×10^{02}	5.13×10^{-39}	–	–	1	1	0

27	MIQCQP [24]	–	–	-	-	**−4.36** × **10^{02}**	**2.54** × **10$^{−48}$**	–	–	1	0	1
28	CCPSO [12]	-	-	-	-	–	–	−5.96 × **10^{01}**	1.34 × **10$^{−27}$**	1	0	1
29	C-GRASP-SaDE [25]	**−4.61** × **10^{02}**	**6.87** × **10$^{−48}$**	-	-	-	-	−6.16 × **10^{01}**	6.04 × **10$^{−28}$**	2	0	2

Numbers in bold signatures that MG-PSO algorithm is statistically significant with respect to the corresponding algorithms.

Exploration and *Exploitation*. The combination of two phases provides a balance between global and local search spaces. With extensive simulation studies, performance of the MG-PSO algorithm was compared with GPSO-*w* algorithm and several existing competitive algorithms and its superiority is demonstrated in terms of mean cost, convergence rate and consistency. In case of PGS-1, the proposed MG-PSO algorithm provides the best mean cost when compared to eight competitive ECTs. In PGS-2, it provides the second best mean cost results when compared with twenty other ECTs. The best mean cost is achieved by our earlier proposed OPSO algorithm. In case of two large PGSs with more constraints, the proposed algorithm achieves the best mean cost when compared with several ECTs. In addition, statistical tests were carried out to demonstrate the effectiveness of the proposed algorithm. Thus, the MG-PSO algorithm has proved to be a powerful and highly effective algorithm that is capable of solving multi-modal, discontinuous and non-convex functions.

Appendix

```
  Begin MG-PSO Algorithm
    Choose $N_{iter}$, $N_{grad}$, $w_{ini,k}$, $w_{fin,k}$, $k = 1, 2, \ldots, N_{grad}$
    Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (25) and (26), respectively.
  Step 1: Initialization: Iteration, $t = 0$
    Obtain $G_{best}(0)$ using (16)–(19)
  Step 2: Begin Exploration phase
    for $k = 1, 2, \ldots, N_{grad} - 1$
    begin of episode $k$
    Determine $grad_k$ using (27)
      for $t = 1, 2, \ldots, N_{iter,xplore}$
      Determine $w_k(t)$ using (30)
        for $i = 1, 2, \ldots, m$
        Update the particle's velocity and position vectors as follows
```

$$V_i^k(t) = w_k(t)\ V_i^k(t-1) + c_1\ r_1(t)\left[G_{pers,i}^k(t-1) - X_i^k(t-1)\right] + c_2\ r_2(t)\left[G_{best}^k(t-1) - X_i^k(t-1)\right] \tag{A-1}$$

$$X_i^k(t) = X_i^k(t-1) + V_i^k(t) \tag{A-2}$$

```
        Evaluate the particle's performance by substituting (A-2) in $f(x)$
        Update $G_{pers,i}$ as follows
```

$$G_{pers,i}^k(t) = \begin{cases} X_i^k(t) & if\quad f\left(X_i^k(t)\right) \le f\left(G_{pers,i}^k(t-1)\right) \\ G_{pers,i}^k(t-1) & Otherwise \end{cases} \tag{A-3}$$

```
        end i loop
      Obtain $f(G_{best}^k(t))$
      $f(G_{best}^k(t)) = min\{f(G_{pers,i}^k(t))\}$, $i = 1, 2, \ldots, m$
      Obtain $G_{best}^k(t)$ corresponding to $f(G_{best}^k(t))$
      end t loop
    Obtain $G_{best}^k(N_{iter,xplore})$ and $f(G_{best}^k(N_{iter,xplore}))$
    end of episode $k$
    end k loop
    Obtain minimum f($G_{best,xplore}(N_{iter,xplore})$) by
    $f(Gbest, xplore) =$
    $\min\{f(G_{best}^{k}(N_{iter,xplore}))\},\ k = 1, 2, \ldots N_{grad} - 1$
    Obtain BEST($G_{best,xplore}$) corresponding to f($G_{best,xplore}$)
    Obtain new search space (neighborhood) by taking "Floor" and "Ceil" of each element of $BEST(G_{best,xplore})$
End Exploration phase
Begin Exploitation phase
Step 3: Initialization: Iteration, $t = 1$
    for $i = 1, 2, \ldots, m$
    $V_i(1) = V_i(N_{iter,xplore})$ corresponding to BEST($G_{best,xplore}$)
    $X_i(1) = X_i(N_{iter,xplore})$ corresponding to BEST($G_{best,xplore}$)
    $G_{pers,i}(1) = G_{pers,i}(N_{iter,xplore})$ corresponding to BEST($G_{best,xplore}$)
    $G_{best,xploit}(1) =$ BEST($G_{best,xplore}$)
    end $i$ loop
    Determine $grad_{N_{grad}}$ using (28)
Step 4: Update
    for $t = 2, 3, \ldots, N_{iter,xploit}$
    Determine $w_k(t)$ using (30)
        for $i = 1, 2, \ldots, m$
        Update the particle's velocity and position vectors as follows:
```

$$V_i(t) = w_{N_{grad}}(t)\ V_i(t-1) + c_1 r_1(t)\left[G_{pers,i}(t-1) - X_i(t-1)\right] + c_2 r_2(t)\left[G_{best,xploit}(t-1) - X_i(t-1)\right] \tag{A-4}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{A-5}$$

```
        Evaluate the particle's performance by substituting (A-5) in $f(x)$
        Update $G_{pers,i}(t)$ as follows
```

$$G_{pers,i}(t) = \begin{cases} X_i(t) & \text{if } f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & \text{Otherwise} \end{cases} \tag{A-6}$$

```
        end $i$ loop
    Obtain f($G_{best,xploit}(t)$)
    $f(G_{best,xolit}(t)) = min\{\ f(G_{pers,i}(t))\ \}$, $i = 1, 2, \ldots, m$
    Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
    end $t$ loop
    Optimum solution = $G_{best,xploit}(N_{iter,xploit})$
    Optimum value = f($G_{best,xploit}(N_{iter,xploit})$)
End of Exploitation phase
```

References

[1] Mahdi F, Vasant P, Kallimani V, Watada J, Fai P, Abdullah-Al-Wadud M. A holistic review on optimization strategies for combined economic emission dispatch problem. Renew Sustain Energy Rev 2017;81:3006–20.

[2] Lynn N, Suganthan PN. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evolut Comput 2015;24:11–24.

[3] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol. 4; 1995. p. 1942–8.

[4] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science; 1995. p. 39–43.

[5] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation; 1999. p. 1945–50.

[6] Zhan Z, Zhang J, Li Y, Shi Y. Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput 2011;6:832–47.

[7] Gao W, Liu S, Huang L. A novel artificial bee colony algorithm based modified search equation and orthogonal learning. IEEE Trans Cybern 2013;43(3):1011–24.

[8] Leung Y, Wang Y. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans Evol Comput 2011;5(1):41–53.

[9] Shi YH, Eberhart RC. A modified particle swarm optimizer. In: Proceedings of the IEEE congress on evolutionary computation; 1998. p. 69–73.

[10] Al-Bahrani LT, Patra JC. Orthogonal particle swarm optimization for economic dispatch of thermal generating units under various power grid constraints in smart power grid. Appl Soft Comput 2017;58:401–26.

[11] Subbaraj P, Rengaraj R, Salivahanan S, Senthilkumar T. Parallel particle swarm optimization with modified stochastic acceleration factors for solving large scale economic dispatch problem. Int J Electr Power Energy Syst 2010;32:1014–23.

[12] Park J, Jeong Y, Shin J. An Improved particle swarm optimization for non-convex economic dispatch problems. IEEE Trans Power Syst 2010;25(1):156–66.

[13] Ling SH, Iu C, Chan K, Lam K, Yeung B, Leung F. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. IEEE Trans Syst Man Cybern 2008;38(3):743–63.

[14] Elsayed W, Hegazy Y, El-bages M, Bendary F. Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem. IEEE Trans Ind Inf 2017;13(3):1017–26.

[15] Kuo C. A novel coding scheme for practical economic dispatch by modified particle swarm approach. IEEE Trans Power Syst 2008;23(4):1825–35.

[16] Selvakumar A, Thanushkodi K. Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems. Int J Electric Power Syst Res 2008;78:2–10.

[17] Cai J, Ma X, Li L, Peng H. Chaotic particle swarm optimization for economic dispatch considering the generator constraints. Energy Convers Manag 2007;48:645–53.

[18] Sun J, Palade V, Wu X, Fang W, Wang Z. Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization. IEEE Trans Power Syst 2014;13(1):519–26.

[19] Ding T, Bo R, Li F, Sun H. A bi-level branch and bound method for economic dispatch with disjoint prohibited zones considering network losses. IEEE Trans Power Syst 2015;30(6):2841–55.

[20] Adhinarayanan T, Sydulu M. Efficient Lambda logic based optimization procedure to solve the large scale generator constrained economic dispatch problem. Int J Electr Eng Technol 2009;4(3):301–9.

[21] Singh N, Dhillon J, Kothari D. Synergic predator-prey optimization for economic thermal power dispatch problem. Appl Soft Comput 2016;43:298–311.

[22] Secui D. A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point effects. Energy 2016;113:366–84.

[23] Cai J, Li Q, Li L, Peng G, Yang Y. A hybrid FCASO-SQP method for solving the economic dispatch problems with valve-point effects. Energy 2012;38:346–53.

[24] Adarsh BR, Raghunathan T, Jayabarathi T, Yang X-S. Economic dispatch using chatic bat algorithm. Energy 2016;96:666–75.

[25] Neto J, Reynoso-Meza G, Ruppel T, Mariani V, Coelho L. Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution. Int J Electr Power Energy Syst 2017;84:13–24.

[26] Meng A, Hu Hanwu H, Yin H, Zhuanghi Guo Peng. Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects. Energy 2015;93:2175–90.

[27] Kavousi-Fard A, Khosravi A. An intelligent θ-modified bat algorithm to solve non-convex economic dispatch problem considering practical constraints. Int J Electr Power Energy Syst 2016;82:189–96.

[28] Labbi Y, Attous D, Gabbar H, Mahdad B, Zidan A. A new rooted tree optimization algorithm for economic dispatch with valve-point effect. Int J Electr Power Energy Syst 2016;79:298–311.

[29] Noel MM, Jannett TC. Simulation of a new hybrid particle swarm optimization algorithm. In: Proceedings of the 36th southeastern symposium on system theory; 2004. p. 150–3.

[30] Azizipanah-Abarghooee R, Niknam T, Gharibzadeh M, Golestaneh F. Robust, fast and optimal solution of practical economic dispatch by a new enhanced gradient-based simplified swarm optimisation algorithm. IET Gener, Transm Distrib 2013;7(6):620–35.

[31] Azizipanah-Abarghooee R, Dehghaniab P, Terzija V. Practical multi-area bi-objective environmental economic dispatch equipped with a hybrid gradient search method and improved Jaya algorithm. IET Gener, Transm Distrib 2016;10(14):3580–96.

[32] Al-Bahrani LT, Patra JC, Kowalczyk R. Multi-Gradient PSO algorithm for economic dispatch of thermal generating units in smart grid. In: Proceedings of IEEE innovative smart grid technologies-Asia (ISGT-Asia) Melbourne; 2016. p. 258–63. Australia.

[33] Gherbi Y, Bouzeboudja H, Gherbi F. The combined economic dispatch environmental dispatch using new hybrid metaheuristic. Energy 2016;115:468–77.

[34] Kheshti M, Kang X, Bie Z, Zaibin Jiao, Wang Xiuli. An effective lightning flash algorithm solution to large scale non-convex economic dispatch with valve-point and multiple fuel options on generation units. Energy 2017;129:1–15.

[35] Niu Q, Zhang H, Li K, Irwin G. An efficient harmony search with new pitch

adjustment for dynamic economic dispatch. Energy 2014;65:25–43.

[36] Niknam T, Mojarrad H, Meymand H, Firouzi B. A new honey bee mating optimization algorithm for non-smooth economic dispatch. Energy 2011;36: 896–908.

[37] Mohammadi B, Rabiee B, Soroudi A, Ehsan M. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. Energy 2012;44:228–34.

[38] Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Scrip Intell 2007: 33–57.

[39] Chen P, Chang H. Large-scale economic dispatch by genetic algorithm. IEEE Trans Power Syst 1995;10(4):1919–26.

[40] Sheskin DJ. Handbook of parametric and nonparametric statistical procedures. USA: Western Connecticut State University; 2000.

Paper H: Applied Soft Computing 2018

Multi-gradient PSO algorithm with enhanced exploration and exploitation

This paper is under review in Applied Soft Computing Journal (2018)

Multi-gradient PSO algorithm with enhanced exploration and exploitation

Loau Tawfak Al-Bahrani [a, b], Jagdish Chandra Patra [a]

[a] Swinburne University of Technology, Melbourne, Australia, [b] University of Diyala, Iraq

E-mail: lalbahrani@swin.edu.au; JPatra@swin.edu.au

Abstract

Over the last two decades a large number of optimization techniques have been proposed for solving complex unimodal and multimodal problems. One popular population-based optimization technique is global particle swarm optimization with inertia weight (GPSO-*w*) algorithm. In this paper, a novel multi-gradient PSO (MG-PSO) algorithm is proposed to solve such complex problems. In MG-PSO algorithm, two phases called *Exploration* phase and *Exploitation* phase are used. In the *Exploration* phase, the *m* particles are called *Explorers* and undergo multiple episodes. In each episode, the *Explorers* use a different negative gradient to explore new neighbourhood whereas in the *Exploitation* phase, the *m* particles are called *Exploiters* and they use one negative gradient that is less than that of the *Exploration* phase, to exploit a best neighborhood. This diversity in negative gradients provides a balance between global search and local search of the *Explorers* and *Exploiters*. The effectiveness of the MG-PSO algorithm is verified using ten selected shifted and rotated benchmark functions with dimensions of 30 and 100 taken from congress on evolutionary computation (CEC) 2015. In addition, the MG-PSO algorithm is evaluated using a real-world problem (case study), i.e., economic dispatch of South Korea power generating system. Superior performance of the MG-PSO algorithm has been shown over the GPSO-*w* algorithm and several existing optimization techniques in terms of several performance measures, e.g., fitness value, convergence rate, and consistency. In addition, by using unpaired t-test, the statistical significance of the MG-PSO algorithm has been shown against several contending algorithms including top-ranked CEC 2015 algorithms.

Keywords: Multi-gradient particle swarm optimization; exploration and exploitation phases; unimodal and multimodal benchmark functions, economic dispatch problem.

1 Introduction

Many optimization problems are generally classified into two groups, unimodal and multimodal problems. Some of the proposed evolutionary computation techniques (ECTs), i.e., population-based techniques to solve such problems are particle swarm optimization (PSO) [1-4] algorithm, genetic algorithm [5-7], and differential evolution (DE) [8-10]. In ECTs, finding the optimum solution of an objective function is based on two phases, namely *Exploration* phase and *Exploitation* phase. In the *Exploration* phase, global search particles, i.e., individuals, exploring all over the search space as much as possible is carried out to find promising new neighbourhoods. Whereas, in the *Exploitation* phase, local search particles exploiting the best neighbourhood to fine-tune the search space is carried out to obtain the optimum solution. The best performance of an ECT is achieved when an appropriate balance between these two phases is maintained [11-12]. Focusing more on *Exploration* will lead to excessive search time because of wastage of time in searching over inferior neighbourhoods, whereas focusing more on *Exploitation* will cause loss of diversity, thereby possibly getting stuck into a local minimum.

One of such popular ECTs named global PSO with inertia weight (GPSO-*w*) algorithm has been proposed [13], for solving unimodal and multimodal functions. The negative gradient of inertia weight, *w*, in GPSO algorithm is used to boost the global search and local search abilities and to make a balance between the *Exploration* and *Exploitation* phases [14]. Thus, the *w* helps the particles to control the convergence tendency and to quickly convergence to optimum solution.

In unimodal optimization problems, there is only one global minimum. Movement of the particles in the direction of negative of the gradient leads to the global minimum. Since an objective function decreases by the largest amount possible in the direction of the negative gradient, movement of the particles in this direction will cause a greater decrease of the function. Therefore, the GPSO-*w* algorithm has performed well on smooth and convex unimodal problems. However, under high-dimensional complex unimodal problems, e.g., shifted and rotated benchmark functions taken from the congress on evolutionary computation (CEC) 2015 [15-16], the GPSO-*w* algorithm may suffer from the curse of dimensionality and overcoming the premature convergence remains a challenge, in spite of existence of negative gradient of *w*.

In multimodal problems, which have multiple local minima, all of its particles in GPSO-*w* algorithm share the swarm's best experience, i.e., global best and this may lead the particles to cluster around the global best. If the global best is located near a local minimum, then escaping from it becomes hard, because of loss of balance between the local search guide (personal experience of each particle) and the global search guide (global best) [17-20]. Thus, the GPSO-*w* algorithm suffers diversity loss near a local minimum.

Recently, some of the notable ECTs that have been used to improve the GPSO-*w* algorithm through enhancing their *Exploration* and *Exploitation* processes are as follows. In [21], a shrinking hypersphere PSO with gravitational search algorithm (SHPSO-GSA) using a gravitational search to enhance the performance of PSO algorithm was proposed. A directionally driven self-regulating PSO (DD-SRPSO) [22] applies cooperation between two strategies: directional update and rotational invariant. In [23], extraordinariness PSO (EPSO) algorithm considering an extraordinary motion of the particles was proposed. Based on this motion, the particles can move toward a global optimum which can be global best, local bests, or even the worst individual. In [24-25], orthogonal PSO (OPSO) algorithm divides the particles into an active group and another passive group, and uses an orthogonal diagonalization process. Here, the orthogonality is used in active group particles to enhance global and local search processes to achieve optimum solution.

Some other categories of ECTs that have been applied to solve complex unimodal and multimodal problems are self-optimization based adaptive DE with linear population (L-SHADE) and eigenvector-based crossover and successful-parent-selecting (SPS-L-SHADE-EIG) [26], DE with success-based parameter adaption (DEsPA) algorithm [27], mean-variance mapping optimization (MVMO) algorithm [28-29], tuned covariance matrix evolution strategy (TunedCMAES) [30], local Lipschits underestimate DE (LLUDE) [31], strategy adaptation DE (SaDE) [31], JADE adaptive DE [31], composite DE (CoDE) [30], and self-adaptive binary variant DE (SabDE) [32], chaotic sequence and crossover PSO (CCPSO) algorithm [33], continuous greedy randomized adaptive search procedure (C-GRASP) with self-adaptive DE (C-GRASP-SaDE) algorithm [34], and C-GRASP with modified DE (C-GRASP-MDE) algorithm [34].

The gradient method is one of the classical methods. It is used to solve linear and unimodal problems. In general, the classical methods including gradient method have major disadvantages like they are inefficient to solve unimodal problems with high-dimensional search space, they suffer from the “curse of dimensionality”, they are inefficient to solve multimodal problems, e.g., non-convex, non-smooth, and discontinuous problems, they are sensitive to an initial starting point, they requires a monotonically increasing objective function, and they are often trap into local minima. However, in the recent years, the gradient method is successfully integrated and combined with few optimization techniques to create hybrid optimization techniques. This combination is used to achieve faster convergence without getting trapped into local minima. The gradient method helps particles to move faster toward optimum solution, whereas the optimization algorithm controls the movement of the particles from falling into a local minimum. Some of the recently proposed such techniques are hybrid gradient algorithm [35], enhanced gradient simplified swarm optimization algorithm [36], and gradient-based Jaya algorithm [37].

In this paper, a novel algorithm called multi-gradient PSO (MG-PSO) algorithm is proposed in which multiple negative gradients are used by *m* particles while searching for optimum solution. The multiple negative gradients help to prevent the global best particle to fall in a local minimum. In MG-PSO algorithm two phases are used, i.e., *Exploration* phase and *Exploitation* phase. In *Exploration* phase, a particle is called an *Explorer*. The *Explorers* operate in several episodes. In each episode, the *Explorers* use a different negative gradient to explore a new neighbourhood. *Explorers* enhance global search ability of the MG-PSO algorithm. At the end of *Exploration* phase, the *Explorers* provide a search boundary which becomes the new search space in the *Exploitation* phase. In the *Exploitation* phase, a particle is called an *Exploiter*. *Exploiters* use one negative gradient which is less than that of the *Exploration* phase to exploit the best neighborhood. The small negative gradient leads to small incremental change in the velocity and position vectors during updating process. This helps the particles to move steadily towards optimum solution. Thus, *Exploiters* enhance local search ability of MG-PSO algorithm. This diversity in negative gradients helps the best particle from falling into a local minimum. The combination of two phases provides a balance between *Exploration* and *Exploitation* in search space. In addition, the combination of two phases is successfully applied to overcome the disadvantages of the gradient methods.

In a recent work, the effectiveness of the proposed MG-PSO algorithm has been shown in solving real-world non-convex problem, i.e., economic dispatch (ED) of small, medium and large power generating systems (PGSs) under several power constraints in smart power grid applications [36-37]. Whereas, in the current study, the MG-PSO algorithm is applied to solve different complex problems with 30 and 100 dimensions, i.e., CEC 2015 shifted and rotated unimodal and multimodal benchmark functions as well as solving ED problem of the South Korea PGS (case study). Furthermore, the mathematical analysis and theoretical justification of MG-PSO algorithm is provided. With extensive simulated experiments, superior performance of the MG-PSO algorithm has been shown in terms of several performance measures, e.g., fitness values, convergence rate and consistency, compared to GPSO-*w* algorithm and several competitive ECTs. In addition, with unpaired t-test, the statistical significance of MG-PSO algorithm has been shown, over several competing algorithms including top-ranked CEC 2015 algorithms.

The rest of the paper is organized as follows. Explanation of the GPSO-*w* algorithm is presented in Section 2. Details of the proposed MG-PSO algorithm are provided in Section 3. In Section 4, application of the MG-PSO algorithm to CEC 2015 benchmark functions is presented. In Section 5, solving ED problem of the South Korea PGS is presented. Finally, conclusion of this study is provided in Section 6.

2 The GPSO-*w* algorithm

The optimization mechanism of the GPSO-*w* algorithm depends on the distribution of the particles in a swarm [13]. It is indicated by a fully connected network, in which each particle has access to the information of the swarm population, as follows. Firstly, each particle flying in the multi-dimensional search space adjusts its flying trajectory according to two guides, its personal experience, $G_{pers,i}$ and its neighborhood's best experience, G_{best}. Secondly, when seeking an optimum solution (global solution), each particle learns from its own historical experience and its neighborhood's historical experience. In such a case, a particle while choosing the neighborhood's best experience uses the best experience of the whole swarm as its neighbor's best experience. Therefore, the GPSO algorithm is named as global PSO, because the position of each particle is affected by the best-fit particle in the entire swarm. The following steps explain the mechanism of the GPSO-*w* algorithm.

Consider a swarm population with m particles ($m > 1$) searching for optimum solution (minimum) of an objective function $f(x)$ in a d-dimensional search space. Let the total number of iterations is N_{iter}. The objective is to minimize the given $f(x)$. A particle, i ($i = 1, 2, ..., m$), has one d-dimensional velocity vector V_i and one d-dimensional position vector X_i and these are denoted by

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}] \tag{1}$$

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \tag{2}$$

```
Step 1: Initialization: Iteration, t = 0.
    for $i = 1, 2, \ldots, m$
    Initialize the $V_i$ and $X_i$ randomly within a defined range of $d$-dimensional search space and denote these by $V_i(0)$ and $X_i(0)$, respectively.
    Initialize the personal position vector of particle $i$, $G_{pers,i}(0)$, as follows:
    $G_{pers,i}(0) = X_i(0)$                                                        (3)
    Evaluate the $f(x)$ using $X_i(0)$.
    end $i$ loop
    Determine the global best position vector, $G_{best}(0)$. It is the best position vector among all the $m$ personal position vectors. The $G_{best}(0)$ is denoted by
    $G_{best}(0) = [g_{b,1}, g_{b,2}, \ldots, g_{b,d}]$                              (4)
Step 2: Update:
    for $t = 1, 2, \ldots, N_{iter}$
        for $i = 1, 2, \ldots, m$
        Determine inertia weight, $w(t)$ [40], as given below:
        $w(t) = -\frac{0.5}{N_{iter}} t + 0.9$                                        (5)
        Update $V_i$ and $X_i$ as follows:
        $V_i(t) = w(t)\, V_i(t-1) + c_1 r_1(t)\, [G_{pers,i}(t-1) - X_i(t-1)] + c_2 r_2(t)\, [G_{best}(t-1) - X_i(t-1)]$   (6)
        $X_i(t) = X_i(t-1) + V_i(t)$                                                  (7)
        where $c_1$ and $c_2$ are positive coefficients, called acceleration constants which are commonly set to 2.0 [40]. The $r_1(t)$ and $r_2(t)$ are two randomly generated values with a uniform distribution in the range of [0,1].
        Evaluate $f(x)$ for particle $i$ using $X_i(t)$.
        Update $G_{pers,i}(t)$, as follows:
```

```
        $G_{pers,i}(t) = \begin{cases} X_i(t) & if \quad f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & Otherwise \end{cases}$        (8)
        end $i$ loop
    Obtain $f(G_{best}(t))$ as follows:
    for $i = 1, 2, \ldots, m$ {obtain $f(G_{pers,i}(t))$}
    $f(G_{best}(t)) = min\{f(G_{pers,i}(t))$
    Obtain $G_{best}(t)$ corresponding to $f(G_{best}(t))$
    end $t$ loop
Step 3: End of iteration: $t = N_{iter}$
    Optimum solution = $G_{best}(N_{iter})$ and optimum value = $f(G_{best}(N_{iter}))$        (9)
```

3 Learning strategy and MG-PSO algorithm

Here, the details of the proposed MG-PSO algorithm and explanation of its learning strategy are provided.

3.1 *Learning strategy*

The learning strategy of MG-PSO algorithm depends on the following considerations. Consider a swarm population with m particles, whereas $m > 1$, flying in a d-dimensional space searching for an optimum solution, i.e., global optimum. Two fundamental phases, "*Exploration* and *Exploitation*" are used by the m particles. In *Exploration* phase, a particle is called *Explorer*. In each episode, the *Explorers* use a different negative gradient to explore new neighbourhood in a d-dimensional search space. The *Explorers* enhance a global search ability of MG-PSO algorithm using several episodes. The purpose of *Explorers* is to obtain a new neighbourhood within the d-dimensional search space in each episode, and to obtain the best neighborhood among episodes.

