

COPYRIGHT AND USE OF THIS THESIS

This thesis must be used in accordance with the provisions of the Copyright Act 1968.

Reproduction of material protected by copyright may be an infringement of copyright and copyright owners may be entitled to take legal action against persons who infringe their copyright.

Section 51 (2) of the Copyright Act permits an authorized officer of a university library or archives to provide a copy (by communication or otherwise) of an unpublished thesis kept in the library or archives, to a person who satisfies the authorized officer that he or she requires the reproduction for the purposes of research or study.

The Copyright Act grants the creator of a work a number of moral rights, specifically the right of attribution, the right against false attribution and the right of integrity.

You may infringe the author's moral rights if you:

- fail to acknowledge the author of this thesis if you quote sections from the work
- attribute this thesis to another author
- subject this thesis to derogatory treatment which may prejudice the author's reputation

For further information contact the University's Director of Copyright Services

sydney.edu.au/copyright

Combinatorial Auction-based Mechanisms for Composite Web Service Selection

Mahboobeh Moghaddam



A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering and Information Technologies at The University of Sydney

June 2015

Abstract

Composite service selection presents the opportunity for rapid development of complex applications using existing web services. It refers to the problem of selecting a set of web services from a large pool of available candidates to logically compose them to achieve valueadded composite services. With the growing number of web services on the Internet, it is very likely to find services, which provide a similar functionality but are differentiated based on their Quality of Service offers and price. The aim of service selection is to choose the best set of services based on the requirements of a composite service requester. The current service selection approaches mostly assume that web services are offered as single independent entities. Thus, if a service provider is interested in offering a combination of services, there is no possibility for bundling. Such an assumption ignores the dependencies between constituent services of a composition, which can strongly affect the service providers' preferences in offering bundles of web services. Moreover, the current research has mainly focused on solving the problem for a single composite service. There is a limited research to date on how the presence of *multiple*, *simultaneous requests* for composite services affects the performance of service selection approaches. Addressing these two aspects significantly enhances the application of composite service selection approaches for real-world practices. Accordingly, our central aim in this thesis is to develop new approaches for the composite web service selection problem by addressing both the bundling and multiple requests issues. In particular, we propose two mechanisms based on combinatorial auction models: a single auction mechanism and a simultaneous auction mechanism. We build on well-established theories from mechanism design and auction theory. The proposed approach based on combinatorial auctions allows multiple items to be auctioned at the same time and providers can bid to offer a combination of services.

The *single auction* mechanism aims to procure the composite service at the lowest price subject to a set of allocation constraints. The set of constraints addresses the service requester's preferences and constraints about different aspects of the composite service, such as quality of service or budget. We performed extensive experiments through simulation to study the impact of bundling on the cost of the composite service. The results indicate that bundling can reduce the cost compared to when services are offered independently. However, the results show that the cost reduction applies only up to a certain threshold for the size of bundles. Beyond this threshold, with more crowded bundles, the cost tends to increase rather than decrease.

The proposed *simultaneous auction* mechanism aims to solve the problem for multiple requests, by matching the requests with offers at specific intervals. The objective of the mechanism is to procure services at the lowest price given the requesters' set of allocation constraints. To prevent the complete failure of the mechanism in the case of some requests being infeasible due to the imposed allocation constraints, the proposed mechanism minimizes the cost for the largest set of feasible requests. The simultaneous auction mechanism is supported by empirical evaluations through conducting extensive simulation experiments. The experiments compare the performance of the simultaneous auction with two greedy mechanisms which allocate services to requests one at a time: firstly, the *single auction* mechanism when applied to a set of requests one at a time, and secondly, a *fixed-price* mechanism where each service requester fixes the price to be paid for the requested composite service. The evaluations show that, firstly, the simultaneous auction mechanism achieves significantly higher success rate (that is the ratio of the feasible service requests to all requests) in allocating service offers to composite service requests compared to the other two mechanisms. Secondly, despite the greedy strategy of the single auction mechanism, the average price of a composite service achieved by simultaneous auction is not significantly different from that of the single auction. More importantly, both auction-based mechanisms achieve significantly lower prices compared to the fixed price mechanism. Thirdly, the simultaneous auction mechanism obtains more homogenous prices for the set of composite services while the prices achieved by single auction mechanism are seriously affected by their order of being considered for service selection. To summarize, the proposed simultaneous auction mechanism enhances the success rate of composite service selection, without losing its optimality in terms of the price of the composite service. By considering all composite services at the same time, it achieves more homogenous prices which can be a determining factor for the service requester when choosing which composite service selection mechanism to attend.

Dedication:

This thesis is dedicated to my beloved husband and son, Behrang and Sina, who shared with me the ups and downs of doing a PhD for four years. I love you dearly.

Acknowledgment:

I thank my advisor, Professor Joseph G. Davis, for his unwavering belief in my abilities, his willingness to listen to me, and for allowing me the freedom and support to explore new ideas. I also thank my co-supervisor, Dr. Anastasios Viglas, for his endless encouragement, support and delicate attention to details.

This thesis was co-funded by the University of Sydney and National ICT Australia (NICTA). I would like to thank both organizations for their generous support. From NICTA, I would like to thank the lab leader Professor Len Bass, and Dr. Vladimir Tosic for their insightful discussions about my research. At University of Sydney, I would like to thank Dr. James Curran for his kind support and sense of humor, and also Professor Alan Fekete, for his guidance and generosity. During the critically important final six months of my PhD, when I was a visitor at the University of Queensland, I received the kind support of Professor Shazia Sadiq, for which I truly thank her.

I would like to thank my husband, Behrang Assemi who has been supportive, encouraging, patient and a true friend to me all along. Despite going through the same stressful PhD experiences yourself, you always showed genuine interest in my problems and research discussions, and motivated, cajoled and encouraged me not to give up. I will be forever grateful for having you in my life. Thanks to my five years old son, Sina, who has demonstrated far more understanding than can be expected from one so young. When mommy was away, busy with work or tired, your generous offers of affection, kisses and hugs gave me the strength to continue.

A good support system is critical to surviving a PhD and keeping one's motivation level above zero. Especially as an international student with my husband doing a PhD himself and a little son, I was blessed to have my friends giving me the love, support and help I needed both on the good and bad days of this journey. Thanks to my friends, Shaghayegh Sharif for being a terrific friend and my patience stone; Waiho Wong for all the insightful discussions; Leila Zamani for being my source of positive energy; Vahid Yaghini for being there for the whole family in all sorts of troubles; Maryam Fathi for singing, dancing and crying with me; Elnaz Kalaly, Paria Jalili and Hamdam Ebrahimi for being Sina's aunties when my sister was far away; Kelly Webster for her patience with my complaints and driving; Mashud Rana for being too good a grad student to be believable; Banafsheh Pazoki for her golden heart; Paul Zervos for his ability in non-stop creative discussions; Evelyn for being kind and gentle and supportive; and my new friends in Brisbane, Rashida and Valerie, who didn't let Sina feel the brunt of my late hours of staying at school during the last few months of writing the thesis.

Finally, to my parents, Vajihe and Mohammad Reza, who paved this long journey with me from 10,000 kilometers away and supported me with their unconditional love and care. You are my true heroes. To my mother in law, Jaleh, who is as dear as my mother, for her love and support. My deepest love and thanks go to my kind brother, Hussein, and my lovely little sister, Mana, for their constant love despite being so far from me. I miss you all every single moment of my life.

Table of Contents

| Abstract | ii |
|-----------------|------|
| Acknowledgment | v |
| List of Tables | xi |
| List of Figures | xiii |

| 1 |] | Introduction | 1 |
|---|--------|--|---|
| | 1.1 Pr | eliminary | 1 |
| | 1.2 M | otivation: Issues in Composite Service Selection | 3 |
| | 1.2.1 | Dependencies between Constituent Web Services of a Composite Service | 3 |
| | 1.2.2 | Price Determination for Web Services | 4 |
| | 1.2.3 | Web Services' Market: Solving the Problem for Multiple Requests | 5 |
| | 1.3 Re | esearch Objectives | 6 |
| | 1.4 Co | ontributions | 6 |
| | 1.4.1 | A Combinatorial Auction Mechanism for Composite Service Selection | 6 |
| | 1.4.2 | 2 Composite Service Selection in the Presence of Multiple Requests | 8 |
| | 1.4.3 | B Design of a Comprehensive Simulation-based Evaluation Process | 0 |
| | 1.5 Re | esearch Methodology 1 | 1 |
| | 1.6 Tł | nesis Organization | 2 |
| 2 |] | Foundation and Basic Concepts1 | 3 |
| | 2.1 In | troduction1 | 3 |

| | 2.2 | Web | Services Technology | 14 |
|---|-----|-------|--|----------|
| | 2. | 2.1 | Web Services | 14 |
| | 2. | 2.2 | Web Service Composition | 14 |
| | 2. | 2.3 | WSC Lifecycle | 16 |
| | 2.3 | Mark | ets and Auctions | 17 |
| | 2. | 3.1 | Dynamic Pricing | 19 |
| | 2. | 3.2 | Auction Categories | 21 |
| | 2. | 3.3 | Combinatorial Auctions | 23 |
| | 2.4 | Mech | anism Design | 26 |
| | 2. | 4.1 | Auction Properties | 27 |
| | 2. | 4.2 | Auction Design Elements | 33 |
| | 2.5 | Conc | lusion | 38 |
| 3 | | Li | terature Review | 40 |
| | 3.1 | Intro | luction | 40 |
| | 3.2 | Resea | arch Problem: Composite Web Services Selection | 40 |
| | 3.3 | Web | Service Selection Challenges | 42 |
| | 3.4 | Basis | of the Categorization | 49 |
| | 3.5 | Optin | nization-based Approaches | 53 |
| | 3. | 5.1 | Local Optimization | 53 |
| | 3. | 5.2 | Global Optimization | 54 |
| | 3.6 | Nego | tiation-based Approaches | 59 |
| | 3. | 6.1 | Negotiation Object | 62 |
| | 3. | 6.2 | Negotiation Protocol | 63 |
| | 3. | 6.3 | Decision Making Model | 64 |
| | 3. | 6.4 | Coordination Model | 67 |
| | 3.7 | | | <u> </u> |
| | | Hybr | id Approaches | 68 |

