

# Niching Ant Colony Optimisation

Daniel John Angus, BEng. BSci.

Complex Intelligent Systems Laboratory  
Faculty of Information & Communication Technologies  
Swinburne University of Technology  
Melbourne, Australia

Submitted in partial fulfilment for the degree of  
Doctor of Philosophy

July 2008



# Abstract

Optimisation, in the mathematical sense, is the process of finding solutions to a problem such that one or many objectives are minimised or maximised. Optimisation problems are diverse in form, necessitating the need for many different optimisation algorithms. These algorithms can be defined in two categories: deterministic and non-deterministic algorithms. Deterministic algorithms usually have set execution schedules and are fairly exhaustive search methods. Non-deterministic algorithms use randomness and prove useful for problems where it may not be possible to execute a deterministic algorithm due to the size, or nature of the problem search space. In these cases a deterministic algorithm may take days or months to find an optimal solution, whereas a non-deterministic algorithm can usually find an approximate but still near-optimal solution in a matter of minutes or seconds.

Ant Colony Optimisation (ACO), a non-deterministic algorithm class, aims to mimic (and exploit) the behaviours of real ant colonies to solve real-world optimisation problems. ACO algorithms are a class of constructive heuristic algorithms, which build solutions to a given optimisation problem, one solution component at a time, according to a defined set of rules (heuristics), i.e. starting with an 'empty' solution add solution components until a complete solution is built. ACO algorithms are unique in this class by their use of past solutions in manipulating an artificial 'pheromone'. The pheromone being a measure associated to every unique solution component which reflects the estimated utility of this solution component. These pheromone values are used to bias solution construction by influencing the probability of a solution component being added to a growing solution based on the amount of pheromone it contains.

The Population-based Ant Colony Optimisation (PACO) algorithm is a recently developed ant-inspired algorithm which, unlike traditional ACO algorithms, maintains a finite population of solutions as well as pheromone information. It has been demonstrated to be an efficient optimisation algorithm when applied to a range of difficult single-objective, multi-objective and dynamic problem instances. In this thesis a review of existing PACO algorithms is offered and an identification of common features is used in the development of a Population-based ACO framework.

Using the new Population-based ACO framework, several new PACO algorithms imbued with a diversity preservation technique known as niching are defined. Niching has been used extensively in the field of Evolutionary Computation, but to the best knowledge of the author, has never been explicitly applied to an ACO algorithm per se. An empirical analysis of these novel

implementations is presented using a variety of single and multiple objective continuous function and combinatorial optimisation problems. These optimisation problems have been chosen since they demonstrate the advantages and disadvantages of adding niching to a PACO algorithm. To conclude, two of these new PACO algorithms are applied to a real-world optimisation problem.

# Acknowledgments

A body of work such as this is never completed without a number of challenges and difficulties along the way. By far my greatest challenge was pursuing my studies without a scholarship for my first year. Although tricky, stressful and at times depressing this first year taught me many lessons on life, love and how fulfilling it can be to feed yourself for a week on \$5. My ability to sense the free campus BBQ also became finely tuned! I wish to firstly thank those people who were there for me during this time who ensured that I continued to pursue my studies. Although I can't say it has been clear sailing from there, it has certainly been made easier thanks to the understanding and care of so many family members, friends and colleagues.

To my wife Katherine, your patience, care and understanding have made these last three years some of the most memorable in my life so far. You were with me from day one, and have stood beside me the whole way, for this I thank you.

Anne and Clive Angus, as my parents you have always encouraged me and given me the support I need. You have given me the strength and courage to pursue my dreams, for this I thank you. To Kate and Leigh Thomson, as my sister and brother in law you have always encouraged my studies and showed a genuine interest in my academic pursuits, thank you.