In each episode using a different negative gradient, the *Explorers* obtain best position vector following its neighbourhood in the d-dimensional search space. Its neighborhood is obtained by taking "Floor" and "Ceil" of each element of the best position vector. These operations create a new search space (best neighborhood) within the d-dimensional search space that will be used in the *Exploitation* phase. In *Exploitation* phase, a particle is called an *Exploiter*. The *Exploiters* use one negative gradient which is less than that of the *Exploration* phase. The *Exploiters* enhance the local search ability of MG-PSO algorithm. The purpose of this phase is to obtain an optimum position by exploiting the *Exploiters* in the best neighborhood obtained from the *Exploration* phase.

3.2 *The MG-PSO algorithm*

In MG-PSO algorithm, number of negative gradients (N_{grad}) are used while the swarm population searches for an optimum solution. In *Exploration* phase, $N_{grad} - 1$ negative gradients are used and one negative gradient is used in *Exploitation* phase. In each episode, the inertia weight follows one negative gradient.

Let N_{iter} be number of iterations in MG-PSO algorithm. The number of iterations in *Exploration* phase is given by

$$N_{iter,xplore} = \gamma \times N_{iter} \tag{10}$$

where γ is a real and positive number in a range [0,1]. The number of iterations in *Exploration* phase is given by:

$$N_{iter,xploit} = (1 - \gamma) \times N_{iter} \tag{11}$$

The initial and final values of the inertia weight for *k*th negative gradient ($k = 1, 2, \ldots, N_{grad}$) are denoted by $w_{ini,k}$ and $w_{fin,k}$, respectively. These values are real and positive numbers within a range [0,1] and $w_{ini,k} > w_{fin,k}$. The *k*th negative gradient ($k = 1, 2, \ldots, N_{grad} - 1$) in *Exploration* phase is given by:

$$grad_k = \frac{w_{fin,k} - w_{ini,k}}{N_{iter,xplre}} \tag{12}$$

In *Exploitation* phase, the negative gradient is given by:

$$grad_{N_{grad}} = \frac{w_{fin,N_{grad}} - w_{ini,N_{grad}}}{N_{iter,xploit}} \tag{13}$$

The N_{grad} gradients are selected such that (14) is satisfied.

$$\left| grad_1 \right| > \left| grad_2 \right| \cdots > \left| grad_{N_{grad}} \right| \tag{14}$$

The inertia weight for *k*th negative gradient ($k = 1, 2, \ldots, N_{grad}$) at iteration *t* is given by:

$$w_k(t) = grad_k \times t + w_{ini,k} \tag{15}$$

The flowchart of the MG-PSO algorithm is shown in Fig. 1. The detailed steps explaining pseudocode of the MG-PSO algorithm are given in Appendix.

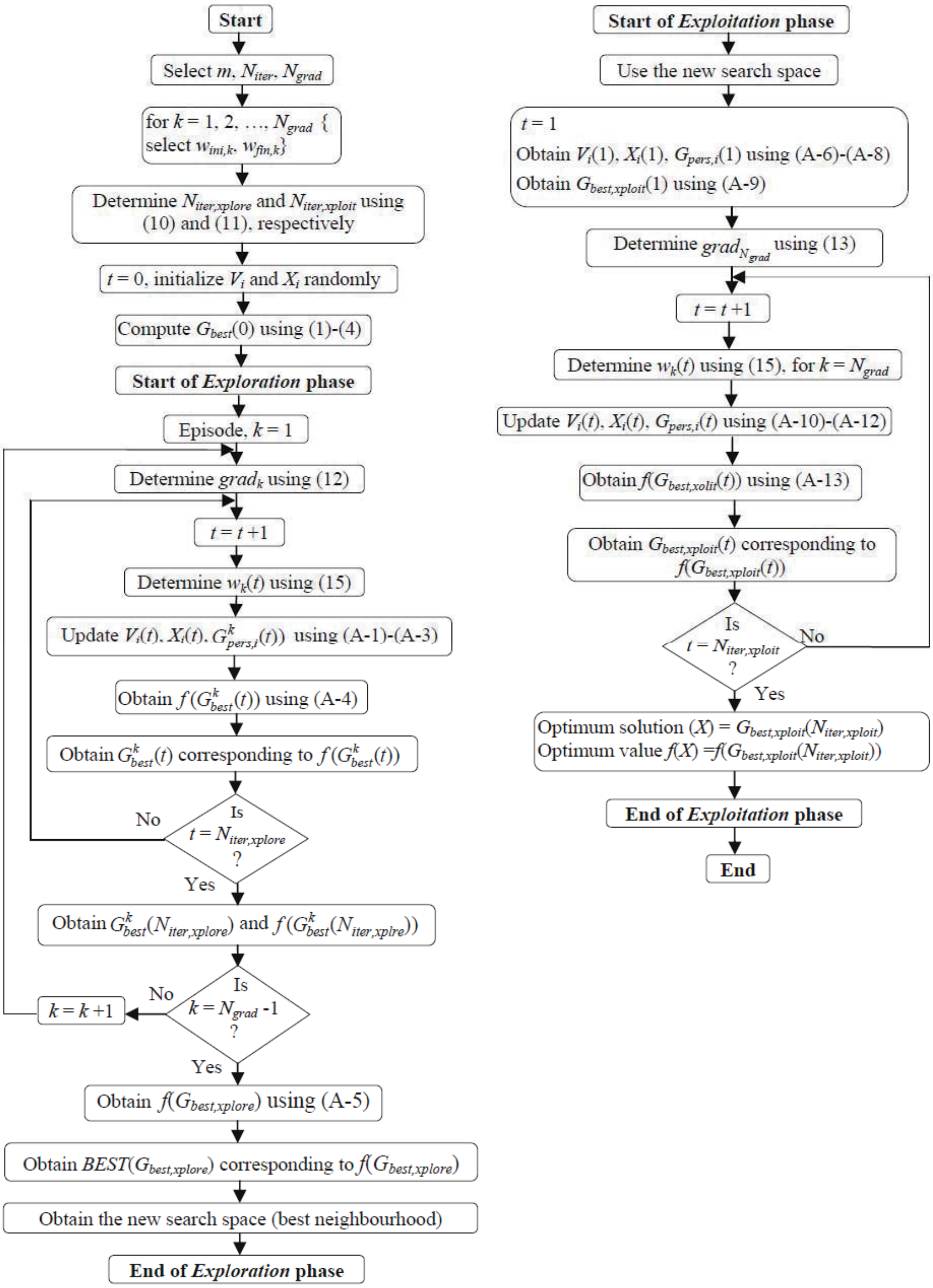

Start
Select m, N_{iter}, N_{grad}
for $k = 1, 2, \ldots, N_{grad}$ { select $w_{ini,k}$, $w_{fin,k}$}
Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (10) and (11), respectively
$t = 0$, initialize V_i and X_i randomly
Compute $G_{best}(0)$ using (1)-(4)
Start of *Exploration* phase
Episode, $k = 1$
Determine $grad_k$ using (12)
$t = t+1$
Determine $w_k(t)$ using (15)
Update $V_i(t)$, $X_i(t)$, $G^k_{pers,i}(t)$) using (A-1)-(A-3)
Obtain $f(G^k_{best}(t))$ using (A-4)
Obtain $G^k_{best}(t)$ corresponding to $f(G^k_{best}(t))$
Is $t = N_{iter,xplore}$?
No
Yes
Obtain $G^k_{best}(N_{iter,xplore})$ and $f(G^k_{best}(N_{iter,xplre}))$
Is $k = N_{grad}$ -1 ?
No
$k = k+1$
Yes
Obtain $f(G_{best,xplore})$ using (A-5)
Obtain $BEST(G_{best,xplore})$ corresponding to $f(G_{best,xplore})$
Obtain the new search space (best neighbourhood)
End of *Exploration* phase
Start of *Exploitation* phase
Use the new search space
$t = 1$
Obtain $V_i(1)$, $X_i(1)$, $G_{pers,i}(1)$ using (A-6)-(A-8)
Obtain $G_{best,xploit}(1)$ using (A-9)
Determine $grad_{N_{grad}}$ using (13)
$t = t+1$
Determine $w_k(t)$ using (15), for $k = N_{grad}$
Update $V_i(t)$, $X_i(t)$, $G_{pers,i}(t)$ using (A-10)-(A-12)
Obtain $f(G_{best,xolit}(t))$ using (A-13)
Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
Is $t = N_{iter,xploit}$?
No
Yes
Optimum solution $(X) = G_{best,xploit}(N_{iter,xploit})$
Optimum value $f(X) = f(G_{best,xploit}(N_{iter,xploit}))$
End of *Exploitation* phase
End

Fig. 1. Flowchart of the MG-PSO algorithm.

3.3 *An Illustrative Example:*

In order to explain the mechanism of MG-PSO algorithm, an example of a 2-dimensional shifted function, $f(x,y) = (x - 2)^2 + (y + 3)^2 + 9$ is illustrated in Fig. 2. It can be seen that x and y are shifted from the origin (0,0) by (2,-3). The optimum solution of the given function equals to 9 at $x = 2$ and $y = -3$. The purpose of the MG-PSO algorithm is to find the solutions x and y for which the $f(x,y)$ is minimized. The

MG-PSO algorithm was implemented using MATLAB in a personal computer with Intel (R) core (TM) 2 Duo CPU T6570 @ 2.1 GHz. RAM of 4GB and Windows 7, 64-bit operating system.

The MG-PSO algorithm was executed with m = 6, N_{iter} = 60, γ = 0.4, N_{grad} = 4, $N_{iter,xplore}$ = 24 and $N_{iter,xploit}$ = 36. The $w_{ini,k}$ and $w_{fin,k}$ are chosen within a range of [0,1] and are shown in Table 1. Selection of $grad_k$ (k = 1, 2, …, N_{grad}) was done by trial and error method. The results of the *Exploration* phase with three episodes (k = 1, 2 and 3) are shown in Table 1. At the end of episodes, $f(G_{best,xplore})$ is found as 9.0566 that is corresponding to episode 1. The $BEST(G_{best,xplore})$ corresponding to episode 1 is found as (2.1625,-2.8262). The range of new search space (best neighborhood) is obtained by taking “Floor” and “Ceil” of 2.1625, i.e., [2,3] and “Floor” and “Ceil” of -2.8262, i.e., [-3,-2]. Thus, the new search space is given by a range of x as [2,3] and a range of y as [-3,-2].

In *Exploitation* phase, the *Exploiters* navigate in the newly found search space, i.e., [2,3] [-3,-2], using one negative gradient (k = 4) that is less than that of *Exploration* phase. As shown in Table 1, at the end, the *Exploiters* obtain the optimum value $G_{best,xploit}$(36)) = 9.0000 and $G_{best,xploit}$(36) = (2.0,-3.0) give the optimum value of $f(x,y)$ and the optimum solution of (x,y) as 9.0 and optimum solution as (x,y) = (2.0, -3.0).

The dynamics of the MG-PSO algorithm in *Exploration* phase in three episodes are shown in Fig. 3. Figs. 3(a), (b) and (c) illustrate variation of inertia weight with iteration for the episodes 1, 2 and 3, respectively. It can be seen that the inertia weights follow different negative gradients. The convergence characteristics, i.e., variation $f(G_{best}^{k}(t))$ with iteration are also shown in this figure. At the end of iteration the optimum values obtained from the three episodes are given by 9.0566, 11.3179 and 10.6136. As shown in Figs. 3(d), (e) and (f) and Table 1, the corresponding optimum solutions for the three episodes are given by (2.1625,-2.8262), (3.4993,-2.7352) and (1.3299,-1.9208), respectively.

The movement of best particle $G_{best}^{k}(t)$ for three episodes, k = 1, 2 and 3 over 24 iterations are shown in Figs. 3(d), (e) and (f), respectively. The $G_{best}^{k}(t)$ follows its $grad_k$ in each episode. This diversity in negative gradients makes the MG-PSO algorithm to obtain different solutions, i.e., $G_{best}^{1}(t)$, $G_{best}^{2}(t)$ and $G_{best}^{3}(t)$ which are close to optimum solution. This means that the global best particle prevents the swarm from falling into a local minimum.

The dynamics of the MG-PSO algorithm in *Exploitation* phase is shown in Fig. 4. The variation of $f(G_{best,xploit})$ and inertia weight w_4 over 36 iterations are shown in Fig. 4(a). Fig. 4(b) shows movement of the best particle $G_{best,xploit}(t)$ in *Exploitation* phase within the new search space range of x and y as [2,3] and [-3,-2], respectively. The $G_{best,xploit}$ gives the optimum solution (x,y) = (2.0,-3.0) and optimum value of $f(x,y)$ = 9.0000.

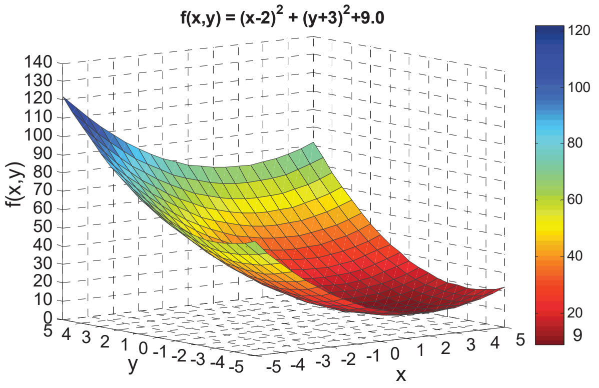
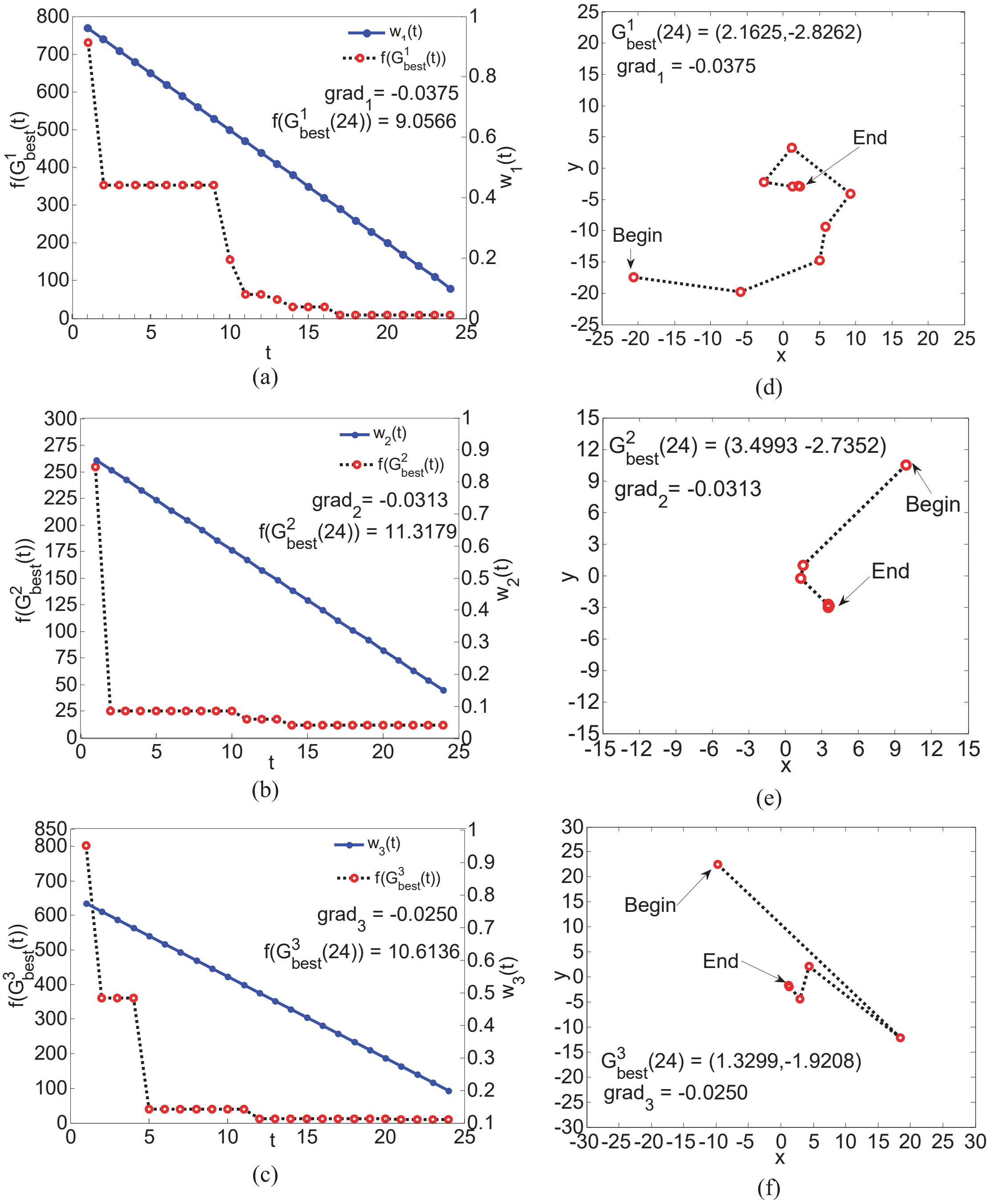


Fig. 2. Two-dimensional shifted function with minimum $f(x,y)$ = 9.0 at x = 2.0 and y = -3.0.

Table 1
Set of parameters and performance of the MG-PSO algorithm with d = 2, m = 6, N_{iter} = 60

Set of parameters	*Exploration* phase			*Exploitation phase*
	***k* = 1**	k = 2	k = 3	k = 4
$w_{ini,k}$	**1.0**	0.90	0.80	0.50
$w_{fin,k}$	**0.1**	0.15	0.20	0.35
$grad_k$	**-0.0375**	-0.0313	-0.0250	-0.0042
G^k_{best}	**(2.1625, -2.8262)**	(3.4993, -2.7352	(1.3299, -1.9208)	**(2.0000,-3.0000)**
$f(G^k_{best})$	**9.0566**	11.3179	10.6136	**9.0000**

The bold numbers indicate the best solution of the *i*th particle in a swarm during *Exploration* and *Exploitation* phases.



(a)

(d)

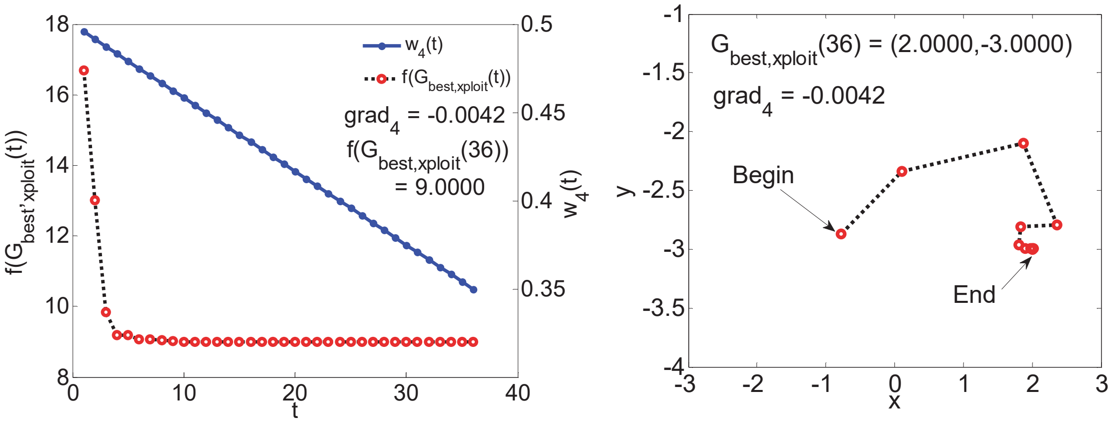
(b)

(e)

(c)

(f)

Fig. 3. Movement of the best particle in *Exploration* phase based on three different negative gradients. (a), (b) and (c): the change of $f(G^k_{best}(t))$ and $w_k(t)$, k =1, 2, 3, with iteration t ; (d), (e) and (f): movement of G^k_{best} in three episodes.



(a) (b)

Fig. 4. Movement of the best particle in *Exploitation* phase with one negative gradient. (a): the change of $f(G_{best,xploit}(t))$ and $w_4(t)$ with iteration t; (b): movement of G_{best} over 36 iterations.

3.4 *Observations:*

Some of the important observations of the MG-PSO algorithm are detailed, as follows:

3.4.1 Observation 1

Because of using a different negative gradient in each episode, in *Exploration* phase, the *Explorers* have ability to find a new neighbourhood within a *d*-dimensional search space. The global best particle is able to prevent the swarm from falling into a local minimum. In addition, the diversity in negative gradients enhances the local search ability of the *Exploiters* to obtain optimum solution. In addition, the combination of two phases is successfully applied to overcome the disadvantages of the gradient methods, as shown in Figs. 3 and 4.

3.4.2 Observation 2

In case of GPSO-*w* algorithm (6), two guides, $G_{pers,i}$ and G_{best}, are used to update the velocity vector $V_i(t)$. The continuous conflict between them until end of the iteration leads to loss of balance between global search and local search. However, in case of MG-PSO algorithm two phases are used. In the *Exploration* phase, using several episodes (each one with a different negative gradient) a new search space (new neighborhood) is obtained. This search space is used in *Exploitation* phase to achieve the optimum solution. In this way a balance is maintained between the global and local search spaces.

3.4.3 Observation 3

The $w_{N_{grad}}(t)$ that follows negative gradient, $grad_{N_{grad}}$ in the *Exploitation* phase is used to diminish the contribution of $X_i(t-1)$ while updating $V_i(t)$, $i = 1, 2, ..., m$. As $t \rightarrow \infty$, assume that the algorithm has converged. In such case,

$$\lim_{t\to\infty} X_i(t-1) = G_{pers,i}(t-1)$$

$$\lim_{t\to\infty} X_i(t-1) = G_{best,xploit}(t-1) \tag{16}$$

Then, Equation (A-10) becomes

$$\lim_{t\to\infty} V_i(t) = w_{N_{grad}}(t)\, V_i(t-1) = 0 \tag{17}$$

which implies that

$$\lim_{t\to\infty} X_i(t) = X_i(t-1) \tag{18}$$

Thus,

$$\lim_{t\to\infty} X_i(t) = \lim_{t\to\infty} G_{pers,i}(t) = \lim_{t\to\infty} G_{best,xploit}(t) \tag{19}$$

Thus, when iteration becomes large and the algorithm has converged, all the position vectors X_i and personal vectors $G_{pers,i}$, $i = 1, 2, \ldots, m$, move towards the best position vector, $G_{best,xploit}$. Fig. 5 shows performance of MG-PSO algorithm in the *Exploitation* phase at different iterations. At $t = 36$, as the algorithm converges, it can be seen that X_i, $G_{pers,i}$ ($i = 1, 2, \ldots, 6$) and $G_{best,xploit}$, all converge to the optimum solution (2.00,-3.00).

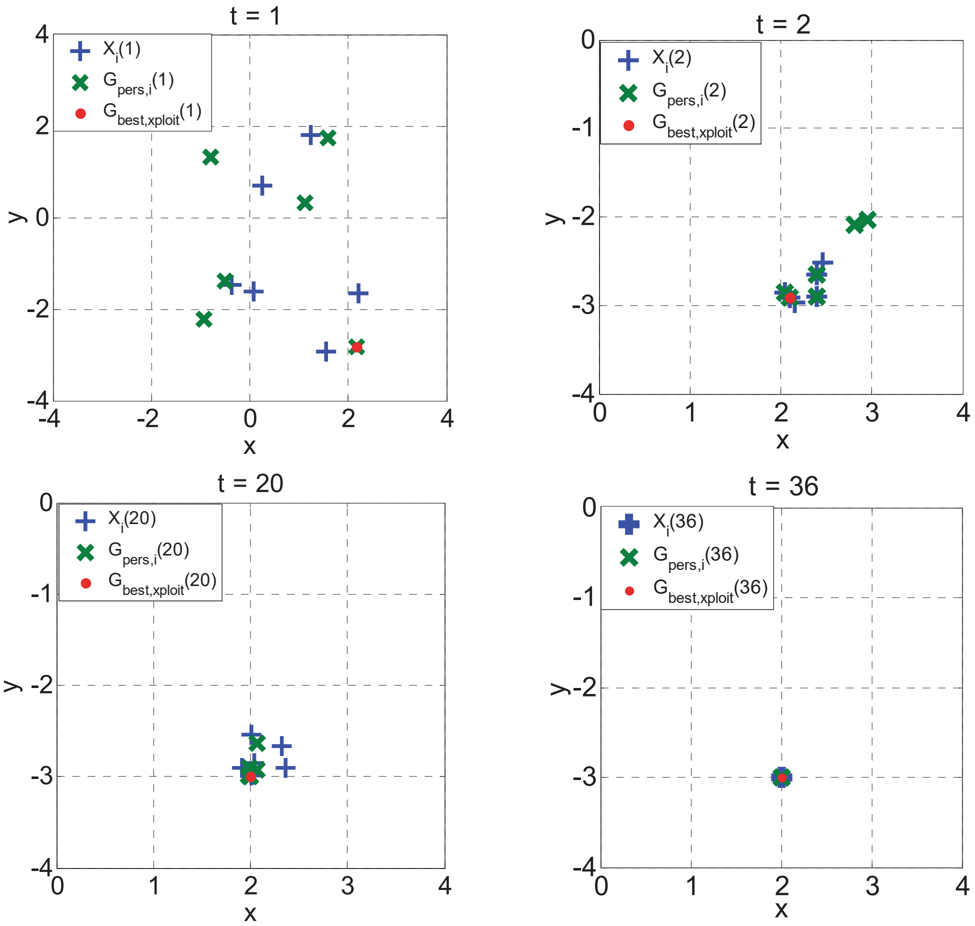


Fig. 5. Performance of the MG-PSO algorithm in *Exploitation* phase showing movement of X_i, $G_{pers,i}$, (i = 1, 2, ..., 6) and $G_{best,xploit}$ at different iterations to achieve a solution of (2.0,-3.0).

4 Application of MG-PSO algorithm to unimodal and multimodal CEC 2015 benchmark functions

Here, the simulation results carried out on ten selected benchmark functions that are taken from CEC 2015 [15-16]. Here we describe these benchmark functions and investigate performance of MG-PSO and GPSO-*w* algorithms along with a several competitive ECTs.