| 3.7.2 | Optimization + Negotiation Approach | 69 | |
|-----------|---|---------|--|
| 3.8 Auct | tion-based Approaches | 70 | |
| 3.8.1 | Auction Models | 71 | |
| 3.8.2 | Bidding Language | 76 | |
| 3.8.3 | Allocation Rules | 78 | |
| 3.8.4 | Payment Rule | 80 | |
| 3.9 Con | clusion and Discussion | | |
| 4 T | heory and Methodology | 85 | |
| 4.1 Intro | duction | 85 | |
| 4.2 Theo | pretical Foundations | | |
| 4.2.1 | Web Service Technology | | |
| 4.2.2 | Auction Theory | | |
| 4.2.3 | 4.2.3 Mechanism Design | | |
| 4.2.4 | Mathematical Optimization | 88 | |
| 4.3 Rese | earch Methodology | 89 | |
| 4.3.1 | Designing the Auction-based Mechanism | 89 | |
| 4.3.2 | Modelling the Proposed Auction-based Mechanism | 95 | |
| 4.3.3 | The Simulation-based Evaluation Process | 95 | |
| 4.4 Cone | clusion | | |
| 5 A | n Auction-based Mechanism for Composite | Service | |
| Selection | 1 | 101 | |
| 5.1 Intro | duction | 101 | |
| 5.2 A M | echanism based on Combinatorial Auctions | | |
| 5.2.1 | Cohesion in a Composite Service | 103 | |
| 5.2.2 | Configuration of the Composite Service Provisioning | 107 | |
| 5.3 The | Design of the Combinatorial Auction Mechanism | | |
| 5.3.1 | Multi-attribute Bid Specification | | |

| | 5.3.2 | The Winner Determination Problem (WDP) | |
|---|-----------|--|-----|
| | 5.1.1 | The Payment Rule | 111 |
| | 5.2 Expe | eriment Design | 114 |
| | 5.3 Resu | llts | 115 |
| | 5.3.1 | The Impact of the Cohesion Constraint on Cost and Success Rate | 115 |
| | 5.3.2 | The Impact of Bundling on Cost | 119 |
| | 5.3.3 | Analysis of the Solve Time | |
| | 5.4 Cond | clusion | |
| 6 | Α | Market for Composite Services | 130 |
| | 6.1 Intro | duction | |
| | 6.2 A M | arket for Web Services | 131 |
| | 6.3 The | Design of a Marketplace | 133 |
| | 6.3.1 | Specification of Offers and Requests | 135 |
| | 6.3.2 | Full-Matching of Requests and Offers | 137 |
| | 6.3.3 | Partial-Matching of Requests and Offers | 139 |
| | 6.3.4 | Providers' Resource Limitation | 143 |
| | 6.4 Expe | eriment Design | 144 |
| | 6.4.1 | Performance Metrics | 144 |
| | 6.4.2 | Establish a Baseline | 145 |
| | 6.4.3 | Scenarios to Investigate: Market Sections | 147 |
| | 6.4.4 | The Stochastic Model to Generate Data | 152 |
| | 6.5 Expe | eriment Execution | 157 |
| | 6.6 Resu | lts | |
| | 6.6.1 | Success Rate (SR) | |
| | 6.6.2 | Cost per Composite Service (CPC) | 167 |
| | 6.6.3 | Solve Time | 177 |
| | 6.7 Cond | clusion | 179 |

| 7 Conclusion | | | 182 |
|--------------|----------|---|-----|
| | 7.1 Cont | ributions and Summary of Results | 182 |
| | 7.1.1 | A Combinatorial Auction Mechanism for Composite Service Selection | 182 |
| | 7.1.2 | A Simultaneous Auction Mechanism for Trading Composite Services | 186 |
| | 7.1.3 | A Comprehensive Evaluation Process | 190 |
| | 7.1.4 | Summary Remarks | 191 |
| | 7.2 Limi | tations | 192 |
| | 7.2.1 | Solve Time | 192 |
| | 7.2.2 | Economic Efficiency and Incentive Compatibility | 193 |
| | 7.2.3 | Data | 198 |
| | 7.3 Impl | ications and Future Work | 198 |
| 8 | Re | eferences | 201 |

List of Tables

| Table 2.1. Categorization of auctions | .23 |
|---|-----|
| Table 3.1. Auction-based service selection approaches | .82 |
| Table 4.1. The four market sections | .98 |
| Table 4.2. The implementation of the evaluation process for the proposed auction-base | ed |
| mechanisms | .99 |
| Table 5.1. Calibrating the local and interactive cohesion factors (LCF and ICF) of the offere | ed |
| bundles based on the BP structure | 106 |
| Table 5.2. Configuration of the simulation parameters for the experiment on the cohesion | on |
| constraint1 | 116 |
| Table 5.3. Configuration of the simulation parameters for the experiment on bund | ile |
| crowdedness, Stage 1 | 119 |
| Table 5.4. Configuration of the simulation parameters for the experiment on bund | lle |
| crowdedness, Stage 2 | 121 |
| Table 5.5. Configuration of the simulation parameters for the experiment on bund | ile |
| crowdedness, Stage 31 | 124 |
| Table 5.6. Configuration of the simulation parameters to evaluate the solve time | 127 |
| Table 6.1. Notation used for the formulation of Full-Matching and Partial-Matchin | ng |
| mechanisms1 | 143 |
| Table 6.2. The four market sections in the experiments | 147 |
| Table 6.3. Seeding the experiment's input parameters | 157 |
| Table 6.4. The evaluation space 1 | 160 |
| Table 6.5. The Kruskal-Wallis test (dependent variable: SR, independent variable: type | of |
| mechanism)1 | 165 |
| Table 6.6. The Conover-Iman procedure (dependent variable: SR, independent variable: typ | pe |
| of mechanism) | 166 |
| Table 6.7. The Kruskal-Wallis test (dependent variable: SR, independent variable: type | of |
| mechanism, size of economy and requests' complexity) | 166 |
| Table 6.8. The Conover-Iman procedure (dependent variable: SR, independent variable: typ | pe |
| of mechanism, size of economy and requests' complexity) | 167 |

| Table 6.9. The Kruskal-Wallis test (CPC) (dependent variable: CPC, independent variable: |
|---|
| type of mechanism)175 |
| Table 6.10. The Conover-Iman procedure (dependent variable: CPC, independent variable: |
| type of mechanism)175 |
| Table 6.11. The Kruskal-Wallis test (dependent variable: CPC, independent variable: type of |
| mechanism, size of economy and requests' complexity)175 |
| Table 6.12. The Conover-Iman procedure (dependent variable: CPC, independent variable: |
| type of mechanism, size of economy and requests' complexity) |
| Table 7.1. An example scenario for the full matching mechanism 196 |
| Table 7.2. An example scenario for the partial matching mechanism |

List of Figures

| Fig 2.1. Web service composition lifecycle |
|--|
| Fig 2.2. Categorization of dynamic pricing, adopted from Stein et al. (2003)20 |
| Fig 3.1. Classification of composite service selection approaches based on QoS Profile 50 |
| Fig 3.2. WSC negotiation framework adapted from the general negotiation framework |
| (Mueller 1996; Faratin et al. 1998)62 |
| Fig 3.3. The realization of the WSC negotiation framework based on the current literature64 |
| Fig 3.4. Different perspectives on applying negotiation for service selection during WSC 70 |
| Fig 3.5. An example bid in the form of volume discount curve77 |
| Fig 4.1. Designing an auction-based mechanism for composite service selection90 |
| Fig 4.2. The evaluation process96 |
| Fig 5.1. An example business process with five tasks106 |
| Fig 5.2 The dependency matrix of the example business process |
| Fig 5.3. Success Rate of the auction mechanism for different cohesion constraints117 |
| Fig 5.4. Impact of different levels of cohesion constraint on cost117 |
| Fig 5.5. Cost as a function of cohesion |
| Fig 5.6. Impact of bundle crowdedness on (a) cost and (b) cost ratio (CATS arbitrary |
| distribution) |
| Fig 5.7. Impact of bundle crowdedness on cost (discounted pricing function, IPV |
| providers)122 |
| Fig 5.8. Impact of bundle crowdedness on cost ratio (discounted pricing function, IPV |
| providers) |
| Fig 5.9. Number of winning bids with 1 service, 2 services, up to 20 services versus alpha 125 |
| Fig 5.10. Number of winning bids with 1 service, 2 services, and more than 2 services versus |
| alpha |
| Fig 5.11. The cost of the composite service versus alpha |
| Fig 5.12. Solve time of the auction mechanism |
| Fig 6.1. Example of a bid for a bundle of services |
| Fig 6.2. Example of a request for a composite service137 |
| Fig 6.3. Web services' market sections (Weinhardt et al. 2011a)149 |

| Fig 6.4. A mobile computing scenario: FuelBuddy150 |
|--|
| Fig 6.5. Scientific workflows scenario: protein sequence analysis (M.B. Monteiro 2008) 152 |
| Fig 6.6. SR achieved by the three mechanisms in the four market sections161 |
| Fig 6.7. Impact of number of tasks on the simultaneous auction's SR for (a) simple and (b) |
| complex requests |
| Fig 6.8. Impact of number of providers on simultaneous auction's SR for (a) simple and (b) |
| complex requests |
| Fig 6.9. CPC achieved by the three mechanisms in the four market sections168 |
| Fig 6.10. Cost of procuring the first arriving and the last arriving request, in a simultaneous or |
| single auction mechanism, for (a) simple and (b) complex requests170 |
| Fig 6.11. CPT achieved by the three mechanisms in the four market sections |
| Fig 6.12. Impact of number of tasks on the simultaneous auction's CPT for (a) simple and (b) |
| complex requests |
| Fig 6.13. Impact of number of providers on the simultaneous auction's CPT for (a) simple and |
| (b) complex requests |
| Fig 6.14. Solve Time of the three mechanisms to find the best allocation for the four market |
| sections |

Chapter 1

1 Introduction

1.1 Preliminary

Web service technology has transformed the Internet from solely being a source of information to a more advanced level of the "Internet of services", presenting Internet users with access to a wide range of services. The service-enabled Internet is realized through the open, Internet-oriented and standards-based interfaces of web services and the standards-based technologies that they use to communicate with one another. These characteristics enable the creation of services which can be easily discovered and consumed by external users independent of their hardware, operating system or programming environment.

