An Open Market for Trading Cloud Services

Sergei Chichin

Submitted in fulfilment of the requirements of the degree of Doctor of Philosophy

Faculty of Science, Engineering and Technology
Swinburne University of Technology
Australia

June 6, 2016
Abstract

The cloud computing model allows computing resources to be provided and accessed as services. That model has been widely accepted, and the increasing number of providers and consumers are forming a large-scale marketplace of cloud services. In such a marketplace, cloud providers aim to deliver their services in the most cost-effective way, while the consumers seek value for money and wish to procure the required resources in a fair market. Due to the dynamic and large-scale nature of the cloud, the efficient provision of cloud services and one that supports the objectives of the consumers and providers in cloud marketplaces is critical in realising the full benefits of cloud computing. Intelligent agent technology has proven useful in marketplace support and automation, and mechanisms based on economic principles show promise with respect to addressing the problem of efficient cloud resource provisioning, including its allocation and pricing. Therefore, the automation of cloud service provisioning can be realised in an agent-based cloud marketplace, where the primary resources, that is, cloud infrastructure, are offered in a similar way to the trading of other utilities, such as energy, between buyers and sellers in open markets.

This thesis investigates the problem of provisioning cloud infrastructure services, with a focus on an effective and efficient computational market mechanism design for a cloud marketplace. In order to determine a set of realistic requirements, a systematic analysis of the state of the current cloud market is conducted, and a conceptual framework is proposed. The four cases of marketplace provisioning of cloud infrastructure services are derived based on (i) the classification of the cloud resources, that is, homogeneous and heterogeneous, and (ii) the type of a cloud market economic platform, that is, single-sided and double-sided. From an efficiency perspective, the market mechanism’s design objective is to achieve the important economic properties that we categorise into (i) feasibility constraints, such as budget balance, individual rationality, and computational tractability, and the (ii) desirable properties, including allocative efficiency and incentive compatibility/truthfulness.

The thesis proposes four economics-inspired computational market mechanisms for
cloud service allocation and pricing to address each of the determined marketplace provisioning cases. Given the non-deterministic polynomial time (NP) complexity of the considered problem and the computational tractability requirement for the practical solutions, approximation mechanisms are designed and investigated. In particular, greedy mechanisms are proposed for cloud service allocation, and the buyer and seller pricing mechanisms are designed to derive prices based on critical value and proportional value, accordingly. The analysis of economic properties reveals that the proposed mechanisms maintain all the economic feasibility constraints, and some achieve truthfulness. The approximation quality of the proposed allocation schemes is analysed in extensive experiments, which show near-optimal results in the majority of simulated markets. Strategic manipulation in untruthful mechanisms is analysed in a variety of experimental scenarios, which reveal that only a relatively small strategic misreporting opportunity exists in some limited market scenarios.

In summary, this thesis makes original contributions to the knowledge in the area of computational market mechanism design for cloud marketplaces. To the best of our knowledge, we are the first to address the problem of infrastructure cloud services distribution in a set of practical market provisioning cases. The proposed allocation and pricing techniques for the markets of homogeneous services allow for reducing the computation complexity of the mechanisms, compared to the traditional techniques, while attaining near-optimal allocations, and achieving truthfulness. The designed market mechanisms for heterogeneous cloud services rely on a novel iterative approach that allows achieving better approximation quality than conventional approaches, while limiting strategic manipulation of the participants. The study is supported by a solid theoretical analysis, and proofs of the maintained economic properties, as well as an extensive simulation, intended to investigate the performance of the proposed mechanisms in different experimental settings.
Acknowledgements

The challenging and rewarding journey that I undertook in the past years of research could not be possible without the incredible people who provided their support and motivation.

First of all, I am indebted to my wonderful wife for who this PhD was as hard as for me: Johana, thank you for your love, encouragement, care, and strength that kept me going, I would not have completed it without you. I am also very thankful to my family for their support, affection, and empathy. I express my deepest regards to my parents, Natalia and Victor Chichin, who have always supported my decisions, as well as my brother and sister Sergey and Vasilisa, thank you for the many joys and love. I also would like to thank my parents in law, Brigitte and Jean-Luc Lambla for their help, support and affection.

I am especially grateful to my thesis supervisors Professor Ryszard Kowalczyk and A/Professor Bao Quoc Vo, who have provided a strong continuous support, guidance, and encouragement throughout the course of my PhD. Thank you for believing in me and giving me a chance to prove myself as a member of your research group. I am very grateful to Dr. Alan Colman for introducing me to my supervisors, which made a significant turn in my life. I also would like to thank all the members of the Intelligent Agent research group at Swinburne University and the staff of Faculty of Science Engineering and Technology. I would like to thank my dearest friends and colleagues with who I shared some insightful discussions. I wish to acknowledge Smart Services CRC, Swinburne University and Suncorp Bank for the provided financial support and research facilities.

I dedicate this thesis to my grandmother, Nina Tochalna, a generous and dedicated grandmother who gave me the love for Science.
Declaration

This is to certify that this thesis contains no material which has been accepted for the award of any other degree or diploma and that 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 thesis.

Sergei Chichin
March 1, 2015
List of Publications

1. **Sergei Chichin**, Bao Quoc Vo, Ryszard Kowalczyk: *Double-sided Market Mechanism for Trading Cloud Resources*, In the Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, Poland, 11-14 August 2014, pp. 198-205.

   Nominated for Best Student Paper Award in IEEE/WIC/ACM IAT 2015


   Winner of Best Student Paper Award in IEEE SCC 2015


   Winner of Student Paper Award in IEEE Victorian Section 2014

5. **Sergei Chichin**, Marcin Stepniak, Mohan Baruwal Chhetri, Bao Quoc Vo, Ryszard Kowalczyk: *Smart Cloud Marketplace - Agent-based Platform for Trading Cloud Services*, Accepted for publication In the Special Session on Agent-based Simulation Supporting Decision-making to be held in conjunction with The 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 388-395.

Accepted with minor revision on 04.04.2016
# Contents

1 Introduction .......................... 1
   1.1 Research Objectives and Research Questions ....................... 4
   1.2 Approach and Methodology ........................................... 4
   1.3 Research Requirements and Assumptions .............................. 6
   1.4 Research Contributions ............................................. 8
   1.5 Thesis Overview ................................................... 11
   1.6 Related Publications .............................................. 13

2 Background and Related Work ........ 15
   2.1 From Service Bureaus to the Fifth Utility of Cloud ................ 15
   2.2 Characterizing Cloud Services ..................................... 16
      2.2.1 Cloud Services Stack ........................................... 17
      2.2.2 Market of Infrastructure Cloud Services ...................... 18
   2.3 Market Mechanisms ................................................ 26
      2.3.1 Why Markets for Infrastructure Cloud Services? .............. 26
      2.3.2 Taxonomy of market mechanisms ................................. 27
      2.3.3 Auction Theory .................................................. 29
      2.3.4 Market-based Compute Resource Management Approaches ........ 39
   2.4 Agent-based Marketplaces ........................................... 42
      2.4.1 Why Agent-based Marketplace for Cloud Services? ............ 42
      2.4.2 Overview of Existing Agent-based Marketplaces ............... 43
   2.5 Summary ........................................................... 48

3 IaaS Market: Conceptual Framework and Problem Formalisation .... 49
   3.1 IaaS Cloud Marketplace Conceptual Framework ...................... 49
      3.1.1 Marketplace Participants ...................................... 49
      3.1.2 Cloud Services Market ......................................... 50
3.1.3 Provisioning Scenarios ........................................ 52
3.2 Problem Formalisation .......................................... 54
  3.2.1 Technical Requirements .................................... 54
  3.2.2 Market Model ................................................. 55
  3.2.3 General Cloud Infrastructure Allocation Problem ............ 58
3.3 Mechanism Design Framework .................................... 60
  3.3.1 Basic Aspects .................................................. 60
  3.3.2 Design Desiderata ............................................. 61
3.4 Approach Description ............................................ 64
  3.4.1 Allocation and Pricing Mechanisms .......................... 64
  3.4.2 Common Aspects of Evaluation Principles .................... 65
3.5 Summary .......................................................... 66

4 Market Mechanisms for Homogeneous Infrastructure Cloud Services 67
  4.1 Single-sided Market Mechanism ............................... 68
    4.1.1 Allocation Mechanism ....................................... 68
    4.1.2 Pricing Mechanism .......................................... 73
    4.1.3 Theoretical Investigation .................................. 78
    4.1.4 Experimental Investigation ................................. 83
  4.2 Double-sided Market Mechanism .............................. 93
    4.2.1 Allocation Mechanism ....................................... 93
    4.2.2 Pricing Mechanism .......................................... 99
    4.2.3 Theoretical Investigation .................................. 107
    4.2.4 Experimental Investigation ................................. 114
  4.3 Summary .......................................................... 125

5 Market Mechanisms for Heterogeneous Infrastructure Cloud Services 129
  5.1 Single-sided Market Mechanism ............................... 130
    5.1.1 Allocation Mechanism ....................................... 130
    5.1.2 Pricing Mechanism .......................................... 139
    5.1.3 Theoretical Investigation .................................. 144
    5.1.4 Experimental Investigation ................................. 149
  5.2 Double-sided Market Mechanism .............................. 158
    5.2.1 Allocation Mechanism ....................................... 158
    5.2.2 Pricing Mechanism .......................................... 165
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.3 Theoretical Investigation</td>
<td>173</td>
</tr>
<tr>
<td>5.2.4 Experimental Investigation</td>
<td>178</td>
</tr>
<tr>
<td>5.3 Summary</td>
<td>190</td>
</tr>
<tr>
<td>6 Smart Cloud Marketplace - A Platform for Trading Cloud Services</td>
<td>193</td>
</tr>
<tr>
<td>6.1 IaaS Marketplace Platform Requirements</td>
<td>194</td>
</tr>
<tr>
<td>6.2 Smart Cloud Marketplace Architecture</td>
<td>196</td>
</tr>
<tr>
<td>6.2.1 Facilitating Technology Choice</td>
<td>196</td>
</tr>
<tr>
<td>6.2.2 Reference Architecture</td>
<td>197</td>
</tr>
<tr>
<td>6.3 Use-case Scenarios &amp; Proof of Concept</td>
<td>201</td>
</tr>
<tr>
<td>6.3.1 Buyer Procurement Scenario</td>
<td>201</td>
</tr>
<tr>
<td>6.3.2 Seller Trading Scenario</td>
<td>204</td>
</tr>
<tr>
<td>6.4 Summary</td>
<td>207</td>
</tr>
<tr>
<td>7 Conclusions and Future Directions</td>
<td>209</td>
</tr>
<tr>
<td>7.1 Summary of Contributions</td>
<td>209</td>
</tr>
<tr>
<td>7.2 Answers to Research Questions</td>
<td>211</td>
</tr>
<tr>
<td>7.3 Possible Extensions and Open Questions</td>
<td>213</td>
</tr>
<tr>
<td>7.3.1 Real-world Data</td>
<td>213</td>
</tr>
<tr>
<td>7.3.2 Software Agents with Learning Capabilities</td>
<td>214</td>
</tr>
<tr>
<td>7.3.3 Irrational Behaviours</td>
<td>214</td>
</tr>
<tr>
<td>7.3.4 Heterogeneous Environments</td>
<td>215</td>
</tr>
<tr>
<td>7.3.5 Bayes-Nash Equilibria in Cloud Markets</td>
<td>215</td>
</tr>
<tr>
<td>7.3.6 Bidding Problem in a Distributed Cloud Marketplace</td>
<td>215</td>
</tr>
<tr>
<td>7.4 Final Remarks</td>
<td>216</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Cloud Marketplace Basic Concept ............................................. 2
1.2 Thesis Organisation and Main Contributions ................................. 12

2.1 Cloud Services Stack ............................................................... 17
2.2 Auction Mechanisms Characterisation .......................................... 30
2.3 Classic Auction Mechanisms Illustrated ...................................... 35

3.1 Conceptual Framework of Infrastructure Cloud Marketplace ............. 51
3.2 Examples of Cloud Service Offers and Requests ............................. 53

4.1 Insight into Parameter $m \geq 0$ (comparative sorting criteria value) .... 70
4.2 Closeness to Optimality in terms of Social Welfare (a, b, c) and Average Resource Utilisation (d, e, f). Representative Market Scenario: $N = 512, z = 0.0, p_1 = p_2 = p_3 = 0.25$. ................................................. 86
4.3 Impact of Resource Provisioning on Average Resource Utilisation (a, b, c, d) and Approximation Ratio (e, f, g, h). Market Scenario: mechanism with $m = 1.0, N = 512, z = 0.0$. ......................................................... 87
4.4 Resource Provisioning and Mechanism Parametrisation: Social Welfare Optimality (a) and Average Resource Utilisation Optimality (b). Market Scenario: $N = 512, z = 0.75$. ......................................................... 88
4.5 Computation Time of Market Allocation. Representative Market Scenario: $m = 1.0, z = 0.25, p = 0.5$. ......................................................... 89
4.7 Impact of Mechanism Parametrisation on Market Pricing (a), Average Resource Utilisation (b), and Buyer Fulfilment (c). Market Scenario: $N = 512, z = 0.25, p = 0.25$. ......................................................... 91
4.8 Computation Time of HO-SS Market Mechanism. Representative Market Scenario: \( m = 1.0, z = 0.25, p = 0.5 \). ................................. 92

4.9 Closeness to Optimality in terms of Social Welfare (a, c) and Average Resource Utilisation (b, d). Market Scenario for \( m = 0.25 \). ................................. 117

4.10 Impact of Mechanism Parametrisation on Social Welfare (a), Average Resource Utilisation (b) and Buyer Fulfilment (c): Closeness to Best Approximation Result. Market Scenario: \( C = 512, z = 0.0, p = 0.25 \). ................................. 118

4.11 Impact of Resource Provisioning on Average Resource Utilisation. Market Scenario: mechanism with \( m = 1.0, G = 512, M = 4, z = 0.0 \). ................................. 119

4.12 Resource Provisioning and Mechanism Parametrisation: Closeness to Best Approximation Result in terms of Social Welfare (a) and Average Resource Utilisation (b). Market Scenario: \( C = 512, z = 0.25, M = 4 \). ................................. 119

4.13 Computation Time of Market Allocation. Representative Market Scenario:
\( m = 1.0, z = 0.25, p = 0.5 \). ................................. 120

4.14 Seller Misreporting Opportunity in various market scenarios. Initial Market Scenario: \( \mu = 0.5, C = 512, M = 8, Z = 0.5, P = 1.0 \) ................................. 123

4.15 Seller Misreporting Opportunity in HO-DS Surplus Distribution (a, b, c, d) and K-pricing Mechanism (e, f, g, h) with different parametrisations. Market Scenario: \( G = 512, M = 8, Z = 0.5, P = 0.5 \) ................................. 124

5.1 Closeness to Optimality in terms of Social Welfare (a, b, e, f, i, j) and Average Resource Utilisation (c, d, g, h, k, l). Representative Market Scenario:
\( N = 512, z = 0.0, p = 0.25 \) ................................. 152

5.2 Social Welfare Improvement of HE-SS-Adaptive mechanism over HE-SS-Static in various Market Scenarios: \( l = 4.0, m = 0.25 \). ................................. 153

5.3 Resource Utilisation of various Allocation Mechanisms achieved across all market scenarios. ................................. 154

5.4 Computation Time of Market Allocation. Representative Market Scenario:
\( m = 1.0, z = 0.25, p = 0.25 \). ................................. 155

5.5 Seller Power over Market Pricing Outcomes: Social Welfare Distribution. Market Scenario: \( m = 1.0, l = 1.0, N = 512 \). ................................. 156

5.6 Impact of Mechanism Parametrisation on Market Pricing (a), Average Resource Utilisation (b), and Buyer Fulfilment (c). Market Scenario: \( m = 1.0, l = 1.0, N = 512, z = 0.25, p = 0.25 \). ................................. 156
   Representative Market Scenario: \( m = 1.0, l = 1.0, z = 0.25, p = 0.25 \).
5.8 Closeness to Optimality in terms of Social Welfare (a, c) and Average Resource Utilisation (b, d).
   Market Scenario for \( l = 2.0, m = 0.25 \).
5.9 Closeness to the Best Approximation Result in terms of Social Welfare (a, b, e, f, i, j) and Average Resource Utilisation (c, d, g, h, k, l).
   Representative Market Scenario: \( C = 512, M = 4, z = 0.0, p = 0.25 \).
5.10 Resource Utilisation of various Allocation Mechanisms achieved across all market scenarios with optimal result.
5.11 Computation Time of Market Allocation. Representative Market Scenario:
   \( z = 0.25, p = 0.5 \).
5.12 Buyer Misreporting Opportunity. Market Scenario: mechanism with \( m = 1.0, l = 1.0, C = 256, M = 8, z = 0.5, p = 1.0 \).
5.13 Seller Misreporting Opportunity in various market scenarios. Initial Market Scenario: \( \mu = 0.5, C = 256, M = 8, Z = 0.5, P = 1.0 \).
5.14 Seller Misreporting Opportunity in HE-DS Surplus Distribution (a, b, c, d) and K-pricing Mechanism (e, f, g, h) with different parametrisations.
   Market Scenario: \( C = 256, M = 4, Z = 0.5, P = 1.0 \).

6.1 Smart Cloud Marketplace Mobile: Sample Screenshots
6.2 Smart Cloud Marketplace: Reference Architecture
6.3 Smart Cloud Marketplace: Buyer Procurement Scenario
6.4 Smart Cloud Marketplace: Custom Buyer Recommender Agent
6.5 Smart Cloud Marketplace: Seller Trading Scenario
## List of Tables

2.1 IaaS Cloud Providers in Industry (brief characterization) .......................... 19
2.2 Infrastructure Cloud Service Types .......................................................... 20
2.3 IaaS Pricing Models ................................................................................... 21
2.4 Double-sided Economic Platforms for Trading Infrastructure Cloud Services
    in Industry .................................................................................................. 24
2.5 Characteristics of Market Mechanisms ......................................................... 27
2.6 Truthful Combinatorial Auctions Illustrated ................................................ 38
2.7 Summary of Related Market-based Approaches in Literature ....................... 41
2.8 Brief Characterisation of Existing Agent-based Marketplaces ..................... 47
3.1 CIAP Problem Notation ............................................................................... 56
3.2 Mechanism Design Framework Notations ................................................... 61
3.3 Problem Use-case Scenarios ....................................................................... 64
4.1 HO-SS Mechanism Example: Market Input .................................................. 72
4.2 HO-SS Mechanism Example: Allocation Procedure ($m = 1.0$) ............... 73
4.3 HO-SS Mechanism Example: Pricing Procedure ($m = 1.0$) ....................... 77
4.4 HO-SS Experimental Setting ....................................................................... 84
4.5 HO-DS Mechanism Example: Market Input ................................................ 97
4.6 HO-DS Mechanism Example: Allocation Procedure ($m = 1.0$) ............... 98
4.7 HO-DS Mechanism Example: Buyer Pricing Procedure ($m = 1.0$) .......... 103
4.8 HO-DS Mechanism Example: Seller Pricing Procedure ............................ 106
4.9 HO-DS Experimental Setting ....................................................................... 115
4.10 Market Mechanisms for Homogeneous Cloud Services: Summary .......... 126
5.1 Insight into parameters $l > 0$ and $m > 0$ (comparative sorting criteria value
    analysis) ....................................................................................................... 135
5.2 HE-SS Mechanism Example: Market Input .................................................. 136
5.3 HE-SS Mechanism Example: Allocation Procedure \((m = 1, l = 1)\) . . . . . . 137
5.4 HE-SS Mechanism Example: Pricing Procedure \((m = 1, l = 1)\) . . . . . . 143
5.5 HE-SS Experimental Setting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
5.6 HE-DS Mechanism Example: Market Input . . . . . . . . . . . . . . . . . . 163
5.7 HE-DS Mechanism Example: Allocation Procedure \((m = 1, l = 1)\) . . . . . . 164
5.8 HE-DS Mechanism Example: Buyer Pricing Procedure \((m = 1, l = 1)\) . . . 169
5.8 HE-DS Mechanism Example: Buyer Pricing Procedure \((m = 1, l = 1)\) . . . 170
5.9 HE-DS Mechanism Example: Seller Pricing Procedure . . . . . . . . . . . . 173
5.10 HE-DS Mechanism: Seller Individual Rationality vs. Buyer Truthfulness . . 174
5.11 HE-DS Experimental Setting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 179
5.12 Market Mechanisms for Heterogeneous Cloud Services: Summary . . . . . 191
Chapter 1

Introduction

Development of web technologies and maturing virtualization techniques have shifted the way businesses and individuals provide and consume the compute power. The novel and increasingly popular cloud computing model allows for providing and accessing the computing resources as services in a convenient any-time on-demand fashion. In cloud computing environments there are two main stakeholders, which are (i) the Cloud Provider who manages and delivers the services, and (ii) the Cloud Consumer who requests and utilises the services.

Many cloud providers, such as Amazon EC2, Google Compute Engine, Rackspace Cloud, Windows Azure, and others, supply their customers with the primary cloud resources, i.e. cloud infrastructure. They offer the hardware and software services that power it all according to the infrastructure as a service (IaaS) model, where the cloud consumer gains full control over application, platform, and infrastructure levels [142, 13]. The IaaS model issues the virtual access to a remote pool of shared computing infrastructure, which is typically delivered in a form of virtual machines (VMs), characterised by a number of low-level attributes, such as the number of processing cores (CPU), amount of memory (RAM), storage capacity (HDD), networks, etc.

The numerous benefits of using cloud computing, such as elasticity, high accessibility, and flexible pay-as-you-go pricing, have resulted in an increasing interest among consumers leading to a significant rise of demand for cloud services [70]. In order to address the consumer needs, a number of cloud service providers has grown significantly. A wide general acceptance of cloud computing and the growing number of providers and consumers are forming a large-scale marketplace of cloud services. Due to the elastic and scalable nature of the cloud, such a marketplace is a dynamic, rapidly changing environment where a lot of participants provide and consume the infrastructure cloud services. In such a marketplace,
the cloud providers aim to provision their services in the most cost-effective way. The cloud consumers, in turn, seek value for money and wish to procure services in a fair market, where the selfish participants cannot benefit at the cost of others. Therefore, in order to realise full benefits of cloud computing, it is critical that the cloud market determines an efficient distribution of services and limits the harmful manipulation of the self-interested strategic participants. Given the large-scale and dynamic nature of the cloud market environment, it is a difficult problem that can be efficiently addressed only in an automated way. Computational market mechanisms based on economic principles and intelligent agent technology have proven useful in automating marketplaces [25, 34] and some predictions have been made towards a wide adoption of intelligent software agents in the future electronic markets [114]. Therefore, the automation of infrastructure cloud services provisioning could be realized in an agent-based cloud marketplace, where the primary resources, i.e. cloud infrastructure, are offered in a similar way to trading other utilities, such as electricity or gas. We illustrate the basic concept of a cloud marketplace in Figure 1.1.

The mechanisms based on economic concepts, such as auctions and commodities markets show promise for addressing the problem of computing resources allocation and pricing. There are several advantages when employing the economic models for the market of cloud services, which include:

- **Improved social welfare** and **enhanced cloud resource utilization**, achieved due to efficient distribution of the cloud resources based on market competition, when the traded goods are granted to the market participants who value them the most.

- **Dynamic pricing** which is based on the laws of supply and demand that reflect the

![Figure 1.1: Cloud Marketplace Basic Concept](image-url)
rapidly changing market conditions in a dynamic cloud marketplace environment.

- Limited strategic manipulation which establishes a healthier and more efficient ecosystem by aligning the market participants’ dominant strategy with the market’s objective. It also has a positive impact on the overall efficiency of the market and the social welfare.

There are two essential design perspectives when approaching the problem of computational market mechanisms design for infrastructure cloud services: achieving effectiveness (technical factors) and efficiency (economic factors). Effectiveness implies that the designed mechanisms are practical and can be applied to address the real-world cloud market scenarios. It includes the characteristics of the traded resources, i.e. VM classification, as well as the economic platform types, commonly used in industry. Efficiency typically relates to the market design from a theoretical perspective, which is characterized by the important economic properties targeted by the mechanism design. It involves (i) the design feasibility constraints, i.e. the essential economic properties that must be achieved (Budget Balance, Individual Rationality, and Computational Tractability), as well as (ii) the desirable properties that need to be approximated if cannot be guaranteed by the mechanism design (Allocative Efficiency and Incentive Compatibility/Truthfulness).

The numerous studies of economic models used in different markets for trading utilities, such as electricity markets, or grid computing, reveal that they are not suitable for today’s market of cloud infrastructure due to a complex nature of the infrastructure cloud services [20, 119, 82]. Despite the growing interests in academia and industry, the attempts to offer effective market mechanisms for efficient infrastructure cloud services trading have not yet been elaborate enough to address the problem. The conducted research has either made the simplified assumptions about the cloud services characteristics, or disregarded the important economic properties of the mechanism design. Hence, there is a need for investigation and development of more effective and efficient market mechanisms for trading infrastructure cloud services.

Therefore, this thesis investigates the problem of infrastructure cloud services allocation and pricing with a focus on an effective and efficient computational mechanism design for cloud marketplace. We intend to come up with an effective mechanism design framework in order to ensure the practical feasibility of the proposed solutions. The economic properties of the computational mechanism design is an important consideration of our study. Apart from achieving the economic feasibility constraints, the objective is to design the market mechanisms that attain the most socially efficient allocation of services and
Chapter 1. Introduction

restrain the strategic manipulation of self-interested participants. We aim to conduct a proper theoretical analysis of the designed mechanisms and to prove that they are economically efficient, as well as to study the strengths and limitations of the proposed solutions by simulating various market trading scenarios.

1.1 Research Objectives and Research Questions

The aim of this thesis is to investigate and develop effective computational mechanisms for efficient trading of infrastructure cloud services in a cloud marketplace composed of self-interested strategic participants.

The main research questions that this thesis aims to answer are:

- Can we achieve the desirable economic properties, including allocative efficiency, individual rationality, budget balance, and incentive compatibility, in a market for trading infrastructure cloud services?
- Can these economic properties in such a cloud market be achieved within a tractable computational time?
- How to design a tractable mechanism for a cloud market that achieves the desirable economic properties, attains the most socially efficient distribution of cloud services, and limits the strategic manipulation of its participants?

1.2 Approach and Methodology

This thesis considers the problem of infrastructure cloud resource provisioning in economic terms and proposes a number of market and economics-inspired mechanisms for effective and efficient cloud service delivery. It investigates a cloud marketplace, which is a distributed open and dynamic environment, populated by self-interested agents that will deviate from the truthful behaviour and apply strategic manipulation, if it can allow them to improve their individual outcomes.

We approach the problem from economic perspective for a number of reasons. Firstly, most economic models are well-studied and the existing knowledge can be used for further design and analysis. The market mechanisms are also proven to be highly effective for dynamic large-scale environments due to the efficient allocation of goods and the ability to adapt the prices based on supply and demand principles[146]. A healthier and more efficient market ecosystem can be established by a well-designed mechanism that eliminates or
limits the strategic manipulation of its self-interested participants [107, 36]. Furthermore, market-based resource pricing ensures that the resources are consumed only when needed and provides incentives to their users to adapt their demand to offpeak periods (when the prices are lower) as well as to return the idle resources back to the market in exchange for the market price [131], which has a positive impact on the market dynamics. Finally, the market mechanisms are known to be trusted by the traders due to the well-understood and transparent rules and regulations that they provide [17].

The general area addressed in this thesis is that of computational market mechanism design [107, 46], which is a multidisciplinary research at the interface between mechanism design [105] and computer science [140]. Example domains vary from computational applications (e.g. market-based resource allocation, such as grid computing or wireless spectrum) to e-commerce (e.g. Internet auctions, electronic markets). The mechanism design deals with problems in which multiple selfish agents are to be organised in the way that the global outcome meets socially desired goals, such as Truthfulness, Pareto Optimality, Individual Rationality, etc. The computer science, in turn, investigates computational and communication complexity of real-time and large-scale search algorithms design [110, 107, 49]. Specifically, in this thesis we deal with such related research areas as auction design [105], combinatorial optimisation [44], computational complexity theory [121], intelligent agents and multiagent systems [149, 56].

The evaluation and analysis of the proposed computational market mechanisms for cloud services allocation and pricing is carried out through the following methodologies:

- **Literature Review:** In order to gather a set of practical requirements for this research, a systematic literature review and market analysis were conducted, so that to come up with a conceptual framework to approach the problem in an effective way. It includes (i) the classification of cloud resources and cloud markets, as well as (ii) the critical analysis and evaluation of the existing methods presented in literature.

- **Theoretical Investigation:** The economic properties of the designed computational market mechanisms are analysed theoretically and proofs of the maintained properties are derived based on mathematical reasoning. The following important economic properties are studied theoretically: Truthfulness, Individual Rationality, Budget Balance, and Computational Complexity.

- **Experimental Simulation:** The performance and dynamics of the proposed market mechanisms are examined experimentally. Since the cloud providers do not publicly
reveal the information about the cloud services requests received from their customers, we use large-scale extensive simulation approach, where the mechanisms are evaluated in a variety of market scenarios. In particular, we investigate the allocation quality of the designed mechanisms, the seller’s power over the prices in the market and the strategic misreporting opportunity in the proposed market mechanisms.

- **Prototyping**: We also validate/demonstrate the proposed mechanisms by implementing a proof-of-concept prototype of an agent-based marketplace of cloud infrastructure services. The developed open platform implements the proposed market mechanisms and allows the distributed users to engage in market activities, such as create markets for trading cloud services as well as buy and sell resources themselves or by using custom-designed intelligent agents.

### 1.3 Research Requirements and Assumptions

In this thesis, we consider an open marketplace for trading infrastructure cloud services, which consists of three building blocks: (i) the market, (ii) the traded goods, and (iii) the market participants. We outline the research requirements and assumptions related to each of these components below:

**Market** provides a medium for trading and guides the process of infrastructure cloud services provisioning. In this work we consider two market types/models which clear the market on a periodic basis.

- **Market-based Resource Allocation and Pricing.** Markets allow to quickly achieve coordinated market decisions in large-scale dynamic environments, like cloud marketplace. Furthermore, due to a complete information about the supply and demand, a centralised market is said to achieve more socially efficient outcomes compared to a decentralised market [40].

- **Market models/types Economic Platform.** In the real-world cloud markets, the cloud providers typically offer their infrastructure cloud services to their consumers directly in a single-provider market. Although such an approach is more common nowadays, the double-sided markets begin to emerge, aiming to establish a fully competitive and more efficient cloud market ecosystem. In this thesis, we follow the real-world examples and consider two market types, which include single-sided direct market and double-sided market.
Chapter 1. Introduction

- **Periodic Clearance.** In a periodic (discrete time) market, the clearance occurs at some pre-defined time after the market opening. In other words, there is a time period when the market collects the participants’ requests and, once the market expires, the allocation and pricing outcomes are determined based on the collected orders. The periodic markets are said to be more efficient compared to their continuous counterparts due to the decisions being made based on more complete information about supply and demand, rather than considering requests for allocation at their individual arrival times [112].

**Goods** are the infrastructure cloud services that need to be distributed among the participants and priced according to the supply and demand principles. In this thesis, we consider two types of traded infrastructure cloud services.

- **Infrastructure Cloud Service Classification.** In today’s cloud market, the infrastructure cloud resources are delivered to the consumers in a form of virtual machines, which represent compound goods with a number of characteristics. The cloud providers either allow constructing complex cloud infrastructures by offering (i) pre-defined VMs of different sizes (i.e. *homogeneous type*), where the VMs’ configurations are fixed by the cloud provider (e.g. micro, small, medium, large instances), or deliver (ii) custom user-defined VMs (i.e. *heterogeneous type*), where the characteristics of underlying low-level resources (e.g. CPU, RAM, HDD) are specified by the consumers based on their individual needs. In this research, we address the problem for both types of infrastructure cloud services, i.e. homogeneous and heterogeneous.

**Participants** join the market and engage in trading activities in order to access or provision services. The market participants can either be the services consumers or providers who buy or sell the infrastructure cloud services, respectively. They are self-interested agents who know their preferences and aim to improve their individual outcomes.

- **Needs-awareness.** We assume that the market participants are aware of their market needs and preferences. In particular, the consumers of cloud resources know their infrastructure cloud service requirements and the budget constraints; and the cloud providers know the reservation prices for their traded services. The needs identification and user preferences elicitation is out of scope of this thesis.

- **Self-interested participants.** The consumers and providers are self-interested and strategic agents with quasi-linear utility functions in price values. They will deviate from a truthful strategy and manipulate the market, if it can improve their individual
outcomes. The goal of the mechanism design is to give incentives to the market participants to act truthfully.

- **Participant Preferences.** Similar to the real-world consumers with minimum requirements to the cloud infrastructure, we consider buyers with strict preferences over the resources they request, called single-minded. The needs of such buyers have to be fully satisfied (indivisible buyer requests), otherwise a less powerful resource is of no use due to its inability to fulfil the minimum requirements. This phenomenon is commonly referred in economics literature as inelastic demand [74]. Sellers are assumed also to have quasilinear preferences for the traded resources. However, they make divisible offers that can be used to provision services to multiple cloud consumers (demand-side aggregation). The seller aims to provision as many resources as possible, subject to resource availability and minimum price constraint.

### 1.4 Research Contributions

This thesis makes the original contributions to the knowledge in the areas of computational market mechanism design, combinatorial optimization, agent-based markets, and cloud computing. A summary of the main contributions of this thesis are provided below:

**Conceptual framework and problem formulation for infrastructure cloud services allocation and pricing**

A systematic analysis of the current and future cloud market is conducted in order to determine a set of effective requirements and establish a systematic view of the problem. A conceptual framework to approach the problem is proposed, which is based on classification of infrastructure cloud resources (homogeneous and heterogeneous VMs), and the type of cloud market type/model (single-sided and double-sided). A generic problem of infrastructure cloud services allocation and pricing is formulated as Integer Program, the corresponding problem complexity class is identified and the four cases of marketplace cloud services provisioning are specified.

**Computational market mechanism design for homogeneous cloud services provisioning**

Two computational market mechanisms for homogeneous cloud services allocation and pricing in single-sided and double-sided markets are designed and evaluated.
Chapter 1. Introduction

- **Single-sided market:** A combinatorial greedy allocation mechanism is proposed for allocation of homogeneous cloud services of a single cloud provider/seller among multiple buyers. The corresponding pricing mechanism based on the critical-value is designed to determine the prices that the buyers pay in the market. The economic properties of the proposed allocation and pricing mechanisms are analysed theoretically, and the corresponding theorems and proofs are presented. The allocation quality of the proposed approximation mechanism, and the seller’s power over the market pricing outcomes are analysed in the extensive simulation experiments.

The proposed mechanism operates in a realistic setting, where the seller can expresses the reserve price constraints for the traded cloud resources. Furthermore, thanks to the proposed approximation technique that maintains a sorted list of bids, the proposed mechanism is more time-efficient (lower time complexity), compared to the state-of-the-art approaches presented in literature.

- **Double-sided market:** A combinatorial greedy allocation mechanism is proposed for resource allocation between multiple buyer and sellers in a double-sided market environment. The pricing mechanism based on critical-value is designed to determine the prices that the buyers pay. The seller pricing mechanism based on market surplus distribution is proposed for seller payments determination. The economic properties of the proposed mechanisms are analysed theoretically, and proofs are provided for budget balance, buyer and seller individual rationality, and buyer truthfulness. The approximation quality of the proposed allocation mechanism and the seller’s strategic manipulation opportunity are analysed in extensive simulation experiments.

To the best of our knowledge, it is the first work to address the problem of cloud services allocation and pricing in a combinatorial double-sided market with indivisible buyer requests. Apart from achieving the economic efficiency, our mechanism design demonstrates an impressive performance in terms of computation time. We also identify a drawback of the proposed greedy allocation heuristic, which we address in our further mechanisms designed for heterogeneous services.

**Computational market mechanism design for heterogeneous cloud services provisioning**

We have designed and evaluated two mechanisms based on economic principles for trading heterogeneous infrastructure cloud services in single-sided and double-sided markets.
• **Single-sided market:** A novel iterative allocation mechanism based on greedy heuristic is proposed for allocation of resources of a single seller among multiple buyers, participating in the market. The corresponding iterative pricing mechanism for buyer price determination based on the critical-value is designed. The economic properties of the proposed mechanisms are analysed theoretically, and the theorems for budget balance, buyer and seller individual rationality, and truthfulness are provided. The quality of allocations made by the proposed mechanism are compared with the optimal results, and the seller’s power over market allocation and pricing decisions are analysed in extensive experiments.

The proposed iterative approach achieves a significant improvement in terms of the approximation quality compared to the conventional one-shot approach. Furthermore, the new iterative pricing mechanism guarantees truthful market outcomes.

• **Double-sided market:** The problem of combinatorial double-sided market allocation and pricing is addressed by an iterative double-sided greedy mechanism. The proposed allocation scheme operates based on greedy selection of candidates in various market iterations. The buyer pricing mechanism is designed to derive payments based on critical-value, subject to seller’s individual rationality constraint. The seller payments are determined by the market surplus distribution based on various mixed rules. The impossibility to achieve seller individual rationality and buyer’s truthfulness in a single design is demonstrated. The Individual Rationality and Budget Balance properties are proven to hold in the proposed design. The quality of approximated allocation, and the buyer and seller strategic misreporting opportunity are analysed in extensive simulation experiments.

The problem of cloud services allocation and pricing has not been addressed for the combinatorial double-sided market of heterogeneous services. We pioneer this field with an effective and efficient mechanism design and identification of the potential future design improvements.

**Proof-of-concept prototype of an agent-based platform for infrastructure cloud services provisioning**

A prototype of an agent-based marketplace platform is designed and developed. The proof-of-concept allows remote users to join the system and trade the cloud services, where the trading process is facilitated by our proposed market mechanisms. The reference architec-
ture and the practical use-case scenarios are presented to demonstrate the advantages of using computation market mechanisms and software agents for cloud services trading.

1.5 Thesis Overview

The main contributions of this thesis in relation to its structure are depicted in Figure 1.2. The organization of the thesis is outlined below:

Chapter 2 presents the background of the main areas related to this thesis, and overviews the related works that address the problems similar to the one considered in this thesis. In particular, this chapter provides a background and literature survey on the cloud computing, market mechanisms, market-based computing resource management, and the agent-based marketplaces.

Chapter 3 summarises and formalises the problem of cloud infrastructure allocation and pricing. Specifically, the chapter provides a structured view to approach the research problem, provides the problem mathematical formalisation, and outlines the market design framework together with the major design principles applied in the thesis.

Chapter 4 investigates the problem of homogeneous cloud services allocation and pricing in the markets populated by self-interested strategic participants. The chapter is split into two parts: mechanism designs for a single-sided and a double-sided markets. For both problem settings, the market mechanism designs are proposed and the mechanisms’ operation is illustrated through simple examples. The rigorous theoretical and extensive experimental evaluation of the designed mechanisms are presented and the results are summarised.

Chapter 5 tackles a heterogeneous cloud services trading problem in the single-sided and double-sided markets populated by strategic participants. The market mechanisms for both problems are presented and investigated. The market mechanism economic properties are theoretically studied, and the performance of the proposed mechanisms is evaluated in various simulation scenarios.

Chapter 6 presents the proof of concept of an open agent-based market for trading cloud services. It outlines the important requirements for an agent-based cloud marketplace, provides the reference architecture, and some illustrative use-case scenarios of the implemented Smart Cloud Marketplace platform.

Chapter 7 concludes the thesis by summarising the presented results and their contributions to the knowledge, answering the research questions, and highlighting the possible extensions and future work.
Figure 1.2: Thesis Organisation and Main Contributions
1.6 Related Publications

The results presented in this thesis have been published and presented in five international conference papers and a journal article. We outline the published manuscripts according to the organization of this thesis. Some of the content of Chapter 4, is derived from the following papers:

- *11th IEEE International Conference on Services Computing (SCC 2014), Anchorage, Alaska, USA [36]*. The paper proposed the design of a single-sided computational market mechanism, and received the Best Student Paper Award from IEEE Victorian Section in December 2014.

- *2015 IEEE/VIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Singapore [37]*. The work focused on a double-sided market mechanism design for homogeneous infrastructure cloud services, and was nominated for the Best Student Paper award in IAT 2015.

A part of Chapter 5 materials, that study the heterogeneous cloud services trading in a marketplace, are derived from the following papers:

- *7th International Conference on Cloud Computing (CLOUD 2014), Anchorage, Alaska, USA [35]*. The conference paper proposed a single-sided market mechanism for heterogeneous cloud services trading based on a novel iterative approach.

- *12th IEEE International Conference on Services Computing (SCC 2015), New York, USA [38]*. The paper investigated the cloud services allocation techniques in a double-sided economic platform; the presented work was granted the Best Student Paper Award from SCC Conference in July 2015.


The work in Chapter 6 is partially derived from:

- *2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Warsaw, Poland [34]*. The conference paper proposed a proof of concept and a prototype implementation of an agent-based marketplace for trading cloud services.

The goal of this thesis is to present a sound, coherent, structured overview of the undertaken research, to describe the original contributions to the knowledge and to point the areas of future research.
Chapter 1. Introduction
Chapter 2

Background and Related Work

In this chapter, we provide the background information for the research problem addressed in this thesis. Specifically, we conduct a systematic analysis of nowadays market of cloud infrastructure, including the types of traded services, commonly used pricing models, and the emerging cloud exchange platforms. We motivate the use of market mechanisms, and auctions in particular, give an overview of auction theory with examples of the existing auction design approaches. Finally, we overview the most relevant market-based approaches for computing resource management found in literature as well as the proposed agent-based marketplace platforms.

2.1 From Service Bureaus to the Fifth Utility of Cloud

Cloud computing relies on the concept of delivering computing services through a network. Cloud services are delivered on-demand which allows the companies and individuals to store and process their data in third-party data centres. Despite today’s rapidly growing popularity and wide general acceptance, cloud computing has a 60-year long history, which took it from service bureaus to its modern form of the fifth utility.

The idea dates mid-sixtieth, when a computer scientist John McCarthy proposed a vision of computation delivered as a public utility, which relied on his time-sharing theory [43]. It evolved in a form of service bureaus which allowed the companies to share the resources of very expensive computation machines. In 1969, J.C.R. Licklider, responsible for Advanced Research Projects Agency Network (ARPANET), proposed the vision of interconnected computer network which permits everyone to access programs and data from anywhere [125]. The networking system project, initiated by U.S. Department of Defence, established the global network development, which conceived Internet.
The *Internet era* began in 1995, and the World Wide Web, that started as static screenfuls loaded into a browser window, rapidly evolved into a form of interactive dynamic pages. Such advancement in web technologies permitted the companies to deliver enterprise applications via simple website interface. The first company to provide such software service over the Internet was Salesforce.com [122]. During Internet era, the world got more interconnected, but the increased computational capacity and reduced size of devices diminished the need to outsource compute power.

The idea of cloud computing returned when the *Mobile era* began. The devices needed more resources to store and process the rapidly generated amounts of data. In 2002, Amazon Web Services provided a suite of cloud services, which included storage, computation, and human intelligence through the Amazon Mechanical Tuck [7]. The maturing virtualization technology and the development of universal high-speed bandwidth allowed Amazon to launched Elastic Compute Cloud (EC2) in 2006 [9], which aimed to support individuals, small and medium enterprises by providing access to Amazon’s cloud infrastructure. The development of web technologies, and establishment of universal interoperability standards allowed the companies to produce reliable and easy to consume browser-based enterprise applications. Therefore, in 2009, Google and Microsoft started to offer various cloud applications that could be delivered on demand. It offered ultimate flexibility for the outsourcing companies and rapidly gained popularity among businesses and individuals.

We are now entering the *Internet of Things (IoT) era*, where the number of interconnected devices is exponentially increasing [134]. As a result, the demand for compute power, which is offered as a public utility, is going to explode, so does the number of datacentres and the cloud providers. Hence, the market of cloud services is going to become more dynamic and large-scale environment, and the efficient cloud services provision will turn into a more and more substantial problem, which has to be addressed.

### 2.2 Characterizing Cloud Services

Cloud computing is a general term to refer to anything that involves delivering of computing services over the Internet. Its objective is to allow a convenient any-time on-demand access to a remote pool of shared computational resources in a similar way to the delivery of public utilities, such as electricity or water [24]. In this section, we describe some general aspects of cloud computing, and provide a more detailed description of infrastructure cloud market, including the types of offered services, the applied pricing models, and the emerging platforms for trading cloud infrastructure.
2.2.1 Cloud Services Stack

In cloud computing environments there are two main stakeholders, which are the Cloud Providers who manage the datacenters and deliver the services, and the Cloud Consumers who request and utilize the services. Cloud providers deliver their services to the consumers according to different models, which follow the "anything as a service" principle of service-oriented architecture (SOA) [142]. The delivery models for the cloud services form the cloud services stack that typically consists of: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service [118]. Please, refer to Figure 2.1 for illustration.

We briefly outline each of the delivery models together with the level of control that the cloud consumers gain over the provisioned service.

- **Software as a Service (SaaS)** is a delivery model for the software and associated data which is centrally hosted on the cloud. The SaaS allows the users to interact with the software application only, and the consumer’s control is limited to the application’s functionality. It is a fairly popular model and such providers as Google Apps [67], Apple App Store [12], Salesforce [122] are offering cloud software.

- **Platform as a Service (PaaS)** provides a platform for application deployment. A consumer can manage the application installation and configuration settings, but has no control over the infrastructure, where the application is running. The Google App Engine [68], AWS Elastic Beanstalk [8], Cloud Foundry [41] are the typical PaaS providers.

![Figure 2.1: Cloud Services Stack](image)
Chapter 2. Background and Related Work

- *Infrastructure as a Service (IaaS)* offers its consumers a full control over application, platform and infrastructure levels. Such third-party providers as Amazon EC2 [6], Microsoft Azure [98], Rackspace Cloud [116] host various infrastructure components (e.g. hardware, software, servers, storage, networks, and other) on behalf of the user, delivered in a form of IaaS.

The IaaS model provides the primary cloud resource (i.e. the software and hardware that power it all), which can be offered to the consumers in a similar way to trading other utilities, such as electricity or gas. In this thesis we investigate the delivery and acquisition of infrastructure cloud services, and in the remainder of this chapter we focus on IaaS provisioning model in particular.

2.2.2 Market of Infrastructure Cloud Services

We conduct a systematic analysis of the current state of the cloud infrastructure market. We analyse more than a dozen of the most notable infrastructure cloud service providers in industry and provide a classification of the offered infrastructure cloud services and the pricing models, commonly applied for the services provisioning. Finally, we discuss some emerging electronic platforms for infrastructure cloud services trading. A quick summary of the reviewed providers and their characteristics is given in Table 2.1.

Offered Infrastructure Cloud Services

The infrastructure cloud service providers own physical equipment, and they are responsible for housing, running and maintaining it. The low-level resources, such as CPU, RAM, HDD are provisioned to the customers in a form of Virtual Machines (VMs). The infrastructure cloud providers commonly offer pre-defined or custom VM configurations. We characterise and provide examples for both VM types below:

- *Pre-defined VMs:* A cloud provider offers various groups of VMs depending on the consumption purpose, such as general, compute optimised, memory or storage optimised. The configurations of VMs in each group are pre-defined by the cloud provider depending on the consumption purpose; for example, memory optimised VMs are built in the way that the instances are better-suited for running memory-intensive applications. However, a common characteristic for all the VM types in a particular group is that the underlying resources in the VMs of different sizes scale according to a strict relationship.
In order to illustrate the concept of pre-defined VMs, we consider an example of general purpose VMs offered by Rackspace Cloud [116] with Windows OS, depicted in Table 2.2a. We can see that there is a strict relationship according to which the amounts of CPUs, memory (RAM) and storage (HDD) scale in the VMs of different sizes. Each pre-defined VM has its associated price which is set by the cloud provider. The consumers select the VM configurations that correspond to their needs and pay
Chapter 2. Background and Related Work

Table 2.2: Infrastructure Cloud Service Types

(a) Pre-defined VMs (Rackspace)

<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>CPU (cores)</th>
<th>RAM (GB)</th>
<th>HDD (GB)</th>
<th>Service Pricing</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Small</td>
<td>2</td>
<td>2</td>
<td>40</td>
<td>$0.08</td>
</tr>
<tr>
<td>General Medium</td>
<td>4</td>
<td>4</td>
<td>80</td>
<td>$0.16</td>
</tr>
<tr>
<td>General Large</td>
<td>8</td>
<td>8</td>
<td>160</td>
<td>$0.32</td>
</tr>
</tbody>
</table>

(b) Custom VMs (CloudSigma)

<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>CPU (cores)</th>
<th>RAM (GB)</th>
<th>HDD (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose</td>
<td>2</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>Compute Intensive</td>
<td>2</td>
<td>3.75</td>
<td>60</td>
</tr>
<tr>
<td>Memory Intensive</td>
<td>2</td>
<td>15</td>
<td>40</td>
</tr>
</tbody>
</table>

| Resource Pricing (per unit) | $0.030 | $0.023 | $0.0001 |

the posted price per hour of usage. This type of infrastructure cloud services is offered by such cloud providers as Amazon Web Services [6], Atlantic.net [15], Google Compute Engine [69] and others.

- **Custom VMs**: A cloud provider offers flexibility to its consumers by allowing to request custom resource configurations. There is no imposed VM configuration and the cloud consumers select the amounts of underlying resources that they need. The cloud provider establishes the prices for each low-level resource separately; hence the final price that the consumer pays is a subject to the custom requested configuration.

Let us consider example of three custom VM requests together with the prices from CloudSigma [42], illustrated in Table 2.2b. We can see that the seller’s offer contains the prices for the underlying resources, while the final consumer’s price depends on the requested custom configuration. The increasing number of cloud providers start to offer custom VMs because it allows their clients to adapt the services to their unique requirements without the need to consume and pay for any extra resources. Some providers who offer custom VMs include AT&T[3], CenturyLink[26], CloudSigma[42], IBM SoftLayer Cloud Servers [77], etc.
The problem of efficient cloud services provisioning is challenging and significant for both types of infrastructure cloud services. Therefore, in this thesis, we aim to address it for both pre-defined and custom VM configurations.

**Pricing Models**

The consumers of infrastructure cloud services are very diverse and have their own specific needs and objectives. It may be (i) a small business aiming to minimize the cost of the outsourced infrastructure, or (ii) a medium company with highly unpredictable workloads and the requirement for a flexible up and down-scaling, or even (iii) a large enterprise, which is looking for a long term commitment. Therefore, the infrastructure cloud providers offer various ways in which their consumers can procure, and pay for the required resources.

We identify three pricing models typically employed for infrastructure cloud services provisioning, which are *subscription-based*, *usage-based* (on-demand) and *market-based* pricing. The characteristics of the pricing models are given in Table 2.3. We analyse the commercial IaaS providers and use the materials from literature [5, 154, 135] to come up with a brief characteristic and application use-cases for each of the three pricing models.

- *Usage-based (consumption-based)*: Cloud computing became so popular due to the provided on-demand access to the resources. The clients request the cloud infrastructure services at any time and pay exactly for the amount of the consumed resources. Such consumption-based pricing has an identical nature to the other utilities, such as electricity or water, which apply metered usage payments. In this model, the prices per unit of the offered resources are fixed by the cloud providers, and the users pay for what has been consumed.

It is a convenient pricing model since it allows an instant access to the resources that can be flexibly scaled up or down based on the changing needs. However, it poses a

<table>
<thead>
<tr>
<th>Pricing Model</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pricing Model</strong></td>
<td><strong>Time</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td>Subscription-based</td>
<td>fixed-term</td>
</tr>
<tr>
<td>Usage-based</td>
<td>open-end</td>
</tr>
<tr>
<td>Market-based</td>
<td>open-end</td>
</tr>
</tbody>
</table>

Table 2.3: IaaS Pricing Models
challenge for the cloud providers, since they have to predict the required usage and ensure that they have sufficient infrastructure up and running. It is a great value model for the testing and development infrastructure, or for hosting next-generation and mobile applications, as it removes the on-premise IT infrastructure setup cost and allows to quickly adapt the compute capacity.

Despite the great value for consumers, such pricing model is generally not the most cost-effective. Due to the static nature of on-demand pricing and highly variable demand, the provider and consumer do not always realize the full benefit of cloud computing. It happens because during the time of a low demand, the consumers overpay for the requested resources, while during a high demand, the provider is disadvantaged by low prices. In order for the pricing model to be more efficient, it should be dynamic and preferably follow the laws of supply and demand.

- **Subscription-based (reservation contract/prepaid scheme):** This model permits the cloud consumers to pay an upfront fee for an agreed amount of cloud resources to be provided for some fixed term of usage. In other words, a user is subscribed to consume the cloud resources for a specific time period, which typically range from a month to several years. Subscription-based pricing is convenient when the resources are required for long-term, and there is a certain predictive threshold for the infrastructure requirements. In general, the cloud providers offer some significant discounts to the consumers who select prepaid pricing scheme depending on the contract duration (e.g. the longer the commitment is, the more discount is given). For example, CloudSigma would offer 10% and 25% discounts for annual and three-year subscription, accordingly. It is a convenient pricing model for the cloud providers, since it allows a more predictive consumption of cloud resources, provides a risk-free income, and ensures a long-term usage commitment to customers.

Very often, the cloud consumers combine the subscription-based and usage-based pricing models, where the former model is employed in order to reduce the total cost of the outsourced infrastructure, and the on-demand instances are used to address the sudden spikes in application requirements. The major drawback of the subscription model is that the user needs to make an accurate estimate for the resource requirements; otherwise, the customer may end up paying for unutilized capacity.