4.1 *Benchmark functions*

Ten benchmark functions listed in Table 2 are selected and used in this study. These benchmark functions are taken from the congress on evolutionary computation, CEC 2015, and are widely used in performance comparison of global optimization algorithms [15-16]. All the ten benchmark functions are minimization tasks. In addition, these are shifted and rotated unimodal and multimodal benchmark functions, and the global optimum solution x as shown in Table 2 is not located at the center of the search space. The optimum solution x is shifted to a new position vector, i.e., shifted global optimum, $O = [o_{i1}, o_{i2}, . . ., o_{id}]$, $i = 1, 2, ..., 10$, where d is the dimension of each benchmark function. In addition, all benchmark functions are rotated by a rotation matrix of size $d \times d$, M_i, $i =1, 2, ..., 10$. The rotation does not affect the shape of the objective function but increases the objective function complexity in finding global optimum. The M_i is applied to obtain the rotation and is generated from standard normally distributed entries using Gram-Schmidt orthnormalization process [15-16].

The ten benchmark functions are divided into two groups based on their significant physical properties. The first group involves three unimodal benchmark functions (f_1-f_3) and the second group consists of seven multimodal functions (f_4-f_{10}).

In Table 2, the name and mathematical expression of (f_1-f_{10}) are shown in columns 2 and 3, respectively. The “Accepted Error” value of each function is available in column 4. The “optimum x” and the minimum value of each objective function, “minimum $f(x)$” are shown in column 5 and column 6, respectively. The solution of each function is judged successful, when the algorithm reaches to a value smaller than the “Accepted Error”, in other words, the algorithm passes the test.

Table 2

Ten benchmark functions selected from CEC 2015 used in this study

f	Name	Objective function	Accepted Error	Optimum (x)	Minimum $f_i(x)$
f_1	Shifted and Rotated High Conditioned Elliptic [15]	$f_1(x)=\sum_{i=1}^{d}(10^6)^{(\frac{i-1}{d-1})}Z_{1,i}^2+F_1,$ $Z_1=M_1\times(x-O_1), x=[x_1,x_2,...,x_d], O_1=[o_{11},o_{12},...,o_{1d}]$	0.001	O_1	$F_1=100$
f_2	Shifted and Rotated Cigar [15]	$f_2(x)=Z_{2,1}^2+10^6\sum_{i=2}^{d}Z_{2,i}^2+F_2,$ $Z_2=M_2\times(x-O_2), x=[x_1,x_2,...,x_d], O_2=[o_{21},o_{22},...,o_{2d}]$	0.001	O_2	$F_2=200$
f_3	Shifted and Rotated Discus [16]	$f_3(x)=10^6Z_{3,1}^2+\sum_{i=2}^{d}Z_{3,i}^2+F_3,$ $Z_3=M_3\times(x-O_3), x=[x_1,x_2,...,x_d], O_3=[o_{31},o_{32},...,o_{3d}]$	0.001	O_3	$F_3=200$
f_4	Shifted and Rotated Ackley [15]	$f_4(x)=-a\ exp\ (-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}Z_{4,i}^2}\ )-exp\ (\frac{1}{d}\sum_{i=1}^{d}cos(cZ_{4,i}))+$ $a+exp\ (1)+F_4, a=20, b=0.2, c=2\pi$ $Z_4=M_4\times(x-O_4), x=[x_1,x_2,...,x_d], O_4=[o_{41},o_{42},...,o_{4d}]$	0.001	O_4	$F_4=300$
f_5	Shifted and Rotated Weierstrass [16]	$f_5(x)=\sum_{i=1}^{d}\left[\sum_{k=1}^{k_{max}}a^k\ cos\,(2\pi b^k(Z_{5,i}+0.5))\text{ - }d\sum_{k=1}^{k_{max}}a^k\ cos\,(\pi b^k)\right]+F_5$ $a=0.5, b=3, k_{max}=20, Z_5=M_5\times\left(\frac{0.5\times(x-O_{5i})}{100}\right), x=[x_1,x_2,...,x_d]$ $O_5=[o_{51},o_{52},...,o_{5d}]$	0.001	O_5	$F_5=300$
f_6	Shifted and Rotated Rastrigin [15]	$f_6(x)=10\,d+\sum_{i=1}^{d}[Z_{6,i}^2-10\,cos\,(2\pi Z_{6,i})]+F_6,$ $Z_6=M_6\times\left(\frac{5.12\times(x-O_6)}{100}\right), x=[x_1,x_2,...,x_d], O_6=[o_{61},o_{62},...,o_{6d}]$	0.001	O_6	$F_6=400$
f_7	Shifted and Rotated Katsuura [16]	$f_7(x)=\frac{10}{d^2}\prod_{i=1}^{d}\left[1+i\sum_{i=1}^{32}\frac{\lvert 2^i\times Z_{7,i}-round\ (2^i\times Z_{7,i})\rvert}{2^i}\right]^{\frac{10}{d^{1.2}}}-\frac{10}{d^2}+F_7$ $Z_7=M_7\times\left(\frac{5\times(x-O_{7i})}{100}\right), x=[x_1,x_2,...,x_d], O_{7i}=[o_{71},o_{72},...,o_{7d}]$	0.001	O_7	$F_7=500$
f_8	Shifted and Rotated HappyCat [16]	$f_8(x)=\left[\sum_{i=1}^{d}Z_{8,i}^2-d\right]^{\frac{1}{4}}+(\frac{0.5\sum_{i=1}^{d}Z_{8,i}^2+\sum_{i=1}^{d}Z_{8,i})}{\mathrm{d}})+0.5+F_8,$ $Z_8=M_8\times\left(\frac{5\times(x-O_{8i})}{100}\right), x=[x_1,x_2,...,x_d], O_{8i}=[o_{81},o_{82},...,o_{8d}]$	0.001	O_8	$F_8=600$
f_9	Shifted and Rotated HGBat [16]	$f_9(x)=\left[(\sum_{i=1}^{d}Z_{9,i}^2)^2-(\sum_{i=1}^{d}Z_{9,i})^2\right]^{\frac{1}{2}}+(\frac{0.5\sum_{i=1}^{d}Z_{9,i}^2+\sum_{i=1}^{d}Z_{9,i})}{d})+0.5+F_9,$ $Z_9=M_9\times\left(\frac{5\times(x-O_9)}{100}\right), x=[x_1,x_2,...,x_d], O_9=[o_{91},o_{92},...,o_{9d}]$	0.001	O_9	$F_9=700$
f_{10}	Shifted and Rotated Expanded Griewank plus Rosenbrock [16]	$f_g(x)=\frac{1}{4000}\sum_{i=1}^{d}Z_{10,i}^2-\prod_{i=1}^{d}cos\ (\frac{Z_{10,i}}{\sqrt{i}})+1$ $f_r(x)=\sum_{i=1}^{d-1}[100\ (Z_{10,i+1}-Z_{10,i}^2)^2+(Z_{10,i}-1)^2]$ $f_{10}(x)=f_g(f_r(Z_{10,1},Z_{10,2})+f_g(f_r(Z_{10,2},Z_{10,3})+\cdots+$ $f_g(f_r(Z_{10,d-1},Z_{10,d})+f_g(f_r(Z_{10,d},Z_{10,1})+F_{10}$ $Z_{10}=M_{10}\times\left(\frac{5\times(x-O_{10})}{100}\right)+1, x=[x_1,x_2,...,x_d],$ $O_{10}=[o_{101},o_{102},...,o_{10d}]$	0.001	O_{10}	$F_{10}=800$

4.2 *Performance measures and experimental setup*

To study the fitness values, convergence rate, accuracy, consistency, robustness and reliability of different algorithms [41-44], eight performance measures as explained below are considered. Let m be the number of particles inside a swarm. Each algorithm is run over N_{run} times with N_{iter} iterations.

1. Number of function evaluations (NFE): It is used as a measure of computational complexity of the algorithms. The NFE is the number of times the objective function $f(x)$ is evaluated in one run of the algorithm and is given by

$$\text{NFE} = m \times N_{iter} \tag{20}$$

2. Fitness value (FV): It is defined as the average value obtained from N_{run} times at each NFE.
3. Best fitness value (BFV): It is defined as the minimum optimized $f(x)$ value obtained from N_{run} independent runs.
4. Worst fitness value (WFV): It is defined as the maximum optimized $f(x)$ value obtained from N_{run} independent runs.
5. Mean fitness value (MFV): It is defined as the average of the N_{run} BFVs.
6. Minimum function error value (MFEV): It is defined as the difference between minimum $f(x)$, i.e., column 6 in Table 2 and MFV. The MFEV is given by

$$\text{MFEV} = \left|\text{minimum} f(x) - \text{MFV}\right| \tag{21}$$

7. Standard deviation (σ): The σ is the standard deviation of the N_{run} BFVs.
8. Success rate (SR): An algorithm is successful if the MFEV of each function falls below the "Accepted Error". The SR is used as a measure of reliability of the algorithm. The SR in percentage is given by

$$\text{SR} = \frac{\text{Number of successful runs}}{N_{run}} \times 100 \tag{22}$$

9. Reliability rate (RR): The RR of an algorithm over all the ten benchmark functions is defined

$$\text{RR} = \frac{1}{10} \sum_{i=1}^{10} \text{SR}_i \tag{23}$$

where SR_i is the success rate of the benchmark function $f_i(x)$, $i = 1, 2, ..., 10$.

10. Average execution time (AET): It is the time consumed by an algorithm until it reaches to MFEV, averaged over N_{run} independent runs.

In order to measure the fitness values, convergence rate, accuracy, consistency, robustness and reliability of each algorithm, the proposed MG-PSO and GPSO-*w* algorithms were evaluated using the ten shifted and rotated unimodal and multimodal benchmark functions given in Table 2. Each benchmark function is tested with d = 30 and d = 100 dimensions where d is dimension of the $f(x)$. Based on the suggestion by the CEC 2015 [16], the optimization task has been carried out for N_{run} = 20 independent runs. The GPSO-*w* and MG-PSO algorithms are terminated when reaching the MFEV of each is smaller than 1.00×10^{-03}. The number of particles is selected as 20, in both MG-PSO and GPSO-*w* algorithms. The MG-PSO and GPSO-*w* algorithms run with maximum NFE = 100,000. Thus, from (20), N_{iter} = 5,000. The acceleration coefficients of c_1 and c_2 in GPSO-*w* and c in MG-PSO algorithms were set at 2.00 and 2.05, respectively, using trial and error method. The parameters $r(t)$, $r_1(t)$ and $r_2(t)$ are generated randomly within a range of [0,1]. In MG-PSO algorithm, in case of ten benchmark functions (f_1-f_{10}) with d = 30, two negative gradients were selected (N_{grad} = 2) by trial and error method, one for *Exploration* phase and another one for *Exploitation* phase. Whereas, in case of f_1-f_{10} with d = 100, three negative gradients were selected (N_{grad} = 3) by trial and error method, two for *Exploration* phase and another one for *Exploitation* phase.

Besides, the shifted global optimum vector, O_i (i =1, 2, …, 10), for each function is randomly distributed in $[-80,80]^d$ and an orthogonal (rotation) matrix M_i (i =1, 2, …, 10) of each function is generated using Gram-Schmidt orthnormalization process. Another set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Tables 3 and 4.

Table 3

Set of parameters of the MG-PSO and GPSO-*w* algorithms used for ten benchmark functions (CEC 2015) with $d = 30$

f	Algorithm		Set of parameters				
			γ	N_{iter}	$w_{ini,k}$	$w_{fin,k}$	$grad_k$
f_1	MG-PSO	*Exploration* phase k =1	0.30	1,500	1.00	0.10	-6.00×10^{-04}
		Exploitation phase k = 2	0.70	3,500	0.50	0.25	-7.14×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_2	MG-PSO	*Exploration* phase k =1	0.30	1,500	1.00	0.20	-5.33×10^{-04}
		Exploitation phase k = 2	0.70	3,500	0.40	0.15	-7.14×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_3	MG-PSO	*Exploration* phase k =1	0.30	1,500	0.95	0.15	-5.33×10^{-04}
		Exploitation phase k = 2	0.70	3,500	0.45	0.25	-5.71×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_4	MG-PSO	*Exploration* phase k =1	0.40	2,000	1.00	0.10	-4.50×10^{-04}
		Exploitation phase k = 2	0.60	3,000	0.50	0.30	-6.66×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_5	MG-PSO	*Exploration* phase k =1	0.30	1,500	0.90	0.10	-5.33×10^{-04}
		Exploitation phase k = 2	0.70	3,500	0.40	0.15	-7.14×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_6	MG-PSO	*Exploration* phase k =1	0.40	2,000	0.90	0.10	-4.00×10^{-04}
		Exploitation phase k = 2	0.60	3,000	0.35	0.10	-8.33×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_7	MG-PSO	*Exploration* phase k =1	0.40	2,000	0.95	0.45	-2.50×10^{-04}
		Exploitation phase k = 2	0.60	3,000	0.45	0.25	-6.66×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_8	MG-PSO	*Exploration* phase k =1	0.40	2,000	1.00	0.10	-4.50×10^{-04}
		Exploitation phase k = 2	0.60	3,000	0.40	0.25	-5.00×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_9	MG-PSO	*Exploration* phase k =1	0.40	2,000	0.90	0.20	-3.50×10^{-04}
		Exploitation phase k = 2	0.60	3,000	0.40	0.15	-8.33×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}
f_{10}	MG-PSO	*Exploration* phase k =1	0.30	1,500	0.90	0.25	-4.33×10^{-04}
		Exploitation phase k = 2	0.70	3,500	0.55	0.35	-5.71×10^{-05}
	GPSO-*w*		-	5,000	0.90	0.40	-1.00×10^{-04}

Table 4
Set of parameters of the MG-PSO and GPSO-*w* algorithms used for ten benchmark functions (CEC 2015) with $d = 100$

f	Algorithm			Set of parameters				
				γ	N_{iter}	$w_{ini,k}$	$w_{fin,k}$	$grad_k$
f_1	MG-PSO	*Exploration* phase	$k = 1$	0.65	3,250	0.95	0.10	-2.61×10^{-04}
			$k = 2$	0.65	3,250	0.85	0.15	-2.15×10^{-04}
		Exploitation phase	$k = 3$	0.35	1,750	0.30	0.15	-8.57×10^{-05}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_2	MG-PSO	*Exploration* phase	$k = 1$	0.70	3,500	1.00	0.15	-2.43×10^{-04}
			$k = 2$	0.70	3,500	0.95	0.25	-2.00×10^{-04}
		Exploitation phase	$k = 3$	0.30	1500	0.40	0.10	-1.66×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_3	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.90	0.10	-3.20×10^{-04}
			$k = 2$	0.50	2,500	0.80	0.15	-2.60×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.45	0.15	-1.20×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_4	MG-PSO	*Exploration* phase	$k = 1$	0.60	3,000	0.90	0.15	-2.50×10^{-04}
			$k = 2$	0.60	3,000	0.85	0.10	-2.16×10^{-04}
		Exploitation phase	$k = 3$	0.40	2,000	0.50	0.30	-1.00×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_5	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.80	0.10	-3.20×10^{-04}
			$k = 2$	0.50	2,500	0.75	0.10	-3.00×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.50	0.25	-2.00×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_6	MG-PSO	*Exploration* phase	$k = 1$	0.55	2,750	1.00	0.15	-3.09×10^{-04}
			$k = 2$	0.55	2,750	0.95	0.15	-2.91×10^{-04}
		Exploitation phase	$k = 3$	0.45	2,250	0.50	0.35	-6.66×10^{-05}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_7	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.95	0.15	-3.20×10^{-04}
			$k = 2$	0.50	2,500	0.85	0.15	-2.80×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.45	0.25	-8.00×10^{-05}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_8	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.95	0.10	-3.40×10^{-04}
			$k = 2$	0.50	2,500	0.75	0.10	-2.60×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.50	0.25	-1.00×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_9	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.95	0.20	-3.00×10^{-04}
			$k = 2$	0.50	2,500	0.92	0.22	-2.80×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.60	0.25	-1.40×10^{-04}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}
f_{10}	MG-PSO	*Exploration* phase	$k = 1$	0.50	2,500	0.90	0.10	-3.20×10^{-04}
			$k = 2$	0.50	2,500	0.85	0.15	-2.6×10^{-04}
		Exploitation phase	$k = 3$	0.50	2,500	0.55	0.35	8.00×10^{-05}
	GPSO-*w*			-	5,000	0.90	0.40	-1.00×10^{-04}

4.3 *Performance comparison between MG-PSO and GPSO-w algorithms*

An extensive performance comparison between MG-PSO and GPSO-*w* algorithms with several performance measures as explained below.

4.3.1 *Fitness values*

Performance comparison between MG-PSO and GPSO-*w* algorithms in terms of BFV, WFV, MFV, MFEV, σ and AET are shown in Tables 5 and 6. One can be seen from Tables 5 and 6 that in case of GPSO-*w* algorithm, the three fitness values, BFV, WFV and MFV differ substantially from their optimum values for all the ten functions (f_1-f_{10}) with d = 30 and d = 100. Whereas, in MG-PSO algorithm, the three fitness values are the same as their optimum values for all the ten functions (f_1-f_{10}). The MFEV of GPSO-*w* algorithm is far from the "Accepted Error" in (f_1-f_{10}). However, in MG-PSO algorithm, the MFEV is smaller than "Accepted Error" as shown in Tables 5 and 6. The MFEV = 0.0 for the ten function (f_1-f_{10}). In terms of the σ, it remains close to 0.0 in MG-PSO algorithm, indicating high consistency and reliability of the MG-PSO algorithm. The results shown in Tables 5 and 6, thus proves that MG-PSO algorithm is more accurate, stable and robust compared to the GPSO-*w* algorithm. In terms of the AET, the MG-PSO algorithm reaches "Accepted Error" within a specific AET as shown in Tables 5 and 6. However, GPSO-*w* algorithm could not reach "Accepted Error", indicating that GPSO-*w* is unable to solve these ten shifted and rotated under high-dimensional search space. Whereas, MG-PSO algorithm successfully achieves the optimum solution for all the ten benchmark functions with d = 30 and d = 100.

Table 5

Performance comparison between MG-PSO and GPSO-*w* algorithms on ten benchmark functions (CEC 2015) with $d = 30$

f	Minimum $f(x)$	Accepted Error	Fitness	GPSO-*w*	MG-PSO	f	Minimum $f(x)$	Accepted Error	Fitness	GPSO-*w*	MG-PSO
f_1	100	1×10^{-03}	BFV	1.0100×10^{02}	1.0000×10^{02}	f_6	400	1×10^{-03}	BFV	5.4213×10^{02}	4.0000×10^{02}
			WFV	1.3660×10^{08}	1.0000×10^{02}				WFV	8.2457×10^{02}	4.0000×10^{02}
			MFV	1.0267×10^{02}	1.0000×10^{02}				MFV	6.1298×10^{02}	4.0000×10^{02}
			MFEV	2.6700×10^{00}	0.0000				MFEV	2.1298×10^{02}	0.0000
			σ	8.8217×10^{00}	0.0000				σ	1.4678×10^{02}	0.0000
			AET (sec)	-	26.2786				AET (sec)	-	8.9211
f_2	200	1×10^{-03}	BFV	2.0030×10^{02}	2.0000×10^{02}	f_7	500	1×10^{-03}	BFV	5.5525×10^{02}	5.0000×10^{02}
			WFV	2.0032×10^{02}	2.0000×10^{02}				WFV	5.8913×10^{04}	5.0000×10^{02}
			MFV	2.0030×10^{02}	2.0000×10^{02}				MFV	7.4021×10^{03}	5.0000×10^{02}
			MFEV	3.0000×10^{-01}	0.0000				MFEV	6.9021×10^{03}	0.0000
			σ	5.0663×10^{-03}	0.0000				σ	1.7701×10^{04}	0.0000
			AET (sec)	-	36.1856				AET (sec)	-	16.3386
f_3	200	1×10^{-03}	BFV	2.3214×10^{02}	2.0000×10^{02}	f_8	600	1×10^{-03}	BFV	6.7597×10^{02}	6.0000×10^{02}
			WFV	2.0134×10^{02}	2.0000×10^{02}				WFV	1.5060×10^{03}	6.0000×10^{02}
			MFV	2.0754×10^{02}	2.0000×10^{02}				MFV	9.0597×10^{02}	6.0000×10^{02}
			MFEV	7.5400×10^{00}	0.0000				MFEV	3.0597×10^{02}	0.0000
			σ	6.4387×10^{00}	0.0000				σ	2.6147×10^{02}	0.0000
			AET (sec)	-	27.8475				AET (sec)	-	13.6193
f_4	300	1×10^{-03}	BFV	3.2280×10^{02}	3.0000×10^{02}	f_9	700	1×10^{-03}	BFV	1.0023×10^{03}	7.0000×10^{02}
			WFV	3.2283×10^{02}	3.0000×10^{02}				WFV	1.2923×10^{03}	7.0000×10^{02}
			MFV	3.2289×10^{02}	3.0000×10^{02}				MFV	1.0906×10^{03}	7.0000×10^{02}
			MFEV	2.2800×10^{01}	0.0000				MFEV	5.9094×10^{02}	0.0000
			σ	6.4987×10^{-03}	5.0797×10^{-56}				σ	3.906×10^{02}	0.0000
			AET (sec)	-	21.9311				AET (sec)	-	10.7789
f_5	300	1×10^{-03}	BFV	3.8910×10^{02}	3.0000×10^{02}	f_{10}	800	1×10^{-03}	BFV	6.0686×10^{02}	8.0000×10^{02}
			WFV	3.9000×10^{02}	3.0000×10^{02}				WFV	1.1069×10^{03}	8.0000×10^{02}
			MFV	3.8960×10^{02}	3.0000×10^{02}				MFV	9.0618×10^{02}	8.0000×10^{02}
			MFEV	8.9600×10^{01}	0.0000				MFEV	1.0618×10^{02}	0.0000
			σ	2.9843×10^{-01}	1.0957×10^{-61}				σ	1.1905×10^{02}	0.0000
			AET (sec)	-	16.1625				AET (sec)	-	28.7829

Table 6
Performance comparison between MG-PSO and GPSO-*w* algorithms on ten benchmark functions (CEC 2015) with d =100

f	Minimum $f(x)$	Accepted Error	Fitness	GPSO-*w*	MG-PSO	f	Minimum $f(x)$	Accepted Error	Fitness	GPSO-*w*	MG-PSO
f_1	100	1×10^{-03}	BFV	3.5443×10^{03}	1.0000×10^{02}	f_6	400	1×10^{-03}	BFV	1.1166×10^{03}	4.0000×10^{02}
			WFV	2.9834×10^{10}	1.0000×10^{02}				WFV	2.6532×10^{06}	4.0000×10^{02}
			MFV	7.4585×10^{09}	1.0000×10^{02}				MFV	2.8028×10^{05}	4.0000×10^{02}
			MFEV	7.4585×10^{09}	0.0000				MFEV	2.7988×10^{05}	0.0000
			σ	2.3048×10^{09}	0.0000				σ	8.3621×10^{05}	0.0000
			AET (sec)	-	87.5865				AET (sec)	-	42.8296
f_2	200	1×10^{-03}	BFV	2.8756×10^{03}	2.0000×10^{02}	f_7	500	1×10^{-03}	BFV	3.5672×10^{03}	5.0000×10^{02}
			WFV	4.7665×10^{08}	2.0000×10^{02}				WFV	1.0134×10^{06}	5.0000×10^{02}
			MFV	7.1500×10^{07}	2.0000×10^{02}				MFV	4.5238×10^{05}	5.0000×10^{02}
			MFEV	7.1499×10^{07}	0.0000				MFEV	4.5188×10^{05}	0.0000
			σ	1.7462×10^{08}	0.0000				σ	5.1634×10^{05}	0.0000
			AET (sec)	-	126.6174				AET (sec)	-	58.6221
f_3	200	1×10^{-03}	BFV	4.2260×10^{03}	2.0000×10^{02}	f_8	600	1×10^{-03}	BFV	2.0234×10^{03}	6.0000×10^{02}
			WFV	6.0347×10^{08}	2.0000×10^{02}				WFV	8.3452×10^{07}	6.0000×10^{02}
			MFV	1.8104×10^{08}	2.0000×10^{02}				MFV	4.1727×10^{07}	6.0000×10^{02}
			MFEV	1.8104×10^{08}	0.0000				MFEV	4.1726×10^{07}	0.0000
			σ	2.8373×10^{08}	0.0000				σ	4.2622×10^{07}	0.0000
			AET (sec)	-	96.0738				AET (sec)	-	61.2868
f_4	300	1×10^{-03}	BFV	8.1423×10^{04}	3.0000×10^{02}	f_9	700	1×10^{-03}	BFV	2.1435×10^{03}	7.0000×10^{02}
			WFV	4.9753×10^{07}	3.0000×10^{02}				WFV	7.3487×10^{06}	7.0000×10^{02}
			MFV	7.7177×10^{06}	3.0000×10^{02}				MFV	2.0057×10^{06}	7.0000×10^{02}
			MFEV	7.7174×10^{06}	0.0000				MFEV	2.0050×10^{06}	0.0000
			σ	1.8038×10^{07}	8.2341×10^{-55}				σ	3.3489×10^{06}	0.0000
			AET (sec)	-	76.3164				AET (sec)	-	55.1879
f_5	300	1×10^{-03}	BFV	1.2001×10^{04}	3.0000×10^{02}	f_{10}	800	1×10^{-03}	BFV	1.9087×10^{03}	8.0000×10^{02}
			WFV	7.3401×10^{07}	3.0000×10^{02}				WFV	3.6723×10^{06}	8.0000×10^{02}
			MFV	1.1600×10^{07}	3.0000×10^{02}				MFV	1.5312×10^{06}	8.0000×10^{02}
			MFEV	1.1599×10^{07}	0.0000				MFEV	1.5304×10^{06}	0.0000
			σ	2.7494×10^{07}	2.6578×10^{-59}				σ	1.8484×10^{06}	0.0000
			AET (sec)	-	59.4783				AET (sec)	-	121.7515

4.3.2 *Success rate and reliability rate*

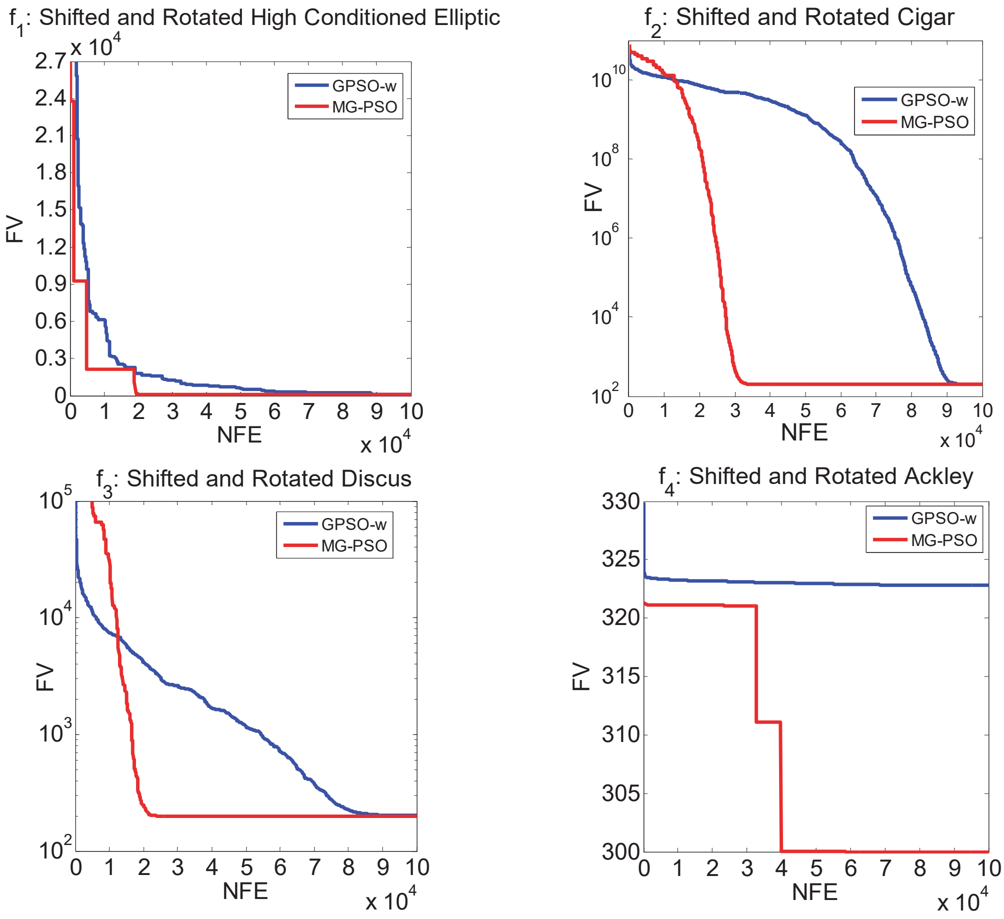
Here, the performance comparison between the MG-PSO and GPSO-*w* algorithms with N_{run} = 20 (independent runs) in terms of SR and RR is carried out. The MG-PSO algorithm was successful in all the ten functions giving rise to SR of 100%. Whereas, the GPSO-*w* algorithm fails in all the ten CEC 2015 benchmark functions. Thus, RR of MG-PSO and GPSO-*w* algorithms are found to be 100% and 0.0%, respectively.