My colleagues at the Complex Intelligent Systems group, you have all helped me at some point and I thank you all for the many meaningful discussions shared over a beer. To my supervisor Prof. Tim Hendtlass, you inspired me from early in my undergraduate research days to pursue a PhD, and here I am now. You have played a most important role as my supervisor and mentor. Thank you for your trust, patience and understanding. To my co-supervisor Assoc. Prof. Marcus Randall your support is also appreciated. To my colleague and old room-mate Jason Brownlee, I have valued your input highly and you deserve an appropriate amount of thanks for your contribution. Your ideas about anything research related always seem to be light-years ahead of mine and I thank you for the positive contribution you have made to my research, and to my life in general.

Family and friends be they university colleagues past or present, high school chums, Rovers and the like, you have all been there at some point to offer encouragement, thank you all.

On a professional note I must thank Adam Deller for his help with Chapter 8 which was derived from his PhD work with the Centre for Astrophysics and Supercomputing at Swinburne University. I also acknowledge the Australian Government for providing me with an Australian Postgraduate Award for a partial component of my PhD program.



# Declaration

I declare that this thesis contains no material that has been accepted for the award of any other degree or diploma and to the best of my knowledge contains no material previously published or written by another person except where due reference is made in the text of this thesis.

Daniel John Angus





# Publications Arising from this Thesis

Angus, D. *Niching for Population-based Ant Colony Optimization*, Proceedings of the 2nd International IEEE Conference on e-Science and Grid Computing, Workshop on Biologically Inspired Optimisation Methods for Parallel and Distributed Architectures: Algorithms, Systems and Applications, 2006.

Angus, D. *Crowding Population-based Ant Colony Optimisation for the Multi-objective Travelling Salesman Problem*, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007), 2007, pp. 333–340.

Angus, D. *Population-based Ant Colony Optimisation for Multi-objective Function Optimisation*, Proceedings of the 3rd Australian Conference on Artificial Life (ACAL2007), 2007, pp. 232–244.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background to the study . . . . .	1
1.2	Research Motives and Contributions . . . . .	2
1.3	Thesis Structure . . . . .	3
<b>2</b>	<b>Optimisation Problems</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Combinatorial Optimisation . . . . .	6
2.2.1	Travelling Salesman Problem . . . . .	6
2.3	Continuous Function Optimisation . . . . .	7
2.4	Modality . . . . .	8
2.5	Multiple Objective Optimisation . . . . .	8
2.5.1	Objective and Solution space . . . . .	9
2.5.2	Pareto Optimality . . . . .	10
2.5.3	Dominance . . . . .	10
2.5.4	Solving Multiple Objective Optimisation Problems . . . . .	11
2.5.5	Existing Algorithms . . . . .	11
2.6	Choice of Problems . . . . .	11
2.7	Performance Statistics . . . . .	13
2.8	Chapter Summary . . . . .	14
<b>3</b>	<b>Ant Colony Optimisation</b>	<b>15</b>
3.1	Introduction . . . . .	15
3.2	Early Development . . . . .	15
3.2.1	Biological Inspiration . . . . .	15
3.2.2	The Ant Systems Algorithm . . . . .	17

## CONTENTS

3.2.3	Ant Colony Optimisation . . . . .	18
3.3	The Ant Colony Optimisation Metaheuristic Framework . . . . .	18
3.3.1	Pheromone mapping . . . . .	20
3.3.2	Ants Generation and Activity . . . . .	20
3.3.3	Pheromone Trail Evaporation . . . . .	21
3.3.4	Daemon Actions . . . . .	21
3.4	ACO Algorithms . . . . .	21
3.4.1	Ant System for the Travelling Salesman Problem . . . . .	21
3.4.2	Ant Colony Systems . . . . .	23
3.4.3	<i>MAX-MIN</i> Ant Systems . . . . .	23
3.4.4	Rank Based Ant Systems . . . . .	24
3.5	Salient Issues in ACO . . . . .	24
3.5.1	Ant Inspired Algorithms . . . . .	24
3.5.2	Local search . . . . .	25
3.6	Chapter Summary . . . . .	26
<b>4</b>	<b>Population-based Ant Colony Optimisation</b>	<b>27</b>
4.1	Introduction . . . . .	27
4.2	FIFO-Queue ACO . . . . .	27
4.2.1	Algorithm Description . . . . .	28
4.2.2	A Useful Population . . . . .	29
4.3	PACO for Dynamic Problems . . . . .	29
4.3.1	Population Update Strategies . . . . .	30
4.4	PACO for Multi-objective Problems . . . . .	30
4.5	ACO for Continuous Domains . . . . .	31
4.6	Hardware Implementation of ACO . . . . .	34
4.7	A Population-based ACO Meta-heuristic Framework . . . . .	34
4.7.1	Process Level Organisation . . . . .	35
4.7.2	Solution Storage and Maintenance . . . . .	35
4.8	Similarity to Genetic Algorithms . . . . .	36
4.8.1	Population Based Incremental Learning . . . . .	36
4.8.2	Similarities and Differences . . . . .	37
4.8.3	General Comments . . . . .	38
4.9	Chapter Summary . . . . .	38