- **Market-based:** It is currently the least common pricing model in industry due to the high complexity of the infrastructure cloud services. It applies a dynamic price-
establishment process, which derives the current prices based on the level of demand in the market. In other words, the high demand is followed by higher prices, while the low demand will result in cheaper resources. Amazon EC2 is currently the only large cloud provider that offers the market-based pricing for infrastructure cloud resources, called Spot Instances [11]. They employ this model in order to sell their spare resources at a lower price compared to the on-demand instances. The market price of a spot instance is derived based on the amounts of available spare resources on Amazon EC2 (supply), and the bids of the potential consumers (demand). While the consumer bid is above the current market price, the user is granted access to the spot instance; however, if the market price gets higher than the consumer’s bid, the resources are taken away by Amazon in order to be provided to another higher-bidding client. The users always pay the current market price for the provisioned resource, which cannot drop below the Amazon’s threshold price (i.e. reserve price).

Even though market-based pricing is the least popular model at the moment, it is the most suitable for such a dynamic, rapidly changing environment as cloud marketplace. Its ability to reflect the changes in supply and demand through the market prices allows achieving better market efficiency. However, the market-based model provided in industry is limited to a specific set of application types, i.e. state-independent workloads. In order to address this issue, a cloud provider could run the market-based mechanism periodically to supply services for some agreed time period in future. For example, the market-based provisioning of VMs for the coming hour could be determined based on currently collected consumer requests.

In this thesis, we consider periodic market-based provisioning of cloud services, where the buyers and sellers requests are collected over the period of time and the market clearance happens at a specific pre-defined time. It is a practical approach which suits a wider range of customers by allocating cloud resources for a guaranteed period of time in future (e.g. next hour usage). Secondly, given a certain time period when the traders’ requests are collected, the periodic mechanisms gather more information about the current supply and demand in the market. It allows to make a better-informed decisions, which result in an improved quality of allocation and pricing outcomes compared to continuous markets, as well as prevents providers’ monopolies [129].
Table 2.4: Double-sided Economic Platforms for Trading Infrastructure Cloud Services in Industry

<table>
<thead>
<tr>
<th>Platform Name</th>
<th>Traded Commodity</th>
<th>Trading Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Compute Exchange (UCX) [139]</td>
<td>VM configurations</td>
<td>Spot/Open-end (pay-as-you-go)</td>
</tr>
<tr>
<td></td>
<td>kWAC - homogeneous unit</td>
<td></td>
</tr>
<tr>
<td>Deutsch Borse Cloud Exchange (DBCE) [4]</td>
<td>Low-level resources</td>
<td>Open-end &amp; Fixed-term (orderbook &amp; OTC)</td>
</tr>
<tr>
<td></td>
<td>CPU, RAM, HDD</td>
<td></td>
</tr>
</tbody>
</table>

**Infrastructure Cloud Exchange Platforms**

The cloud providers, discussed earlier in this section, deliver their cloud services to the consumers individually; therefore, they do not enter in a direct competition with the other providers. Although such single-provider markets are more common nowadays, the double-sided markets for infrastructure cloud services exchange begin to emerge. The primary goals of these exchange platforms are (i) to provide a uniform interface to different cloud providers, (ii) to measure and forecast the cloud consumption, and (iii) to enable cloud trading as a utility. We discuss some of the most notable cloud exchange platforms in industry, such as Universal Compute Exchange (UCX) and Deutsche Borse Cloud Exchange (DBCE). A brief summary is provided in Table 2.4.

- *Universal Compute Exchange (UCX)[139]* provides a centralized electronic marketplace for spot exchange of virtual infrastructure among the cloud providers and consumers. Inspired by the standard electricity unit measurement (kW/h), they introduce a standard unit for the virtual infrastructure metering, called Workload Allocation Cube (WAC)[14]. It provides a homogeneous view of the cloud infrastructures, by assigning a simple value to each complex composite service, i.e. VM. The basic principle is really simple: connect to the marketplace, and utilise the services on-demand (plug-in and consume).

In UCX, the market quantifies the expenditure and allows the consumers to pay the market price for the usage only. The participants enjoy the advantages resulting from a transparent free market, where multiple suppliers and consumers compete directly. The marketplace allows the users to measure the IT consumption, to benchmark the infrastructure which allows comparison of offers from different providers, to
forecast the future consumption based on previously measured results and to trade standardized contracts.

- *Deutsche Borse Cloud Exchange (DBCE)* serves as a marketplace where the low-level cloud infrastructure resources (e.g. CPU, RAM, and HDD) are traded among the infrastructure cloud providers and cloud consumers as utilities. The platform applies a two steps approach: trade and virtualise; where the users exchange the infrastructure resources and manage the virtual infrastructure built on top of the acquired resources.

The marketplace provides the transparency of prices, performance, and datacenter locations. It is responsible for the definition of the offered products and for providing a legal framework. In order to standardize the traded resources across the cloud providers (who can own different hardware infrastructure), a set of microbenchmark tests are conducted, and the results serve as a QoS assurance. There are two mechanisms for infrastructure cloud services exchange in DBCE: by trading open-end (spot/on-demand) or fixed-term (subscription) contracts via the (i) *orderbook*, as well as negotiating custom contracts privately using the (ii) *Over-The-Counter (OTC)* mechanism.

The capacity management platform provides a uniform web interface to manage the purchased resource. The consumers can build the virtual infrastructures based on the purchased resources, monitor the resource usage, and release the unutilised resources back to the market.

These two cloud exchange platforms apply conceptually different approaches for cloud infrastructure trading. The DBCE platform trades low-level cloud infrastructure resources, such as CPU, RAM, and HDD separately. The UCX marketplace, by contrast, uses a special metric (WAC) that gives a homogeneous view to the complex cloud infrastructure services, i.e. VMs. In this thesis, we derive the cases of marketplace provisioning of cloud infrastructure inspired by the principles applied in industry. In particular, we consider (i) homogeneous and heterogeneous types of cloud services, similar to predefined and custom VMs; and (ii) single and double-sided economic platforms, with a unique services provider or a fully competitive cloud services exchange market.
2.3 Market Mechanisms

Markets provide a medium that allows its participants (i.e. buyers, sellers) to engage in interactions in order to facilitate the provisioning and acquisition of specific goods and services. As an intermediary, a market has to gain the participants’ trust by ensuring professionalism and reliability [17], where efficient and fair market mechanism designs are typically preferred [28, 107]. The market mechanism is essentially a set of principles and rules that determine how the market operates. It defines the process by which participating buyers and sellers establish a market price and determine the amount of product or service to be exchanged in the market [40].

In this section we motivate the need for the market mechanisms for infrastructure cloud services provisioning, describe a taxonomy of market mechanisms and provide a detailed characterization of auctions. Some classic auction mechanisms are explained, and the relevant auction market mechanisms in literature are explained.

2.3.1 Why Markets for Infrastructure Cloud Services?

The cloud marketplace environment is a dynamic, large-scale ecosystem, where the supply and demand forces may change very quickly. The fixed pricing scheme is very inflexible and inefficient for the rapidly changing markets with highly-variable demand [89], such as cloud market. For instance, when the demand is low (below the fixed price), there will be an unrealised welfare; similarly, when the demand is high, there will be some unrealised profit since the provider cannot charge more than the fixed price.

Instead of trading computing resources for a fixed price, the allocation and pricing can be determined dynamically by a market mechanism [133, 103]. Considering the cloud resource allocation in economic terms has a number of significant benefits. First of all, market-based mechanisms provide a notion of relative worth, typically abstracted as a price, which is suitable for the infrastructure cloud services market [64]. Such a dynamic valuation-based approach is said to perform well for cloud marketplace, since it makes the allocation and pricing decisions based on dynamically changing supply and demand [126, 130]. As a result, the market-based resource pricing would ensure that the resources are consumed only when needed, would provide incentives to their users to adapt their demand to offpeak periods (when the prices are lower), and to return the idle resources back to the market in exchange for the market price [130]. Furthermore, a well-designed market mechanism can eliminate or limit the strategic manipulation of its participants by giving them an incentive to reveal their valuations truthfully, which is a very important
requirement for achieving efficient resource allocation. The existing knowledge about the
well-studied economic models can be used for further design and analysis related to the
problem of infrastructure cloud services provisioning. Finally, the market mechanisms es-
tablish well-understood and transparent rules and regulations, which allow gaining market
participants trust by ensuring professionalism and reliability [17, 28].

2.3.2 Taxonomy of market mechanisms

We provide a classification of market mechanisms, which is partially derived from [154]. We
make some amendments to the original taxonomy by generalising some of the classifications.
We distinguish four major classes of market mechanisms, which include (i) commodities
markets, (ii) tendering, (iii) bargaining/negotiation, and (iv) auctions. A brief summary
of the most differentiated characteristics of the discussed market mechanisms is provided
in Table 2.5.

Commodities market

*Commodities markets* are used for trading of raw or primary products. It is a very common
type of market, which is widely used to facilitate the investment trade in nearly 100 primary
commodities worldwide. In such markets the producers typically dictate the prices that
the consumers pay. The pricing rates may be of flat or variant type and they are generally
derived based on usage time, consumed quantity and other parameters. The variant-rate
pricing changes overtime, typically based on the balance of supply and demand in the
market. The posted price can be viewed as an extension of commodities markets, where

<table>
<thead>
<tr>
<th>Table 2.5: Characteristics of Market Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Typical Economic Platform</strong></td>
</tr>
<tr>
<td>Commodity market</td>
</tr>
<tr>
<td>Tendering</td>
</tr>
<tr>
<td>Bargaining/Negotiation</td>
</tr>
<tr>
<td>Auction</td>
</tr>
</tbody>
</table>
the "special offers" from resource providers, such as discounted items, irregular offerings with specific conditions, or discounts for large purchases, can be advertised openly in order to attract attention of the consumers.

Many commercial infrastructure cloud providers employ commodity market mechanisms, typically with fixed pricing for their cloud resources. However, fixed pricing mechanism does not achieve an efficient result when the market demand is variable [16, 1]. The dynamic pricing model is hard to apply to the infrastructure cloud services, given their complex nature. Therefore, currently only Amazon EC2 offers the variant-rate pricing for trading Spot Instances [11].

**Tendering**

Tendering is commonly used by large organizations, such as governments, in order to procure a customized product or service at appropriate price. In this market model, a consumer announces the details of the required product, and the provider companies analyse the requirements and make their price offers, or ignore the deal if they are not interested or not capable to fulfil the customer needs. The consumer chooses the best offer from the potential providers. The selected producer delivers the product according to the requirements in the customer’s tender for the proposed price.

Competitive tendering is not a feasible market mechanism type for cloud infrastructure provisioning, since it typically applies to the markets with large organisations and can be a very time-consuming process. The cloud market, in turn, is a rapidly changing environment populated by a lot of small and medium enterprises, where the provisioning decisions need to be made quickly.

**Bargaining/Negotiation**

Bargaining/Negotiation is a different form of market mechanism, where both the provider and consumer negotiate in order to reach the mutually acceptable price. Typically, the negotiation process starts with a high price from seller (aiming to maximize profit) and a low price from buyer (aiming to minimize cost). The negotiation process runs in iterations either until the agreeable price is achieved, or one of the negotiating parties is not willing to negotiate further. It is a long process, which is commonly used when it is hard to determine the prices based on supply and demand.

Bargaining, being a long price-determination process, is not a very effective market mechanism type for cloud marketplace environments, where the resources are often required
immediately. Deutsche Borse Cloud Exchange operates a negotiation mechanism, called over-the-counter (OCT), which offers a bargaining process that may take up to a week, which can be very inconvenient for the consumers.

**Auction**

*Auction* is a common name to refer to a process of buying and selling, where goods and services are offered up for a bid. Auctions are used in different domains in order to determine the value of a resource whose price is unknown. The auction mechanisms are typically designed to fit a specific setting with the particular problem requirements. However, a common aspect of any auction is that the price leader is an independent party, i.e. auctioneer, who does not intend to act in favour of consumer or provider, but rather determines the market results based on some pre-established transparent rules.

Auction mechanisms are fairly popular in research community for compute resource allocation, such as grid [131, 45], wireless spectrum [60, 156], and cloud [155, 102]. It is the best known market mechanism type for infrastructure cloud services provisioning, since auctions determine the market outcomes based on laws of supply and demand and can operate efficiently in a large-scale, dynamic environments, such as cloud marketplace. Furthermore, auction mechanisms can be designed to reflect the specific infrastructure cloud services requirements related to the resource classifications and the market type.

### 2.3.3 Auction Theory

Auctions have a long history and there is an evidence of auctions as early as 500 B.C. Nowadays, auctions are most often associated with procurement processes of some rare items, such as a painting by Picasso or Van Gogh; or with electronic procurement system, such as eBay. In fact, auctions can be of many different types and can be applied in various domains.

The price in an auction is neither predefined nor derived as a result of negotiation, but discovered through the process of competitive bidding. In other words, an auction is a protocol which determines the allocation of goods and the pricing outcomes, given the indications of its participants in terms of their interests in the traded resources [127]. We provide a brief classification of auctions, the desirable properties of auction mechanism design and some examples of classic auctions in mechanism design theory. Most materials presented in this subsection are derived from the academic literature on classic auction design [113, 107, 40, 74, 112].
Chapter 2. Background and Related Work

Figure 2.2: Auction Mechanisms Characterisation

**Auctions Classification**

An auction has a number of features. We list the most commonly recognized features, that either characterize the traded good or the market itself. A summary of the described auction characteristics is provided in Figure 2.2.

**Characteristics of the Traded Good (resource/service)**

1. **Aspects: single/multi-attribute (single/multi-dimensional auction)**

   This characteristic is linked with the number of aspects of the traded good, which are being the subject of auctioning. In a single-dimensional auction, there is only one aspect considered, typically price. Multi-dimensional auctions deal with multiple distinguished aspects of a good, such as quality of service (QoS), delivery time, etc.

2. **Instances: (single/multi-unit auction)**

   This classification refers to the number of instances of the traded goods. In a single-unit auction, each traded good is unique (it may be a single item or a bundle of items grouped together). A multi-unit auction can trade several copies of the same good. For example, in a multi-unit auction, where 200GB of cloud storage is offered, it is possible to bid for 50GB at 10c per GB; a single-item auction, by contrast, would allow to buy 200GB of storage as a single indivisible disk.

3. **Types: single-item/combinatorial auction(multi-item)**

   Based on the number of good types offered in a market, we distinguish between single-item and combinatorial auctions. In a combinatorial auction, multiple heterogeneous goods are traded simultaneously, and the bidders are able to specify an arbitrary combination of goods. For example, if compute resources of several different types, such as CPU, RAM, and HDD are auctioned, a buyer may bid $1 for a bundle of \( \{ 1 \text{ CPU core, 2 GB of RAM, and 50 GB of HDD} \} \), while another bidder may offer $0.5
for 500 GB of HDD. It implies the complementarity concept, which suggests that the value extracted from a bundle of resources is greater than the sum of her valuations for the individual goods, i.e. CPU, RAM, and HDD.

**Characteristics of the Market:**

1. **Economic platform / Market Type: single/double-sided**

   The type of economic platform defines whether only one side of the market is competing for resources, or the market is fully competitive. Single-sided auctions allow only one side of the market to place the bids on the offered goods. It may be either buyers competing for resources (direct market) or sellers providing competitive offers (reverse auction). The auctioneer’s role is to select the winning bid(s) among the submitted ones and establish the buyer prices. In double-sided auctions both buyer and seller sides submit bids (seller’s bid is typically called ask) and the auctioneer matches the buy requests with sell offers and sets the prices for exchanged goods.

2. **Revealed Information: open-cry/sealed-bid:**

   This auction classification is linked with the information that is disclosed in the market to the participants. In open-cry auctions, every bidder knows all the other bids, whereas in a sealed-bid auction, only the auctioneer has a complete information about the reported bids.

3. **Clearance period: periodic(discrete time)/continuous:**

   The clearance period defines the moment when the auctioneer establishes the final allocations and the market prices. In a discrete-time/periodic auction, there is a time period during which the market collects the participants’ bids. When the auction time expires, the auctioneer makes allocation and pricing decision based on the gathered information. In a continuous auction, the bids can be submitted at any time, and the clearance happens instantly based on the currently available information.

4. **Pricing type: discriminatory/uniform:**

   There are two ways in which the auction can establish the trading prices in the market: it can either be a single price applied to all the allocated trades (uniform pricing), or an individual price for each matched participant (discriminatory pricing). While the former ensures that no market participant is discriminated; the latter is a more personalized approach that derives the price based on individual buyer bids,
which provides a better reflection of the participant’s valuation for the allocated goods.

This categorization allows us to classify the auctions, for instance as multi-attribute single-sided sealed-bid combinatorial auction.

**Chapter 2. Background and Related Work**

Auction Mechanism Design Desiderata

Apart from the classification, the auction mechanism design aims at achieving a number of economic properties [107, 130]. An ideal design of an auction mechanism should satisfy a number of economic properties, such as:

- **Individual Rationality (IR)** ensures that the buyers and sellers can always obtain as much expected utility from trading in the market as if they avoid participating. In other words, the participants of such a market cannot suffer any loss in utility from taking part in an auction, i.e. it is individually rational to participate. Individually rational mechanisms are more trusted by the participants; therefore, this property is commonly considered as a design feasibility constraint.

- **Budget Balance (BB)** implies that the market neither accumulates surplus, nor runs in deficit and does not need to be subsidised by outside payments. In other words, a budget balanced market mechanism determines such prices, that the buyers’ prices are exactly sufficient to cover the payments to the sellers.

- **Truthfulness (T) / Incentive Compatibility (IC)** is linked with the strategic manipulation opportunity of the market participants. A truthful mechanism ensures that the dominant strategy is to reveal the truth, which means that a participant cannot improve her individual outcomes (benefit) by applying misreporting strategy. It is an important property since it is closely related to the market’s efficiency and simplifies the strategic decisions of the participants by giving them an incentive to act truthfully.

- **Economic/Allocative Efficiency (AE)** is generally the main objective of market mechanisms for resource allocation. An allocative efficient mechanism determines socially efficient distribution of the resources, in the way that the utility across all the participants, i.e. social welfare or overall "happiness", is maximised.

- **Computational Tractability** is concerned with the time complexity of the designed auction mechanisms. Based on the considered problem, this property may be a
feasibility requirement; for example, in a cloud marketplace, time-efficient market mechanisms are essential, since the outcomes in such a large-scale environment have to be determined quickly.

According to the conducted study [100], it is not possible to achieve all the economic properties at the same time for the majority of mechanism design problems. Therefore, a specific compromise should be found that allows to address a specific problem in the most efficient and effective manner. We outline some of the classic auctions, their strengths and weaknesses and explain the commonly applied techniques in auction design, which allow achieving specific economic requirements.

**Single-Item Auctions**

**Classic Auctions:** Some of the most commonly known classic auctions are English, and Dutch auctions. They trade single-attribute goods, where the only subject of bidding is the price. These are single-sided auctions, typically with a unique seller (direct auction), where all the participating bidders know all the bids (open-cry).

- *English auction*[23] begins with the seller’s starting price (reservation price), and the participating bidders cry-out their bids in an ascending order (ascending price auction) until there is no one who can outbid the current highest bid. The winning bidder obtains the auctioned item and pays the price she bid in order to win.

- In a *Dutch auction*[113], the process runs in a reverse order: the auctioneer starts at a price that no-one is likely to accept, and gradually drops the price (descending price auction) until there is a bidder who wishes to comply to it.

**Sealed-bid Auctions:** Other very popular auction mechanisms are 1st price sealed-bid auction, and Vickrey auction. Unlike the classic auctions, there is no sequential interaction among the bidders and the participants privately reveal their bids to the auctioneer in a sealed-bid fashion. Typically, there are two distinct phases: (i) a bidding period, when the participants submit their bids and (ii) a clearance phase, when the auctioneer determines and announces the winner. Depending on the way the auction selects the price that the winner pays, we distinguish between the 1st price and Vickrey auction (2nd price sealed-bid auction). We provide a brief explanation with an illustrative example in Figure 2.3a.

- *1st price sealed-bid auction*[52] defines the payment of the winning bidder based on her reported value. This auction is commonly criticized for the strategic incentives
that it offers to the bidders. In a sealed-bid auction, the participants have certain beliefs about the valuations of the other bidders. Therefore, the bidder with the highest true willingness to pay will try to guess what everyone else is going to bid in order to report a bid, which is slightly higher than the next highest one.

For example, given three buyer bids: $\beta_1 = $20, $\beta_2 = $23, and $\beta_3 = $18, the buyer with the bid $\beta_2$ will receive the auctioned good for the price of his bid - $23. However, the winning bidder could have a belief about the second highest bid, and could have an incentive to lower his bid from $23 to $21 in order to get the item at a cheaper price.

- **Vickrey auction (2nd price sealed bid auction)**[52] also assigns the item to the highest bidding participant, but the winning buyer pays the price of the second highest reported bid. Therefore, the participants do not have to estimate the willingness to pay of the other bidders, but should simply report the maximum price they are willing to pay. In other words, there is no incentive for the bidder to say anything else than the truth, because the participants always end up paying what it takes to win the auctioned good.

In the considered example, the buyer with the bid $\beta_2 = $23 is going to win the good at the price of the second highest bid $\beta_1 = $20. There is no incentive for the buyer to bid a different value than the true one: (i) if he lowers his bid, he may either lose the auction (e.g. $\beta_2 = $19 will lose), or still end up paying the price of the second highest bid (e.g. $\beta_2 = $21 will still pay $20), (ii) if the buyer reports a higher bid, he may be forced to pay the price greater than his true value (e.g. if $\beta_1 + = $24 then the buyer will have to pay $23, while his true valuation is at most $20).

**Double-sided auctions:** The previously discussed auction mechanisms operate in a single-sided economic platform, where only one side of the market (e.g. supply or demand) is bidding for the traded good. Double-sided auctions allow both supply and demand-sides to compete in the market, i.e. submit bids, and asks.

In order to determine the auction winners, the auctioneer has to select the buyers who receive goods and the sellers who sell. Therefore, in classic double-sided (single-item) auctions, the buy and sell orders are typically ranked according to the reported price: the buyers follow descending order of their bids and the sellers are sorted in ascending order of their asks. The mechanism selects the top k buyers and sellers (breakeven index k), such that the bid prices satisfy the ask offers, which ensures allocative efficiency. There is
Chapter 2. Background and Related Work

(a) First and Second Price Auctions

Figure 2.3: Classic Auction Mechanisms Illustrated

a number of different ways proposed for price determination in double-auctions. Please, refer to Figure 2.3b for illustration of our further discussion.

- **k-double auction**\(^1\) \([124]\) applies the surplus distribution principle based on \(\kappa\) value. This value is selected between 0 and 1, i.e. \(\kappa \in [0, 1]\) before the auction commences, and the price is determined as follows: \(\rho = \kappa \beta + (1 - \kappa)\alpha\). Such auction treats all trades of a given type alike, where the fraction of surplus depending on the \(\kappa\)-value goes to buyer and seller. For example, when \(\kappa = 1\), the buyer’s bid is deterministic for the price (called buyer bid double auction). \(\kappa\)-double auction can be implemented in a form of **uniform pricing**, when the values of the lowest matched bid and the highest matched ask are used: \(\rho = \kappa \beta_k + (1 - \kappa)\alpha_k\); as well as in a **discriminatory price** auction, where each matched couple of bid and ask are considered individually: \(\rho_i = \kappa \beta_i + (1 - \kappa)\alpha_i\), \(\forall i: \beta_i > \alpha_i\).

This type of auction is Individually Rational, Budget Balanced, Allocative Efficient, but not Truthful, and the participants have incentives to misreport their actual valuation for the auctioned goods.

- **McAfee auction** \([97, 152]\) addresses the truthfulness issue of \(\kappa\)-double auction and provides dominant strategies for both buyers and sellers. It is a uniform price auction, which applies Vickrey principle for price determination. Unlike the \(\kappa\)-double auction, McAfee auction does not derive the final price based on the declarations (bids and asks) of the winning participants. Instead, the bid and ask of the first losing buyer and seller are used, as follows: \(\rho = (\beta_{k+1} + \alpha_{k+1})/2\). If such price satisfies the \(k\)-th

\(^1\)Typically, the notation \(\kappa\) is used in a continuous double auction; we apply this notation in our explanation in order to distinguish it from the breakeven index \(k\), which is used to refer to the last supply-demand match.
buyer and seller, i.e. $\beta_k \geq \rho \geq \alpha_k$, then the first $k$ buyers and sellers will trade at this price. Otherwise, the least valuable trade, i.e. $\langle \beta_k, \alpha_k \rangle$ is prohibited by the mechanism, and the price for the first $k - 1$ buyers is set to the $\alpha_k$, and the first $k - 1$ sellers trade at $\beta_k$ price. Such denial of the possible trade is called trade reduction, and it is commonly used in order to achieve some specific properties by compromising on allocative efficiency.

As a result, McAfee mechanism maintains Individual Rationality, Truthfulness, but does not guarantee Budget Balance, and Allocative Efficiency (when trade reduction is applied).

Combinatorial Auctions

Combinatorial is the most generic class of auction, which permits multiple types of goods to be traded simultaneously. The auction participants are allowed to place bids on combinations of items, commonly called bundles or packages, rather than individual items. Combinatorial auctions are very popular and applied in many fields due to their high effectiveness for solving problems with complex requirements [44]. Combinatorial auctions are often applied for real-word problems, such as the radio-spectrum auctions, airport takeoff and landing slots allocation, cargo transportation services procurement, industrial procurement, for the London bus routes market [44, 127], etc. However, their effectiveness comes at a cost: in a case of combinatorial auctions it is a cost of computation. The computational complexity is a common challenge for large instances of combinatorial winner determination problem. In particular, there is no known polynomial-time optimal algorithm, i.e. the problem is NP-hard [123, 63]. Therefore, the combinatorial optimization problem has attracted a lot of attention in academia and industry, and various mechanisms have been proposed by researchers across the globe to address the problem. The key combinatorial auction ideas, including the problem formulation, connection with standard optimisation problems, such as set-packing problem and the issue of computation complexity were first discussed by Rassenti, Smith, and Bulfin in 1982 [117]. We outline some of the most notable and commonly applied techniques for combinatorial auctions below:

- **Vickrey-Clarke-Groves (VCG)** [144, 71, 39] is a great achievement of mechanism design theory. Its modifications have been applied in various computational task and resource allocation problems. In VCG mechanism, the selected winning bids maximise the social welfare in the market (i.e. an optimal allocation outcome is needed), while the payment of the winning bids is defined by the corresponding
social cost. It shows the externality imposed on the other bidders, i.e. the value of loss that the other agents experienced due to the winner's participation. Therefore, the VCG mechanism determines the social welfare if the winner did not participate in the market $W_{\beta, \cdot}$, and the initial social welfare of all the bidders except for the currently considered winner $W_{\beta} - \beta$. The difference between these two values is the welfare loss imposed on the others.

Let us consider an example of a single-unit combinatorial auction with three bidders, illustrated in Table 2.6. According to an optimal allocation, there will be two winners, $\beta_1$ and $\beta_3$ with the social welfare of $W_{\beta} = \$11$. If $\beta_1$ was not to participate in the auction, there would be only one winner $\beta_2$ and the corresponding social welfare would be $W_{\beta, 1} = \$10$. The welfare without the value of $\beta_1$ is $W_{\beta} - \beta_1 = \$5$; hence, the imposed externality is $\rho_1 = \$5$. The payment for the bidder $\beta_3$ is determined in a similar fashion.

We can see that the bidder payments depend on the total market efficiency; hence the choice of the efficient outcome is in participants' interest. Therefore, the VCG mechanism achieves allocative efficiency, truthfulness and individual rationality, but does not guarantee budget balance and suffers from a high computational complexity. It has been shown that if the optimal allocation mechanism is replaced by an approximation which achieves the outcomes in a feasible computation time, the auction loses its truthfulness [106].

- **Greedy Combinatorial Auctions** [90] offer a practical solution to difficult combinatorial optimisation problems and ensure truthful incentives for the auction participants. The greedy mechanisms are fairly popular due to their good approximation quality and low computation time, especially for the large scale problems [107, 21, 66]. The idea is inspired by time-efficient and near-optimal greedy heuristics and incentive compatible Vickrey principles. Initially, the combinatorial bids are sorted and allocated in a greedy fashion based on their value per item, which approximates the optimal solution. The payments of each winning bid are derived based on the minimum price by which the bid can win, i.e. critical value.

Given the previously considered example in Table 2.6, we illustrate the procedure of a classic greedy combinatorial auction. The greedy sorting order is determined based on value per item, for example$^2$: $\beta_2 = \frac{\$10}{\frac{1}{2}} \approx 7.07$. Based on the sorted order

$^2$In this example, we use a square root of the total number of items, as the most efficient parametrisation of this mechanism determined by Lehmann et al. [90]; a linear value per item would be $\beta_2 = \frac{\$10}{2} = 5$.
Table 2.6: Truthful Combinatorial Auctions Illustrated

<table>
<thead>
<tr>
<th>Bid</th>
<th>Compute Opt. VM</th>
<th>Memory Opt. VM</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>$6$</td>
<td>$0$</td>
<td>$6$</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>$0$</td>
<td>$0$</td>
<td>$10$</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>$0$</td>
<td>$5$</td>
<td>$5$</td>
</tr>
</tbody>
</table>

VCG Mechanism

Winners: $\beta_1$, $\beta_3$, $W_{\beta} = 6 + 5 = 11$

<table>
<thead>
<tr>
<th>$W_{\beta}$</th>
<th>$W_{\beta} - \beta_0$</th>
<th>Payment/Externality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>$10$</td>
<td>$11 - 6 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$10 - 5 = 5$</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>$10$</td>
<td>$11 - 5 = 4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$10 - 6 = 4$</td>
</tr>
</tbody>
</table>

Greedy Comb. Auction

Winners: $\beta_2$, $W_{\beta} =$ $10$

$L = \{ \beta_2(7), \beta_1(6), \beta_3(5) \}$

<table>
<thead>
<tr>
<th></th>
<th>Payment/Critical-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLB* $\beta_2$</td>
<td>$\beta_1(6)$</td>
</tr>
<tr>
<td></td>
<td>$6 \times 2^{0.5} \approx 8.5$</td>
</tr>
</tbody>
</table>

in the list $L$, there will be only one winner $\beta_2$ with the approximated social welfare of $W_{\beta} =$ $10$. The payment of $\beta_2$ is derived based on the first losing bid (i.e. $\beta_1$) by scaling her valuation according to the number of items in the winning bid as follows: $\rho_2 = 6 \times 2^{0.5} \approx 8.5$. The determined payment is the critical value, i.e. the minimum price that the winning bidder $\beta_2$ could have proposed in the auction in order to remain the winner.

This class of auction mechanisms is truthful, individually rational, budget balance and computationally tractable. Due to greedy heuristic approximation, the optimal outcomes are not guaranteed by the mechanism, but near-optimal solutions are commonly achievable. Greedy combinatorial auctions are very practical for large-scale problems where the possibility of strategic manipulation of the participants needs to be limited.

Greedy Combinatorial Auctions show promise for addressing the problem of market-based infrastructure cloud services allocation and pricing. This approach can allow achieving good quality allocations in a fast computation time [66, 21], and can elicit truthful/honest participants’ behaviour [90].
2.3.4 Market-based Compute Resource Management Approaches

Several researchers have investigated the problem of compute resource allocation and pricing and proposed different-market-based and game theoretic approaches to address it. Mechanism design theory has been applied a lot in the area of wireless spectrum allocation, where a primary licence holder, such as government, trades the usage rights for specific frequency band. The auction-based truthful mechanisms were proposed for addressing the problem in a number of scientific manuscripts [156, 157, 83, 151]. Unfortunately, the proposed mechanisms cannot be applied for cloud services allocation, because wireless spectrum is the only type of resource considered in their design, while a cloud service is a complex good, composed of a number of resources (e.g. CPU, RAM, HDD).

Jia et al. [80] consider several uniform channels partitioned into small cells for allocation among the bidders, where a simple greedy metric is proposed for user ordering. By analogy, cells can be viewed as different types of auctioned infrastructure cloud services. However, the presented work limits the number of uniform channels, whereas the cloud services are composed of several types of resources and should not be limited.

There is a body of research, investigating the cloud market from provider’s perspective, aiming to maximize the seller’s profit [91, 137]. In particular, Toosi et al. [136] combine the auction design with the proposed mechanism for dynamically calculating seller’s reserve prices based on electricity costs and Power Usage Effectiveness (PUE) of the data center. These works offer an interesting perspective to addressing the problem, but a truly fair and competitive market has to aim at maximizing the social welfare.

We discuss some of the related works for market-based cloud and grid resource allocation and pricing presented in literature. We split our overview into the discussions of single-sided and double-sided market mechanisms; a summary of discussed works is provided in Table 2.7.

**Single-sided Market Setting**

Roovers [119] investigates the problem of infrastructure cloud services allocation in a cloud market environment. The problem is approached from practical perspective based on a realistic market setting. The proposed solution is a continuous reverse auction, where the buyer’s complex requests are multicast to all the market sellers, who reply with their price offers. The presented work provides a well-suited approach for cloud environments. However, no regard is made to the economic efficiency, which is a very important design perspective. In this thesis, we aim at achieving both practical and economically efficient market mechanism designs for infrastructure cloud services trading.
Huu et al. [75] consider a combinatorial auction-based mechanism with multiple objectives, including energy consumption and operational cost minimisation. The authors propose three algorithms used for winner determination and payments computation, including exhaustive search algorithm, linear relaxation based on randomised rounding, and green greedy algorithm. The theoretical investigation reveals that the proposed mechanisms are monotone and truthful, while the experimental results show that the proposed greedy mechanism allows the seller to generate a higher revenue. In our work, we consider social welfare to be the major objective in the market and propose a number of mechanisms for its maximisation.

Zaman et al. [155] study combinatorial auction mechanisms for VM allocations in the cloud with the unique resource provider. In order to address the problem of services allocation, they propose two mechanisms: greedy heuristic scheme and a linear programming relaxation and randomized rounding mechanism. The pricing approach used in their work derives buyer payments based on critical-values. They prove that the greedy mechanisms are truthful and linear programming mechanism is truthful in expectation. From effectiveness perspective, their mechanism disregards an important requirement - the provider’s minimum price for the auctioned resources. In this thesis, we address a similar problem of homogeneous cloud services trading with seller’s reservation price constraint. We propose an updated allocation and pricing technique that allows reducing the mechanism’s computation time.

Nejad et al. [102] consider the problem of cloud marketplace, where the low-level resources, i.e. CPU, RAM, HDD, are traded as bundles. They propose a combinatorial auction mechanism based on greedy allocation and critical-value pricing. Their greedy approximation derives the sorting order of the bids based on initial scarcity of resources in the market, which allows achieving a good allocative performance. They show that the proposed mechanism is effective and prove that it ensures truthful bidding. However, due to the seller’s inability to influence the market pricing, the seller is disadvantaged (receives zero payment) in the markets when supply exceeds demand, which is highly undesirable. We address this issue and propose a novel iterative approach for winner determination that considers the scarcity of resources after each allocation. It allows to significantly improve the mechanism’s allocative performance and the utilisation of cloud resources.

Double-sided Market Setting

Xingwei et al. [153] develop the economic mechanisms to address the problem of resource allocation and pricing in double-sided cloud markets. They design a genetic
algorithm for services allocation and derive the market prices based on the limited English combinatorial auction. In their work, the economic market properties are not studied, and the mechanism’s outcomes can be influenced by untruthful bidding, which is an important property that we aim to achieve in our mechanism design.

Fujiwara [59] addresses a combinatorial auction problem for trading cloud services in different timeslots. The authors design an optimal combinatorial mechanism using a linear mixed integer program for resource allocation and uses a k-pricing scheme to derive the buyer and seller prices. The proposed mechanism is computationally hard and can be affected by untruthful bidding. It causes a serious issue for the mechanism’s economic efficiency, since the buyers and sellers can benefit from socially-harmful strategic behaviour. In our work, we aim at achieving truthful market mechanism designs and study the strategic incentives of the market participants.

Li et al. [92] apply a combinatorial exchanges model for resource allocation in the grid and describe a greedy allocation mechanism based on the average price for the requested bundle. Although the designed mechanism represents incentive compatible characteristic, there are some restrictive assumptions of their mechanism design. In particular, the authors consider a cloud service to be a divisible good, where a single buyer request can be satisfied by several asks. However, in order for the market mechanism to be applicable in cloud computing environments, the consumer requests have to be indivisible bundles. By analogy, a VM is an indivisible entity composed of CPU, RAM, HDD and other resources; this VM
needs to be provisioned entirely by a single provider.

Stober et al. [132] consider a realistic scenario, where the grid resources are traded in a double-sided market in a form of indivisible combinatorial bids. They design a greedy heuristic for large-scale grid resource allocation, where the computing power is considered to be a central scarce resource. In their considered problem, the central scarce resource is the only one being priced in the market, while the capacity constraints apply to the other resource types (e.g., memory, storage). Their proposed allocation mechanisms operate based on greedy sorted orders of bids and asks. The buyer pricing is derived based on critical-values required to win the resources, and the seller pricing is realised in a form of proportional payment. In such a pricing mechanism, each resource supplier receives a portion of surplus equivalent to her proportional contribution of resources sold in the market. The central scarce resource pricing is a restrictive assumption in cloud market environments, which can result in the buyer over-consuming the unpriced resources. In this thesis, we address this drawback by designing the market mechanisms that consider several types of resources when making allocation and pricing decisions.

We can conclude that the market mechanisms proposed in literature either fail to address the complexity of cloud services, or disregard important economic properties of mechanism design. Therefore, there is a need for more effective and efficient market mechanisms for infrastructure cloud services trading.

2.4 Agent-based Marketplaces

2.4.1 Why Agent-based Marketplace for Cloud Services?

The cloud environment is highly diverse and dynamic, with cloud providers offering different flexible resource provisioning schemes and consumers having highly variable workloads. This makes efficient resource allocation a very important and challenging problem. Recognizing this challenge, there has been a growing stream of interdisciplinary research including economics and computer science, with researchers proposing various market-based approaches for trading cloud services in different market settings. However, at present, there is no marketplace platform that allows automated trading of cloud services between multiple independent users.

Intelligent agents show promise for effective automation of electronic market processes and facilitation of trading decisions for its participants. The complex cloud marketplace environment, where various individual users trade the resources in different markets, makes
intelligent decision making especially important. Moreover, intelligent agents are often used and prove themselves to be efficient in service offers search (e.g. setting up travel arrangements [57]), or to bargain the price for specific product [150], which is similar to our problem. According to Gartner’s predictions for Digital Future 2016 and beyond, the autonomous agents/personal assistants will participate in five percent of all economic transactions outside of human control; the smart agents are expected to facilitate around 40% of mobile transactions by 2020 [114]. Therefore, addressing the problem of infrastructure cloud services trading as an agent-based marketplace platform can offer an efficient solution with future perspectives. In this section, we provide the summary of the agent-based marketplaces proposed in literature and developed in industry.

2.4.2 Overview of Existing Agent-based Marketplaces

An electronic marketplace, or e-marketplace, refers to an internet based platform for carrying out e-commerce. In recent years, e-marketplaces such as eBay, Alibaba and Amazon, as well as hundreds of thousands of e-commerce businesses have been shifting the landscape of the retail and distribution businesses from physical market environments to Internet-enabled market environments [141]. The rising complexity of electronic markets’ related activities necessitates decision-support and decision-making tools that can automate, augment and coordinate some or all of the decision processes leading to the service exchange. Agent-mediated e-commerce has been proposed and studied for over a decade by many researchers who have used the popular consumer buying behaviour (CBB) model [72, 73, 109] to develop intelligent-agent-based technology solutions that can help market participants (and in particular human end-consumers) in the different phases of the product purchase cycle. Starting from simple electronic platforms that provide the communication infrastructure to a more sophisticated automated trading systems, agent-based technology is going to take a leading role in electronic marketplaces of the future [114].

Agent-Based Marketplaces In Literature

First generation agent-based markets: The intelligent agents started to be applied for electronic commerce in mid-90s. Different agent-based marketplaces began to appear in research papers, proposing relatively simple agents negotiation frameworks. The major focus of the early agent-based marketplace investigation was on the communication infrastructure, information and interaction models, standards for resource description, etc. Among some of the most notable marketplaces proposed at that time are Kasbah, Magma,
Chapter 2. Background and Related Work

METU-EMar, and SICS MarketSpace; the works that inspired the research community for further exploration in this field.

- **Kasbah**[27] is the first marketplace which automated the trading process by providing the agents who autonomously buy and sell goods on behalf of their users. The proactive seller agents were proposed: they contact the interested buying agents and negotiate to find the best deal directly.

- The **Magma**[138] marketplace approaches the problem from practical perspective and provides two economic mechanisms in which the market participants can establish the deals: (i) via **direct negotiation**, as well as (ii) through the **broker** with pre-established transparent rules. The negotiation model allows to negotiate prices manually as well as to employ the software agents for an automated process. The brokered producer-consumer transactions implement Vickrey auction.

- The **METU-EMar**[50] marketplace proposes a two step market process. Initially, the **resource discovery agent** finds the best seller who can provide the required products and services. After that, the buyer and the seller agents are introduced to each other for further **direct negotiation** of price.

- **SICS MarketSpace**[55] allows **one-to-many bilateral negotiation**, i.e. direct negotiation with multiple sellers simultaneously, unlike the previously mentioned **one-to-one negotiation** marketplaces. This agent interaction model allows emulating such popular auction protocols as English or Dutch auctions.

**New era of agent-based marketplaces:** While the marketplaces of simple forms automated negotiation over the price (single-attribute), it is only one aspect of retail markets. In order to address the real-world problems, there was a need for more sophisticated and elaborate techniques for realizing the agent-based marketplaces, which could deal with complex user preferences and could allow negotiation over complex goods. Some of the most significant contributions that followed in early 21st century include such marketplaces as FuzzyMAN, PumaMart and EMarketRL.

- **FuzzyMAN**[88] allows the users to express their preferences in fuzzy terms. The negotiation happens over multi-attribute subjects and the process is realized between multiple parties (multiple sellers and buyers). The mediator agent performs a pre-selection procedure for negotiation; followed by the direct multilateral negotiation.
The *PumaMart*[147] marketplace allows for automatic evaluation of the offers based on fuzzy user preferences and applies the mechanisms for providing fast response to the consumers. It employs an *auction-like* one-to-many bilateral negotiation model based on fuzzy evaluation criterion, where the consumer agent is the auctioneer and shop agents are the bidders.

The *EMarketRL*[120] marketplace enables trading based on three main factors – quality, price, and delivery time. The mediator agent conducts the pre-selection process and determines the sellers who offer the requested product. The further negotiation takes a form of reverse auction, where the buyers multicast their requests to the sellers who offer the required product. The entire trading process is based on sellers/buyers constructing the reputation of the other buyers/sellers, as well as sellers learning in order to adjust the price, delivery-time and the product-quality to maximize their profits.

**Service-based marketplaces:** The service-based approach has also gained attention in the field of market-based trading. The major difference is that instead of multi-agent system, the marketplace is based on service-oriented architecture [86], where one application components provide services to other components through a communication protocol. We discuss some of the service-oriented marketplaces for trading utility computing and cloud services discussed in literature, such as Mandi, DRAGON-Lab, and CloudSim.

- *Mandi*[62] provides a service-based market framework to facilitate the trading between the cloud consumers and cloud providers. The platform allows the consumers to search for and aggregate the products from various sources based on requirements. The major feature of this marketplace is that it allows simultaneous existence of multiple markets as well as gives flexibility in terms of market mechanisms (e.g. commodities market, auction).

- *DRAGON-Lab* [61] is an independent autonomous system connected to multiple real networks in order to provide both testbed network and testbed system. It is designed to provide a cooperative research platform for consideration based large scale internet.

- *CloudSim* [25] is a framework for modelling and simulation of cloud computing infrastructure and services. The framework supports system and behaviour modelling of various cloud system components, including data centres, virtual machines, services provisioning and pricing policies.
Agent-based Marketplaces In Industry

**Intelligent Agents for Product and Merchant Brokering:** Initially, the intelligent agents started to be used in e-Marketplaces for product and merchant brokering [84]. It is a kind of pre-negotiation step that intends to find the suitable partners to negotiate.

- *Frictionless Commerce* [22] online shop was one of the first e-marketplaces of this kind. This system allowed its users to choose what to buy and who they want this product from. The consumers of the services could initialize their agents with their individual criteria (e.g. product features and merchant services) as well as specify their vague preferences. Frictionless Commerce utilized *multi-attribute utility theory* [51] for ranking the crisp sellers’ proposals according to the consumers’ preferences.

- *Multi-Agent Trading Environment (MATE)* [108] is another example of commercial agent-based e-Market. The key differentiation point of MATE was that a *purchasing assistant* and the merchants’ agents were added to the marketplace architecture, which are acting on behalf of their human users. The system utilized *fuzzy-logic matching* [85] to automate the complex process of product selection so that both the buyer’s criteria and the merchant’s business goals are satisfied.

**Intelligent Agents in Auction-driven Marketplaces:** Later, agent-based online auction-driven marketplaces started to appear, and the negotiation process was replaced by a bidding procedure realised via a mediator/auctioneer. Some the examples include:

- *eBay Auction Web* [53] provides *English auction* for trading valued items online. For a bidder, the system offered an optional "phantom" bidding service, called *proxy bidding*, which is a kind of a buying agent. The English auction offered by eBay can be viewed as a multi-agent marketplace where the agents negotiate over the price issue.

- *Fishmarket electronic auction house* [87] is another example of auction-driven marketplace. This marketplace employed *Dutch auction*, where the price is being lowered down until some participant is willing to accept the auctioneer’s price. What make the Fishmarket different is that it allows the users to employ their own buying and selling agents, i.e. *custom trading agents*, of arbitrary complexity, as well as allows to use the market-owned agents for bidding.

- *Amazon EC2 Spot Instances* [11] allows their customers to bid for the spare computing resources, where the market outcomes, including allocation and pricing are
<table>
<thead>
<tr>
<th>Marketplace/Platform</th>
<th>Trading Model</th>
<th>Multi-market Support</th>
<th>Custom Trading Policy</th>
<th>Execution Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Direct</td>
<td>Mediated</td>
<td></td>
<td>Real-time</td>
</tr>
<tr>
<td>Kasbah</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Magma</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>METU-EMar</td>
<td>✓</td>
<td>x*</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>SICS MarketSpace</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>FuzzyMan</td>
<td>✓</td>
<td>x*</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>PumaMart</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>EMarketRL</td>
<td>x</td>
<td>✓*</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>In Academia</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandi</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DRAGON-Lab</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CloudSim</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Service-based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictionless Commerce</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>✓</td>
</tr>
<tr>
<td>MATE</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>In Industry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eBay Auction Web</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fishmarket e-auction</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Amazon EC2 Spot Instances</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* pre-selection is performed by a mediator/resource discovery agent.
determined based on Amazon’s proprietary auction mechanism. Their auction-driven marketplace operates in a single-sided setting, where the resources of a unique seller, i.e. Amazon, are auctioned among multiple bidders. Due to a high complexity of infrastructure cloud service, Amazon trades the VMs of different configurations separately, which significantly simplifies the problem. They also offer a programming interface (API) and various monitoring tools that allow the users to build their custom software agent which follow their individual strategies for cloud services procurement.

A summary of the discussed marketplace platforms is presented in Table 2.8. We can notice that the more advanced platforms and marketplaces in industry support simultaneous existence of multiple markets and allow for custom user-defined trading policies. Both components are important for a fully competitive open market, where the users are can initialise their markets with different mechanisms and apply individual trading strategies.

2.5 Summary

This chapter provided a systematic analysis of nowadays market of infrastructure cloud services and presented an overview of market-based methods for cloud resource management. Multi-unit combinatorial auction seems to be the most adequate model for realising the cloud services market provisioning process; it allows to reflect the nature of the traded service and to determine the market prices based on the laws of supply and demand. Greedy combinatorial auction looks like the most promising approach for addressing the problem because it allows achieving good allocation quality in a fast computation time and can elicit truthful participants’ behaviour. Based on current state-of-the-art mechanisms proposed in literature, there is no truly effective and efficient solution, which motivates the work presented in this thesis. Therefore, in the following chapters we design the market mechanisms that improve the practicality of the solutions proposed in literature by considering more realistic problem settings, and achieve more efficient allocation and pricing outcomes by applying some improved and novel approaches. Finally, in this chapter, we described the agent-based market platforms in literature and industry. It allowed us to identify a number of important features for the agent-based marketplace for infrastructure cloud services, such as simultaneous existence of multiple markets, and custom automated trading strategies. In this thesis, we apply agent-based approach in order to achieve the desired functional characteristic of a marketplace platform. Therefore, we propose an agent-based cloud marketplace architecture, which is generic enough to accommodate the aforementioned requirements.
Chapter 3

IaaS Cloud Marketplace: Conceptual Framework and Problem Formalisation

This chapter provides a conceptual framework for the marketplace of infrastructure cloud services. We derive our problem requirements based on the analysis of current IaaS market, provided in Chapter 2. A general problem of Cloud Infrastructure Allocation and Pricing (CIAP) is formulated as Integer Program and the mechanism design framework is outlined. It includes some important design aspects and desirable economical properties of the market mechanism design, i.e. design desiderata. Finally, a general approach for addressing the described problem is outlined.

3.1 IaaS Cloud Marketplace Conceptual Framework

The proposed conceptual framework of infrastructure cloud marketplace is depicted in Figure 3.1. The framework contains three major building blocks that interact with each other and establish a dynamic interrelated ecosystem. The roles and characteristics of the marketplace components are outlined below.

3.1.1 Marketplace Participants

The market participants are separated into two distinctive groups. Depending on their role in the market, we differentiate the services consumers/buyers and resource providers/sellers. Both join the market in order to engage in trading activities and access or provision services. The market participants are assumed to know their preferences and act selfishly in
order to improve their individual outcomes.

The services consumers are aware about their cloud infrastructure requirements and have some budget constraints that they need to satisfy. The providers, in turn, possess some cloud infrastructure resources with the associated costs for running services. This information is private and each participant is only aware of her own requirements and resource valuation.

The market participants are considered as strategic self-interested agents with incentives to maximise their market outcomes even at the cost of other traders and the social well-being. Such a selfish behaviour may be harmful for the market and may have a negative impact on the well-being of the other participants and the overall market’s efficiency. Hence, it is essential that the self-interested participants are given the incentives to act truthfully in the market.

3.1.2 Cloud Services Market

A market provides a medium that allows its participants, i.e. consumers and providers, to engage in interactions, and facilitates the trading process of specific goods and services. An IaaS market has a general anatomy, i.e. underlying structure, defined by the type of traded infrastructure cloud service and the type of economic platform in place. The market trading decisions and other associated processes are facilitated by the market mechanisms which operate based on a specific market anatomy.

Infrastructure Cloud Service Type

The infrastructure cloud service refers to a high-level service offered to the cloud consumer, such as a Virtual Machine/Cloud Server. The infrastructure cloud resource is the underlying hardware resource, such as CPU power, RAM, or storage, which are offered for consumption as services. In the infrastructure cloud environments, there are typically two distinctive types of offered services: pre-defined and custom VMs. As discussed in Chapter 2, pre-defined VMs are the services with pre-built configurations; whereas custom VMs have flexible configurations selected by the consumers. Therefore, in our conceptual framework, we consider two types of services, based on the way their configurations are defined: homogeneous and heterogeneous infrastructure cloud services.

- **Homogeneous Infrastructure Cloud Service** is a Virtual Machine of a particular purpose group (e.g. general, compute optimised, memory intensive, etc.) with a pre-defined resource configuration. Similar to the real-world, there may exist several sizes
Figure 3.1: Conceptual Framework of Infrastructure Cloud Marketplace

of VM configurations (e.g. small, medium, large). However, the underlying resources in VMs of different sizes scale according to a unique strong relationship (e.g. large = 2 medium = 4 small) (example in Table 2.2a). We call this service type homogeneous because such a relationship allows us to establish a one-dimensional view of a complex service. Please, note that Universal Compute Exchange [139] applies a similar approach by abstracting a virtual infrastructure into a standard measurement unit, called WAC [14].

- **Heterogeneous Infrastructure Cloud Service** is a Virtual Machine with flexible resource configuration selected by the cloud consumers based on their individual needs. The relationship between such custom-built VMs is not always clear and possible to define. Therefore, such a service has to be considered as a complex heterogeneous good, composed of low-level resources, (e.g. CPU, RAM, HDD) (example in Table 2.2b). In other words, a market of heterogeneous infrastructure cloud services is trading heterogeneous bundles of the underlying resources. Please, note that Deutsche Borse Cloud Exchange [4] applies such principles in their platform for cloud infrastructure exchange.
Economic Platform Type

Economic platform defines whether only one or both sides of the market are competing for resources. As discussed in Chapter 2, there are two economic platform types (market types) applied in nowadays market of cloud infrastructure: the single-sided direct market and double-sided market.

- In a single-sided direct market there is a unique cloud service provider who offers the services to the customers (e.g. Amazon EC2 Spot Instances [11]). The provider also takes the role of an auctioneer, who establishes the rules of market procedure, while the cloud consumers compete for the right to use resources.

- In a double-sided market both consumers and providers engage in a fully competitive market (e.g. DBCE [4], UCX [139]), where supply and demand sides are in a direct contest with each other for provisioning and accessing the cloud services.

Market Mechanism

The market mechanism is used to guide and facilitate the market trading process and decisions, including the winner and price-determination. As an intermediary, the market has to gain the participants’ trust by ensuring professionalism and reliability [17], where efficient and fair market mechanism designs are essential, especially in the presence of strategic participants and incomplete information [107, 28]. Therefore, market mechanism design has to focus on the impact that the market rules can have on the behaviour of the participants, i.e. the behaviour elicited by the rules of the market mechanism [107]. For example, truthful mechanisms ensure that the participants do not have an incentive to lie in the market. We outline and formulate the most important economic properties of the market mechanism design for trading infrastructure cloud services in Section 3.2.