4.3.3 *Convergence characteristics*

Fig. 6 shows the convergence characteristics of MG-PSO and GPSO-*w* algorithms for ten shifted and rotated CEC 2015 benchmark functions $f_1 - f_{10}$ with d = 30. The comparison is obtained in terms of FV averaged over N_{run} times at each NFE. It can be seen that, in case of MG-PSO algorithm, the FV reduces to minimum value with NFE less than 3.0×10^{04} (i.e., N_{iter} = 1,500) except for f_4. Whereas, in case of GPSO-*w* algorithm, the FV fails to converge and remains above the “Accepted Error”, indicating failure of the algorithm.

In order to highlight the superior performance of the MG-PSO algorithm over the GPSO-*w* algorithm, in Fig. 7 the MFEV over N_{run} times at each NFE with d = 30 have been provided for the four selected CEC 2015 benchmark functions, f_1, f_7, f_8 and f_{10}. The MFEVs obtained by GPSO-*w* algorithm are much above “Accepted Error”. Whereas, the MG-PSO algorithm was successful as the MFEVs in these four benchmark functions remain below the “Accepted Error”. Similar observations were made for the remaining benchmark functions with d = 100.

The above mentioned observations provide the evidence of superior performance of the MG-PSO over GPSO-*w* algorithm in terms of three fitness values, σ, convergence, SR, RR and convergence rate.



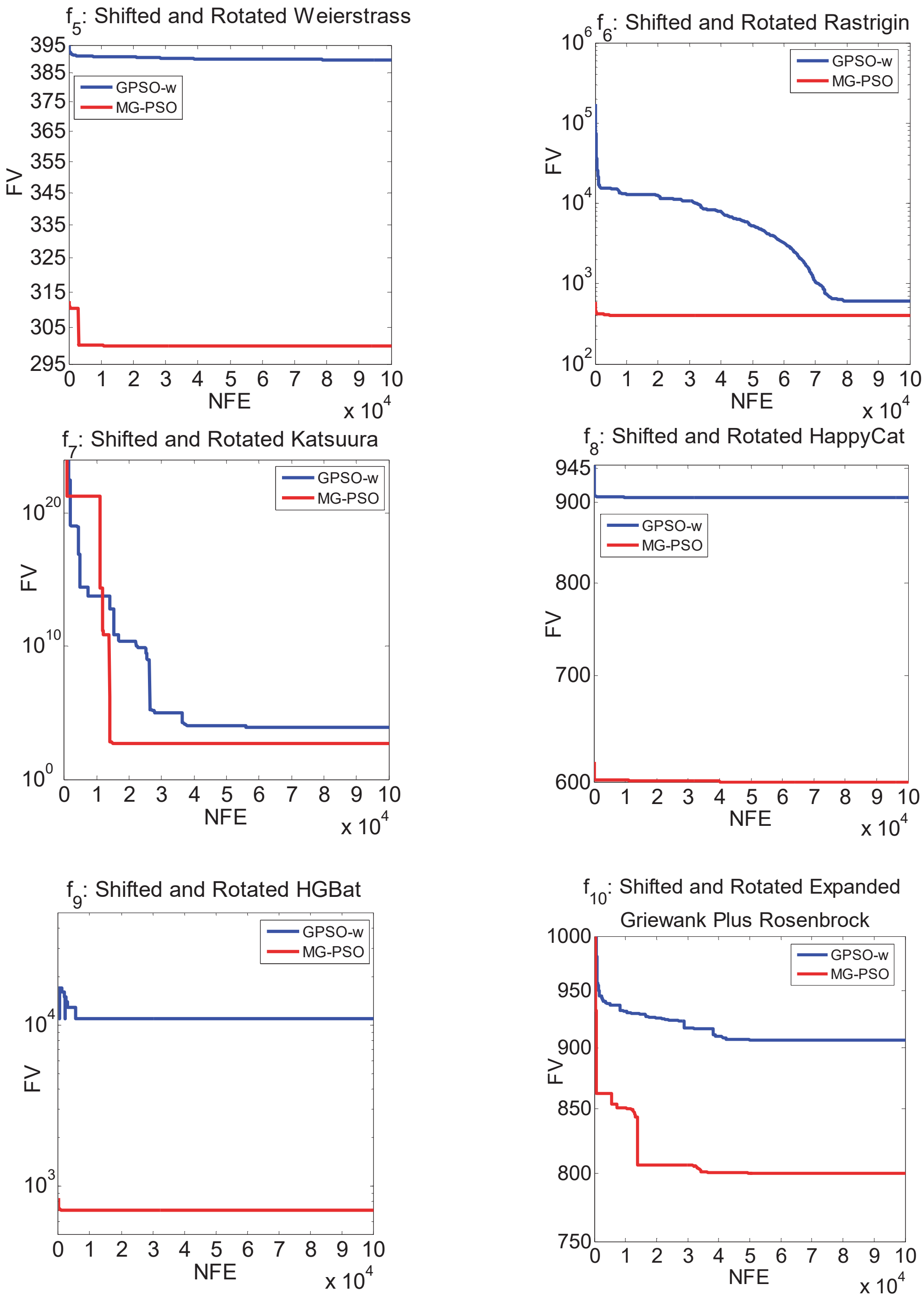
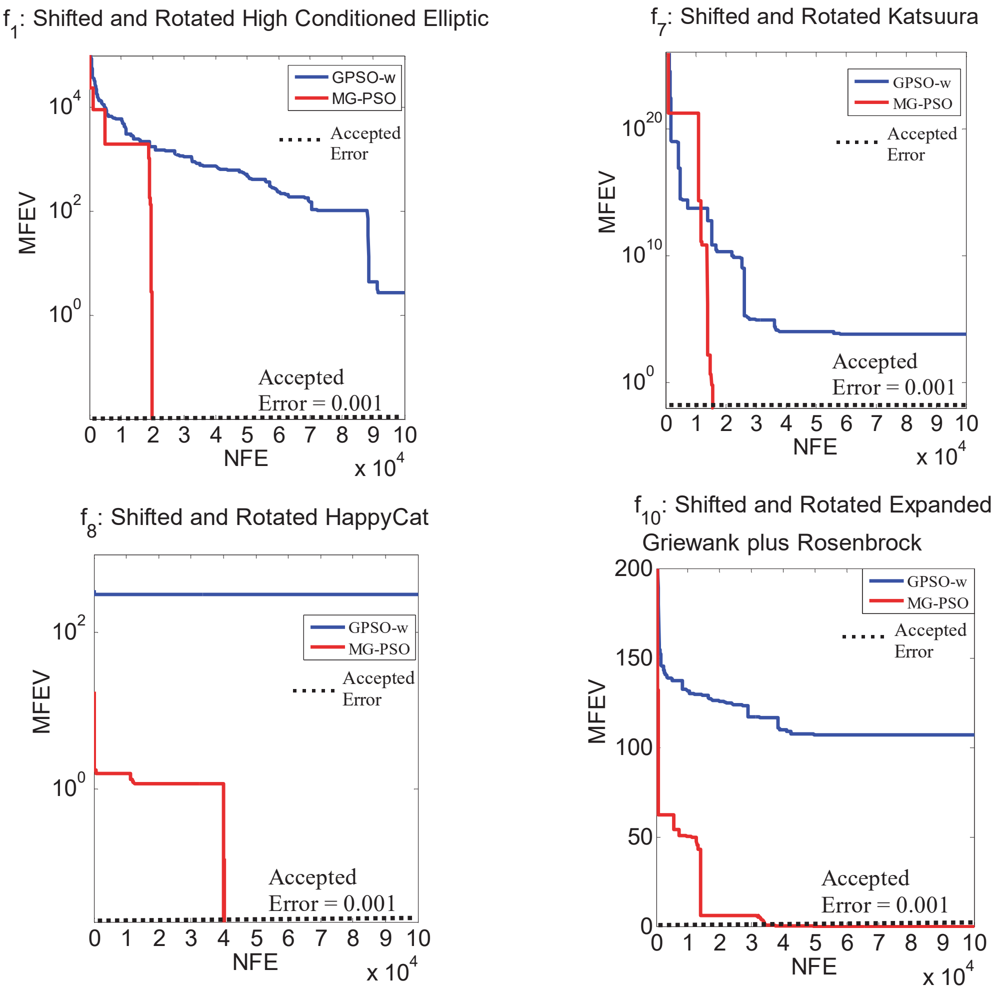


Fig. 6. Comparison of convergence characteristics between MG-PSO and GPSO-*w* algorithms for f_1-f_{10}.



f_1: Shifted and Rotated High Conditioned Elliptic

f_7: Shifted and Rotated Katsuura

f_8: Shifted and Rotated HappyCat

f_{10}: Shifted and Rotated Expanded Griewank plus Rosenbrock

Fig. 7. The FEVs averaged over 20 runs at each NFE obtained by MG-PSO and GPSO-*w* algorithms for f_1, f_7, f_8 and f_{10}.

4.4 *Sensitivity analysis of MG-PSO algorithm against swarm size*

In order to study the sensitivity analysis of the proposed MG-PSO algorithm against change in swarm population *m*, four benchmarks functions f_1, f_7, f_8 and f_{10} were selected. The test was carried out with N_{iter} = 5,000, N_{run} = 20 and d = 30. The set of parameters given in Table 3 were used in MG-PSO algorithm.

Table 7 shows sensitivity analysis of the MG-PSO algorithm with variation of swarm population, *m*, from 5 to 25 in terms of BFV, WFV, MFV and AET for f_1, f_7, f_8 and f_{10}. When *m* increases from 5 to 15, the fitness values do not yield optimum solution. Whereas, *m* = 20, the fitness values provide optimum solution. Any further increase of *m* (*m* = 25), increase the AET without giving any further improvement in performance. Therefore, the appropriate value of *m* = 20 is selected in this study.

Table 7
Sensitivity analysis of the MG-PSO algorithm with increase size of swarm

f	Minimum $f(x)$	Fitness	m = 5	AET (sec)	m = 10	AET (sec)	m = 15	AET (sec)	m = 20	AET (sec)	m = 25	AET (sec)
f_1	100	BFV	100.00	15.83	100.00	18.80	100.00	21.23	100.00	25.34	100.00	27.56
		WFV	104.75		100.52		100.01		100.00		100.00	
		MFV	101.63		100.10		100.00		100.00		100.00	
f_7	500	BFV	500.00	11.65	500.00	13.03	500.00	14.97	500.00	16.76	500.00	18.32
		WFV	503.15		500.23		500.03		500.00		500.00	
		MFV	501.28		500.10		500.00		500.00		500.00	
f_8	600	BFV	600.00	12.34	600.00	14.20	600.00	15.76	600.00	17.61	600.00	20.54
		WFV	601.87		600.01		600.00		600.00		600.00	
		MFV	600.09		600.02		600.00		600.00		600.00	
f_{10}	800	BFV	800.00	13.45	800.00	18.45	800.00	20.87	800.00	22.56	800.00	23.97
		WFV	802.32		800.85		800.02		800.00		800.00	
		MFV	800.16		800.12		800.00		800.00		800.00	

4.5 *Performance comparison between MG-PSO algorithm and other ECTs*

Here, we verify the performance of the proposed MG-PSO algorithm by comparing it with few ECTs recently reported by other authors [21-24] and [26-32].

4.5.1 *Shifted and rotated benchmark functions: f_1, f_2, f_4 and f_6 with d* = 30

The three top-ranked algorithms in the CEC 2015 [15] learning based papers (CEC 2015-LBP) are SPS-L-SHADE-EIG [26], DEsPA [27], and MVMO [28], in which the four shifted and rotated benchmark functions, i.e., f_1, f_2, f_4 and f_6 with d = 30, are considered. The set of parameters given in Table 3 were used in MG-PSO algorithm. Here, the performance of the MG-PSO algorithm against the above three top-ranked algorithms and a few competitive algorithms from [31] as well as OPSO [24] for the four functions, has been compared.

The comparison has been achieved with N_{run} = 20 and for each run the maximum NFE is 10,000×d, i.e., 10,000×30 = 300,000, where d is dimension of $f(x)$. The criterion used in this comparison depends on the values of maximum NFE and MFEV (21). When the algorithm reaches NFE = 300,000, the MFEV is recorded as a better result. The algorithm obtains a best result when the MFEV is 0.0 or close to 0.0.

Table 8 presents the results of the MFEV and the corresponding σ obtained by MG-PSO algorithm and eight ECTs. We can see that the MG-PSO algorithm is significantly superior to LLUDE, SaDE, JADE and CoDE [31] in solving f_1, f_2, f_4 and f_6. While comparing with CEC 2015-LBP [15] ranked algorithms, performance of the MG-PSO algorithm found similar or better than that of the three top-ranked algorithms for the four functions. Furthermore, performance of the MG-PSO algorithm is found similar to that of OPSO algorithm [24] for the functions f_1, f_2, f_4 and f_6.

Table 8
Performance comparison between MG-PSO algorithm and eight ECTs using four CEC 2015-LBP [15] benchmark functions with $d = 30$

Sl. No.	ECTs	Performance measure	f_1	f_2	f_4	f_6
1	**MG-PSO (proposed)**	MFEV	**0.00**	**0.00**	**0.00**	**0.00**
		σ	**0.00**	**0.00**	$\mathbf{5.07\times10^{-56}}$	$\mathbf{3.73\times10^{-44}}$
2	OPSO [24]	MFEV	**0.00**	**0.00**	**0.00**	**0.00**
		σ	5.94×10^{-50}	3.32×10^{-54}	6.11×10^{-44}	3.73×10^{-44}
3	SPS-L-SHADE-EIG [26] Rank #1 CEC 2015-LBP [15]	MFEV	**0.00**	**0.00**	2.00×10^{01}	1.03×10^{01}
		σ	**0.00**	**0.00**	7.29×10^{-05}	1.41×10^{01}
4	DEsPA [27] Rank #2 CEC 2015-LBP [15]	MFEV	**0.00**	**0.00**	2.01×10^{01}	9.71×10^{00}
		σ	**0.00**	**0.00**	4.36×10^{-02}	3.02×10^{00}
5	MVMO [28] Rank #3 CEC 2015-LBP [15]	MFEV	**0.00**	**0.00**	2.00×10^{01}	9.54×10^{00}
		σ	**0.00**	**0.00**	5.42×10^{-04}	3.53×10^{00}
6	LLUDE [31]	MFEV	5.93×10^{-01}	2.84×10^{-14}	2.03×10^{01}	2.59×10^{01}
		σ	2.47×10^{-01}	2.69×10^{-14}	2.33×10^{-02}	3.28×10^{00}
7	SaDE [31]	MFEV	1.78×10^{03}	2.38×10^{-11}	2.05×10^{01}	3.46×10^{01}
		σ	1.43×10^{03}	7.22×10^{-11}	5.99×10^{-02}	6.44×10^{00}
8	JADE [31]	MFEV	6.23×10^{00}	3.41×10^{-14}	2.03×10^{01}	2.61×10^{01}
		σ	1.55×10^{01}	1.17×10^{-14}	2.86×10^{-02}	3.39×10^{00}
9	CoDE [31]	MFEV	1.58×10^{04}	6.02×10^{-13}	2.00×10^{01}	2.97×10^{01}
		σ	1.16×10^{04}	9.88×10^{-13}	9.98×10^{-02}	1.08×10^{01}

The bold numbers indicate the best solution found by corresponding algorithm

4.5.2 *Shifted and rotated benchmark functions: f_1, f_2, f_4 and f_6 with d* = 100

The three top-ranked algorithms in the CEC 2015-LBP [15] are SPS-L-SHADE-EIG [26], DEsPA [27], and MVMO [28], in which the four shifted and rotated benchmark functions, i.e., f_1, f_2, f_4 and f_6, are considered. Here, the performance of the MG-PSO algorithm is compared with the above three top-ranked algorithms with $d = 100$.

The comparison has been achieved with $N_{run} = 20$ and for each run the maximum NFE is 300,000. The criterion used in this comparison depends on the values of maximum NFE and MFEV (21). When the algorithm reaches NFE = 300,000, the MFEV is recorded as a better result. The algorithm obtains a best result when the

MFEV is 0.0 or close to 0.0. The set of parameters given in Table 3 were used in MG-PSO algorithm.

Table 9 presents the results of the MFEV and the corresponding σ obtained by MG-PSO algorithm and three top-ranked algorithms in the CEC 2015-LBP [15]. We can see that the MG-PSO algorithm is significantly superior to SPS-L-SHADE-EIG [26], DEsPA [27] and MVMO [28] in solving f_4 and f_6. While for f_1 and f_2, the MG-PSO algorithm is found similar or better than that of the three top-ranked algorithms.

Table 9
Performance comparison between MG-PSO algorithm and three top-ranked CEC 2015-LBP [15] benchmark functions with $d = 100$

Sl. No.	ECTs	Performance measure	f_1	f_2	f_4	f_6
1	**MG-PSO (proposed)**	MFEV	**0.00**	**0.00**	**0.00**	**0.00**
		σ	**0.00**	**0.00**	**1.03×10^{-53}**	**3.73×10^{-41}**
2	SPS-L-SHADE-EIG [26] Rank #1 CEC 2015-LBP [15]	MFEV	**0.00**	**0.00**	2.00×10^{01}	3.80×10^{01}
		σ	**0.00**	**0.00**	2.25×10^{-02}	1.08×10^{01}
3	DEsPA [27] Rank #2 CEC 2015-LBP [15]	MFEV	4.60×10^{05}	3.07×10^{01}	2.03×10^{01}	4.51×10^{01}
		σ	1.34×10^{05}	1.30×10^{02}	2.12×10^{-01}	7.08×10^{00}
4	MVMO [28] Rank #3 CEC 2015-LBP [15]	MFEV	**0.00**	**0.00**	2.00×10^{01}	1.66×10^{02}
		σ	**0.00**	**0.00**	6.02×10^{-07}	2.12×10^{01}

The bold numbers indicate the best solution found by corresponding algorithm

4.5.3 *Shifted and rotated benchmark functions: f_3, f_5, $f_7 - f_{10}$ with d* = 30

The benchmark functions in CEC 2015 expensive optimization papers (CEC 2015-EOP) [16] are highly competitive and require efficient optimization algorithms to provide fast solutions with a high accuracy. Six shifted and rotated benchmark functions f_3, f_5, $f_7 - f_{10}$ are considered in the two top-ranked algorithms are MVMO [29] and TunedCMAES [30].

Here, we compare the performance of the MG-PSO algorithm with that of the above two top-ranked algorithms and few other competitive algorithms from [21-23], [32]. The comparison has been done with $N_{run} = 20$ and for each run the exact maximum NFE is set at 1,500, as given in [16]. The dimension of each tested function d is 30. In the SHPSO-GSA [21] and DD-SRPSO [22] algorithms 50 particles are used whereas 60 particles are used for the EPSO [23] algorithm. In this experiment, the MG-PSO algorithm uses 20 particles and set of parameters were given in Table 3.

Table 10 shows the MFEV, the corresponding σ and AET for the eight ECTs. Among the eight ECTs, the MG-PSO algorithm achieves the best MFEV performance for the five functions, f_5, f_7-f_{10}, whereas the MVMO [29] gives the best MFEV performance for the function f_3. In terms of σ, the performance of the MG-PSO algorithm is the best in case of the four functions f_5, f_7-f_9 and is the second best for the function f_3 and f_{10}. Thus, the performance of the MG-PSO algorithm is found to be superior to the two CEC 2015-EOP [16] algorithms. In terms of AET, the MG-PSO algorithm performance is better than the OPSO [24] and SabDE [32] for all the six functions, f_3, f_5, f_7 -f_{10}. The AET information for other algorithms is not available.

Table 10

Performance comparison between MG-PSO algorithm and seven ECTs using six CEC 2015-EOP [16] benchmark functions with d = 30

Sl. No.	ECTs	Performance measure	f_3	f_5	f_7	f_8	f_9	f_{10}
1	**MG-PSO (proposed)**	MFEV	1.67×10^{-02}	**1.24×10^{-02}**	**5.00×10^{-03}**	**1.24×10^{-05}**	**2.24×10^{-03}**	**1.12×10^{00}**
		σ	1.58×10^{-02}	**4.22×10^{-03}**	**2.35×10^{-03}**	**1.82×10^{-05}**	**1.83×10^{-03}**	1.07×10^{-02}
		AET (sec)	**1.34×10^{00}**	**1.23×10^{00}**	**1.15×10^{00}**	**1.07×10^{00}**	**1.03×10^{00}**	**1.49×10^{00}**
2	SHPSO-GSA [21]	MFEV	6.96×10^{-01}	1.80×10^{01}	1.06×10^{00}	5.26×10^{-01}	1.36×10^{-01}	2.78×10^{00}
		σ	2.11×10^{01}	9.03×10^{-01}	5.55×10^{-01}	4.62×10^{-01}	3.11×10^{-01}	**1.04×10^{-04}**
		AET (sec)	-	-	-	-	-	-
3	DD-SRPSO [22]	MFEV	2.59×10^{04}	2.24×10^{01}	2.71×10^{00}	5.61×10^{-01}	5.43×10^{-01}	3.32×10^{02}
		σ	1.05×10^{04}	2.12×10^{00}	7.58×10^{-01}	1.00×10^{-01}	2.03×10^{-01}	2.12×10^{02}
		AET (sec)	-	-	-	-	-	-
4	EPSO [23]	MFEV	6.37×10^{04}	3.38×10^{02}	5.04×10^{02}	6.02×10^{02}	7.21×10^{02}	1.27×10^{05}
		σ	-	-	-	-	-	-
		AET (sec)	-	-	-	-	-	-
5	OPSO [24]	MFEV	6.81×10^{-02}	1.37×10^{00}	6.40×10^{-01}	1.49×10^{-02}	1.54×10^{-02}	1.29×10^{00}
		σ	2.96×10^{-02}	3.35×10^{-01}	4.00×10^{-01}	5.97×10^{-03}	7.45×10^{-03}	6.74×10^{-02}
		AET (sec)	2.32×10^{00}	1.98×10^{01}	4.51×10^{00}	1.47×10^{00}	1.43×10^{00}	1.60×10^{00}
6	MVMO [29] Rank #1 CEC 2015-EOP [16]	MFEV	**6.93×10^{-03}**	3.79×10^{01}	1.67×10^{01}	5.20×10^{-01}	4.39×10^{-01}	4.03×10^{02}
		σ	**3.24×10^{-04}**	3.85×10^{0}	5.04×10^{-01}	1.32×10^{-01}	9.93×10^{-02}	2.63×10^{02}
		AET (sec)	-	-	-	-	-	-
7	TunedCMAES [30] Rank #2 CEC 2015-EOP [16]	MFEV	1.17×10^{05}	3.21×10^{02}	5.05×10^{02}	6.00×10^{02}	7.00×10^{02}	8.22×10^{02}
		σ	2.19×10^{04}	5.06×10^{0}	5.91×10^{-01}	2.35×10^{-01}	2.86×10^{-01}	1.09×10^{-01}
		AET (sec)	-	-	-	-	-	-
8	SabDE [32]	MFEV	2.54×10^{09}	2.00×10^{01}	4.33×10^{02}	4.56×10^{04}	7.52×10^{01}	2.39×10^{07}
		σ	5.00×10^{09}	2.04×10^{02}	9.46×10^{01}	4.07×10^{04}	4.15×10^{03}	5.43×10^{07}
		AET (sec)	1.76×10^{01}	1.65×10^{01}	1.68×10^{01}	1.71×10^{01}	1.78×10^{01}	1.72×10^{01}

The bold numbers indicate the best solution found by corresponding algorithm.