<b>5</b>	<b>Niching for PACO</b>	<b>41</b>
5.1	Introduction . . . . .	41
5.2	Evolutionary Computation . . . . .	41
5.2.1	Genetic Algorithm . . . . .	42
5.3	Niching in Evolutionary Computation . . . . .	43
5.3.1	Crowding . . . . .	44
5.3.2	Fitness Sharing . . . . .	44
5.3.3	Difference Measures . . . . .	46
5.3.4	Pros and Cons of Niching . . . . .	48
5.3.5	Sequential versus Parallel Niching . . . . .	48
5.4	Niching Ant Colony Optimisation . . . . .	49
5.4.1	Constraints . . . . .	49
5.4.2	Fitness Sharing Population-based Ant Colony Optimisation . . . . .	49
5.4.3	Crowding Population-based Ant Colony Optimisation . . . . .	50
5.4.4	Alternative ACO Diversity Preservation Methods . . . . .	50
5.5	Chapter Summary . . . . .	52
<b>6</b>	<b>Single Objective Optimisation</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	Niching PACO for the Travelling Salesman Problem . . . . .	54
6.2.1	Algorithm Details . . . . .	54
6.2.2	Results and Discussion: Crown Problem . . . . .	55
6.2.3	Results and Discussion: Standard Problems . . . . .	56
6.2.4	Summary . . . . .	63
6.3	Niching PACO for Continuous Function Optimisation . . . . .	63
6.3.1	Algorithm Details . . . . .	63
6.3.2	Test Problems Used . . . . .	67
6.3.3	Performance Metrics . . . . .	77
6.3.4	Results and Discussion: Sensitivity Analysis . . . . .	78
6.3.5	Results and Discussion: Quantitative Analysis . . . . .	86
6.3.6	Summary . . . . .	91
6.4	Chapter Summary . . . . .	93
<b>7</b>	<b>Multiple Objective Optimisation</b>	<b>95</b>

## CONTENTS

7.1	Introduction . . . . .	95
7.2	Existing Algorithms . . . . .	96
7.3	Performance Metrics . . . . .	99
7.3.1	Summary Attainment Surface . . . . .	99
7.3.2	Hypervolume . . . . .	100
7.3.3	Epsilon Indicator . . . . .	100
7.3.4	Use of Performance Metrics . . . . .	101
7.4	Niching PACO for the Multiple Objective Travelling Salesman Problem . . . . .	101
7.4.1	Algorithm Details . . . . .	102
7.4.2	Test Problems . . . . .	103
7.4.3	Results and Discussion . . . . .	104
7.4.4	Summary . . . . .	118
7.5	Niching PACO for Multi-objective Function Optimisation . . . . .	118
7.5.1	Algorithm Details . . . . .	119
7.5.2	Test Problems . . . . .	121
7.5.3	Results and Discussion . . . . .	124
7.5.4	Summary . . . . .	140
7.6	Chapter Summary . . . . .	140
<b>8</b>	<b>Case Study: Antenna Correlation</b>	<b>143</b>
8.1	Problem Details . . . . .	144
8.1.1	Problem Description . . . . .	144
8.1.2	Objective Function . . . . .	146
8.1.3	Problem Data . . . . .	146
8.2	Experiment One: Naive Optimisation . . . . .	147
8.2.1	Experiment Description . . . . .	147
8.2.2	Results and Discussion . . . . .	148
8.3	Experiment Two: Problem Decomposition . . . . .	153
8.3.1	Experiment Description . . . . .	153
8.3.2	Results and Discussion . . . . .	153
8.4	Experiment Three: Application of Heuristics . . . . .	157
8.4.1	Experiment Description . . . . .	157
8.4.2	Results and Discussion . . . . .	157
8.5	Final Solution Analysis . . . . .	162