Single web services usually are designed to offer limited atomic functionalities such as searching in Google, obtaining the weather condition of a region or showing the location of a vehicle on a map by getting its geographic specification. Such a design adheres to the fundamental service-oriented computing design principle of "reusability"; the limited atomic functionality makes it easier to reuse web services as building blocks in different applications. However, what users need is generally a complex functionality which might not be found in a single web service, such as planning a trip, online patient follow-up, applying for admission to a university or protein sequence analysis. The most natural solution to make the complex functionality appears to be integrating the appropriate set

of web services in a proper sequence so that they can jointly achieve the required functionality, a process known as *web service composition*.

Web service composition (WSC) has been an active research area in service-oriented computing for more than 10 years. One major reason for such interest among researchers is the promise of the rapid development of software systems with low development cost. The exciting idea of creating complex applications by composing existing concrete services has the potential to change the way organizations build or procure their software applications. This can also lead to significant time and money savings at both the development and the maintenance stages of the software system lifecycle.

With the constantly growing number of available online web services that can perform similar functionalities at different levels of quality and price, one major challenge in building a composite service is to select the most appropriate web services for composition. This is referred to in the literature as *quality-aware or quality-driven service selection* for *web service composition* (Zeng et al. 2003; Canfora et al. 2005; Michlmayr et al. 2010; He et al. 2014). From now on, we will refer to it as *composite web service selection*. More formally, composite web service selection is the process of selecting the "optimal" set of web services that can collectively achieve a specific complex functionality when logically composed together, from the pool of available services. Optimality is defined based on the composite service characteristics such as the quality and price.

We argue that web services' characteristics and their execution context (including their users, providers and execution environment) present particular challenges to the composite service selection process for which adequate resolutions cannot be found in the current approaches. In this thesis, we have viewed composite service selection as a complex resource allocation problem. We have investigated the application of auction theory to solve this problem which can improve the existing research on web services regarding the vital issues that have been largely ignored so far.

1.2 Motivation: Issues in Composite Service Selection

Composite service selection is widely acknowledged to be a complex problem. Several issues contribute to this complexity. While some of these issues have been extensively studied in the literature, some others have not received the attention they deserve. Among the issues receiving the attention of the research community, we can name: the NP-hardness and scalability of composite service selection; the non-sequential structures in a composite service; aggregation models for quality attributes; and the difficulty of determining the trade-off between various quality attributes from the requester's perspective, to be discussed in more details in section 3.3 (Web Service Selection Challenges). In this research, we have identified and addressed three additional important issues that we believe have not received the attention they deserve. These issues are discussed below:

1.2.1 Dependencies between Constituent Web Services of a Composite Service

The dominant assumption in the existing service selection literature is that web services are offered as independent entities. Even if a provider offers more than one service, the offers are considered to be independent. In other words, providers cannot offer bundles of web services. We argue that this assumption does not consider the dependencies that exist between web services participating in a composition.

Atomic web services that form the composite service are dependent on each other based on different factors such as input/output data, execution time or domain related constraints (Yang and Papazoglou 2002; Milanovic and Malek 2004; Verma et al. 2004; Omer and Schill 2009). The existence of these dependencies has implications for web service providers: the dependencies can lead to complementarity effects among web services. For example, the complementarity effects may help service provider offer a bundle of services with a lower price or improved quality level. Such possibilities create strong motivation for the service providers to offer their services in bundles, rather than offering individual services separately. A composite service selection approach cannot achieve the optimal solution for the service requester if it does not support the needs of service providers in expressing their preferences for bundling of web services.

1.2.2 Price Determination for Web Services

In the extant literature, there are two trends in the assumption about the pricing models of web services. The first, which is also more dominant, is that web services are offered at a fixed predetermined price. The second model is that the price of each web service is determined through a negotiation process between the service requester and service providers. The first assumption is the basis of the optimization-based composite service selection approaches and the second one is the basis for the negotiation-based approaches, each of which will be discussed in detail in Chapter 3 (Literature Review).

In the first model, the price is fixed for all consumers and, in the best scenario, we can imagine that when providers realize the need to change the price of their offers, they have to determine a new price and update the web service specification accordingly. This is a pricing strategy known as posted-price or fixed pricing (Wang 1993). This means that the complexity of determining the price of a web service is completely left to service providers.

Such a pricing strategy has major complications for web service providers. From the pricing theory perspective, web services are considered to be products with low specificity, meaning that it is possible to sell them over and over with a very low marginal cost. This is due to the open, Internet-oriented and standards-based interfaces of web services. Therefore, service providers face the problem of pricing their web services based on the supply and demand from the requesters and providers' sides, rather than the cost of production. However, such information about supply and demand is neither readily available, nor easy to obtain.

There is a constant fluctuation in the supply and demand of web services offered over the Internet due to its open and changing nature. This means that the service providers need to constantly monitor the market to be able to set the prices at the most profitable level. Clearly, such a continuous monitoring of the market will cost providers considerable time and money. Considering the nature of web services in typically offering limited functionality at a relatively low price, such a pricing strategy is unlikely to be profitable for service providers.

In the second pricing model, the price is completely flexible and determined through (automated) negotiation process between the service providers' and requester's software agents. This pricing model still has not found practical applications due to the complexity involved in an automated negotiation process. The current proposals with a negotiation-based pricing model have tried to reduce the complexity by creating a simplified model of the negotiation process through imposing restricting assumptions on the strategies, tactics and utility functions of the negotiators. As a result, it is not very likely that these approaches find practical applications in the web services domain in near future.

1.2.3 Web Services' Market: Solving the Problem for Multiple Requests

Composite services have been recognized as a crucial part of web service marketplaces (Papazoglou 2003; Yarom et al. 2004; Weinhardt et al. 2011b). These marketplaces create the opportunities for service providers and requesters to meet and trade single and composite web service (Papazoglou 2003). However, very limited research has been done to examine how such marketplaces affect composite service selection. The presence of multiple composite services extends the composite service selection problem to what we call the "*multiple composite service selection*" problem.

The existing approaches mainly solve the problem for a single request with no further discussion about solving the problem for multiple requests, neither simultaneously nor one by one. To the best of our knowledge, our work is one of the very first studies to investigate composite service selection in the presence of multiple requests for composite services.

This is an important issue to study in service selection that its resolution has the potential to enhance the practicality of web service selection approaches as the research considers a more realistic setting where there are likely to be multiple requests for composite services. This consideration and the results can significantly impact the design and development of web services' marketplaces.

1.3 Research Objectives

Our central objective in this thesis is to develop new approaches for the composite web service selection problem taking into account the complementarities between web services forming a composition, the necessity for more dynamic pricing mechanisms, and the need to cater for the presence of multiple requests. We address the research gap by conducting this study with three main objectives:

- Identify the important issues that need to be considered in a composite service selection approach, namely, complementarity effects among web services forming a composition, the complexities related to price determination in web services, and the presence of multiple composite services,
- Introduce and develop new approaches based on auction theory to incorporate the above-mentioned requirements in the process of selecting web services,
- Demonstrate the specific properties of these approaches which make them suitable for web services and their execution environment through performing comprehensive and objective evaluations of the proposed approaches.

Thus, the aim of this thesis is to investigate how auction theory can be used to facilitate composite service selection and to improve our understanding of the fundamentals behind it. Building on the current theories and developments in web service technology, auction theory, mechanism design and mathematical optimization, this study proposes a novel approach to address the web service selection problem.

1.4 Contributions

1.4.1 A Combinatorial Auction Mechanism for Composite Service Selection

We propose an auction-based mechanism to solve the composite service selection problem. As the proposed mechanism aims to solve the problem for a single composite service request, we refer to it as the *"single auction mechanism"* (Chapter 5).

A design based on auction models in general, and combinatorial auctions in particular achieves these desirable properties: (1) enhances dynamic pricing for composite services compared to a fixed pricing strategy; (2) facilitates price determination of single and composite services by sending constant feedback about the status of supply and demand obtained from the information revealed after each auction; and (3) accommodates the need for bundling web services due to the inter-service dependencies between constituent services of a composition.

By considering the dependencies between web services of a composition and offering services in bundles according to those dependencies, service providers are enabled to: (1) offer discount over the price of the bundled services by internalizing some of the cost of service provisioning, and (2) improve the quality of bundled services by having more control over the communication and execution of the bundle. These can enhance the providers' competitive power in the market as well as the consumer's loyalty.

The *single auction* mechanism aims to procure the composite service at the lowest price subject to a set of allocation constraints. The set of constraints addresses the service requester's preferences and constraints about different aspects of the composite service, such as quality of service or budget. The proposed mechanism is formulated as an Integer Linear Programming (ILP) problem.

• Studying the Impact of Bundling on the Cost of a Composite Service

While there are other proposals on application of combinatorial auctions to solve the composite service selection problem, to the best of our knowledge no other research has studied the impact of bundling, in terms of the bundle size, on the performance of the composite service selection approach, in terms of the achieved cost for the composite service. The result of this study is important for both service providers and requesters. On one hand, service providers can increase their chance of winning the service selection auction by choosing the right size for their bundles which consequently leads to increase in their profit. On the other hand, in a market where many providers claim to offer discounts over bundle of items, service requesters can reduce their cost of service provisioning by understanding how the bundle size affects the cost of the composite services.

• Introducing and Measuring the Cohesion of a Composite Service

We introduce the concept of "cohesion" of a composite service and propose a technique to measure it. The proposed notion of cohesion enables the service requesters to manage the dependency of a composite service to its service providers which affects important quality (non-functional) requirements such as the maintainability, reliability and (provider-) dependability of the composite service. The cohesion is defined based on direct data dependencies between the participant services offered by the same provider. It is measured as the sum of the cohesion of the bundles winning the auction to execute the composition. We have developed a resource allocation constraint that enables the service requester to define a lower and an upper bound for the cohesion of the composite service. The single auction mechanism checks this constraint while looking for the optimal service allocation for the composite service.

• Identify the Need and Develop Constraints to Manage the Configuration of Composite Service Provisioning

We recognize the service requester's need to manage the configuration of service providers in the execution of a composite service. More specifically, we identified two important patterns of the service providers' involvement in the composition: a set of tasks need to be executed by "the same provider" or by "different providers". These patterns are very important in the context of service requester's security and privacy concerns. We have developed two resource allocation constraints corresponding to each of the patterns, which are added to the ILP formulation of the single auction mechanism.