4.5.4 *Shifted and rotated benchmark functions: f_3, f_5, $f_7 - f_{10}$ with d* = 100

The benchmark functions in CEC 2015-EOP [16] are highly competitive and require efficient optimization algorithms to provide fast solutions with a high accuracy. Six shifted and rotated benchmark functions f_3, f_5, $f_7 - f_{10}$ are considered. The test has been done with N_{run} = 20 and for each run the exact maximum NFE is set at 1,500, as given in [16]. The MG-PSO algorithm has been run for f_3, f_5, $f_7 - f_{10}$ with d = 100. It uses 20 particles and set of the parameters were given in Table 3. The results are shown in Table 11. It can be seen that the MG-PSO algorithm perform quite satisfactorily. However, as there is no other reported paper, we are unable to compare with other algorithms.

Table 11
Performance of the MG-PSO algorithm for six CEC 2015-EOP [16] benchmark functions with d = 100

Sl. No.	ECTs	Performance measure	f_3	f_5	f_7	f_8	f_9	f_{10}
1	**MG-PSO (proposed)**	MFEV	3.85×10^{-02}	2.79×10^{-02}	7.34×10^{-03}	2.48×10^{-05}	3.36×10^{-03}	2.31×10^{00}
		σ	1.69×10^{-02}	5.45×10^{-03}	4.43×10^{-03}	2.78×10^{-05}	3.43×10^{-03}	2.06×10^{-02}
		AET (sec)	4.48×10^{00}	5.36×10^{00}	4.67×10^{00}	4.34×10^{00}	4.23×10^{00}	5.01×10^{00}

4.6 *Statistical significance of the MG-PSO algorithm for CEC 2015 benchmark functions*

In order to determine the statistical significance of the proposed MG-PSO algorithm, we carried out two sets of unpaired one-tailed t-test [45] with a significance level of α = 0.05. The results of the t-Test for CEC 2015-LBP [15] are shown in Table 12. Here, the statistical results of the comparison between MG-PSO algorithm and eight competitive algorithms for f_1, f_2, f_4 and f_6 are provided. The comparison is made with a degree of freedom equals to 19. The MG-PSO algorithm is considered to be statistically significant against the contender algorithm when t-value < 0 and p-value less than 0.05. The general merit over contender is shown in the last row of Table 12. It is calculated as the difference between the number of times the MG-PSO algorithm is found to be statistically significant and statistically not significant among four tested functions. It can be seen that out of the eight algorithms, the MG-PSO algorithm is statistically significant against four algorithms, i.e., LLUDE, SaDE, JADE and CoDE [31]. However, against the three top-ranked algorithms CEC 2015-LBP [16], SPS-L-SHADE-EIG [26], DEsPA [27] and MVMO [28], the MG-PSO algorithm is statistically significant for f_4 and f_6, whereas it is statistically not significant for f_1 and f_2. In addition, the MG-PSO algorithm is statistically not significant against the OPSO algorithm [24] for f_1, f_2, f_4 and f_6.

Table 13 shows the statistical results of the comparison between MG-PSO algorithm and seven competitive algorithms for the six functions f_3, f_5, $f_7 - f_{10}$ CEC 2015-EOP. The comparison is made with a degree of freedom equals to 19. The general merit over contender is shown in the last row of Table 13. It can be seen that the MG-PSO algorithm is statistically significant against all the seven algorithms. Note that, numbers in bold signatures that MG-PSO algorithm is statistically significant with respect to the corresponding algorithms.

Table 12

Statistical results of unpaired t-test of MG-PSO algorithm against eight ECTs for CEC 2015-LBP [15]

Sl. No.	f	Statistical Results		Competitive Algorithms						
			OPSO [24]	SPS-L-SHADE-EIG [26] Rank #1 CEC 2015-LBP [15]	DEsPA [27] Rank #2 CEC 2015-LBP [15]	MVMO [28] Rank #3 CEC 2015-LBP [15]	JADE [31]	CoDE [31]	LLUDE [31]	SaDE [31]
1	f_1	t-value	0.0	0.0	0.0	0.0	**-1.53×10^{16}**	**-∞**	**-∞**	**-∞**
		p-value	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	**7.99×10^{-297}**	**0.0**	**0.0**	**0.0**
2	f_2	t-value	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		p-value	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}	5.00×10^{-01}
3	f_4	t-value	0.0	**-∞**	**-2.47×10^{16}**	**-∞**	**-2.49×10^{16}**	**-2.47×10^{16}**	**-2.49×10^{16}**	**-2.00×10^{16}**
		p-value	5.00×10^{-01}	**0.0**	**9.05×10^{-301}**	**0.0**	**7.50×10^{-301}**	**9.05×10^{-301}**	**7.50×10^{-301}**	**4.74×10^{-299}**
4	f_6	t-value	0.0	**-2.53×10^{16}**	**-1.19×10^{16}**	**-1.17×10^{16}**	**-1.08×10^{16}**	**-∞**	**-1.06×10^{16}**	**-2.12×10^{16}**
		p-value	5.00×10^{-01}	**5.67×10^{-301}**	**9.13×10^{-295}**	**1.28×10^{-294}**	**7.35×10^{-294}**	**0.0**	**8.51×10^{-294}**	**1.56×10^{-299}**
t = negative t < 0			0	2	2	2	3	3	3	3
t = positive t = ≥ 0			4	2	2	2	1	1	1	1
General Merit Over Contender			4	0	0	0	**2**	**2**	**2**	**2**

Numbers in bold signatures, that MG-PSO algorithm is statistically significant with respect to the corresponding algorithm.

Table 13

Statistical results of unpaired t-test MG-PSO algorithm against seven ECTs for CEC 2015-EOP [16]

Sl. No.	*f*	Statistical Results		Competitive Algorithms					
			SHPSO-GSA [21]	DD-SRPSO [22]	EPSO [23]	OPSO [24]	MVMO [29] Rank #1 CEC 2015-EOP [16]	TunedCMAES [30] Rank #2 CEC 2015-EOP [16]	SabDE [32]
1	f_3	t-value	**-9.41×10^{01}**	**-3.88×10^{06}**	**-9.54×10^{06}**	**-9.89×10^{03}**	9.15×10^{00}	**-1.75×10^{07}**	**-3.80×10^{11}**
		p-value	**7.89×10^{-27}**	**1.63×10^{-114}**	**6.14×10^{-122}**	**3.10×10^{-65}**	2.15×10^{-08}	**5.90×10^{-127}**	**2.37×10^{-209}**
2	f_5	t-value	**-∞**	**-1.37×10^{16}**	**-∞**	**-∞**	**-2.32×10^{16}**	**-∞**	**-∞**
		p-value	**0.0**	**6.05×10^{-296}**	**0.0**	**0.0**	**$2.77\times 10^{--300}$**	**0.0**	**0.0**
3	f_7	t-value	**-∞**	**-2.06×10^{16}**	**-∞**	**-2.51×10^{16}**	**-2.04×10^{16}**	**-∞**	**-∞**
		p-value	**0.0**	**2.14×10^{-301}**	**0.0**	**6.34×10^{-301}**	**3.06×10^{-299}**	**0.0**	**0.0**
4	f_8	t-value	**-2.06×10^{16}**	**-∞**	**-∞**	**-3.74×10^{16}**	**-1.02×10^{16}**	**-∞**	**-∞**
		p-value	**2.63×10^{-299}**	**0.0**	**0.0**	**3.24×10^{-304}**	**1.71×10^{-293}**	**0.0**	**0.0**
5	f_9	t-value	**-1.06×10^{16}**	**-2.13×10^{16}**	**-∞**	**-3.86×10^{16}**	**-3.44×10^{16}**	**-∞**	**-2.30×10^{16}**
		p-value	**7.29×10^{-294}**	**1.44×10^{-299}**	**0.0**	**1.73×10^{-304}**	**1.56×10^{-303}**	**0.0**	**3.22×10^{-300}**
6	f_{10}	t-value	**-1.36×10^{16}**	**-∞**	**-∞**	**-1.26×10^{16}**	**-∞**	**-∞**	**-∞**
		p-value	**6.94×10^{-296}**	**0.0**	**0.0**	**2.86×10^{-295}**	**0.0**	**0.0**	**0.0**
t = negative t < 0			6	6	6	6	5	6	5
t = positive t = ≥ 0			0	0	0	0	1	0	0
General Merit Over Contender			**6**	**6**	**6**	**6**	**4**	**6**	**6**

Numbers in bold signatures, that MG-PSO algorithm is statistically significant with respect to the corresponding algorithm.

5 Case study: Solving economic dispatch problem of South Korea PGS

The economic dispatch of active power of online thermal generating units (TGUs), also termed as ED problem, performs an important part in economic operation of PGS. It is considered as a non-linear constrained optimization problem. The problem becomes multimodal, i.e., non-convex, non-smooth and discontinuous, when the TGUs are subjected to valve-point loading (VPL) effects and a set of prohibited operating zones (POZs) operating power constraints.

In this section, we study the ED problem of large-scale TGUs for South Korea PGS under four practical operating power constraints, i.e., VPL effects, generation limits, ramp rate limits (RRLs) and a set of POZs. The objective function of ED problem is the fuel cost. Explanation of the fuel cost function and its four operating practical power constraints, i.e., VPL effects, generation limits, ramp rate limits (RRLs) and a set of POZs are considered, as follow:

5.1 *Fuel cost function (objective function)*

The aim of ED problem is to guess the optimum arrangement of power generation of all online TGUs in order to minimize the entire generation fuel cost objective function subjected to the online TGUs. In general, the fuel cost function of each on-line TGU is characterized by a quadratic function given by [46]:

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{24}$$

where $F(P_j)$ is the fuel cost function of *j*th TGU in (\$/h), P_j is the output active power in (MW) at current time interval, and a_j, b_j, and c_j are fuel cost coefficients.

5.2 *Operating power constraints imposed by thermal generating units*

Here, four practical operating power constraints imposed on the online TGUs are considered.

5.2.1 *Valve-point loading effects*

Under valve-point loading effects, sinusoidal functions are added to the quadratic cost function (1) [34]. This makes the cost function, non-smooth and non-convex with multiple modes, as follows:

$$F(P_j) = a_j + b_j P_j + c_j P_j^2 + \left| e_j \times sin\left(f_j \times \left(P_{j,min} - P_j\right)\right)\right| \tag{25}$$

where e_j and f_j are the coefficients reflecting VPL effects and $P_{j,min}$ is the minimum output active power of *j*th TGU. The symbol $| \; |$ corresponds to absolute value. The total fuel cost function considering all on-line TGUs is given by:

$$F_{cost} = \sum_{j=1}^{N_{gen}} F(P_j) \tag{26}$$

where N_{gen} is the number of scheduled online TGUs and F_{cost} is the total fuel cost function to be minimized.

An illustrative example: To explain the impact of VPL effects on the fuel cost function, let us suppose two TGUs, TGU$_1$ and TGU$_2$, with a set of parameters as shown in Table

14 [47]. The TGU$_1$ and TGU$_2$ are steam powered turbo generators with multiple valves. Practically, the valves of steam-turbine control the steam entering through separate nozzle groups. Each nozzle group provides best efficiency when its operating at maximum output. Thus, when increasing the output power, the valves of seam-turbine are opened and closed in sequence in order to achieve a highest possible efficiency for a given output power. Subsequently, causing ripple-like effects and then the fuel cost function becomes non-linearity of higher order. Fig. 8 shows the total fuel cost of TGU$_1$ and TGU$_2$ under VPL effects. It can be concluded that multiple local minima are caused by the sinusoidal functions (25).

Table 14
Parameters of TGU$_1$ and TGU$_2$

TGU$_j$	a_j (\$/h)	b_j (\$/MWh)	c_j (\$/MW^{2}h)	e_j (\$/h)	f_j (MW^{-1})	$P_{j,min}$ (MW)
1	958.29	21.60	0.00043	450	0.041	150
2	1,313.60	21.05	0.00063	600	0.036	135

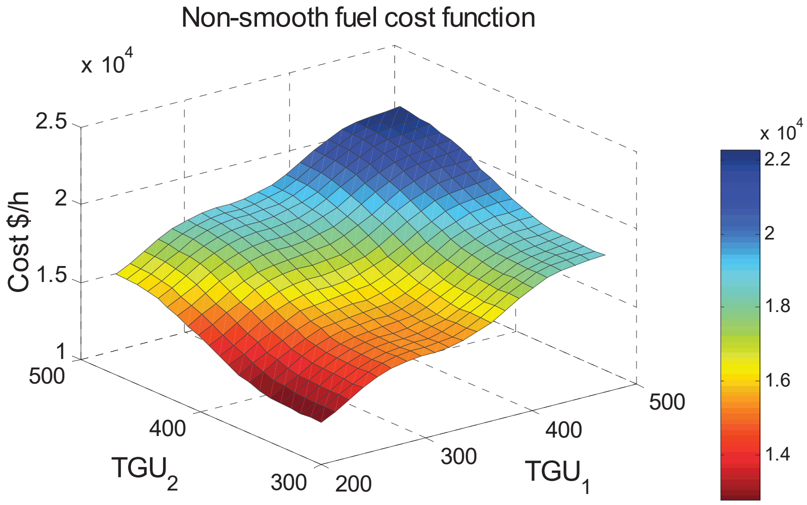


Fig. 8. Total fuel cost function of TGU$_1$ and TGU$_2$ under VPL effects.

5.2.2 *Generation limits*

Each online TGU has a specified range within which its operation is stable. Therefore, it is desired that the TGUs be run within this range in order to maintain system stability. Normally, the generation limits of the *j*th TGU is given by

$$P_{j,min} \leq P_j \leq P_{j,max} \qquad j = 1, 2, ..., N_{gen} \tag{27}$$

5.2.3 *Ramp rate limits*

The operating range of online TGU is restricted by its ramp rate limits, RRLs, due to its physical limitations [48-49]. For any sudden change in the P_D, TGU increase or decrease its power generation in order to satisfy system stability. However, the TGU can change its power generation only at a certain rate determined by its up-ramp and down-ramp rate. If a TGU is operating at a specific point, then its point of operation can be changed only up to a certain rate determined by the ramp rate. Therefore, a change in TGU output active power from one specific interval to the next cannot exceed a specified limit.

If power generation need to increase, then per unit time rate of increase $P_j - P_j^0$ must satisfy:

$$P_j - P_j^0 \leq UR_j \tag{28}$$

If power generation need to decrease, then per unit time rate of decrease $P_j^0 - P_j$ must satisfy:

$$P_j^0 - P_j \leq DR_j \tag{29}$$

where P_j^0 is the TGU output active power at the previous time interval. The UR_j and DR_j are the up-ramp and down-ramp limits of TGU j in (MW/h), respectively.

By substituting (28) and (29) in (27), we obtain the following constraints.

$$max\{P_{j,min}, (P_j^0 - DR_j)\} \leq P_j \leq min\{P_{j,max}, (P_j^0 + UR_j)\} \tag{30}$$

where

$$P_{j,low} = max\{P_{j,min}, (P_j^0 - DR_j)\}, \tag{31}$$

$$P_{j,high} = min\{P_{j,max}, (P_j^0 + UR_j)\}\text{ , and} \tag{32}$$

$P_{j,low}$ and $P_{j,high}$ are the new lower and higher limits of jth TGU, respectively.

5.2.4 *A set of prohibited operating zones*

The physical limitations due to the steam valve operation or vibration in shaft bearing of TGU may result in the generation units operating within prohibited operating zones [49]. The POZs make the fuel cost function discontinuous in nature. In such case, it is difficult to determine the shape of the cost curve of the fuel cost function under a set of POZs through actual performance testing. In addition, if the TGU operates within the POZ range then it may lead to loss of the stability. Therefore, these regions are usually avoided during generation. By using (27), the feasible operating zones (FOZs) of the *j*th TGU are given by:

$$P_{j,min} \le P_j \le P^l_{j,1}$$

$$P^u_{j,k-1} \le P_j \le P^l_{j,k} \qquad k = 2, 3, ..., N_{j,PZ} \tag{33}$$

$$P^u_{j,N_{j,PZ}} \le P_j \le P_{j,max}$$

where $P^l_{j,k}$ and $P^u_{j,k}$ are the lower and upper bound of the *k*th POZs of the *j*th TGU, and $N_{j,PZ}$ is number of POZs of the *j*th TGU. Incorporating these power constraints in (30)-(33), we get the final set of inequality power constraints imposed on TGU as given below.

$$P_{j,low} \le P_j \le P^l_{j,1},$$

$$P^u_{j,k-1} \le P_j \le P^l_{j,k} \qquad k = 2, 3, ..., N_{j,PZ} \tag{34}$$

$$P^u_{j,N_{j,PZ}} \le P_j \le P_{j,high}$$

Equation (34) gives the final set of the operating power constraints imposed on *j*th TGU in terms of new lower and upper generation power limits with RRLs and FOZs and avoiding all POZ$_S$. Thus, all online TGUs will have a set of operation limits (OLs) that satisfies all the operating power constraints.

An illustrative example: In order to illustrate new lower and upper generation limits and FOZs obtained due to presence RRLs and POZs of *j*th TGU, an example of specifications of TGU$_2$ per one-hour generation is given below [46]:
$P_2{}^0$= 170 MW; $P_{2,min}$ = 50 MW; $P_{2,max}$ = 200 MW; UR_2 = 50 MW; DR_2 = 90 MW. The TGU$_2$ has two POZs: POZ$_1$ = [90,110] and POZ$_2$ = [140,160].

From (34), the new lower and upper limits of TGU$_2$ based on RRLs are:

$$P_{2,low} = 80 \text{ MW and } P_{2,high} = 200 \text{ MW},$$

and the three FOZs are:

$$\text{FOZ}_1\text{: } 80 \le P_2 \le 90$$

$$\text{FOZ}_2\text{: } 110 \le P_2 \le 140$$

$$\text{FOZ}_3\text{: } 160 \le P_2 \le 200$$

Fig. 9 shows that TGU_2 has minimum and maximum OLs given by 50 MW and 200 MW, respectively. However, due to presence of RRL power constraint, TGU_2 operates within new lower and higher OLs given by $P_{2,low}$ = 80 MM and $P_{2,high}$ = 200 MW. In addition, three FOZs: FOZ_1 = [80,90] MW, FOZ_2 = [110,160] MW and FOZ_3 = [160,200] MW, and two POZs: POZ_1 = [90,110] MW and POZ_2 = [140,160] MW in dark color are shown in Fig. 9. The intermittent zone ([50,80] MW) is out of OL of the TGU_2.

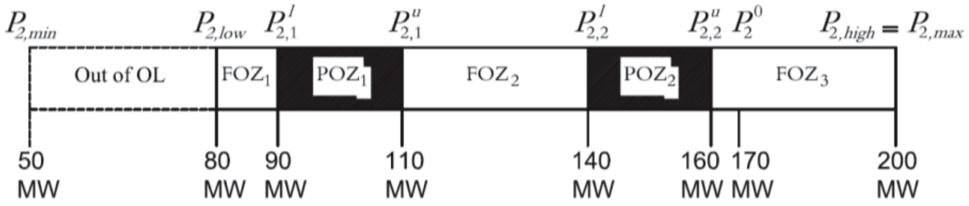


Fig. 9. Lower and upper generation limits, POZs and FOZs for TGU_2.

5.3 *Performance measures*

To study the accuracy, consistency and robustness of different algorithms in solving ED problem of South Korea PGS, several fitness values as illustrated below are considered. Every algorithm is executed over N_{run} runs each with N_{iter} iterations.

1. Ensemble average cost (F_{cost}): At each iteration, it is the average cost value obtained from N_{run} independent runs.
2. Minimum fuel cost (F_{min}): Defined as the minimum of the optimized F_{cost} values obtained from N_{run} independent runs.
3. Maximum fuel cost (F_{max}): Defined as the maximum of the optimized F_{cost} values obtained from N_{run} independent runs.
4. Mean fuel cost (F_{mean}): Defined as the average of the optimized F_{cost} values obtained from N_{run} independent runs.
5. Standard deviation (σ): The σ is the standard deviation of the optimized F_{cost} values obtained from N_{run} independent runs.
6. Range (R): The range (R) is defined as the difference between F_{max} and F_{min}.
7. Average execution time (AET): It is the time consumed by an algorithm after convergence, averaged over N_{run} independent runs.

5.4 *Test case study: South Korea power generation system*

The South Korea power generation system is a very large-scale PGS [34]. It consists of gas, stream, diesel, and nuclear power stations. The South Korea PGS is a complex with 140 TGUs each having generation limits and RRLs. In addition, the cost fuel functions of 12 TGUs have VPL effects and 4 TGUs have 11 POZs. The maximum load demand under steady-state and normal operations is 49,342 MW. The South Korea PGS data are available in [34].

5.4.1 *Comparison in terms of fitness values*

The The South Korea PGS has been tested with three existing ECTs, i.e., CCPSO [33], C-GRASP-SaDE [34], and C-GRASP-MDE [34]. Here, the performance of MG-PSO algorithm is compared with these three algorithms as well as GPSO-*w*. The set of parameters used in MG-PSO and GPSO-*w* algorithms are shown in Table 15. In addition, both are run with m = 20, d = 140 and N_{run} = 25. In MG-PSO algorithm, four negative gradients were selected (N_{grad} = 4) by trial and error method, three for *Exploration* phase and another one for *Exploitation* phase as shown in Table 15.

The fitness values of the five ECTs are listed in Table 16. It can be seen that in GPSO-*w*, F_{mean} = \$2,529,855.79/h and σ = \$358,126.35/h. These results indicate that GPSO-*w* is unable to solve South Korea PGS. Whereas, the MG-PSO algorithm is efficient in obtaining the best result in terms of F_{mean} over 25 independent runs. In addition, in terms of σ, the performance of the MG-PSO algorithm is the second best. This shows that the MG-PSO algorithm provides optimum and consistent results. In addition, the range *R* of MG-PSO algorithm is the second lowest among the five ECTs, thus indicating that it provides solution with low dispersion. In terms of AET, the MG-PSO algorithm shows the second best performance. These results indicate that among the five ECTs, the MG-PSO algorithm is stable and robust and is able to provide optimum solution.

Table 15

Set of parameters used in MG-PSO and GPSO-*w* algorithms for South Korea PGS

Set of parameters	MG-PSO				GPSO-*w*
	Exploration phase			*Exploitation* phase	
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	
γ	0.40	0.40	0.40	0.40	-
c_1, c_2	2.05	2.05	2.05	2.05	2.00
N_{iter}	1,200	1,200	1,200	1,800	3000
$w_{ini,k}$	0.80	0.80	0.80	0.35	0.90
$w_{fin,k}$	0.10	0.20	0.30	0.20	0.40
$grad_k$	-5.83×10^{-4}	-5.00×10^{-4}	-4.16×10^{-4}	-8.33×10^{-5}	-1.67×10^{-04}

Table 16

Comparison of cost performance between MG-PSO algorithm and other 3 ECTs for South Korea PGS

Sl.No.	Algorithm	Min. Cost F_{min} ($/h)	Max. Cost F_{max} ($/h)	Mean Cost F_{mean} ($/h)	σ ($/h)	R ($/h)	AET (sec)
1	CCPSO [33]	1,657,962.7300	1,657,962.7300	1,657,962.7300	**0.00**	**0.00**	150.00
2	C-GRASP-SaDE [34]	1,657,962.7268	1,658,583.5267	1,658,006.2712	NA	620.79	NA
3	C-GRASP-MDE [34]	1,666,667.7400	1,897,207.1500	1,,685,973.32	NA	230,539.41	NA
4	GPSO-*w*	1,933,419.8873	3,366,473.6288	2,529,855.7978	358,126.35	1,433,053.74	**31.29**
5	MG-PSO	1,656,515.4715	1,656,917.3113	**1,656,667.4650**	8.01	401.83	48.37

The bold numbers indicate the best solution found by corresponding algorithm.

5.4.2 *Convergence characteristics of MG-PSO and GPSO-w algorithms*

Fig. 10 shows the convergence characteristics of MG-PSO and GPSO-*w* algorithms for South Korea PGS. It shows ensemble average F_{cost} values at each iteration obtained from 25 independent runs. It can be seen that MG-PSO algorithm settles at about 1,300 iterations and achieves F_{mean} of about $1,656,700/h. Whereas, the GPSO-*w* algorithm settles at a non-optimum F_{mean} of about $2,529,855.79/h since the beginning of its learning. Early convergence of the GPSO-*w* algorithm indicates that it has trapped into a local minimum of at about $2,529,855.79/h. This indicates that the GPSO-*w* algorithm is unable to solve ED problem of South Korea PGS with such a high dimension (d = 140) and under such a large number of operating power constraints. Whereas, it is clear that for this complex PGS, the MG-PSO algorithm efficiently converges to the vicinity of the optimum solution with four operating power constraints imposed by online TGUs.

Fig. 11 shows the variation of optimized F_{cost} over 25 independent runs achieved by the MG-PSO and GPSO-*w* algorithms. It shows that the optimized F_{cost} of MG-PSO algorithm varies between $1,656,515/h and $1,656,917/h, whereas in GPSO-*w* algorithm, it varies between $1,933,419.88/h and $3,366,473.62/h. This indicates that MG-PSO algorithm is capable of providing consistent and reliable optimum solution.

Whereas, the GPSO-*w* algorithm is unable to provide optimum solution due to the high complexity of the problem.

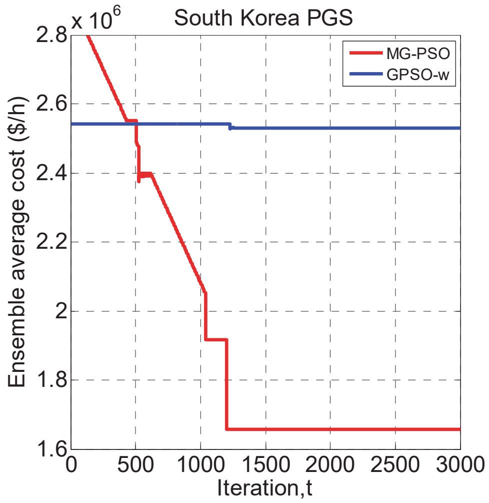


Fig. 10. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for South Korea PGS.

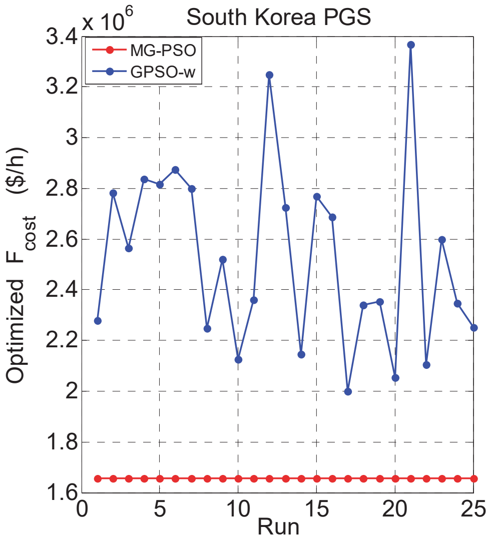


Fig. 11. Convergence characteristics of MG-PSO and GPSO-*w* algorithms for South Korea PGS.