8.6 Chapter Summary . . . . .	164
<b>9 Summary and Final Remarks</b>	<b>165</b>
9.1 Summary . . . . .	165
9.2 Contributions . . . . .	166
9.3 Future Work . . . . .	167
<b>References</b>	<b>169</b>
<b>10 Appendix</b>	<b>183</b>





# List of Figures

2.1	Optimal tour of the PCB3038 TSP . . . . .	7
2.2	Illustrations of different Pareto fronts plotted in the objective space. . . . .	10
3.1	Double bridge experimental setup . . . . .	16
3.2	Graphical reproduction of the double bridge experiment using paths of equal length. . . . .	17
3.3	Graphical reproduction of the double bridge experiment using paths of unequal length. . . . .	17
3.4	Process organisation of the Ant Colony Optimisation Metaheuristic Framework .	19
3.5	Early development of the field of ACO . . . . .	19
4.1	An illustration of the new solution creation procedure of AC OCD. . . . .	33
4.2	Process organisation of original ACO framework versus the PACO framework .	35
5.1	Generic algorithmic structure of Evolutionary Computation . . . . .	42
6.1	Average number of similar edges amongst the best 100 and a randomly selected 100 solutions to the Burma14 TSP . . . . .	55
6.2	Crown TSP problem visual representation . . . . .	56
6.3	Optimal solutions to Crown problem . . . . .	56
6.4	Occurrence of the two optimal paths over time for a single experiment run of the Crown TSP. . . . .	57
6.5	An illustration of the new solution creation procedure for FSPACO-CFO and CPACO-CFO. . . . .	66
6.6	Example of standard deviation used for new solution creation in FSPACO-CFO and CPACO-CFO. . . . .	67
6.7	Three Pot Holes function defined in two dimensions. . . . .	69
6.8	Ackley's function defined in two dimensions. . . . .	70
6.9	Branin's R-Cos function defined in two dimensions. . . . .	71

LIST OF FIGURES

6.10	Himmelblau’s function defined in two dimensions. . . . .	72
6.11	Rastrigin’s function defined in two dimensions. . . . .	73
6.12	Schaffer’s function defined in two dimensions. . . . .	74
6.13	Schwefel’s function defined in two dimensions. . . . .	75
6.14	Shekel’s foxholes function defined in two dimensions. . . . .	76
6.15	Plots indicating the effect on the max-peak-like performance metric when the niche radius and history exponent of FSPACO-CFO are varied. . . . .	79
6.16	Plots indicating the effect on the max-peak-like performance metric when the crowding window size and history exponent of CPACORank-CFO are varied. . . . .	80
6.17	Plots indicating the effect on the max-peak-like performance metric when the crowding window size and history exponent of CPACOQuality-CFO are varied. . . . .	81
6.18	Plots indicating the effect on the max-peak-like performance metric when the crowding window size of CPACOUntity-CFO is varied. . . . .	82
6.19	Example highlighting the effect of replacement error. . . . .	83
6.20	Results of CFO quantitative analysis . . . . .	88
6.20	Results of CFO quantitative analysis . . . . .	89
6.21	Results of CFO analysis as number of dimensions increases . . . . .	92
6.22	Results from Three Pot Holes defined in 100 dimensions with a varied number of function evaluations. . . . .	93
7.1	Example of the resultant ranks assigned to a set of solutions using the NSGA-II dominance ranking procedure . . . . .	97
7.2	Example of two hypervolumes created from two different sets, A and B. In this example A would be better than B since A’s hypervolume is larger. . . . .	100
7.3	1% (best) attainment surface for kroA100 and kroB100 using PACO-MO & CPACO-MOTSP . . . . .	105
7.4	50% (average) attainment surface for kroA100 and kroB100 using PACO-MO & CPACO-MOTSP . . . . .	105
7.5	1% (best) attainment surface for kroC100 and kroD100 using PACO-MO & CPACO-MOTSP . . . . .	106
7.6	50% (average) attainment surface for kroC100 and kroD100 using PACO-MO & CPACO-MOTSP . . . . .	106
7.7	1% (best) attainment surface for kroA150 and kroB150 using PACO-MO & CPACO-MOTSP . . . . .	107
7.8	50% (average) attainment surface for kroA150 and kroB150 using PACO-MO & CPACO-MOTSP . . . . .	107