3.1.3 Provisioning Scenarios

The goal of a market mechanism is, given the information provided by the participants in terms of their cloud infrastructure requirements and price valuations, to determine an efficient distribution of services and establish such market pricing that gives truthful incentives. We provide some use-case scenarios of infrastructure cloud provisioning in a marketplace, and illustrate the types of seller offer and buyer request, given in Figure 3.2.
Chapter 3. IaaS Market: Conceptual Framework and Problem Formalisation

Figure 3.2: Examples of Cloud Service Offers and Requests

<table>
<thead>
<tr>
<th></th>
<th>SELLS OFFER</th>
<th></th>
<th>BUYS REQUEST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMALL VM</td>
<td>MEDIUM VM</td>
<td>LARGE VM</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>$0.08</td>
<td>$0.16</td>
<td>$0.32</td>
</tr>
</tbody>
</table>

(a) Homogeneous Cloud Services

<table>
<thead>
<tr>
<th></th>
<th>SELLS OFFER</th>
<th></th>
<th>BUYS REQUEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>540</td>
<td>540</td>
<td>10,800</td>
</tr>
<tr>
<td>RAM</td>
<td>540</td>
<td>540</td>
<td>10,800</td>
</tr>
<tr>
<td>HDD</td>
<td></td>
<td></td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>$0.03</td>
<td>$0.023</td>
<td>$0.0001</td>
</tr>
</tbody>
</table>

(b) Heterogeneous Cloud Services

Market Seller Offer

The infrastructure cloud providers have some limited infrastructure cloud resources that they are willing to provision to the cloud consumers. Consider a company that is running a datacentre to address their own needs. During the last month, due to a highly optimised business processes, they start to experience excess of idle infrastructure resources. In order to make use of not utilised computing assets, their strategic department decides to provision their spare resources to the consumers in a cloud marketplace. They estimate the cost associated for running the infrastructure cloud resources and do not wish to sell them at a lower price. The company joins the marketplace and makes the infrastructure cloud services offer. The examples of seller offer for trading homogeneous and heterogeneous infrastructure cloud services are given in Figures 3.2a and 3.2b, respectively. Both offers have the same structure: the goods of different types (resources or services) are offered with the corresponding minimum prices, specified per unit of each good type. In both examples, there is the same amount of offered infrastructure cloud resource in terms of CPU, RAM, and HDD; however, the homogeneous service offer abstracts these resources into small, medium, and large VMs according to the configurations provided in Table 2.2a.

Market Buyer Request

The infrastructure cloud consumers are willing to outsource their in-house infrastructure to the cloud. Such infrastructure may be complex and can contain a number of interconnected VMs. Imagine a medium size company that is running an e-commerce website. Their commercial enterprise application is composed of a number of servers, which include
(i) an application server for hosting the business logic, (ii) a database server for keeping a persistent track of their customers, and (iii) a secure server for handling electronic transactions. According to their IT department, they require two small VMs for hosting the database and the secure server, and one large VM for the application. They have a budget constraint which allows them to pay at most 80c/hours for the entire infrastructure, if their website is fully functional. Hence, they join the marketplace and make their request for cloud services. Figures 3.2a and 3.2b provide examples of buyer requests for homogeneous and heterogeneous infrastructure cloud services. Both requests specify the requirements for the cloud infrastructure as well as the maximum willingness to pay for the entire requested bundle. The infrastructure cloud resources, specified in the heterogeneous cloud server request, can be used to build multiple custom VMs, including 2 small and 1 large, based on configurations in Table 2.2a. Such custom-built configurations can provide a better fit for the specific buyer requirements.

3.2 Problem Formalisation

In order for the market to be effective, it needs to operate in a realistic setting. Therefore, in this section, we outline a number of technical requirements that are essential for the cloud consumers and cloud providers, describe the market model, and formulate the problem based on the listed requirements. We provide a general problem formulation for the double-sided market, and provide additional explanations and amendments for single-sided economic platform, as appropriate.

3.2.1 Technical Requirements

We have analysed multiple infrastructure cloud providers\(^1\) and a number of emerging infrastructure cloud marketplaces\(^2\) in order to come up with a realistic set of technical requirements for our problem formulation.

- **Bundling**: Typically, commercial infrastructure cloud providers allow constructing complex cloud infrastructures by offering Virtual Machines of different sizes (e.g. micro, small, medium, large). These VMs, in turn, are composed of a number of various infrastructure cloud resources (e.g. CPU, RAM, HDD) (Figure 3.2). Therefore, a

\(^1\)Amazon EC2, AT&T Cloud Solutions, Century Link/Savvis, GoGrid, Ninefold, Rackspace, Verizon/Terremark Cloud, Windows Azure

\(^2\)Universal Compute Exchange, Deutsche Borse Cloud Exchange, Equinix Cloud Exchange, Open Cloud Exchange from CoreSite
cloud marketplace has to allow trading of multiple types of goods in bundles (i.e. combinatorial requests), similar to cloud providers in industry.

- **Request Indivisibility**: Infrastructure cloud service is indivisible in its nature, and cannot be provisioned partially or supplied by a number of different providers. For example, CPU power cannot be provisioned separately from RAM, due to the technology limitations. Hence, the goods in the market are considered to be complementary, and a set of goods requested in a bundle must be acquired entirely from a single infrastructure cloud provider.

- **Demand-side Aggregation**: The infrastructure cloud providers, as the datacentre owners, manage their infrastructure and deliver their services to a large number of consumers. Accordingly, a cloud marketplace has to permit the cloud providers to make divisible offers that can be used to satisfy more than a single consumer request, similar to a real-world cloud market.

- **Fast Market Clearance**: A market of infrastructure cloud services is a dynamic, and rapidly changing environment with a large number of users. The provisioning decisions in such environment have to be made quickly; thus, the market mechanism has to be able to clear the market instances in a feasible time.

### 3.2.2 Market Model

#### Problem Notation

The problem notation used in this thesis is summarised in Table 3.1. In this thesis, we use capital letters to refer to sets, except for the following notations $K, N, M$ that correspond to the total numbers of goods, buyers and sellers, respectively; the lowercase letters represent variables. The upper indexes in variables signify the type of variable, i.e. the subject to which a variable relates; the bottom indexes are used for referencing, that allow to locate the right variable from a set. For example, $p_{sg}$ would refer to a unit reservation price of a seller with index $s$ for a good with index $g$. We omit the notion of seller $s$ in a single-sided market, since there is only one unique participating seller.

#### Participants

The market allows trading of $K$ different good types $G = \{1, \ldots, K\}$, among its participants. There are two categories of market participants, depending on the role they perform in the market: *buyer* and *seller*. The buyers $B = \{1, \ldots, N\}$ aim to procure goods in the
### Table 3.1: CIAP Problem Notation

#### Market Sets:
- **G**: set of good types \( \{1, \ldots, K\} \)
- **B**: set of buyers \( \{1, \ldots, N\} \)
- **S**: set of sellers \( \{1, \ldots, M\} \)

#### Bid and Ask Model:
- \( c_{bg} \) capacity of good \( g \in G \) demanded by the buyer \( b \in B \)
- \( p_{vb}^v \) price valuation for the bundle of capacities from the buyer \( b \in B \)
- \( c_{sg} \) capacity of good \( g \in G \) supplied by the seller \( s \in S \)
- \( p_{sr}^{ur} \) unit reservation price for a good \( g \in G \) from the seller \( s \in S \)
- \( p_{sb}^{br} \) bundle reservation price of seller \( s \in S \) for the bundle of buyer \( b \in B \)

#### Market Outcomes:
- \( X \): allocation decision vector, where \( x_{sb} \) is an allocation decision for the couple of buyer \( b \in B \) and seller \( s \in S \)
- \( \mathcal{P}^b \): buyers’ pricing vector, where \( p^b_s \) is buyer’s \( b \in B \) final price
- \( \mathcal{P}^s \): sellers’ pricing vector, where \( p^s_s \) is seller’s \( s \in S \) final payment
- \( W \): social welfare

Market by making requests, and the sellers \( S = \{1, \ldots, M\} \) are willing to trade their goods by making offers. Since the participants in the market are assumed to be self-interested and strategic agents, their true (private) knowledge about their valuation for the traded goods can be different from the one declared to the market. Therefore, the information in regards to the participants’ requirements and valuation has to be modelled as two different types, called private and declared.

### Declared Type

A declared type is the information that a participant reveals to the market, which may be a result of participant’s strategic behaviour.

The buyers’ declared type takes form of a bid (i.e. market request). They declare their requirements by requesting indivisible bundles of goods \( \hat{C}^d_b = (\hat{c}^d_{b1}, \ldots, \hat{c}^d_{bK}) \), so that \( \hat{c}^d_{bg} \geq 0, \forall b \in B, \forall g \in G \), and express their price valuation \( \hat{p}^v_b \geq 0, \forall b \in B \) for the whole requested bundle, which represents their maximum willingness to pay. The buyer’s declared type (i.e. bid) is expressed as follows:

\[
\hat{\beta}_b = \langle \hat{C}^d_b, \hat{p}^v_b \rangle = \langle \langle \hat{c}^d_{b1}, \ldots, \hat{c}^d_{bK} \rangle, \hat{p}^v_b \rangle
\]
The sellers’ declared type takes form of an ask (i.e. market offer). They report to the market the capacities of goods of different types that they are willing to trade $\check{C}_s = \langle \hat{c}_{s1}, \ldots, \hat{c}_{sK} \rangle$, so that $\hat{c}_{sg} \geq 0, \forall s \in S, \forall g \in G$, and declare their unit reservation prices $\check{P}_s = \langle \hat{p}_{s1}, \ldots, \hat{p}_{sK} \rangle$, so that $\hat{p}_{sr} \geq 0, \forall s \in S, \forall g \in G$, which are minimum prices for a unit of good that the seller agrees to accept. The sellers’ declared type (i.e. ask) is expressed as follows:

\[
\check{\alpha}_s = \langle \check{C}_s, \check{P}_s \rangle = \langle \langle \hat{c}_{s1}, \ldots, \hat{c}_{sK} \rangle, \langle \hat{p}_{s1}, \ldots, \hat{p}_{sK} \rangle \rangle
\]

The minimum price that the seller wishes to obtain for a specific bundle of goods requested by the buyer, the bundle-specific reservation price, is determined as follows:

\[
\check{p}_{br} = \sum_{g \in G} \hat{p}_{sg} c_{bg}
\]

the declared unit reservation prices of the seller $s \in S$ are scaled according to the amounts of goods, requested in the bid of buyer $b \in B$.

**Private Type**

A private type is the participant’s private information about her actual true price values for the goods in the market. The buyer and seller private types have an identical structure to their declared types but they are denoted by the symbols without hats “$\hat{\ldots}$”.

The buyer’s private type is expressed as follows:

\[
\beta_b = \langle \check{C}_b, \check{P}_b \rangle = \langle \langle \hat{c}_{b1}, \ldots, \hat{c}_{bK} \rangle, \langle \hat{p}_{b1}, \ldots, \hat{p}_{bK} \rangle \rangle
\]

The buyers in the market are assumed to be single-minded, which is an extreme case of complementarity [90]. Such a buyer type is a good reflection of the real-world cloud consumer with a strong minimum requirement to the cloud infrastructure. The consumer is interested in a specific bundle of goods, and a partial fulfilment of the request does not produce any value to her. Formally, single-minded buyer wishes to obtain exactly the requested bundle of goods $C^d_b$, and derives her true price value $p^v_b$, if she obtains the desired bundle or any superset of it; otherwise her price valuation is zero:

\[
p^v_b(\hat{C}^d_b) = \begin{cases} 
p^v_b, & \text{if } C^d_b \subseteq \hat{C}^d_b \\
0, & \text{otherwise} \end{cases}
\]
Chapter 3. IaaS Market: Conceptual Framework and Problem Formalisation

The seller’s private type is expressed as follows:

\[ \alpha_s = \left( C_s^s, P_s^p \right) = \left( \{c_{s1}, \ldots, c_{sK}\}, \{p_{s1}^{ur}, \ldots, p_{sK}^{ur}\} \right) \]

The seller’s true private value function, for the bundle of resources requested in the bid, derives the bundle-specific reservation price based on seller’s true unit reservation prices as follows:

\[ p_s^r(C_b) = \sum_{g \in G} p_{sg} c_{bg} \]

### 3.2.3 General Cloud Infrastructure Allocation Problem

We formulate the general Cloud Infrastructure Allocation Problem (CIAP) as an Integer Program as follows:

\[
\begin{align*}
\text{Maximize } W & \quad \text{(Obj)} \\
\text{Subject to: } & \\
\sum_{b \in B} c_{bg} x_{sb} & \leq c_{sg}^s, \forall s \in S, \forall g \in G \quad \text{(ARC)} \\
\sum_{s \in S} x_{sb} & \leq 1, \forall b \in B \quad \text{(BAC)} \\
x_{sb} & = \{0, 1\}, \forall s \in S, \forall b \in B \quad \text{(BIC)} 
\end{align*}
\]

CIAP aims to determine the allocation decision \( X = x_{sb}, \forall s \in S, \forall b \in B \), so that the social welfare\(^3\) \( W \) of the market is maximised (Obj), subject to the problem constraints (ARC, BAC, BIC that will be explained later). Maximising the social welfare is a standard objective in combinatorial exchanges [107] and auctions. The allocation decision \( x_{sb} \) defines the couples of buyers and sellers who will exchange the goods, which is a binary variable defined as follows:

\[
x_{sb} = \begin{cases} 
1, & \text{if } C_b^d \text{ is allocated from } s \in S \text{ to } b \in B \\
0, & \text{otherwise}
\end{cases}
\]

We provide more details related to social welfare and problem constraints below.

**Problem Constraints**

There is a number of problem constraints that the allocation mechanism must satisfy:

\(^3\)The social welfare definition depends on the considered economic platform type: single or double-sided.
• **Available Resource Constraint (ARC):** is introduced in order to ensure that the sellers’ supplied capacities of goods are not exceeded for all types of goods.

• **Binary Allocation Constraint (BAC):** implies that a bid is allocated at most once. Please, note that the seller’s offer can satisfy more than a single buyer’s request.

• **Bid Indivisibility Constraint (BIC):** addresses the technical requirement of request indivisibility. BIC imposes that the bid can be either entirely allocated to a single seller or not allocated at all.

### Social Welfare

The social welfare defines the total value of satisfaction of the market participants after the market is cleared. It relies on the participants’ private valuation functions for the goods they own after allocation. Depending on the type of market economic platform, the social welfare is defined differently. In economic literature [107, 65, 18, 96, 100], the social welfare is typically derived based on the satisfaction/happiness of the competing sides of the market. Therefore, in a single-sided market it is based on buyers’ valuations only, while in a double-sided market, the social welfare is the welfare of both buyers and sellers. The buyers’ total welfare $W^b$ is determined as a sum of all the true price valuations of allocated buyers, as follows:

$$W^{ss} = W^b = \sum_{b \in B} \sum_{s \in S} p_b^i (\hat{C}_b^d) x_{sb}$$  \hspace{1cm} (3.1)

From the optimisation problem perspective, a good has to be allocated to the participant who values it the most. Hence, if a seller values her goods higher than the bidding buyers, the ownership rights will remain with the seller. Therefore, the seller’s welfare $W^s$ is determined as her true valuation for the goods that remain with her after allocation. The total seller’s welfare is expressed as a difference between their true price value for all the goods they offer in their asks, and their true price value for the allocated goods, as follows:

$$W^s = \sum_{s \in S} p_s^i (\hat{C}_s^a) - \sum_{s \in S} \sum_{b \in B} p_s^i (\hat{C}_b^d) x_{sb}$$  \hspace{1cm} (3.2)

Hence, the social welfare in a double-sided market is determined as follows:

$$W^{ds} = W^b + W^s = \left( \sum_{b \in B} \sum_{s \in S} p_b^i (\hat{C}_b^d) x_{sb} \right) + \left( \sum_{s \in S} p_s^i (\hat{C}_s^a) - \sum_{s \in S} \sum_{b \in B} p_s^i (\hat{C}_b^d) x_{sb} \right)$$
We can notice that, in this social welfare function, the argument $\sum_{s \in S} p^s_r(C^s_r)$ is a constant, which is independent of the allocation decision variable $x_{sb}$. Hence, this argument can be removed, and the objective function can be transformed into a simpler form, as follows:

$$W^{ds} = \left( \sum_{b \in B} \sum_{s \in S} p^d_b(C^d_b) \cdot x_{sb} \right) - \sum_{s \in S} \sum_{b \in B} p^s_r(C^s_r) \cdot x_{sb}$$

$$= \sum_{s \in S} \sum_{b \in B} \left( p^d_b(C^d_b) - p^s_r(C^s_r) \right) \cdot x_{sb}$$

(3.3)

Hence, the double-sided market aims to maximise the total market generated value, measured as a surplus produced by the allocated couples of buyers and sellers.

**Closeness to Standard Optimization Problem**

The formulated CIAP problem is a combinatorial optimisation problem [44]. In particular, a single-sided market can be considered as a special case of Multi-dimensional Knapsack Problem (MKP) [95], and a double-sided market can be viewed as a Multiple Multi-dimensional Knapsack Problem [128]. By analogy, the sellers are regarded as individual knapsacks, and the buyers represent the items to be allocated. In such case, the sellers’ offered capacities of goods can be considered as the knapsack dimensions, and the unit reservation price is a cost associated with each dimension occupied in the knapsack. The buyers’ requested bundles are the dimensions of the items, and the price valuations are the item values. Each item is indivisible and can be allocated only once, while each knapsack can fit multiple items. The goal is to allocate the items to knapsacks, maximising the social welfare. MKP and MMKP have non-deterministic polynomial time complexity (NP-hard) [121] and the optimal solutions cannot be found in a feasible time.

**3.3 Mechanism Design Framework**

**3.3.1 Basic Aspects**

A market mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ consists of two steps: allocation $\mathcal{A}$, and pricing $\mathcal{P}$. The allocation scheme $\mathcal{A}$ aims to produce the market allocation decisions, and select the couples of buyers and sellers who will exchange the resources. The pricing scheme $\mathcal{P}$ intends to establish the prices that the buyers need to pay $\mathcal{P}^b = \{\rho^b_1, \ldots, \rho^b_N\}$, and the payments that the sellers receive $\mathcal{P}^s = \{\rho^s_1, \ldots, \rho^s_M\}$. We provide the notations used in our mechanism design framework in Table 3.2.

While the private valuation functions show the individual value of the goods possessed
Table 3.2: Mechanism Design Framework Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta = {\beta_1, \ldots, \beta_N} )</td>
<td>a vector of all the buyers' private types</td>
</tr>
<tr>
<td>( \alpha = {\alpha_1, \ldots, \alpha_M} )</td>
<td>a vector of all the sellers' private types</td>
</tr>
<tr>
<td>( \beta_{-b} = {\beta_1, \ldots, \beta_{b-1}, \beta_{b+1}, \ldots, \beta_N} )</td>
<td>a vector of all buyers' types except for the buyer ( b \in B )</td>
</tr>
<tr>
<td>( \alpha_{-s} = {\alpha_1, \ldots, \alpha_{s-1}, \alpha_{s+1}, \ldots, \alpha_M} )</td>
<td>a vector of all sellers' types except for the seller ( s \in S )</td>
</tr>
</tbody>
</table>

by a participant, utility represents an additional monetary value generated after trade [106]. Hence, the utility of a market participant is a result of both allocation and pricing market outcomes. The buyer’s utility function is defined as a difference between her true price valuation and the buyer’s final price \( \hat{p}_b \) (i.e. granted discount) determined by the market, as follows:

\[
  u_b^b(\beta, \alpha) = \sum_{s \in S} \hat{p}_b^s(C_b^d) x_{sb} - \hat{p}_b^b
\]

The seller’s utility function is defined as the difference between the final seller’s payment and her true price reservation for the allocated resources (i.e. generated profit determined by the market):

\[
  u_s^s(\beta, \alpha) = \hat{p}_s^s - \sum_{b \in B} \hat{p}_s^b(C_b^d) x_{sb}
\]

### 3.3.2 Design Desiderata

A rational market participant aims to maximise her utility when trading in the market, which may involve strategic manipulation that can be harmful for the market’s socio-economic outcomes [111]. Selfish behaviour of participating traders, which aims to improve their individual utility, can cause the market to make inefficient allocation decisions, and disadvantage the other participants. Market mechanism design aims to achieve a number of desired properties, which define the efficiency, robustness to manipulation, and practical feasibility of the mechanism [107, 130]. We categorise the most important properties of the market mechanism design into (i) feasibility constraints that are essential in our mechanism design, and (ii) desirable properties, which can be approximated, if cannot be guaranteed.
Mechanism Design Feasibility Constraints

- **Individual Rationality (IR):** is a property concerned with each market participant individually. Individually rational market mechanisms ensure that the participating buyers and sellers can always obtain as much expected utility from trading in the market as if they avoid participation. Formally, buyer and seller individual rationality is defined as follows:

\[
\begin{align*}
  u^b_b(\beta, \alpha) &\geq \bar{u}^b_b(\beta_b) & \text{IR}^b \\
  u^s_s(\beta, \alpha) &\geq \bar{u}^s_s(\alpha_s) & \text{IR}^s
\end{align*}
\]

where \( u^b_b(\beta, \alpha) \), and \( u^s_s(\beta, \alpha) \) are the expected utilities of the participating buyer and seller, respectively. The expected utility for non-participation is considered to be zero: \( \bar{u}^b_b(\beta_b) = \bar{u}^s_s(\alpha_s) = 0 \). In other words, individual rationality implies that the market participants do not experience negative utility from participation. Given our definition of buyer’s and seller’s utility, we get:

\[
\begin{align*}
  \text{IR}^b = \sum_{s \in S} p^c_b(\tilde{C}_b) x_{sb} - \rho^b_b & \geq 0 \\
  \quad \text{if } x_{sb} = 0 & \Rightarrow \\
  \rho^b_b &\leq \sum_{s \in S} p^c_b(\tilde{C}_b) & \text{if } x_{sb} = 1 \\
  \text{IR}^s = \rho^s_s - \sum_{b \in B} p^c_s(\tilde{C}_b) x_{sb} & \geq 0 \\
  \quad \rho^s_s &\geq \sum_{b \in B} p^c_s(\tilde{C}_b) x_{sb}
\end{align*}
\]

In other words, in order to achieve buyers’ IR property, the mechanism has to determine such final buyer prices that, non allocated buyer pays nothing, and the final price of the allocated buyer does not exceed her price valuation. The sellers’ IR property is ensured, if the total seller’s payment is not lower than her price reservation for the allocated resources. IR property is considered to be the feasibility constraint in our mechanism design since IR mechanisms gain better participant’s trust and encourage participation.

- **Budget Balance (BB):** is a market property which guarantees that the market does not generate surplus, and does not run in deficit. In other words, the market determines such prices that the payments made by the buyers cover exactly the payments to the sellers. Formally, BB property is defined as follows:

\[
\sum_{b \in B} \rho^b_b = \sum_{s \in S} \rho^s_s
\]
• **Computational Tractability (CT):** implies that the market can calculate the outcome in a feasible amount of time. Optimal mechanisms that achieve allocative efficient results are often computationally intractable, and approximation mechanisms are needed to address this requirement. The cloud market mechanism design must be computationally tractable in order to be able to address the large-scale allocation and pricing problems.

**Desirable Mechanism Design Properties**

• **Allocative Efficiency (AE):** is a common objective in market mechanism design. Allocative efficient mechanisms determine the allocations, so that the market’s social welfare, i.e. the overall satisfaction across all participating buyers and sellers, is maximised. Achieving allocative efficiency requires determination of optimal allocation solution, which may be computationally hard; therefore, near-optimal approximation mechanisms, that exhibit feasible time complexity, are commonly designed.

• **Incentive Compatibility/Truthfulness:** The participants in markets may be tempted to reveal their private type untruthfully (e.g. overstate or understate their true price value for the traded goods). Misreporting and lying about the true types in order to increase the individual utility, can have a negative impact on the quality of allocation decisions. Truthful / Incentive Compatible market mechanisms guarantee that reporting a true type is a dominant strategy for the market participants. A mechanism \( \mathcal{M} \) is truthful, if a true type declaration of any market participant results in a higher utility, given the other participants’ declarations. Formally, it includes buyers and sellers truthfulness, defined as follows:

\[
\begin{align*}
    u^b_b((\hat{\beta}_b, \hat{\beta}_{-b}), \hat{\alpha}) & \geq u^b_b((\hat{\beta}_b, \hat{\beta}_{-b}), \hat{\alpha}) \quad \text{T}^b \\
    u^s_s((\hat{\beta}, \alpha_s, \alpha_{-s})) & \geq u^s_s((\hat{\beta}, \alpha_s, \alpha_{-s})) \quad \text{T}^s
\end{align*}
\]

where \( u^b_b((\hat{\beta}_b, \hat{\beta}_{-b}), \hat{\alpha}) \) is the utility of the buyer \( b \in B \) when she reports her private type truthfully \( \beta_b \), and \( u^b_b((\hat{\beta}_b, \hat{\beta}_{-b}), \hat{\alpha}) \) is the buyer’s utility for any declared type \( \hat{\beta}_b \). A similar formulation is applied for the sellers.
3.4 Approach Description

The problem of cloud infrastructure provisioning needs to be addressed by effective and efficient market mechanism design for various types of infrastructure cloud services and economic platforms. Based on the presented Conceptual Framework, we derive four market provisioning cases for trading homogeneous and heterogeneous infrastructure cloud services in a single-sided and double-sided markets. We address each use-case scenario by designing computational market mechanisms. Please, refer to Table 3.3. The general approach to address the described problem as well as some common aspects related to theoretical and experimental evaluation used in Chapters 4 and 5 are discussed below.

3.4.1 Allocation and Pricing Mechanisms

The considered Cloud Infrastructure Allocation Problem is NP-hard optimisation problem, which cannot be solved optimally in a feasible time. Therefore, we adapt greedy approximation approach for infrastructure cloud allocation. Greedy mechanisms are quick and often give good approximation to optimum [66, 21]. Furthermore, they are widely applied in multiple fields, such as mobile networking, auction design, machine learning, artificial intelligence, and business intelligence, which proves their practicality [21, 66]. We design and analyse a number of parametrised greedy heuristic functions for ranking the allocation candidates, and study their approximation quality.

The proposed pricing mechanisms apply Vickrey principles [144, 71, 39] and ideas aiming to give truthful incentives to the market participants. In particular, we aim to develop such pricing schemes where the determined participants’ payments are decoupled from their own declarations (declared types), and derived based on the reported types of the other agents in the market. In particular, the designed buyer payment schemes are based on critical-value, where the final payment is the minimum price required in order to

<table>
<thead>
<tr>
<th>Market Type</th>
<th>Service Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Homogeneous</td>
</tr>
<tr>
<td>Single-sided</td>
<td>Section 4.1</td>
</tr>
<tr>
<td>Direct Market</td>
<td>Section 4.2</td>
</tr>
<tr>
<td>Double-sided</td>
<td>Exchange Market</td>
</tr>
</tbody>
</table>
remain the market winner. The seller payments in the double-sided markets are based on the distribution of market accumulated surplus, where we propose and analyse different mixed distribution rules.

3.4.2 Common Aspects of Evaluation Principles

Theoretical Investigation

In our theoretical investigation of the market properties of the proposed market mechanisms, we provide proofs of guaranteed economic desiderata. For a market mechanism truthfulness, we show that the minimum required truthfulness conditions hold. In particular, we prove that the allocation function is monotone and the pricing function determines the prices based on critical-value [90, 105]. The truthfulness are based on the following definitions.

**Definition 3.4.1 (Monotonicity).** A monotone allocation function implies that (i) a winning bid cannot lose by offering a higher price valuation for a smaller requested bundle, and (ii) a losing bid cannot win by offering a lower price valuation for a bigger requested bundle. Formally, given two reported bids of the same buyer $\hat{\beta}_{b*} = (C_{b*}, \hat{p}_{b*})$, and $\hat{\beta}_b = (C_b, \hat{p}_b)$, s.t. $\hat{\beta}_{b*} \geq \hat{\beta}_b$, meaning that $\hat{p}_{b*} \geq \hat{p}_b$, and $C_{b*} \geq C_b$, where $c_{b*}^g \leq c_b^g \forall g \in G$, we need to prove that the two following relationships hold:

- if $\sum_{s \in S} x_{sb} = 1$ for $\hat{\beta}_b$, then the allocation for $\hat{\beta}_{b*}$ would result in $\sum_{s \in S} x_{sb*} = 1$
- if $\sum_{s \in S} x_{sb*} = 0$ for $\hat{\beta}_{b*}$, then the allocation for $\hat{\beta}_b$ would result in $\sum_{s \in S} x_{sb} = 0$

**Definition 3.4.2 (Critical-value).** When a mechanism has a monotone allocation function, each bid would have a unique value, called critical-value, such that the reported valuation above would result in the buyer winning the requested goods, but any valuation below would make the buyer lose. In other words, critical-value pricing implies that the mechanism determines the minimum prices required for the buyer to conform to the determined allocation decision: specifically, the losing buyer has to pay nothing, and the allocated buyer has to pay the minimum price required for her to remain the winner. Formally, we need to show that any price valuation below the final price would make the buyer lose, while a greater value results in the bid being allocated:

$$
\begin{aligned}
\hat{p}_b^* < \rho_b^b & \Rightarrow \sum_{s \in S} x_{sb} = 0 \\
\hat{p}_b^* \geq \rho_b^b & \Rightarrow \sum_{s \in S} x_{sb} = 1
\end{aligned}
$$
Experimental Investigation

Due to the unavailability of the publicly-revealed information about the cloud services requests and offers in the infrastructure cloud market, we use large-scale extensive simulation approach, where the designed mechanisms are evaluated in a variety of simulated market scenarios.

We approach the mechanisms’ investigation from both the buyer’s and seller’s perspectives, and analyse the socio-economic dynamics in the designed market mechanisms. From the buyer’s viewpoint we are concerned with such aspects as the the number of addressed buyer requests, the discounts granted in the market and the power that the seller has over the market outcomes, including allocation and pricing decisions. From the seller’s perspective, we investigate the achieved utilisation of resources, total profit from trade, as well as the opportunities provided by misreporting, including the best strategies for improving the individual utility.

The developed experimental environment is in Java programming language using IBM CPLEX solver library [76] for the optimal mechanism implementation. The simulations are conducted on an Intel 2.20GHz with 2GB RAM instances with Linux OS, which is part of Swinburne High-Performance Computing system.

3.5 Summary

This chapter has described a conceptual framework for the marketplace of infrastructure cloud services, where the market has to allow trading of homogeneous and heterogeneous cloud services in single-sided and double-sided markets. The problem of Cloud Infrastructure Allocation and Pricing has been formulated as Integer Program. Our investigation has revealed that the considered problem can be considered as a Multiple Multi-dimensional Knapsack Problem, which is strongly NP-complete. We have identified the design desiderata, where Individual Rationality, Budget Balance, and Computational Tractability are the mechanism design feasibility constraints that have to be achieve in our mechanism design, and Allocative Efficiency and Incentive Compatibility/Truthfulness are the desirable mechanism design properties that can be approximated. Finally, we have described the common approaches applied in our mechanism design; in particular, greedy approximation is used for allocation mechanisms, and Vickrey principles and the critical-value pricing approach are applied for price-determination. In the next chapters, we design and evaluate the market mechanisms for trading infrastructure cloud services.
Chapter 4

Market Mechanisms for
Homogeneous Infrastructure Cloud Services

In this chapter, we investigate the market of Homogeneous Infrastructure Cloud Services in two settings: a single-side market with a single infrastructure cloud service provider and multiple competing buyers (Section 4.1) and a double-sided fully competitive open cloud exchange market including multiple providers and multiple consumers (Section 4.2). In both sections, we (i) describe our computational mechanisms for cloud services allocation and pricing, (ii) provide illustrative examples of mechanism’s operational procedure, (iii) conduct theoretical investigation of the proposed algorithms’ economical properties, including Individual Rationality, Budget Balance, Computational Tractability and Buyer Truthfulness, and (iv) investigate the allocative performance and strategic manipulation opportunities of the market mechanisms in extensive simulation experiments. Finally, we provide the summary of the designed market mechanisms, including the major design aspects and the qualitative performance characteristics.
4.1 Single-sided Market Mechanism

In this section, we design a computational market mechanism for homogeneous infrastructure cloud services trading in the market with a single seller and multiple competing buyers. Please, recall that homogeneous infrastructure cloud services are the virtual machines of pre-defined configurations of various sizes, e.g. small, medium, large, that can be traded as combinatorial bundles of services of different types.

4.1.1 Allocation Mechanism

The allocation scheme proposed for a single-sided market of homogeneous infrastructure cloud services is given in Algorithm 1. The allocation scheme aims to determine the final allocation decision \( X \) for all the submitted buyers, given a vector of buyer bids \( \hat{\beta} \), and the seller’s ask \( \hat{\alpha} \). The mechanism also returns the sorted list of allocation candidates \( L \), which is used by the pricing scheme in its operation, in order to reduce the mechanism’s time complexity.

Algorithm Description

The allocation mechanism needs to determine a subset of bids that will obtain the goods offered by the cloud provider. It is a non-trivial problem, given the combinatorial nature of the buyers’ requests, and a feasible time complexity requirement. In order to address the problem, we apply the idea of greedy allocation, which aims to rank the allocation candidates (i.e. bids) based on some simple value, called sorting criteria value. Therefore, our proposed allocation scheme runs in a number of steps:

- **Initialization Phase (lines 3-4):** Firstly, the algorithm initializes the variables required for its operation, such as the final allocation decision vector \( X \), and the list \( L \) which is used to keep the allocation candidates in the sorted order.

- **Sorting Phase (line 5):** Secondly, the allocation scheme sorts the bids in the list \( L \) in decreasing order of their sorting criteria values, determined by the sorting criteria function \( e(\hat{\beta}_b) \). This function aims to determine such values associated with each allocation candidate, that would result in a social-welfare maximising sorting order. A detailed explanation of the proposed sorting criteria function and its relationship to the problem’s objective is given below.

- **Allocation Phase (lines 6-13):** Finally, the mechanism considers the bids in the sorted order in the list \( L \) and makes the allocation decisions for all the candidates. If all
Algorithm 1 Allocation Scheme for Single-sided Market of Homogeneous Cloud Infrastructure Services (HO-SS-A)

1: **Input:** \( \hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N) \), where \( \hat{\beta}_b = (\hat{C}_b^d, p_b^i) \)

2: **Input:** \( \hat{\alpha} = (\hat{C}_s^d, \hat{P}_s^r) \)

### # Step 1: Initialization Phase
3: \( X \leftarrow \emptyset \) // final allocation decision vector
4: \( L \leftarrow \hat{\beta} \) // list of allocation candidates for re-ordering

### # Step 2: Sorting Phase
5: Re-order \( \hat{\beta}_b \) in list \( L \) in decreasing order of \( e(\hat{\beta}_b) \)

### # Step 3: Allocation Phase
6: for all \( \hat{\beta}_b \) in \( L \) do
7: if \( \text{ARC}(\hat{\beta}_b, \hat{\alpha}, X) \& \text{RPC}(\hat{\beta}_b, \hat{\alpha}) \) then
8: \( x_b = 1 \)
9: else
10: \( x_b = 0 \)
11: end if
12: \( X \leftarrow X \cup x_b \)
13: end for
14: **Output:** \( X, L \)

the problem constraints are satisfied, the bid is allocated and the buyer receives the requested bundle of services \( \hat{C}_b^d \); otherwise the bid is denied by the mechanism and the buyer receives nothing. The mechanism verifies two constraints:

- **Available Resource Constraint** (ARC) is verified in order to ensure the availability of sufficient resources to satisfy the buyer’s request, as follows:

\[
\hat{c}_b^d \leq \hat{c}_s^{sa}, \forall g \in G, \text{ where } \hat{c}_s^{sa} = \hat{c}_s^s - \sum_{b \in B} \hat{c}_b^d x_b
\]

- **Reserve Price Constraint** (RPC) is used to ensure that the seller’s minimum prices are satisfied by the buyer’s price valuation, as follows:

\[
\hat{p}_b^v \geq \hat{p}_b^{br}, \text{ where } \hat{p}_b^{br} = \sum_{g \in G} \hat{c}_b^d \hat{p}_g^{ur}
\]

**Sorting Criteria Function**

The sorting criteria function \( e(\hat{\beta}_b) \) is used in order to determine the sorting criteria value, which allows the mechanism to rank the candidates for allocation in the list \( L \). So as to determine the efficient allocation results, the mechanism has to produce the social welfare-
maximising sorting order, where more socially efficient bids are preferred and placed at the higher positions in the sorted list \( L \). Therefore, an efficient sorting criteria function has to be tightly linked with the problem’s objective function. A single-sided market mechanism aims to maximise the sum of price valuations of all the allocated buyers (Formula 3.1). Hence, the more socially efficient bid would be the one that offers a greater price valuation, while requesting less resource from the market. Our proposed sorting criteria function is defined as follows:

\[
e(\hat{\beta}_b) = \frac{\hat{p}_v}{w(\hat{\beta}_b)}, \text{ where } w(\hat{\beta}_b) = \left( \sum_{g \in G} c_{bg} f_{r_g} \right)^m, m \geq 0
\]

This sorting criteria function is based only on the bid’s declared type \( \hat{\beta}_b \), which includes the price valuation \( \hat{p}_v \) and the requested bundle of services \( \hat{C}_d \). The idea is to determine the price value contributed by a bid per unit of traded good. We use the resource relationship factor in order to get a homogeneous view of the requested bundle, which allows to obtain comparable values across the combinatorial bids with unique bundles. Therefore, the function calculates the weight of the requested bundle \( w(\hat{\beta}_b) \) as a sum of requested capacities of goods scaled according to the goods relationship \( F^r = \{ f_1^r, \ldots, f_K^r \} \), where \( f_g^r \) - is a relationship factor for the good type \( g \in G \). Such sorting criteria function considers a bid to be more efficient if it offers a greater price valuation for fewer requested goods.

The parameter \( m \geq 0 \) has an impact on the entire calculated bundle weight. It is used to allow controlling the significance of the traded resource size. Please, refer to Figure 4.1 for a simple illustrative example of parameter \( m \) impact. In the presented example,
we consider two bids, which have different bundle size ($\hat{\beta}_2$ requests a bigger bundle of services) and an equal linear sorting criteria value, when $m = 1.0$ (we consider the following goods relationship $F^r = (1, 2, 4)$). We plot the relative competitiveness of the two bids derived based on their sorting criteria values. We can clearly see that when $m < 1.0$, the mechanisms gives a competitive advantage the larger request $\hat{\beta}_2$, while $m > 1.0$ makes the sorting criteria function favour smaller bid $\hat{\beta}_1$. In other words, adjusting parameter $m$ 0 indicates to the mechanism which kind of buyer bids to favour: the ones that request a larger or smaller bundle of services. While the parameter $m$ does not guarantee the improved social welfare, or the better resource utilization, it provides flexibility to the market owner, by allowing to choose the types of preferred buyers. Please, note that a non-linear sorting criteria value (i.e. $m \neq 1$) does not guarantee split- or merge-proofness [99]. As we could see, when the conversion is large, smaller buyer requests are favoured by the mechanism, meaning that the buyer can decompose her bids into a number of smaller ones in order to gain the competitive advantage. However, given the complementarity of resources in the requested bundle, splitting the bid into a number of smaller ones has an associated risk. The buyer may end up with only some of her bids being granted, and not all the desired amount of resources.

**Allocation Mechanism Example**

We illustrate the way our allocation mechanism works by giving a simple example. We assume, that a cloud provider offers general purpose VMs of three different sizes: small, medium, and large. The VM types and the corresponding resource configurations are provided in the Table 4.1a. According to the VMs composition in terms of their underlying resources, the following strict relationship emerges: Large = $2 \times$ Medium = $4 \times$ Small, which is reflected in the mechanism’s relationship factor $F^r = (1, 2, 4)$. The seller offers limited amounts of VMs of each type and reports a minimum price for a unit of each VM type, i.e. unit price reservations (see Table 4.1b). A number of prospective cloud consumers, from large companies to individuals, join the market in order to procure the traded cloud infrastructure services. In our example, six buyers enter the market with their different cloud infrastructure requirements and price valuations: two large consumers (LC), two medium consumers (MC), and two phd students (PhD). The cloud provider’s offer and the consumers’ requests are depicted in Table 4.1c.

The allocation procedure of our proposed single-sided mechanism for trading homogeneous cloud services in this example is depicted in Table 4.2. The allocation scheme
receives the specified seller’s offer, and the buyer requests as the input. The bids sorting order is determined based on the proposed sorting criteria function. For simplicity, we consider a linear form of our proposed sorting criteria function, where $m = 1.0$. The weights of the buyer’s requested bundles are determined given the goods relationship factor $F^{r} = (1, 2, 4)$, and the sorting criteria value is further calculated for each participating buyer. For instance, the sorting criteria value of the LC1 is determined as follows:

$$e(LC1) = \frac{9.0}{(0 \times 1 + 0 \times 2 + 6 \times 4)^{1.0}} = 0.375$$

The allocation candidates are reordered in the list $L$ in decreasing order of their sorting criteria values. In our example, LC1 is the most competitive candidate, which is ordered at the first position in the sorted list $L$ and she is the first to be considered for allocation. In order to make the allocation decision for a candidate, the mechanism verifies whether there is enough resource available to satisfy the candidate’s requested bundle (ARC), and checks if the reported price valuation satisfies the seller’s desired bundle price reservation.
Table 4.2: HO-SS Mechanism Example: Allocation Procedure ($m = 1.0$)

<table>
<thead>
<tr>
<th>Sorted List, $L$</th>
<th>Values</th>
<th>Constraints</th>
<th>$x_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w(\hat{\beta}_b)$</td>
<td>$e(\hat{\beta}_b)$</td>
<td>$\bar{C}^d_b$</td>
<td>$\bar{C}^{sa}_b$</td>
</tr>
<tr>
<td>LC1</td>
<td>24.0</td>
<td>0.375</td>
<td>(0, 0, 6)</td>
</tr>
<tr>
<td>MC1</td>
<td>12.0</td>
<td>0.350</td>
<td>(4, 4, 0)</td>
</tr>
<tr>
<td>LC2</td>
<td>22.0</td>
<td>0.250</td>
<td>(0, 1, 5)</td>
</tr>
<tr>
<td>PhD1</td>
<td>1.0</td>
<td>0.220</td>
<td>(1, 0, 0)</td>
</tr>
<tr>
<td>MC2</td>
<td>8.0</td>
<td>0.200</td>
<td>(8, 0, 0)</td>
</tr>
<tr>
<td>PhD2</td>
<td>4.0</td>
<td>0.025</td>
<td>(4, 0, 0)</td>
</tr>
</tbody>
</table>

We can see that both problem constraints are satisfied, and the mechanism selects LC1 as a market winner. The process repeats for all the bids in the sorted list $L$.

As a result, the mechanism has determined three winning bids (i.e. LC1, MC1, and PhD1), which will receive the goods from the cloud provider. The remaining three bids were rejected by the market either due to the insufficient amounts of available goods (i.e. LC2, MC2), or the low price valuation, which does not satisfy the seller’s desired minimum price (i.e. PhD2). The losing buyers do not receive any goods from the provider.

The mechanism allocated the following amounts of goods $\langle 0, 0, 6 \rangle + \langle 4, 4, 0 \rangle + \langle 1, 0, 0 \rangle = \langle 5, 4, 6 \rangle$; the proportion of the allocated goods to the corresponding seller’s initially supplied amounts, i.e. $\langle 50\%, 40\%, 60\% \rangle$ is referred to as resource utilization. The social welfare produced by the mechanism is $W = $9.0 + $4.2 + $0.22 = $13.42. Please, note that although the proposed greedy allocation has reached an optimal solution in the considered example, it is an approximation mechanism and the optimal allocation is not guaranteed. This example demonstrates that a highly efficient allocation can be determined by our proposed heuristic mechanism. A detailed analysis of the mechanism’s approximation quality is presented in experimental section, where we conduct a large scale extensive simulation experiment.

### 4.1.2 Pricing Mechanism

The proposed pricing scheme is given in Algorithm 2. The pricing scheme aims to determine the final prices that the buyers have to pay $\mathcal{P}^b$, given a vector of buyer bids $\hat{\beta}$, the seller’s ask $\hat{\alpha}$, and the previously determined allocation decision $X$ and the sorted list of allocation candidates $L$. 
Algorithm Description

The pricing mechanism needs to determine the final prices that all the participating buyers pay for the resources granted to them by the allocation scheme. The mechanism has to determine the prices based on the laws of supply and demand, which means that the buyer prices are derived based on the current competition for resources in the market.

Our pricing scheme determines the final prices for all the participating buyers, where the rejected ones are not required to pay (lines 7, 26). The final prices that the allocated buyers pay for the granted resources are determined based on the competition in the market. The idea is to determine the minimum prices that they could have offered in order to remain the market winners, called critical-value payments. Therefore, the pricing mechanism derives the prices based on the declarations of the first losing competitors associated with each winning buyer. In order to determine such competitors and calculate the final prices, our proposed pricing mechanism runs in a number of steps. We describe the price determination procedure repeated for each winning buyer individually.
Algorithm 2 Pricing Scheme for Single-sided Market of Homogeneous Cloud Infrastructure Services (HO-SS-P)

1: Input: $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$, where $\hat{\beta}_b = (C_d^b, p^b)$
2: Input: $\hat{\alpha} = (\hat{C}_s^b, P_s^b)$
3: Input: $X = \{x_b\}, \forall b \in B$
4: Input: $L = \{\hat{\beta}_b\}, \forall b \in B$  // sorted list of bids
5: $\mathcal{P}^b \leftarrow \emptyset$
6: for all $\hat{\beta}_b \in \hat{\beta}$ do
7:   $\rho^b = 0$
8:   if $x_b = 1$ then
9:     # Step 1: Initialization Phase
10:    $\check{X} \leftarrow \emptyset$  // auxiliary allocation result
11:    $\check{L} \leftarrow \{\hat{\alpha}_\sigma | \sigma \neq b \land \forall \hat{\alpha}_\sigma \in L\}$  // all bids except $\hat{\beta}_b$ in sorted order
12:    $\check{L} \leftarrow \check{L} \cup \{\hat{\beta}_b^{br}\}$  // insert bundle reservation bid in order of $e(\hat{\beta}_b)$
13:    $\hat{\beta}_b^c = \hat{\beta}_b$
14:    break
15:   end if
16:   $x_b = 1$
17:   $\check{X} \leftarrow \check{X} \cup \check{x}_b$
18: end if
19: end for
20: # Step 2: Competitor Determination Phase
21: for all $\check{\beta}_b$ in $\check{L}$ do
22:   if $\text{ARC}(\hat{\beta}_b, \check{\alpha}, \check{X}) \& \text{RPC}(\hat{\beta}_b, \check{\alpha})$ then
23:     if $\text{ARC}(\hat{\beta}_b, \check{\alpha}, \check{X}) \& \text{RPC}(\hat{\beta}_b, \check{\alpha})$ then
24:       if $(x_b = 0) \lor (\hat{b} = b)$ then
25:         $\check{\beta}_b^c = \hat{\beta}_b$
26:       end if
27:     end if
28:   end if
29: end for
30: # Step 3: Price Determination Phase
31: $\rho^b_{\check{\beta}_b} = \{w(\hat{\beta}_b) \times e(\hat{\beta}_b^c)\}$
32: end if
33: $\mathcal{P}^b \leftarrow \mathcal{P}^b \cup \rho^b_{\check{\beta}_b}$
34: end for
35: Output: $\mathcal{P}^b = \{\rho^b_1, \ldots, \rho^b_N\}$

- Initialization Phase (lines 8-11): Firstly, the mechanism initialises the auxiliary allocation decision vector $\check{X}$ used to keep track of an ongoing auxiliary allocations; and constructs a new sorted list of candidates $\check{L}$. This list contains all the bids in the previously determined sorted order\(^1\), except for the currently considered winning bid $\hat{\beta}_b$ (line 10), which is replaced by its bundle-reservation bid $\hat{\beta}_b^{br}$ (line 11), and inserted (using binary search) in the list $\check{L}$ at the position, so that the sorted order based on $e(\hat{\beta}_b)$ is respected.

\(^1\)Please, note that reusing the initially sorted order of bids in the list $L$, allows to improve the time complexity of the mechanism.
A bundle reservation bid $\hat{\beta}_{br}^b = (\hat{\beta}_b, \hat{\beta}_{br})$ is a special type of candidate, which requests the same bundle of services as $\hat{\beta}_b$, but the buyer’s initial price valuation $\hat{\beta}_{pv}^b$ is replaced by the seller’s bundle-specific price reservation $\hat{\beta}_{br}^b = \sum_{g \in G} \hat{\beta}_{pr}^{gr} \hat{\beta}_{bg}^g$. It is the least competitive price valuation value which could be used to satisfy the seller’s desired minimum price for the bundle.

- **Competitor Determination Phase (lines 12-23):** Secondly, the mechanism searches for the first losing competitor $\hat{\beta}_c$ of the currently considered winning bid $\hat{\beta}_b$. Therefore, the pricing mechanism runs a new allocation phase with a new sorted list of candidates $\bar{L}$. The goal is to identify the first competitor with a unique competition condition as described below:

  - **Allocation feasibility condition** (lines 13-14): The bid $\hat{\beta}_b$ is considered to be a competitor of the bid $\hat{\beta}_b$, if both can be allocated (i.e. satisfy ARC and RPC) when considered at a particular position in the sorted list. Please, note that the auxiliary allocation decision variable $\hat{X}$ is updated according to the ongoing new allocation of the candidates in the list $\bar{L}$ (lines 17, 18) to keep track of available goods.

  - **Unique competition condition** (line 15): The mechanism aims to locate the first losing competitor, which exhibits a unique competition condition, compared to the initial allocation. Such competitor may be either (i) a bid which initially lost but would have won, if the currently considered winner did not participate in the market ($x_b = 0$), or (ii) a bundle reservation bid of the currently considered winner ($b = \hat{b}$). This bid becomes a competitor only if there is no other competitor for the bid $\hat{\beta}_b$, which signifies that in order to become a winner it is sufficient to bid the reserve price.

Once the first losing competitor is found, the mechanism moves to the next phase.

- **Price Determination Phase (line 24):** Finally, the mechanism determines the final price for the winning buyer bid $\hat{\beta}_b$ based on the declaration of the associated first losing competitor $\hat{\beta}_c$. Therefore, the mechanism scales the bundle weight of the currently considered winner according to the sorting criteria value of her first losing competitor. It allows to determine the minimum price for $\hat{\beta}_b$ to outbid her competitor.
## Pricing Mechanism Example

We continue the initial example, which was used to illustrate the procedure of our proposed allocation scheme. Once the market winners are determined, the mechanism aims to establish the prices that the buyers have to pay. The pricing procedure of our proposed single-sided mechanism for trading homogeneous cloud services is depicted in Table 4.3.

The final buyer prices are determined for all the participating buyers in order. If a buyer was initially allocated, the mechanism searches for her first losing competitor bid. Therefore, in the considered example, the mechanism constructs a new sorted list of bids $\tilde{L}$ for the winning bid $LC1$. As you can see, this list contains all the initial bids but the bundle-reservation candidate $LC1BR$ replaces $LC1$. Given the sorted order in $L$, the mechanism

<table>
<thead>
<tr>
<th>Buyer, $b$ $x_b$</th>
<th>New Sorted List, $\tilde{L}$</th>
<th>First Losing Competitor, $\beta^c$</th>
<th>Final Price, $\rho^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LC1$ 1</td>
<td>$MC1$ 0.350 1 1</td>
<td>$LC2$</td>
<td>$0.250 \times 24 = $6.00$</td>
</tr>
<tr>
<td></td>
<td>$LC2$ 0.250 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$PhD1$ 0.220 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MC2$ 0.200 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$LC1BR$ 0.075 - 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$PhD2$ 0.025 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$MC1$ 1</td>
<td>$LC1$ 0.375 1 1</td>
<td>$MC2$ 0.200 0 0</td>
<td>$0.200 \times 12 = $2.40$</td>
</tr>
<tr>
<td></td>
<td>$LC2$ 0.250 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$PhD1$ 0.220 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MC2$ 0.200 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MC1BR$ 0.070 - 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$PhD2$ 0.025 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$LC2$ 0</td>
<td>- - - - -</td>
<td>-</td>
<td>$$0.0$</td>
</tr>
<tr>
<td></td>
<td>$LC1$ 0.375 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MC1$ 0.350 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$PhD1$ 1</td>
<td>$LC2$ 0.250 0 0</td>
<td>$PhD1BR$ 0.06 $\times 1$</td>
<td>$$0.06$</td>
</tr>
<tr>
<td></td>
<td>$MC2$ 0.200 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- - - - -</td>
<td></td>
<td>$$0.0$</td>
</tr>
<tr>
<td>$MC2$ 0</td>
<td>- - - - -</td>
<td>-</td>
<td>$$0.0$</td>
</tr>
<tr>
<td></td>
<td>$PhD2$ 0.025 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$PhD2$ 0</td>
<td>- - - - -</td>
<td>-</td>
<td>$$0.0$</td>
</tr>
</tbody>
</table>
determines the new allocation outcomes, aiming to find the first losing competitor. For LC1, the mechanism selects LC2 as the competitor, since it satisfies the unique competitor condition: it was initially denied in \( L \) but got allocated in a new setting \( \tilde{L} \). Hence, the final price that LC1 pays is determined by scaling her bundle weight according to the sorting criteria value of LC2 as follows:

\[
e(LC2) \times w(LC1) = 0.250 \times (0 \times 1 + 0 \times 2 + 6 \times 4)^{1.0} = 6.0
\]

It is the minimum price that the LC1 could have declared to the market in order to outbid LC2 and win. Similarly, the mechanism determines the first losing competitor for MC1 and the final price is calculated is a similar fashion. Please note that LC1, and MC1 have different first losing competitors since they request different scarce resource types. While LC1, and LC2 compete for Large VMs, given that their aggregated demand is greater than the supply for this resource type \((6 + 5 > 10)\), the buyers MC1, and MC2 compete for Small VMs \((8 + 4 > 10)\). In case of PhD1, her corresponding bundle reservation bid PhD1RB is her first losing competitor, since there is no competitor for PhD1. Hence, the buyer could be granted the requested goods even with the lowest possible price. Therefore, the final price determined based on the sorting criteria value of the bundle reservation bid is always the reserve price. Finally, the unallocated buyers do not pay anything.