5.4.3 *Comparison in terms of four operating power constraints*

Tables 17 presents a solution vector P_j (j = 1, 2, ..., 140) corresponding to the best solution obtained from MG-PSO and GPSO-*w* algorithms. Four practical operating power constraints imposed on 140 TGUs in South Korea PGS, i.e., VPL effects, RRLs, generation limits and a set of POZs, are considered. In case of GPSO-*w* algorithm, it is unable to solve non-smooth, non-convex cost function due to VPL effects, as shown from the values of fuel cost (minimum, maximum and mean) in Table 16. In addition, 11 TGUs violate RRLs and generation limits, as shown in red color in Table 17. The details of the 11 TGUs for which GPSO-*w* algorithm failed to satisfy RRLs and generation limits are shown in Table 18. These 11 TGUs must operate within the range of RRLs and generation limits, based on (34).

In addition, TGU #136 violates POZ [50-74] MW, based on (34), as shown in blue color in Table 17. This means that GPSO-*w* algorithm fails to solve ED problem for South Korea PGS and is unable to solve the four operating power constraints imposed on a large-scale TGUs. However, the MG-PSO algorithm avoids the 11 POZs imposed on 4 TGUs and working within RRLs and generation limits of each TGU and solving non-smooth cost function due to VPL effects imposed on 12 TGUs.

Table 17

Optimized output active power in (MW) for each TGU obtained by GPSO-*w* and MG-PSO algorithms for South Korea PGS

GPSO-*w*								MG-PSO							
TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j	TGU$_j$	P_j
1	119.0	39	611.1	77	513.1	115	368.2	1	118.9029	39	773.9550	77	389.9391	115	244.0759
2	150.8	40	707.6	78	336.4	116	286.4	2	163.9317	40	768.9389	78	330.8755	116	244.3029
3	167.7	41	4.7	79	196.7	117	348.2	3	189.8356	41	3.5417	79	530.7148	117	244.0177
4	137.8	42	12.1	80	272.4	118	160.9	4	189.8119	42	3.1421	80	530.9548	118	95.1031
5	183.4	43	170.4	81	525.4	119	119.8	5	189.8391	43	249.7563	81	541.6917	119	95.2248
6	129.3	44	185.4	82	90.8	120	189.3	6	189.9790	44	249.7986	82	56.1218	120	116.1326
7	320.7	45	222.9	83	156.1	121	300.7	7	489.9250	45	249.9315	83	115.4416	121	175.2108
8	285.0	46	197.2	84	236.1	122	11.0	8	489.9932	46	249.7574	84	115.3084	122	2.0596
9	392.6	47	178.2	85	241.9	123	17.6	9	495.8361	47	249.9664	85	115.4370	123	4.1350
10	266.4	48	191.0	86	239.3	124	63.7	10	495.9449	48	249.8982	86	207.4802	124	15.1293
11	441.0	49	224.3	87	211.7	125	16.2	11	495.8284	49	249.8841	87	207.0506	125	9.3245
12	329.6	50	234.8	88	295.0	126	20.6	12	495.8931	50	249.8927	88	175.2168	126	12.3679
13	351.5	51	365.2	89	193.6	127	13.3	13	505.9485	51	165.0090	89	175.2709	127	10.0243
14	467.3	52	484.4	90	239.3	128	230.0	14	508.9545	52	165.0718	90	175.3078	128	112.1335
15	424.4	53	383.5	91	324.6	129	19.0	15	505.9317	53	165.2701	91	175.3082	129	4.0748
16	377.6	54	457.9	**92**	**516.7**	130	34.9	16	504.8797	54	165.3485	92	575.3236	130	5.3571
17	309.7	55	269.9	**93**	**509.8**	131	6.2	17	505.9219	55	180.0918	93	547.2814	131	5.3220
18	360.9	56	380.7	**94**	**983.9**	132	86.9	18	505.9252	56	180.2213	94	836.2569	132	50.0785
19	369.5	57	278.5	95	813.1	133	6.8	19	504.9787	57	103.0974	95	837.2834	133	5.1698
20	348.6	58	521.0	96	634.2	134	71.9	20	504.8703	58	198.6536	96	681.8989	134	42.1825
21	263.6	59	230.9	97	712.4	135	65.6	21	504.9936	59	311.8011	97	719.9877	135	42.3435
22	352.5	60	216.3	98	683.5	**136**	**51.2**	22	504.9871	60	311.6648	98	717.6493	136	41.2530
23	470.7	61	425.2	99	705.5	137	23.1	23	504.8732	61	163.7961	99	719.6865	137	17.2795
24	272.5	62	298.6	100	891.5	138	17.8	24	504.8014	62	95.8232	100	963.8141	138	7.0127
25	505.1	63	478.2	101	945.7	139	7.6	25	536.8458	63	510.9918	101	957.9583	139	7.0810
26	507.6	64	470.4	102	862.4	140	34.6	26	536.9566	64	510.9026	102	946.5203	140	26.1599
27	481.4	65	205.1	**103**	**976.1**	FAIL		27	548.9488	65	489.6328	103	933.7727	SUCCESS	
28	420.2	66	458.6	**104**	**984.6**			28	548.9923	66	252.6819	104	934.8718		
29	324.9	67	413.8	**105**	**1,014.3**			29	500.9955	67	489.7562	105	876.4437		
30	463.6	68	293.6	**106**	**906.0**			30	498.9895	68	489.6741	106	880.1277		
31	430.2	69	175.3	**107**	**938.4**			31	505.9392	69	130.8168	107	873.4553		
32	285.0	70	297.0	108	874.2			32	505.8717	70	296.6273	108	877.2801		
33	473.8	71	337.9	**109**	**879.0**			33	505.9013	71	142.7573	109	871.2103		
34	455.5	72	411.5	**110**	**919.5**			34	505.8016	72	367.6518	110	864.6802		
35	461.1	73	345.3	**111**	**969.3**			35	499.9410	73	195.6819	111	882.3793		
36	293.9	74	473.3	112	165.3			36	499.8281	74	219.6960	112	94.3471		
37	164.8	75	483.2	113	193.4			37	240.9999	75	217.8117	113	94.2248		
38	216.7	76	523.9	114	113.6			38	240.9666	76	267.8831	114	94.2292		

Table 18

List of 11 TGUs that violate RRLs and generation limits based on output active power P_j obtained by GPSO-*w* for South Korea PGS

TGU$_j$	92	93	94	103	104	105	106	107	109	110	111
$P_{j,low}$ (MW)	539.4	511.5	795.0	844.0	875.0	816.5	820.9	813.7	799.5	795.0	810.0
$P_{j,high}$ (MW)	575.4	547.5	836.8	934.0	935.0	876.5	880.9	873.7	871.7	864.8	882.0
P_j (MW)	516.7	509.8	983.9	976.1	984.6	1,014.3	906.0	938.4	879.0	919.5	969.3

5.5 *Statistical significance of the MG-PSO algorithm for South Korea PGS*

Here, the statistical significance of the proposed MG-PSO algorithm has been determined, one set of unpaired one-tailed t-Test is carried out [45] with a significance level of α = 0.05. The MG-PSO algorithm is considered to be statistically significant against the contender algorithm when t-value < 0 and p-value less than 0.05. The general merit over contender is shown in the last row of Table 19.

Table 19 shows the t-Test results for South Korea PGS. The one-tailed unpaired (α = 0.05 with a degree of freedom of 24) is performed against four competitive algorithms. As seen from the data in the last column in Table 19, the proposed MG-PSO algorithm is found to be statistically significant against the GPSO-*w* algorithm for South Korea PGS. One can see from the statistical data in the last column of Table 19 that the MG-PSO algorithm is statistically significant against the four contending algorithms for solving ED problem of the South Korea PGS. These results give enough evidence that the proposed MG-PSO algorithm is statistically significant against all four contending algorithms.

Table 19

Statistical results of unpaired t-Test of MG-PSO algorithm against four ECTs for South Korea PGS

Sl. No.	Competitive Algorithms	South Korea PGS		t = negative t < 0	t = positive t ≥ 0	General Merit Over Contender
		t-value	p-value			
1	GPSO-*w*	**-4.02×10^{04}**	**1.84×10^{-95}**	1	0	**1**
2	CCPSO [33]	**-5.96×10^{01}**	**1.34×10^{-27}**	1	0	**1**
3	C-GRASP-SaDE [34]	**-6.16×10^{01}**	**6.04×10^{-28}**	1	0	**1**
4	C-GRASP-MDE [34]	**-6.05×10^{01}**	**6.13×10^{-28}**	1	0	**1**

Numbers in bold signatures, that MG-PSO algorithm is statistically significant with respect to the corresponding algorithm.

6 Conclusion

An algorithm called multi-gradient PSO (MG-PSO) algorithm is proposed and applied to optimize unimodal and multimodal problems. In MG-PSO algorithm, several negative gradients are used by *m* particles while searching for an optimum solution in two phases called *Exploration* and *Exploitation*. The combination of two phases provides a balance between global and local search spaces. Thus, this combination is successfully applied to overcome the disadvantages of the gradient methods. With extensive simulation studies, performance of the MG-PSO algorithm was compared with GPSO-*w* algorithm and several existing competitive algorithms and its superiority is demonstrated in terms of several performance measures. The MG-PSO algorithm was applied to ten selected unimodal and multimodal benchmark functions (CEC 2015) with $d = 30$ and $d = 100$ as well as to South Korea power generating system (case study). The proposed MG-PSO algorithm outperformed several existing ECTs including top-ranked CEC 2015 algorithms. In addition, the sensitivity analysis and statistical tests were carried out to demonstrate the effectiveness of the proposed algorithm. Thus, the MG-PSO algorithm proved to be a powerful and highly effective algorithm that is capable of solving complex unimodal and multimodal functions.

Appendix

Here, pseudocode of the MG-PSO algorithm is provided.

```
Begin MG-PSO Algorithm
  Let $f(x)$ be the function to be minimized.
  Choose $N_{iter}$, $N_{grad}$, $w_{ini,k}$, $w_{fin,k}$, $k = 1, 2, \ldots, N_{grad}$
  Determine $N_{iter,xplore}$ and $N_{iter,xploit}$ using (10) and (11), respectively.
Step 1: Initialization: Iteration, $t = 0$
  Obtain $G_{best}(0)$ using (1)-(4)
Step 2: Begin Exploration phase
  for $k = 1, 2, ..., N_{grad} - 1$
  begin of episode $k$
  Determine $grad_k$ using (12)
    for $t = 1, 2, \ldots, N_{iter,xplore}$
    Determine $w_k(t)$ using (15)
      for $i = 1, 2, \ldots, m$
      Update the particle's velocity and position vectors as follows
      $V_i^k(t) = w_k(t) V_i^k(t-1) + c_1 r_1(t)[G_{pers,i}^k(t-1) - X_i^k(t-1)] + c_2 r_2(t)[G_{best}^k(t-1) - X_i^k(t-1)]$   (A-1)
      $X_i^k(t) = X_i^k(t-1) + V_i^k(t)$   (A-2)
      Evaluate the particle's performance by substituting (A-2) in $f(x)$
      Update $G_{pers,i}$ as follows
      $G_{pers,i}^k(t) = \begin{cases} X_i^k(t) & if \quad f(X_i^k(t)) \le f(G_{pers,i}^k(t-1)) \\ G_{pers,i}^k(t-1) & Otherwise \end{cases}$   (A-3)
      end $i$ loop
      for $i = 1, 2, ..., m$
      Obtain $f(G_{pers,i}^k(t))$
      end loop $i$
    $f(G_{best}^k(t)) = min\{\, f(G_{pers,i}^k(t)) \,\}$   (A-4)
    Obtain $G_{best}^k(t)$ corresponding to $f(G_{best}^k(t))$
    end $t$ loop
  Obtain $G_{best}^k(N_{iter,xplore})$ and $f(G_{best}^k(N_{iter,xplore}))$
  end of episode $k$
  end $k$ loop
  for $k = 1, 2, \ldots, N_{grad}$-1
  Obtain $f(G_{best}^k(N_{iter,xplore}))$
  end $k$ loop
```

```
    $f(G_{best,xplore}) = min\{ f(G^{k}_{best}(N_{iter,xplore})) \}$
(A-5)
    Obtain $BEST(G_{best,xplore})$ corresponding to $f(G_{best,xplore})$
    Obtain new search space (neighbourhood) by taking “Floor” and “Ceil” of each element of
    $BEST(G_{best,xplore})$
End Exploration phase


Begin Exploitation phase
Use the new search space
Step 3: Initialization: Iteration, $t = 1$
    for $i = 1 , 2, \ldots, m$
    $V_i(1) = V_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$    (A-6)
    $X_i(1) = X_i(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$    (A-7)
    $G_{pers,i}(1) = G_{pers,i}(N_{iter,xplore})$ corresponding to $BEST(G_{best,xplore})$    (A-8)
    end $i$ loop
    $G_{best,xploit}(1) = BEST(G_{best,xplore})$    (A-9)
    Determine $grad_{N_{grad}}$ using (13)
Step 4: Update
    for $t = 2, 3, \ldots, N_{iter,xploit}$
    Determine $w_k(t)$ using (15)
        for $i = 1, 2, \ldots, m$
        Update the particle’s velocity and position vectors as follows:
        $V_i(t) = w_{N_{grad}}(t)\ V_i(t-1) + c_1 r_1(t)\ [G_{pers,i}(t-1) - X_i(\mathrm{t}-1)]$
                        $+ c_2 r_2(t)\ [G_{best,xploit}(t-1) - X_i(\mathrm{t}-1)]$    (A-10)
        $X_i(\mathrm{t}) = X_i(t-1) + V_i(t)$    (A-11)
        Evaluate the particle’s performance by substituting (A-11) in $f(x)$
        Update $G_{pers,i}(t)$ as follows
        $G_{pers,i}(t) = \begin{cases} X_i(t) & if\ f(X_i(t)) \le f(G_{pers,i}(t-1)) \\ G_{pers,i}(t-1) & Otherwise \end{cases}$    (A-12)
        end $i$ loop
        for $i = 1, 2, ..., m$
        Obtain $f(G_{pers,i}(t))$
        end loop $i$
    $f(G_{best,xolit}(t)) = min\{ f(G_{pers,i}(t))\}$    (A-13)
    Obtain $G_{best,xploit}(t)$ corresponding to $f(G_{best,xploit}(t))$
    end $t$ loop
    Optimum solution = $G_{best,xploit}(N_{iter,xploit})$
    Optimum value = $f(G_{best,xploit}(N_{iter,xploit}))$
End of Exploitation phase
```

References

[1] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks; 1995 (4), pp. 1942-1948.

[2] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the sixth International Symposium on Micro Machine and Human Science; 1995, p. 39-43.

[3] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation; 1999, pp. 1945-50.

[4] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transaction on Evolutionary Computation 6 (1) 2002 658-73.

[5] S.S. Sathya, M.V. Radhika, Convergence of nomadic genetic algorithm on benchmark mathematical functions, Applied Soft Computing 13 (5) (2013) 2759-2766.

[6] Y. Liang, K.S. Leung, Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization, Applied soft computing 11 (2) (2011) 2017-2034.

[7] Z. Ma, Chaotic population in genetic algorithm, Applied soft computing 12 (8) (2012) 2409-2424.

[8] C.M Fu, C. Jaing, C.G Chen, Q.M. Lin, An adaptive differential evolution algorithm with an aging leader and challenges mechanism, Applied Soft computing 57 (2017) 60-73.

[9] Y. Cai, C. Sun, T. Wang, H. Tian, Y. Chen, J. Wang, Neighbourhood-adaptive differential evolution for global numerical optimization, Applied Soft Computing 59 (2017) 659-706.

[10] S.S. Jadan, R. Tiwari, H. Sharma, J. Bansal, Hybrid artificial bee colony algorithm with differential evolution, Applied Soft Computing 58 (2017) 11-24.

[11] N. Lynn, P. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm and Evolutionary Computation 24 (2015) 11-24.

[12] M. Islam, X.D Li, Y. Mei, A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO, Applied Soft Computing, 59 (2017) 182-196.

[13] Y.H, Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation; 1998, pp. 69-73.

[14] Y Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation; 1999, pp. 1945-50.

[15] J.J. Liang, B.Y. Qu, P.N. Suganthan, Q. Chen, Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization, Zhengzhou University/Nanyang Technological University, Zhengzhou, China/Singapore, 2014, Technical Report.

[16] Q. Chen, B. Liu, Q. Zhang, J.J Liang, P.N Suganthan, B.Y. Qu, Problem definition and evaluation criteria for CEC 2015 special session and competition on bound constrained

single-objective computationally expensive numerical optimization, in: IEEE Conference Evolutionary Computation, Sendai, Japan, 2014, Special session.

[17] Z. Zhan, J. Zhang, Y. Li, Y. Shi, Orthogonal learning particle swarm optimization, IEEE Transactions on Evolutionary Computation 6 (2011) 832-47.

[18] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based modified search equation and orthogonal learning. IEEE Transactions on Cybernetics 43 (3) (2013) 1011-24.

[19] Y. Leung, Y. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions on Evolutionary Computation 5 (1) (2011) 41-53.

[20] P. Liu, J. Liu, Multi-leader (MLPSO): a new PSO variants for solving global optimization, Applied Soft Computing 61 (2017) 256-263.

[21] A. Yadav, K. Deep, J. Kim , A. Nagar, Gravitational swarm optimizer for global optimization. Swarm and Evolutionary Computation 31 (2016) 64-89.

[22] M. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self- regulating particle swarm optimization algorithm, Swarm and Evolutionary Computation 28 (2016) 98-116.

[23] T. Ngo, A. Sadollah, J. Kim, A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems, Journal of Computational Science 13 (2016) 68-82.

[24] L.T. Al-Bahrani, J.C. Patra, Orthogonal particle swarm optimization for economic dispatch of thermal generating units under various power grid constraints in smart power grid, Applied Soft Computing 58 (2017) 401-26.

[25] L.T. Al-Bahrani, J.C. Patra, A novel orthogonal PSO algorithm based on orthogonal diagonalization, Swarm and Evolutionary Computation. In press, corrected proof.

[26] S.M. Guo, H. Tsai, C.C. Yang, P.H. Hsu, A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1003-1010.

[27] N. Awad, M.Z. Ali, R.G. Reynolds, A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1098-05.

[28] J.L. Rueda, I Erlich, Testing MVMO on learning-based real-parameter single objective benchmark optimization problems, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1025-32.

[29] J.L. Rueda, I Erlich, MVMO for bound constrained single-objective computationally expensive numerical optimization, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1011-17.

[30] V. Berthier, Experiments on the CEC 2015 expensive optimization testbed, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1059-66.

[31] X.-G. Zhou, G.-J. Zhan, X.-H. Hao, D.-W. Xu, L. Yu, Enhance differential evolution using local Lipschits underestimate strategy for computationally expensive optimization problems, Applied Soft Computing 48 (2016) 169-81.

[32] A. Banitalebi, M. Aziz, Z. Aziz, A self-adaptive binary differential evolution algorithm for large scale binary optimization problems, Information Sciences 367-368 (2016) 487-511.

[33] J. Park, Y Jeong, J Shin, An improved particle swarm optimization for nonconvex economic dispatch problems, IEEE Transactions on Power Systems 25 (1) (2010) 156-66.

[34] J Neto, G. Reynoso-Meza, T. Ruppel, V. Mariani, L. Coelho, Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution. International Journal of Electrical Power and Energy Systems 84 (2017) 13-24.

[35] M.M. Noel, T.C. Jannett, Simulation of a new hybrid particle swarm optimization algorithm, in Proceedings of the 36th Southeastern Symposium on System Theory, 2004, pp. 150-53.

[36] R. Azizipanah-Abarghooee, T. Niknam, M. Gharibzadeh, F. Golestaneh, Robust, fast and optimal solution of practical economic dispatch by a new enhanced gradient-based simplified swarm optimisation algorithm, IET Generation, Transmission & Distribution 7 (6) (2013) 620-35.

[37] R. Azizipanah-Abarghooee, P. Dehghaniab, V. Terzija, Practical multi-area bi-objective envirronmental economic dispatch equipped with a hybrid gradient search method and improved Jaya algorithm, IET Generation, transmission & distribution 10 (14) (2016) 3580-96.

[38] L.T Al-Bahrani, J.C Patra, R Kowalczyk, Multi-gradient PSO algorithm for economic dispatch of thermal generating units in smart grid, in: Proceedings of IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia) Melbourne, Australia, 2016, pp. 258-63.

[39] L.T Al-Bahrani, J.C Patra, Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid, Energy. In press, accepted manuscript.

[40] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization an overview, Swarm Intelligence (1) (2007) 33-57.

[41] Y. Tuppadung, W. Kurutach, Comparing non-linear inertia weights and construction factors in particle swarm optimization, International Journal of Knowledge-based and Intelligent Engineering Systems 15 (2011) 65-70.

[42] J. Gou, Y-X. Lei, W-P. Guo, C. Wang, Y-Q. Cai, W Luo, A novel improved particle swarm optimization algorithm based on individual difference evolution, Applied Soft Computing 57 (2017) 468-481.

[43] A.R. Jordehi, Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems, Applied Soft Computing 26 (2015) 401-417.

[44] J. Li, Y. Tan, The bare bones fireworks algorithm: A minimalist global optimizer, Applied Soft Computing 62 (2018) 454-462.

[45] D.J. Sheskin, Handbook of parametric and nonparametric statistical procedures, Western Connecticut State University, USA, Champman and Hall/CRC 2000.

[46] J Sun, V. Palade, W. Wu X, Fang, Z. Wang, Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization, IEEE Transactions on Power Systems 13 (1) (2014) 519-26.

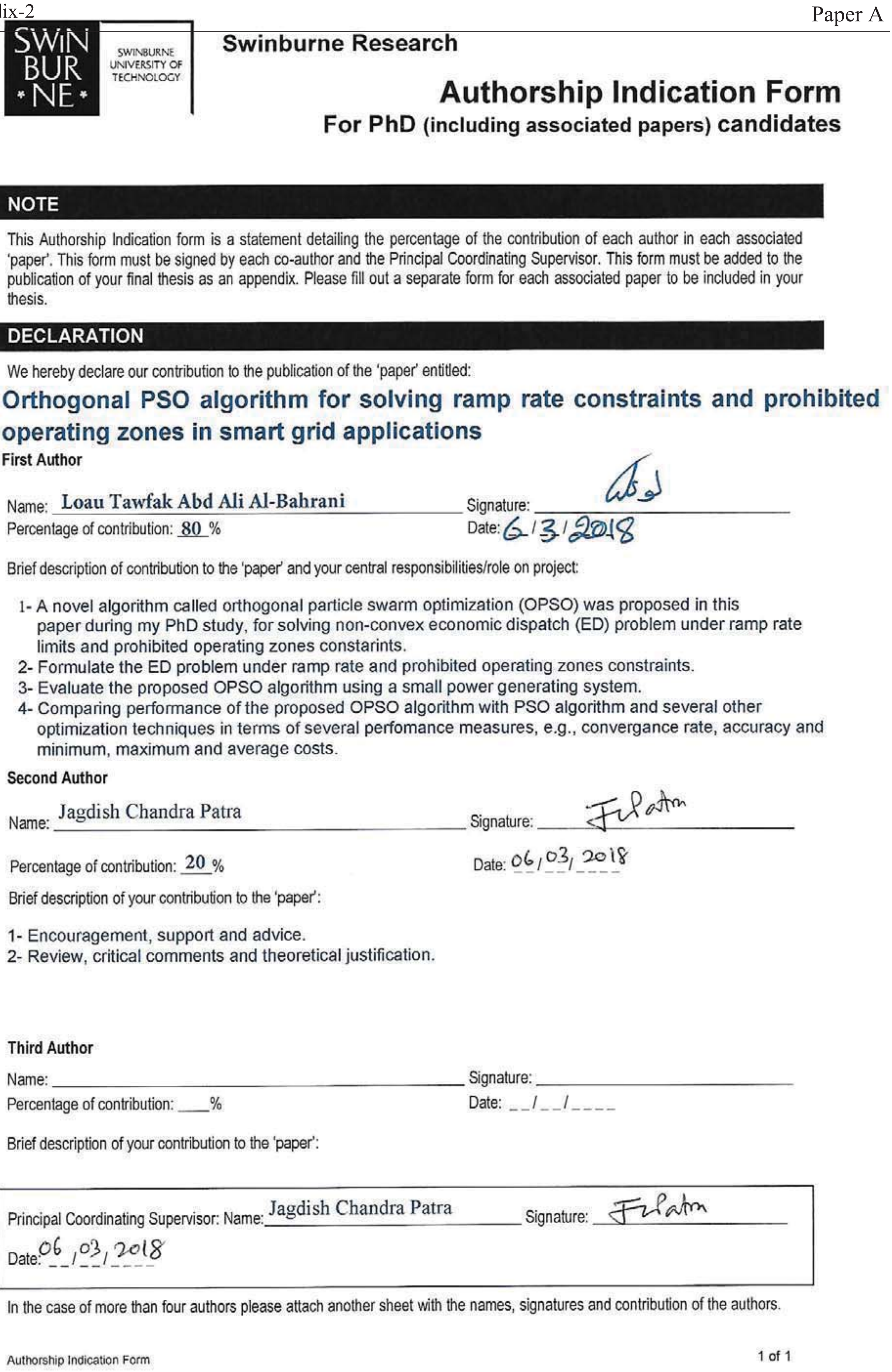
[47] A. Meng, H. Hanwu, H Yin, G. Zhuanghi, Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects, Energy 93 (2015) 2175-90.

[48] Niknam T, Mojarrad H, Meymand H, Firouzi B. A new honey bee mating optimization algorithm for non-smooth economic dispatch. Energy 2011;896-08.

[49] Mohammadi B, Rabiee B, Soroudi A, Ehsan M. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. Energy 2012;44:228-34.