1.4.2 Composite Service Selection in the Presence of Multiple Requests

We introduce *the multiple composite service selection problem* by extending the composite service selection to include multiple requests for composite services. To the best of our knowledge, this is the first study to consider, investigate and propose solution for composite service selection in the presence of multiple requests (Chapter 6).

The results of this study is critically important for designing and managing web services' marketplaces where service requesters and providers meet to trade single and composite services. However, very limited study has been done to examine how such a marketplace impacts the composite service selection process. All existing service selection approaches solve the problem for a single request and no discussion exists about solving the problem for multiple requests, neither simultaneously nor one by one.

• Proposing Two Simultaneous Auction Mechanisms to Solve the Multiple Composite Service Selection Problem

We propose a novel mechanism based on combinatorial auctions to match multiple composite service requests with the web service offers simultaneously. The proposed "*simultaneous auction*" mechanism comes in two variations: *Full-Matching* and *Partial-Matching* mechanisms.

The Full-Matching mechanism aims to procure services for all the requests at the lowest price, given the requesters' set of allocation constraints. Consequently, if there are any requests that are not feasible due to the service requester's budget or quality constraints, the whole auction fails and no request, even the feasible ones, will be assigned any services. The Partial-Matching mechanism aims to prevent the auction failure in such circumstances by solving the composite service selection problem for the largest set of feasible requests. More specifically, the Partial-Matching mechanism minimizes the cost for the largest set of feasible requests. The proposed Partial-Matching simultaneous auction mechanism is supported with empirical evaluations by conducting extensive simulations.

• Studying the Impact of Simultaneous Consideration of Multiple Requests on the Performance of the Composite Service Selection Approach

We conducted extensive simulations to study the impact of simultaneously considering multiple requests on the performance of the service selection mechanism. To perform this study, we compared the simultaneous auction (the Partial-Matching mechanism) with two other mechanisms which solve the composite service selection problem for multiple requests, one at a time. We also defined the important performance metrics for our problem domain which are: (1) the success rate of the mechanism in finding the optimal service allocation for the composite service requests, (2) the average cost of procuring a composite service, and (3) the time to find the optimal allocations.

This study is performed in specific sections of the web services' market. This is the first study that aims to focus on particular market sections for web services rather than a generic market. We believe that such a setting is a more realistic scenario for the evaluation the proposed approaches, compared to a generic setting for web services' market.

This study and its results have important implications for web service market makers (independent parties who create and maintain the web service markets) as well as the market's participants (service requesters and providers). On one hand, it provides insightful guidelines for market makers on designing the appropriate service selection mechanism based on their target market participants. On the other hand, the results of this study help service providers and requesters making more informed decisions about the type of service selection mechanism to attend, considering their priorities for the different performance metrics such as the need to a fast allocation mechanism or finding the lowest prices for the composite services.

1.4.3 Design of a Comprehensive Simulation-based Evaluation Process

We designed a simulation-based evaluation process that improves the clarity of the evaluation process of service selection approaches. The evaluation process enforces a clear specification for a specific set of elements that can be used as a framework for simulation-based experiments on composite service selection approaches. These elements are:

- 1. Defining the performance metrics
- 2. Establishing the baseline for comparison
- 3. Determining the scenarios to be investigated
- 4. Specifying the simulation's data generation model
- 5. Determining the seeding of the simulation's parameters

We have also designed a baseline which can be used for the evaluation of service selection approaches that are based on dynamic pricing strategy, that is, the auction-based and negotiation-based approaches. Being an alternative to the dynamic pricing strategy, the baseline is founded on a fixed pricing strategy where the service requesters fix the price to be paid for the composite services. This design is based on the specific characteristics of the composite service selection problem and will be discussed in more details in subsection 6.4.2.2 (Fixed-price Mechanism).

Moreover, we have designed the simulation to be performed in particular sections of the web services markets, rather than a generic market. Considering of such sections is particularly important in the evaluation of web service selection approaches as the publicly available data for web service offers and composite service requests are very

limited. Therefore, it is very important to make the data generation part of the simulation as realistic as possible.

Based on the information we collected from the existing web services directories online, we decided to perform experiments in specific market sections for web services, rather than a general market as it would be rather difficult to have any estimate for the number of service providers, requesters and the type of the web service offers and requests in a generic market for web services. The market sections are designed based on important factors distinguishing these sections. This will be further discussed in subsection 6.4.3 (Scenarios to Investigate: Market Sections).

1.5 Research Methodology

To address the proposed research objective, we designed and employed a research methodology which has three parts: (1) designing of an auction-based mechanism, (2) modelling of the proposed mechanism, and (3) evaluating the proposed mechanism using simulations.

The design of an auction-based mechanism for composite service selection requires answering two important questions: (1) what are the elements of an auction model? (2) how is an auction model for service selection different from auction models in other domains such as transportation, communication networks or cloud computing? The answer to the first question defines the elements of an auction model, to be discussed in subsection 2.4.2 (Auction Design Elements). The answer to the second question specifies the particular requirements of the composite service selection problem to an auction-based solution which differentiate the auction model in this domain from other domains, to be discussed in subsection 4.3.1 (Designing the Auction-based Mechanism).

In the second part, the elements of the auction-based mechanism are modelled mathematically. The proposed mechanism is mapped to an Integer Linear Programming problem and implemented mathematically using a language called AMPL¹.

Finally, the proposed auction-based mechanism is evaluated by conducting simulations. The design of the evaluation process was a challenge for our study, as this research is one

¹ A Modelling Language for Mathematical Programming <<u>http://www.ampl.com/</u>>

of the first studies, which apply auction theory to composite service selection and its extension, namely multiple composite service selection. First we designed an evaluation process for the single auction mechanism. The evaluation process was later revised and extended to be applied on the proposed simultaneous mechanism.

1.6 Thesis Organization

Chapter 2 presents the fundamental concepts about web service technology, auctions and mechanism design. Chapter 3 expands on the extant literature on composite service selection, covering optimization-based, negotiation-based and auction-based approaches to service selection. Chapter 4 describes the four theoretical pillars of our research and the research methodology that we followed to develop an auction-based approach for the composite service selection problem. Chapter 5 introduces the design of an auction mechanism for service selection for a single composite service, along with the findings of the proposed mechanism's evaluation. Chapter 6 extends the composite services. Two mechanisms are introduced: Full-Matching and Partial-Matching. The results of the experimental simulation to evaluate the proposed mechanisms are also presented in this chapter. Chapter 7 concludes with a summary of general research issues, contributions, empirical analysis, limitations, and future outlook.

Chapter 2

2 Foundation and Basic Concepts

2.1 Introduction

In this chapter, we present the basic concepts related to the three main streams of the current study, covering concepts from web service technologies, auction theory and mechanism design.

In section 2.2, we give an introduction to web service technology, including a discussion of the definition and characteristics of web services and a review of the web service composition (WSC) process and its objective. This section closes by outlining a lifecycle for the WSC process to give a clearer understanding of what is involved in a typical WSC process. The WSC lifecycle was proposed by Moghaddam (2011).

Section 2.3 introduces auction theory. We present a broader picture of markets before drilling down to auctions as the most dominant trading mechanism in markets. This section includes a discussion of the definition and classification of auctions and of one of their important characteristics, namely, dynamic pricing. We then discuss in more detail a specific type of auction, called the combinatorial auction, which has a central role in our study as it allows for the auctioning of multiple items simultaneously.

Section 2.4 presents mechanism design for auctions and the main challenge of mechanism designers: how to design a mechanism that achieves the designer's set of desirable properties despite the selfish behavior of rational participants. We then introduce the most important set of desirable properties for auctions. The three design elements of an auction,

namely the bidding language, the allocation rules and the payment rule, are discussed at the end of this section.

2.2 Web Services Technology

2.2.1 Web Services

Web services have been advancing as the technology of choice for realizing serviceoriented computing. Web services are self-contained, modular business applications with open, Internet-oriented, standards-based interfaces (UDDI Consortium 2001). A technical definition, focusing on the technologies involved, defines web services as:

"A Web service is a software system designed to support interoperable machineto-machine interaction over a network. It has an interface described in a machineprocessable format (specifically WSDL²). Other systems interact with the Web service in a manner prescribed by its description using SOAP³-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Webrelated standards." (W3C Working Group Note 11 2004)

Web services communicate directly with each other via standards-based technologies such as XML messaging. Open standards-based communications give customers and suppliers the opportunity to access different web services independent of their hardware, operating system or even programming environment. This flexibility means that businesses can expose their business applications as web services which can be easily discovered and consumed by external parties.

2.2.2 Web Service Composition

Although a single web service has its own value for its users, the functionalities offered by the individual web services are generally limited atomic functionalities to follow the fundamental service-oriented computing design principles, such as "reusability" (Erl

² Web Services Description Language (WSDL) is an XML-based interface description language which is used to specify the functionality offered by a web service.

³ Simple Object Access Protocol (SOAP) is an XML-based protocol for exchange of information in a distributed decentralized environment.

2005, chap.9). For example, an atomic web service may retrieve a map of a location, obtain the weather conditions of a region, search in Google or show the location of a vehicle on a map.

However, what the service requesters mostly require are complex functionalities that cannot be obtained from a single web service, such as planning a trip (McIlraith et al. 2001), online patient follow-up (Omer and Schill 2009), protein sequence analysis (Thakkar et al. 2005; Medjahed and Atif 2007) or constructing a map of the landing area annotated with weather, meteorology and tidal conditions (Kim et al. 2009).

Therefore, it is desirable to logically connect several atomic web services to satisfy complex functional requirements. The true potential of web services can only be achieved through assembling web services into more powerful workflows and other applications with more sophisticated functionalities, leveraging the loose coupling characteristics of service-oriented architecture (SOA). This process is called *web service composition*.

Web service composition (WSC) refers to the process of combining different web services to provide a value-added service (Medjahed and Bouguettaya 2005). It does not involve the physical integration of all components; rather, the basic components that participate in the composition remain separate from the composite web service (Charfi and Mezini 2004).

Composition of web services enables the building of cross-enterprise applications on the Web (Alonso et al. 2004). This is mainly motivated by three factors. First, tomorrow's Web is expected to be highly populated with web services. Second, the adoption of XML-based messaging over well-established protocols enables communication among disparate systems. Third, the use of a document-based messaging model in web services caters for loosely coupled relationships among applications owned by different organizations (Medjahed and Bouguettaya 2005).