7.9	1% (best) attainment surface for kroA200 and kroB200 using PACO-MO & CPACO-MOTSP . . . . .	108
7.10	50% (average) attainment surface for kroA200 and kroB200 using PACO-MO & CPACO-MOTSP . . . . .	108
7.11	Example of sampling behaviour of PACO-MO . . . . .	110
7.12	Example of sampling behaviour of CPACO-MOTSP . . . . .	110
7.13	1% (best) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 (isolating KroA100 & KroB100) with 100,000, 250,000 and 500,000 solution evaluations. . . . .	111
7.14	50% (average) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 (isolating KroA100 & KroB100) with 100,000, 250,000 and 500,000 solution evaluations. . . . .	111
7.15	1% (best) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 (isolating KroA100 & KroB100) with 100,000, 250,000 and 500,000 solution evaluations. . . . .	112
7.16	50% (average) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 (isolating KroA100 & KroB100) with 100,000, 250,000 and 500,000 solution evaluations. . . . .	112
7.17	1% (best) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 ( <i>CPACO_ABCD</i> ), and CPACO-MOTSP applied to only kroA100 & kroB100 ( <i>CPACO_AB</i> ) . . . . .	113
7.18	50% (average) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 ( <i>CPACO_ABCD</i> ), and CPACO-MOTSP applied to only kroA100 & kroB100 ( <i>CPACO_AB</i> ) . . . . .	113
7.19	1% (best) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 ( <i>CPACO_ABCD</i> ), and CPACO-MOTSP applied to only kroC100 & kroD100 ( <i>CPACO_CD</i> ) . . . . .	114
7.20	50% (average) attainment surface for CPACO-MOTSP applied to kroA100, kroB100, kroC100 & kroD100 ( <i>CPACO_ABCD</i> ), and CPACO-MOTSP applied to only kroC100 & kroD100 ( <i>CPACO_CD</i> ) . . . . .	114
7.21	Example of selection probability using only pheromone information. . . . .	117
7.22	Example of selection probability using pheromone and heuristic information. . . . .	118
7.23	Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP1 . . . . .	125
7.24	Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP2 . . . . .	126
7.25	Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP3 . . . . .	127

LIST OF FIGURES

7.26 Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP4 . . . . . 128

7.27 Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP6 . . . . . 129

7.28 Distribution of final population of PACO-MOFO and NSGA-II on MOP2 and MOP3 . . . . . 131

7.28 Distribution of final population of PACO-MOFO and NSGA-II on MOP4 and MOP6 . . . . . 132

7.29 Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to the benchmark function, MOP2, in 2, 5, 10, 15 and 20 dimensions . . . . 133

7.30 (cont'd) Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to the benchmark function, MOP4, in 5, 10, 15 and 20 dimensions . . . 134

7.30 (cont'd) Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to the benchmark function, MOP4, in 5, 10, 15 and 20 dimensions . . . 135