4.1.3 Theoretical Investigation

In this section we study the designed HO-SS market mechanism theoretically and analyse the following economic properties: Individual Rationality, Budget Balance, Computational Tractability, and Truthfulness. The provided proofs are based on the mathematical formulations and definitions introduced in Chapter 3.

**Individual Rationality (IR)**

**Theorem 1.** The proposed pricing scheme maintains Individual Rationality property for single-minded buyers declared types.

**Proof.** We need to show that IR property holds as defined in Formula 3.4. Since, our proposed pricing scheme does not charge losing buyers, we focus on the allocated buyer pricing, i.e. second inequality. We assume that it does not hold as follows \( \rho_b^k > \hat{\rho}_b^c \) and
provide the proof by contradiction. In such a situation, we get the following:

\[ e(\hat{\beta}^c) \times w(\hat{\beta}_b) > \hat{p}_b^r \implies e(\hat{\beta}^c) > \frac{\hat{p}_b^r}{w(\hat{\beta}_b)} \implies e(\hat{\beta}^c) > e(\hat{\beta}_b) \]

It means that the currently considered winning bid \( \hat{\beta}_b \) is less competitive than her first losing competitor \( \hat{\beta}^c \). Therefore, the sorted lists \( L \) and \( \bar{L} \) would be identical at least until the position \( z \leq N \) of the first losing competitor \( \hat{\beta}^c \):

\[
\bigcup_{i=1}^{z} L_i = \bigcup_{i=1}^{z} \bar{L}_i \implies X_z = \bar{X}_z
\]

As a result, the allocation decision variables \( X_z \) and \( \bar{X}_z \) must have been the same at least until the candidate in the position \( z \leq N \). Thus, the allocation decision for the first losing bid \( \hat{\beta}^c \) would not have been different in \( X \) and \( \bar{X} \), which contradicts our unique competitor condition. Therefore, the final buyer’s price is always lower than her reported price valuation: \( \rho_b^k \leq \bar{p}_b^r \).

Hence, the proposed pricing mechanism is IR for single-minded buyers.

\[ \blacksquare \]

**Theorem 2.** The proposed pricing scheme maintains Individual Rationality property for the sellers’ reported types.

**Proof.** We need to show that IR property holds as defined in Formula 3.5. We will show that IR property for the seller holds by proving that the final price of any allocated buyer cannot be lower than the seller’s bundle price reservation: \( \rho_b^k \geq \bar{p}_b^r x_b, \forall b \in B \). We assume that it is false (i.e. \( \rho_b^k < \bar{p}_b^r \), where \( x_b = 1 \)) and provide proof by contradiction. In such situation, we get:

\[ w(\hat{\beta}_b) \times e(\hat{\beta}^c) < \bar{p}_b^r \implies e(\hat{\beta}^c) < \frac{\bar{p}_b^r}{w(\hat{\beta}_b)} \implies e(\hat{\beta}^c) < e(\hat{\beta}_b^b) \]

It means that the first losing competitor \( \hat{\beta}^c \) was less competitive than the bundle reservation bid \( \hat{\beta}_b^b \). As a result, the bundle reservation bid would have appeared earlier in the sorted list \( \bar{L} \), and would have been selected instead. Hence, the lowest possible final price of a winning bid can be derived based on her bundle reservation bid: \( \rho_b^k \geq \bar{p}_b^r \). Consequently,
given that the seller’s payment is the sum of all the buyers’ payments, we conclude:

\[
\begin{align*}
\rho^s_b &\geq \hat{p}_b x_b, \forall b \in B \\
\rho^s &\geq \sum_{b \in B} \rho^b_b x_b
\end{align*}
\]

Hence, the proposed pricing mechanism is IR for the seller’s reported type.

\[\square\]

**Budget Balance (BB)**

We know that all the payments collected from the buyers go entirely to the seller. Formally, the seller payment is defined as follows:

\[
\rho^s = \sum_{b \in B} \rho^b_b
\]

We can see that it complies with the budget balance definition (Formula 3.6). Hence, the proposed market mechanism maintains Budget Balance property.

**Computational Tractability (CT)**

The proposed single-sided market mechanism for trading homogeneous cloud services clears in polynomial time in the size of its input. Specifically, the HO-SS-A mechanism clears in \(O(N \log N)\) time, required for sorting the allocation candidates in the list \(L\). The HO-SS-P scheme determined buyers’ prices in \(O(N^2)\) time due to the reconsidered initial order of the allocation candidates in the list. Therefore, the market mechanism’s time complexity is \(O(N^2)\) which is improved compared to the time complexity of the approaches proposed in literature, typically being \(O(|N^2 \log |N||)\).

**Truthfulness (T)**

We derive truthfulness by proving that the minimum required truthfulness conditions, outlined in [90], hold. In particular, we prove that the designed allocation mechanism in monotone in buyers’ declarations, and the proposed pricing scheme determines critical-value payments.

**Theorem 3.** The proposed allocation scheme is monotone in declarations of single-minded buyers.

*Proof.* The provided proof is derived based on Definition 3.4.1. We consider two reported
Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

bids of the same buyer $\hat{\beta}_{b*} = (\hat{C}^d_{b*}, \hat{p}^v_{b*})$, and $\hat{\beta}_{b} = (\hat{C}^d_{b}, \hat{p}^v_{b})$, s.t. $\hat{\beta}_{b*} \geq \hat{\beta}_{b}$. When changing the declared type from $\hat{\beta}_{b*}$ to $\hat{\beta}_{b}$, the sorted list of bids may differ only in the changed bid moving up or down in the list (all the other declarations remain unchanged, and preserve their positions). We denote by $L^*$, and $L^-$, the sorted lists of bids with $\hat{\beta}_{b*}$, and $\hat{\beta}_{b}$, respectively. We assume that the monotonicity property does not hold, and provide the proof by contradiction.

**Scenario 1:** We consider a situation when $\hat{\beta}_{b}$ was allocated $x_{b} = 1$, but its changed type $\hat{\beta}_{b*}$ had lost $x_{b*} = 0$. Hence, the bid $\hat{\beta}_{b*}$ must have violated at least one of the two problem constraints:

- **RPC:** Since $\hat{\beta}_{b}$ was allocated $x_{b} = 1$, it has satisfied the RPC, i.e. $\hat{p}^v_{b} \geq \hat{p}^v_{b*}$. Given that $\hat{c}^d_{b*g} \leq \hat{c}^d_{b*g}, \forall g \in G$, we get that $\hat{p}^v_{b} \geq \hat{p}^v_{b*}.$

$$
\begin{align*}
\hat{p}^v_{b} &\geq \hat{p}^v_{b*}, \\
\hat{p}^v_{b} &\leq \hat{p}^v_{b*} \\
\hat{p}^v_{b} &\geq \hat{p}^v_{b*}
\end{align*}
$$

Hence, $\hat{\beta}_{b*}$ would also satisfy RPC.

- **ARC:** Since $\hat{\beta}_{b}$ was allocated, it means that there was enough available resource to satisfy her requested bundle $\hat{C}^d_{b}$ at the considered position in the sorted list $L^-$. The position of $\hat{\beta}_{b}$, in the list $L^*$ is defined by its sorting criteria function and since $\hat{\beta}_{b*} \geq \hat{\beta}_{b}$, we get:

$$
\begin{align*}
\hat{p}^v_{b} &\geq \hat{p}^v_{b*}, \\
\hat{c}^d_{b*g} &\leq \hat{c}^d_{b*g}, \forall g \in G
\end{align*}
\Rightarrow 
\left( \frac{\sum_{g \in G} \hat{c}^d_{b*g} f^r_{g}}{m} \right)^m \geq \left( \frac{\sum_{g \in G} \hat{c}^d_{b*g} f^r_{g}}{m} \right)^m \Rightarrow e(\hat{\beta}_{b*}) \geq e(\hat{\beta}_{b})
$$

Hence, $\hat{\beta}_{b*}$ would appear in the same or better position in the sorted list $L^*$ compared to $\hat{\beta}_{b}$ in the list $L^-$. In this case, $\hat{\beta}_{b*}$ is guaranteed to satisfy ARC because its requested bundle is smaller $\hat{C}^d_{b*} \geq \hat{C}^d_{b}$ and at a higher position in the list there can be only more resource available.

We have shown that if a bid $\hat{\beta}_{b}$ is allocated $x_{b} = 1$, then the modified bid $\hat{\beta}_{b*}$, s.t. $\hat{\beta}_{b*} \geq \hat{\beta}_{b}$ would also satisfy all the problem constraints, and would also be allocated $x_{b*} = 1$.

**Scenario 2:** We consider the opposite situation, when a bid $\hat{\beta}_{b*}$ was denied $x_{b*} = 0$, and the modified bid $\hat{\beta}_{b}$ was allocated $x_{b} = 1$. If the reason for $\hat{\beta}_{b*}$ to be denied is the violated RPC, the bid $\hat{\beta}_{b}$ would have to violate it as well due to $\hat{p}^v_{b} \geq \hat{p}^v_{b*}$, and
\( \hat{c}_{b,g}^{d} \leq \hat{c}_{b,g}^{d}, \forall g \in G \). If \( \hat{b}_{b}^{*} \) was rejected due to the insufficient amount of available resources, i.e. violated ARC, the modified bid \( \hat{b}_{b}^{v} \) must have been moved up in the sorted list in order to satisfy ARC, and to be allocated. Therefore, its sorting criteria value has to be more competitive, which is false \( ^{\hat{b}} \). Therefore, if a bid \( \hat{b}_{b}^{*} \) was denied \( x_{b} = 0 \), then the modified bid \( \hat{b}_{b}^{v} \), s.t. \( \hat{b}_{b}^{v} \geq \hat{b}_{b} \) would also be denied in the market \( x_{b} = 0 \).

Hence, the HO-SS-A scheme is monotone in single-minded buyers' declarations.

\( \square \)

**Theorem 4.** *The proposed pricing scheme determines critical-value payments for single-minded buyers.*

*Proof.* The provided proof is derived based on Definition 3.4.2. We assume that the required critical-value conditions do not hold and provide the proof by contradiction.

**Scenario 1:** We consider a situation when a buyer wins \( x_{b} = 1 \) with the price valuation \( \hat{p}_{b}^{v} < \hat{p}_{b}^{h} \). Therefore, we get the following:

\[
\hat{p}_{b}^{v} < w(\hat{b}_{b}) \times e(\hat{b}^{c}) \quad \Rightarrow \quad \frac{\hat{p}_{b}^{v}}{w(\hat{b}_{b})} < e(\hat{b}^{c}) \quad \Rightarrow \quad e(\hat{b}_{b}) < e(\hat{b}^{c})
\]

Thus, the sorting criteria value of the bid \( \hat{b}_{b} \) would be less competitive compared to its first losing competitor \( \hat{b}^{c} \). It means that the bid \( \hat{b}_{b} \) would be placed at a lower position in the sorted list \( L \) compared to \( \hat{b}^{c} \). In other words, the sorted list \( L \) and \( \hat{L} \) would be identical at least to the position \( z \leq N \) of \( \hat{b}^{c} \) in the sorted list:

\[
\bigcup_{i=1}^{z} L_{i} = \bigcup_{i=1}^{z} \hat{L}_{i} \quad \Rightarrow \quad X_{z} = \hat{X}_{z}
\]

It means that the allocation decision must have been the same for the first \( z \) candidates in both lists \( L \) and \( \hat{L} \), which contradicts the unique competition condition \( (x_{b} = 0 \& x_{b} = 1) \). However, the first losing competitor could have been a bundle reservation bid \( \hat{b}^{br}_{b} \). Given that \( \hat{p}_{b}^{v} < \hat{p}_{b}^{h} \), we get:

\[
\hat{p}_{b}^{v} < w(\hat{b}_{b}) \times e(\hat{b}^{br}_{b}) \quad \Rightarrow \quad \hat{p}_{b}^{v} < w(\hat{b}_{b}) \times \frac{\hat{p}_{b}^{br}}{w(\hat{b}_{b})} \quad \Rightarrow \quad \hat{p}_{b}^{v} < \hat{p}_{b}^{br}
\]

Thus, the bid \( \hat{b}_{b} \) would have violated RPC, and would not win \( x_{b} = 1 \). Therefore, any price valuation \( \hat{p}_{b}^{v} < \hat{p}_{b}^{h} \) would result in the bid \( \hat{b}_{b} \) being denied \( x_{b} = 0 \).

**Scenario 2:** We consider the second case and analyse the situation when a bid is denied \( x_{b} = 0 \) with the price valuation \( \hat{p}_{b}^{v} \geq \hat{p}_{b}^{h} \). We know that the first losing competitor
\( \hat{c} \) satisfies the problem constraints. Similarly, we get that any price valuation \( \hat{\beta}^c \geq p \) would result in the bid being more competitive \( \epsilon(\hat{\beta}^c) \geq \epsilon(\hat{\beta}) \). Therefore, it would appear earlier than her first losing competitor \( \hat{\beta} \) in the sorted list \( L \). As a result, the bid \( \hat{\beta} \) would satisfy the problem constraints and would be allocated. Hence, any price valuation \( \hat{\beta} = p \) would result in the bid \( \hat{\beta} \) being allocated \( x = 1 \).

Hence, the HO-SS-P mechanism determines buyer critical-value payments.

4.1.4 Experimental Investigation

In this section, we conduct the experimental investigation of the proposed mechanism. We aim to analyse the allocative quality of our approximation allocation mechanism, and to study the impact of the seller’s price reservations on the market outcomes.

Experimental Setting

The experimental setup used in our simulation investigation is depicted in Table 4.4. We analyse our proposed mechanism’s behaviour in a wide range of simulated market scenarios. Therefore, we define the model of the market, and experiment with various demand and supply models.

**Market Model:** We consider a single-sided market, which allows trading of infrastructure cloud services of three different types \( K = 3 \), which are Small, Medium and Large VM types\(^2\). We assume that the resource configurations of these three VMs are such that there is a strict relationship (resource multiplier), which we define as the following goods relationship factor: \( F^r = (1, 2, 4) \). A real-world example of such an offering is the general purpose pre-defined VMs for running Windows OS on Rackspace Cloud\(^3\); it includes three VM types where the underlying resources scale based on the aforementioned relationship.

**Demand Model:** In order to model the market demand, i.e. the buyers’ bids, we vary the number of buyers \( N \) participating in the market, and randomly generate the requested bundles of services and corresponding price valuations. The bundles of services requested in the bids are drawn from a uniform distribution function \( U(0, 10) \) and rounded to their natural values. We also use the uniform distribution function in order to generate the price valuations of the buyers; where the random values are scaled according to the linear weight of the requested bundle. Such approach allows to reflect variability for continuous spaces.

\(^2\)Our preliminary experimental results revealed a similar mechanism’s behaviour when the number of traded services change; therefore, we select a representative number of server configurations in a server purpose group.

\(^3\)https://www.rackspace.com/cloud/public-pricing
Table 4.4: HO-SS Experimental Setting

<table>
<thead>
<tr>
<th>Market Model:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Types</td>
<td>$G = {1: \text{Small}, 2: \text{Medium}, 3: \text{Large}}$</td>
</tr>
<tr>
<td>$K = 3$</td>
<td></td>
</tr>
<tr>
<td>Relative Relation</td>
<td>$F^r = {1, 2, 4}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Demand Model:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Buyers</td>
<td>$N = {32, 64, 128, 256, 512}$</td>
</tr>
<tr>
<td>Demanded Capacity</td>
<td>$c_{bg}^d = \mathcal{U}(0, 10), \forall g \in G$</td>
</tr>
<tr>
<td>Price Valuation</td>
<td>$p_b^v = \mathcal{U}(0.0, 0.25) \times \sum_{g \in G} c_{bg}^d f_g^r$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supply Model:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sellers</td>
<td>$M = {1}$</td>
</tr>
<tr>
<td>Supplied Capacity</td>
<td>$c_{sg}^s = \left( \sum_{b \in B} c_{bg}^d \right) \times p_g, \forall g \in G$</td>
</tr>
<tr>
<td>$p_g = {0.25, 0.5, 0.75, 1.0}$</td>
<td></td>
</tr>
<tr>
<td>Price Reservation</td>
<td>$p_{s1}^r = \mathcal{U}(0.0, 0.25) \times z$</td>
</tr>
<tr>
<td>$p_{s2}^r = \mathcal{U}(0.25, 0.5) \times z$</td>
<td></td>
</tr>
<tr>
<td>$p_{s3}^r = \mathcal{U}(0.5, 1.0) \times z$</td>
<td></td>
</tr>
<tr>
<td>$z = {0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5}$</td>
<td></td>
</tr>
</tbody>
</table>

| Metrics and Mechanisms: |  |
| Metrics | Social Welfare, Resource Utilisation, Total Cost, Seller Revenue, Computation Time, Buyer Fulfilment |
| Allocation Schemes | HO-SS where $m = \{0.0, 0.25, \ldots, 3.75, 4.0\}$, Optimal CPLEX solver |
| Pricing Schemes | HO-SS-P |

with huge number of combinations; furthermore, uniform distribution is a common choice for modelling valuations in economic literature [96].

**Supply Model:** In the considered single-sided market there is a single service provider, i.e. $M = 1$. In this experiment, we achieve various provisioning scenarios by balancing the supply against the previously generated demand. The market level of provisioning is determined by the provisioning factor $p_g, \forall g \in G$. The values $p_g < 1.0$ indicate that the service is scarce, while $p_g \geq 1.0$ result in a sufficient service provisioning. In our simulation, we consider all combinations of four provisioning levels\(^4\) $p_g = \{0.25, 0.5, 0.75, 1.0\}$ for the

\(^4\)We omit the zero provisioning case, i.e. $p_g = 0.0$, since it results in a limited number of feasible allocations (due to ARC) and optimal allocation outcomes.
three resource types, i.e. $4^3 = 64$ combinations. The seller’s price reservations are drawn from a uniform distribution function and the results are scaled according to the price comparative ratio $z$. It is the ratio between the average buyer’s generated price valuation and the average seller’s price reservation value. It permits us to experiment with various price reservation scenarios.

Overall, we simulate 448000 market inputs: where 2240 market scenarios are repeated 200 times each. The simulated market inputs are solved by 17 different parametrisations of our proposed greedy mechanism as well as by the optimal solver for comparison. In order to avoid statistical noise, we calculate means of our measured metrics for each market scenarios (across the conducted 200 trials).

**Allocation Mechanism Experiment**

In this experiment, we analyse the allocation quality of our approximation allocation mechanism, including various parametrizations. Therefore, we measure the following metrics: social welfare, resources utilisation, buyer fulfilment and computation time; and compare our allocation mechanism with the optimal results obtained by the IBM CPLEX solver.

**Closeness to Optimality**

In Figure 4.2, we plot the closeness to optimality in terms of the Social Welfare and Average Resource Utilisation achieved by various mechanism parametrisations of our proposed mechanism, i.e. different values for parameter $m$. Please recall that this parameter is used to control the significance of the traded resource size, where smaller values of $m$ give a competitive advantage to the larger requests. We select a representative market case where $N = 512$, $z = 0.0$, $p_1 = p_2 = p_3 = 0.25$ and show the dynamics of the mechanism’s performance by varying one of the three variables.

The approximation quality in terms of social welfare depends (i) on the size of the buyers’ requests compared to the seller’s offer (bid granularity) as well as (ii) on the number of buyer bids feasible for allocation, i.e. the bids that satisfy the problem constraints. When the bids are more granular to the size of the seller’s offer (e.g. $N = 512$), the negative impact of a non-optimal allocation decision will be smaller, while a fewer bids of a bigger relative size ($N = 32$) are more difficult to allocate and the allocation quality is slightly lower (Figure 4.2a). More optimal allocations are commonly achieved when the price comparative ratio $z$ is higher, since more buyer bids fail to satisfy the RPC (Figure 4.2c); similarly, when the resources are less scarce (the resource provisioning $p$ is higher),

\footnote{It is the hardest case for the mechanism, since the resources in the market are scarce, and the seller’s price reservation does not have an impact on the outcomes.}
the attained approximation quality improves due to the lower chance to deny a bid which is a part of an optimal solution (Figure 4.2b).

Overall, we observe a near-optimal performance, when the most efficient mechanism’s parametrisation is used, which is typically $0.75 \leq m \leq 1.0$. In Figure 4.2a, we can see that when the number of buyers is relatively small ($N = 32$), a more efficient allocation is achieved when $m = 0.75$. Such a parametrisation ensures a higher chance of allocating larger competitive buyer bids, while leaving some spare goods to be consumed by smaller
buyer requests. It is also confirmed by the resource utilisation (Figure 4.2d) which is closer to optimal when \( m = 0.75 \). The high values of mechanism parameter \( (m > 1.0) \) is typically a bad choice, which is recommended only if more buyers need to be satisfied in the market. A better buyer fulfilment is achieved because when \( m > 1.0 \), smaller bids are favoured by the mechanism; thus more of them can be allocated.

We can notice that the average resource utilisation optimality follows a similar pattern to the social welfare approximation. It happens because the allocation decisions are constrained by the amounts of available services, which prevent more optimal allocation decisions.

**Resource Provisioning Impact**

In Figure 4.3 we depict the dynamics of the average resource utilisation (Figures 4.3a-4.3d) and the social welfare optimality (Figures 4.3e-4.3h) based on the varying level of resource provisioning for the services of different types. In the presented graphs, we fix the provisioning level of the small VM type \( p_1 \) and display the obtained results when the provisioning of the other two services change.

![Figure 4.3](image-url)

Figure 4.3: Impact of Resource Provisioning on Average Resource Utilisation (a, b, c, d) and Approximation Ratio (e, f, g, h). Market Scenario: mechanism with \( m = 1.0, N = 512, z = 0.0 \).

We can clearly see that the best average resource utilisation is achieved when all three traded types of services are provisioned in an equal proportion to the market demand,

---

6We omit displaying all the dimensions, since the obtained results are almost identical
i.e. when $p_1 = p_2 = p_3$. It means that the utilisation of services of different types is highly interrelated. Typically, the scarcer service type gets exhausted by the mechanism, while some amount of services of the other types remain unallocated. Due to the resource complementarity in the buyer requests, the chance of making further allocations, after that the scarcest resource has been sold out, diminish significantly, which leads to a lower average resource utilisation. The social welfare has similar dynamics, where the most optimal solution is achieved when the resources are provisioned in an equal way. We can see that the lowest attained approximation ratio of 78% (Figure 4.3e) is an attribute to a rough estimate of resource provisioning $p_1 = 0.25, p_2 = 1.0, p_3 = 1.0$.

In Figure 4.4 we show the impact of the mechanism’s parameter on the social welfare and average resource utilisation optimality in three different provisioning scenarios, where one of the service types is scarce in regards to the others. We can notice that when the Large VM is scarce (i.e. $p_1 = 1.0, p_2 = 1.0, p_3 = 0.25$), the higher values of parameter $m$ allow achieving a better allocation quality (Figure 4.4a) and better average resource utilisation (Figure 4.4b). It happens because the smaller buyer requests, generated in our experiments, statistically contain less large VMs. The mechanism with parameter $m \geq 1.0$, will favour smaller requested bundles, and will not exhaust the largest resource to quickly. Such behaviour results in a higher social welfare and better resource utilisation. The opposite effect can be observed in the other two cases, when the small and medium VMs are scarce in the market.

\textit{Computation Time}

The computation time of the proposed HO-SS allocation mechanism and the Optimal
Figure 4.5: Computation Time of Market Allocation. Representative Market Scenario: $m = 1.0, z = 0.25, p = 0.5$.

solver are presented in Figure 4.5. It can be clearly seen that the approximation mechanism solves in near-real time: the market with the biggest number of buyers was cleared in less than 1ms on average. The optimal mechanism also showed time-efficient performance; however the NP-hard time complexity of the optimal solution is not feasible for large-scale computational markets.

**Pricing Mechanism Experiment**

In this experiment, we study the impact of changing seller’s price reservations on the market outcomes determined by the mechanism. We measure the performance (i) metrics related to allocation, such as social welfare, resource utilisation, buyer fulfilment, as well as (ii) the price-related metrics, such as seller’s revenue, total services reserve price, and the total buyer discount. We compare the results obtained by various parametrisations of our proposed mechanism and draw conclusions about the interrelations between the mechanism’s allocative efficiency and the determined market prices.

**Seller Power over Market Pricing Outcomes**

The social welfare can be defined as a sum of all allocated buyer price valuations for the granted resources. During the pricing phase the social welfare is split into the total payment from all the buyers to the seller and the total discount granted to the buyers. The total seller payment consists of the reservation price as specified in the ask and the generated profit. We study how the market generated social welfare is distributed among the buyers and the seller during the pricing phase, and what is the seller’s power over the pricing outcomes that is achieved through the reserve prices.

In Figure 4.6, we depict how the social welfare is split in the market. The total bar
represents the social welfare, and its portions of buyers’ discount, seller’s profit and total reserve price are indicated by different colours. We can see that the total social welfare is higher when more services are provisioned in the market (Figure 4.6d) since more buyer requests can be fulfilled. Furthermore, the increasing reserve price \((z \geq 0.75)\) diminishes the number of allocated buyers and reduces the social welfare. When the services are very scarce (Figures 4.6a, 4.6b), the seller’s payment is derived based on the competition in the market and the increased reserve price does not allow to improve the seller’s payment. Furthermore, when the services are overpriced \((z = 1.5)\) the seller will experience some serious loss. It is possible to achieve a slight improvement in the seller’s payment due to increased reserve price, when the resources are slightly insufficient to address the market demand (Figure 4.6c). We can see that due to the price reservation, the seller achieves a better payment when \(z = 0.75\), compared to the payment derived purely based on the competition among the buyers \(z = 0.0\). The reserve price is essential for the seller in the situation when the services are overprovisioned (supply \(\geq\) demand)(Figure 4.6c). In such scenario, there is no competition among the buyers, which drives the final prices down to minimum. If the reserve price is zero \((z = 0.0)\), the seller will not obtain any payment.

**Social Welfare Optimality and Market Pricing**

In Figure 4.7, we depict the impact of mechanism’s parametrisation on the market pricing, average resource utilisation and buyer fulfilment. We can observe a very interesting behaviour: the more efficient allocation outcomes result in a higher seller payment (Figure 4.7a). While the social welfare improvement is around 9% between parametrisations \(m = 0.0\) and \(m = 1.0\), there is a significant difference in social welfare distribution. Apart from increased total reserve price for the sold services (due to better resource utilisation, Figure 4.7b), the seller obtains a bigger profit (increased from $477 \((m = 0.0)\) to $631 \((m = 1.0)\)) by the cost of the buyers’ total discount (Figure 4.7a) (reduced from $212 \((m = 0.0)\) to $140 \((m = 1.0)\)). Such behaviour is specific to the way the buyer pricing mechanism works. In simple terms, any mechanism parametrisation rather than one is more likely to result in lower final prices for the buyers, i.e. better offered discount. Let us consider two scenarios:

- **Parametrisation** \(m < 1\): the mechanism favours larger buyer requests; hence, it is likely that the weight of the bundle requested by the winning buyer is greater than the weight of her first losing competitor \(w(\hat{\beta}^w) > w(\hat{\beta}^c)\). As a result, given the parameter \(m < 1\), we can conclude that the determined buyer price will be lower
Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

Figure 4.6: Seller Power over Market Pricing Outcomes: Social Welfare Distribution. Market Scenario: $m = 1.0, N = 512$.

Figure 4.7: Impact of Mechanism Parametrisation on Market Pricing (a), Average Resource Utilisation (b), and Buyer Fulfilment (c). Market Scenario: $N = 512, z = 0.25, p = 0.25$. 
compared to the corresponding linear density price:

\[
\begin{align*}
w(\hat{\beta}^w) &> w(\hat{\beta}^c) \\
m < 1
\end{align*}
\]

\[
\Rightarrow p_c^v \left( \frac{w(\hat{\beta}^w)}{w(\hat{\beta}^c)} \right)^{1.0} > p_c^v \left( \frac{w(\hat{\beta}^w)}{w(\hat{\beta}^c)} \right)^{m<1}
\]

- **Parametrisation** \( m > 1 \): the smaller buyer requests are preferred; thus, there is a higher chance for the first losing competitor bundle weight to be larger than the one of the winning bid. Therefore, \( m > 1 \) will produce a lower buyer price:

\[
\begin{align*}
w(\hat{\beta}^w) &< w(\hat{\beta}^c) \\
m > 1
\end{align*}
\]

\[
\Rightarrow p_c^v \left( \frac{w(\hat{\beta}^w)}{w(\hat{\beta}^c)} \right)^{1.0} > p_c^v \left( \frac{w(\hat{\beta}^w)}{w(\hat{\beta}^c)} \right)^{m>1}
\]

**Computation Time**

The computation time of our proposed market mechanisms is depicted in Figure 4.8. As we can see, that the market clearance happens in milliseconds, even for the hard market cases. Such time-efficiency is achieved thanks to the sorted list of buyers which is being reused by the pricing mechanism instead of recalculating the orders for each winning buyer.

![Computation Time Chart](image.png)

Figure 4.8: Computation Time of HO-SS Market Mechanism. Representative Market Scenario: \( m = 1.0, z = 0.25, p = 0.5 \).
4.2 Double-sided Market Mechanism

In this section, we design a computational market mechanism for homogeneous infrastructure cloud services trading in the market with multiple sellers and multiple buyers. The resources are traded as combinatorial bundles of services of different types, such as the virtual machine configurations of different pre-defined sizes, e.g. small, medium, large.

4.2.1 Allocation Mechanism

The proposed allocation scheme is given in Algorithm 3. The allocation scheme aims to determine the final allocation decision matrix \( X \), which specifies the couples of buyers and sellers who will exchange the services, given a vector of declared buyer bids \( \hat{\alpha} \), and a vector of declared seller asks \( \hat{\beta} \). The mechanism also returns the sorted list of the allocation candidates \( L \), which is used by the pricing scheme in its operation in order to achieve a better time complexity.

Algorithm Description

The allocation mechanism needs to determine the couples of sellers and buyers who will exchange the traded services. It is a difficult problem given a feasible time complexity requirement. Due to the combinatorial nature of the buyer’s request, the traditional greedy approach with two sorted queues (i.e. the queues for the participating buyer bids, and seller’s asks) is not applicable, since there is no single sorted order for the seller’s asks. We propose a new greedy allocation approach suitable for the problems with a similar setting (i.e. combinatorial bids and seller pricing for multiple good types). The idea is to view each couple of a bid and an ask as an allocation candidate \( \varsigma_{sb} \), which are ranked and considered for allocation. Therefore, our proposed allocation scheme runs the following steps:

- **Initialization Phase (lines 3-4):** Firstly, the allocation algorithm initialises the variables required for its operation, such as the final allocation decision \( X \), and the list of all allocation candidates \( L \) among the participating bids and asks. An *allocation candidate* is a binding of the seller’s ask, and the buyer’s bid in a single object, defined as follows:

\[
\varsigma_{sb} = \langle \hat{\alpha}_s, \hat{\beta}_b \rangle = \langle \hat{C}^d_b, \hat{p}^v_b, \hat{p}^{br}_{sb} \rangle
\]

Hence, an allocation candidate is composed of a bundle of services to be exchanged \( \hat{C}^d_b \), the buyer’s price valuation \( \hat{p}^v_b \), and the bundle reservation price \( \hat{p}^{br}_{sb} \) of the associated seller.
Algorithm 3 Double-sided Allocation of Homogeneous Cloud Infrastructure Services (HO-DS-A)

1: Input: $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$, where $\hat{\beta}_b = (\hat{C}^d_b, \hat{p}^v_b)$
2: Input: $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_M)$, where $\hat{\alpha} = (\hat{C}^s, \hat{P}^r)$

# Step 1: Initialization Phase
3: $X \leftarrow \emptyset$ // final allocation decision matrix
4: $L \leftarrow \{\varsigma_{sb} \mid \forall s \in S, \forall b \in B\}$ // all possible allocation candidates

# Step 2: Sorting Phase
5: Re-order candidates $\varsigma_{sb}$ in list $L$ in decreasing order of $e(\varsigma_{sb})$

# Step 3: Allocation Phase
6: for all $\varsigma_{sb} \in L$ do
7: if $\text{ARC}(\varsigma_{sb}, X) \& \text{RPC}(\varsigma_{sb}) \& \text{BAC}(\hat{\beta}_b, X)$ then
8: $x_{sb} = 1$
9: else
10: $x_{sb} = 0$
11: end if
12: $X \leftarrow X \cup x_{sb}$
end for
13: Output: $X, L$

- **Sorting Phase (line 5):** Secondly, the allocation scheme sorts the allocation candidates in the list $L$ in decreasing order of their sorting criteria values, determined by the sorting criteria function $e(\varsigma_{sb})$. The goal is to determine such sorting criteria values that result in the social-welfare maximising sorting order. A detailed explanation about the sorting criteria function and its relationship to the problem’s objective is provided later in this section.

- **Allocation Phase (lines 6-13):** Finally, the mechanism makes the allocation decisions for the candidates given the sorted order of in the list $L$. A candidate is selected as a market winner and the associated buyer and seller will exchange the requested services, if all the problem constraints are satisfied; otherwise a candidate is rejected by the mechanism. The following constraints are verified:

  - **Available Resource Constraint (ARC):** is verified in order to ensure the requested bundle of services can be provided by the corresponding seller. Formally, the constraint is defined as follows:

    $$\hat{c}_{bg}^d \leq \hat{c}_{sg}^a, \forall g \in G, \text{ where } \hat{c}_{sg}^a = \hat{c}_{sg}^s - \sum_{b \in B} \hat{c}_{bg}^d x_{sb}$$

  - **Reserve Price Constraint (RPC):** is used to ensure that the buyer offers the price valuation $\hat{p}^v_b$ for the requested services, which satisfies the seller’s desired
minimum price $p_{sb}^{br}$. Formally, the constraint is defined as follows:

$$p_{b}^{v} \geq p_{sb}^{br}, \text{ where } p_{sb}^{br} = \sum_{g \in G} c_{bg}^{d} \pi_{ur}^{br}$$

- Binary Allocation Constraint (BAC): is verified in order to ensure that each buyer’s bid is allocated at most once. In other words, a bid cannot be granted the requested bundle of services several times and from multiple different sellers. Formally, the constraint is defined as follows:

$$\sum_{s \in S} x_{sb} \leq 1, \quad \forall b \in B$$

Sorting Criteria Function

The sorting criteria function $e(\hat{\beta}_b)$ is used to determine the sorting criteria values associated with each allocation candidate; these values serve as a ranking criteria in the sorted list $L$. So as to determine the efficient allocation results, the mechanism has to determine such order of the allocation candidates that allows to maximise the social welfare. Therefore, the more socially efficient candidates have to be preferred and placed at the higher positions in the sorted list $L$. Consequently, an efficient sorting criteria function has to be tightly linked with the problem’s objective function.

Since a double-sided market mechanism aims to maximise the total market generated value, measured as a potential surplus, produced by the allocation. A more socially efficient allocation candidate would be the one that generates a greater surplus, while exchanging less resources. Therefore, our proposed sorting criteria function is defined as follows:

$$e(\hat{\beta}_b, \hat{\alpha}_s) = e(\varsigma_{sb}) = \frac{\hat{p}_b^{v} - \hat{p}_{sb}^{br}}{w(\hat{\beta}_b)}, \text{ where } w(\hat{\beta}_b) = \left( \sum_{r \in R} \tilde{c}_{br}^{d} F_r^r \right)^m$$

This sorting criteria function is based on the declarations of the bid $\hat{\beta}_b$ and the ask $\hat{\alpha}_s$ from the allocation candidate $\varsigma_{sb}$, including bid’s price valuation for the requested bundle $\hat{p}_b^{v}$, ask’s bundle reservation price for the same set of services $\hat{p}_{sb}^{br}$, and the bundle of services to be exchanges between the buyer and the seller $\hat{C}_{b}^{d}$. The principal idea is to determine the amount of surplus, that the allocation candidate could contribute per unit of exchanged goods, if allocated. Therefore, the function calculates the weight of the exchanged bundle $w(\hat{\beta}_b)$ based on goods relationship factor $F_r^r = \langle f_1^r, \ldots, f_K^r \rangle$, where $f_g^r$ - is a relationship factor for the good type $g \in G$. Such sorting criteria function considers a candidate to be
more efficient if it offers a greater price valuation for fewer requested goods.

Similar to the single-sided setting, the parameter \( m \geq 0 \) is used to allow controlling the significance of the exchanged bundle size, where \( m > 1 \) indicates to the mechanism that smaller request should be favoured, while \( m < 1 \) would give advantage to the larger exchanged bundles.

**Allocation Mechanism Example**

We demonstrate the procedure of our proposed double-sided mechanism for trading homogeneous resources by providing a simple example. We suppose that the market allows trading of general purpose VMs of three different types: small, medium, and large. The VM types and the corresponding resource configurations are provided in Table 4.1a. Based on the underlying resources that compose the VMs of different types, the following strict relationship emerges: Large = 2 × Medium = 4 × Small, which allows to establish the following mechanism’s relationship factor \( F_r = (1, 2, 4) \).

In the considered example, there are two sellers who join the market in order to exchange the services they possess. The corresponding offers/asks are depicted in Table 4.5a. The Small Seller (SS) is willing to trade multiple instances of small and medium VM types, but has only a single large VM to offer. The Large Seller (LS) offers more instances of Large VMs. They report their minimum desired prices for the VM instances of each type, where SS sets a cheaper price for the small VMs, and LS discounts the large VMs. There are six potential consumers of the cloud resources who submit their bids to the market. The considered buyers’ bids, including the requested bundles and the reported price valuations are depicted in Table 4.5b.

The allocation procedure of our proposed double-sided mechanism for trading homogeneous cloud services in the considered example is depicted in Table 4.6. The mechanism collects bids and asks, and calculates the sorting criteria values for all possible allocation candidates. Since there are two seller offers and six buyer requests, the market mechanism will examine twelve allocation candidates. For simplicity, we consider a linear form of the sorting criteria function, when \( m = 1.0 \). The weights of the exchanged bundles of services are determined given the goods relationship factor \( F_r = (1, 2, 4) \). For instance, the sorting criteria value of the allocation candidate among the seller LS and the buyer LC1 is calculated as follows:

\[
e(\text{LS, LC1}) = \frac{\$9.00 - \$1.86}{(0 \times 1 + 0 \times 2 + 6 \times 4)^{1.0}} = 0.2975
\]
Table 4.5: HO-DS Mechanism Example: Market Input

(a) Sellers’ Offers / Asks

<table>
<thead>
<tr>
<th>Seller, s</th>
<th>Offered Services, $\hat{C}_s^o$</th>
<th>Price Reservations, $\hat{P}_s^r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>Small: 5</td>
<td>Medium: 6</td>
</tr>
<tr>
<td>LS</td>
<td>Small: 1</td>
<td>Medium: 5</td>
</tr>
</tbody>
</table>

(b) Buyers’ Requests / Bids

<table>
<thead>
<tr>
<th>Buyer, b</th>
<th>Requested Bundle, $\hat{C}_b^o$</th>
<th>Val., $\hat{P}_b^v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC1</td>
<td>Small: 0</td>
<td>Medium: 0</td>
</tr>
<tr>
<td>LC2</td>
<td>Small: 0</td>
<td>Medium: 1</td>
</tr>
<tr>
<td>MC1</td>
<td>Small: 4</td>
<td>Medium: 6</td>
</tr>
<tr>
<td>MC2</td>
<td>Small: 0</td>
<td>Medium: 0</td>
</tr>
<tr>
<td>PhD1</td>
<td>Small: 1</td>
<td>Medium: 0</td>
</tr>
<tr>
<td>PhD2</td>
<td>Small: 4</td>
<td>Medium: 0</td>
</tr>
</tbody>
</table>

We would like to draw your attention to the following two bids: MC1, and MC2. They request different bundles of services, but they both have the same linear bundle weight and price valuation. However, their associated sorting criteria values will be different due to the difference in unit price reservation from the sellers. For example, the candidate (SS, MC1) will be more competitive than (SS, MC2), since SS offers a more expensive price for large VM type which is not requested by MC1, unlike the buyer MC2 who demands 4 instances of large VMs.

The mechanism re-orders the allocation candidates in the list $L$ in decreasing order of their sorting criteria values. In our example (LS, LC1) is the most socially efficient candidate, who is ranked at the first position in the list $L$. In order to make the allocation decision for the considered candidate, the mechanism verifies the problem constraints. A positive allocation decision is made for (LS, LC1), since (i) LC1 has not been previously allocated (BAC), (ii) the seller LS has a sufficient amount of resource available to satisfy LC1 (ARC), and (iii) the price valuation of LC1 is good enough to address the minimum price, desired by LS (RPC). The process repeats for all twelve allocation candidates in the sorted list $L$.

As a result, the allocation mechanism determines three winning allocation candidates.
Table 4.6: HO-DS Mechanism Example: Allocation Procedure ($m = 1.0$)

<table>
<thead>
<tr>
<th>Sorted List, $L$</th>
<th>Values</th>
<th>Constraints</th>
<th>$x_{ab}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>$b$</td>
<td>$w(\hat{\beta}_b)$</td>
<td>$e(\hat{\beta}_b)$</td>
</tr>
<tr>
<td>LS</td>
<td>LC1</td>
<td>24.0</td>
<td>0.2975</td>
</tr>
<tr>
<td>SS</td>
<td>LC1</td>
<td>24.0</td>
<td>0.2925</td>
</tr>
<tr>
<td>SS</td>
<td>MC1</td>
<td>16.0</td>
<td>0.1875</td>
</tr>
<tr>
<td>LS</td>
<td>MC2</td>
<td>16.0</td>
<td>0.1850</td>
</tr>
<tr>
<td>SS</td>
<td>MC2</td>
<td>16.0</td>
<td>0.1800</td>
</tr>
<tr>
<td>LS</td>
<td>MC1</td>
<td>16.0</td>
<td>0.1775</td>
</tr>
<tr>
<td>LS</td>
<td>LC2</td>
<td>22.0</td>
<td>0.1723</td>
</tr>
<tr>
<td>SS</td>
<td>LC2</td>
<td>22.0</td>
<td>0.1677</td>
</tr>
<tr>
<td>SS</td>
<td>PhD1</td>
<td>1.0</td>
<td>0.1600</td>
</tr>
<tr>
<td>LS</td>
<td>PhD1</td>
<td>1.0</td>
<td>0.1200</td>
</tr>
<tr>
<td>SS</td>
<td>PhD2</td>
<td>4.0</td>
<td>0.0275</td>
</tr>
<tr>
<td>LS</td>
<td>PhD2</td>
<td>4.0</td>
<td>-0.0125</td>
</tr>
</tbody>
</table>
(i.e. \(\langle LS, LC1 \rangle, \langle SS, MC1 \rangle, \) and \(\langle SS, PhD1 \rangle\)), which will exchange the agreed bundles of services. The remaining candidates are rejected by the mechanism either due to insufficient amounts of available goods, bid being previously allocated (i.e. \(\langle SS, LC1 \rangle, \langle LS, MC1 \rangle, \) and \(\langle LS, PhD1 \rangle\)), or low price valuation (i.e. \(\langle LS, PhD2 \rangle\)).

The mechanism allocated the following amounts of goods \((5, 6, 0)\) for the seller SS, and \((0, 0, 6)\) for the seller LS. Hence, the resource utilization was as follows: \((100\%, 100\%, 0\%)\) for SS, and \((0\%, 0\%, 100\%)\) for LS. The achieved social welfare is \(W = (9.00 - 1.86) + (4.20 - 1.20) + (0.22 - 0.06) = 10.30\). In the considered example, the approximation mechanism achieves an optimal solution, which is not always guaranteed. We investigate the allocative performance of the proposed mechanism is experimental setting.

### 4.2.2 Pricing Mechanism

In double-sided markets, the pricing problem becomes more challenging since the payments have to be determined for both the buyers and the sellers. Given combinatorial buyer bids and seller unit price reservations for services of different types, it is a non-trivial problem, especially if we aim to achieve some economic properties of the mechanism design, such as budget-balance, individual rationality, and incentive compatibility. Therefore, we split the problem into two parts and propose the pricing approach where the buyers’ prices are determined first, and the seller payments are derived afterwards.

#### Buyer Pricing Algorithm Description

The proposed buyer pricing mechanism is depicted in Algorithm 4. The pricing scheme intends to compute the final buyer prices \(P^b\), given the reported bids of the participating buyers \(\hat{\beta}\), the submitted offers of the participating sellers \(\hat{\alpha}\), and the previously determined allocation decision variable \(X\) and the sorted list of allocation candidates \(L\).

Our proposed pricing scheme for the buyers calculates the prices for all the participating bids in order. The losing buyer bids are not required to pay and receive zero payment (lines 7, 24). The allocated buyers have to pay the price which depends on the competition in the market (lines 9-22). The critical-value payment method, which derives prices based on the first associated losing competitors, is not applicable to the considered double-sided problem in its standard form. The main reason for this is the combinatorial buyer requests in the presence of multiple sellers with individual minimum prices for multiple types of goods. Therefore, we design a novel double-sided critical-value pricing mechanism, which
derives the minimum buyer payments to remain the market winners, by analysing all the market competitors, regardless of the seller who provisions the goods. For each winning buyer, the proposed buyer pricing mechanism performs the following operations:

- **Initialization Phase (lines 9-12):** Firstly, the mechanism initialises the auxiliary allocation decision variable $\tilde{X}$, which is used to keep track of ongoing auxiliary allocations; the variable $\tilde{L}$ which contains a list of competitors for the winning buyer; and constructs a new sorted list of candidates $\tilde{L}$. This list contains all the allocation candidates in the initially sorted order in $L$, except for the candidates with the currently considered winning bid. The candidates with the currently considered winning bid are replaced by the bundle-reservation candidates and added at the end of the list $\tilde{L}$ in the order in which they appear in the list $L$.

A bundle reservation candidate $\varsigma_{br}^b = (\hat{c}_b^d, p_{br}^b, ^p_{br}^b)$ is a special type of candidate, which aims to exchange the same bundle of services as the initial candidate $\varsigma_b$, but the buyer’s initial price valuation $\hat{p}_b$ is replaced by the seller’s bundle-specific price reservation $^p_{br}^b = \sum_{g \in G} \hat{p}_{sg}^d c_{bg}$. Such a price valuation is just large enough to satisfy the seller’s desired minimum price for the bundle, and the sorting criteria value of a bundle-reservation candidate is always $e(\varsigma_{br}^b) = 0$, since the produced surplus is 0.

The bundle-reservation candidate is needed in order to determine winning buyer’s price in the situation of low competition, e.g. no other competitors, market resource overprovisioning, etc.

- **Competitors Determination Phase (lines 13-21):** Secondly, the mechanism searches for the competitors among the allocation candidates present in the new sorted list $\tilde{L}$. Therefore, the pricing mechanism runs a new allocation phase with a new sorted list of candidates $\tilde{L}$. The goal is to determine all the competitors of the currently considered winning bid by verifying the competitors condition.

  - **Allocation feasibility condition (lines 14-15):** The candidate $\varsigma_{gb}$ is considered to be a competitor of $\varsigma_b$, if both could be allocated, i.e. satisfy all the required constraints, when considered at a particular position in the sorted list. Please, note that the auxiliary allocation decision variable $\tilde{X}$ is updated according to the ongoing new allocation of candidates in the list $\tilde{L}$ (lines 18, 19) in order to keep track of the consumed services.

---

7 We say that the buyer is the winner, if her requested bundle of services was allocated to any seller.

8 Such order ensures that the candidate with lower seller’s price reservation appears earlier in the list $\tilde{L}$. 

Algorithm 4 Double-sided Buyer Pricing of Homogeneous Cloud Infrastructure Services (HO-DS-PB)

1: **Input:** $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$
2: **Input:** $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_M)$
3: **Input:** $X = x_{sb}, \forall s \in S, \forall b \in B$
4: **Input:** $L = \{s_{sb}\}, \forall s \in S, \forall b \in B$ // sorted list of candidates
5: $p^b \leftarrow 0$
6: **for** all $\hat{\beta}_b \in \hat{\beta}$
7: $\rho_b^b = 0$
8: **if** $(!BAC(\hat{\beta}_b, X))$ **then**
9:   **# Step 1: Initialization Phase**
10:   $L^c \leftarrow \emptyset$ // list of competitors
11:   $X \leftarrow \emptyset$ // auxiliary allocation results
12:   $\bar{L} \leftarrow \{s_{\sigma\sigma} \mid \sigma \neq b \land \forall s_{\sigma\sigma} \in L\}$ // all candidates, except for $b$
13:   $\bar{L} \leftarrow \bar{L} \cup \{s_{\sigma\sigma} \mid \sigma = b \land \forall s_{\sigma\sigma} \in L\}$ // bundle-reservation for $b$
14: **end if**
15: **for** all $s_{sb} \in \bar{L}$
16:   **if** $\left(ARC(s_{sb}, \bar{X}) \land RPC(s_{sb}) \land BAC(\hat{\beta}_b, \bar{X})\right)$ **then**
17:     **if** $\left(ARC(s_{sb}, \bar{X}) \land RPC(s_{sb})\right)$ **then**
18:       $L^c \leftarrow L^c \cup s_{sb}$
19:     **end if**
20:   **end if**
21: **end for**
22: **# Step 2: Competitors Determination Phase**
23: **for** all $s_{sb} \in \bar{L}$
24:   $\rho_b^b = \min(\{p_{sb}^{br} + e(s_{sb}) \times w(\hat{\beta}_b)\}), \forall s, b : s_{sb} \in L^c$
25: **end if**
26: **# Step 3: Price Determination Phase**
27: **Output:** $P^b = \{\rho_1^b, \ldots, \rho_N^b\}$

Please, note that unlike a single-sided mechanism, the unique competitor condition is not required since the mechanism considers all the allocation candidates. Once, all the allocation candidates in the list $\bar{L}$ are considered and a subset of all possible competitor candidates is determined, the mechanism proceeds to the next phase.

- Price Determination Phase (line 22): Finally, the mechanism determines the minimum price required to outbid at least one competitor from the list $L^c$. The minimum price needed for the candidate $varsigma_{sb}$ to outbid her competitor $s_{sb}$ is determined by adding her required bundle-reservation price $p_{sb}^{br}$ to the surplus needed to produce better sorting criteria value. Such a surplus is calculated by scaling the exchanged bundle weight $w(\hat{\beta}_b)$ according to the competitors sorting criteria value $e(s_{sb})$. 


Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

Buyer Pricing Mechanism Example

In order to describe the pricing procedure of the proposed mechanism, we continue the initially considered example used for allocation scheme. Once, the market allocation decisions are made, the pricing mechanism aims to establish the prices that the buyers have to pay. The buyer pricing procedure of our proposed double-sided mechanism for trading homogeneous cloud services is depicted in Table 4.7.

The final buyer prices are determined for all the participating buyers in order. The rejected buyers, such as LC2, MC2, and PhD2, are not required to pay anything, since they were not granted their requested bundles of services. The final prices for the allocated buyers, such as LC1, MC1, and PhD1, are calculated based on the competition in the market. Therefore, the mechanism constructs a new sorted list $\tilde{L}$ for each of them, and searches for the competitors. The lowest price required to outbid one of the determined competitors becomes the final buyer’s price.

For example, for the winning buyer LC1, the new sorted list of allocation competitors $\tilde{L}$ would contain all the allocation candidates from the list $L$ in the previously determined order, except for the ones that contain LC1. These candidates are replaced by the bundle reservation candidates and added at the end of the list. The allocation candidates in the list $\tilde{L}$ are examined by the mechanism in the specified order. The allocation feasibility condition is verified and the corresponding competitors of the buyer LC1 are determined. For example, $(SS, MC1)$, sorted at the first position in the list $\tilde{L}$ and the corresponding candidate with the current winner $(SS, LC1)$ are not competitors, since the latter violates ARC due to insufficient amount of large VMs. The process repeats for all the allocation candidates in the list $\tilde{L}$. As a result, there is only one competitor $(LS, MC2)$ for the buyer LC1, and the smallest price required to outbid its declaration is calculated as follows:

$$p^{br}((LS, LC1)) + e((LS, MC2)) \times w((LS, LC1)) = 1.86 + 0.1850 \times 24 = 6.30$$

The prices for the other two winning buyers, i.e. MC1, and PhD1 are determined in a similar fashion. We can see that there were two competitors for the bid MC1, where the smallest price required to outbid at least one of them $1.64 was determined based on the competitor $(SS, PhD2)$. Three competitors were determined for the buyer PhD1, where the lowest price was the seller’s required price reservation $0.06, which was derived based on the corresponding bundle-reservation candidate $(SS, PhD1BR)$.