Due to the promise of rapid development of software systems with a low development cost, web service composition has been an active research area for more than 10 years. Despite the slow progress in adoption of WSC approaches by industry, many researchers in service-oriented computing believe that it has the potential to significantly impact software development and maintenance methodologies and practices.

2.2.3 WSC Lifecycle

To discuss the detail of what is involved in a typical WSC process, we have defined the web service composition (WSC) lifecycle (Moghaddam 2011). The presented lifecycle gives a comprehensive view of all the stages in WSC which will help us to precisely position our research problem.

In our discussion, we will use "service requester" to refer to the user who requires a complex functionality that can be built by composing existing web services offered by service providers. We have divided the WSC process into five stages, as illustrated in Fig 2.1.

The first stage is called *goal specification*. In this stage, the service requester's goal and preferences are defined. Following this, the high-level goal is semi-automatically decomposed into an abstract business process (BP). The BP comprises a set of tasks, each with clearly defined functionality, along with the control and data flow between them. The Quality of Service (QoS) requirements for the end-to-end BP and for each participating task are also specified.



Stage 3: Web Service Selection

Fig 2.1. Web service composition lifecycle

In the next stage, *web service discovery*, web services that match the tasks' functional and non-functional requirements are located by searching a service registry that holds information about available web services. The objective of service discovery is to find a

service match for each of the BP's tasks. At this stage, it is very likely that more than one candidate will be found for each task that, while satisfying the required functionality, may have different levels of quality and price.

The next stage, *web service selection*, aims to select those web services that best match the service requester's preferences and constraints. This can be defined at two levels: first at a local level for each task and its corresponding web service and, second, at the endto-end level for the whole composite service. The preferences and constraints are mainly about the quality of the service and price. After choosing the best match for all tasks in the BP, each task is bound to the selected web service to create the *concrete composite service*.

During the *composite service execution* stage, the composite service is executed. With each execution, an instance of the composite service is created. The service instance is continuously monitored for any failure or change in its status at the final stage of WSC, namely, *composite service maintenance and monitoring*.

This research focuses on the third stage of the WSC lifecycle, namely, web service selection. The application of web service selection is found not only in the context of WSC, but also in the context of the complex processes that are partly made up of web services and partly from legacy systems, such as some scientific workflows. We will present a detailed discussion of the web service selection process in the next chapter. We will also include a detailed review and analysis of the current literature on service selection approaches.

2.3 Markets and Auctions

In general, markets are physical or virtual meeting points where buyers and sellers interact to set prices and exchange goods and services. The prices represent the values of the goods and services in terms of money. People and firms voluntarily exchange different commodities at the price level. Prices also serve as signals to producers and consumers: if, for example, consumers demand more of an item, its price will increase and this signals to producers to increase production. That is, prices balance supply and demand by coordinating the decisions of producers and consumers in a market. In a market, there is no higher authority to direct the behavior of the market participants. Rather, it is the "invisible hand" of the marketplace that allocates goods and sets the prices. This remarkable property of a competitive market economy was first recognized by Adam Smith. In his book, "The Wealth of Nations" (Smith 1904, Book IV, Chapter 2), he argues that "every individual ... neither intends to promote the public interest, nor knows how much he is promoting it ... he intends only his own security; and by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention". Smith's insight about the functioning of the market mechanism has been an important inspiration to neo-classical economy. It has been shown that with perfect competition and no externalities,⁴ a market achieves maximum economic efficiency in allocating goods and services.

Auctions have been used as the major trading mechanism in markets for many years. In the economics literature, an auction is defined as "a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants" (McAfee and McMillan 1987a, p.701). Auctions are used when there is no standard value for the item to be traded and there is a need for price discovery.

The application of auction models to solve computer science problems is not new. They have been proposed to solve resource allocation problems (Ferguson et al. 1996) and distributed control problems in distributed environments. Concrete examples include resource scheduling in manufacturing information systems (Kutanoglu and Wu 1999); flow control in virtual circuit-based computer networks (Ferguson 1989); providing a QoS guarantee in packet networks (Sairamesh et al. 1995); data migration and replication with the objective of minimizing the transaction response time (Ferguson 1989); air-conditioning control (Huberman and Clearwater 1995); and coordination for robot navigation (Sierra et al. 2000). These problems share several common characteristics that lean them towards being modelled as an auction:

⁴ Externalities refer to the cost and/or benefit of an exchange accruing to a third party not involved in the transaction (Mundt 1993). In a situation where externalities exist, an action of an economic entity affects the utility of another entity, in either a positive or negative way, and there is no mechanism to compensate for the impact (Cornes 1996, p.5). Pollution produced by a manufacturer is a typical example of a negative externality which imposes cost on its neighboring community. An example of a positive externality is improving the appearance of one's property where neighbors also benefit from a nicer view.

- Large number of entities, for example, users, applications, machines
- Heterogeneity of entities
- A changing set of resources and users
- Resources belonging to different organizations
- Satisfaction of users determined by the simultaneous allocation of resources

More recently, auctions have been used to solve practical problems. One successful example is the application of auctions to automate online advertisement trading, also known as *ad auction* or *real-time bidding (RTB)*, in which online advertising space is bought and sold (Edelman et al. 2007; Varian 2007). Ad auction has been employed by major companies including Yahoo, MSN, Google and Facebook to improve the profitability of selling their ad spaces. Traditionally, vendors wanting to sell their ad space would segment the audience into bundles according to some characteristics such as age or behavior and sell those segments at predetermined prices to advertisers interested in buying the ad spaces (Ross 2013). Advertisers would then have to pay a fixed price for each impression⁵ in a bundle regardless of their preferences or the relevance of an impression to them. Ad auctions answered the need for more dynamic pricing mechanisms. Their implementation has significantly improved the price determination for these companies: Weide (2013) demonstrated that, with 49% annual growth, RTB spending will grow from \$2.7 million in 2012 to \$20.8 billion by 2017 in the United States. Europe and Japan are heading in the same direction.

2.3.1 Dynamic Pricing

At the heart of auction theory is the concept of dynamic pricing. Dynamic pricing is defined in contrast with the traditional static pricing in which the sellers fix the price. Pricing has been a difficult business problem especially when it comes to pricing an item which does not have a standard value. Setting the right price for a product or service goes beyond the estimation of the cost and a minimum profit: rather, it is governed by a complex set of variables which, among others, include supply and demand, competitor pricing and the lifecycle of the product. More formally, the price determination problem

⁵ An *impression* (in the context of online *advertising*) is a measure of the number of times an *ad* is seen, whether it is clicked on or not (Wikipedia 2014).

is to price a finished product so as to maximize the total expected revenue over the finite sales horizon (Gallego and Ryzin 1997).

Pricing can be divided into two categories: static (fixed) and dynamic. In a fixed pricing approach, the prices are fixed by the sellers and might be changed in long term based on market fluctuations (Schwind 2007, p.30). Market fluctuations are changes in the complex set of variables, mentioned above. For many products and services, these variables change constantly, making it costly for producers to frequently adjust the prices based on the fluctuations. Static pricing for such products incurs implicit and explicit costs for businesses as they need to:

- 1. Spend time and money to obtain relevant information and make decisions about whether, when and how to change their prices,
- 2. Prepare customers for the new prices through further communication, education and efforts to convince them (Bergen et al. 2003).

In dynamic pricing, the price of a good or service is determined by the market. There are four major configurations of dynamic pricing, depending on the number of buyers and sellers involved (see Fig 2.2):

- One buyer, one seller: negotiation
- One buyer, many sellers: reverse or procurement auction
- Many buyers, one seller: direct auction
- Many buyers, many sellers: exchange

| Buvers | Many | (Direct) Auction | Exchange |
|--------|------|----------------------|-----------------|
| | One | Haggle (Negotiation) | Reverse Auction |
| | | One | Many |
| | | Sellers | |

Fig 2.2. Categorization of dynamic pricing, adopted from Stein et al. (2003)

In the context of our problem, negotiation has been employed by some researchers to solve the composite service selection problem. The idea of negotiation-based service selection approaches is to have automated agents performing negotiation on behalf of a service requester and the service providers in order to reach an agreement on the price and quality of the offered services.

However, to address the inherent complexity in a negotiation process, these approaches need to: (1) rely on simplifying assumptions and straightforward techniques to develop automated negotiators, and (2) address the composite service selection problem at a local level for a single web service, rather than at a global level for a composite service. The fact that only simplified models have been actualized means that the application of realistic automated negotiation techniques for the web service selection problem appears to be unfeasible, at least for the near future. We will present a detailed study of these approaches in section 3.6 (Negotiation-based Approaches).

Our research develops an approach for web service selection based on auction models. Auctions are known to be the most widely used mechanism for dynamic pricing (Bichler 2001). They have been proven to be a success in achieving dynamic pricing and also in solving complex problems with the help of well-established theories from economics and mechanism design.

2.3.2 Auction Categories

Auctions can be categorized based on different attributes. Some of the important categorizations which are meaningful in the context of our problem domain have been summarized in Table 2.1. These categories are not necessarily exclusive; for example, an auction can be multi-attribute, iterative and combinatorial.

- Direct/reverse auction (Bichler et al. 2006): Traditionally, auctions have been used for selling products, that is, *(direct) auctions*. However, the same theory applies when the auctioneer aims to buy some products or services. This is called a *reverse* or *procurement* auction. Procurement auctions are popular mechanisms for supply chain management.
- Single-item/multi-item (combinatorial) auctions (de Vries and Vohra 2003; Blumrosen and Nisan 2007): It is possible to simultaneously auction more than one item which is called a *multi-item* auction. In such an auction, bidders might be allowed to bid only for single items (a *non-combinatorial* auction) or they might be able to express their preferences in more complex ways by bidding for a
combination or bundle of items. This leads to an important category of auctions that has attracted the attention of researchers and practitioners in auction theory and mechanism design during recent years, known as *combinatorial auctions*. We will focus on this auction model in the next subsection (2.3.3).