Appendix-2: Authorship Indication Form

Here, the signed authorship indication form for the Papers A-H are presented as follows:



Swinburne Research

SWINBURNE UNIVERSITY OF TECHNOLOGY

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80 % Date: 6/3/2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called orthogonal particle swarm optimization (OPSO) was proposed in this paper during my PhD study, for solving non-convex economic dispatch (ED) problem under ramp rate limits and prohibited operating zones constarints.
2- Formulate the ED problem under ramp rate and prohibited operating zones constraints.
3- Evaluate the proposed OPSO algorithm using a small power generating system.
4- Comparing performance of the proposed OPSO algorithm with PSO algorithm and several other optimization techniques in terms of several perfomance measures, e.g., convergance rate, accuracy and minimum, maximum and average costs.

Second Author

Name: Jagdish Chandra Patra Signature:

Percentage of contribution: 20 % Date: 06/03/2018

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

Name: ______ Signature: ______

Percentage of contribution: ___% Date: __/__/____

Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:

Date: 06/03/2018

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1

SWINBURNE UNIVERSITY OF TECHNOLOGY

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Orthogonal PSO algorithm for economic dispatch of power under power grid constraints

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80 % Date: 6/3/2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called orthogonal particle swarm optimization (OPSO) was proposed to solve economic dispatch (ED) of power of 15 thermal generating units (TGUs) power generating system under several operating power constraints.
2- Formulate the ED problem under several operating power constraints.
3- Evaluate the OPSO and global PSO (GPSO) algorithm using 15 TGUs power generating system.
4- Comparing performance of the OPSO algorithm with GPSO algorithm and several other optimization techniques in terms of several performance measures, e.g., conergence rate, accuracy, equality and inequality power constraints and minimum, maximum and mean costs.

Second Author

Name: Jagdish Chandra Patra Signature:

Percentage of contribution: 20 % Date: 06/03/2018

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

Name: ______ Signature: ______

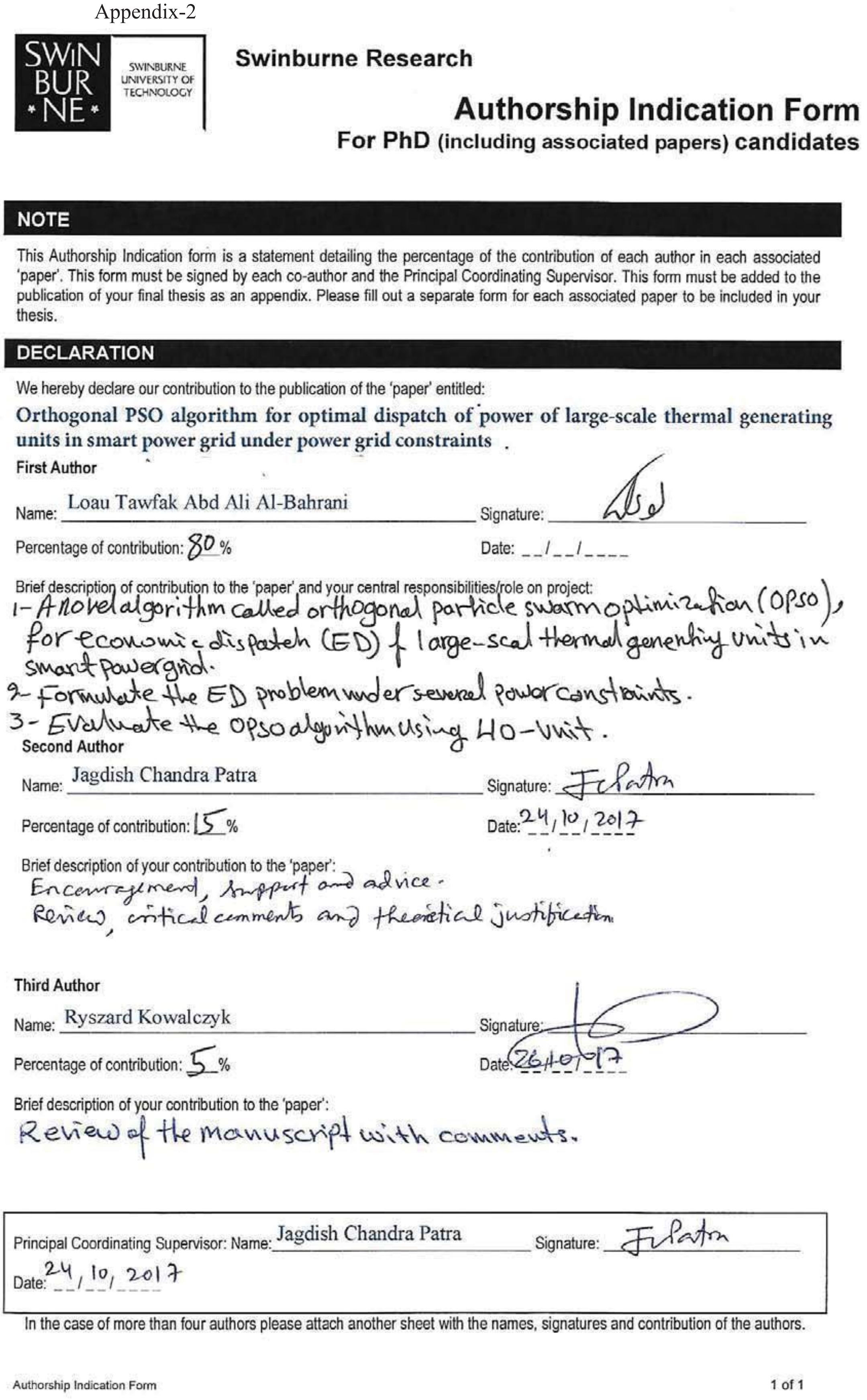
Percentage of contribution: ___% Date: __/__/____

Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:

Date: 06, MAR 2018

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.



Appendix-2

SWINBURNE UNIVERSITY OF TECHNOLOGY

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints .

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80 % Date: __/__/____

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called orthogonal particle swarm optimization (OPSO), for economic dispatch (ED) f large-scal thermal generting units in smart powergrid.

2- Formulate the ED problem under several power constraints.

3- Evaluate the OPSO algorithm using 40-unit.

Second Author

Name: Jagdish Chandra Patra Signature:

Percentage of contribution: 15 % Date: 24/10/2017

Brief description of your contribution to the 'paper':

Encouragement, support and advice.

Review, critical comments and theoretical justification

Third Author

Name: Ryszard Kowalczyk Signature:

Percentage of contribution: 5 % Date: 26/10/17

Brief description of your contribution to the 'paper':

Review of the manuscript with comments.

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:

Date: 24/10/2017

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Multi-gradient PSO algorithm for economic dispatch of thermal generating units in smart grid

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80 % Date: __/__/____

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- Anovel algorithmcalled multi-gradient particle swarm optimization (MG-PSO) was proposed, for economic dispatch (ED) of power of small and medium power systems.
2- Formulate the ED problem under several power constraints imposed by smart power grid and thermal generating units.
3- Evaluate the MG-PSO algorithm using 6 and 15 thermal generating units power systems.

Second Author

Name: Jagdish Chandra Patra Signature:

Percentage of contribution: 15 % Date: 24/10/2017

Brief description of your contribution to the 'paper':

Encouragement, support and advice.
Review, critical comments, theoretical justification

Third Author

Name: Ryszard Kowalczyk Signature:

Percentage of contribution: 5 % Date: 26/10/17

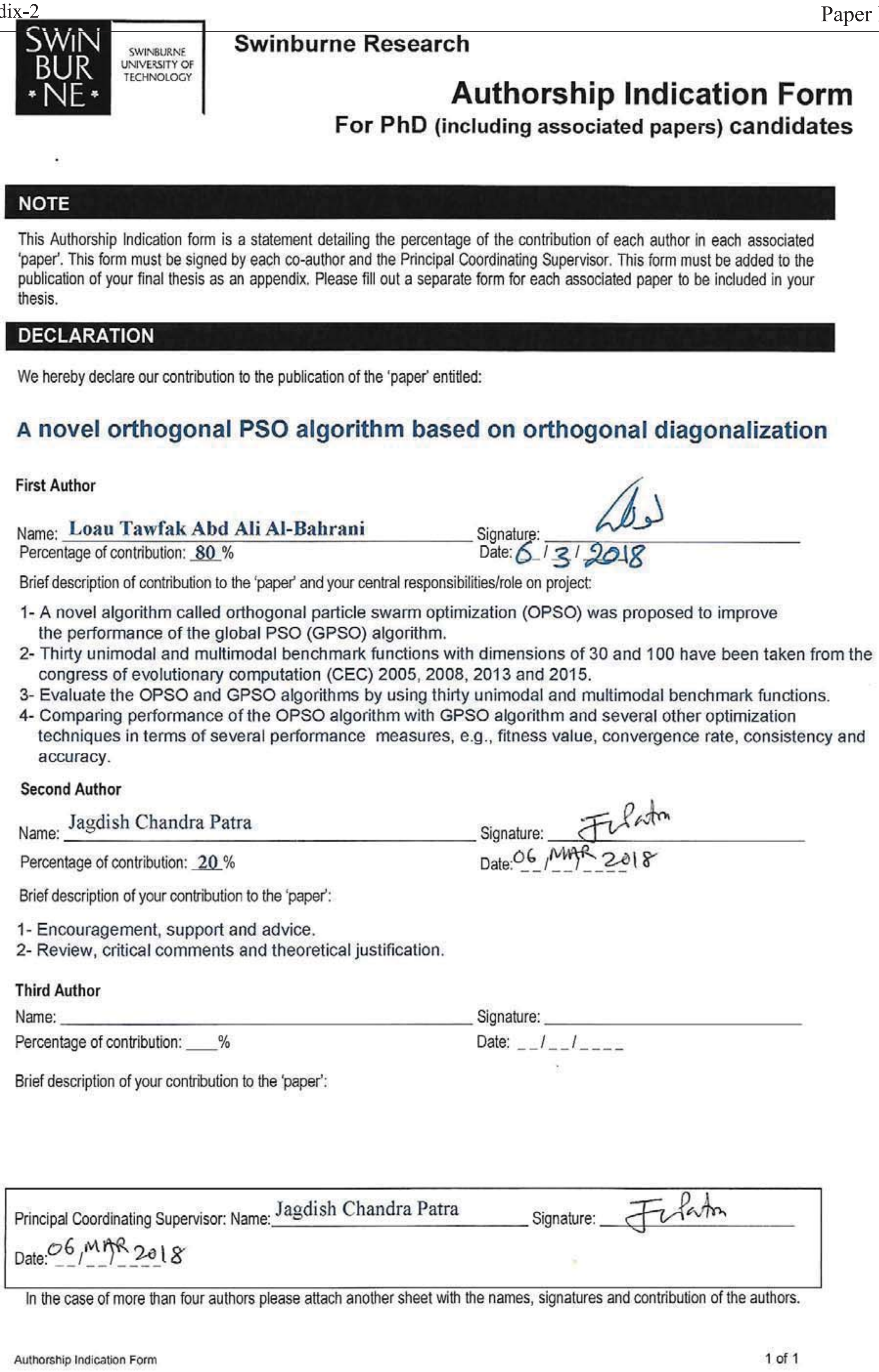
Brief description of your contribution to the 'paper':

Review of the manuscript with comments.

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:

Date: 24/10/2017

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.



SWINBURNE UNIVERSITY OF TECHNOLOGY

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

A novel orthogonal PSO algorithm based on orthogonal diagonalization

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:
Percentage of contribution: 80 % Date: 6 / 3 / 2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called orthogonal particle swarm optimization (OPSO) was proposed to improve the performance of the global PSO (GPSO) algorithm.
2- Thirty unimodal and multimodal benchmark functions with dimensions of 30 and 100 have been taken from the congress of evolutionary computation (CEC) 2005, 2008, 2013 and 2015.
3- Evaluate the OPSO and GPSO algorithms by using thirty unimodal and multimodal benchmark functions.
4- Comparing performance of the OPSO algorithm with GPSO algorithm and several other optimization techniques in terms of several performance measures, e.g., fitness value, convergence rate, consistency and accuracy.

Second Author

Name: Jagdish Chandra Patra Signature:
Percentage of contribution: 20 % Date: 06 / MAR / 2018

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

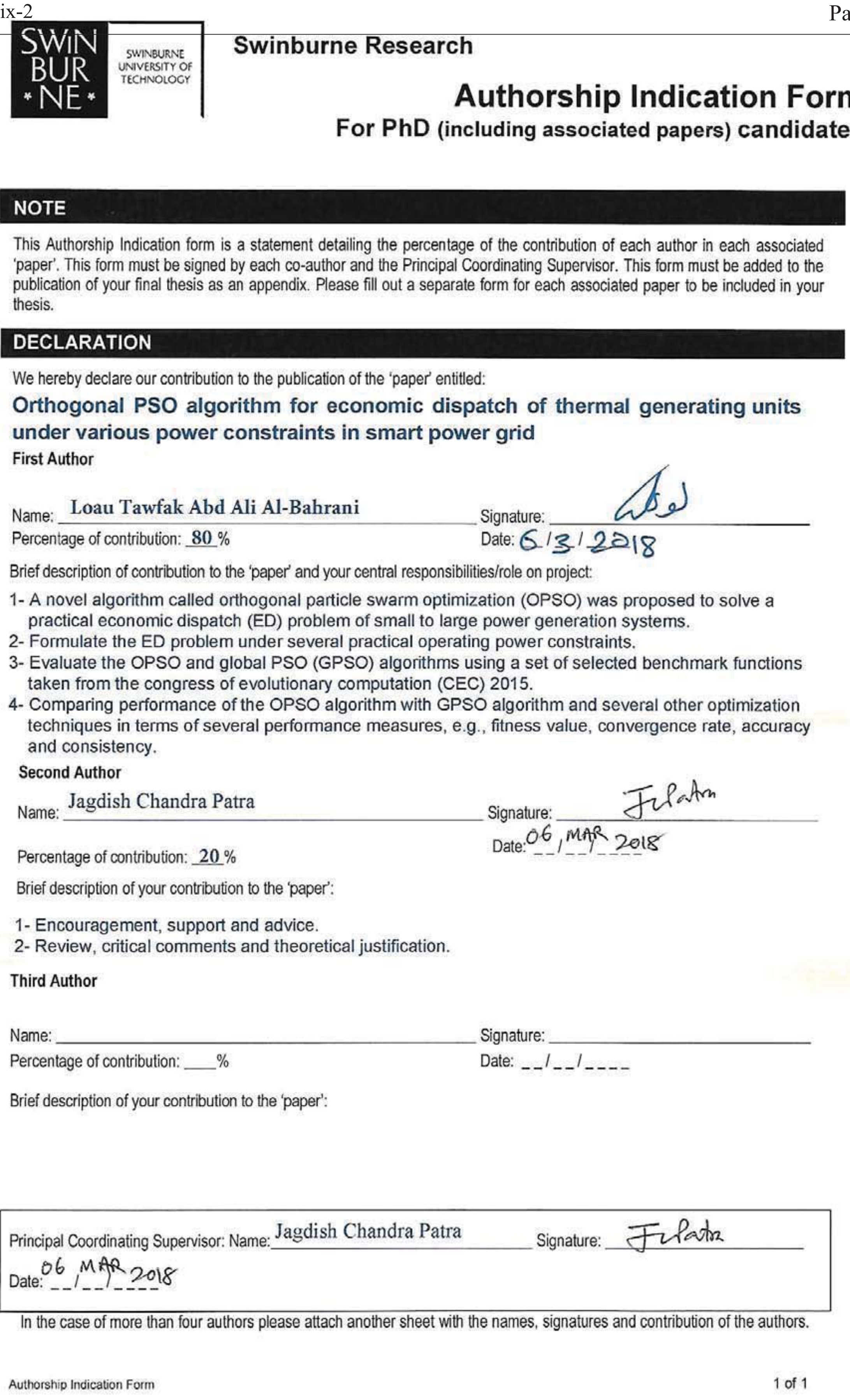
Name: ______ Signature: ______
Percentage of contribution: ____% Date: __/__/____

Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:
Date: 06 / MAR / 2018

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1



SWINBURNE UNIVERSITY OF TECHNOLOGY

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80 % Date: 6/3/2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called orthogonal particle swarm optimization (OPSO) was proposed to solve a practical economic dispatch (ED) problem of small to large power generation systems.
2- Formulate the ED problem under several practical operating power constraints.
3- Evaluate the OPSO and global PSO (GPSO) algorithms using a set of selected benchmark functions taken from the congress of evolutionary computation (CEC) 2015.
4- Comparing performance of the OPSO algorithm with GPSO algorithm and several other optimization techniques in terms of several performance measures, e.g., fitness value, convergence rate, accuracy and consistency.

Second Author

Name: Jagdish Chandra Patra Signature: JCPatra

Date: 06/MAR/2018

Percentage of contribution: 20 %

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

Name: ____ Signature: ____

Percentage of contribution: ___%

Date: __/__/____

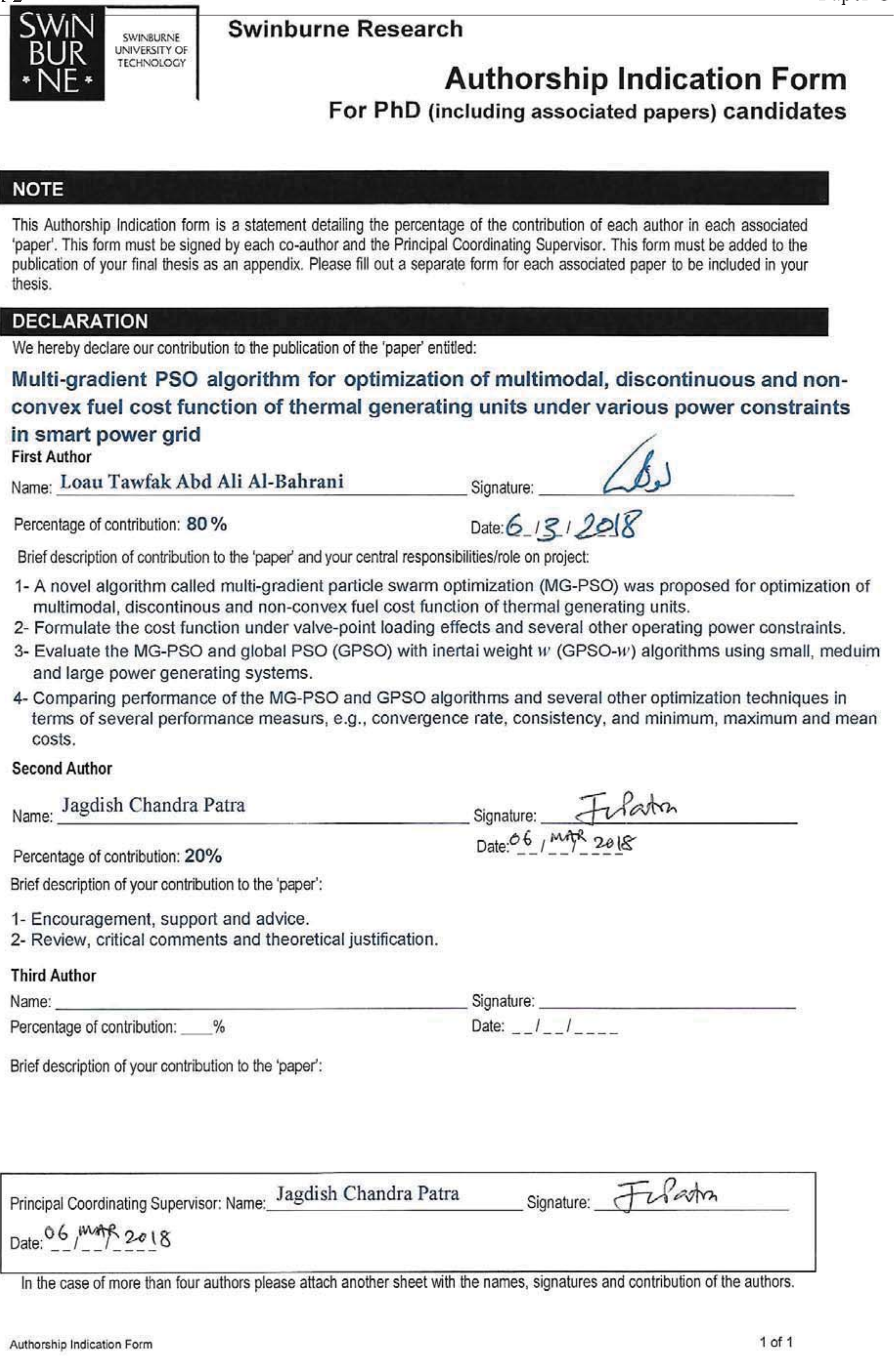
Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature: JCPatra

Date: 06/MAR/2018

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1



Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid

First Author

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: **80 %** Date: 6 / 3 / 2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called multi-gradient particle swarm optimization (MG-PSO) was proposed for optimization of multimodal, discontinous and non-convex fuel cost function of thermal generating units.
2- Formulate the cost function under valve-point loading effects and several other operating power constraints.
3- Evaluate the MG-PSO and global PSO (GPSO) with inertai weight w (GPSO-w) algorithms using small, meduim and large power generating systems.
4- Comparing performance of the MG-PSO and GPSO algorithms and several other optimization techniques in terms of several performance measurs, e.g., convergence rate, consistency, and minimum, maximum and mean costs.

Second Author

Name: Jagdish Chandra Patra Signature: JCPatra

Percentage of contribution: **20%** Date: 06 / MAR 2018

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

Name: ______ Signature: ______

Percentage of contribution: ____% Date: __/__/____

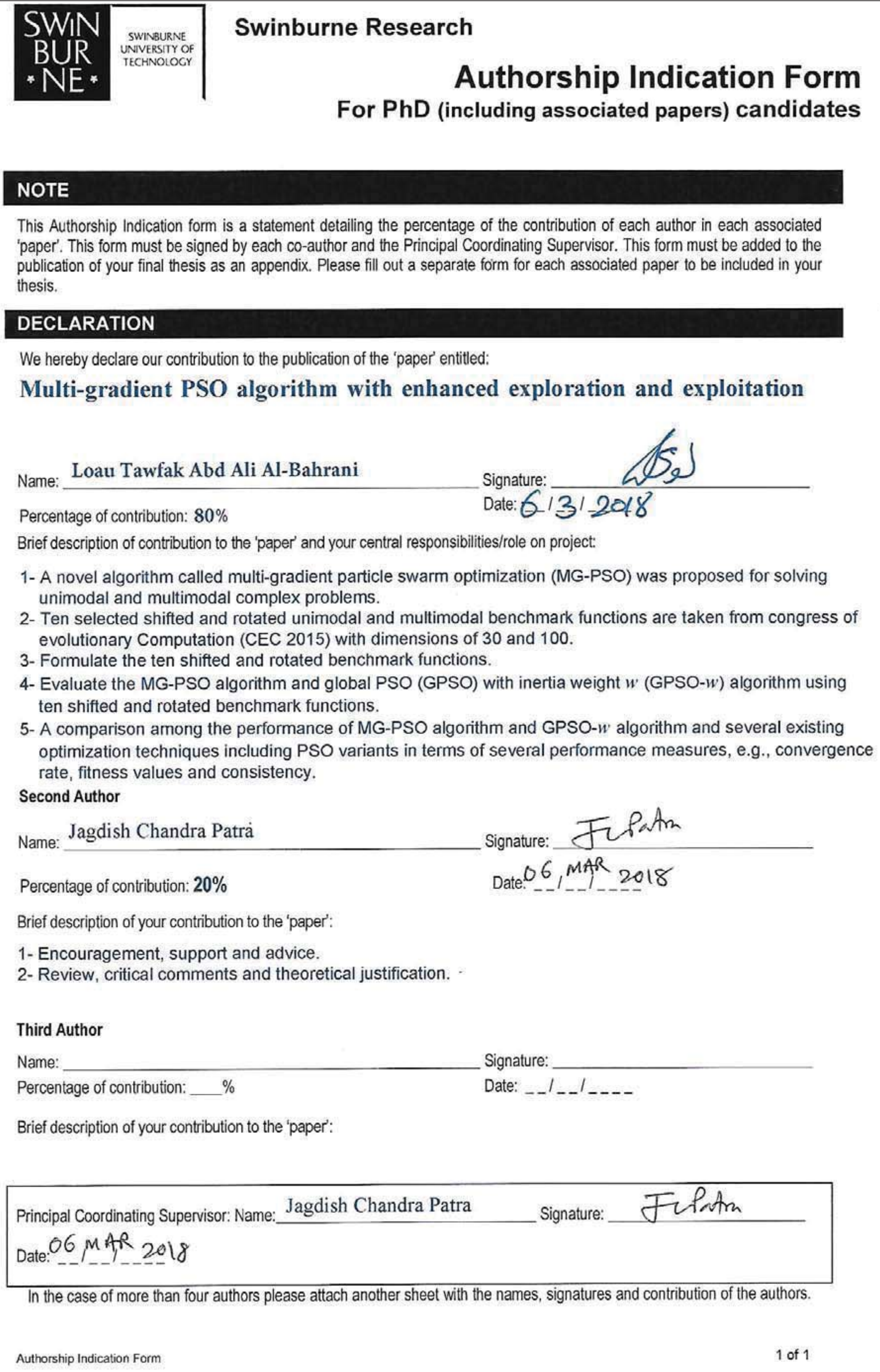
Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature: JCPatra

Date: 06 / MAR 2018

In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1



Swinburne University of Technology

Swinburne Research

Authorship Indication Form

For PhD (including associated papers) candidates

NOTE

This Authorship Indication form is a statement detailing the percentage of the contribution of each author in each associated 'paper'. This form must be signed by each co-author and the Principal Coordinating Supervisor. This form must be added to the publication of your final thesis as an appendix. Please fill out a separate form for each associated paper to be included in your thesis.

DECLARATION

We hereby declare our contribution to the publication of the 'paper' entitled:

Multi-gradient PSO algorithm with enhanced exploration and exploitation

Name: Loau Tawfak Abd Ali Al-Bahrani Signature:

Percentage of contribution: 80% Date: 6/3/2018

Brief description of contribution to the 'paper' and your central responsibilities/role on project:

1- A novel algorithm called multi-gradient particle swarm optimization (MG-PSO) was proposed for solving unimodal and multimodal complex problems.
2- Ten selected shifted and rotated unimodal and multimodal benchmark functions are taken from congress of evolutionary Computation (CEC 2015) with dimensions of 30 and 100.
3- Formulate the ten shifted and rotated benchmark functions.
4- Evaluate the MG-PSO algorithm and global PSO (GPSO) with inertia weight w (GPSO-w) algorithm using ten shifted and rotated benchmark functions.
5- A comparison among the performance of MG-PSO algorithm and GPSO-w algorithm and several existing optimization techniques including PSO variants in terms of several performance measures, e.g., convergence rate, fitness values and consistency.