---

9Please note that the bundle reservation candidates also maintain the original order of the corresponding candidates in the list $L$, which permits the candidate with the lower price reservation to appear first.
Table 4.7: HO-DS Mechanism Example: Buyer Pricing Procedure \((m = 1.0)\)

<table>
<thead>
<tr>
<th>Buyer, (b)</th>
<th>(\sum_{s \in S} x_{sb})</th>
<th>Competitors Determination</th>
<th>New List of Candidates, (\bar{L})</th>
<th>Conditions</th>
<th>Final Price, (\bar{p}^c_{sb})</th>
<th>(\rho^b)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(c_{sb})</td>
<td>(e(c_{sb}))</td>
<td>(\bar{x}_{sb})</td>
<td>(\bar{L}^c)</td>
<td></td>
</tr>
<tr>
<td>LC1</td>
<td>1</td>
<td>(\langle \text{SS, MC1} \rangle)</td>
<td>0.1875</td>
<td>1</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC2} \rangle)</td>
<td>0.1850</td>
<td>1</td>
<td>✓</td>
<td>$6.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC2} \rangle)</td>
<td>0.1800</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC1} \rangle)</td>
<td>0.1775</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC2} \rangle)</td>
<td>0.1723</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC2} \rangle)</td>
<td>0.1677</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC1BR} \rangle)</td>
<td>0.1600</td>
<td>1</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, PhD1} \rangle)</td>
<td>0.1200</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD1} \rangle)</td>
<td>0.0275</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD2} \rangle)</td>
<td>-0.0125</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC1BR} \rangle)</td>
<td>0.0000</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC1BR} \rangle)</td>
<td>0.0000</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC1} \rangle)</td>
<td>0.2975</td>
<td>1</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC1} \rangle)</td>
<td>0.2925</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC2} \rangle)</td>
<td>0.1850</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC2} \rangle)</td>
<td>0.1800</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC2} \rangle)</td>
<td>0.1723</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC2} \rangle)</td>
<td>0.1677</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, PhD1} \rangle)</td>
<td>0.1600</td>
<td>1</td>
<td>✓</td>
<td>$3.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD1} \rangle)</td>
<td>0.1200</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, PhD2} \rangle)</td>
<td>0.0275</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD2} \rangle)</td>
<td>-0.0125</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC1BR} \rangle)</td>
<td>0.0000</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC1BR} \rangle)</td>
<td>0.0000</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC1} \rangle)</td>
<td>0.2975</td>
<td>1</td>
<td>✓</td>
<td>$0.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC1} \rangle)</td>
<td>0.2925</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC1} \rangle)</td>
<td>0.1875</td>
<td>1</td>
<td>✓</td>
<td>$0.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC1} \rangle)</td>
<td>0.1850</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, MC2} \rangle)</td>
<td>0.1800</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, MC1} \rangle)</td>
<td>0.1775</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, LC2} \rangle)</td>
<td>0.1723</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, LC2} \rangle)</td>
<td>0.1677</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, PhD2} \rangle)</td>
<td>0.0275</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD2} \rangle)</td>
<td>-0.0125</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{SS, PhD1BR} \rangle)</td>
<td>0.0000</td>
<td>1</td>
<td>✓</td>
<td>$0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\langle \text{LS, PhD1BR} \rangle)</td>
<td>0.0000</td>
<td>0</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>PhD2</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$0.00</td>
</tr>
</tbody>
</table>
Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

Seller Pricing Mechanism Description

The proposed seller pricing mechanism is depicted in Algorithm 5. The seller pricing scheme intends to compute the final seller payments $P^s$, given the initial market input, including participating buyer bids $\hat{\beta}$ and seller asks $\hat{\alpha}$; and the previously determined allocation decision variable $X$ together with the final buyer prices $P^b$.

There is no seller pricing mechanism complementary to the greedy heuristic in a combinatorial setting, which could be used to generate truthful payments for the sellers [131]. Therefore, our goal is to design such a pricing scheme which would limit the seller’s strategic manipulation as much as possible. In classic auction design, the truthful pricing is achieved when the final price does not depend on the trader’s declaration directly [113, 94]. We follow these principles and propose a pricing mechanism that grants a portion of market generated surplus (in addition to the seller’s required minimum price for the allocated services) based on different mixed rules. In particular, our pricing mechanism relies on the proportional-value pricing and direct payment rules:

- **Direct Payment**: is the most simple way for seller payment determination, where the seller receives the portion of surplus that was generated by her allocated buyers $\Delta_s$. In other words, the seller’s direct payment is the sum of prices that the buyer’s pay for the resources granted from this seller. Formally, it is expressed as follows:

$$\rho^s = \sum_{b \in B} \hat{p}_{sb}^br_{sb}x_{sb} + \Delta_s = \sum_{b \in B} \hat{p}_{sb}^br_{sb}x_{sb} + \left( \sum_{b \in B} (\rho_b^b - \hat{p}_{sb}^br_{sb})x_{sb} \right) = \sum_{b \in B} \rho_b^b x_{sb}$$

- **Proportional-Value Payment**: approach was previously proposed for the markets with a single central-scarce resource [131]. Our proposed proportional-value payment operates in combinatorial markets with multiple resource types. It applies the principle of distributing the market-generated surplus among the sellers based on their contribution of goods after allocation. In other words, the seller who sold more resource in the market, as a proportion to the total exchanged amount, will obtain a greater portion of generated surplus. Formally, such
Algorithm 5 Double-sided Seller Pricing of Homogeneous Cloud Infrastructure Services (HO-DS-PS)

1. **Input:** $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$
2. **Input:** $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_M)$
3. **Input:** $X = x_{sb}, \forall s \in S, \forall b \in B$
4. **Input:** $T^b = \{\rho^b_1, \ldots, \rho^b_N\}$

# Step 1: Market Volume and Surplus

5. $P^s \leftarrow 0$
6. $\Delta = \sum_{b \in B} \rho^b_b - \sum_{s \in S} \sum_{b \in B} \hat{p}^b_{sb} x_{sb}$ // generated surplus
7. $\Omega = \sum_{s \in S} \sum_{b \in B} w(\hat{\beta}_b) x_{sb}$ // total sold resource

# Step 2: Surplus Distribution

8. for all $\hat{\alpha}_s \in \hat{\alpha}$ do
9. $\Omega_s = \sum_{b \in B} w(\hat{\beta}_b) x_{sb}$
10. $\Delta_s = \sum_{b \in B} (\rho^b_b - \hat{p}^b_{sb}) x_{sb}$
11. $\rho^s = \sum_{b \in B} \hat{p}^b_{sb} x_{sb} + \Delta (\mu \frac{\Omega_s}{\Omega} + (1 - \mu) \frac{\Omega_s}{\Omega})$
12. $P^s \leftarrow P^s \cup \rho^s$
13. end for
14. **Output:** $P^s = \{\rho^s_1, \ldots, \rho^s_M\}$

payment is defined as follows:

$$\rho^s = \sum_{b \in B} \hat{p}^b_{sb} x_{sb} + \Delta \frac{\Omega_s}{\Omega} = \sum_{b \in B} \hat{p}^b_{sb} x_{sb} + \Delta \frac{\sum_{s \in S} \sum_{b \in B} w(\hat{\beta}_b) x_{sb}}{\Omega_s}$$

We combine the two approaches for surplus distribution: direct-value and proportional-value payment, in a systematic way by using a weighted average with control parameter $\mu \in [0, 1]$. Therefore, in order to determine the final seller payments, the proposed pricing mechanism runs in a number of steps:

- **Market Volume and Surplus (lines 5-7):** Firstly, the mechanism initialises the final seller pricing variable $P^s$ and calculates the market values required for the mechanism’s operation, such as the market generated surplus $\Delta$, and the total volume of goods exchanged by the market $\Omega$. The market generated surplus $\Delta$ is defined as a difference between the total payment made by all the buyers and the total reserve price desired by the sellers for their allocated services. The total market volume $\Omega$ is measured by the total sum of weights of all the exchanged bundles of services.

- **Surplus Distribution (lines 8-13):** Finally, the pricing scheme distributes the surplus among the sellers in addition to their required reserve price. Therefore, the mecha-
Table 4.8: HO-DS Mechanism Example: Seller Pricing Procedure

(a) Market Values

<table>
<thead>
<tr>
<th>Winning Candidate, ( c_{sb} )</th>
<th>Market Details</th>
<th>( \rho_b^j )</th>
<th>( \rho_{sb}^{jp} )</th>
<th>Surplus</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle \text{LS, LC1} \rangle )</td>
<td>( $6.30 )</td>
<td>( $1.86 )</td>
<td>( $4.44 )</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>( \langle \text{SS, MC1} \rangle )</td>
<td>( $1.64 )</td>
<td>( $1.20 )</td>
<td>( $0.44 )</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>( \langle \text{SS, PhD1} \rangle )</td>
<td>( $0.06 )</td>
<td>( $0.06 )</td>
<td>( $0.00 )</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Total: \( \Delta = \$4.88 \), \( \Omega = 41 \)

(b) Individual Seller Values

<table>
<thead>
<tr>
<th>Seller, ( s )</th>
<th>Individual Seller Values</th>
<th>Final Payment, ( \rho_s^x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume ( \Omega_s )</td>
<td>Surplus ( \Delta_s )</td>
<td>Total Price Res.</td>
</tr>
<tr>
<td>( \mu = 0.0 )</td>
<td>( \mu = 0.5 )</td>
<td>( \mu = 1.0 )</td>
</tr>
<tr>
<td>LS</td>
<td>24</td>
<td>$4.44 $1.86</td>
</tr>
<tr>
<td>SS</td>
<td>17</td>
<td>$0.44 $1.26</td>
</tr>
</tbody>
</table>

The mechanism calculates the individual seller’s surplus and volume contributed to the market (lines 9, 10). The surplus is distributed according to the payment rules, depending on the parameter \( \mu \) (line 11). Please note, if the parameter \( \mu = 0 \), the surplus is distributed exclusively based on direct payment rule; similarly, if \( \mu = 1 \), only the proportional-value payment rule applies; any value in between \( 0 < \mu < 1 \) would result in a mixed surplus distribution rule.

Seller Pricing Mechanism Example

We continue the previously discussed example and illustrate the procedure of our proposed seller payment mechanism. The price determination process, discussed in this example, is depicted in Table 4.8.

The mechanism determines that the total surplus in the market is \( \Delta = \$4.88 \) and the total volume of exchanged goods is \( \Omega = 41 \). There values are derived as sums of the individual surpluses and weights of the exchanged bundles of the allocated candidates (Table 4.8a). The payment is further determined for each participating seller based on the selected value of \( \mu \) (Table 4.8b). If the parameter is \( \mu = 0.0 \), the final payment will be derived based on direct payment rule. For example, the payment of the seller SS when \( \mu = 0.0 \) is calculated as follows: \( \$1.26 + (4.88 \times \frac{\$0.44}{\$1.70}) = \$1.70 \), which is a sum of all the
payments of the buyers allocated to SS. If $\mu = 1.0$, the mechanism applies proportional-value payment rule. For example, the payment of the seller LS when $\mu = 1.0$ is calculated as follows: $1.86 + \left( \frac{4.88 \times 24}{41} \right) = $4.72. It assigns a portion of surplus according to the seller’s contributed market volume. When the mixed rule is applied $\mu = 0.5$, the payment is derived as a weighted sum of both rules. For instance, the payment of LS when $\mu = 0.5$ is calculated as follows:

$$1.86 + 4.88 \times \left( 0.5 \times \frac{24}{41} + (1.0 - 0.5) \times \frac{4.44}{4.88} \right) = $5.51$$

In Table 4.8b we can see how the seller payments change with varying value of parameter $\mu$. In this example, it is more interesting for the seller LS to rely on direct payment rule, since her allocated buyers contribute a significant amount of surplus, meaning that LS offers a good add value services in the market, compared to SS. A more detailed study of the seller payment mechanism parametrisation and its impact on the strategic incentives is provided in the experimental section.

### 4.2.3 Theoretical Investigation

The theoretical investigation of the proposed market mechanisms is conducted in this section. We present the detailed proofs of such economic properties as Individual Rationality, Budget Balance, and Buyer Truthfulness, as well as determine mechanisms time complexity. The provided theorems are based on the definitions and formulations presented in Chapter 3.

**Individual Rationality (IR)**

**Theorem 5.** The proposed pricing scheme maintains Individual Rationality property for single-minded buyers declared types.

**Proof.** We need to show that IR property holds as defined in Formula 3.4. Since our proposed pricing scheme does not charge a losing buyer, we focus on the allocated buyers’ pricing. We assume that the second condition required for buyer IR does not hold as follows $\hat{p}_b > \tilde{p}_b$ and provide the proof by contradiction. In such situation, we get the following:

$$\hat{p}_b^{lb} + e(s_b) \times w(\hat{p}_b) > \tilde{p}_b \Rightarrow e(s_b) > e(s_b)$$

It means that the candidate $s_b$ is less competitive than her weakest competitor candidate $s_b$. Therefore, the two sorted lists of allocation candidates $L$, and $\hat{L}$ are identical at least
until the position $z \leq N$ of the competitor candidate $\zeta_{sb}$:

$$\bigcup_{i=1}^{z} L_i \equiv \bigcup_{i=1}^{z} \bar{L}_i \implies X_z = \bar{X}_z$$

It means that the allocation decision variables would be the same until the position $z \leq N$ in the sorted lists of allocation candidates, including the decision for the competitor candidate $\zeta_{sb}$. We know that the competitor candidate can either be a bundle-reservation candidate, or some other candidate from the list $\bar{L}$. In case the it is a bundle reservation candidate, we get the following:

$$\bar{p}_b^v < \bar{p}_{sb}^{br} + e(\zeta_{sb} \times w(\hat{\beta}_b)) \implies \bar{p}_b^v < \bar{p}_{sb}^{br}$$

Thus, the bid would violate RPC of the seller $\bar{s}$. In case it is some other candidate, we know that both $\varsigma_{sb}$ and $\zeta_{sb}$ have to satisfy the problem constraints in order to be selected as the competitors. Since, there was no weaker candidate than $\zeta_{sb}$, there is not enough resource available to satisfy the the buyer bid $\hat{\beta}_b$ at some position $\bar{z} > z$ in the sorted list $L$. Consequently, the final buyer’s price is always lower than her reported price valuation: $\rho_b^v \leq \bar{p}_b^v$.

Hence, the proposed pricing mechanism is IR for single-minded buyers.

\begin{proof}

We need to show that IR property holds as defined in Formula 3.5. We will show that IR property holds for the seller by proving that the final price of an allocated buyer cannot be lower than the seller’s bundle price reservation: $\rho_b^v \geq \bar{p}_{sb}^{br} x_{sb}, \forall b \in B, \forall s \in S$. We assume that the final price of some allocated buyer $\rho_b^v < \bar{p}_{sb}^{br}$, where $x_{sb} = 1$, and provide proof by contradiction. In such situation, we get:

$$\bar{p}_{sb}^{br} + e(\zeta_{sb}) \times w(\hat{\beta}_b) = \bar{p}_{sb}^{br}$$

We know that $\zeta_{sb}$ has satisfied the problem constraints in order to be selected as a competitor candidate. Thus, her sorting criteria value is at least $e(\varsigma_{sb}) = 0$, since RPC was
satisfied. Consequently, we get:

\[
\begin{align*}
& e(\zeta_{sb}) < \frac{\hat{p}_{br}^{eb} - \hat{p}_{br}^{es}}{w(\beta_b)} \quad \Rightarrow \quad \frac{\hat{p}_{br}^{es} - \hat{p}_{br}^{eb}}{w(\beta_b)} > 0 \quad \Rightarrow \quad \hat{p}_{sb}^{br} > \hat{p}_{sb}^{br}
\end{align*}
\]

Therefore, the bundle reservation price of the initially allocated seller \(s\), has to be greater than the one of the seller \(\tilde{s}\) from a competitor candidate \(s_{\tilde{b}}\). Hence, the following must hold:

\[
\frac{\hat{p}_{b}^{v} - \hat{p}_{sb}^{br}}{w(\beta_b)} < \frac{\hat{p}_{b}^{v} - \hat{p}_{sb}^{br}}{w(\beta_{\tilde{b}})} \quad \Rightarrow \quad e(\zeta_{sb}) < e(\zeta_{sb})
\]

It means that the candidate \(e(\zeta_{sb})\) would be ordered before \(e(\zeta_{sb})\) in the sorted list \(L\), and the bid \(\beta_b\) would have been allocated to the seller \(\alpha_{\tilde{s}}\) instead of \(\alpha_{s}\). Hence, the final buyer price is always \(\rho_{b}^{l} \geq \hat{p}_{sb}^{br} x_{sb}\).

Given the way the seller’s payment is determined (line 11), we need to prove that the distribution of surplus is always non-negative, as follows:

\[
\Delta(\mu \frac{\Omega_{s}}{\Omega} + (1 - \mu) \frac{\Delta s}{\Delta}) \geq 0
\]

Since, \(\tilde{c}_{bg}^{d} \geq 0, \forall g \in G, \forall b \in B\), and \(f_{g}^{r} > 0, \forall g \in G\), the following must hold:

\[
\begin{align*}
\Omega_{s} &= \sum_{b \in B} \sum_{g \in G} (\tilde{c}_{bg}^{d} f_{g}^{r})^{m} x_{sb} \geq 0 \\
\Omega &= \sum_{s \in S} \sum_{b \in B} \sum_{g \in G} (\tilde{c}_{bg}^{d} f_{g}^{r})^{m} x_{sb} \geq 0
\end{align*}
\]

Given that \(\rho_{b}^{l} \geq \hat{p}_{sb}^{br} x_{sb}, \forall s \in S, \forall b \in B, \tilde{c}_{bg}^{d} \geq 0, \forall g \in G, \forall b \in B, \) and \(\hat{p}_{sg}^{br} \geq 0, \forall g \in G, \forall s \in S\), we get the following:

\[
\begin{align*}
\Delta_{s} &= \sum_{b \in B} (\rho_{b}^{l} - \sum_{g \in G} \hat{p}_{sg}^{br} \tilde{c}_{bg}^{d}) x_{sb} \geq 0 \\
\Delta &= \sum_{b \in B} \rho_{b}^{l} - \sum_{s \in S} \sum_{b \in B} \sum_{g \in G} \hat{p}_{sg}^{br} \tilde{c}_{bg}^{d} x_{sb} \geq 0
\end{align*}
\]

Consequently, given that \(0 \leq \mu \leq 1\), we get that surplus distribution is always positive. Hence, the seller payment is always \(\rho_{s}^{*} \geq \sum_{b \in B} \hat{p}_{sb}^{br} x_{sb}, \forall s \in S\).

Hence, the proposed pricing mechanism is IR for the sellers’ declared types.
Budget Balance (BB)

Theorem 7. The proposed market mechanism maintains Budget Balance.

Proof. We need to show that BB property holds as defined in Formula 3.6. To prove the equivalence, we replace the seller pricing by the definition, and transform the right side of the equation, as follows:

\[
\sum_{s \in S} \rho_s^x = \sum_{s \in S} \left( \sum_{b \in B} \rho_{sb}^x x_{sb} + \Delta \left( \frac{\Omega_s}{\Omega} + (1 - \mu) \frac{\Delta_s}{\Delta} \right) \right) = \\
\sum_{s \in S} \sum_{b \in B} \rho_{sb}^x x_{sb} + \Delta \mu \sum_{s \in S} \Omega_s \Omega + \Delta (1 - \mu) \sum_{s \in S} \Delta_s \Delta
\]

Given the definitions of the market and individual seller volumes and surplus, we can derive the following:

\[
\Omega = \sum_{s \in S} \sum_{b \in B} w(\hat{\beta}) x_{sb} \quad \Rightarrow \quad \Omega = \sum_{s \in S} \Omega_s
\]

\[
\Omega_s = \sum_{b \in B} w(\hat{\beta}) x_{sb} \quad \Rightarrow \quad \Omega_s = \sum_{s \in S} \Omega_s
\]

\[
\Delta = \sum_{b \in B} \rho_b^x - \sum_{s \in S} \sum_{b \in B} \rho_{sb}^x x_{sb} \quad \Rightarrow \quad \Delta = \sum_{s \in S} \Delta_s
\]

Consequently, we can reduce the fractions corresponding to the volumes and surpluses, and replace the generated surplus \( \Delta \) by its definition. As a result, we get the following:

\[
\sum_{s \in S} \sum_{b \in B} \rho_{sb}^x x_{sb} + \Delta = \sum_{s \in S} \sum_{b \in B} \rho_{sb}^x x_{sb} + \sum_{b \in B} \rho_b^x - \sum_{s \in S} \sum_{b \in B} \rho_{sb}^x x_{sb} = \sum_{b \in B} \rho_b^x
\]

Hence, the proposed mechanism maintains BB property. \( \square \)

Computational Tractability (CT)

The designed double-sided market mechanism for trading homogeneous cloud services determines market outcomes in polynomial time in the size of its input. In particular, the HO-DS-A scheme makes the allocation decisions in \( O(|MN| \log |MN|) \) time, where the major computation is in order to find the sorted order of allocation candidates in the list \( L \). The HO-DS-PB mechanism calculates the buyer prices in \( O(|MN^2|) \) due to reusing the initial sorting order of the allocation candidates in the list \( L \). The HO-DS-PS scheme computes the seller payments in linear time \( O(|M|) \). Hence, the total mechanism’s time...
complexity is $O(|MN^2|)$ which is an extremely time-efficient mechanism, given the non-trivial problem complexity.

**Buyer Truthfulness (BT)**

The truthfulness in buyers’ declarations is demonstrated by achieving the minimum required truthfulness conditions, outlined in [90]. Specifically, we prove that the proposed allocation function is monotonic in buyers’ declarations, and the proposed buyer pricing mechanism derives payments based on critical-value.

**Theorem 8.** *The proposed allocation scheme is monotone in declarations of single-minded buyers.*

\textit{Proof.} The provided proof is derived based on Definition 3.4.1. We consider two reported bids of the same buyer $\hat{\beta}_{b*} = (\hat{C}_v^d, \hat{p}_v^*)$, and $\check{\beta}_{b^*} = (\check{C}_v^d, \check{p}_v^*)$, s.t. $\check{\beta}_{b^*} \succeq \hat{\beta}_{b^*}$. When changing the reported type from $\check{\beta}_{b^*}$ to $\hat{\beta}_{b^*}$, the sorted list of allocation candidates may differ only in the candidates associated with the changed bid moving up or down in the sorted list (all the other allocation candidates remain unchanged and preserve their positions). We denote by $L^*$, and $L^-$, the sorted lists of allocation candidates with the reported bids $\check{\beta}_{b^*}$, and $\hat{\beta}_{b^*}$, respectively. We assume that the monotonicity property does not hold and provide the proof by contradiction.

**Scenario 1:** We consider the situation when $\hat{\beta}_{b^*}$ was allocated $\sum_{s \in S} x_{sb^*} = 1$, but its changed type $\check{\beta}_{b^*}$ has lost $\sum_{s \in S} x_{sb^*} = 0$. Hence, all the allocation candidates associated with this bid $\check{\beta}_{b^*}$, $\forall s \in S$ have violated at least one of the following problem constraints:

- **RPC:** Since $\hat{\beta}_{b^*}$ was allocated $\sum_{s \in S} x_{sb^*} = 1$ to some seller $s \in S$, it has satisfied RPC of this seller, i.e. $\hat{p}_v^* \geq \check{p}_v^*$. Given that $c_{d*}^d \leq c_{d^*}^d, \forall g \in G$, we get that $\check{p}_v^* \geq \check{p}_v^*$.

Thus, we can conclude the following:

\[
\begin{align*}
\hat{p}_v^* &\geq \check{p}_v^* \\
\check{p}_v^* &\leq \check{p}_v^* \\
\check{p}_v^* &\geq \check{p}_v^*
\end{align*}
\]

Hence, if the bid $\hat{\beta}_{b^*}$ satisfied RPC of the seller $s \in S$, then $\check{\beta}_{b^*}$ would also satisfy RPC of this seller, which is true for any seller $\forall s \in S$.

- **ARC:** Since $\hat{\beta}_{b^*}$ was allocated $\sum_{s \in S} x_{sb^*} = 1$, there was enough available resource from the seller $s \in S$ at the considered position $z^* \leq MN$ in the list $L^*$. When $\hat{\beta}_{b^*}$ is
changed to $\hat{\beta}_{b^*}$, all the sorting criteria values of the associated allocation candidates change. Since, $\hat{\beta}_{b^*} \geq \hat{\beta}_{b'}, f^*_g > 0, \forall g \in G$ and $m \geq 0$, we get:

$$
\begin{align*}
\hat{p}_{b^*}^v < \hat{p}_b^v \\
\sum_{g \in G} c^d_{bg} f^*_g m^m \geq \sum_{g \in G} c^d_{b'g} f^*_g m^m \\
\Rightarrow e(\xi_{sb^*}) \geq e(\xi_{sb'}), \forall s \in S
\end{align*}
$$

It means that all the allocation candidates associated with the bid $\hat{\beta}_{b^*}$ would become more competitive than the ones associated with the bid $\hat{\beta}_{b'}$. Therefore, the sorted list of candidates $L^*$ would be different from $L'$ in the candidates associated with $\hat{\beta}_{b^*}$ moving into higher positions in the list. As a result, a candidate associated with $\hat{\beta}_{b^*}$ would be considered at some position $z^* \leq z'$ in the list $L^*$, and would also satisfy ARC.

We have shown that if a bid $\hat{\beta}_{b'}$ is allocated $\sum_{s \in S} x_{sb'} = 1$, then the modified bid $\hat{\beta}_{b^*}$, s.t. $\hat{\beta}_{b^*} \geq \hat{\beta}_{b'}$ would also satisfy all the problem constraints and would also be allocated $\sum_{s \in S} x_{sb^*} = 1$.

**Scenario 2:** We consider the opposite situation, when a bid $\hat{\beta}_{b^*}$ was denied $\sum_{s \in S} x_{sb^*} = 0$, and the modified bid $\hat{\beta}_{b'}$ was allocated $\sum_{s \in S} x_{sb'} = 1$. In case $\hat{\beta}_{b^*}$ was denied due to violated RPC, then $\hat{\beta}_{b^*}$ would violate it as well due to $\tilde{p}_{b^*}^v \geq \tilde{p}_b^v$ and $\tilde{p}_{sb'}^v \geq \tilde{p}_{sb^*}^v, \forall s \in S$. If $\hat{\beta}_{b^*}$ was denied due to insufficient amount of available services, when considered at position $z^*$ in the list $L^*$, then $\hat{\beta}_{b^*}$ would violate ARC as well at some position $z' \geq z^*$ in the list $L'$ due to lower resource availability and $\tilde{C}^d_{b^*} \geq \tilde{C}^d_{b'}$. Hence, if a bid $\hat{\beta}_{b^*}$ is denied $\sum_{s \in S} x_{sb^*} = 0$, then the modified bid $\hat{\beta}_{b'}$, s.t. $\hat{\beta}_{b^*} \geq \hat{\beta}_{b'}$ will also be denied in the market $\sum_{s \in S} x_{sb'} = 0$.

Therefore, the HO-DS-A scheme is monotone in single-minded buyer declarations.

\[\square\]

**Theorem 9.** The proposed pricing scheme determines critical-value payments for single-minded buyers.

**Proof.** The provided proof is based on Definition 3.4.2. We assume that the required critical-value conditions do not hold and provide the proof by contradiction.

**Scenario 1:** We consider a situation when a buyer wins $\sum_{s \in S} x_{sb} = 1$ with the price valuation $\tilde{p}_b^v < \tilde{p}_b^v$. Since the final buyer’s price $\tilde{p}_b^v$ is derived as the minimum price required
to outbid her weakest competitor \( \xi_{sb} \in \tilde{L}^c \), we get the following:

\[
\tilde{p}_b^v < \tilde{p}_{sb}^{br} + e(\xi_{sb}) \times w(\tilde{\beta}_b) \quad \Rightarrow \quad \frac{\tilde{p}_b^v - \tilde{p}_{sb}^{br}}{w(\tilde{\beta}_b)} < e(\xi_{sb}) \quad \Rightarrow \quad e(\xi_{sb}) < e(\xi_{sb})
\]

As a result, a candidate \( \xi_{sb} \) would be placed at a lower position in the sorted list \( L \) compared to her weakest competitor \( \xi_{sb} \). We know that the competitor candidate satisfies the problem constraints and gets allocated in the list \( L \). Since, \( x_{sb} = 1 \) and \( e(\xi_{sb}) < e(\xi_{sb}) \), we conclude that the two sorted lists of allocation candidates \( L \) and \( \tilde{L} \) are identical at least until the position \( z \leq MN \) of the competitor candidate \( \xi_{sb} \):

\[
\bigcup_{i=1}^{z} L_i = \bigcup_{i=1}^{z} \tilde{L}_i \quad \Rightarrow \quad X_z = \tilde{X}_z
\]

Therefore, the allocation decision would be the same until the allocation candidate in position \( z \leq MN \), including the weakest competitor candidate \( \xi_{sb} \), which would be \( x_{sb} = 1 \). However, it is not possible for both candidates \( \xi_{sb} \) and the corresponding weakest competitor \( \xi_{sb} \) to be allocated at the same time. Regardless of the allocated seller, either \( s = \bar{s} \) or \( s \neq \bar{s} \), if there is enough resource available to allocate both in the list \( L \), the corresponding bundle-reservation candidate \( \xi_{sb}^{br} \) would have been allocated in \( \tilde{L} \) and selected as the weakest competitor.

If the weakest competitor is the bundle-reservation candidate, given that \( \tilde{p}_b^v < \tilde{p}_b^b \), we get the following:

\[
\tilde{p}_b^v < \tilde{p}_{sb}^{br} + e(\xi_{sb}^{br} \times w(\tilde{\beta}_b)) \quad \Rightarrow \quad \tilde{p}_b^v < \tilde{p}_{sb}^{br}
\]

Such bid would violate RPC of the seller \( \bar{s} \), and would not be allocated at the first place.

What if there is some other seller \( s^* \neq \bar{s} \), s.t. the bundle reservation of the bid \( \tilde{\beta}_b \) is lower: \( \tilde{p}_{s^*b}^{br} < \tilde{p}_{sb}^{br} \). We know that the bundle reservation candidates are inserted in the list \( \tilde{L} \) based on the sorted order in \( L \). Hence, the following holds:

\[
e(\xi_{s^*b}) < e(\xi_{sb}) \quad \Rightarrow \quad \frac{\tilde{p}_b^v - \tilde{p}_{s^*b}^{br}}{w(\tilde{\beta}_b)} < \frac{\tilde{p}_b^v - \tilde{p}_{sb}^{br}}{w(\tilde{\beta}_b)} \quad \Rightarrow \quad p_{s^*b}^{br} > p_{sb}^{br}
\]

Hence, there is no bundle reservation candidate with the price valuation below the one of the selected bundle reservation competitor candidate.

Hence, any price valuation \( \tilde{p}_b^v < \tilde{p}_b^b \) results in the bid \( \tilde{\beta}_b \) being denied \( \sum_{s \in S} x_{sb} = 0 \).

**Scenario 2:** We consider the second case, and analyse the situation when a bid is denied \( \sum_{s \in S} x_{sb} = 0 \) with the price valuation \( \tilde{p}_b^v \geq \tilde{p}_b^b \). We know that the weakest candidate \( \xi_{sb} \) has satisfied the problem constraints in order to be selected. Similarly, we get that
any price valuation $\hat{p}^v_i \geq \rho^b_i$ would result in a candidate associated with the bid $\hat{\beta}_b$ being more competitive: $e(\varsigma_{sb}) \geq e(\varsigma_{s_b})$. Therefore, it would appear in the sorted list $L$ before her weakest competitor $\varsigma_{s_b}$. As a result, there will be at least one allocation candidate associated with the bid $\hat{\beta}_b$ that would satisfy the problem constraints and get allocated. Hence, any price valuation $\hat{p}^v_i \geq \rho^b_i$ would result in the bid $\hat{\beta}_b$ being allocated $\sum_{s \in S} x_{sb} = 1$.

Therefore, the proposed pricing mechanism determines the buyer prices based on critical-value.

\[ \Box \]

4.2.4 Experimental Investigation

In this section, we analyse the performance of the proposed mechanism in an experimental setting. We investigate the allocation quality of the proposed approximation allocation scheme, and to study the seller’s strategic misreporting opportunity in the market.

Experimental Setting

The setup of our simulation experiment is presented in the Table 4.4. We study the dynamics of the market and the associated outcomes in a wide range of market scenarios. The considered market setup, including the simulated supply and demand models, is provided below.

**Market Model:** We consider a double-sided market, which allows its participants, i.e. buyers and sellers to exchange the infrastructure cloud services of three different types $K = 3$: Small, Medium and Large VMs. The underlying low-level resources of these VMs scale according to the following strict relationship: $F^s = \{1, 2, 4\}$. This model was inspired by general purpose pre-defined VMs for running Windows OS on Rackspace Cloud.

**Demand Model:** We experiment with different amounts of buyer bids in the market $N$, which are determined as the number of allocation candidates $C$ per each participating seller $M$. It allows us to control the number of allocation candidates and keep a consistent number of allocation candidates across the experimental scenarios with different numbers of sellers. We employ uniform distribution functions in order to generate the buyer’s requested bundle of services and the corresponding price valuation, which allows us to reflect variability for continuous spaces with huge number of combinations.

**Supply Model:** In order to model the market supply, we simulate different number of sellers $M$. We experiment with various provisioning scenarios $p_g$, achieved by balancing the total market’s supply against the previously generated demand. The amounts of supplied
Table 4.9: HO-DS Experimental Setting

**Market Model:**

<table>
<thead>
<tr>
<th>Resource Types</th>
<th>( G = { 1: \text{Small}, 2: \text{Medium}, 3: \text{Large} } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K = 3 )</td>
<td></td>
</tr>
<tr>
<td>Relative Relation</td>
<td>( F^r = { 1, 2, 4 } )</td>
</tr>
</tbody>
</table>

**Demand Model:**

<table>
<thead>
<tr>
<th>Number of Buyers</th>
<th>( N = C/M ), where</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C = { 32, 64, 128, 256, 512 } )</td>
<td></td>
</tr>
<tr>
<td>Demanded Capacity</td>
<td>( c_{bg}^d = U(0, 10), \forall \in G )</td>
</tr>
<tr>
<td>Price Valuation</td>
<td>( p_b^V = U(0.0, 0.25) \times \sum_{g \in G} c_{bg}^d f_s^r )</td>
</tr>
</tbody>
</table>

**Supply Model:**

<table>
<thead>
<tr>
<th>Number of Sellers</th>
<th>( M = { 1, 2, 4, 8, 16 } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplied Capacity</td>
<td>( p_g = { 0.25, 0.5, 0.75, 1.0 } )</td>
</tr>
<tr>
<td>( F^p = (f_{1}^p, \ldots, f_{M}^p) ), where ( f_{g}^p = U(0.0, 1.0) )</td>
<td></td>
</tr>
</tbody>
</table>

| Price Reservation | \( p_{s1}^r = U(0.0, 0.25) \times z \) |
|                   | \( p_{s2}^r = U(0.25, 0.5) \times z \) |
|                   | \( p_{s3}^r = U(0.5, 1.0) \times z \) |
| Price Reservation | \( z = \{ 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5 \} \) |

| Seller Misreporting | \( q_g = \{ 0.5, 0.75, 1.0, 1.25, 1.5 \}, \forall \in G \) |

**Metrics and Mechanisms:**

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Social Welfare, Resource Utilisation, Total Cost, Seller Utility, Computation Time, Buyer Fulfilment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation Mechs.</td>
<td>HO-DS where ( m = { 0.0, 0.25, \ldots, 3.75, 4.0 } )</td>
</tr>
<tr>
<td></td>
<td>Optimal CPLEX solver</td>
</tr>
<tr>
<td>Pricing Mechs.</td>
<td>HO-DS where ( \mu = { 0.0, 0.25, 0.5, 0.75, 1.0 } )</td>
</tr>
<tr>
<td></td>
<td>K-pricing where ( k = { 0.0, 0.25, 0.5, 0.75, 1.0 } )</td>
</tr>
</tbody>
</table>
services in each seller’s ask are determined as a random portion of the total market’s supply, defined as follows:

\[ ||F^p||_s := \frac{f_s^p}{\sum_{s' \in S} f_{s'}^p} \]

where \( F^p = (f^p_1, \ldots, f^p_M) \), \( f_s^p = \mathcal{U}(0,0,1,0) \) is a vector of randomly generated numbers from 0 to 1, and \( ||F^p||_s \) is its normalising factor to sum to unity. The seller’s price reservations are derived by uniform distribution function. The generated random values are scaled according to the price comparative ratio \( z \), which permits us to simulate different price reservation scenarios in the market.

Overall, we simulate 2240000 market inputs, where 11200 market scenarios are repeated 200 times each. In the allocation experiment, the market inputs are solved by 17 different parametrisations of our proposed HO-DS mechanism and by the optimal solver. In the pricing experiment, we simulate five misreporting scenarios for each traded service type, i.e. \( 5^3 = 125 \) misreporting situations. These scenarios are solved by 5 different parametrisations of our proposed HO-DS CV-SD mechanism and by 5 parametrisations of a well-known K-pricing mechanism [124]. In order to avoid statistical noise, the average metrics are calculated for each market scenario across the conducted 200 trials.

**Allocation Mechanism Experiment**

We study the quality of allocation outcomes achieved by our proposed HO-DS allocation mechanism and determine the best mechanism’s parametrisation based on various objectives, such as better resource utilisation, social welfare, or the buyer fulfilment.

**Closeness to Optimality**

Due to the high problem complexity and the computation time constraint, we could collect only some optimal results. In Figure 4.9, we plot representative results for the closeness of the social welfare and average resource utilisation to the calculated optimal results. We can observe that the number of allocation candidates \( C \) and the the buyer requests granularity are the two major factors that affect the allocation quality. When the number of allocation candidates is high, the chance that the approximation mechanism denies a candidate which is a part of an optimal solution is greater. When there are fewer bids of a bigger size relative to the sellers’ offers (low bids granularity), there is a higher negative impact of non-optimal allocation decisions on the social welfare. Overall, the proposed greedy heuristic determines near-optimal allocations in majority of market scenarios. The worst case approximation of 87% was observed in the large markets with
Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

Figure 4.9: Closeness to Optimality in terms of Social Welfare (a, c) and Average Resource Utilisation (b, d). Market Scenario for \( m = 0.25 \).

(a) \( z = 0.0, p = 0.25 \)  
(b) \( z = 0.0, p = 0.25 \)  
(c) \( C = 64, M = 2 \)  
(d) \( C = 64, M = 2 \)

In Figure 4.9c we can see that when the resource in the market is not scarce \( (p = 1.0) \), the heuristic mechanism essentially allocates all the buyer’s requests similar to the optimal allocation scheme and the deviation from the optimal solution is low (the slight deviation from the optimal is mainly due to buyer request indivisibility constraint). When the resource becomes scarce \( (p < 1.0) \), our proposed heuristic allocation mechanism is more likely to produce suboptimal allocations. We can also see that our mechanism is more efficient when the price comparative ratio is high \( (z \geq 1.0) \), because the number of feasible allocation candidates is reduced due to the increased sellers’ price reservations.

The closeness to optimality of the average resource utilisation is very related to the social welfare optimality. The constraints applied to the amounts of offered resources play a crucial role in the attained allocation efficiency.
Mechanism’s Parametrisation

We investigate the impact that the mechanism’s parametrisation has on the approximation quality of the allocation outcomes. Since some of the optimal outcomes are unknown, we compare the allocation outcomes determined by our approximation mechanism among themselves, where 100% indicates the most efficient approximation result.

In Figures 4.10a-4.10b, we can observe that the social welfare and the average resource utilisation have a very similar dynamics. The most efficient parametrisation varies based on the market scenario but it is typically achieved when \( m \leq 1.0 \). When there are fewer number of sellers \( M \) (the bids granularity is high), the better allocations are achieved with 0.75 \( \leq m \leq 1.0 \) parametrisation. However, when the bids become less granular, favouring larger requests has a positive impact on the market allocation. It happens because, when the bids are comparatively large, allocating larger bundles first improves the chance for them to be allocated, while leaving some spare services to be consumed by
Chapter 4. Market Mechanisms for Homogeneous Infrastructure Cloud Services

\( p_1 = 0.25 \)  
\( p_1 = 0.5 \)  
\( p_1 = 0.75 \)  
\( p_1 = 1.0 \)

Figure 4.11: Impact of Resource Provisioning on Average Resource Utilisation. Market Scenario: mechanism with \( m = 1.0, G = 512, M = 4, z = 0.0 \).

(a) Social Welfare  
(b) Average Resource Utilisation

Figure 4.12: Resource Provisioning and Mechanism Parametrisation: Closeness to Best Approximation Result in terms of Social Welfare (a) and Average Resource Utilisation (b). Market Scenario: \( C = 512, z = 0.25, M = 4 \).

smaller buyer requests. In Figure 4.10c we plot the relative buyer fulfilment achieved by different mechanism’s parametrisations. It can be clearly seen that bigger parameter \( m \) allows to improve the number of satisfied buyers, since it favours smaller requests.

**Resource Provisioning Impact**

In Figure 4.11, we plot the interrelation between the provisioning level of services of different types and the average resource utilisation. We can observe a very similar pattern to the single-sided market results, where a more well-balanced provisioning results in a better resource utilisation. At these graphs, we show a snapshot of the resource utilisation in the whole market which includes all the participating sellers. However, an individual seller with a more balanced provisioning is likely to achieve a better resource utilisation than the seller who makes a bad estimate in terms of market demand. We should also note that the best average resource utilisation gradually drops when the number of seller offers
increase due to the growing number of available resource constraints imposed by different sellers. Nevertheless, the lowest best resource utilisation is 95%, which is observed in the market scenario when the resources are very scarce \( p_1 = 0.25, p_2 = 0.25, p_3 = 0.25 \) (Figure 4.11a).

In Figure 4.12, we depict how the relative optimality of the mechanism (in terms of the social welfare and the average resource utilisation) changes with various mechanism parametrisations. We observe a similar behaviour to the single-sided market, where the most efficient mechanism’s parametrisation depends on the size of the scarce resource. When small or medium VMs are scarce, favouring larger requests with \( m \leq 0.5 \) results in a more efficient allocation. However, when the number of bids is small and they are of a bigger size relative to the sellers’ offers (\( C = 512, M = 16 \)), it is always more efficient to favour larger requests, even if the large VMs are scarce.

**Computation Time**

In Figure 4.13, we present the computation time of our proposed HO-DS allocation mechanism compared to the optimal solver time. Due to a high problem complexity, the optimal results were not achieved within a feasible time. However, an approximation mechanism is extremely time-efficient, where all the markets were cleared within 1 ms on average.

**Pricing Mechanism Experiment**

In this experiment, we intend to investigate the seller’s strategic misreporting and identify the pricing mechanism’s parametrisation that limits the strategic opportunity the most. We also compare the results obtained by our proposed HO-DS mechanism with different
parametrisations of K-pricing mechanism.

We measure and compare the utility of a misreporting seller agent in a wide range of misreporting scenarios. Figures 4.14 and 4.15 illustrate the utility obtained by a misreporting seller relative to her utility when telling truth (Y axis). Any value above 1.0 indicates that the seller was successful in applying a misreporting strategy and her utility was improved; similarly, values below 1.0 suggest that the seller was penalised and experienced loss in utility. Over the X axis, we depict different degrees of over and understating strategies based on the mean square difference of the three price reservations from the truthful prices.

**Seller Misreporting and Market Scenarios**

We plot some representative results for our proposed pricing mechanism which uses a mixed rule with $\mu = 0.5$ in Figure 4.14, and analyse the dominant strategic behaviour in various market scenarios.

- **Price Comparative Ratio $z$**: In Figures 4.14a, 4.14b, and 4.14c, we show how the seller misreporting opportunity changes with various price comparative ratios $z$. When the price ratio is low $z = 0.25$ (Figure 4.14a), the overstating strategy allows a seller to improve her utility for up to 20%. It would encourage the participating sellers to increase their price reservations, which in other words would result in a market situation with higher $z$. However, in Figures 4.14b and 4.14c, we can clearly see that when $z \geq 0.75$, the seller strategic opportunity becomes very slim (almost completely eliminated). It happens because the market generated surplus is so small that overstating strategy is likely to result in reduced number of allocations (lower payments and reduced utility), and understating is likely to result in payments below the truthful price reservation (negative utility). Therefore, the sellers have an incentive to overstate their truthful price reservation until a natural balance is achieved in the market.

- **Number of sellers $M$ and buyers $N$**: We can observe that a seller has a greater strategic power when the competition at the supply side of the market is restricted to a small number of sellers (Figure 4.14d). We can see that the utility obtained by overstating strategy can be almost double of a truthful utility. An extreme case would be a monopoly market with a single seller, the market scenario that was carefully studied in the previous section. However, in Figure 4.14e, we can see that with only four competitive sellers the strategic power of an individual seller is significantly reduced.
Given the same provisioning market scenario, the smaller number of buyers $N$ result in a higher seller misreporting opportunity (Figure 4.14g). It happens because due to our experimental setup, the buyer bids are of a bigger size relative to the sellers’ offers. Therefore, there are less sellers capable to satisfy the buyers in terms of available resources. As we observed before, when the number of competitive sellers is smaller, the corresponding misreporting opportunity will be higher.

- **Provisioning level $p$:** Figures 4.14j, 4.14k, and 4.14l show the seller’s strategic opportunity in the markets with different resource provisioning levels. We can see that with reducing services scarcity, the strategic opportunity tend to decrease, and the seller gets a more and more severe penalty for misreporting (Figure 4.14l). In the following discussion we will see that the opportunity presented by the overstating strategy (Figure 4.14j) is mainly due to the negative impact of proportional-value pricing rule.

**Seller Misreporting and Mechanism’s Parametrisation**

Figure 4.15 provides an illustrative comparison of the seller’s strategic opportunity allowed by HO-DS and K-pricing mechanisms, including various parametrisations.

**HO-DS mechanism:** We can see that the understating strategy is not a good strategic choice in our proposed mechanism and such strategy does not allow the seller to improve her utility, but rather harms the seller (Figures 4.15a-4.15d). It holds true in all the tested market scenarios because when the price reservation is lowered, the payments determined for the buyers may only reduce, since the competition on the buyer side reduces. As a result, the surplus that goes to the seller can only get lower compared to the surplus generated when the seller reports truthfully. The overstating strategy is not very efficient when the surplus distribution relies more on a direct-value payment $\mu < 0.5$ (Figure 4.15b) and only a marginal utility improvement can be achieved. It happens because the buyer-based pricing always grants to the seller the exact amount of surplus that she contributed to the market and overstating strategy is likely to lower it down. Proportional value payment rule $\mu = 1.0$ (Figure 4.15d) allows the seller to benefit from overstating her price reservations, mainly in the markets with scarce resources ($p < 1.0$) and competitive buyer price valuations ($z < 1.0$). In such situation, the difference between the sellers’ price reservations and the buyers’ price valuations is relatively large, and overstating the price reservation permits the seller to secure a greater portion of buyers’ payments (by setting the reserve price bound higher), while contributing less surplus to the market. Due to the high scarcity of resources in the market, the seller is still likely to maintain a similar proportion.
Figure 4.14: Seller Misreporting Opportunity in various market scenarios. Initial Market Scenario: \( \mu = 0.5, C = 512, M = 8, Z = 0.5, P = 1.0 \)
Figure 4.15: Seller Misreporting Opportunity in HO-DS Surplus Distribution (a, b, c, d) and K-pricing Mechanism (e, f, g, h) with different parametrisations. Market Scenario: $G = 512, M = 8, Z = 0.5, P = 0.5$
of sold resources, which permits to obtain a similar portion of market surplus in addition to the increased price reservation for the sold resources. We can notice that the mixed rule $\mu = 0.5$ (Figure 4.15b) provides a lower misreporting opportunity than the proportional-value $\mu = 1.0$ (Figure 4.15d), but has a more severe punishment for misreporting compared to the direct-value payment, which demonstrates its practical application.

**K-pricing:** We can observe a very interesting behaviour of k-pricing mechanism when $k \geq 0.5$ (Figures 4.15f-4.15h), which permits to benefit from understating strategy. In the markets with highly competitive buyer price valuations ($z \leq 0.5$), reducing the price reservation allows the seller to become more competitive and to sell more resources (when resource is overprovisioned) and/or be allocated the buyers with higher valuation (when resource is underprovisioned). Given that the seller secures more than 50% of the price difference, the final seller payment is likely to improve leading to higher utility. When resource is less scarce (increasing $p$), the potential gain out of understating is increasing due to the increased competition among the sellers for the buyers. Although, the seller’s strategic misreporting achieved when $k = 0.25$ (Figures 4.15e) is comparable to the one of our proposed mechanism when $\mu \leq 0.25$, our pricing mechanism is truthful in buyer declarations, which is not guaranteed by K-pricing.

### 4.3 Summary

This chapter provided two market mechanisms for trading homogeneous cloud services in single-sided and double-sided markets. The major design aspects and the qualitative characteristics of the proposed market mechanisms are summarised in Table 4.10.

The single-sided market mechanism (HO-SS) applies greedy heuristic for combinatorial bids allocation, where the seller can express the minimum desired prices, i.e. reservation prices, for the traded services. The corresponding pricing mechanism is developed based on critical-value payment principles. The double-sided market mechanism applies an approach for allocation of composite candidates, i.e. ask-bid couples, in a greedy sorted order. The pricing scheme derives the buyer payments based on a new double-sided critical-value payment mechanism, while the seller payments are realised via market surplus distributed based on mixed rules.

Our theoretical evaluation of the designed mechanisms revealed that both maintain all the required mechanism design feasibility constraints. In particular, the mechanisms are Individually Rational (IR) for buyers and sellers, maintain Budget Balance (BB) and clear in polynomial computation time, i.e. Computationally Tractable (CT). We have also
Table 4.10: Market Mechanisms for Homogeneous Cloud Services: Summary

(a) Mechanism Design Aspects

<table>
<thead>
<tr>
<th>Market Mechanism</th>
<th>Allocation</th>
<th>Pricing</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO-SS</td>
<td>Combinatorial Greedy (sorted order of buyer bids)</td>
<td>Critical-value pricing (seller reserve price)</td>
</tr>
<tr>
<td>HO-DS</td>
<td>Combinatorial Greedy (sorted order of composite candidates: ask-bid couples)</td>
<td>Buyer pricing: Double-sided critical-value, Seller Payment: Mixed Rule Surplus Distribution</td>
</tr>
</tbody>
</table>

(b) Mechanism Design Qualitative Characteristics

<table>
<thead>
<tr>
<th>Economic Properties</th>
<th>IR</th>
<th>BB</th>
<th>CT</th>
<th>AE</th>
<th>TB</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO-SS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>□</td>
<td>●</td>
<td>n/a</td>
</tr>
<tr>
<td>HO-DS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>□</td>
<td>●</td>
<td>□</td>
</tr>
</tbody>
</table>

- ● - property is theoretically guaranteed
- □ - property is approximated (near-optimal / near-truthful)

(c) Mechanism Parametrization: Market Operators Decision Support

<table>
<thead>
<tr>
<th>Param. Value</th>
<th>Impact</th>
<th>Description / Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75 ≤ m ≤ 1.0</td>
<td>Favour larger bids</td>
<td>Better Social Welfare and Resource Utilisation</td>
</tr>
<tr>
<td>m ≥ 1.0</td>
<td>Favour smaller bids</td>
<td>Satisfy more Buyers’ Requests</td>
</tr>
<tr>
<td>μ = 0.0</td>
<td>Direct-value Payment</td>
<td>Seller overstating opportunity, Severe punishment for misreporting</td>
</tr>
<tr>
<td>μ = 0.5</td>
<td>Mixed Pricing Rule</td>
<td>Little overstating opportunity, Reasonable punishment for misreporting</td>
</tr>
<tr>
<td>μ = 1.0</td>
<td>Proportional-value Payment</td>
<td>Marginal misreporting opportunity, Weak punishment for misreporting</td>
</tr>
</tbody>
</table>
proved that both mechanisms are truthful in declarations of single-minded buyers, which is a challenging design task given the complexity of the considered problems. Although, allocative efficient (optimal) outcomes are not guaranteed by the designed approximation allocation mechanisms, the experimental results reveal near-optimal allocation quality.

We have also analysed the impact of the seller’s reservation prices on the market outcomes determined by the single-sided mechanism. The results show that the seller cannot improve her payment by adjusting the price reservations in the markets of scarce resource, while the reserve prices become price-deterministic if the services in the market are over-provisioned (supply exceeds demand). The seller’s strategic manipulation opportunity was also analysed in extensive simulation experiments. The obtained results show near-truthful performance of the proposed pricing mechanisms, where only the price overstating strategy may be slightly beneficial in some restricted market scenarios. We have studied the parametrisation of the proposed market mechanisms. Our results are summarized in a Table 4.10c as decision support for potential market operators.
Chapter 5

Market Mechanisms for
Heterogeneous Infrastructure Cloud Services

In this chapter, we investigate the market of Heterogeneous Infrastructure Cloud Services with a single infrastructure cloud service provider (Section 5.1) and a fully competitive double-sided marketplace with multiple buyers and sellers (Section 5.2). We address the considered problems by proposing two computational market mechanism designs and conducting a qualitative evaluation of their performance. Therefore, we (i) provide a detailed description of the designed mechanisms, (ii) illustrate their operation with simple examples, (iii) investigate theoretically such economic properties as Individual Rationality, Budget Balance, Computational Tractability and Buyer Truthfulness, and (iv) conduct extensive simulation experiments to analyse the quality of achieved allocations and to study the strategic manipulation opportunities of the market participants. Finally, we summarise the major design aspects of the proposed market mechanisms and their corresponding economic properties.
5.1 Single-sided Market Mechanism

In this section, we design a computational market mechanism for heterogeneous infrastructure cloud services trading in the market with a single seller and multiple competing buyers. Please, recall that heterogeneous infrastructure cloud service is a custom-defined virtual machine, where the underlying configuration in terms of CPU, RAM, and HDD is selected by the consumer, and it is traded in a form of combinatorial bundles.

5.1.1 Allocation Mechanism

The allocation mechanism proposed for a single-sided market of heterogeneous infrastructure cloud services is given in Algorithm 6. The allocation mechanism intends to determine the final allocation decision $X$ for all the participating buyer’s bids, given the vector of submitted buyer requests $\vec{\beta}$ and the seller’s ask $\vec{\alpha}$. The mechanism also returns the list $W$ that contains the determined weights of the buyer requested bundles of resources, which is used by the pricing mechanism in its operation.

Algorithm Description

The allocation mechanism has to identify a subset of bids that will receive the requested resources from the cloud provider. Given the heterogeneous nature of the traded goods, combinatorial buyer requests and a feasible time complexity requirement, the winner determination becomes a non-trivial problem. In order to address the problem, we apply greedy allocation principles, where the basic idea is to rank the bids based on some simple value, called a sorting criteria value.