- Single-dimensional/multi-dimensional auctions (Parsons et al. 2011): In conventional auctions, the bidders only express the price of what they are willing to buy or sell. This is known as a *single-dimensional* auction where the only important aspect of a bid is the price. In a *multi-dimensional* auction, other aspects of the item are also part of the bid, such as the quality.
- Single-sided/double-sided (exchange) auctions (McAfee and McMillan 1987b; Parsons et al. 2011): In a *single-sided* auction, only one side (either seller or buyer) submits their bids and the auctioneer decides the winners of the auction. In a *double-sided* or *exchange* auction, both buyers and sellers submit their bids and the auctioneer's job is to match the buyers and sellers.
- One-shot/iterative auctions (Parkes 2006): A *one-shot* auction consists only of a single round of bidding during which the bidders submit their bids. In an *iterative* auction, there are multiple rounds of bidding. At the end of each round, there is a flow of information from the auctioneer to the bidders about the current status of the auction, for example, the amount of the current winning bid. This information helps the bidders to adjust their bids for the next round. The outcome of the auction will be determined at the end of the last round.
- Single-unit/multi-unit auction (Klemperer 1999): In *single-unit* auctions, there is only one copy of each item being auctioned, whereas *multi-unit* auctions have many copies of the same item being auctioned.

Due to the particular characteristics of our problem domain, combinatorial auctions have a central role in developing a solution for the composite service selection problem. Therefore, we have provided a detailed study of this category of auctions in the next section.

| Basis for Categorizing | Different Categories | |
|--|--|---|
| Sell or buy | (Direct) auction | Reverse or procurement auction |
| Number of items under auction | Single item | Multiple items |
| Simultaneous bidding for multiple items | Single items (non-combinatorial auctions) | Combinatorial auctions |
| Bids information | Single-dimensional | Multi-dimensional or multi-attribute |
| Bidding participants | Single-sided | Double-sided or exchange |
| Rounds in an auction | One-shot or single-round | Iterative auctions or multi-round |
| Number of units under auction | Single unit | Multiple units |

Table 2.1. Categorization of auctions

2.3.3 Combinatorial Auctions

In traditional auctions, as we know them, one item is auctioned at a time. However, in many auctions, bidders care in complex ways about the combination of items that they want to win. Imagine a buyer who wants to purchase a return ticket to a particular destination from an online auction site that sells airline tickets. In the traditional way of auctioning, she has to attend two separate auctions and win two tickets, one to and one from that destination. If she wins in only one of the auctions, she will end up with a one-way ticket which is of no value to her. Such a bidder strongly prefers an auction model that allows her to bid for the two items together as a bundle. In other words, the satisfaction of such a bidder is determined by the simultaneous allocation of the items.

In combinatorial auctions, multiple distinct items are simultaneously auctioned and the bidders can bid for any combination of items, or *bundles*. Bundling is particularly important when bidders have preferences not just for specific items but for bundles due to the complementarity or substitutability effects that exist among the items (de Vries and Vohra 2003).

Two items are said to be substitutes (have substitutability effect) if their combined value is less than the sum of their individual values (Shoham and Leyton-Brown 2009, p.362).

An example of items being substitutes is two tickets to two movies which are shown at the same time.

Complementarity is the opposite effect of substitutability: two items are said to be complementary if their joint value exceeds the sum of their individual values (Shoham and Leyton-Brown 2009, p.362). As an example consider a left shoe and a right shoe. The combinatorial auctions where items to be bundles have complementarity effect have been categorized, based on the types of complementarity dependencies between items, into the following groups:

1. Path in space:

In this class, the bidders are interested in purchasing the connection between two points. The points are equivalent to the items under auction and they are connected to each other if they have accessibility relationship. Examples of auctions with this type of dependency are auctions to allocate truck routes, gas pipeline networks, network bandwidth and right to railway tracks.

2. Proximity in space:

Here, the complementarity arises from adjacency in two-dimensional space. Example of this class of auction includes: sale of adjacent pieces of real states, drilling right (in adjacent lots) and the spectrum auctions (to some extend).

3. Temporal matching:

Here, the complementarity arises from a temporal relationship between items. In the general temporal matching dependency with single quantity items, there are *m* distinct items, and each bidder wants 1 time slice from a set of $j \le m$ items with some constraints over how the times of different items relate to one another. Example includes the auction over airport take-off and landing rights where j=2.

4. Temporal scheduling:

In this class, a bidder has a job, requiring some amount of one or more resources' time, with a deadline by which the job should have been completed. The auction is over the time slots of the resources. Example includes distributed job-shop scheduling with one resource, and also allocating grid resources to the tasks.

5. Arbitrary dependencies:

In this category, the dependencies are due to some kind of regularity in the complementarity relationships between the items. Example includes any auction of different, indivisible goods which have dependencies to each other, such as semiconductor parts, or collectables, or the right to emit some quantity of different pollutants produced by the same industrial process. The combinatorial auction for procuring a composite service is another example in this group (Leyton-Brown et al. 2000).

Bundling of complementary or substitute items allows the bidders to more fully express their preferences which often leads to greater economic efficiency (allocating items to those who value them most) and greater auction revenue (Cramton et al. 2006, p.8).

Combinatorial auctions have been proposed and/or applied for practical applications in various industries. Examples include combinatorial auctions for supply chain management (industrial procurement) (Chen et al. 2011); procurement of school meals (Olivares et al. 2012); procuring transportation (logistics) services (Sheffi 2004; Srivastava et al. 2008); allocating bus routes to private operators (Cantillon and Pesendorfer 2006); allocating airport arrival and departure slots to competing airlines (Rassenti et al. 1982); and resource allocation in the cloud (Zaman and Grosu 2013).

Combinatorial auctions can be either direct or procurement auctions. In the direct combinatorial auction, there are multiple items or service for sale. While in the combinatorial procurement auction, there is a buyer who is interested in a combination of products or services and the sellers bid to provision these products or services.

In practice, combinatorial procurement auctions have been successfully applied by online platforms for industrial procurement. Examples of sourcing companies who have implemented combinatorial procurement auctions for strategic sourcing and supply chain include Logistics.com⁶, CombineNet now part of SciQuest⁷, and TradeExtensions⁸. The motivations behind designing a combinatorial procurement auction have been described as:

⁶ <<u>http://www.logistics.com/</u>>

⁷ <<u>http://www.sciquest.com/</u>>

⁸ <<u>http://www.tradeextensions.com/</u>>

- Cost saving: Combinatorial bids represent the complementarity or substitutability effects among the items which lead to production and/or transportation cost savings for the bidders. This eventually improves the procurement cost for the bid-taker.
- 2. Time efficiency: Combinatorial bidding allows all sides to instantaneously express their complex preferences for the items through package bids and saves them from the need to attend different auctions to attain what they want.
- 3. Impacting on the market structure: In complex procurement scenarios with many items to be procured, refusing the combinatorial bids will restrict the competition to only big suppliers who are able to offer all the items. On the other hand, combinatorial auctions allow the splitting of a big contract into smaller parts thus making it possible for smaller suppliers to enter the competition. This, in turn, can lead to more cost saving for the bid-taker (Bichler et al. 2006).

The bid-taker of the combinatorial auction (buyer in a direct auction or the seller in a reverse or procurement auction) receives a set of price offers for various combinations of auctioned items and faces the problem of choosing the set of offers which maximizes the bid-taker's revenue or economic efficiency, as will be discussed in more detail in subsection 2.4.2.2. This problem, known as the winner determination problem (WDP), is NP-complete in the general case and intractable (Sandholm 2002). We will discuss this aspect of combinatorial auctions later in subsection 7.2.1, when discussing the limitations of our proposed auction-based approach for composite service selection.

2.4 Mechanism Design

The 2007 Nobel Memorial Prize in economic sciences was awarded to Leonid Hurwicz, Eric Maskin and Roger Myerson "for having laid the foundations of mechanism design theory" (Royal Swedish Academy of Sciences 2007). The Nobel Prize was awarded for their work spanning 50 years in this field.

Mechanism design is a sub-field of microeconomics and game theory that has an engineering perspective. In this field, the focus is on problems involving multiple, rational, self-interested players with private information about their preferences. The objective of mechanism design is to design a mechanism to achieve a given outcome. This objective is the reverse of the focus in traditional economics and game theory which

is to analyze the performance or outcome of a given mechanism. That is the reason mechanism design is sometimes called *reverse game theory*. The basic question in mechanism design is how to design an economic system so that the selfish behavior of participants leads to "desirable properties".

The importance of mechanism design is that it studies real problems and it has found real applications. Recently, mechanism design has found several important applications in electronic market design, distributed scheduling problems and combinatorial resource allocation problems. Mechanism design has also laid the principles for designing auctions.

2.4.1 Auction Properties

Here, we introduce the important set of desirable properties for auctions. Auction designers may consider any subset of the following desirable properties for the auction to achieve. These properties are neither mutually exclusive nor it is always possible to achieve them simultaneously, that is, some of these properties conflict. In this regard, we introduce an important impossibility theorem, proven by Myerson and Satterthwaite (1983), which we will refer to for the analysis of our proposed auction models in subsections 4.3.1.3 and 7.2.2.5.

Please note that in the discussion of these properties, the assumption is that participants have quasi-linear utility or quasi-linear preferences in the form of $u_i = v_i(a) - p_i$, where u_i is the utility of participant *i* after attending the auction. Also, v_i is the valuation function of participant *i* that maps the possible outcomes of the auction (the allocation of the items to bidders), *A*, to real numbers, that is, $v_i: A \to R$. The quantitative value $v_i(a)$ represents the value that participant *i* assigns to the outcome $a \in A$. We also define the monetary transfer by participant *i* to/from the auction as $p_i \in R$. If $p_i > 0$ then participant *i* will pay some money, and if $p_i < 0$ then participant *i* will receive the money.

Having a quasi-linear utility function means participant i's utility is their value for the auction outcome less their payment. Assuming a quasi-linear utility for the participants implies three important things:

1. The mechanism can charge the participants or award them some monetary amount.

- 2. A participant's degree of preference for any outcome is independent from their degree of preference for having to pay some amount to the mechanism. This means that a participant's utility for an outcome is independent from the amount of money they have in their pocket.
- A participant's utility depends only on their own monetary transfer, and they do not care about the money paid or received by other agents (Shoham and Leyton-Brown 2009, p.280).

2.4.1.1 Efficiency

In auction theory, the design of an auction aims to achieve either of the two objectives: efficiency or revenue maximization (Parsons et al. 2011; Ausubel 2003). These two objectives are not mutually exclusive, yet, it is not always possible to achieve them simultaneously.

An efficient auction design is concerned with achieving allocative efficiency, that is to maximize the total value over all bidders (Parsons et al. 2011). Here, the goal is to design a mechanism that collects the participants' reported valuation functions and, based on these reports, selects an outcome a^* that maximizes the total value. More precisely: $a^* \in argmax_{a \in A} \sum_i v_i(a)$. This efficiency is referred to as the ex post efficiency as it is tested at the ex post stage of the auction.