Second Author

Name: Jagdish Chandra Patra Signature:

Percentage of contribution: **20%** Date: 06/MAR/2018

Brief description of your contribution to the 'paper':

1- Encouragement, support and advice.
2- Review, critical comments and theoretical justification.

Third Author

Name: Signature:

Percentage of contribution: ___% Date: __/__/____

Brief description of your contribution to the 'paper':

Principal Coordinating Supervisor: Name: Jagdish Chandra Patra Signature:

Date: 06/MAR/2018

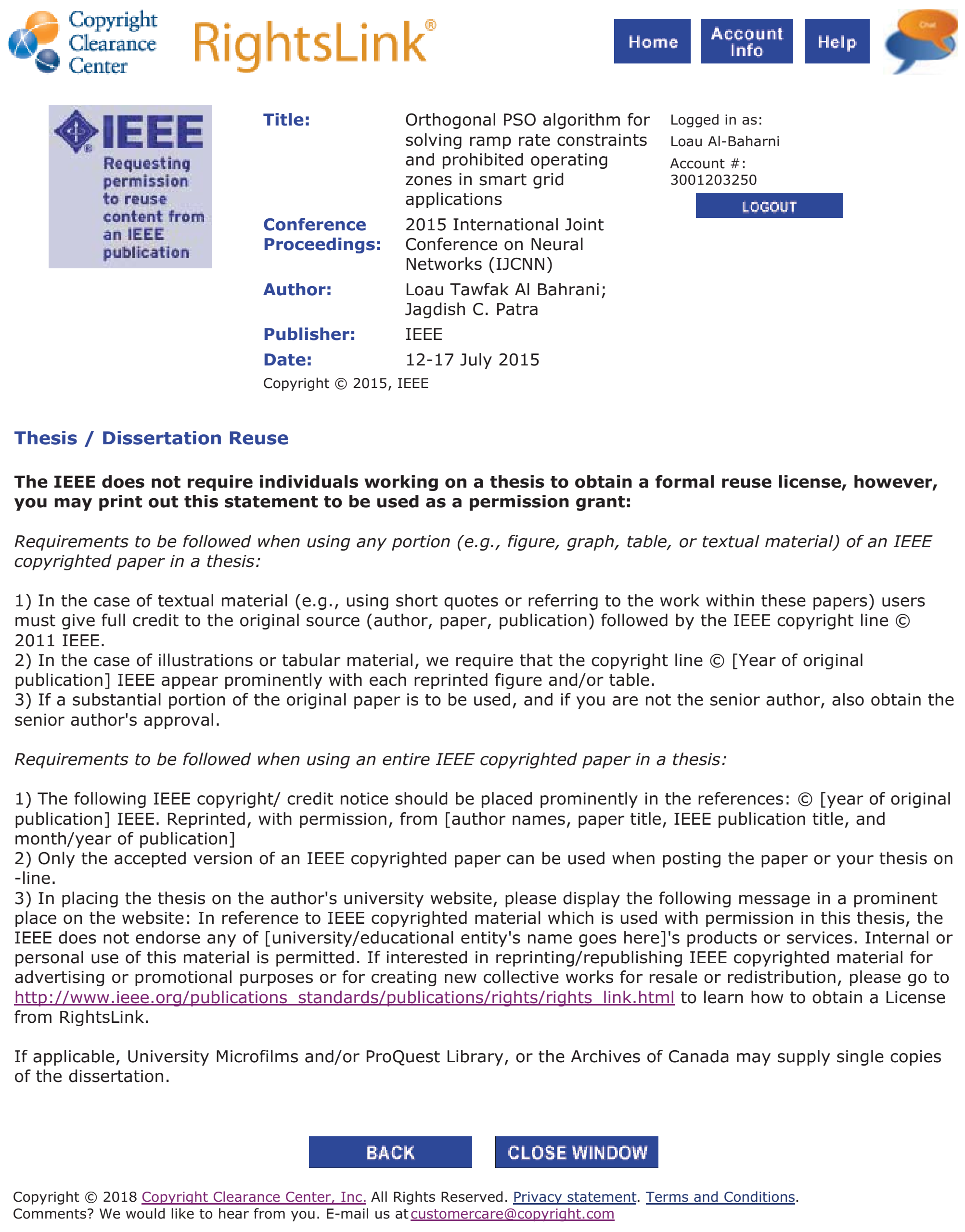
In the case of more than four authors please attach another sheet with the names, signatures and contribution of the authors.

Authorship Indication Form 1 of 1

Appendix-3: Publisher Permission

Here, publisher permission for the Papers A-G (Paper H is under review) are presented as follows.

Rightslink® by Copyright Clearance Center


Copyright Clearance Center
RightsLink®
Home
Account Info
Help
IEEE
Requesting permission to reuse content from an IEEE publication
LOGOUT
BACK
CLOSE WINDOW

Title: Orthogonal PSO algorithm for solving ramp rate constraints and prohibited operating zones in smart grid applications

Conference Proceedings: 2015 International Joint Conference on Neural Networks (IJCNN)

Author: Loau Tawfak Al Bahrani; Jagdish C. Patra

Publisher: IEEE

Date: 12-17 July 2015

Copyright © 2015, IEEE

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

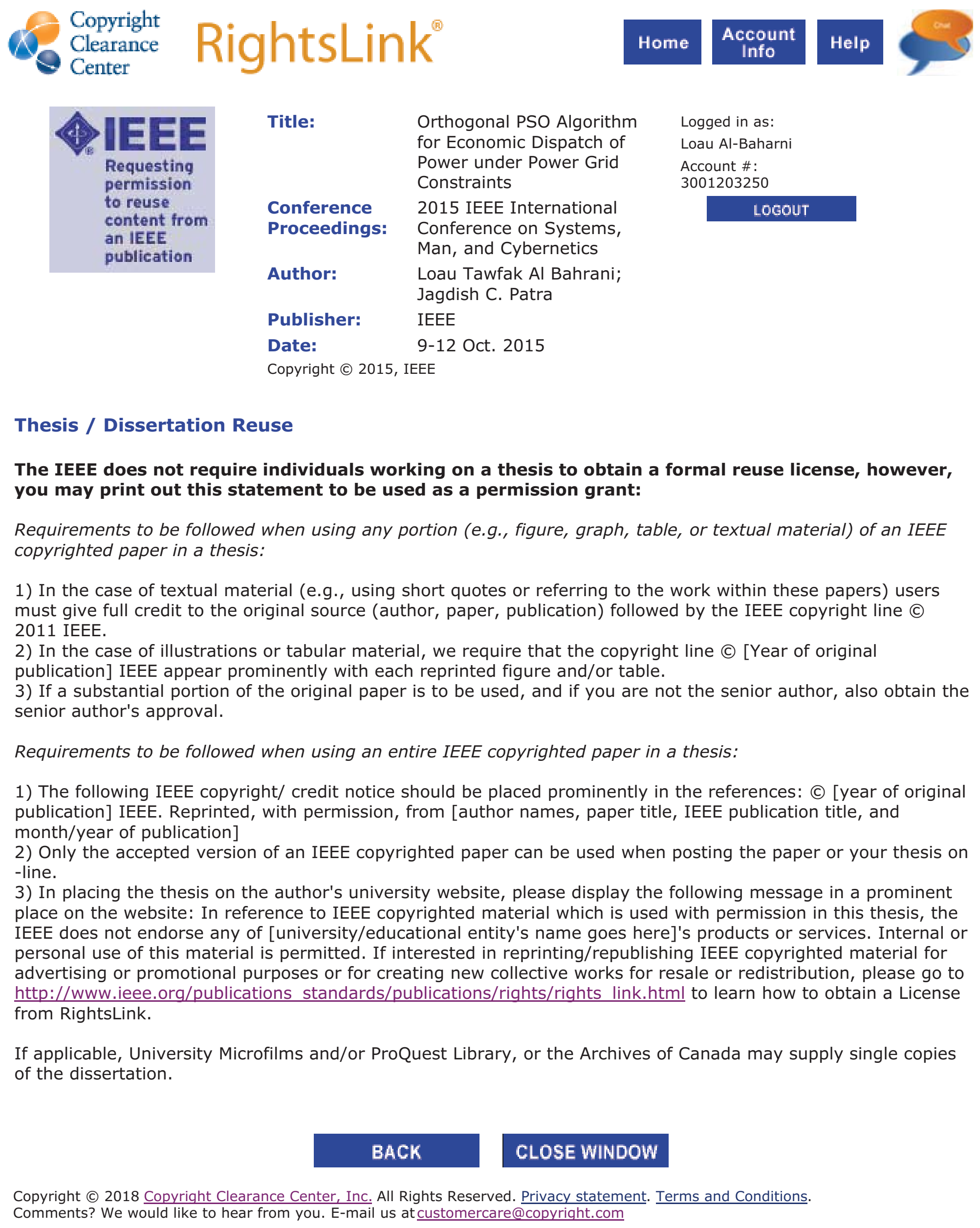
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on -line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Rightslink® by Copyright Clearance Center


Copyright Clearance Center
RightsLink®
Home
Account Info
Help
Chat
IEEE
Requesting permission to reuse content from an IEEE publication
LOGOUT
BACK
CLOSE WINDOW

Title: Orthogonal PSO Algorithm for Economic Dispatch of Power under Power Grid Constraints

Conference Proceedings: 2015 IEEE International Conference on Systems, Man, and Cybernetics

Author: Loau Tawfak Al Bahrani; Jagdish C. Patra

Publisher: IEEE

Date: 9-12 Oct. 2015

Copyright © 2015, IEEE

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

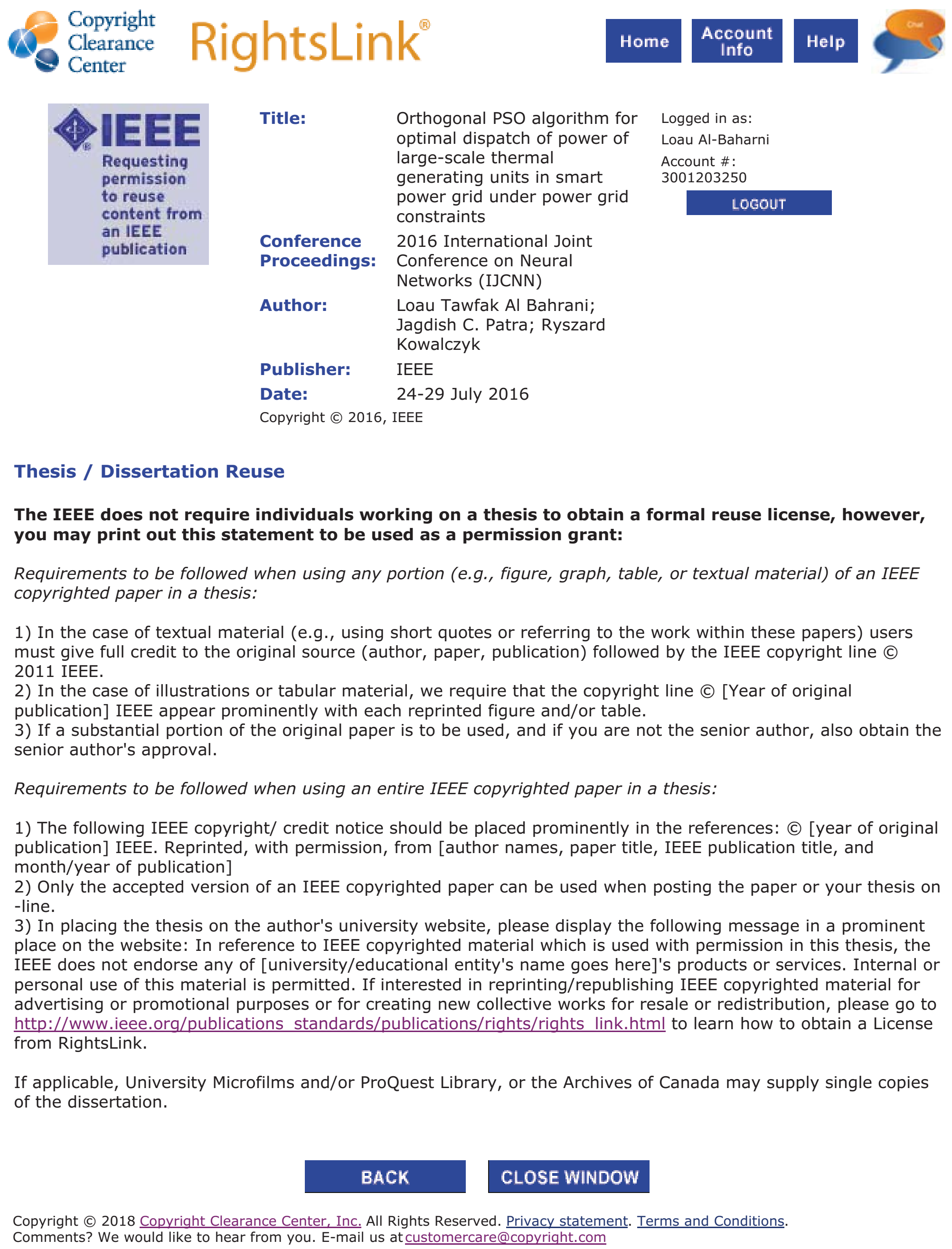
Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on -line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

https://s100.copyright.com/AppDispatchServlet 22/02/2018

Rightslink® by Copyright Clearance Center


Copyright Clearance Center
RightsLink®
Home
Account Info
Help
IEEE
Requesting permission to reuse content from an IEEE publication
LOGOUT
BACK
CLOSE WINDOW

Title:	Orthogonal PSO algorithm for optimal dispatch of power of large-scale thermal generating units in smart power grid under power grid constraints
Conference Proceedings:	2016 International Joint Conference on Neural Networks (IJCNN)
Author:	Loau Tawfak Al Bahrani; Jagdish C. Patra; Ryszard Kowalczyk
Publisher:	IEEE
Date:	24-29 July 2016

Copyright © 2016, IEEE

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

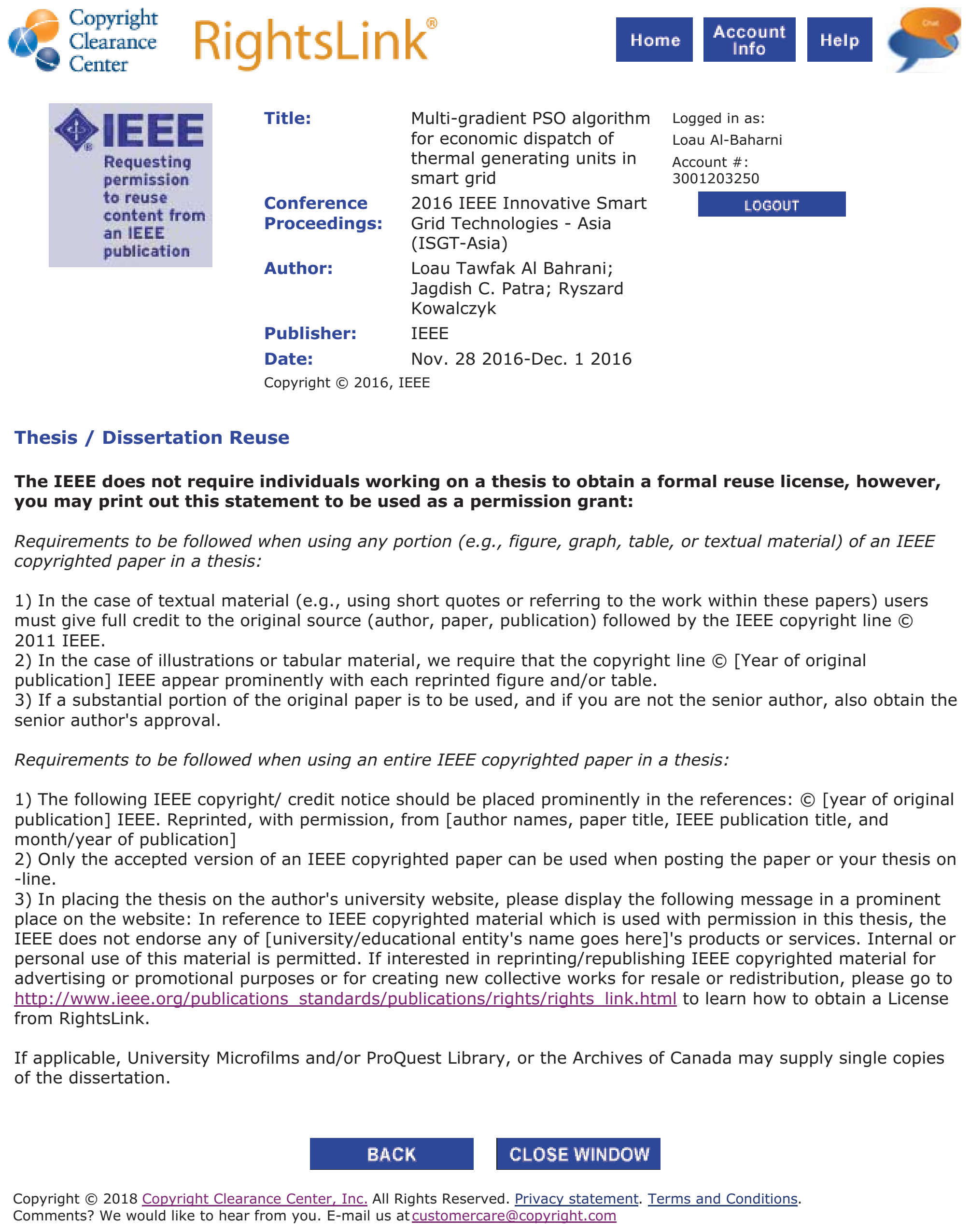
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on -line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

https://s100.copyright.com/AppDispatchServlet 22/02/2018


Copyright Clearance Center
RightsLink®
Home
Account Info
Help
Chat
IEEE
Requesting permission to reuse content from an IEEE publication
LOGOUT
BACK
CLOSE WINDOW

Title: Multi-gradient PSO algorithm for economic dispatch of thermal generating units in smart grid

Conference Proceedings: 2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)

Author: Loau Tawfak Al Bahrani; Jagdish C. Patra; Ryszard Kowalczyk

Publisher: IEEE

Date: Nov. 28 2016-Dec. 1 2016

Copyright © 2016, IEEE

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

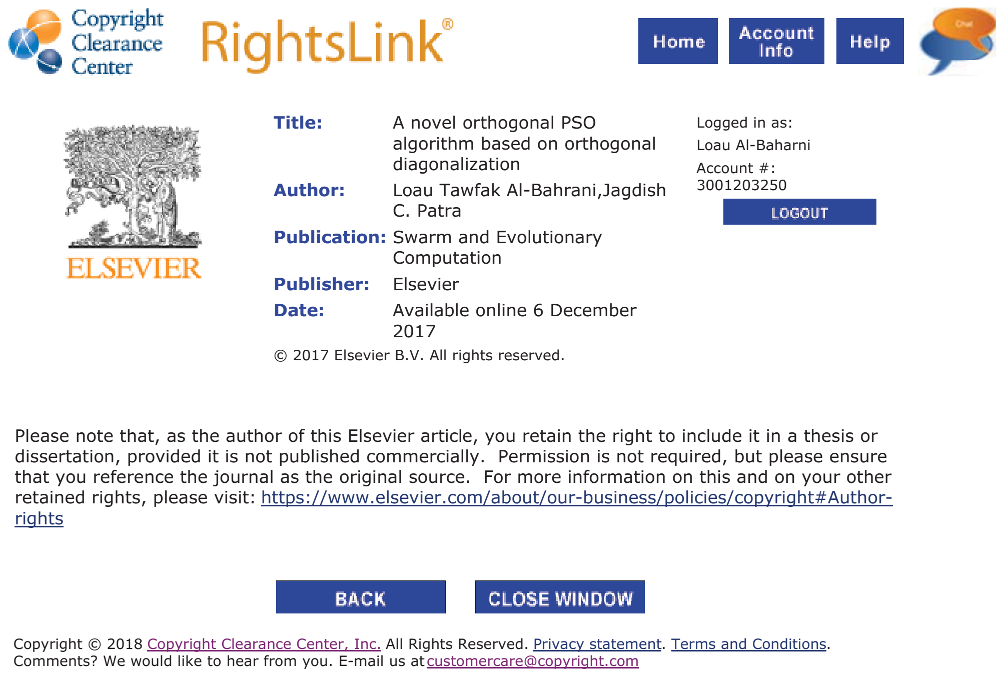
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on -line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Rightslink® by Copyright Clearance Center



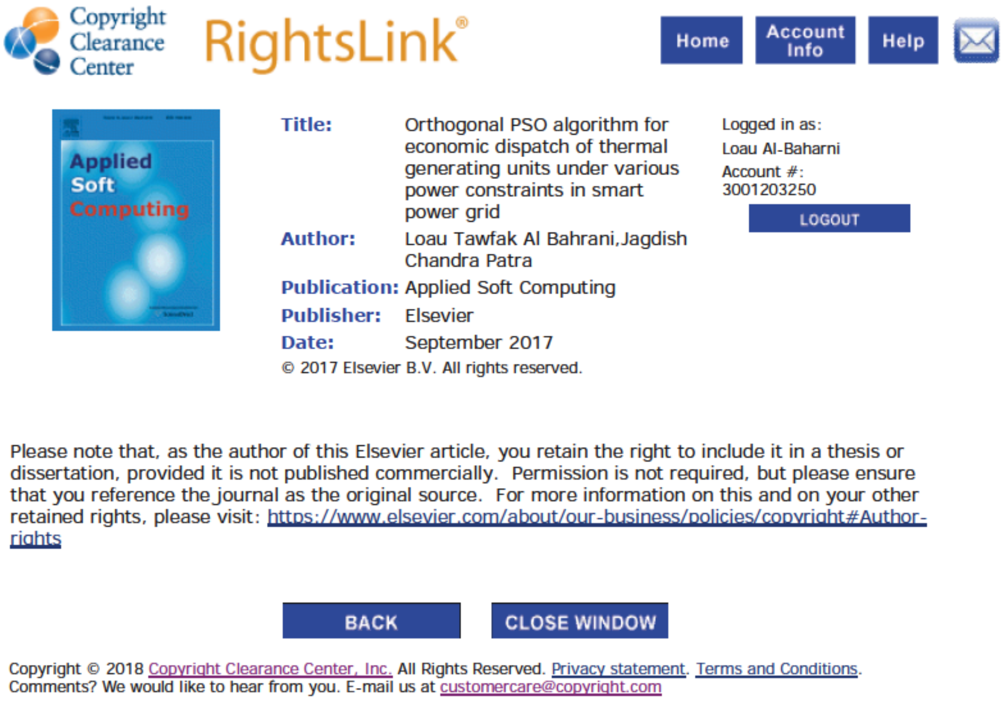
Title:	A novel orthogonal PSO algorithm based on orthogonal diagonalization
Author:	Loau Tawfak Al-Bahrani,Jagdish C. Patra
Publication:	Swarm and Evolutionary Computation
Publisher:	Elsevier
Date:	Available online 6 December 2017

© 2017 Elsevier B.V. All rights reserved.

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

Rightslink® by Copyright Clearance Center

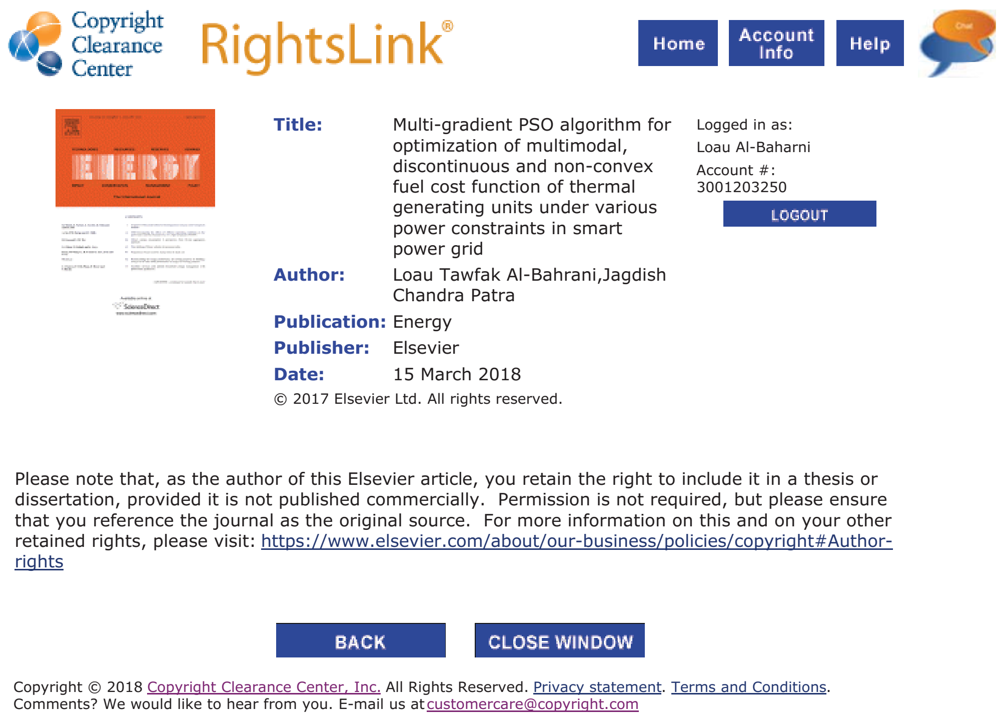


Title:	Orthogonal PSO algorithm for economic dispatch of thermal generating units under various power constraints in smart power grid
Author:	Loau Tawfak Al Bahrani,Jagdish Chandra Patra
Publication:	Applied Soft Computing
Publisher:	Elsevier
Date:	September 2017

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

Rightslink® by Copyright Clearance Center



Title:	Multi-gradient PSO algorithm for optimization of multimodal, discontinuous and non-convex fuel cost function of thermal generating units under various power constraints in smart power grid
Author:	Loau Tawfak Al-Bahrani,Jagdish Chandra Patra
Publication:	Energy
Publisher:	Elsevier
Date:	15 March 2018

Logged in as:
Loau Al-Baharni
Account #:
3001203250

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

https://s100.copyright.com/AppDispatchServlet 22/02/2018