In Chapter 4, we have identified the major issue of the conventional (single-shot) greedy allocation approach for allocation of combinatorial indivisible requests. The main problem is that such greedy mechanisms can exhaust one of the traded goods too quickly, which significantly diminishes the chance of making any further allocations, due to the resource complementarity in the bid. In order to address this issue, the market mechanism has to consider the scarcity of the available resources when making the allocation decision. Furthermore, the traditional greedy approaches determine the sorted order of the bids only once and consider the candidates for allocation in this order. It may not always be the best approach for allocating the resources in a marketplace, since the market conditions (e.g. resources available for allocation) may change significantly after a couple of candidates are allocated; which may result in an inefficient allocation. Therefore, we propose an iterative approach for winner determination, where the most socially efficient candidate is
Algorithm 6 Iterative Allocation Scheme for Single-sided Market of Heterogeneous Cloud Infrastructure Services (HE-SS-A)

1: Input: $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$, where $\hat{\beta}_b = (\hat{C}_b^d, \hat{p}_b^v)$
2: Input: $\hat{\alpha} = (\hat{C}_a^s, \hat{P}_s^a)$

# Step 1: Initialization Phase
3: $X \leftarrow \emptyset$  \hspace{1em} // final allocation decision vector
4: $L \leftarrow \emptyset$  \hspace{1em} // list of best efficiency bids in considered order

# Step 2: Iterative Allocation Phase
5: for $i = 1$ to $N$ do

  # Step 2a: Determine Best Efficiency Bid
6:   $b = \text{arg max}_{\sigma} e(\hat{\beta}_\sigma, \hat{\alpha}, X)$, where $\forall \hat{\beta}_\sigma \in \{\hat{\beta} \setminus L\}$
7:   $L \leftarrow L \cup \hat{\beta}_b$

  # Step 2b: Verify Problem Constraints
8:   if $\text{ARC}(\hat{\beta}_b, \hat{\alpha}, X) \& \text{RPC}(\hat{\beta}_b, \hat{\alpha})$ then
9:     $x_b = 1$
10:    else
11:     $x_b = 0$
12:  end if
13:  $X \leftarrow X \cup x_b$
14:  end for

15: Output: $X, L$

Determined and considered for allocation in each allocation iteration. In order to determine the market allocations, our proposed allocation mechanism runs in a number of steps:

- **Initialization Phase (lines 3, 4):** Firstly, the mechanism initialises the variable required for its operations, such as the final allocation decision variable $X$ and the variable $L$, which will contain the list of best efficiency bids in the considered order.

- **Iterative Allocation Phase (lines 5 - 14):** The allocation mechanism determines the market winners in $N$ iterations, where at each round only a single bid, called best efficiency candidate is selected and considered for allocation.

  - **Best Efficiency Candidate Determination (lines 6-7):** The mechanism uses the sorting criteria function $e(\hat{\beta}_b, \hat{\alpha}, X)$ in order to determine the sorting criteria values associated with all participating bids, except the previously selected best efficiency candidates in $L$. Afterwards, the allocation scheme selects the most desirable bid for allocation, i.e. the bid with the highest sorting criteria value, and adds it to the list $L$ (line 7). The sorting criteria function aims to associate such values with the allocation candidates that the more socially-efficient
bids are preferred by the mechanism. A more detailed explanation of the proposed sorting criteria function and its relationship to the problem’s objective is provided below.

– Problem Constraints Verification (lines 8-12): Finally, the selected best efficiency candidate $\hat{b}_b$ is considered for allocation. If all the problem constraints are satisfied, the bid will be allocated and the buyer will receive the requested bundle of resources $\hat{C}_b^d$; otherwise, the bid is denied by the mechanism and the buyer receives nothing. The proposed allocation scheme verifies the following two constraints:

* Available Resource Constraint (ARC) is checked to ensure that the amounts of available resources are sufficient to satisfy the buyer’s request. Formally, the ARC is defined as follows:

$$\hat{c}_b^d \leq \hat{c}_g^{sa}, \forall g \in G,$$

where

$$\hat{c}_g^{sa} = \hat{c}_g^s - \sum_{b \in B} \hat{c}_bg^d x_b$$

* Reserve Price Constraint (RPC) is verified in order to ensure that the buyer’s offered price valuation satisfies the seller’s desired bundle-specific price reservation. Formally, the RPC is defined as follows:

$$\hat{p}_b^{br} \geq \hat{p}_b^{hr},$$

where

$$\hat{p}_b^{hr} = \sum_{g \in G} \hat{c}_bg^d \hat{p}_g^{ur}$$

Sorting Criteria Function

The sorting criteria function $e(\hat{b}_b, \hat{\alpha}, X)$ is used to determine the sorting criteria values associated with each allocation candidate. These values serve as a selection criteria of the best efficiency candidates in various allocation iterations. In order for the allocation mechanism to determine the efficient allocation outcomes, it has to select the best efficiency candidates in such order that maximises the market social welfare, where the more socially-efficient bids are preferred. In other words, the sorting criteria function has to be tightly linked with the problem’s objective function. The single-sided market mechanism aims to maximise the sum of price valuations of all the allocated buyers, as defined in Formula 3.1. Therefore, a more socially efficient candidate would consume less resources, while offering a greater price valuation. Hence, the proposed sorting criteria function is defined as follows:

$$e(\hat{b}_b, \hat{\alpha}, X) = \frac{\hat{p}_b^{br}}{w(\hat{b}_b, \hat{\alpha}, X)}$$
This sorting criteria function is based on the declaration of the considered bid \( \hat{\beta}_b \), the reported seller’s ask \( \hat{\alpha} \) and the allocation decision variable \( X \), which contains the allocation decisions made until the current allocation iteration. The idea is to determine the price valuation contributed by a candidate per unit of traded good, defined by function \( w(\hat{\beta}_b, \hat{\alpha}, X) \) which is explained below. Unlike the homogeneous infrastructure cloud service, considered in Chapter 4, the relationship between the heterogeneous resources of different types is unknown, and a new way of measuring the resource relevance has to be defined. In the markets, which trade limited amounts of resources (especially with goods complementarity), resource scarcity is a good indicator for this purpose. We define the resource scarcity factor \( F_s \), as inverse of available capacities of resources, as follows:

\[
F_s = (f_1^s, \ldots, f_K^s), \quad \text{where} \quad f_g^s(\hat{c}_g^s, X) = (\hat{c}_g^s)^{-1} = (\hat{c}_g^s - \sum_{b \in B} c_{bg} x_b)^{-1}
\]

The weight of the requested bundle of resources is determined given the information about the traded resources \( C_b^d \) in relation to the amounts of available resources of corresponding types. In other words, the more scarce the available resource is in relation to the other resources, the more weight it is going to contribute to the requested bundle. Therefore, a candidate which requests scarcer resources will have to offer a very interesting price valuation in order to be competitive. Formally, the weight of the requested bundle is defined as follows:

\[
w(\hat{\beta}_b, \hat{\alpha}, X) = \left( \sum_{g \in G} \left( \frac{f_d^g(\hat{c}_g^s, X)}{\hat{c}_{bg}} \right)^l \right)^m
\]

The proposed sorting criteria function favours such allocation candidates that do not overconsume the scarce resources in the market, aiming to avoid the situation when the market runs out of one of its complementary resource types to quickly. For example, while CPU is the scarcest market resource, the mechanism would favour the buyer bids that request memory-intensive configurations, aiming to keep the consumption balance between the complimentary resource types.

**Parameters** \( l \geq 0 \) and \( m \geq 0 \)

We parametrise our proposed sorting criteria function in order to allow the control system over the market mechanism, depending on different allocation objectives. While adjusting the values of the two parameters \( l \) and \( m \) does not guarantee the improved social welfare or better resource utilisation, it gives flexibility for the market owner in adapting the mechanism to the specific market conditions and specific allocation requirements. We
provide an insight into the impact of introduced parameters on the sorting criteria values through illustrative examples in Table 5.1, where we plot the relative competitiveness of two bids, derived based on their sorting criteria values.

- **Parameter $l \geq 0$ (example 1):** The parameter $l \geq 0$ has a local impact on each of the resource types, and it is introduced in order to permit controlling the importance of the resource balance in the requested bundle. A well-balanced candidate would not request a lot of scarce resources, where the resource balance of the requested bundle is considered in regards to the amounts of resources available in the market, as follows: $c_{b_1}^d f_1^s = \cdots = c_{b_K}^d f_K^s$. We can observe that with increasing $l > 1$, the bid $\hat{\beta}_1$ is getting competitive advantage because the resources traded in $\hat{C}_1^d$ are better balanced in regards to the seller’s available resources, compared to the resource balance in $\hat{C}_2^d$.

- **Parameter $m \geq 0$ (example 2):** The parameter $m \geq 0$ has an impact on the entire calculated bundle weight, and it is used to allow controlling the significance of the traded resource size. In other words, adjusting this parameter indicates to the mechanism which kind of buyer bids to favour: the ones that request a larger or smaller bundle of services. In the considered example 2, we can observe the following: (i) when $m > 1$, the bid $\hat{\beta}_1$, which requests twice less resource than $\hat{\beta}_2$, gains competitive advantage (due to smaller traded bundle), while (ii) the parameter $m < 1$ makes the mechanism to favour the larger request $\hat{\beta}_2$ (bigger bundle).

- **Side-effect of parameter $l$ (example 3) and parameters interrelation $m = 1/l$ (example 4):** In the provided example, the allocation candidate $\hat{\beta}_1$ is better-balanced compared to $\hat{\beta}_2$. However, when the value of parameter $l$ increases, the bid $\hat{\beta}_2$ is favoured by the mechanism (example 3). Such undesired side-effect is due to the smaller requested bundle in the buyer bid $\hat{\beta}_2$. In order to address this issue, we can counter-balance the negative side-effect of parameter $l$ by adjusting the parameter $m$. Specifically, the following interrelation $m = 1/l$ allows to establish a more fair balance (example 4). Such parametrization would still emphasise the importance of the bundle balance according to the parameter $l$, but the sorting criteria result would not be affected by the size of the requested bundle.

As has been noted in Chapter 4, a non-linear parametrization of the sorting criteria function does not guarantee fairness in terms of merge-proofness. The buyers may split their bids into a number of smaller ones in order to gain the competitive advantage in
Table 5.1: Insight into parameters $l > 0$ and $m > 0$ (comparative sorting criteria value analysis)

<table>
<thead>
<tr>
<th>Example 1: Impact of Parameter $l$</th>
<th>Example 2: Impact of Parameter $m$</th>
<th>Example 3: Side-effect of $l$</th>
<th>Example 4: Interrelation $m = 1/l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\alpha} = \langle(20, 40, 1000), \langle0.05, 0.04, 0.001\rangle\rangle$</td>
<td>$\hat{\beta}_1 = \langle(2, 4, 100), 0.72\rangle$</td>
<td>$\hat{\beta}_1 = \langle(2, 4, 100), 0.72\rangle$</td>
<td>$\hat{\beta}_1 = \langle(20, 40, 1000), 7.2\rangle$</td>
</tr>
<tr>
<td>Bids</td>
<td>$\hat{\beta}_2 = \langle(4, 2, 100), 0.84\rangle$</td>
<td>$\hat{\beta}_2 = \langle(4, 8, 200), 1.44\rangle$</td>
<td>$\hat{\beta}_2 = \langle(10, 2, 50), 1.44\rangle$</td>
</tr>
<tr>
<td>Illustration</td>
<td><img src="image1" alt="Illustration" /></td>
<td><img src="image2" alt="Illustration" /></td>
<td><img src="image3" alt="Illustration" /></td>
</tr>
</tbody>
</table>

Illustration: Relative Competition

<table>
<thead>
<tr>
<th>$l$</th>
<th>$m$</th>
<th>Relative Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.00</td>
<td>25%</td>
</tr>
<tr>
<td>0.25</td>
<td>1.00</td>
<td>25%</td>
</tr>
<tr>
<td>0.50</td>
<td>1.00</td>
<td>25%</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
<td>25%</td>
</tr>
<tr>
<td>2.00</td>
<td>1.00</td>
<td>25%</td>
</tr>
<tr>
<td>4.00</td>
<td>1.00</td>
<td>25%</td>
</tr>
</tbody>
</table>
Table 5.2: HE-SS Mechanism Example: Market Input

(a) Seller’s Offer / Ask

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>CPU (cores)</th>
<th>RAM (GB)</th>
<th>HDD (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{C}_s )</td>
<td>10</td>
<td>11</td>
<td>1000</td>
</tr>
<tr>
<td>( \hat{P}_r )</td>
<td>$0.05</td>
<td>$0.04</td>
<td>$0.00025</td>
</tr>
</tbody>
</table>

(b) Buyers’ Requests / Bids

<table>
<thead>
<tr>
<th>Buyer</th>
<th>Requested Res., ( \hat{C}_d^b )</th>
<th>Val., ( \hat{p}_d^b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB1</td>
<td>2 4 100 $1.00</td>
<td></td>
</tr>
<tr>
<td>CB2</td>
<td>5 1 50 $1.30</td>
<td></td>
</tr>
<tr>
<td>CB3</td>
<td>4 4 150 $0.70</td>
<td></td>
</tr>
<tr>
<td>CB4</td>
<td>1 1 500 $1.10</td>
<td></td>
</tr>
<tr>
<td>CB5</td>
<td>2 8 300 $1.80</td>
<td></td>
</tr>
<tr>
<td>CB6</td>
<td>1 1 100 $0.10</td>
<td></td>
</tr>
</tbody>
</table>

the market. However, such behaviour has an associated risk of not obtaining the desired resources in full and may not be practical in the markets with resource complementarity.

Allocation Mechanism Example

We demonstrate the operational procedure of our proposed mechanism by giving a simple example. We assume that a cloud provider is willing to trade three different resource types: CPU, RAM, and HDD. The seller offers some amounts of CPU cores, GBs of RAM and HDD, and reports the minimum desired prices for a unit of each resource of different types, i.e. unit price reservations. The seller’s offer is depicted in Table 5.2a. A number of prospective cloud consumers join the market in order to procure the traded cloud infrastructure resources. In the considered example, there are six buyers requesting custom bundles of resources together with their price valuations for the demanded bundles. The cloud consumer requests, used in this example, are depicted in Table 5.2b.

The allocation procedure of our proposed single-sided mechanism for trading heterogeneous cloud services in the considered example is depicted in Table 5.3. The allocation mechanism collects the seller’s offer and the buyers’ requests, and aims to select the market winners who will obtain the traded cloud resources. For simplicity, we consider a linear form of the sorting criteria function, where \( l = 1.0, m = 1.0 \). The mechanism determines the winners in 6 iterations. In each iteration, the mechanism identifies a single best efficiency candidate, which is considered for allocation. If all the problem constraints are satisfied, the bid is granted the resources; otherwise, it is rejected.

In the first iteration, the sorting criteria values are calculated for all the participating
Table 5.3: HE-SS Mechanism Example: Allocation Procedure ($m = 1, l = 1$)

<table>
<thead>
<tr>
<th>$i$</th>
<th>Candidates, $e(\hat{\beta}<em>\sigma)$, where $\forall \hat{\beta}</em>\sigma \in {\hat{\beta} \setminus L}$</th>
<th>Best Candidate $\hat{\beta}_b$ in $L$</th>
<th>Constraints</th>
<th>$x_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^*$</td>
<td>CB1 1.51, CB2 2.03, CB3 0.77, CB4 1.59, CB5 1.47, CB6 0.34</td>
<td>CB2 1.51</td>
<td>$\hat{C}_b = (5, 1, 50)$, $\hat{C}_sa = (10, 11, 1000)$</td>
<td>✓ $1.3000$</td>
</tr>
<tr>
<td>2</td>
<td>1.10 - 0.52, CB4 1.33, 1.19, 0.25</td>
<td>CB4 1.33</td>
<td>$\hat{C}_b = (1, 1, 500)$, $\hat{C}_sa = (5, 10, 950)$</td>
<td>✓ $1.1000$</td>
</tr>
<tr>
<td>3</td>
<td>0.86 - 0.39, CB5 0.88, 0.17</td>
<td>CB5 0.88</td>
<td>$\hat{C}_b = (2, 8, 300)$, $\hat{C}_sa = (4, 9, 450)$</td>
<td>✓ $1.8000$</td>
</tr>
<tr>
<td>4</td>
<td>0.18 - 0.10, CB1 0.05</td>
<td>CB1 0.10</td>
<td>$\hat{C}_b = (2, 4, 100)$, $\hat{C}_sa = (2, 1, 150)$</td>
<td>✗ $1.0000$</td>
</tr>
<tr>
<td>5</td>
<td>- - 0.10, CB3 0.05</td>
<td>CB3 0.10</td>
<td>$\hat{C}_b = (4, 4, 150)$, $\hat{C}_sa = (2, 1, 150)$</td>
<td>✗ $0.7000$</td>
</tr>
<tr>
<td>6</td>
<td>- - - - 0.05, CB6 0.05</td>
<td>CB6 0.05</td>
<td>$\hat{C}_b = (1, 1, 100)$, $\hat{C}_sa = (2, 1, 150)$</td>
<td>✓ $0.1000$</td>
</tr>
</tbody>
</table>

* The order in which the allocation candidates are considered in a conventional (single-shot) approach is based on the sorting criteria, equal to the values determined in the first allocation iteration in our proposed mechanism, i.e. $L^{conv.} = \{CB2, CB4, CB1, CB5, CB3, CB6\}$
buyers. The buyers’ requested bundle weights are determined, given the current resource scarcity factor, i.e. $F_i^{s} = \langle 10^{-1}, 11^{-1}, 1000^{-1} \rangle = \langle 0.10, 0.09, 0.001 \rangle$ for the iteration $i = 1$. Please, note that the smaller the available capacity of a particular resource, the larger the scarcity factor will be; thus, the greater weight this resource type will contribute to the bundle. The sorting criteria values are further derived as a price valuation offered per bundle weight. For instance, the sorting criteria value of CB1 is calculated as follows:

$$e(CB1) = \frac{\$1.00}{((2 \times 0.1)^1 + (4 \times 0.09)^1 + (100 \times 0.001)^1)^1} \approx 1.51$$

The mechanism selects CB2 as the best efficiency bid in the allocation iteration $i = 1$ because it is the candidate with the highest sorting criteria value. The buyer bid CB2 becomes a market winner, since both constraints are satisfied. Once the allocation decision is made, the amounts of available resources change and the scarcity factor adjusts accordingly, $F_2^{s} = \langle 5^{-1}, 10^{-1}, 950^{-1} \rangle = \langle 0.2, 0.1, 0.0011 \rangle$. Therefore, in the following allocation iteration $i = 2$, the mechanism recalculates the sorting criteria values of the remaining allocation candidates (i.e. CB2 is not considered). Consequently, CB4 becomes the best efficiency candidate in the allocation iteration $i = 2$; the problem constraints are verified, and the bid becomes the winner. The process repeats until all the participating bids are considered and the corresponding allocation decisions are made.

As a result, the mechanism determines three winning bids (i.e. CB2, CB4, CB5), which are granted the requested resources of the cloud provider. The remaining three bids are denied by the mechanism either due to the insufficient amounts of available resources (i.e. CB1, CB3), or due to the low price valuation which does not satisfy the seller’s desired minimum price (i.e. CB6). The rejected buyers do not get any resource from the seller.

The mechanism allocated the following amounts of goods $(8, 10, 850)$; hence, the resource utilization in our example is $(80\%, 91\%, 85\%)$. The social welfare produced by the mechanism is $\mathcal{W} = \$1.3 + \$1.1 + \$1.8 = \$4.2$. By contrast, a conventional approach, where the sorted order is determined only once, would select the following market winners in the considered example: CB2, CB4, and CB1. The corresponding social welfare is $\mathcal{W}^{conv} = \$1.0 + \$1.3 + \$1.1 = \$3.3$ and the resource utilization is $(80\%, 55\%, 65\%)$. We can see that the iterative approach achieves a better resource utilization and social welfare due to the updated sorting criteria values in each allocation iteration. We study how the allocation quality of these two approaches compares in experimental section.
5.1.2 Pricing Mechanism

The proposed pricing scheme is given in Algorithm 7. The mechanism aims to determine the final buyer prices for the granted resources $P^b$, given the declarations of the participating buyers $\hat{\beta}$ and the seller’s offer $\hat{\alpha}$; as well as the previously determined allocation decision $X$ and the list of the best efficiency candidates in the considered order $L$.

Algorithm Description

The critical-value pricing schemes, complementary with the conventional single-sided greedy allocation mechanisms, derive the buyer prices based on the sorting criteria value of the first losing competitor (see Chapter 4). Such approach would not work when applied to our proposed iterative greedy allocation scheme due to the dynamic nature of the sorting criteria values that change in each iteration based on the availability of resources. In such an iterative allocation, a unique competition condition should be verified in various allocation iterations. Therefore, we propose a novel approach for critical-value payments determination, that is complementary with the iterative greedy allocation mechanism.

Our proposed buyer pricing scheme calculates the final buyer prices for all the participating buyer bids in order. The initially rejected bids are not granted any resource and receive zero payment (lines 7, 29). The final prices of the winning allocation candidates are determined based on the competition for the resources in the market (lines 9-27). Therefore, the mechanism searches for the corresponding competitors in various allocation iterations, and derives the minimum price required to outbid the weakest competitor. The following steps are performed for each winning buyer, aiming to determine the final prices:

- **Initialization Phase (lines 9-13):** Firstly, the mechanism initialises the variables required for its operation, including the variable $P^c$, which is used to keep the list of the minimum prices required to outbid the competitors, the auxiliary allocation decision variable $\bar{X}$, and the vector $\bar{L}$ used to keep the new list of the best efficiency candidates in the considered order. The pricing scheme, also constructs a new set of allocation candidates $\bar{L}^i$, which contains all the initially participating bids, except for the currently considered winner $\hat{\beta}^b$. Her bid is replaced by the corresponding bundle-reservation candidate $\hat{\beta}^br$. This candidate is the weakest possible competitor for the winning bid. It is used in order to establish the lower bound for the buyer’s price, i.e. bundle reservation price desired by the seller.

- **Iterative Competitors Determination Phase (lines 14-26):** Given the iterative allo-
Algorithm 7 Pricing Scheme for Single-sided Market of Heterogeneous Cloud Infrastructure Services (HE-SS-P)

1. **Input:** $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N)$, where $\hat{\beta}_b = \langle \hat{C}_b^d, \hat{p}_b^c \rangle$
2. **Input:** $\hat{\alpha} = \langle \hat{C}_b^a, \hat{P}_b^c \rangle$
3. **Input:** $X = \{x_b\}$, $\forall b \in B$
4. **Input:** $L = \{\hat{\beta}_b\}$ // list of best efficiency bids in considered order
5. $P^b \leftarrow \emptyset$
6. for all $\hat{\beta}_b \in \beta$ do
7. $\rho^b_0 = 0$
8. if $x_b = 1$ then

   # Step 1: Initialization Phase
9. $P^c \leftarrow \emptyset$ // list of minimum competitive prices
10. $X \leftarrow \emptyset$ // new allocation results
11. $\hat{L} \leftarrow \emptyset$ // best efficiency bids in considered order
12. $\hat{L}^i \leftarrow \{\hat{\beta}_\sigma \mid \sigma \neq b \land \forall \hat{\beta}_\sigma \in \hat{\beta}\}$ // all bids except $b$
13. $L^i \leftarrow L^i \cup \{\hat{\beta}_b^x\}$ // bundle reservation for $b$

   # Step 2: Iterative Competitors Determination Phase
14. for $i = 1$ to $N$ do
15.   $\hat{\beta} = \arg \max_{\sigma} e(\hat{\beta}_\sigma, \hat{\alpha}, \hat{X})$, where $\forall \hat{\beta}_\sigma \in \{L^i \setminus \hat{L}\}$
16.   $\hat{L} \leftarrow \hat{L} \cup \hat{\beta}_b^x$
17.   if $\left(\text{ARC}(\hat{\beta}_b, \hat{\alpha}, \hat{X}) \& \text{RPC}(\hat{\beta}_b, \hat{\alpha})\right)$ then
18.     if $\left(\text{ARC}(\hat{\beta}_b, \hat{\alpha}, \hat{X}) \& \text{RPC}(\hat{\beta}_b, \hat{\alpha})\right)$ then
19.       if $\left(\bigcup_{\tau = 1}^i L_\tau \neq \bigcup_{\tau = 1}^i \hat{L}_\tau \lor (\hat{b} = b)\right)$ then
20.         $P^c \leftarrow P^c \cup \{e(\hat{\beta}_b, \hat{\alpha}, \hat{X}) \times w(\hat{\beta}_b, \hat{\alpha}, \hat{X})\}$
21.       end if
22.     end if
23.     $\hat{x}_b = 1$
24.   end if
25.   $X \leftarrow X \cup \{\hat{x}_b\}$
26. end if
27. end for

   # Step 3: Price Determination Phase
28. $\rho^b_0 = \min(P^c)$
29. end if
30. $P^b \leftarrow P^b \cup \rho^b_0$
31. end for

Output: $P^b = \{\rho^b_1, \ldots, \rho^b_N\}$
cation mechanism, the competitors of the winning buyers have to be determined in a similar iterative fashion. Therefore, the mechanism verifies the competitors in \( N \) allocation iterations, where the candidates are considered one by one in each round:

- **Best Efficiency Bid Determination (lines 15-16):** Given the currently available capacities of resources of different types and the corresponding scarcity factors, the mechanism calculates the sorting criteria values for all not-considered candidates \( \{\bar{L}^i \setminus \bar{L}\} \). The candidate with the highest sorting criteria value becomes the best efficiency bid \( \bar{b}_{\text{b}} \) which is saved in the list \( \bar{L} \). This candidate will be verified to be the competitor of the current winning bid.

- **Competitors Determination (lines 17-25):** The mechanism verifies a number of conditions in order for the two candidates - the selected best efficiency bid \( \bar{b}_{\text{b}} \) and the currently considered winning bid \( \bar{b}_{\text{w}} \), to be the competitors.

  * **Allocation feasibility condition (lines 17-18):** For the two candidates to be considered as competitors, the allocation in the considered iteration has to be feasible for both bids. Therefore, both of them have to satisfy the problem constraints, including ARC and RPC.

  * **Unique competition condition (line 19):** The pricing scheme aims to find the particular cases, when the competition in the market is unique, compared to the competition in the initial allocation process. There are two possible conditions for the competition to be unique: (i) either the competitor is the bundle-reservation bid \( \bar{b}_{\text{br}} \) (which was not present in the initial allocation), or (ii) the resource scarcity factors in the considered allocation iteration \( 1 \leq i \leq N \) are different from the ones in the initial allocation procedure. In the latter situation, the lists of the best efficiency candidates have to contain different bids as follows:

\[
\bigcup_{\tau=1}^{i} \bar{L}_{\tau} \neq \bigcup_{\tau=1}^{i} \bar{L}_{\tau}
\]

For each determined competitor candidate of the currently winning bid, the mechanism determines the minimum price required to outbid the competitor, which is kept in the list of competitor prices \( \bar{P}^{c} \) (line 20). Please, note that the auxiliary allocation decision \( \bar{X} \) is updated according to the new ongoing allocation of the candidate in the list \( \bar{L}^i \) (lines 23, 24) in order to keep track of resource scarcity.
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

- **Price Determination Phase (line 27):** Finally, when all the bids in the list $\hat{L}^i$ are considered and the minimum prices to outbid the determined competitors are calculated, the mechanism derives the final buyer price. This price is simply the smallest price among all the competitor prices in the list $\hat{P}^c$. In other words, the mechanism selects the smallest price required to outbid at least one of the determined competitors.

**Pricing Mechanism Example**

We continue the initial example, which was used to illustrate the procedure of the proposed allocation mechanism. When the market winners are identified, the pricing scheme aims to determine the final prices that the buyers pay. The pricing procedure of our proposed single-sided mechanism for trading heterogeneous cloud services is depicted in Table 5.4.

The final prices are calculated for all six participating buyer bids in order. The denied candidates, such as CB1, CB3, and CB6, are not required to pay, since they do not receive their requested bundle of resources. The allocated buyers, i.e. CB2, CB4, and CB5, will have to pay the prices determined based on the competition for the resources in the market. Therefore, the mechanism runs an iterative competitors determination procedure with the new set of candidates, which contains all the participating bids, except for the currently considered winner, who is replaced by her bundle reservation candidate.

For example, for CB2, the mechanism would consider the following new set of bids: $\langle CB1, CB2BR, CB3, CB4, CB5, CB6 \rangle$. These candidates are considered in six allocation iterations, where the best efficiency bid is determined in each. In the first allocation iteration, the bid CB4 has the highest sorting criteria value and becomes the best efficiency bid. Hence, the mechanism verifies whether both CB4 and the currently considered winning bid CB2 are the competitors. We can see that both of them satisfy the problem constraints and could be allocated in this allocation iteration $\bar{x}_b = \bar{x}_b = 1$. Furthermore, the unique competition condition is valid, since the two lists of the best efficiency bids are not the same, i.e. $L_1 = \langle CB2 \rangle \neq (L_1 = \langle CB4 \rangle)$. Therefore, the bids are competitors and the minimum price required for the buyer CB2 to outbid her unique competitor CB4 is calculated given the current scarcity factor $F^*_1 = \langle 0.10, 0.09, 0.001 \rangle$ as follows:

$$e(\hat{\beta}_b, \hat{\alpha}, X) \times w(\hat{\beta}_b, \hat{\alpha}, X) = \frac{\$1.1000}{(1 \times 0.1)^1 + (1 \times 0.09)^1 + (500 \times 0.001)^1} \times (5 \times 0.1)^1 + (1 \times 0.09)^1 + (50 \times 0.001)^1}^1 = \$1.0204$$

The process repeats until all the allocation candidates are considered. We can see that
### Table 5.4: HE-SS Mechanism Example: Pricing Procedure ($m = 1, l = 1$)

<table>
<thead>
<tr>
<th>Buyer, $x_b$</th>
<th>Iterative Competitors Determination</th>
<th>Final Price $\rho_b^c$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best Efficiency Bids in $\bar{L}$</td>
<td>Conditions</td>
</tr>
<tr>
<td></td>
<td>$i$</td>
<td>$\hat{\beta}_b$</td>
</tr>
<tr>
<td>CB1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>CB2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>CB3</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>CB4</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>CB5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>CB6</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

* The candidate is not a competitor due to the violated unique competition condition, which means that the initial list of candidates $L_i$ contained the same best efficiency bids as the list $\bar{L}_i$ at the allocation iteration $i > 0$.

The best efficiency candidates in the initially considered order in the list: $L = (CB2, CB4, CB5, CB1, CB3, CB6)$
the mechanism has determined three unique competitors, i.e. CB4, CB1 and CB3 for the winning buyer CB2. The corresponding minimum prices in order for CB2 to outbid them were determined, i.e. $1.0204, $0.9189, and $0.4365, respectively. The smallest competitive price, i.e. $0.4365 is selected as the final price of CB2.

The mechanism determines the final prices for the bids CB4 and CB5 in a similar fashion. Please, note that the final price of the winning buyer CB4 was derived based on her bundle-reservation bid CB4BR. It means that the bid could have won even with the price reservation value. We would like to draw your attention to the cases noted by asterisk sign. For example, in the allocation iteration $i = 2$ for the winning bid CB5, the lists of considered best efficiency candidates were as follows: $L_2 = \{CB2, CB4\} = \bar{L}_2$. Even though, both the best efficiency candidate CB4 and the currently considered winning buyer CB5 satisfy the problem constraints, they are not considered as competitors because the unique competition condition is violated. It is used to avoid the buyer prices that exceed the winning buyer’s price valuation (buyer Individual Rationality), which we prove theoretically in the next section.

5.1.3 Theoretical Investigation

In this section, the economic properties of the proposed HE-SS market mechanism are analysed theoretically. In particular, we investigate the following properties: Individual Rationality (IR), Budget Balance (BB), Computational Tractability (CT) and Truthfulness (T). The mathematical proofs provided in this sections are based on the definitions in Chapter 3.

Individual Rationality (IR)

Theorem 10. The proposed pricing scheme maintains Individual Rationality property for single-minded buyers declared types.

Proof. We need to show that IR property holds as defined in Formula 3.4. Since our proposed pricing scheme does not charge a losing buyers, we focus on the allocated buyer pricing. We assume that second buyer IR condition does not hold as follows $\hat{p}_h > \hat{p}_c$ and provide the proof by contradiction. In such situation, we get the following:

\[ e(\hat{\beta'}, \hat{\alpha}, \hat{X}) \times w(\hat{\beta}_h, \hat{\beta}, \hat{X}) < \hat{p}_c \Rightarrow e(\hat{\beta'}, \hat{\alpha}, \hat{X}) < e(\hat{\beta}_h, \hat{\alpha}, \hat{X}) \]

It means that the currently considered winning bid $\hat{\beta}_h$ is less competitive than her weakest
competitor bid $\tilde{\beta}^c$. Therefore, the sorted lists $L$ and $\tilde{L}$ would be identical at least until the position $z \leq N$ of the weakest competitor $\tilde{\beta}^c$ as follows:

$$\bigcup_{\tau=1}^{z} L_{\tau} = \bigcup_{\tau=1}^{z} \tilde{L}_{\tau}$$

Therefore, in order to satisfy the unique competitor condition, the weakest competitor must have been a bundle reservation bid $\beta^{br}_b$. In such situation, we get:

$$e(\beta_b, \tilde{\alpha}, X) \leq \beta^{br}_b, \tilde{\alpha}, X \quad \Rightarrow \quad \frac{\tilde{p}^{br}_b}{w(\beta_b, \tilde{\alpha}, X)} > \frac{\tilde{p}^v_b}{w(\beta_b, \tilde{\alpha}, X)} \quad \Rightarrow \quad \beta^{br}_b > \tilde{p}^v_b$$

Such a bid could not have been initially allocated due to violated RPC. Consequently, the final buyer’s price is always lower than her reported price valuation: $\rho^b_b \leq \tilde{p}^v_b$.

Hence, the proposed pricing mechanism is IR for single-minded buyers.

\[\square\]

**Theorem 11.** The proposed pricing scheme maintains Individual Rationality property for the seller’s declared type.

**Proof.** We need to show that IR property holds as defined in Formula 3.5. We will show that IR property holds for the seller by proving that the final price of an allocated buyer cannot be lower than the seller’s bundle price reservation: $\rho^b_b \geq \tilde{p}^{br}_b, x_b, \forall b \in B$. We assume that it is false (i.e. $\rho^b_b < \tilde{p}^{br}_b$, where $x_b = 1$) and provide proof by contradiction. In such situation, we get:

$$e(\tilde{\beta}^c, \tilde{\alpha}, X) \times w(\beta_b, \tilde{\alpha}, X) < \tilde{p}^{br}_b \quad \Rightarrow \quad e(\tilde{\beta}^c, \tilde{\alpha}, X) < e(\beta^{br}_b, \tilde{\alpha}, X)$$

It means that the weakest competitor $\tilde{\beta}^c$ was less competitive that the bundle reservation bid $\beta^{br}_b$. As a result, at the allocation iteration $z \leq N$ when the bid $\beta^c$ was considered, the lists of the best efficiency bids $L$ and $\tilde{L}$ would contain the same candidates:

$$\bigcup_{\tau=1}^{z} L_{i} = \bigcup_{\tau=1}^{z} \tilde{L}_{\tau}$$

Since, it violates one of the unique competitor conditions, such bid could not have become the competitor. Hence, the lowest possible final price of a winning bid can be derived based on her bundle reservation bid: $\rho^b_b \geq \tilde{p}^{br}_b$. Consequently, given the way the seller’s payment
is derived, we can conclude:

\[
\begin{align*}
\rho_b^b &\geq \beta_b^b x_b, \forall b \in B \\
\rho^s &= \sum_{b \in B} \rho_b^b \\
\Rightarrow \rho^s &\geq \sum_{b \in B} \rho_b^b x_b
\end{align*}
\]

Hence, the proposed pricing mechanism is IR for the seller’s declared type.

**Budget Balance (BB)**

In a proposed single-sided market, the payments collected from the buyers are paid entirely to a single market seller, which is formally defined as follows:

\[
\rho^s = \sum_{b \in B} \rho_b^b
\]

Such a payment complies with the budget balance definition provided in Formula 3.6, which means that the HE-SS market mechanism maintains Budget Balance property.

**Computational Tractability (CT)**

The designed HE-SS market mechanism for trading heterogeneous infrastructure cloud services has a polynomial time complexity. The proposed iterative HE-SS-A mechanism determines the market winners in \(O(|N^2|)\) time, where the mechanism selects \(N\) best efficiency bids in \(N\) allocation iterations. The corresponding HE-SS-P scheme determines the buyer payments in \(O(|N^3|)\) time where the major computation is for iterative competitors determination process. Therefore, the market mechanism’s time complexity is \(O(|N^3|)\).

Please note that the computation time can be improved by reducing the number of allocation iterations in which the best efficiency bids are determined from \(N\) to \(N/\mathcal{T}\), where \(1 \leq \mathcal{T} \leq N\); hence the corresponding time complexity will be reduced to \(O(|N^3/\mathcal{T}|)\).

**Truthfulness (T)**

The truthfulness of the proposed HE-SS market mechanism is demonstrated by proving the sufficient truthfulness conditions, as outlined in [90]. Specifically, we provide two theorems that demonstrate that the proposed allocation function is monotone in buyers’ declarations, and the pricing mechanism derives buyer payments based on critical-value.

**Theorem 12.** The proposed allocation scheme is monotone in declarations of single-minded buyers.
Proof. The proposed proof is derived based on Definition 3.4.1. We consider two reported bids of the same buyer \( \hat{\beta}_{b^*} = \langle \hat{C}_{b^*}, \hat{p}_{b^*} \rangle \), and \( \hat{\beta}'_{b^*} = \langle \hat{C}'_{b^*}, \hat{p}'_{b^*} \rangle \), s.t. \( \hat{\beta}_{b^*} \succeq \hat{\beta}'_{b^*} \). We assume that the monotonicity property does not hold and provide the proof by contradiction.

Scenario 1: We consider the situation when \( \hat{\beta}'_{b^*} \) was allocated \( x_{b^*} = 1 \), but her changed type \( \hat{\beta}_{b^*} \) had lost \( x_{b^*} = 0 \). Hence, the bid \( \hat{\beta}_{b^*} \) must have violated at least one of the two problem constraints:

- **RPC:** Since \( \hat{\beta}'_{b^*} \) was allocated \( x_{b^*} = 1 \), it has satisfied PRC, i.e. \( \hat{p}'_{b^*} \geq \hat{p}_{b^*}^{br} \). Given that \( \hat{c}'_{b^*g} \leq \hat{c}_{b^*g} \), \( \forall g \in G \), we get that \( \hat{p}'_{b^*} \geq \hat{p}_{b^*}^{br} \). Thus, we can conclude the following:

\[
\begin{align*}
\hat{p}_b^c & \geq \hat{p}_{b^*}^{br} \\
\hat{p}_b^c & \leq \hat{p}_{b^*} \\
\hat{p}_{b^*}^{br} & \geq \hat{p}_{b^*}^{br}
\end{align*}
\]

Hence, \( \hat{\beta}_{b^*} \) would also satisfy RPC.

- **ARC:** Since \( \hat{\beta}_{b^*} \) was allocated \( x_{b^*} = 1 \) at some allocation iteration \( z < N \), there was enough available resource for her to satisfy ARC. In each new allocation iteration, the amounts of available resources can only decrease due to the ongoing new allocations. Therefore, given that \( \hat{c}'_{b^*g} \leq \hat{c}_{b^*g} \), \( \forall g \in G \), the bid \( \hat{\beta}_{b^*} \) must have been considered in some allocation iteration \( \hat{z} > z \) in order to be denied \( x_{b^*} = 0 \). Since \( \hat{\beta}_{b^*} \succeq \hat{\beta}'_{b^*} \), we get the following:

\[
\begin{align*}
\hat{p}_b^c \geq \hat{p}_b^c \\
\left( \sum_{g \in G} (\hat{c}_{b^*g}^{d} f^*(\hat{c}_g^*, X_z)) \right)^m & \leq \left( \sum_{g \in G} (\hat{c}_{b^*g}^{d} f^*(\hat{c}_g^*, X_z)) \right)^m \\
\Rightarrow e(\hat{\beta}_{b^*}, \hat{\alpha}, X_z) & \geq e(\hat{\beta}'_{b^*}, \hat{\alpha}, X_z)
\end{align*}
\]

It means that \( \hat{\beta}_{b^*} \) would be the best efficiency bid in some allocation iteration \( \hat{z} \leq z \), and it would also satisfy ARC. We have shown that if a bid \( \hat{\beta}_{b^*} \) is allocated \( x_{b^*} = 1 \), then the modified bid \( \hat{\beta}_{b^*} \), s.t. \( \hat{\beta}_{b^*} \succeq \hat{\beta}_{b^*} \) would also satisfy all the problem constraints and would also be allocated \( x_{b^*} = 1 \).

Scenario 2: We consider the opposite situation, when a bid \( \hat{\beta}_{b^*} \) was denied \( x_{b^*} = 0 \), and the modified bid \( \hat{\beta}'_{b^*} \) was allocated \( x_{b^*} = 1 \). Similarly, if \( \hat{\beta}_{b^*} \) was rejected due to the violated RPC, then the bid \( \hat{\beta}_{b^*} \) would violate it as well, since \( \hat{p}_{b^*}^{br} \geq \hat{p}_{b^*} \), and \( \hat{c}_{b^*g}^{d} \leq \hat{c}_{b^*g}^{d} \), \( \forall g \in G \). In case, there was not enough available resources for \( \hat{\beta}_{b^*} \) at some allocation iteration \( z \leq N \) (violated ARC), then the modified bid \( \hat{\beta}'_{b^*} \) must be considered at some
allocation iteration $\bar{z} < z$ in order to be allocated $x_{br} = 1$. Since $\hat{\beta}_{br} \geq \beta_{br}$, we get that $e(\hat{\beta}_{br}, \hat{\alpha}, X_z) \geq e(\beta_{br}, \hat{\alpha}, X_z)$ meaning that $\hat{\beta}_{br}$ would have been considered at some allocation iteration $\bar{z} \geq z$ and would also violate ARC. Therefore, if a bid $\hat{\beta}_{br}$ was denied $x_{br} = 0$, then the modified bid $\hat{\beta}_{br}'$, s.t. $\hat{\beta}_{br}' \geq \hat{\beta}_{br}$ would also be denied $x_{br} = 0$.

Hence, the HE-DS-A scheme is monotone in single-minded buyer declarations.

\[ \Box \]

**Theorem 13.** The proposed pricing scheme determines critical-value payments for single-minded buyers.

**Proof.** The provided proof is derived based on Definition 3.4.2. We assume that the required critical-value conditions do not hold and provide the proof by contradiction.

**Scenario 1:** We consider a situation when a buyer wins $x_b = 1$ with the price valuation $\hat{p}_b^v < \rho_b^v$. Given that $\rho_b^v$ is the minimum price valuation required in order to outbid the weakest competitor (is referred to as $\hat{\beta}^c$) in some allocation iteration $z \leq N$, we get:

\[
\hat{p}_b^v < e(\beta^c, \hat{\alpha}, \bar{X}_z) \times w(\hat{\beta}_b, \hat{\alpha}, \bar{X}_z) \quad \Rightarrow \quad e(\hat{\beta}_b, \hat{\alpha}, \bar{X}_z) < e(\beta^c, \hat{\alpha}, \bar{X}_z)
\]

It means that the bid $\hat{\beta}_b$ is less competitive than $\hat{\beta}^c$, and it must have become the best efficiency bid at some allocation iteration $\bar{z} > z$. In other words, the lists of the best efficiency bids $L$ and $\bar{L}$ would have contained the same allocation candidates at least until the allocation iteration $z \leq N$:

\[
\bigcup_{\tau=1}^{\bar{z}} L_{\tau} = \bigcup_{\tau=1}^{z} L_{\tau}
\]

Thus, such bid must have been a bundle reservation bid $\beta_{br}^b$ in order to satisfy the unique competition conditions. In this case, given that $\hat{p}_b^v < \rho_b^v$ we get the following:

\[
\hat{p}_b^v < e(\beta_{br}^b, \hat{\alpha}, \bar{X}_z) \times w(\hat{\beta}_b, \hat{\alpha}, \bar{X}_z) \quad \Rightarrow \quad \hat{p}_b^v < \beta_{br}^b
\]

It means that the bid $\hat{\beta}_b$ would have violated RPC and would not win $x_b \neq 1$. Therefore, any price valuation $\hat{p}_b^v < \rho_b^v$ would result in the bid $\hat{\beta}_b$ being denied $x_b = 0$.

**Scenario 2:** We consider the second case and analyse the situation when a bid is denied $x_b = 0$ with the price valuation $\hat{p}_b^v \geq \rho_b^v$. We know that the weakest competitor $\hat{\beta}^c$ has satisfied the problem constraints. Similarly, we get that any $\hat{p}_b^v \geq \rho_b^v$ would result in the bid being more competitive $e(\hat{\beta}_b, \hat{\alpha}, \bar{X}_z) < e(\beta^c, \hat{\alpha}, \bar{X}_z)$ at some allocation iteration $z \leq N$. As a result, the bid $\hat{\beta}_b$ would become the best efficiency bid in allocation iteration $\bar{z} \leq z$. Therefore, the bid would satisfy the problem constraints and would be allocated.
Hence, any price valuation $\hat{p}_b^k \geq \hat{\beta}_b^k$ would result in the bid $\hat{\beta}_b$ being allocated $x_b = 1$.

Hence, the HE-SS-P mechanism determines buyer critical-value payments.

5.1.4 Experimental Investigation

In this section, we conduct qualitative analysis and study dynamics of our proposed mechanism in extensive simulation experiments. In particular, we analyse the allocative quality of our iterative/adaptive greedy approximation allocation scheme and study the seller’s market power over the pricing outcomes.

Experimental Setting

The experimental setup, used in our extensive simulation, is outlined in Table 5.5. In order to analyse the dynamics of our proposed mechanism, we simulate a wide range of market scenarios as discussed below.

**Market Model:** A single-sided market with a single service provider is considered. There are three types of infrastructure cloud resources $K = 3$ traded in the market, namely CPU, RAM, and HDD.

**Demand Model:** In this investigation, we experiment with varying numbers of buyers $N$ participating in the market. The buyer bids are randomly generated based on uniform distribution functions within specific ranges. The minimum and maximum values of the distribution functions, used for requested bundle generation, are derived based on real-world cloud provider offerings, such as Amazon EC2 [6] and Rackspace [116]. The bid valuations are generated for each resource type individually based on random values. These values are scaled according to the amounts of requested resources in order to determine the valuation for the requested bundle $p_v^b$. The maximum limits in the distribution function are selected based on the prices from such cloud providers as AT&T Cloud [3] and Verizo/Terremark Cloud [143].

**Supply Model:** There is a single resource provider $M = 1$, offering services to the consumers. We experiment with various levels of resource provisioning by adjusting the supply in regards to the previously generated total market demand. It is achieved by the provisioning factor $p_g$, $\forall g \in G$. We consider four possible provisioning levels for the resources of each type, which totals to $4^3 = 64$ combinations. In this experiment, the seller’s
### Market Model:

**Resource Types**

\[ G = \{ 1 : \text{CPU}, 2 : \text{RAM}, 3 : \text{HDD} \} \]

\[ K = 3 \]

### Demand Model:

**Number of Buyers**

\[ N = \{ 32, 64, 128, 256, 512 \} \]

\[ c_{b1}^{d} = U(1, 32) \]

**Demanded Capacity**

\[ c_{b2}^{d} = U(1, 240) \]

\[ c_{b3}^{d} = U(50, 1200) \]

**Price Valuation**

\[ p_{b}^{v} = \sum_{g \in G} p_{bg}^{v}, \text{where} \]

\[ p_{b1}^{v} = U(0.0, 0.1) \]

\[ p_{b2}^{v} = U(0.0, 0.08) \]

\[ p_{b3}^{v} = U(0.0, 0.0005) \]

### Supply Model:

**Number of Sellers**

\[ M = \{ 1 \} \]

\[ c_{sg}^{s} = \left( \sum_{b \in B} c_{bg}^{d} \right) \times p_{g}, \forall g \in G \]

**Supplied Capacity**

\[ p_{g} = \{ 0.25, 0.5, 0.75, 1.0 \} \]

**Price Reservation**

\[ p_{s1}^{r} = U(0.0, 0.1) \times z \]

\[ p_{s2}^{r} = U(0.0, 0.08) \times z \]

\[ p_{s3}^{r} = U(0.0, 0.0005) \times z \]

\[ z = \{ 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5 \} \]

### Metrics and Mechanisms:

**Metrics**

Social Welfare, Resource Utilisation, Total Cost, Seller Revenue, Computation Time, Buyer Fulfilment

**Allocation Mechanisms**

HE-SS Iterative, where \( m, l = \{ 0, 0.25, 0.5, 1, 2, 4 \} \)

HE-SS Static [101]

Optimal CPLEX solver

**Pricing Mechanisms**

HE-SS Iterative Critical-value

HE-SS Static Critical-value [101]
price reservations for the traded resources are drawn from uniform distribution function in the same ranges as the buyers valuations. We simulate various price reservation scenarios by scaling the random price reservation values according to the price comparative ratio $z$.

Overall, we simulate 448000 market inputs, where 2240 market scenarios are repeated 200 times each. We experiment with 36 different parametrisations of our proposed mechanism (parameters $l$ and $m$), and compare the results with the static allocation mechanism from literature [101] and with the optimal result. For each market scenario, we calculate some basic statistics across the conducted 200 trials in order to avoid statistical noise.

Allocation Mechanism Experiment

In this experiment, we study the quality of allocation results achieved by our proposed mechanism, determine the most efficient parametrisation based on allocation objective and compare the allocative performance with a static mechanism from literature. In particular, we measure the following metrics: social welfare, resource utilisation, buyer fulfilment and computation time.

Closeness to Optimality

Figure 5.1 depicts the closeness to optimality of the results obtained by our approximation allocation mechanism in terms of social welfare and average resource utilisation. Each graph shows the optimality dynamics with varying values of mechanism parameters $l$ and $m$. We pick a representative market scenario where $N = 512$, $z = 0.0$, $p = 0.25$ and change one of the variables at a time in order to analyse the impact.

In terms of social welfare, we can observe a great allocation quality achieved by the adaptive approximation mechanism (Figure 5.1 left side). A similar pattern emerges across different market scenarios, which suggests that the best parametrisation in order to attain the most socially-efficient allocation is when $(l = 2.0, m = 0.5)$ and $(l = 4.0, m = 0.25)$. It complies to our theoretical predictions, where the bundle balance is emphasised by greater values of parameter $l$ and the negative side-effect of this parameter is counter balanced by adjusting the value of parameter $m = 1/l$. Another important observation is that the approximation quality drops significantly when both parameters are set to high values, e.g. $(l = 4.0, m = 4.0)$. It happens because a greater importance is given to the bundle of requested resource, while a smaller regard is given to the actual price valuation. However, such a parametrisation has a positive impact on the resource utilisation as depicted in the right side of Figure 5.1. Please, note that the socially-efficient parametrisation also achieves a near-optimal performance in terms of resource utilisation, which proves that the
approximation mechanism behaviour is very similar to the optimal solver.

The market instances get easier for approximation mechanism with increasing provisioning level (Figures 5.1a and 5.1b) and higher price comparative ratio (Figures 5.1i and 5.1j). It happens because the chance to select a bid which is a part of optimal allocation is higher. In the former case, there is less bids to be denied by the mechanism due to a higher resource availability; in the latter case, there is less number of bids feasible for allocation due to violated RPC. The number of buyers in the market have an insignificant impact on the approximation allocation quality (Figures 5.1e and 5.1f), where the results for the market instances with a small number of bids are typically slightly less efficient.

Our experimental outcomes reveal that the average resource utilisation achieved by our proposed mechanism is close to the optimal allocation result even when the resources of different types are provisioned in an unbalanced proportion. Such an outstanding result
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

Comparing Static and Adaptive Approximation Approaches

In Figure 5.2, we depict the social welfare improvement, achieved by our proposed adaptive greedy approximation mechanism (most efficient parametrisation $\langle l = 4.0, m = 0.25 \rangle$) over the static approach. It can be clearly seen that our proposed mechanism has a significant advantage over the static one, where the social welfare improvement reaches up to 11% on average in certain market scenarios. As was discussed earlier, the hardest market cases are the ones with a low level of provisioning and small price comparative ratio. In Figure 5.2a, we can see that the best social welfare improvement over the static method is achieved when $p = 0.25$ and $z = 0.0$ due to the mechanism’s ability to make the allocation decisions based on changing scarcities of resources. In Figure 5.2b, we can see that when the number of bids is small $N = 64$, the improvement is smaller. It happens due to the reduced number of iterations of our adaptive approach which allows less time for our mechanism to achieve a better result.

Figure 5.3 provides a snapshot of resource utilisation, achieved by different mechanisms across all market scenarios (mean values). The goal is to provide a global view of how the resource utilisation of CPU, RAM and HDD compares across the market mechanisms. First of all, we can notice an impressively well-balanced resource utilisation by our proposed adaptive mechanism’s parametrisation $\langle l = 4.0, m = 4.0 \rangle$. Due to a high importance given to the requested bundles, such parametrisation is able to achieve 67% utilisation on average for each type of traded resource. The performance of the static mechanism is comparable to the $\langle l = 0, m = 0 \rangle$ parametrisation of our proposed mechanism, which relies
Figure 5.3: Resource Utilisation of various Allocation Mechanisms achieved across all market scenarios.

Figure 5.4: Computation Time for different allocation mechanisms.

on bid valuations only for ranking. The resource utilisation and the balance across the resources of different type are very similar in optimal mechanism and our mechanism with parameters \( l = 1, m = 1 \) and \( l = 4.0, m = 0.25 \). The latter parametrisation has an identical average performance to the optimal, while the linear parametrisation is slightly worse due to the lower importance given to the bundle-balance.