This is achieved when the items under auction are put in the hands of those who value them most (Ausubel 2003). In the mechanism design language, the total value of the outcome to all the participants is also referred to as *the social welfare*, and therefore, this objective is also known as maximizing the social welfare.

As an example, consider an auction of one item where the bidders' valuation functions are $v_i: \{T, F\} \rightarrow R$, from either receiving (*T*) or not receiving (*F*) the item, and for all participants, the valuation of not receiving the item is zero: $\sum_i v_i(F) = 0$. As a result, the social welfare is equal to the valuation of the one bidder who has won the item. This means that in order to maximize the social welfare, the item should be awarded to the bidder who values it most. A well-known efficient auction is the second-price (Vickrey) auction for a single item where the bidder with the highest bid wins the auction, but pays the second highest bid (Vickrey 1961). This auction will be discussed further in subsection 2.4.2.3. When an auction is designed with the objective of achieving allocative efficiency, it means that the auctioneer is assumed not to be seeking their own benefit. Rather, the auctioneer aims to maximize the happiness (social welfare) across all participants. An example can be a government interested in maximizing the social welfare (Shoham and Leyton-Brown 2009, p.300). Even in the case of private companies, they might decide to maximize allocation efficiency rather than to maximize the profit on the basis that the long-term relationship with customers and customers' loyalty are more important than the short-term profit (Karlin and Peres 2014, p.272).

2.4.1.2 Revenue Maximization (Cost Minimization)

The other objective function discussed in the literature is to maximize the revenue for the seller in a direct auction, or to minimize the cost for the buyer in a reverse (procurement) auction. Auctions with this objective are called optimal auctions and were first studied by Myerson in his seminal paper (1981).

In general, allocation efficiency and revenue maximization are in conflict (Ausubel and Cramton 1999; Parkes 2001). A broad research area in auction theory has focused on studying the relationship of these two objectives and how to design auctions that can simultaneously get close to the two objectives as much as possible, such as the analysis performed by Aggarwal et al. (2009).

Revenue maximization seems to be a natural choice in auction design as the assumption is that all participants, including the auctioneer, are self-interested entities who are interested in maximizing their revenue rather than efficiency. This objective is more likely to be pursued if the auctioneer (or the bid-taker) benefits from high market power, for example, due to being a monopoly. However in reality, even private companies aim to maximize their profit in long-term rather than a single auction which means that their objective includes both revenue maximization and economic efficiency (Likhodedov and Sandholm 2003).

2.4.1.3 Budget-balanced

An auction is said to be budget-balanced if, regardless of participants' types (preferences over possible outcomes), the auction collects and distributes the same amount of money from and to the participants (Shoham and Leyton-Brown 2009, p.286). In other words, no net payment "into" or "out of" the system is required (no funding or subsidy, nor the

loss of money). More precisely, the auction is (strongly) budget-balanced if, regardless of participants' types, we have: $\sum_i p_i = 0$.

There is a weaker sense of budget-balanced where it is possible to have a net payment from participants to the mechanism, but no net payment from the mechanism to the participants. In other words, the auction will never take a loss, but it may take a profit (Shoham and Leyton-Brown 2009, p.286). This is called a weakly budget-balanced auction and we have: $\sum_i p_i \ge 0$.

Budget-balanced is an important property to consider in an auction design, as there is little incentive to run an auction that loses money or needs to be subsidized.

2.4.1.4 Pareto Optimal

An auction is Pareto optimal if its allocation rule achieves a Pareto optimal outcome. An outcome is said to be Pareto optimal if there is no other outcome that will make at least one participant better off without making at least one other participant worse off (Pardalos et al. 2008, p.482). In other words, with a Pareto optimal outcome, no participant can be made happier without making at least one other participant less happy. Accordingly, an outcome is not Pareto-efficient if there is another outcome that can make at least one participant happier while keeping everyone else at least as happy as in the non-Pareto-efficient outcome. More specifically for the context of auction design, an auction is proven to be Pareto-efficient if and only if it is allocative-efficient and budget-balanced (Garg et al. 2008; Parkes 2001).

2.4.1.5 Individual Rationality

Another important property of an auction is individual rationality, also known as voluntary participation constraint (Parkes 2001, p.34). It is about the extent of the willingness of participants to attend the auction, based on the information they have about the auction rules. More precisely, this property defines the level of expected utility that a participant can receive by attending the auction and generally requires that the utility of the participants be greater or equal than their utility before participation in the mechanism (Shoham and Leyton-Brown 2009, p.286). This implies that, in an auction with the individual rationality property, the participants prefer to attend the auction and are not forced to do so.

The most natural form is called interim individual rationality. It states that no individual wants to leave the auction when they know their own type, but only have expectations over the other individuals' types and the auction outcome in terms of allocations and payments (Jackson 2003). In other words, participants expect a positive utility from attending the auction: $E[u_i] \ge \overline{u}_i$, where u_i is the utility of participant *i* after attending the auction, and \overline{u}_i is the utility that participant *i* receives by withdrawing from the auction (or not participating in the auction at all).

The strong form is called ex-post individual rationality and it requires that no individual participant wishes to walk away from the auction after the auction outcome is fully specified at the ex-post stage of the auction (Jackson 2003). More precisely: $u_i \ge \bar{u}_i$.

2.4.1.6 Incentive Compatibility

Incentive compatibility respects the fact that participants have private information about their valuation for the items under auction which they might or might not report truthfully. An auction is said to be *incentive compatible* (aka *truthful*, or *strategy proof*) if, for each participant *i*, truthful reporting maximizes their utility, regardless of other participants' choices. In this setting, truthfulness is said to be the dominant strategy for participant *i* (Jackson 2003).

In the real world, selfish participants tend to strategically report their value untruthfully, in order to maximize their own utility. Therefore, having a truthful mechanism does not seem to be very realistic. The immediate question that comes to mind is then "why has incentive compatibility been an important auction property to study throughout the mechanism design research?"

To answer this question, we need to go back to the principal question that mechanism design tries to answer: "how can an economic system be designed so that the selfish behavior of the participants leads to desirable properties?". Suppose that the goal of the auction designer is to achieve an outcome that maximizes the social welfare, that is, the cumulative value of the outcome to all participants. The designer should design an auction that collects the bidders' reported valuations, which might be truthful or not, and selects an outcome (an allocation of items to bidders) and a payment function so that the social welfare is maximized. This means that the designer needs to analyze all possible strategic behaviors of bidders in order to predict the outcome. Analyzing such a complex space is

not easy, if at all possible. However, there is a simple, yet extremely useful principle in mechanism design called *the revelation principle* that simplifies the design of auctions.

The revelation principle (Gibbard 1973; Myerson 1979; Myerson 1981) states that under quite weak conditions, there is an equivalent direct⁹ incentive-compatible auction for any arbitrary auction that achieves the same outcome. This principle made it possible for researchers to focus exclusively on direct incentive-compatible mechanisms to do the theoretical analysis of what is possible (*possibility results*) and what is not (*impossibility results*) in the space of direct mechanisms.

The design problem thus becomes easier for the auction designer. If they want to achieve a set of desirable outcomes, they need to install the right incentives for the bidders to make them behave in a certain way that will lead to these outcomes. Giving the right incentives is usually done by designing the right payment rule which will be further discussed in subsection 2.4.2.3.

2.4.1.7 Myerson–Satterthwaite Impossibility Theorem

As we explained before, some of the discussed desirable properties cannot be achieved simultaneously. One such set is recognized in the *Myerson–Satterthwaite Impossibility theorem* (Myerson and Satterthwaite 1983). The impossibility result states that even in a simple bargaining problem between one seller and one buyer for a single item, there is no allocation mechanism that is incentive compatible and (interim) individually rational that can achieve (ex post) efficiency without outside subsidies (being budget-balanced). The assumption is that both buyer and seller's valuation for the item is independent private value (IPV), that is, they have a private value for the item which is not known to the others and is also independent from the valuations of others for the item (Parsons et al. 2011).

As a result, the auction designer has to trade off some of the properties to achieve the others in the auction. As individual rationality seems to be the principal requirement in many auction settings (if the bidders are not to be forced to attend the auction), the trade-

⁹ In mechanism design, a direct mechanism is one in which the only action available to the participants is to announce their private information (or their types) (Shoham and Leyton-Brown 2009, p.277). With this definition, we can see that a sealed-bid auction is a direct mechanism, but an iterative auction is an indirect one.

off is usually done between the three other properties: incentive compatibility, allocation efficiency and budget-balanced.

2.4.2 Auction Design Elements

Any auction designer needs to make decisions regarding three aspects of an auction, which we have called auction design elements. Many researchers, such as Wurman et al. (1998) and Bichler et al. (2006), have recognized these aspects, although under different names. These elements are:

- 1) *The auction protocol* which may be referred to as *the bidding language*: The syntax, semantics and sequence of message exchange in the auction.
- Allocation rule(s), also known as the winner determination problem (WDP) in single-sided auctions, or the matching algorithm in double-sided auctions: It specifies how the item(s) will be allocated to bidders.
- Payment rule(s), also referred to as the pricing model/scheme or the incentive implications: The payment to (from) bidders in a direct (reverse) auction is determined here.

2.4.2.1 The Bidding Language (The Auction Protocol)

The auction protocol determines the structure of the messages sent from the bidders to the auctioneer (the bid), the structure of the information feedback sent from the auctioneer to the bidders, and the sequence of the message exchange between the auction's participants.

In the simplest form, the bid structure contains a price offer (or request) to buy (or sell) a single item under auction. The simplest feedback from the auctioneer to the bidders can be a signal indicating whether the bid is accepted or rejected. This is the auction protocol followed by sealed-bid auctions for a single-unit item. In the first round, the bidders submit their bids, and in the next round, the winner is determined.

This simple protocol can be extended along different aspects. The first aspect is along the bidding language. If the auction's requirements dictate that the bid structure needs to transfer more information about the bidder's offer rather than simply the price, then there should be enough support in the bidding language. For example, in a procurement auction, the bidders might also need to specify the quality level for their offers, or the delivery

time. A bidding language supporting such a structure is called a multi-attribute bidding language.