7.31 Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP5 . . . . . 138

7.32 Attainment surfaces generated from 50 runs of PACO-MOFO and NSGA-II applied to MOP7 . . . . . 139

8.1 The observed residual delay for one observing band of a baseline (Parkes and ATCA) tracked over a 24 hour period taken from data set STA-131AV. . . . . 145

8.2 Arrangement of baselines for the STA-131AU and STA-131AV datasets. . . . . 147

8.3 Figures illustrating 100 interesting features of two known problems and one unknown problem. . . . . 149

8.4 Results (RMS error value) presented from 100 repeats of the Crowding PACO algorithms and the control algorithm (Crowding Genetic Algorithm) applied to the 35 Dimension problems with each algorithm allowed 100,000 solution evaluations per experimental run. . . . . 151

8.5 Figures illustrating possible problem compositions. . . . . 154

8.6 Results obtained by plotting the predicted (modelled) residual delays against the observed residual delays. . . . . 155

8.6 Results obtained by plotting the predicted (modelled) residual delays against the observed residual delays. . . . . 156

8.7 Figure indicating the performance of test and control algorithms on 35D, 28D and 7D variants of the STA-131AU and STA-131AV problems using a heuristic. . 159

8.8 Results (RMS error value) presented from 100 repeats of the variable search strategy versus the fixed search strategy applied to the 4×7D problem with each algorithm allowed 1,000,000 solution evaluations per experimental run. . . . . 161

# List of Tables

2.1	5 city TSP represented using 2D Euclidean coordinates . . . . .	6
2.2	5 city TSP edge weight matrix . . . . .	7
3.1	Research Papers published before the formalisation of ACO . . . . .	18
5.1	Example candidate solution 1 . . . . .	47
5.2	Example candidate solution 1 adjacency matrix . . . . .	47
5.3	Example candidate solution 2 . . . . .	47
5.4	Example candidate solution 2 adjacency matrix . . . . .	47
5.5	AND operation applied to solution 1 & 2 adjacency matrices . . . . .	47
6.1	Statistical comparison of FSPACO-TSP, CPACO-TSP, MMAS and PACO on various benchmark TSP. . . . .	58
6.2	Parameter settings used in testing of the TSP . . . . .	59
6.3	Path lengths of solutions contained in the population of CPACO-TSP (with a large crowding window size) after initialisation and the final population after 1000 iterations for Berlin52. . . . .	61
6.4	Path lengths of solutions contained in the population of CPACO-TSP (with a small crowding window size) after initialisation and the final population after 1000 iterations for Berlin52. . . . .	62
6.5	Path lengths of solutions contained in the population of CPACO-TSP (with a medium crowding window size) after initialisation and the final population after 10000 iterations for Eil101. . . . .	62
6.6	Parameter settings used for sensitivity analysis . . . . .	78
6.7	Effect of varying crowding size and history exponent on Shekel's foxholes . . . . .	85
6.8	Parameter settings used for quantitative analysis . . . . .	87
7.1	Algorithm parameter settings for Bi-objective TSP Problems . . . . .	104