**Computation Time**

The computation time required to determine the resource allocation outcomes is depicted in Figure 5.4. We plot the results of the static approach, our proposed mechanism and the optimal solver. Due to the iterative nature of our proposed allocation approach, the time required in order to determine the allocation outcomes is higher than in a static mechanism. The optimal solver has impressively time-efficient performance as well; however, when the problem size becomes bigger, the mechanism’s time to derive the solution increases exponentially. Please, note that the time complexity of our proposed adaptive allocation approach can be improved by reordering the sorted queue of bids not after each new allocation iteration, but after several allocations were made. Thus, a compromise can be made between the enhanced allocation quality and an improved computation time.

**Pricing Mechanism Experiment**

In this experiment, the seller’s power over the determined market outcomes, including allocation and pricing decisions, is investigated. We measure (i) the allocation-related metrics, such as social welfare, resource utilisation and buyer fulfilment, as well as (ii) the pricing outcomes, such as the seller’s revenue, total reserve price and the total buyer discount. We compare the results of our proposed mechanism with a static mechanism.
that was extended to consider the seller’s reserve price.

**Seller Power over Market Pricing Outcomes**

The social welfare consists of buyer price valuations for the resources obtained after allocation. The pricing mechanism shares the market’s social welfare between the seller and the buyers by determining the payment amount and the granted discount. The seller’s payment consists of the minimum price reservation, required by the seller, and the profit, determined by the mechanism. We analyse how the social welfare is distributed in the market and to which extent the sellers has control over the market pricing outcomes.

Figure 5.5 illustrates how the social welfare is split in the market, where the portions of discount, profit and reserve price are indicated by different colours. When more resource is provisioned in the market $p = 1.0$, the social welfare is going to be higher (Figure 5.5d) due to a higher resource availability and more satisfied buyers. Moreover, the seller’s reserve price has a significant impact on the social welfare: when the seller’s minimum prices are set too high ($z \geq 0.75$), less buyers can satisfy the RPC and get denied by the market, which results in a reduced social welfare.

We can clearly see that the seller’s reserve price does not allow to achieve a higher payment in the markets of scarce resource (Figures 5.5a-5.5c). It happens because the buyer prices are derived based on competition in the market, and the seller’s reserve price can only eliminate some of the bids with low valuations, resulting in a reduced demand-side competition. However, when there is enough resource available in the market to satisfy the buyers (supply exceeds demand), the competition is absent or very weak. Therefore, the buyer payments drop to the minimum prices required to acquire the resources, which

![Figure 5.4: Computation Time of Market Allocation. Representative Market Scenario: $m = 1.0, z = 0.25, p = 0.25$.]
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

Figure 5.5: Seller Power over Market Pricing Outcomes: Social Welfare Distribution. Market Scenario: $m = 1.0, l = 1.0, N = 512$.

Figure 5.6: Impact of Mechanism Parametrisation on Market Pricing (a), Average Resource Utilisation (b), and Buyer Fulfilment (c). Market Scenario: $m = 1.0, l = 1.0, N = 512, z = 0.25, p = 0.25$. 
is the seller’s reserve price. If the reserve price is zero $z = 0.0$, the resulting seller’s payment will be zero as well (Figure 5.5d), which shows the importance of the seller’s reserve price.

**Comparing Static and Adaptive Mechanisms**

Figure 5.6 illustrates the experimental results obtained by static and adaptive mechanisms for a representative market scenario with high competition for resources: $N = 512$, $z = 0.25$, $p = 0.25$. In Figure 5.6a, we can observe how the social welfare is distributed in the market by the two considered mechanisms. As discussed in our previous experiment, the adaptive approach achieves a better quality allocation, i.e. higher social welfare. The part of social welfare that belongs to the seller’s reserve price is greater in our mechanism because of a bigger amount of sold resources (as confirmed by better resource utilisation in Figure 5.6b). Although, there is a significant difference in resource utilisation, a relatively small contrast in total reserve price can be explained by a low price comparative ratio.

An interesting observation is that the seller profit and the granted buyer discount are distributed by static and adaptive mechanism in a different way. The adaptive mechanism grants a bigger portion of social welfare to the seller’s profit. Although, a much higher percentage of buyers are satisfied by our proposed mechanism (Figure 5.6c), a smaller total discount is granted to the buyers. Such mechanism’s behaviour is specific to the iterative nature of our pricing approach. A static mechanism considers the same resource scarcity for buyer price determination, while our adaptive mechanism searches for the weakest competitor who typically appears in the last allocation iterations when the resources are likely to be scarcer. Therefore, the buyer prices in our proposed mechanism tend to be higher compared to the static one. Such mechanism’s behaviour can have a positive impact on the seller’s incentives, since the better pricing outcomes motivate her to prefer a more socially efficient allocation outcome.

**Computation Time**

In Figure 5.7, we plot the computation time for winner and price determination required by static and adaptive market mechanisms. The static approach is more time-efficient compared to our proposed mechanism; however, both of them clear the market in a feasible computation time. The hardest market scenario with $N = 512$ can be solved by the adaptive mechanism within two minutes.
5.2 Double-sided Market Mechanism

In this section, we design a computational market mechanism for heterogeneous infrastructure cloud services trading in the market with multiple sellers and multiple buyers. The heterogeneous infrastructure cloud services are traded as combinatorial bundles that define the virtual machine configurations in terms of CPU, RAM, and HDD in a custom way.

5.2.1 Allocation Mechanism

The proposed mechanism for heterogeneous infrastructure cloud services allocation is provided in Algorithm 8. The allocation scheme collects the bids, declared by the participating buyers $\hat{\beta}$, and the asks, submitted to the market by the trading sellers $\hat{\alpha}$. It aims to determine the final allocation decision vector $X$, which specifies the couples of buyers and sellers who will exchange the services. The mechanism also returns the list $W$ that contains the actual weights of the resource bundles, traded by the allocation candidates, which is used by the pricing mechanism in its operation.

Algorithm Description

The proposed allocation scheme has to determine the best matching buyers and sellers, who will exchange the services. It is a challenging problem, due to the heterogeneous nature of the traded cloud services, combinatorial buyer requests and the seller pricing for resources of different types. We apply the same principles for candidates allocation in a greedy sorting order as in the HO-DS market mechanism, proposed in Section 4.2. In particular, we consider a couple of bid and ask as an allocation candidate $c_{sb}$. In order to address the problem, we propose an iterative greedy allocation approach based on the

![Figure 5.7: Computation Time of Market Mechanisms: HE-SS-Static vs HE-SS-Adaptive. Representative Market Scenario: $m = 1.0, l = 1.0, z = 0.25, p = 0.25$.](image)
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

Algorithm 8 Iterative Allocation Scheme for Double-sided Market of Heterogeneous Cloud Infrastructure Services (HE-DS-A)

1: Input: \( \hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N) \), where \( \hat{\beta}_b = (\hat{C}_b^d, p_{br}^b) \)
2: Input: \( \hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_M) \), where \( \hat{\alpha} = (\hat{C}_s^d, P_s^r) \)

\# Step 1: Initialization Phase
3: \( X \leftarrow \emptyset \) // final allocation decision matrix
4: \( L \leftarrow \emptyset \) // list of best efficiency candidates
5: \( W \leftarrow \emptyset \) // list of best efficiency candidates’ weights
6: \( L_i \leftarrow \{ \sigma_{\phi\sigma} \mid \forall \phi \in S \land \forall \sigma \in B \} \) // initial list of candidates

\# Step 2: Iterative Allocation Phase
7: for \( i = 1 \) to \( MN \) do
8: \( (s, b) = \arg \max_{(\phi, \sigma)} e(\sigma_{\phi\sigma}, X) \), where \( \forall \sigma_{\phi\sigma} \in \{ L_i \setminus L \} \)
9: \( L \leftarrow L \cup \sigma_{sb} \)
10: \( W \leftarrow W \cup \{ w_{sb} \} \), where \( w_{sb} = w(\hat{\beta}_b, \sigma_s, X) \)

\# Step 2a: Determine Best Efficiency Candidate
11: if \( \text{ARC}(\sigma_{sb}, X) \wedge \text{RPC}(\sigma_{sb}) \wedge \text{BAC}(\hat{\beta}_b, X) \) then
12: \( x_{sb} = 1 \)
13: else
14: \( x_{sb} = 0 \)
15: end if
16: \( X \leftarrow X \cup x_{sb} \)
17: end for
18: Output: \( X, W \)

sorted list of candidates. The iterative approach allows for the market to select the winners in an adaptive way, based on the changing market conditions, such as resources scarcity.

Our proposed double-sided allocation mechanism runs in a number of steps:

- **Initialization Phase (lines 3-6):** Firstly, the allocation scheme initialises the variables required for its operation, such as the final allocation decision vector \( X \), the list of determined best efficiency candidates \( L \), and the list \( W \) which keeps the actual weights of the resource bundles traded in the market (at the moment when they are considered for allocation)\(^1\). The mechanism also constructs the initial list \( L_i \) with all possible allocation candidates among the participating buyers and sellers. Similar to HO-DS mechanism, an allocation candidate is a couple of seller’s ask and the buyer’s bid \( \sigma_{sb} = (\hat{\alpha}_s, \hat{\beta}_b) = (\hat{C}_s^d, \hat{p}_b^s, p_{br}^b) \). It consists of the bundle of resources to be exchanged \( \hat{C}_b^d \), the buyer’s price valuation \( \hat{p}_b^s \) and the bundle reservation price of the corresponding seller \( p_{br}^b \).

\(^1\)The information about the traded weight of resources is needed in order to realise the seller pricing mechanism, which we outline later.
• **Iterative Allocation Phase** (lines 7-17): The allocation scheme makes allocation decisions in $MN$ allocation iterations. In each iteration, only the candidate with the highest sorting criteria value is selected and considered for allocation. The following steps are involved in the process:

  – **Best Efficiency Candidate Determination** (lines 8-10): In order to determine the most socially efficient allocation candidate in a particular allocation iteration, the mechanism calculates the sorting criteria values associated with all (previously not considered $L^i \setminus L$) candidates. We provide a detailed explanation of the sorting criteria function below. Please note that the mechanism keeps a list of the considered candidates $L$ (line 9) and saves their associated weights in $W$ (line 10).

  – **Problem Constraints Verification** (lines 11-16): Once, the best efficiency candidate is identified, the mechanism checks if the resources could be exchanged between the corresponding buyer and seller. Therefore, the allocation scheme verifies the problem constraints and makes the allocation decision. The following constraints are verified:

    * **Available Resource Constraint** (ARC): is verified to make sure that the seller has enough resource available in order to satisfy the buyer request in current allocation iteration, as follows:
      
      $$\hat{c}_{bg}^d \leq \hat{c}_{sg}^a, \forall g \in G,$$
      
      where $\hat{c}_{sg}^a = \hat{c}_{sg} - \sum_{b \in B} \hat{c}_{bg} x_{sb}$

    * **Reserve Price Constraint** (RPC): makes sure that the buyer’s price valuation is sufficient to satisfy the seller’s corresponding bundle price reservation as follows: $\hat{p}_b^a \geq \hat{p}_{ab}^b$.

    * **Binary Allocation Constraint** (BAC): checks whether the buyer has been previously allocated to some other seller. Formally, the constraint is defined as follows: $\sum_{s \in S} x_{sb} \leq 1$.

**Sorting Criteria Function**

The sorting criteria function $e(\hat{\beta}_b, \hat{\alpha}_s, X)$ is used to associate the sorting criteria values with the allocation candidates. It guides the greedy allocation mechanism in its best efficiency candidate selection. The candidate with the highest sorting criteria value will become the best efficiency candidate in a particular allocation iteration. In order for the mechanism to
determine the efficient allocations of bids to asks, the sorting criteria function has to assign such values to each candidate that the market’s social welfare is maximised. The double-sided market mechanism aims to maximise the total market generated surplus produced by the allocations. Therefore, a more socially efficient allocation candidate will exchange less resource, while generating a greater surplus. Based on these principles, our sorting criteria function is defined as follows:

\[ e(\varsigma_{sb}, X) = e((\hat{\beta}_b, \hat{\alpha}_s), X) = \frac{\hat{p}^v_b - \hat{p}^{br}_{sb}}{w(\beta_b, \alpha_s, X)} \]

The proposed sorting criteria function is based on the declarations of the associated buyer bid and seller ask as well as on the allocation decisions, made until the current allocation iteration \( X \). The basic idea is to determine the amount of surplus that the candidate can potentially contribute to the market, given the weight of the bundle of services exchanged by the candidate. The potential surplus is determined as the difference between the buyer’s proposed price valuation \( \hat{p}^v_b \) and the seller’s desired bundle price reservation \( \hat{p}^{br}_{sb} \). The weight of the exchanged bundle depends on the amounts of resource, requested by the buyer \( \hat{C}^d_b \) and the scarcity of these resources from a corresponding seller. The resource scarcity factor \( F^s \), is defined as inverse of available capacities of resources as follows:

\[ F^s = \{f^s_1, \ldots, f^s_K\}, \text{ where } f^s_{sg}(\hat{c}^s_{sg}, X) = \left(\hat{c}^{sa}_{sg}\right)^{-1} \left(\hat{c}^s_{sg} - \sum_{b \in B} \hat{c}^d_{bg} \hat{p}^{br}_{sb}\right)^{-1}, \forall g \in G \]

In other words, the scarcer resource will have a larger scarcity factor, thus it will contribute a greater weight to the traded bundle. However, in the double-sided market, there may be multiple sellers who offer their resources and one seller could provide more resources than the others. In such situation, the larger sellers will gain significant advantage over the smaller ones, since their resources are less scarce compared to the smaller sellers. In order to address this issue and to ensure a fair competition among the sellers, we introduce a normalization factor, which normalises the resource scarcity across the sellers. The proposed normalization factor is defined as follows:

\[ f^n_s(\hat{C}^s, X) = \left(\sum_{g \in G} f^s_{sg}(\hat{c}^s_{sg}, X)\right)^{-1}, \forall s \in S \]

As a result, the weight of the exchanged bundle of resources is defined as follows:

\[ w(\hat{\beta}_b, \hat{\alpha}_s, X) = \left(\sum_{g \in G} (\hat{c}^d_{bg} \times f^s_{sg}(\hat{c}^s_{sg}, X) \times f^n_s(\hat{C}^s, X))^l\right)^m \]
The normalised scarcity factor shows how the seller’s resources are scarce in comparison to one another. When applied on the amounts of requested resource in the bundle, the bundle weight would reflect how well the requested bundle would fit the seller’s currently available resources. For example, if the candidate aims to exchange a lot of scarce resource, the associated bundle weight will be high; and vice versa.

Similar to the single-sided iterative market mechanism, provided in Section 5.1, the introduced parameters $l \geq 0$ and $m \geq 0$ allow controlling the market mechanism operation, depending on different allocation objectives. Specifically, the parameter $l \geq 0$ allows controlling the significance of the resource balance in the traded bundle, and the parameter $m \geq 0$ allows controlling the importance of the traded resource size. Please refer to Section 5.2 for examples of the mechanism’s parameters impact.

**Allocation Mechanism Example**

We illustrate the allocation decision-making process of our proposed double-sided market mechanism for trading heterogeneous cloud services by providing a simple example. We assume that the market allows trading the cloud resources of three following types: CPU, RAM, and HDD. The two cloud providers join the market with their cloud services offers and five prospective cloud consumers are willing to exchange the services. The sellers’ asks and the buyers’ bids, used in this example, are depicted in the Table 5.6. We can see that the cloud providers supply different amounts of resources and have different unit price reservations. The participating buyers, in turn, report their custom bundles of required resources and express their price valuations. The allocation process of the proposed double-sided mechanism is summarized in Table 5.7. For simplicity, we consider a linear form of the sorting criteria function, where $l = 1.0$ and $m = 1.0$.

Initially, given two seller asks and five buyer bids, the allocation scheme constructs ten possible allocation candidates in the list $\overline{L}^i$. For example, the candidate between the ask $S1$ and the bid $B1$, will look as follows $(S1, B1) = ((2, 1, 150), \$1.2, \$0.193)$. The constructed candidates are considered for allocation in 10 allocation iterations, where the best efficiency candidate (with highest sorting criteria value) is determined in each iteration round. In the first allocation iteration, the sorting criteria values associated with each candidate are determined based on the bundle of requested resources, the associated seller’s available resources and the corresponding price valuations. For instance, the sorting criteria value
of the candidate \((S_1, B_1)\) (in the first allocation iteration) is determined as follows:

\[
e(\langle S_1, B_1 \rangle) = \frac{\$1.2 - \$0.193}{2.7138 \times (2 \times 5^{-1} + 1 \times 6^{-1} + 150 \times 550^{-1})} \approx 0.4421
\]

\[F_{S1}^a = (5^{-1}, 6^{-1}, 550^{-1})\]

\[F_{S1}^w = (5^{-1} + 6^{-1} + 550^{-1})^{-1} = 2.7138\]

Please, note that even though S2 offers a more competitive bundle reservation price for the bundle requested by B1, compared to the reservation price of S1, the sorting criteria value of \((S_2, B_1)\) is less competitive than \((S_1, B_1)\) in this allocation iteration. It happens because, the CPU resource is scarcer on the seller S2, which results in a greater bundle weight associated with the candidate. As a result, the \((S_2, B_2)\) becomes the best efficiency candidate in the first allocation iteration. We can see that there is enough available resource in order to allocate the candidate, the price reservation of the seller S2 is satisfied and the bid B2 was not previously allocated. Therefore, the mechanism makes the decision to allocate the selected best efficiency candidate.

In the following allocation iteration, the sorting criteria values of the remaining allocation candidates are updated according to the new capacities of available resources. Since,

**Table 5.6: HE-DS Mechanism Example: Market Input**

(a) Sellers’ Offers / Asks

<table>
<thead>
<tr>
<th>Seller, (s)</th>
<th>Offered Resources, (\hat{C}_s^o)</th>
<th>Price Reservations, (\hat{P}_s^{pur})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (core)</td>
<td>RAM (GB)</td>
</tr>
<tr>
<td>S1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>S2</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

(b) Buyers’ Requests / Bids

<table>
<thead>
<tr>
<th>Buyer, (b)</th>
<th>Requested Resources, (\hat{C}_b^r)</th>
<th>Val., (\hat{p}_b^v)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (core)</td>
<td>RAM (GB)</td>
</tr>
<tr>
<td>B1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>B4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>B5</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 5.7: HE-DS Mechanism Example: Allocation Procedure \((m = 1, l = 1)\)

<table>
<thead>
<tr>
<th>Allocation Candidate in (L^i)</th>
<th>Best Effic. Cand. (\hat{C}_{sb}^d)</th>
<th>(\hat{C}_{sb}^{sa})</th>
<th>ARC</th>
<th>(\hat{p}_b^v)</th>
<th>(\hat{p}_sb^{br})</th>
<th>RPC</th>
<th>BAC</th>
<th>(x_{sb})</th>
</tr>
</thead>
<tbody>
<tr>
<td>((S1, B1))</td>
<td>((S1, B2))</td>
<td>((S1, B3))</td>
<td>((S1, B4))</td>
<td>((S1, B5))</td>
<td>((S2, B1))</td>
<td>((S2, B2))</td>
<td>((S2, B3))</td>
<td>((S2, B4))</td>
</tr>
<tr>
<td>1</td>
<td>0.44</td>
<td>0.54</td>
<td>0.04</td>
<td>0.14</td>
<td>0.13</td>
<td>0.43</td>
<td>0.55</td>
<td>0.04</td>
</tr>
<tr>
<td>2</td>
<td>0.44</td>
<td>0.54</td>
<td>0.04</td>
<td>0.14</td>
<td>0.13</td>
<td>0.45</td>
<td>-</td>
<td>0.04</td>
</tr>
<tr>
<td>3</td>
<td>0.44</td>
<td>-</td>
<td>0.04</td>
<td>0.14</td>
<td>0.13</td>
<td>0.45</td>
<td>-</td>
<td>0.04</td>
</tr>
<tr>
<td>4</td>
<td>0.44</td>
<td>-</td>
<td>0.04</td>
<td>0.14</td>
<td>0.13</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>0.04</td>
<td>0.14</td>
<td>0.13</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>-</td>
<td>0.12</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
</tr>
</tbody>
</table>
the resources of the seller S2 were allocated, only the values of the candidates associated with this seller change. The candidate \( (S1, B2) \) becomes the best efficiency candidate, but since the bid \( B2 \) was previously allocated, this candidate is rejected. The process repeats until all ten allocation candidates are considered. As a result, three allocation candidates are allocated by the mechanism and the corresponding sellers and buyers will exchange the resources: \( (S2, B2) \), \( (S2, B1) \) and \( (S1, B4) \). The remaining allocation candidates are rejected either due to insufficient amount of available resource or due to being previously allocated. In this example, all the candidates satisfied RPC, but this constraint is also verified and can be the reason for rejection.

The mechanism allocates the following amounts of cloud resources: \( (4, 5, 450) \) and \( (3, 5, 400) \), for the seller S1 and S2, respectively. The corresponding resource utilization values are \( (80\%, 83\%, 82\%) \) and \( (75\%, 83\%, 0.89\%) \). The social welfare produced by the allocation scheme is \( W = (\$2.2 - \$0.27) + (\$1.2 - \$0.162) + (\$1.5 - \$0.539) = \$3.929 \). In this example, we can see rather well-balanced resource utilisation on both sellers; a detailed study of the proposed mechanism allocative performance is provided in experimental section.

### 5.2.2 Pricing Mechanism

A double-sided market mechanism has to determine the corresponding prices for both the buyers and sellers. We propose a composite pricing scheme, which initially establishes the prices that the buyers pay based on the competition in the market, and derive the final sellers’ payments by distributing the surplus generated in the market based on mixed rules. We provide a detailed explanation of our proposed pricing mechanisms with illustrative examples, and investigate the mechanism theoretical properties.

#### Buyer Pricing Algorithm Description

The proposed pricing scheme for the buyer price determination is given in Algorithm 9. The mechanism receives the declarations of the participating buyers \( \hat{\beta} \), the offers of the participating sellers \( \hat{\alpha} \), and the previously determined allocation decision vector \( X \). The goal is to establish the prices that the buyers have to pay for the granted resources \( P^b \).

Our proposed buyer pricing scheme establishes the final prices that the buyers have to pay for the granted resources in order. The initially rejected buyers, that were not allocated to any seller, receive zero payment (lines 6, 26). The mechanism derives the final prices based on competition in the market for all the buyer bids that were allocated (lines 8-24). The basic idea is to determine all the competitors of the winning buyer bid
in different allocation iterations, and to calculate the minimum price required to outbid at least one of them (the weakest competitor). Therefore, our proposed buyer pricing scheme determines the payments for each individual buyer in a number of steps:

**Algorithm 9** Double-sided Buyer Pricing of Heterogeneous Cloud Infrastructure Services (HE-DS-PB)

1: Input: \( \bar{\beta} = (\bar{\beta}_1, \ldots, \bar{\beta}_N) \)
2: Input: \( \bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_M) \)
3: Input: \( X = x_{sb}, \forall s \in S, \forall b \in B \)
4: \( p^b \leftarrow 0 \)
5: for all \( \bar{\beta}_b \in \bar{\beta} \) do
6: \( \rho^b_b = 0 \)
7: if \(!\text{BAC}(\bar{\beta}_b, X)\) then

   # Step 1: Initialization Phase
8: \( P^c \leftarrow \emptyset \) // minimum competitive prices
9: \( X \leftarrow \emptyset \) // new allocation results
10: \( L \leftarrow \emptyset \) // best efficiency candidates in order
11: \( L^i \leftarrow \{s_{\phi \sigma} \mid \forall \phi \in S \land \forall \sigma \in B \land \sigma \neq b\} \)
12: \( L^i \leftarrow L^i \cup \{s_{\phi \sigma} \mid \forall \phi \in S\} \) sorted in increasing order of \( \rho^b_{sb} \)

   # Step 2: Iterative Competitors Determination Phase
13: for \( i = 1 \) to \( MN \) do
14: \( \langle \bar{s}, \bar{b} \rangle = \arg \max_{\langle \phi, \sigma \rangle} e(s_{\phi \sigma}, X) \), where \( \forall s_{\phi \sigma} \in \{L^i \setminus L\} \)
15: \( \bar{L} \leftarrow \bar{L} \cup s_{\bar{s} \bar{b}} \)

   # Step 2a: Best Efficiency Candidate Determination
16: if \( \text{ARC}(s_{\bar{s} \bar{b}}, \bar{X}) \land \text{RPC}(s_{\bar{s} \bar{b}}) \land \text{BAC}(\bar{\beta}_b, \bar{X}) \) then
17: if \( \text{ARC}(s_{\bar{s} \bar{b}}, \bar{X}) \land \text{RPC}(s_{\bar{s} \bar{b}}) \) then
18: \( P^c \leftarrow P^c \cup \{s_{\bar{s} \bar{b}} \mid e(s_{\bar{s} \bar{b}}, \bar{X}) \times w(s_{\bar{s} \bar{b}}, \bar{X})\} \)
19: end if
20: \( x_{\bar{s} \bar{b}} = 1 \)
21: \( \bar{X} \leftarrow \bar{X} \cup x_{\bar{s} \bar{b}} \)
22: end if
23: end for

   # Step 2b: Competitors Verification
24: if \( \text{ARC}(s_{\bar{s} \bar{b}}, \bar{X}) \land \text{RPC}(s_{\bar{s} \bar{b}}) \land \text{BAC}(\bar{\beta}_b, \bar{X}) \) then
25: \( p^b \leftarrow p^b \cup \rho^b_b \)
26: end if
27: end for

   # Step 3: Price Determination Phase
28: \( \rho^b_b = \max \left( \sum_{\phi \in S} p^c_{\phi \sigma} x_{\phi \sigma}, \min(p^c) \right) \)
29: end if
30: \( P^b \leftarrow P^b \cup \rho^b_b \)
31: end for
32: Output: \( P^b = \{\rho^b_1, \ldots, \rho^b_N\} \)

- **Initialization Phase (lines 8-12):** Firstly, the mechanism initialises the variables required for its operation, such as the auxiliary allocation decision variable \( \bar{X} \) to keep track of the new ongoing allocations, the variable \( \bar{L} \) to keep track of the consid-
ered best efficiency candidates, and the variable $\bar{P}^c$ which will contain the minimum competitive prices for the considered winning buyer $\hat{\beta}_b$. The pricing scheme also constructs a new initial set of allocation candidates $\tilde{L}^i$, which contains all candidates among the participating sellers and buyers, except for the candidates with the currently considered winning bid (line 11). These candidates are replaced by the corresponding bundle-reservation candidates (line 12) and placed at the end of the list $\tilde{L}^i$ in increasing order of their bundle reservation price $\bar{p}_{\text{br}}^b$. It ensures that the bundle-reservation candidate with the lower price reservation is considered first.

A bundle reservation candidate $\zeta_{sb} = (\tilde{C}_b, \bar{p}_{\text{br}}^b, \bar{p}_{\text{br}}^i)$ aims to exchange the same bundle of resources as the initial candidate $\zeta_{sb}$, but the buyer’s initial price valuation $\bar{p}_b^b$ is replaced by the seller’s bundle-specific price reservation $\bar{p}_{\text{br}}^b$.

- *Iterative Competitors Determination Phase (lines 13-23):* The competitors of the winning buyer have to be determined in an iterative fashion, similar to the way allocation scheme operates. Therefore, the mechanism runs $MN$ iterations and a single best efficiency candidate is verified to be a competitor in each round.

  - *Best Efficiency Candidate Determination (lines 14-15):* Given the capacities of resources available from different sellers at the current allocation iteration $1 \leq i \leq MN$, the sorting criteria values are determined for each not-considered allocation candidate $\{\tilde{L}^i \setminus \tilde{L}\}$. The best efficiency candidate $\zeta_{sb}$ with the highest sorting criteria value, is selected by the mechanism. This candidate will be verified to be the competitor of the currently considered winning bid.

  - *Competitors Determination (lines 16-23):* In order for the current best efficiency candidate $\zeta_{sb}$ and the candidate with the considered winning bid $\zeta_{sb}$ to be the competitors, they need to satisfy the allocation feasibility condition.

    * Allocation feasibility condition (lines 16-17):* ensures that both candidates are feasible for allocation in the current allocation iteration. Therefore, both of them have to satisfy all the required problem constraints.

Once, the competitor candidate is found, the minimum price required for the currently considered winning bid to outbid her competitor is calculated and saved in the list of competitor prices $\bar{P}^c$ (line 18). Please, note that the auxiliary allocation decision $\hat{X}$ is updated according to the new ongoing allocation of the candidates in the list $\tilde{L}^i$ (lines 20-21), which is used in order to determine the

\[\text{2Please, recall that the sorting criteria value of any bundle-reservation candidate is always zero}\]
ongoing sorting criteria values of the remaining candidates.

- Price Determination Phase (line 24): Finally, when all the allocation candidates in the list $L$ are considered, and their competitors with the corresponding minimum prices are identified, the mechanism selects the smallest required price as the final buyer’s price $\min(P_c)$. Please, note that the mechanism makes sure that the final buyer price is at least the bundle-reservation price, desired by the allocated seller. This condition is verified in order to ensure seller’s Individual Rationality property by compromising on buyer Truthfulness. A more detailed discussion on impossibility to achieve both design properties are provided in theoretical investigation section.

Buyer Pricing Mechanism Example

We continue the initial example and illustrate the buyer pricing procedure. Once the allocation decisions have been made, the corresponding prices that the buyers pay for the granted resources need to be determined. The procedure of the proposed buyer pricing scheme is given in the Table 5.8.

The buyers who were not allocated the resources in the market, such as B3 and B5, do not pay anything (price is zero). The prices for B1, B2 and B4 are determined based on the competition in the market. Therefore, the pricing scheme runs an iterative competitors determination procedure for each of the winning buyers individually. The goal is to determine the competitors and the minimum prices required for the buyer to outbid them.

For example, for the bid B1, the pricing mechanism constructs a new set of allocation candidates, which include all possible candidates among the participating buyers and sellers, where the ones with the currently considered winning buyer B1 are replaced by the corresponding bundle reservation candidates, e.g. (S1, B1BR) and (S2, B1BR). The allocation candidates in $L$ are verified to be the competitors of the currently considered winning buyer B1 in ten allocation iterations. In the first allocation iteration $i = 1$, the best efficiency candidate is (S2, B2). This candidate is feasible for allocation $\bar{x}_{s2} = 1$ (satisfies all constraints). We can see that the corresponding candidate with the currently considered winning buyer (S2, B1) is also feasible for allocation; hence, the candidates are the competitors. Therefore, the mechanism calculates the minimum price valuation, required for the buyer B1 in order to outbid her competitor, as follows:
Table 5.8: HE-DS Mechanism Example: Buyer Pricing Procedure ($m = 1, l = 1$)

<table>
<thead>
<tr>
<th>Buyer, $b$</th>
<th>Iterative Competitors Determination</th>
<th>Final Price $\rho^*_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterative Efficiency Bids in $\hat{L}$</td>
<td>Conditions</td>
</tr>
<tr>
<td></td>
<td>$i$</td>
<td>$c_{bi}$</td>
</tr>
<tr>
<td>B1 1</td>
<td>1</td>
<td>(S2, B2)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(S1, B2)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(S1, B4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(S2, B4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(S1, B5)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>(S2, B5)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>(S2, B3)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>(S1, B3)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>(S2, B1BR)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>(S1, B1BR)</td>
</tr>
</tbody>
</table>

| B2 1       | 1   | (S1, B1) | (5, 6, 550) | 1 1 ✓ | $\bar{p}^c$ = $1.8600$ |
|           | 2   | (S2, B1) | (4, 6, 450) | 0 - X | - |
|           | 3   | (S2, B4) | (4, 6, 450) | 1 1 ✓ | $\bar{p}^c$ = $0.7969$ |
|           | 4   | (S2, B5) | (0, 1, 0) | 0 - X | - |
|           | 5   | (S1, B4) | (3, 5, 400) | 0 - X | - |
|           | 6   | (S1, B5) | (3, 5, 400) | 0 - X | - |
|           | 7   | (S2, B3) | (0, 1, 0) | 0 - X | - |
|           | 8   | (S1, B3) | (3, 5, 400) | 1 1 ✓ | $\bar{p}^c$ = $0.3972$ |
|           | 9   | (S2, B2BR) | (0, 1, 0) | 0 - X | - |
|           | 10  | (S1, B2BR) | (2, 3, 250) | 0 - X | - |

B3 0 - - - - - - - - - $\bar{p}^c$ = $0.0000$
Table 5.8: HE-DS Mechanism Example: Buyer Pricing Procedure \((m = 1, l = 1)\)

<table>
<thead>
<tr>
<th>Buyer, (b)</th>
<th>Iterative Competitors Determination</th>
<th>Final Price (\rho^b)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best Efficiency Bids in (\hat{L})</td>
<td>Conditions</td>
</tr>
<tr>
<td></td>
<td>(i)</td>
<td>(\varsigma_ib)</td>
</tr>
<tr>
<td>(S1, B5)</td>
<td>5</td>
<td>6 (S2, B2)</td>
</tr>
<tr>
<td>(S1, B4)</td>
<td>6</td>
<td>1 (S1, B2)</td>
</tr>
<tr>
<td>(S1, B3)</td>
<td>7</td>
<td>1 (S2, B1)</td>
</tr>
<tr>
<td>(S1, B1)</td>
<td>8</td>
<td>1 (S1, B1)</td>
</tr>
<tr>
<td>(S2, B5)</td>
<td>9</td>
<td>1 (S1, B5)</td>
</tr>
<tr>
<td>(S2, B4)</td>
<td>10</td>
<td>1 (S1, B4)</td>
</tr>
<tr>
<td>(S2, B3)</td>
<td>11</td>
<td>1 (S2, B3)</td>
</tr>
<tr>
<td>(S1, B1)</td>
<td>12</td>
<td>1 (S2, B1)</td>
</tr>
<tr>
<td>(S1, B3)</td>
<td>13</td>
<td>1 (S1, B3)</td>
</tr>
<tr>
<td>(S2, B5)</td>
<td>14</td>
<td>1 (S2, B5)</td>
</tr>
<tr>
<td>(S1, B4)</td>
<td>15</td>
<td>1 (S1, B4)</td>
</tr>
</tbody>
</table>

The process repeats until all the allocation candidates are considered. We can see that the mechanism has determined the following three competitors: \(\langle S2, B2\rangle\), \(\langle S1, B4\rangle\) and \(\langle S2, B3\rangle\) for the winning buyer B1, with the corresponding minimum prices $1.4729, $0.5220 and $0.2430. The smallest competitive price is selected as the final price that the buyer would pay.

\[
\hat{p}^{br}(\langle S2, B1\rangle) + e(\langle S2, B2\rangle, \tilde{X}) \times w(\langle S2, B1\rangle, \tilde{X}) = \$0.162 + 3.5146 \times 2.3873 = \$1.4729
\]

\[
\hat{p}^{br}(\langle S2, B1\rangle) = \$0.04 \times 2 + \$0.04 \times 1 + \$0.00028 \times 150 = \$0.162
\]

\[
e(\langle S2, B2\rangle, \tilde{X}) = \frac{\$2.20 - \$0.27}{2.3873 \times ((1 \times 4^{-1})^1 + (4 \times 6^{-1})^1 + (250 \times 450^{-1})^1)^1} = 3.5146
\]

\[
w(\langle S2, B1\rangle, \tilde{X}) = 2.3873 \times ((1 \times 4^{-1})^1 + (1 \times 6^{-1})^1 + (150 \times 450^{-1})^1)^1 = 2.3873
\]
B1 have to pay. It means that in order for the buyer B1 to be allocated, she has to outbid at least the buyer B3 on the seller S2 with the price of $0.2430.

The mechanism determines the final prices for the buyer B2 and B4 in a similar fashion. We would like to draw your attention to the third allocation iteration $i = 3$ when the price of B4 is determined. The candidates are not considered as competitors, since the allocation for the candidate with the winning buyer (S2, B4) is not feasible due to insufficient available resource from the seller S2. We can see that despite that some bids are initially allocated to one seller, the final price that they pay may be derived based on the competition for the resources from the other seller. For example, the price that B2 have to pay for the resources allocated from the seller S2, is derived based on the competition for the resources from the seller S1, since it is the minimum price in order for B2 to remain the winner.

**Seller Pricing Mechanism Description**

The seller pricing scheme is given in Algorithm 10. The mechanism receives the declarations of the market participants, including bids $\hat{\beta}$ and asks $\hat{\alpha}$, the previously determined allocation decision variable $X$, the list of actual weights of the best efficiency candidates $W$ and the final buyer prices $P^b$. The goal is to determine the final seller payments $P^s$.

Due to the multi-dimensional seller valuations, where the unit price reservations are set for the resources of different types, the proposed allocation scheme is not a monotonic function in sellers’ declarations. Therefore, there is no seller’s critical-value payment; thus, the pricing algorithms, complementary to the proposed iterative greedy allocation mechanism, would not generate truthful seller payments. Hence, our goal is to design such a seller pricing mechanism that would limit the seller’s strategic manipulation. We apply the seller pricing approach proposed for HO-DS-PS mechanism in Section 4.2. The goal is to derive the seller payments so that the final prices do not depend directly on the seller’s declaration. The proposed pricing mechanism distributes the surplus among the sellers either based on their proportional-value of the contributed market volume, or based on direct payment from the allocated buyer. The degree of importance of the two considered surplus distribution rules is controlled by the market parameter $0 \leq \mu \leq 1$. For a more detailed description of the surplus distribution rules, please refer to the Section 5.

In order to determine the final seller payments, the proposed pricing mechanism performs the following steps:

- **Market Volume and Surplue (lines 6-8):** Firstly, the mechanism calculates the essential market values, including the total surplus generated by the buyer payments...
Algorithm 10 Double-sided Pricing of Homogeneous Cloud Infrastructure Services

1: Inputs: \( \hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_N) \)
2: Input: \( \hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_M) \)
3: Input: \( X = x_{sb}, \forall s \in S, \forall b \in B \)
4: Input: \( W \) // actual weights of the best efficiency candidates
5: Input: \( P^b = \{p^b_1, \ldots, p^b_N\} \)

# Step 1: Market Volume and Surplus
6: \( P^s \leftarrow 0 \)
7: \( \Delta = \sum_{b \in B} p^b_s - \sum_{s \in S} \sum_{b \in B} p^b_{sb} x_{sb} \) // generated surplus
8: \( \Omega = \sum_{s \in S} \sum_{b \in B} w_{sb} x_{sb} \) // total market volume

# Step 2: Surplus Distribution
9: for all \( \hat{\alpha}_s \in \hat{\alpha} \) do
10: \( \Omega_s = \sum_{b \in B} w_{sb} x_{sb} \)
11: \( \Delta_s = \sum_{b \in B} (p^b_s - p^b_{sb}) x_{sb} \)
12: \( \rho^s = \sum_{b \in B} p^b_{sb} x_{sb} + \Delta (\mu \frac{\Omega}{\Omega} + (1 - \mu) \frac{\Delta_s}{\Delta}) \)
13: \( P^s \leftarrow P^s \cup \rho^s \)
14: end for
15: Output: \( P^s = \{\rho^s_1, \ldots, \rho^s_M\} \)

\( \Delta \) and the total market volume of sold resources \( \Omega \). The major difference between the HO-DS-PS and HE-DS-PS mechanisms is that the weights of the traded bundles have to be determined at the moment of allocation by HE-SS-A scheme, taking into account the scarcity of resources at a particular allocation iteration. Hence, the market volume is derived based on the previously determined weights.

- **Surplus Distribution (lines 9-14):** The mechanism decides the amount of surplus to be granted to each participating seller in addition to their desired minimum price reservations. It is realised via the mixed surplus distribution rule, controlled by the parameter \( 0 \leq \mu \leq 1 \), where \( \mu = 0 \) grants the surplus based on direct payment, \( \mu = 1 \) relies on the proportional-value payment rule, and any value in between \( 0 < \mu < 1 \) result in a mixed rule.

**Seller Pricing Mechanism Example**

We follow the initial example and illustrate the seller payment determination procedure of our proposed HE-DS-PS mechanism, as depicted in Table 5.9.

The pricing mechanism determines the total market generated surplus \( \Delta = 1.0443 \) and the market exchanged volume \( \Omega = 12.4538 \) based on the individual contributions of each allocated candidate (Table 5.9a). Afterwards, the mechanism determines the payments
Table 5.9: HE-DS Mechanism Example: Seller Pricing Procedure

(a) Market Values

<table>
<thead>
<tr>
<th>Winning Candidate, $s_{ab}$</th>
<th>Market Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\rho_{b}^{p}$</td>
</tr>
<tr>
<td>(S2, B1)</td>
<td>$0.2430$</td>
</tr>
<tr>
<td>(S2, B2)</td>
<td>$0.3972$</td>
</tr>
<tr>
<td>(S1, B4)</td>
<td>$1.3752$</td>
</tr>
</tbody>
</table>

$\Delta = 1.0443 \quad \Omega = 12.4538$

(b) Individual Seller Values

<table>
<thead>
<tr>
<th>Seller, $s$</th>
<th>Individual Seller Values</th>
<th>Final Payment, $\rho_{s}^{p}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Volume $\Omega_{s}$</td>
<td>Surplus $\Delta_{s}$</td>
</tr>
<tr>
<td>S1</td>
<td>6.6530</td>
<td>$0.8362$</td>
</tr>
<tr>
<td>S2</td>
<td>5.8009</td>
<td>$0.2081$</td>
</tr>
</tbody>
</table>

of each seller depending on the choice of parameter $\mu$. We can clearly see that the seller S1 has sold a bigger volume of resources and has contributed more surplus to the market. Therefore, regardless of the value of parameter $\mu$, S1 will receive a bigger portion of market surplus. If a mixed rule with $\mu = 0.5$ is applied, the payment of S1 is calculated as follows:

$$\rho_{S1}^{p} = 0.5390 + 1.0443 \times \left( 0.5 \times \frac{6.6530}{12.4538} + (1.0 - 0.5) \frac{0.8362}{1.0443} \right) = 1.2360$$

In Table 5.9b, we depict how the seller payments change with different values of parameter $\mu$. We can see that S1 is better off relying on direct payment rule, since her allocated buyers have contributed a larger portion of surplus in the market. The seller S2 has sold slightly less resources despite contributing much smaller surplus, which is the reason why the proportional-payment rule is a better choice for S2 in this example.

5.2.3 Theoretical Investigation

Seller Individual Rationality (IR) vs. Buyer Truthfulness (TB)

In order to illustrate the impossibility to achieve both economic properties: Seller Individual Rationality and Buyer Truthfulness, we consider a simple example. We assume that
there is a single buyer B1 requesting resources in the market and there are two sellers S1 and S2, making their market offers. The market setup, considered in this example, is depicted in Tables 5.10a and 5.10b.

We can clearly see that, while the seller S2 offers a better price reservation for the bundle of resources, requested by B1, the bundle-balance in regards to the sellers’ resource scarcities is much better on S1 (Table 5.10c). For simplicity, a linear mechanism’s parametrisation \( l = 1.0, m = 1.0 \) is considered. The mechanism selects the candidate \((S1, B1)\) as the best efficiency candidate which will be allocated and exchange the resources. In this example, the critical-value payment is the lowest possible reservation price $0.09, which is the minimum price desired by the seller S2. However, we can notice that such price is below the reservation price required for the resources allocated from the selected seller S1, and this payment would violate the Seller’s Individual Rationality. Since,
IR is the design feasibility constrain, we compromise on the Buyer Truthfulness in order to guarantee a practical mechanism design. Therefore, in the buyer price determination phase, we ensure that the lowest buyer price is the reservation price of the corresponding allocated seller. In the considered example, instead of the critical-value payment, the price that the buyer B1 pays is the reservation price of the seller S1, i.e. $0.16.

**Individual Rationality (IR)**

**Theorem 14.** *The proposed pricing scheme maintains Individual Rationality property for single-minded buyers declared types.*

**Proof.** We need to show that buyer IR property holds as defined in Formula 3.4. We focus on the pricing of allocated buyers because our proposed pricing mechanism does not charge the denied bids, which complies to the IR definition. We assume that the IR condition for the buyers does not hold as follows $\rho_b^h > \tilde{p}_b^v$, and provide the proof by contradiction. The buyer payment is defined as the larger value between the bundle reservation price of the allocated seller and the minimum price in order to outbid the weakest competitor:

- **Bundle reservation price:**

  \[
  \rho_b^h > \tilde{p}_b^v \\
  \rho_b^h = \sum_{\phi \in S} \rho_{\phi b}^v x_{\phi b} \Rightarrow \sum_{\phi \in S} \rho_{\phi b}^v x_{\phi b} > \tilde{p}_b^v
  \]

  It means that the bundle reservation price of the allocated seller is higher than the buyer’s price valuation. However, such a bid could not be allocated at the first place due to the violated RPC.

- **Competitor-based price:** Based on the way the minimum competitive prices are derived, we can conclude the following:

  \[
  \rho_b^h > \tilde{p}_b^v \\
  \rho_b^h = \tilde{p}_b^{\phi_b} + e(\varsigma_{\phi_b}, \tilde{X}) \times w(\varsigma_{\phi_b}, \tilde{X}) \Rightarrow e(\varsigma_{\phi_b}, \tilde{X}) < e(\varsigma_{\phi_b}, \tilde{X})
  \]

  It means that the candidate $\varsigma_{\phi_b}$ is less competitive than her weakest associated competitor $\varsigma_{\phi_b}$. It means that the lists of best efficiency candidates $L$ and $\tilde{L}$ would contain the same candidates at least until the allocation iteration $z \leq MN$ of the competitor $\varsigma_{\phi_b}$ as follows:

  \[
  \bigcup_{i=1}^z L_i = \bigcup_{i=1}^z \tilde{L}_i \Rightarrow X_z = \tilde{X}_z
  \]
As a result, the corresponding allocation decisions would be the same until the allocation iteration $z \leq MN$. We know that when the candidate $\varsigma_{sb}$ is allocated in the list $L$, the competition conditions (i.e. the seller’s bundle-reservation price and the resource scarcity that affects the bundle weight) for $\beta_b$ are the best to win. Therefore, if $e(\varsigma_{sb}, X) < e(\varsigma_{sb}, \bar{X})$, any candidate associated with $\beta_b$ would become the best efficiency candidate after $\varsigma_{sb}$ in the list $L$. However, we know that there is not enough available resource to satisfy the buyer $\beta_b$ at the position $\bar{z} > z$ in the list $L$, and the initial allocation would be feasible, if $\rho_{ls}^b \leq \bar{p}_{ls}^n$.

Hence, the proposed pricing mechanism is IR for single-minded buyers.

\[ \square \]

**Theorem 15.** The proposed pricing scheme maintains Individual Rationality property for the sellers’ declared types.

**Proof.** We need to show that seller IR property holds as defined in Formula 3.5. We know that the payment of each individual buyer is always at least the seller’s desired bundle-reservation price for the allocated resources $\rho_{ls}^b = \max \left( \sum_{\phi \in S} \hat{p}_{\phi b}^r x_{\phi b}, \min(\hat{p}_{\phi c}^c) \right)$ as ensured in buyer pricing mechanism.

Therefore, in order to prove seller IR property, we need to show that the market surplus distribution is always positive, as follows:

\[ \Delta \left( \frac{\Omega_s}{\Omega} + (1 - \mu) \frac{\Delta_s}{\Delta} \right) \geq 0 \]

We know that the buyer requested bundles contain only positive amounts of resources $\hat{c}_{bg}^d \geq 0, \forall g \in G, \forall b \in B$ and the amounts of available resources cannot be negative $\hat{c}_{sg}^{sa} \geq 0, \forall g \in G$ (as ensured by ARC), we can conclude the following:

\[
\begin{align*}
& f_g^s(\hat{c}_{sg}^s, X) > 0, \forall g \in G, \forall s \in S \\
& f_s^n(\hat{c}_{s}^s, X) > 0, \forall s \in S \\
& l \geq 0, m \geq 0
\end{align*}
\]

\[ \Rightarrow w(\beta_b, \alpha_s, X) > 0 \Rightarrow \frac{\Omega_s}{\Omega} \geq 0 \]

Similarly, given the constraints on the problem variables, we can conclude the following:

\[
\begin{align*}
& \rho_{ls}^b \geq \hat{p}_{ls}^r x_{\phi b}, \forall s \in S, \forall b \in B \\
& \hat{c}_{bg}^d \geq 0, \forall g \in G, \forall b \in B \\
& \hat{p}_{sg}^c \geq 0, \forall g \in G, \forall s \in S
\end{align*}
\]

\[ \Rightarrow \frac{\Delta_s}{\Delta} \geq 0 \]
Consequently, given that \(0 \leq \mu \leq 1\), we get that surplus distribution is always positive. Hence, the seller payment is always \(\rho_s^b \geq \sum_{b \in B} \tilde{p}_{x_{sb}}^{br}, \forall s \in S\).

Hence, the proposed pricing mechanism is IR for the sellers’ declared types.

\[\square\]

**Budget Balance (BB)**

**Theorem 16.** The proposed market mechanism maintains Budget Balance property.

**Proof.** We need to show that BB property holds as defined in Formula 3.6. To prove the required equivalence of buyer prices and seller payments, we replace the seller pricing by the definition, and transform the right side of the equation, as follows:

\[
\sum_{s \in S} \rho_s^b = \sum_{s \in S} \left( \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} + \Delta \left( \frac{\Omega_s}{\Omega} + \frac{(1 - \mu) \frac{\Delta_s}{\Delta}} \right) \right) = \\
\sum_{s \in S} \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} + \Delta \sum_{s \in S} \frac{\Omega_s}{\Omega} + \Delta (1 - \mu) \frac{\sum_{s \in S} \Delta_s}{\Delta}
\]

Given the definitions of the market and individual seller volumes and surplus, we can derive the following:

\[
\Omega = \sum_{s \in S} \sum_{b \in B} w_{sb} x_{sb} \\
\Omega_s = \sum_{b \in B} w_{sb} x_{sb} \Rightarrow \Omega = \sum_{s \in S} \Omega_s
\]

\[
\Delta = \sum_{b \in B} \rho_b^b - \sum_{s \in S} \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} \\
\Delta_s = \sum_{b \in B} (\rho_b^b - \tilde{p}_{x_{sb}}^{br}) x_{sb} \Rightarrow \Delta = \sum_{s \in S} \Delta_s
\]

Consequently, we can reduce the fractions corresponding to the volumes and surpluses, and replace the generated surplus \(\Delta\) by its definition. As a result, we get the following:

\[
\sum_{s \in S} \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} + \Delta = \sum_{s \in S} \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} + \sum_{b \in B} \rho_b^b - \sum_{s \in S} \sum_{b \in B} \tilde{p}_{x_{sb}}^{br} x_{sb} = \sum_{b \in B} \rho_b^b
\]

Hence, the proposed mechanism maintains BB property. \(\square\)

**Computational Tractability (CT)**

The proposed HE-DS market mechanism for trading heterogeneous infrastructure cloud services determines allocation and pricing decisions in polynomial time. In particular, the
iterative HE-DS-A mechanism solves the allocation problem in $O(|M^2N^2|)$ time, where $MN$ number of allocation candidates are considered for allocation in $MN$ market iterations. The HE-DS-PB mechanism determines the payments for $N$ buyers by iteratively searching for the competitor minimum prices, which results in $O(|M^2N^3|)$ time complexity. The corresponding HE-DS-PS scheme determines seller payments in linear time $O(|M|)$. Therefore, the market mechanism’s time complexity is $O(|M^2N^3|)$. We can see that the major computation is spent on buyer payment determination. However, in order to reduce the computation time required for large problem instances, a binary search could be used in order to quickly approximate the critical-value buyer payment. Another approach could be to reduce the number of allocation iterations and derive allocations based on partially sorted lists of candidates, which can allow to reduce the mechanism’s time complexity to $O(|M^2N^3/T|)$, where $1 \leq T \leq MN$.

5.2.4 Experimental Investigation

Experimental Setting

The experimental setup used in this study is summarised in Table 5.11. The dynamics of the proposed market mechanism are investigated in an extensive simulation setting with various market scenarios.

**Market Model:** We consider a double-sided market, populated by multiple buyers and sellers. The market allows trading of resources of three different types $K = 3$, including CPU, RAM and HDD.

**Demand Model:** In this experiment, we simulate different numbers of buyer bids $N$, as a number of allocation candidates $C$ per each participating seller $M$. In this way we can control the number of allocation candidates in the market scenarios with different numbers of sellers. In order to reflect variability for continuous spaces, we employ uniform distribution functions for modelling the buyer bids. In particular, the requested bundles of resources are drawn from uniform distribution with ranges derived based on real-world cloud provider offerings (e.g. Amazon EC2 [6], Rackspace Cloud [116]). The price valuations $p^i_b$ consist of randomly generated price valuations for each individual resource type $p^i_{bg}, \forall g \in G$ and scaled according to the amounts of the resource in the bundle. The selected maximum limits used in our distribution functions for buyer price valuations are based on price offers from commercial cloud providers, such as AT&T Cloud [3] and Verizo/Terremark Cloud [143].

**Supply Model:** We simulate various numbers of sellers $M$, participating in the mar-
Table 5.11: HE-DS Experimental Setting

<table>
<thead>
<tr>
<th><strong>Market Model:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Types</td>
<td>$G = {1: \text{CPU}, 2: \text{RAM}, 3: \text{HDD}}$</td>
</tr>
<tr>
<td>$K = 3$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Demand Model:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Buyers</td>
<td>$N = C/M$, where</td>
</tr>
<tr>
<td>$C = {32, 64, 128, 256, 512}$</td>
<td></td>
</tr>
<tr>
<td>Demanded Capacity</td>
<td>$c^d_{b1} = U(1, 32)$</td>
</tr>
<tr>
<td></td>
<td>$c^d_{b2} = U(1, 240)$</td>
</tr>
<tr>
<td></td>
<td>$c^d_{b3} = U(50, 1200)$</td>
</tr>
<tr>
<td>Demanded Capacity</td>
<td>$\sum_{g \in G} p^v_{bg}$, where</td>
</tr>
<tr>
<td></td>
<td>$p^v_{b1} = U(0.0, 0.1)$</td>
</tr>
<tr>
<td></td>
<td>$p^v_{b2} = U(0.0, 0.08)$</td>
</tr>
<tr>
<td></td>
<td>$p^v_{b3} = U(0.0, 0.0005)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Supply Model:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sellers</td>
<td>$M = {1, 2, 4, 8, 16}$</td>
</tr>
<tr>
<td>$c^*<em>{sg} = \left( \sum</em>{b \in B} c^d_{bg} \right) \times p_g \times |F^p|_s$, $\forall g \in G$</td>
<td></td>
</tr>
<tr>
<td>Supplied Capacity</td>
<td>$p_g = {0.25, 0.5, 0.75, 1.0}$</td>
</tr>
<tr>
<td></td>
<td>$F^p = (f^p_1, \ldots, f^p_M)$, where $f^p_s = U(0.0, 1.0)$</td>
</tr>
<tr>
<td>Price Reservation</td>
<td>$p^r_{s1} = U(0.0, 0.1) \times z$</td>
</tr>
<tr>
<td></td>
<td>$p^r_{s2} = U(0.0, 0.08) \times z$</td>
</tr>
<tr>
<td></td>
<td>$p^r_{s3} = U(0.0, 0.0005) \times z$</td>
</tr>
<tr>
<td></td>
<td>$z = {0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5}$</td>
</tr>
<tr>
<td>Seller Misreporting</td>
<td>$q_g = {0.5, 0.75, 1.0, 1.25, 1.5}$, $\forall g \in G$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Metrics and Mechanisms:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics</td>
<td>Social Welfare, Resource Utilisation, Total Cost, Seller Utility, Computation Time, Buyer Fulfilment</td>
</tr>
<tr>
<td>Allocation Mechanisms</td>
<td>HE-DS where $m, l = {0.0, 0.25, 0.5, 1, 2, 4}$</td>
</tr>
<tr>
<td></td>
<td>Optimal CPLEX solver</td>
</tr>
<tr>
<td>Pricing Mechanisms</td>
<td>HE-DS where $\mu = {0.0, 0.25, 0.5, 0.75, 1.0}$</td>
</tr>
<tr>
<td></td>
<td>K-pricing where $k = {0.0, 0.25, 0.5, 0.75, 1.0}$</td>
</tr>
</tbody>
</table>
ket. The goal is to experiment with different market provisioning and price reservation scenarios. Therefore, we balance the market generated supply against the previously generated market demand by adjusting the parameter $p_g$. The amount of resources offered by each participating seller is determined by a random portion of the total market’s supply defined as follows:

$$||F^p||_s := \frac{f^p_s}{\sum_{s' \in S} f^p_{s'}}$$

The vector $F^p$ contains randomly generated numbers from 0 to 1, and $||F^p||_s$ is its normalising factor to sum to unity. The seller’s price reservations are derived based on the same uniform distribution functions as the buyers’ price valuations for individual resources. The random value is scaled according to the price comparative ratio $z$, which allows us to experiment with different price reservation scenarios.

Overall, we simulate 2240000 market inputs, where 11200 market scenarios are repeated 200 times each. Each generated market is solved by 36 different parametrisations of our proposed allocation mechanism and by the optimal solver. In the buyer pricing experiment, the actual individually rational prices are compared with the critical-value truthful prices. We simulate $5^3 = 125$ seller misreporting scenarios in our second pricing experiment, where five levels of misreporting are applied to each resource type. The markets are cleared by five different parametrisations of our proposed HE-DS market mechanism and five parametrisations of well-known K-pricing scheme [124]. The basic statistics for the measured metrics are calculated based on the results obtained in 200 trials, in order to avoid statistical noise.

### Allocation Mechanism

We investigate the allocation quality achieved by our proposed allocation scheme and identify the mechanism’s parametrisations that allow achieving various objectives in the market, such as better social welfare, resource utilisation or buyer fulfilment.

### Closeness to Optimality

Only some limited market scenarios could be solved by the optimal solver within the feasible time due to the high problem complexity. Figure 5.8 illustrates the closeness to optimality of our proposed approximation mechanism (one of the most efficient parametrisations ($l = 2.0, m = 0.25$)) in terms of social welfare and average resource utilisation.

The number of feasible allocation candidates is the main factor that affects the allo-
Figure 5.8: Closeness to Optimality in terms of Social Welfare (a, c) and Average Resource Utilisation (b, d). Market Scenario for $l = 2.0, m = 0.25$. 

cation quality, because when it is high, the chance that the approximation mechanism denies a candidate which is a part of optimal solution is greater (Figure 5.8a). In the considered example, the social welfare is more optimal when there are more sellers in the market ($M = 16$) because the number of feasible allocation candidates is lower due to the reduced size of each seller’s offer (violated ARC), compared to the generated bids. Similarly, in Figure 5.8c, we can see that our mechanism is more efficient when the price comparative ratio is high ($z \geq 0.75$), because the number of feasible allocation candidates is reduced due to the increased sellers’ price reservations. The resource provisioning level has an insignificant impact on the mechanism’s approximation quality. However, when the resource in the market is not scarce, i.e. $p > 1.0$, our approximation mechanism essentially allocates all the participating buyers. A little deviation from optimal outcome can be due to some buyers being allocated not to the same seller as in the optimal solution. Overall, we observe a near-optimal allocation outcomes, where the worst social welfare approximation, obtained in our experiments, was 90%, observed in the market scenario with high competition for resources ($z = 0.0, p = 0.25$).
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

As in the experiments discussed in the previous sections, the optimality of the average resource utilisation has an identical pattern to the social welfare optimality (Figures 5.8b, 5.8d) because of the resource availability constraints that limit the approximation mechanism.

**Mechanism’s Parametrisation**

Since the optimal allocation result is unknown in the majority of the generated market

Figure 5.9: Closeness to the Best Approximation Result in terms of Social Welfare (a, b, e, f, i, j) and Average Resource Utilisation (c, d, g, h, k, l). Representative Market Scenario: \( C = 512, M = 4, z = 0.0, p = 0.25 \)
scenarios, we compare the allocation outcomes achieved by various parametrisations of our approximation mechanism among themselves (100% indicates the best approximation result). Figure 5.9 depicts the closeness to the best approximation result in terms of social welfare and average resource utilisation, obtained in a representative market scenario with $M = 4$, $C = 512(N = 128)$ and a high competition for resources $z = 0.0, p = 0.25$.

The social welfare and the average resource utilisation have conflicting dynamics, where the best solutions are typically achieved at the intersecting peak for social welfare and resource utilisation ($m = 1/l$). We can observe the opposite dynamics because while the higher values of parameters $l$ and $m$ make the mechanism give a higher importance to the exchanged bundle of resources in a candidate, the lower values make the mechanism rely more on the surplus that can be potentially contributed by a candidate. However, relying only on the candidate’s surplus (parametrisations $(l = 0, m = 0)$) is not the best approach neither. The most socially-efficient allocation decisions are made when there is a certain balance between the two parameters (e.g. Figure 5.9m). In particular, when the importance of the exchanged bundle-balance is emphasised by higher values of parameter $l \geq 2.0$ and its negative side-effect is counter-balanced by adjusting the value of parameter $m = 1/l$. The best resource utilisation is typically achieved when $(l = 4.0, m = 4.0)$ (e.g. Figure 5.9l) with some exceptions in the market scenarios with a small number of allocation candidates (Figure 5.9g) and with relatively small seller offers (Figure 5.9p). In such cases, smaller values of both parameters make the mechanism favour larger requested bundle, which results in a better resource utilisation.

**Resource Utilisation**

A snapshot of resource utilisation achieved by various parametrisations of our proposed

![Resource Utilisation Chart](image)

Figure 5.10: Resource Utilisation of various Allocation Mechanisms achieved across all market scenarios with optimal result.
Chapter 5. Market Mechanisms for Heterogeneous Infrastructure Cloud Services

Figure 5.11: Computation Time of Market Allocation. Representative Market Scenario: $z = 0.25, p = 0.5$.

In this chapter, we discuss the computation time of the proposed HE-DS allocation scheme and the corresponding result of the optimal solver. We can see that the optimal mechanism was not able solve the hard problem instances, while our approximation mechanism has proven to be highly time-efficient. The hardest market instance with $C = 512$ candidates was cleared within 10 seconds.

Pricing Mechanism: Buyer Misreporting Opportunity

In this experiment, we investigate the strategic misreporting opportunity of buyers, participating in the market that clears based on the proposed HE-DS mechanism.

We compare the payments determined for all the simulated buyers in the market with the corresponding critical-value payments (the ones that guarantee truthful behaviour).
valuation (critical-value payment).

We have noticed that the number of feasible allocation candidates in the market is the major factor that results in an increased buyer misreporting opportunity. For example, with the increased number of candidates in the market, statistically, the chance of getting the bids to benefit is increased (Figure 5.12a). Similarly, when the number of sellers is large ($M = 16$), their generated offers are smaller (due to experimental setting) and the number of feasible allocation candidates is reduced (due to violated ARC), which results in a reduced misreporting opportunity. We should notice that when there is a single seller in the market, the mechanism ensures truthful dominant buyer strategy (Figure 5.12d).

The price comparative ratio has a small impact on the buyer’s misreporting opportunity, but when the price reservations are too high, there is less feasible candidates and the chance to benefit becomes smaller (Figure 5.12b). With increasing level of resource provisioning, the buyer misreporting opportunity increases because more payments rely on the sellers’ price reservations, since the buyer-side competition is lowered (Figure 5.12c). Overall, we observe a buyer near-truthful average performance in the proposed mechanism, with some accidental spikes for maximum misreporting opportunity, which allows a buyer to improve her utility up to 20%. We have to stress that these values are derived based on complete information about the bids and asks submitted to the market, which is available only to the auctioneer. Therefore, it is very hard for the buyer to calculate a good misreporting bid and the chance to improve the utility is very small.

**Pricing Mechanism: Seller Misreporting Opportunity**

In this experiment, we analyse the opportunity presented to the market sellers when applying various misreporting strategies. We investigate the performance of various parametrisations of our proposed HE-DS market mechanism in a wide range of simulated market scenarios and compare the results with K-pricing mechanism.

Figures 5.14 and 5.13 illustrate the seller’s utility achieved from misreporting strategy relative to her truthful utility (Y axis). The values above 1.0 indicate that the strategy was successful and the seller improved her utility, while the values below 1.0 suggest that the seller was penalised for misreporting. Various degrees of over and understating strategies, based on the mean square difference of the three price reservations from the truthful ones, are depicted over X axis.

**Seller Misreporting and Market Scenarios**

We plot the representative results obtained in various market scenarios by a mixed rule
parametrisation of our proposed mechanism when $\mu = 0.5$ in Figure 5.13.

- **Price Comparative Ratio $z$:** We show how the seller’s misreporting opportunity changes in the market scenarios with different price comparative ratios (Figures 5.13a, 5.13b and 5.13). Typically, when the average buyer prices are much higher than the average seller prices in the market, the final buyer prices are going to be much higher due to a strong buyer-side competition. As a result, the market will generate more surplus and its bigger portion can be drawn by the sellers who apply overstating misreporting strategy. An incentive to raise the prices up in the market will result in a market situation with higher $z$. However, as we can see in Figure 5.13b, the seller has no incentive to lie in such a market scenario.

- **Number of seller $M$ and buyer $N$:** The competition on both sides of the market play an essential role in seller’s misreporting opportunity. The closer the supply side is to an extreme case of monopoly market ($M = 1$), the greater the benefit that the seller can gain from misreporting (Figures 5.13d, 5.13e and 5.13f). However, with just four sellers, there is no incentive to be strategic. When the number of buyers is getting smaller, their simulated bundles become larger in comparison to the sellers’ offers;
hence, fewer sellers can satisfy them (violated ARC). It results in a more monopolistic market with just a few competitive sellers and an increased misreporting opportunity (Figures 5.13g, 5.13h and 5.13i).

- **Provisioning level \( p \):** In Figure 5.13j, we can observe that the sellers have some tiny incentive to overstate their price reservations in the market of very scarce resources, while the increasing resource availability results in no strategic incentive (Figures 5.13k and 5.13l). It happens because when the resources are scarce, the buyer-side competition and the determined buyer payments are higher. As a result, the seller can secure a payment by increasing the price reservations for resources.

**Seller Misreporting and Mechanism’s Parametrisation**

In Figure 5.14, we plot the seller strategic opportunity, presented by different parametrisations of HE-DS mechanism and K-pricing scheme.

**HE-DS mechanism:** In all simulated market scenarios, the understating strategy did not result in a better seller utility. Typically, in the markets of scarce resources, reducing the price reservations can only result in lower buyer payments due to the reduced demand-side competition. Consequently, the surplus that the seller secures can only become smaller. Overall, we can see a near-truthful performance achieved by the proposed HE-DS mechanism, where a seller has a very small incentive to apply overstating strategy, when the pricing mechanism relies more on the proportional-value payment rule when \( \mu = 1.0 \) (Figure 5.14d). In the markets with high demand-side competition, the seller is likely to sell a similar amount of resources for an increased reservation price. As a result, a similar portion of surplus is drawn by the seller in addition to the overstated prices that result in a higher total reservation price. The payment mechanism parametrisation, that relies more on a direct buyer payment rule, generates more truthful seller payments (Figure 5.14a); however, the proportional-value payment rule has a more severe punishment for misreporting (Figure 5.14c).

**K-pricing:** We can observe a very similar performance of K-pricing scheme as in experiments for homogeneous double-sided market mechanism (Section 4.2.4). The understating strategy allows the seller to benefit when \( k \geq 0.5 \), since the seller secures more than 50% of the price difference, and the lower price reservation allow the seller to be more competitive and to sell more resources (resource overprovisioning) and/or be allocated more competitive buyers (resource underprovisioning). Although the seller misreporting opportunity is negligible when \( k = 0.25 \), the buyer strategic misreporting opportunity is increased in such mechanism. It may result in harmful market dynamics and highly inefficient allocation.
Figure 5.13: Seller Misreporting Opportunity in various market scenarios. Initial Market Scenario: $\mu = 0.5, C = 256, M = 8, Z = 0.5, P = 1.0$
Figure 5.14: Seller Misreporting Opportunity in HE-DS Surplus Distribution (a, b, c, d) and K-pricing Mechanism (e, f, g, h) with different parametrisations. Market Scenario: $C = 256, M = 4, Z = 0.5, P = 1.0$
5.3 Summary

This chapter investigated the problem of heterogeneous infrastructure cloud services trading and proposed two market mechanism designs for single-sided and double-sided markets. The aspects of the proposed mechanism designs and the corresponding qualitative characteristics are summarised in Table 5.12.

The designed mechanism for a single-sided market setting applies a novel iterative combinatorial greedy allocation approach. The pricing mechanism, complementary to the proposed allocation scheme, derives the buyer critical-value payments in an iterative manner. The double-sided market mechanism design also applies an iterative combinatorial greedy allocation approach for best efficiency candidates (composed of the bid and ask pair) determination. The corresponding pricing mechanism derives buyer prices based on an iterative double-sided critical-value payment mechanism, which is a subject to the seller’s RPC. The seller payment is calculated via distribution of the generated market surplus based on various mixed rules.

The designed mechanisms both guarantee all the required mechanism design feasibility constraints, as demonstrated in our theoretical evaluation. Specifically, both designs ensure Individual Rationality of the participating buyers and sellers, guarantee Budget Balance and determine the market outcomes in polynomial time. Our theoretical analysis revealed that the HE-SS mechanism for a single-side market is Truthful and discourages the buyers to act strategically. We have also shown that the seller IR and the buyer Truthfulness cannot be achieved at the same time in our double-sided market mechanism design.

The conducted extensive simulation experiments demonstrate that the proposed mechanisms determine the allocation outcomes that are very close to the optimal solutions. The proposed novel iterative approach showed to be highly efficient and has a significant improvement over the conventional (one-shot) allocation approach. We have studied the buyer’s and seller’s strategic misreporting opportunity in the proposed HE-DS mechanism. The experimental results reveal near-truthful incentives of the market participants. In particular, computing an efficient lie is almost impossible for the buyers, while sellers have some very small incentives to apply overstating strategy in a very limited set of market scenarios. We provide a quick reference for the potential market operators in regards to the market parametrisation decisions in Table 5.12c.
### Table 5.12: Market Mechanisms for Heterogeneous Cloud Services: Summary

#### (a) Mechanism Design Aspects

<table>
<thead>
<tr>
<th>Market Mechanism</th>
<th>Allocation</th>
<th>Pricing</th>
</tr>
</thead>
<tbody>
<tr>
<td>HE-SS</td>
<td>Single-sided Iterative Combinatorial Greedy (best efficiency bids)</td>
<td>Iterative critical-value pricing (seller reserve price)</td>
</tr>
<tr>
<td>HE-DS</td>
<td>Double-sided Iterative Combinatorial Greedy (best efficiency candidates candidate: ask-bid couple)</td>
<td>Buyer pricing: Iterative double-sided critical-value (subject to seller RPC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seller Payment: Mixed Rule Surplus Distribution</td>
</tr>
</tbody>
</table>

#### (b) Mechanism Design Qualitative Characteristics

<table>
<thead>
<tr>
<th>Economic Properties</th>
<th>IR</th>
<th>BB</th>
<th>CT</th>
<th>AE</th>
<th>TB</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HE-SS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>n/a</td>
</tr>
<tr>
<td>HE-DS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

- ● - theoretically guaranteed economic property
- ● - approximated property (near-optimal / near-truthful)

#### (c) Mechanism Parametrization: Market Operators Decision Support

<table>
<thead>
<tr>
<th>Param. Value</th>
<th>Impact</th>
<th>Description / Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\langle l = 2.0, m = 0.5\rangle)</td>
<td>Favour better-balanced bids</td>
<td>Best Social Welfare</td>
</tr>
<tr>
<td>(\langle l = 4.0, m = 0.25\rangle)</td>
<td>Bundle-based allocation (little regard to valuation)</td>
<td>Best Resource Utilisation</td>
</tr>
<tr>
<td>(\langle l = 4.0, m = 4.0\rangle)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mu = 0.0)</td>
<td>Direct-value Payment</td>
<td>Seller overstating opportunity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Severe punishment for misreporting</td>
</tr>
<tr>
<td>(\mu = 0.5)</td>
<td>Mixed Pricing Rule</td>
<td>Little overstating opportunity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reasonable punishment for misreporting</td>
</tr>
<tr>
<td>(\mu = 1.0)</td>
<td>Proportional-value Payment</td>
<td>Marginal misreporting opportunity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Weak punishment for misreporting</td>
</tr>
</tbody>
</table>
Chapter 6

Smart Cloud Marketplace - A Platform for Trading Cloud Services

In this chapter, we design and develop a prototypical platform, called Smart Cloud Marketplace, for automated trading of cloud services based on intelligent agent technology. Firstly, we outline the important requirements for an open marketplace of infrastructure cloud services and discuss the modern technologies that can address them. We provide the reference architecture of the designed cloud marketplace platform and give some implementation details. Finally, we illustrate the practical use of the developed system by considering a number of use-case scenarios that illustrate the automation of infrastructure cloud services trading by using computational market mechanisms and intelligent agents.
Chapter 6. Smart Cloud Marketplace - A Platform for Trading Cloud Services

6.1 IaaS Marketplace Platform Requirements

The emerging cloud exchange platforms in industry [4, 139] focus on addressing the technical challenges, related to infrastructure cloud services provisioning, such as quality of service assurance, issues linked with virtualisation, contracts standardisation, management of service level agreements and other. However, an important next step in cloud marketplace evolution would be to allow an automated exchange of infrastructure cloud services in an open marketplace environment. At present, there is no open marketplace platform that allows automated trading of cloud services between multiple independent users. In order to provide an effective marketplace for infrastructure cloud services, we need to determine a set of important features and design requirements for such a platform. The aspects discussed in this section are based on our analysis of the infrastructure cloud services market and the proposed agent-based marketplaces, provided in Chapter 3.

- **Heterogeneous environment (human and software agents):** In recent years, we are witnessing a gradual shift from traditional electronic marketplaces to the more intelligent next-generation platforms that are increasingly reshaped by disruptive technologies, such as social computing, cognitive analytics, digital engagement, cloud computing and Internet of Things (IoT). The products in such markets are tailored to individual preferences and often need to be procured on the fly and based on rapidly changing requirements. It poses cognitive challenges for the human users and necessitates the tools for decision support and decision-making that automate, augment and coordinate some or all of the decision processes related to the services exchange. The growing acceptance of intelligent personal assistants and the important advances in the area of artificial intelligence and cognitive computing, is likely to lead to a semi or fully-automated electronic marketplace where some or all market-related activities are delegated to intelligent software agents. Therefore, we envision the next-generation marketplace of infrastructure cloud services as heterogeneous environment, where human users can trade by themselves or can delegate their market activities to the intelligent agents.

- **Trading strategies (embedded and custom):** The marketplace platform can offer some built-in intelligent software agents that facilitate the trading activities for the buyers and sellers. For example, a resource monitoring agent, similar to Amazon EC2 CloudWatch [10], or market price monitoring agent that makes the right adjustments by buying or selling services as needed. However, the cloud consumers and
providers have their individual requirements and objectives that have to be addressed. Therefore, a cloud marketplace has to provide an interface which allows the market participants to employ their custom trading agents of arbitrary complexity. It will allow expressiveness and freedom to the consumers in reaching mutually acceptable agreements over the services usage terms and conditions.

- **Market mechanisms:** The rapid development in the multidisciplinary research between computer science and economics has resulted in a large variety of proposed market mechanism designs that aim to facilitate the market trading process in the most effective and efficient way. Although, a universal mechanism is unlikely to be developed, some designs can make a better fit for specific purposes, and can be preferred over the others, depending on the scenario. While the mechanism designers aim at developing economically efficient and practical mechanisms, the actual choice of the best market mechanisms has to be done by the users themselves. Therefore, the platform has to allow a free choice of the market mechanisms used for trading. In this way, the most preferred mechanisms go through a natural selection as a result of the user experience in terms of practicality, fairness, and efficiency.

- **Multiple markets:** The simultaneous existence of multiple markets is an essential requirement for an open marketplace for trading infrastructure cloud services. First of all, multiple markets allow trading of different types of services (e.g. homogeneous/heterogeneous) in different types of markets (e.g. single-sided/double-sided), using various market mechanisms. Furthermore, both sellers and buyers can instantiate new markets for addressing their trading requirements and expressing their individual preferences. The ability to create multiple simultaneous markets permits the buyers and sellers to choose the preferred market mechanisms. For example, the buyers are not forced to join the market that operates disadvantaging K-pricing scheme. Instead, they would rather create a new market with a more fair auction design, such as HO-DS market mechanism. As a result, such a marketplace is a dynamic and open system that provides equal rights to both buyers and sellers.

- **Execution Environment (real-time trading and simulation testing):** Typically, the proposed platforms for electronic trading are developed to allow the real-time trading of goods and services. Such execution environment permits the users to buy and sell services remotely and in real-time. However, an important commonly overlooked feature of an open marketplace platform is a simulation environment. Simulation
environment is essential for researchers in order to evaluate the efficiency of the
designed market mechanisms as well as to analyse the performance of the developed
software agents.

Apart from the marketplace requirements listed above, there is a standard set of tech-
nical system quality attributes that need to be achieved by the platform design. Besides
allowing for multiple market mechanisms and custom trading strategies, the marketplace
has to ensure extensibility. In other words, the platform has to take the future growth
into consideration, and should be easily extendible by new market mechanism designs and
trading strategies. Given a dynamic nature of the cloud services and multiple consumers
and providers, the system must be scalable and should allow trading among a large number
of buyers and sellers. Apart from that, the marketplace platform has to be secure and fault
tolerant to ensure seamless performance with protection from harmful activities. Finally,
since the system allows direct interactions with human users, it has to ensure usability and
a smooth learning curve for its human participants.

6.2 Smart Cloud Marketplace Architecture

6.2.1 Facilitating Technology Choice

In order to address the aforementioned requirements for an open marketplace of infras-
tructure cloud services, we propose an agent-based system with a modular design and
web-based interaction interfaces.

We select an agent-based system design because it is capable of addressing our system
requirements in an effective and efficient way. It allows modelling multiple market mecha-
nisms and trading strategies as agents’ behaviours. Furthermore, the multi-agent systems
are proven to perform well in terms of reliability, extendibility, and security [148].

We use Java programming language for our system implementation because it is object-
oriented and allows creating modular programs which run based on platform-independent
byte-code. The communication model in the system is based on JSON objects which
ensures simplicity and extensibility of the system design. The web services provide a
single standardised point of access to the system, where a human user can interact via the
provided web interface, and the custom-designed personal software agents communicate
with the system via web services directly. In order to develop a user-friendly web-interface
of the system, we apply the fundamental usability principles [104] and get inspiration from
the commercial marketplaces, such as UCX [139] and DBCE [4].
The proposed modular design ensures extensibility and scalability. Each module can be deployed on a separate cloud instance, which scales up and down based on the changing need. In order to extend the system by new market mechanisms, the module with implemented market mechanisms (java library) has to be updated without any changes to the multi-agent system. The new trading strategies of the embedded trader agent can be added by developing new agent behaviours without any changes to the rest of the system.

6.2.2 Reference Architecture

The backend engine, which has been used for our simulation experiments in order to produce the results discussed in Chapters 4 and 5, has been extended in order to allow real-time cloud services trading. It is important to note that the simulator functionality of the system is still available for access via the provided application programming interface (API), while real-time trading functionality has been added to the system.

The reference architecture of the proposed Smart Cloud Marketplace platform is given in Figure 6.2. In this section, we omit the explanation of some common design functions, such as register, login, change password, and discuss the major aspects of the system design. We can see that the user can interact with the system in two different ways: (i) direct interaction via the web-based user interface (the system offers a manual trading option as well as can provide some simplistic embedded software agents) and (ii) by delegating the trading tasks to a custom trading agent of arbitrary design, which interacts with the agent-based environment via the web services. The actual platform consist of three major building modules, which are (i) the web interface for the user interaction, (ii) the multi-agent environment, where the trading decisions are facilitated by the market mechanisms available in the library, and (iii) the database for persistent storage of the marketplace activity and trading outcomes. We discuss some of the major system components below.

Web Interface

The developed prototype system can be accessed at http://smartcloudmarket.com/. The system is available on all devices from mobile phones to computer web-browsers. We provide the sample screenshots of mobile web-site version in Figure 6.1. The system’s web interface is split into two major zones: public and authorised. The public zone contains some promotional material and ground explanations about the proposed system. The authorised zone allows the registered users to interact with the system by browsing the existing markets, creating new markets for trading specific goods and services, joining
Chapter 6. Smart Cloud Marketplace - A Platform for Trading Cloud Services

(a) Public Zone Index Page

(b) Authorised Zone Main Page

Figure 6.1: Smart Cloud Marketplace Mobile: Sample Screenshots

the markets by submitting sell and buy orders/trade requests, monitoring the purchased resources, and running market simulation. A more detailed discussion about the possible operations allowed via the user interface is given in the next section, where we provide some illustrative use-case scenarios.

Multi-agent Environment

The Smart Cloud Marketplace core system consists of a number of agent types that facilitate the trading processes in the platform. The system is based on Java Agent DEvelopment Environment (JADE) [19], which is provided in order to simplify the implementation of multi-agent systems. The entire system is exposed to the users as a number of web-services that offer the major market functions, such as (i) creating markets, (ii) submitting and withdrawing trade requests, including buy and sell orders, (iii) retrieving the information about the markets and (iv) running various market simulations so that to make better-informed trading decisions in future. The web services provide a flexible design and allow the users to build and employ their custom trading agents in order to address their unique market objectives. We outline the roles of each agent type in our system and explain the system communication workflow.
Figure 6.2: Smart Cloud Marketplace: Reference Architecture

- **Participant Agent**: Each user, registered in the Smart Cloud Marketplace system, is represented by an individual participant agent that initiates all the market-related processes, and manages the corresponding trader agents of this user. The participant agent is responsible for (i) communication with the market manager agent, requesting the initiation of new markets, (ii) creation of new trader agents with corresponding behaviours and instructions, as well as (iii) market information retrieval, such as the currently active markets or the past market prices.
• **Trader Agent:** This agent is provided by the marketplace platform in order to perform some simplistic trading behaviours. The trader agent interacts with the market agent and shares the information related to the sell and buy orders. In its trivial form, this agent would receive the exact instructions in regards to the bid/ask to be declared to a specific market agent. Some more intelligent embedded trader agents can construct the market requests based on the user preferences and objectives. For example, the user can specify the type of application to be outsourced to the cloud, and the trader agent would select the most appropriate VM configuration for this purpose. Regardless of the selected behaviour, all the trade requests, submitted to the market, are sent by the embedded trader agent. If human principal employs a custom trader agent, which makes the decisions and submits the trade request to the platform, the trivial trader agent will simply follow the specified instructions. The trader agents are created for the time of the trade and die once they fulfil their mission.

• **Market Manager Agent:** It is a unique agent and there is a single instance running in the system. This agent is responsible for managing the market agents and the related market information in the platform. Market manager agent is responsible for (i) creation of market agents upon eligible request of the participant agents and (ii) communication with the persistence agent in order to keep a record of the market status (e.g. active, closed, planned), the information about supply, demand, and allocation and pricing decisions.

• **Market Agent:** This agent represents an individual market for trading some specific goods or services. The market agent is created upon the market manager agent request, and it has its own market anatomy in terms of market type (e.g. single-sided, double-sided), the type of traded service (e.g. homogeneous, heterogeneous), as well as the market mechanism (selected behaviour). The communication model of the market agent is defined based on JSON objects, and the Java library of the implemented market mechanisms is used as an engine for making decisions and clearing the market. Due to such a generic and modular design, it is really easy to extend the platform with new mechanisms, which requires no change to the actual multi-agent environment. The market agent reports about any changes in supply, demand and the market decisions to the market manager agent.

• **Persistence Agent:** These is a single entity of persistence agent running the the
environment. This agent is responsible for all the database-related operations, including the data retrieval and storage. Persistence agent is listening to the requests from participant agent and market manager agent and acts accordingly by providing the required information from the database or saving the market-related information in persistent storage.

### 6.3 Use-case Scenarios & Proof of Concept

The Smart Cloud Marketplace system allows trading of two different types of assets, including hardware resources (e.g. CPU, RAM, HDD) and virtual machines (pre-defined configurations). Regardless of the traded asset type, the system allows on-demand and future exchange. On-demand trading is equivalent to a continuous auction, which clears at the moment the request (i.e. bid, ask) arrives. Such a market trades the resources in present time, and updates the allocation and pricing decisions once a change in supply and demand occurs (a new trade request arrives or an old request is withdrawn). Futures market, analogous to periodic auction, aims to determine allocation and pricing decisions for the resources traded in future (e.g. next day). The clearance in such a market happens only once when the market closes. In both cases, the market-based mechanisms, proposed in this thesis, are used in order to determine the allocation and pricing outcomes, but other mechanisms can be added in future. Please note that our proof-of-concept prototype system is currently not attached to the real cloud resources.

In this section, we demonstrate the developed prototype of Smart Cloud Marketplace system by providing a set of illustrative use-case scenarios. In particular, we describe a buyer procurement scenario and a seller trading scenario in order to provide the supply and demand side perspectives of the trading process in Smart Cloud Marketplace.

#### 6.3.1 Buyer Procurement Scenario

We imagine that the Smart Cloud Marketplace system begins to attract commercial customers. We assume that there has already been a couple of cloud providers who joined the marketplace, and more buyers come to procure cloud infrastructure in SCM. Our Swinburne Intelligent Agent Technology group is one of the active buyers in the marketplace, procuring resources for research purposes.

Smart Cloud Broker [33, 32, 31, 29, 30] (http://smartcloudbroker.com/) is one of our highly successful projects, which allows to schedule the automated benchmarking process on different cloud platforms in order to get an insight into the performance of various virtual
machine configurations. In order to conduct a new set of software benchmarking tests, our group needs a cloud infrastructure composed of different VM configurations. Therefore, we decide to trade the virtual machines using the Smart Cloud Marketplace system. Our benchmarking tests are state independent and can run in an automated fashion. Therefore, we can buy resources on-demand during the time periods when the market-based prices satisfy our budget constraint. We search for the markets that trade general purpose virtual machines of small, medium and large sizes. In Figure 6.3a, we can see a single market for trading Virtual Machine instances on-demand, called "On-demand VM Exchange". There are two sellers, namely Amazon EC2 and Rackspace Cloud, currently offering their services in this market. We browse the market details, and see that it satisfies our requirements in terms of offered VM configurations (Figure 6.3b); thus, our trade request can be submitted in order to buy a bundle of VMs (Figure 6.3c).

During the working hours, there has been some buying and selling activity in the market (this activity was generated by us for illustration purposes). At the end of the day, we browse the trading outcomes for our IAT Group request in "Contracts" section (Figure 6.3d). This window visualises the amounts of bought resources and allows us to monitor the corresponding contracts and resource prices. We can see that in the early morning (from 8:15 am to 9 am), the requested cloud infrastructure was provisioned according to our requirements with the price of $0.83. At 9 am, the price of the provisioned resource increased up to $2.645 due to the increased competition for resources. During the period between 10 am to 5 pm the traded VMs were allocated to some other more competitive users. Finally, at the end of the working day (5 pm) and until the market closing time (8 pm), the requested infrastructure was provisioned to us again.

In this example, we engaged in a manual trade, where the resource preferences were known and the trading decisions were made by an expert. However, the market-related activity in Smart Cloud Marketplace can be automated or facilitated for ordinary users by means of custom-designed intelligent agents. For our proof of concept purpose, we have developed a simple recommender system to assist the buyer in defining the appropriate VM configurations based on the high level user preferences (Figure 6.4). The software agent prompts the buyer to reveal the hosting purpose and the type of workload to be handled by a VM. Based on this information, the agent suggests the most appropriate class of VMs to be used as well as the most common VM configurations in this class (Figure 6.4b). Once the buyer has finalised the cloud infrastructure requirements, the recommender will search the SCM markets that can address the buyer’s request (Figure 6.4a). Finally, the software
Chapter 6. Smart Cloud Marketplace - A Platform for Trading Cloud Services

(a) On-demand Markets for Virtual Machines Trading

(b) Market Information and Details

(c) Buy Request Submission

(d) Monitoring Contracts for Virtual Machines

Figure 6.3: Smart Cloud Marketplace: Buyer Procurement Scenario
agent will schedule the buy request submission to the selected market for the required period. It is an example of a simple agent that can facilitate the buyer market trading. Some really sophisticated agents can perform synchronised simultaneous trading in several markets aiming to achieve various buyer objectives. Design of such truly intelligent agents is out of scope of this thesis and can be considered in future work.

6.3.2 Seller Trading Scenario

CloudSigma, a global cloud provider, is currently selling the infrastructure cloud services to the consumers by applying a usage-based and subscription pricing models. Fascinated by the growing discussions and benefits of market-based cloud services trading, their strategic department decides to use Smart Cloud Marketplace to have an additional sales channel for their hardware resources.

The CloudSigma administrator registers in the system with some verified capacities of owned cloud resources. CloudSigma’s objective is to maximise the amount of resources consumed by the market buyers at any given time. Therefore, the administrator applies the following strategy: the resources are initially traded in futures market for the next day usage (which provides better guarantees for the cloud provider), and the remaining unsold capacity is provisioned in an on-demand market.

In order to trade the hardware resources for the next day usage, CloudSigma has to either join an existing market or create a new one. Since there is no suitable market in the system, CloudSigma creates a new futures market in order to address the specific
Chapter 6. Smart Cloud Marketplace - A Platform for Trading Cloud Services

trading requirements (Figure 6.5a). In particular, CloudSigma would like to trade the cloud infrastructure resources for the next working day (9 am - 5 pm). The market outcomes have to be known by the evening today (9 pm), so that CloudSigma has some reasonable time to make further trading decisions based on the outcomes. Afterwards, CloudSigma submits its offer to the newly created market, where all the available capacity is attempted to be sold.

Since CloudSigma has just joined the market, all the available resources owned in current time will be traded in an on-demand market. An on-demand exchange market for trading hardware resources, called "On-demand Hardware Exchange", currently exists in Smart Cloud Marketplace. ElasticHosts is the only seller in this market at the moment, meaning that the competition is relatively small and CloudSigma can try its chances in this market. Therefore, CloudSigma’s administrator submits a new offer for all of the available resources with corresponding prices (Figure 6.5b). The trade results are known immediately and can be viewed on "Hardware Contracts" page.

In Figure 6.5c, we show the contracts of ElasticHosts and the fluctuation in price overtime. We can see that due to the buyer’s activity in the market, the amounts of sold resources and the corresponding prices of ElasticHosts were changing every 15 minutes. We can observe a steady rise in price from $7 at 10 am up to $16 by 11:15 am. It can be clearly seen that at 12 am, the contract price of ElasticHosts has dropped to $11. It happened due to a new market offer from CloudSigma, which resulted in a higher resource availability in the market and lower buyer prices. This interface allows the market sellers and buyers to analyse the market outcomes and adapt their trade requests based on the changing market conditions. For example, if the seller’s reserve price is too high to compete with other sellers, the current offer can be withdrawn and replaced by a new one with lower, more competitive prices.

At the end of the day, the administrator checks the results of trading in Smart Cloud Marketplace. The seller can observe that most of the on-demand resource was consumed during the day, and around 60% of resource was sold for the next day usage (Figure 6.5d). This resource monitor provides an aggregated view of cloud provider’s resource consumption, and allows for an easy analysis of trading outcomes and better trading decision in future.

Apart from the provided Smart Cloud Marketplace web interface, the sellers can design and use their personal software agents. A design of an intelligent software agent for seller’s trading in Smart Cloud Marketplace is a complex research problem, which involves some
Chapter 6. Smart Cloud Marketplace - A Platform for Trading Cloud Services

(a) Create Market Summary

(b) Sell Request Submission

(c) On-demand Hardware Contracts Monitoring

(d) Cloud Provider Resource Monitoring

Figure 6.5: Smart Cloud Marketplace: Seller Trading Scenario
sophisticated scheduling mechanisms design and application of machine learning techniques
for price prediction and best trading strategy selection.

6.4 Summary

This chapter provided the proof-of-concept implementation of Smart Cloud Marketplace -
a system for infrastructure cloud services trading based on computational market mech-
anisms, proposed in this thesis. We have identified a list of requirements for an open
marketplace platform for trading cloud services. In our vision a truly competitive open
marketplace has to allow a simultaneous existence of multiple markets for resource trading
and provide a free selection of the market mechanisms for their operation. Furthermore, the
marketplace environment has to be heterogeneous and allow both human and intelligent
software agents to engage in market activities. Therefore, we have developed a functional
software engine that operates based on intelligent agents and provides web interface for
human interaction with the marketplace. The architecture of the system was briefly out-
lined in this chapter, where we motivated the technology choice and described our designed
multi-agent environment.

Finally, we provided a number of use-case scenarios, where the designed market mech-
anisms were used for real-time and simulation trading of infrastructure cloud services. We
have shown that the system can be used by both cloud providers and consumers and the
market-based allocation and pricing mechanisms allow both buyers and sellers to trade
services based on laws of supply and demand. The developed system allows for an easy
monitoring of possessed, reserved, and sold cloud resources, as well as analysis of contracts
and market price fluctuation overtime (Figure 6.5c). Furthermore, we have demonstrated
that the trading activities in Smart Cloud Marketplace can be facilitated or automated by
the means of custom-designed intelligent software agents. Finally, the developed system
still allows the researchers to conduct custom simulation experiments by using the provided
application programming interface (API). We conclude that Smart Cloud Marketplace is a
proof-of-concept platform with a big potential and can evolve into a commercial system for
infrastructure cloud services exchange as well as can be used by researchers as a simulation
tool for market mechanisms evaluation.
Chapter 7

Conclusions and Future Directions

This chapter makes the final conclusions to the work presented in this thesis. We summarise the main research contributions in regards to the research questions from Chapter 1. Afterwards, we discuss the potential improvements and extensions of the presented work, and outline the future research to be undertaken.

7.1 Summary of Contributions

The major objective of this thesis has been to investigate the problem of infrastructure cloud services provisioning and acquisition in an open marketplace. The aim of this work has been to come up with the effective and efficient computational market mechanism designs that are both practical and theoretically sound. Typically, in order to develop the economically efficient market mechanism, the considered problem settings rely on multiple simplifying assumptions that are often too abstract to be applicable for solving real-world problems. On the other hand, practical problem settings can be very complex for designing and analysing theoretically robust market mechanisms. Therefore, in this thesis we have attempted to achieve the best possible compromise between an effective and efficient mechanism design, and we have made the following related contributions:

- General Problem Definition: We have conducted a systematic analysis of the infrastructure cloud services market. The analysis has revealed the common characteristics of the infrastructure cloud services, offered in industry, including homogeneous and heterogeneous VMs with pre-defined and custom configurations, as well as the types of typically used market models, such as single-sided and double-sided. Based on these outcomes, we have defined a conceptual framework and formulated a general problem of infrastructure cloud service provisioning. Afterwards, the problem has
been split into a set of specific services trading scenarios to be addressed.

- **Market Mechanism Design**: We have investigated each of the defined scenarios and proposed four economics-inspired computational mechanisms to address the problems of market provisioning of infrastructure cloud services. Overall, the proposed market mechanisms either apply the novel techniques that significantly improve over the existing methods, proposed in research literature (i.e. iterative combinatorial greedy allocation, buyer critical-value pricing in a double-sided combinatorial setting) or extend the existing mechanisms, aiming to provide a better fit for the considered problem setting (i.e. seller reserve price constraint, proportional-value payment rule for combinatorial markets with multiple resource types).

- **Mechanism Analysis and Evaluation**: As a result, we have tackled the problem from two perspectives: practical and theoretical, and have achieved a reasonable accord between the two perspectives of mechanism design. In particular, we have evaluated the design mechanisms in extensive numerical simulation experiments, we have conducted a theoretical analysis of the essential economic market design properties, as well as have validated our mechanisms in a proof-of-concept prototype of an electronic marketplace. From practical perspective, our mechanism design problem has accounted for the specifics of the cloud computing infrastructure environments, such as the services characteristics and commonly used economic market models. From the theoretical perspective, we have applied the existing knowledge in economics and auction design to the real-world cloud infrastructure provisioning models.

We provide a juxtaposition of the proposed mechanisms with related work below:

- **Homogeneous Single-sided Market Mechanism (Section 4.1)**: The presented mechanism extends the work of Zaman et al. [155] by considering a more realistic problem setting. In particular, our mechanism allows the seller to express the minimum desired prices for the traded services of different types (i.e. reservation prices). Both greedy allocation heuristic and the critical-value pricing preserve the important economic properties, such as Budget Balance, Individual Rationality and Truthfulness, unlike the approaches from the relevant literature [119, 75].

- **Homogeneous Double-sided Market Mechanism (Section 4.2)**: To the best of our knowledge, the proposed market mechanism is the first to address the problem of cloud services allocation and pricing in a combinatorial double-sided market with
indivisible buyer requests. Unlike the work of Stober et al. [132] where the authors use central scarce resource pricing, we address the problem where the market prices are derived based on competition for services of multiple different types. In our work, we extend the buyer critical-value pricing and seller proportional-value payment rule, proposed by Stober et al. [132], to operate in combinatorial markets with multiple resource types, which is a more robust mechanism. Furthermore, the proposed mechanism is superior to the following related works [92, 153, 59] since it maintains Budget Balance, Individual Rationality, Buyer Truthfulness and Computational Tractability in a double-sided combinatorial setting.

- **Heterogeneous Single-sided Market Mechanism (Section 5.1):** The proposed market mechanism is based on a novel iterative combinatorial greedy allocation heuristic accompanied by a corresponding iterative critical-value pricing mechanism. The mechanism has shown to be more allocative efficient in an extensive experimental setting compared to the approach proposed by Nejad et al. [102]. Furthermore, the proposed mechanism is Budget Balanced, Individually Rational and Truthful, unlike the relevant works in literature [119, 75].

- **Heterogeneous Double-sided Market Mechanism (Section 5.2):** As far as we are concerned, the problem of cloud services allocation and pricing has not been addressed before for the combinatorial double-sided market of heterogeneous services. We designed a buyer critical-value pricing for a double-sided combinatorial setting, which aims to attain the most truthful pricing outcomes possible. By contract to the relevant literature, our work focusses on a more realistic cloud market scenario, where the buyer requests are indivisible (unlike divisible goods in [92]) and the pricing is based on multiple different resource types (unlike central scarce resource pricing in [132]). Furthermore, we design the market mechanism that is Computationally Tractable (unlike the mechanisms in [59, 92]). Although, our theoretical investigation uncovers the cases when Buyer Truthfulness is not guaranteed by the proposed critical-value pricing, the misreporting opportunity is marginal, as revealed in our simulation experiments.

### 7.2 Answers to Research Questions

The considered research questions, presented in Chapter 1, can be answered based on the results of the work presented in this thesis. We provide concise responses to each of the
investigated research questions below:

**Research Question 1:**

Can we achieve the desirable economic properties, including allocative efficiency, individual rationality, budget balance, and incentive compatibility, in a market for trading infrastructure cloud services?

In order to answer this research question, we have defined a conceptual framework and formalised the problem of infrastructure cloud services trading. To ensure practicality of the considered infrastructure cloud services trading problem, its formulation has been based on the results of our market analysis (Chapter 2). This formulation has taken into consideration the classification of cloud services and the common market types for trading. As a result, the infrastructure cloud services provisioning problem has been modelled as a multi-unit combinatorial auction (Chapter 3). The economic literature that considers similar types of problems reveal that all four economic properties are unachievable in a single mechanism design [100], and some compromise should be made to ensure the best design possible [18][65]. Therefore, we have classified the mechanism design economic properties into feasibility constraints, essential for a practical design, and the desirable properties which have to be approximated, if cannot be guaranteed.

**Research Question 2:**

Can these economic properties in such a cloud market be achieved within tractable computational time?

To address this research question, we have formulated the cloud infrastructure allocation and pricing problem (CIAP) as an Integer Program in Chapter 3, which takes form of multi-unit combinatorial auction. We have compared the formulated problem with the standard problems in optimisation, and have demonstrated that it is a special case of (Multiple) Multi-dimensional Knapsack Problem (M)MKP [95], by analogy. This type of optimisation problems has non-deterministic polynomial time complexity (NP-hard)[121]; hence, the allocative efficient outcome (optimal solution) cannot be found in a tractable computation time, and some near-optimal approximation techniques have to be developed in order to solve the problem in a feasible time.
Research Question 3:

How to design a tractable mechanism for a cloud market that achieves the desirable economic properties, attains the most socially efficient distribution of cloud services, and limits the strategic manipulation of its participants?

In order to answer this research question, a systematic literature survey has been conducted in Chapter 2. Among the reviewed methods for solving the multi-unit combinatorial auctions, greedy combinatorial approach has showed promise for addressing the considered problem. Such approach typically allows achieving good approximation quality in a fast computation time [66][17], as well as can elicit truthful participants’ behaviour [90]. However, there is no truly effective and efficient solution in literature that can address the problem formulated in this thesis. Therefore, we have develop a set of computational market mechanisms based on greedy combinatorial principles for allocation, and pricing based on critical-value payment for buyers and surplus distribution based on mixed rules for sellers. A more detailed summary of the applied mechanism design aspects can be found in Tables 4.10a and 5.12a. Our theoretical analysis of the designed mechanisms has revealed that they maintain the mechanism design feasibility constraints, such as Individual Rationality, Budget Balance and Computational Tractability. Furthermore, some of the mechanism designs guarantee Truthfulness property in buyer declarations, which is very difficult to achieve in such a complex setting. In order to evaluate the quality of approximated properties, including the allocative efficiency and the strategic manipulation, we have conducted extensive simulation experiment. The results have revealed that the proposed mechanism designs attain near-optimal distribution of cloud services and provide near-truthful incentives to the market participants. A more detailed summary of the qualitative characteristics of the market mechanisms, proposed in this thesis, can be found in Tables 4.10b and 5.12b.

7.3 Possible Extensions and Open Questions

The work, presented in this thesis, can be extended and continued in several research directions. We list some of the most significant and fascinating areas below.

7.3.1 Real-world Data

As we have noted earlier, the cloud providers and the emerging cloud exchange platforms keep private the information about the cloud services requests, coming from customers
and cloud provider offers. Due to unavailability of public information, we had to rely on extensive simulation experiments in order to evaluate the performance of our mechanisms. Although, average performance based on modelled workloads have been analysed, it could be of great interest and value to evaluate the allocative performance and pricing outcomes of the proposed market mechanisms based on real data. While the benefits of using combinatorial market mechanisms, such as the ability to express complex requirements and achieve efficient market outcomes, are really significant, the application and validation of their performance in a real-world environment is an essential future work.

7.3.2 Software Agents with Learning Capabilities

We find the domain of strategic dynamics in the electronic markets and auctions to be a truly fascinating and relatively unexplored area of research. In the presented work, our experimental investigation of strategic misreporting focussed on a single agent applying a pre-defined set of misreporting strategies exhaustively. However, it could be really exciting to investigate the strategic dynamics in the market where the software agents have some learning capabilities and adapt their strategies based on their past experience and rapidly changing market conditions. It is very interesting to know whether theoretically truthful market mechanisms are going to motivate such intelligent agents to converge to a truthful strategy, or each individual selfish participant in the community is going to converge to a collective lie. This study can also investigate the participants’ collective power, where the major subject of analysis is the impact of the different groups of various software agent designs on the market outcomes.

7.3.3 Irrational Behaviours

The mechanism design theory is based on a very strong assumption, that the participants in competitive markets are rational decision-makers. In this thesis, we departed from the same viewpoint and considered the market participants as rational selfish decision-makers. However, recent empirical results [2] show a marked deviation in the behaviour of the participants in contradiction to theoretical predictions. The reason for this is not always rational behaviour of human traders in electronic markets. The market outcomes as well as the strategic behaviour dynamics of the market participants can be significantly affected by such unpredicted behaviours. Therefore, it is important to understand the impact of such human behaviour on the other market participants behaviour and on socio-economic outcomes of the proposed market mechanisms. The knowledge about the participants’
behaviour and the corresponding market outcomes is an important future consideration.

7.3.4 Heterogeneous Environments

According to expert predictions [114], in the nearest future the electronic marketplaces are going to be populated by intelligent software agents trading by the side of human users. We are witnessing a gradual shift to intelligent next-generation platforms, where the market-related processes are automated and facilitated for humans. As a result, a complex heterogeneous human-software agents environment is very likely to emerge. In this thesis, we have proposed the Smart Cloud Marketplace platform for trading cloud services, which is a heterogeneous environment where the humans either engage in manual trade or employ the intelligent software agents to trade on their behalf. The socio-economic and strategic dynamics in such heterogeneous platforms is a significant, interesting and highly challenging area of research to be investigated. Given various human behaviours and custom trading strategies, the cloud marketplace is a very complex environment with a mixture of strategic behaviours. These behaviours play an important role in establishing the dynamics of the market mechanisms. The knowledge about interrelations between the trading strategies and the socio-economic outcomes is crucial in realising the vision of the next-generation open marketplace for trading cloud services.

7.3.5 Bayes-Nash Equilibria in Cloud Markets

In the recent years some advancements have been made in mechanism design, including the area of multidimensional preferences and Bayes-Nash implementations. Given the new body of knowledge, it could be interesting to investigate the problem of cloud resource allocation and pricing in markets of incomplete information, where the proposed solution concept is relaxed from dominant strategies / truthfulness to Bayes-Nash equilibria. It would allow a different game-theoretic perspective on the problem of cloud services allocation and pricing.

7.3.6 Bidding Problem in a Distributed Cloud Marketplace

The cloud market may be established as a stand-alone marketplace where all buyers and sellers come together to exchange cloud services. On the other hand, it may also be formed as a number of marketplaces, where different competing sellers and buyers are trading cloud services. A bidding problem in such a distributed environment becomes more complex and needs to be investigated.
7.4 Final Remarks

Many factors indicate that an open marketplace of infrastructure cloud services is to emerge in the nearest future. We are witnessing an exponential growth of the number of interconnected devices and the coming era of Internet of Things. These devices will require more compute power to be consumed in a seamless manner so that to address the rapidly changing needs with little or no human intervention in the process. Luckily, the nature of cloud services, which is similar to other utilities, makes such service delivery technically viable. However, a large-scale, dynamic and distributed nature of such cloud provisioning environment will turn the services delivery process into a more significant problem which will need to be addressed in an efficient way. An open marketplace for trading the primary cloud resource, which is cloud infrastructure, is likely to emerge in order to address this problem by facilitating the cloud services exchange process.

Inspired by the vision of the next-generation marketplace for infrastructure cloud services, we have attempted to address the problem in this thesis. Starting from a conceptual framework to approach the cloud services provisioning scenarios, to the design and evaluation of the economics-inspired computational mechanisms, finishing by a functional proof of concept prototype of the Smart Cloud Marketplace system. Although, there is a room for further improvement and a vast unexplored field of research, we have made a step towards effective and efficient automation of market provisioning by applying economic principles for services allocation and pricing. Our contributions to the field of market mechanism design for trading infrastructure cloud services can be used by researchers in the areas of optimisation, mechanism design and computational complexity. The developed system prototype has shown that the cloud service provisioning and acquisition process can be automated by computational market mechanisms and facilitated by custom software trading agents. At current stage, we are looking for technology adopters in industry who would help us to realise our vision of the next-generation marketplace for effective and efficient cloud services provision.
Bibliography


[14] A Askarunisa, Mr N Ganesh, and Mr A Athiraja. Measuring the usage of cloud.


[34] Sergei Chichin, Mohan Baruwal Chhetri, Quoc Bao Vo, Ryszard Kowalczyk, and Marcin Stepniak. Smart cloud marketplace-agent-based platform for trading cloud


[126] Jeffrey Shneidman, Chaki Ng, David C Parkes, Alvin AuYoung, Alex C Snoeren, Amin Vahdat, and Brent N Chun. Why markets could (but don’t currently) solve resource allocation problems in systems. In *HotOS*, 2005.


[133] IE SUTHERLAND. A futures market in computer time. 1968.