LIST OF TABLES

7.2	Statistical comparison of PACO-MOFO and PACO-MO for various two-objective TSP . . . . .	109
7.3	Average final population size of PACO-MO versus static population size of CPACO-MOTSP for all test problems . . . . .	116
7.4	Algorithm parameter settings for MOFO problems . . . . .	124
7.5	Statistical comparison of PACO-MOFO and NSGA-II for MOP1, MOP2, MOP3, MOP4 and MOP6 defined in two dimensions . . . . .	124
7.6	Statistical comparison of PACO-MOFO and NSGA-II for MOP2 . . . . .	130
7.7	Statistical comparison of PACO-MOFO and NSGA-II for MOP4 . . . . .	136
7.8	Statistical comparison of PACO-MOFO and NSGA-II for MOP5 and MOP7 . . .	137
8.1	The default range of the search space per individual problem dimension for all datasets. . . . .	146
8.2	Quantitative comparison of the Random, CGA and CPACO algorithms applied to the STA-131AU and STA-131AV problems. . . . .	150
8.3	Parameter settings of test and control algorithms . . . . .	150
8.4	Statistical comparison of PACO-MOFO and NSGA-II for STA-V131AU and STA-V131AV . . . . .	152
8.5	Parameter settings of the multiple objective algorithms . . . . .	152
8.6	The average and standard deviation of the best multiple objective solutions when evaluated using the single objective cost routine. . . . .	153
8.7	Quantitative comparison of the 35D, 28D and 7D variants of the STA-131AU and STA-131AV problems. . . . .	155
8.8	The magnitude of the search space per individual problem dimension for the STA-131AU dataset after execution of the heuristic routine. . . . .	158
8.9	The magnitude of the search space per individual problem dimension for the STA-131AV dataset after execution of the heuristic routine. . . . .	158
8.10	Quantitative comparison of the 35D, 28D and 7D variants of the STA-131AU and STA-131AV problems using a heuristic. . . . .	158
8.11	Quantitative comparison of the fixed and variable search strategies applied to the STA-131AU and STA-131AV problems. . . . .	162
8.12	Positional errors obtained for the STA-131AV dataset using an alternative (but more complex) geodesy model. . . . .	163
8.13	Positional errors obtained for the STA-131AV dataset using the 4×7D problem decomposition, heuristic search space reduction and relocation technique from Sec. 8.4.2. . . . .	163

8.14 Magnitude of difference between the positional errors reported in Tab. 8.12 and Tab. 8.13. . . . .	163
10.1 Statistical significance of results for single objective TSP. . . . .	183
10.1 Statistical significance of results for single objective TSP. . . . .	184
10.2 Statistical significance of results for Schaffer's Function. . . . .	185
10.3 Statistical significance of results for Schwefel's Function. . . . .	186
10.4 Statistical significance of results for Himmelblau's Function. . . . .	187
10.5 Statistical significance of results for Branin's R-Cos Function. . . . .	188
10.6 Statistical significance of results for 35D Antenna Problems. . . . .	189
10.7 Statistical significance of results for 35D, 28D and 4×7D Antenna Problems. . . .	189
10.8 Statistical significance of results for 35D, 28D and 4×7D Antenna Problems using heuristics. . . . .	189





# Commonly Used Acronyms

ACO	Ant Colony Optimisation
ACOCD	Ant Colony Optimisation for Continuous Domains
ACS	Ant Colony Systems
AS	Ant Systems
CFO	Continuous Function Optimisation
CGA	Deterministic Crowding Genetic Algorithm
CPACO	Crowding Population-based Ant Colony Optimisation
DE	Differential Evolution
EC	Evolutionary Computation
EDA	Estimation of Distribution Algorithm
ES	Evolutionary Strategies
FSPACO	Fitness Sharing Population-based Ant Colony Optimisation
GA	Genetic Algorithm
GP	Genetic Programming
MMAS	<i>MAX</i> – <i>MZN</i> Ant Systems
MOEA	Multiple Objective Evolutionary Algorithm
MOFO	Multiple Objective Function Optimisation
MOGA	Multiple Objective Genetic Algorithm
MOO	Multiple Objective Optimisation
MOTSP	Multiple Objective Travelling Salesman Problem
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm (version 2)
PACO	Population-based Ant Colony Optimisation
PBIL	Population Based Incremental Learning
PCB	Printed Circuit Board
PMBGA	Probabilistic Model-based Genetic Algorithm
PSO	Particle Swarm Optimisation
QAP	Quadratic Assignment Problem
RBAS	Rank-based Ant Systems
RTS	Restricted Tournament Selection
RVGA	Real Value Genetic Algorithm
SA	Simulated Annealing
SNGA	Sequential Niching Genetic Algorithm
TSP	Travelling Salesman Problem
VEGA	Vector Evaluated Genetic Algorithm
VLBI	Very Long Baseline Interferometry