In some auctions, more than one item is auctioned simultaneously. If items under auction are homogeneous (multiple copies of the same item), bidders should be able to specify in their bids the number of items that they intend to buy or sell. On the other hand, when the items are heterogeneous, bidders might prefer to bid for a combination of the items, or bundles. Multi-unit auctions and combinatorial auctions are the auctions that correspond to these two settings. The bidding language should have enough support for each setting to enable bidders to more fully express their preferences.

In combinatorial auctions, the auction designer might decide to limit the bidding possibilities to prevent communication problems. The issue is that in a combinatorial auction with many items under auction, if bidders are allowed to bid on any combination of items, the number of possible bid submissions can theoretically grow exponentially. This can lead to a huge amount of message exchange between bidders and the auctioneer, significantly affecting the decision-making process involved in the winner determination problem. Therefore, the auction designer might limit the number of bid submissions or the number of items in a bid or impose a special structure (sequence) for the items in a bundle.

Apart from the structure of the bid, the auction designer needs to specify the number of bids that each bidder can submit. If a bidder is allowed to submit more than one bid, the designer needs to specify what the relationship between the bids can be, in other words, how the bids can be combined. This relationship can be one of the following types:

- **OR bids**: With the OR combination of bids, the bidders can submit several bids and they may have any subset of their bids being a winner. The designer might restrict the number of bids in an OR phrase to control the communication effort.
- **XOR bids**: In XOR combination, the bidders can submit several bids. However, they can have at the most one winning bid, that is, the bids are mutually exclusive.
- Combination of OR and XOR bids: It is also possible to allow the bidders to have both OR and XOR types in their bid submissions. Such a language is strongly

expressive; however, the complexity of the language and the communication effort are challenging for bidders and the auctioneer.

OR* bids: This type simultaneously has the simplicity of the OR type and the expressivity of the XOR type by using the concept of dummy items in an auction. The bids are combined by OR; however, if a bidder needs to have an XOR relationship between two bids, they can add the same dummy item to both bids. With the restriction that no item can be won more than once, the dummy items simulate the XOR relationship (Nisan 2000).

The design of the auction protocol is based on the problem domain requirements. Meanwhile, in designing the bidding language, the auction designer specifically needs to consider the trade-off in designing any language which is the expressiveness of the language versus its simplicity.

The information feedback in the simple auction protocol can also be extended when the auction is iterative. In these auctions, the designer needs to specify how much and what information to reveal at the end of each round of bidding. Examples of such information include information about the current winning bid, the current winner's identity and the identities of those who have dropped out. Decisions regarding the information feedback significantly affect the competition among the bidders which, in turn, impacts on the auctioneer's revenue and/or the economic efficiency of the auction. Due to its importance, the study of information feedback has been central to the analysis of multi-round auctions for decades.

2.4.2.2 Allocation Rules (Winner Determination Problem)

The set of allocation rules, more famously known as the winner determination problem (WDP), determines which bidders have won and how the items are allocated to them. The auction designer needs to consider two main components here: the auction objective and the set of allocation constraints.

As discussed in subsections 2.4.1.1 and 2.4.1.2, the two popular objectives that an auctioneer might have are economic efficiency and revenue maximization (de Vries and Vohra 2003). The two objectives are not mutually exclusive; however, it is not possible

to achieve both at the same time in all situations. The auction designer needs to formulate the auction's objective based on the problem domain requirements.

The second component of the allocation rule is the set of allocation constraints. There are several types of allocation constraint that the auction designer might need to add to the WDP formulation. One well-known constraint concerns the reservation price of the seller (direct auction) or budget constraint of a buyer (reverse auction).

There are other allocation constraints which are important in reverse auctions. One such constraint is in relation to the quality of the items to be procured. The quality constraint might be demanded at the level of "a single item" or at the level of "a set of items". For example, the auctioneer might be interested in the delivery time of all items under auction collectively, and not in the individual items' delivery time.

Another important constraint in the context of procurement auctions is about managing the extent of the dependency on suppliers. The two extremes are: (1) procuring all items from as few suppliers as possible, and (2) selecting as many different suppliers as possible. The first case might lead to a high exposure if some of the suppliers are not able to deliver their promise, and the second case can create a high overhead cost of managing too many suppliers (Bichler et al. 2006). Therefore, the auction designer might set limitations on the minimum and maximum number of winning suppliers to avoid depending on too few suppliers or on too many of them. This constraint is also referred to as *the market share* as the designer aims to control the share of the market won by the individual suppliers.

2.4.2.3 Payment Rule (Pricing Scheme)

In the familiar form of auctions, the winners usually have to pay their bid as the price to obtain the item. However, for many auctions, that is not the case. In mechanism design, the pricing scheme is a strong measure employed by mechanism designers to install their desirable properties in the auction. Through the payment rules, they install the appropriate incentives for the mechanism's participants to guide their behavior in a certain way that will lead to the mechanism's desirable properties.

One of the important desirable properties in auctions is incentive compatibility or truthfulness, discussed in subsection 2.4.1.6. Based on the revelation principle (Gibbard 1973; Myerson 1979; Myerson 1981), a mechanism designer is able to study the direct

incentive-compatible equivalent of any arbitrary mechanism to predict the possible outcome. Therefore, the focus of researchers in mechanism design is to design payment rules which lead to incentive-compatible mechanisms.

For a single item, the well-known truthful mechanism is the Vickrey auction, also known as the sealed-bid second-price auction. In this auction, the bidder with the highest bid wins the auction, but they will only pay the second highest bid as the price for the item. For example, if the highest bid is \$10 and the second highest bid is \$8, the bidder with the \$10 bid wins, but only pays \$8. First analyzed through a game-theoretic approach, William Vickrey (1961) demonstrated that in this auction, bidders with independent private values (IPV) cannot increase their utility by manipulating the declared valuation. Therefore, their dominant strategy is to be truthful and bid their true value for the item.

Despite being very popular among researchers in auction theory, the Vickrey auction did not find much practical application (Ausubel and Milgrom 2006). One important factor contributing to this unpopularity in practice is the possibility of very low revenue for the auctioneer even in the case of high competition among the bidders for high-valued items. McMillan (1994) describes a second-price auction for radio spectrum held by the New Zealand government in 1990 which led to embarrassing results. In one extreme case, despite an existing buyer bidding NZ\$100,000 for a license, the final price paid was the second-highest bid of NZ\$6.

The Vickrey auction was later expanded by Clarke (1971) and Groves (1973) to a more general competitive process where multiple items with interdependent values are auctioned. The extension is known as the Vickrey–Clarke–Groves (VCG) mechanism and is both economically efficient and incentive-compatible.

The allocation rule in the VCG mechanism assumes that the received bids are truthful, and therefore, chooses an allocation that maximizes the economic efficiency based on the received bids. The designed payment rule ensures that the bidders have no incentive to not tell the truth. The payment to each winning bidder, i, has two parts:

• a payment made by bidder *i* to the mechanism for winning the item(s) which is equal to the sum of the values of all winning bids if bidder *i* does not participate in the auction. This is calculated by taking bidder *i* out from the competition, and finding the best allocation without their presence,

• a refund paid by the mechanism to bidder *i* to incentivize truthfulness which is equal to the sum of the values of all winning bids except that of bidder *i*.

As an example, consider an auction to sell three items, A, B and C, where we have four bidders with these bids: bidder₁={A, \$20}, bidder₂={B, \$10}, bidder₃={BC, \$80} and bidder₄={ABC, \$80}. The winners will be bidder₁ and bidder₃ with a total value of \$100. The amount that needs to be paid by bidder₁ is 90 - 80 = 10, rather than \$20. The first amount is calculated by taking bidder₁ out from the competition and determining the total value of the winning bids again, and the second amount comes from the sum of the current winning bids without the bid from bidder₁. Similarly, bidder₃ has to pay \$60 (\$80 - \$20) instead of his original bid of \$80. The reduction in the winning bidders' payments is in fact a concession awarded to them by the mechanism for being truthful.

In the VCG mechanism, both parts in the payment function to a bidder are independent of the bidder's declared value. Therefore, bidders have no incentive for strategically manipulating their declared valuations, as the manipulation will not increase their gained utility. The mathematical proof can also be found in Nisan et al. (2007, p.219).

Similar to the second-price sealed-bid auction, the VCG mechanism is "lovely in theory but lonely in practice" (Ausubel and Milgrom 2006). There are a number of serious limitations that have prevented the VCG payment formulation from finding practical applications including (Ausubel and Milgrom 2006; Rothkopf 2007):

- Too complex to compute when the number of bidders is too high,
- Reveals a lot of information,
- Possible to have very low-revenue outcomes,
- Highly susceptible to collusion,
- Requires unlimited budgets (for the bidder side in a direct auction) or else it will cause problems (it will be complex for a bidder with budget constraints to determine how to bid).

2.5 Conclusion

In this chapter, we presented the basic concepts underlying our research along three main pillars: the web service technology, auction theory and mechanism design. In web service technology, we discussed the characteristics of web services, the importance of web service composition and introduced the lifecycle of web service composition. The proposed lifecycle decomposes the WSC into five stages, positioning our research in the second stage, as web service selection.

Then, we presented the broader picture on markets before focusing on auctions as the most widespread allocation mechanism in markets. We discussed dynamic pricing as the central pricing mechanism in auctions. Following this, a general categorization of markets was presented. Among these categories, we discussed combinatorial auctions in more details as the basis of our proposed mechanism for composite web service selection.

Lastly, we discussed the mechanism design approach in designing an auction and an important set of desirable properties that auction designers consider when designing an auction. We also discussed the essential elements in designing an auction, namely, the auction protocol, the allocation rules and the pricing scheme.

Chapter 3

3 Literature Review

3.1 Introduction

In this chapter, we present a comprehensive review of the current literature on composite web service selection approaches. In section 3.2, the research problem, composite web service selection problem, is discussed in more details. In section 3.3, we examine the main challenges and issues involved in this research area.

We have categorized composite service selection approaches to three main categories: optimization-based approaches, negotiation-based approaches and auction-based approaches. We discuss the basis of our categorization in section 3.4. The extant literature on each of these categories are discussed in sections 3.5 (optimization-based approaches), 3.6 (negotiation-based), 3.7 (hybrid of optimization and negotiation approaches) and 3.8 (the emerging auction-based approaches).

Parts of this chapter have been previously published as a book chapter in (Moghaddam and Davis 2014). It has been updated with more recent studies to be included in this thesis. The last two challenges on the list and the literature on auction-based approaches are presented here for the first time.

3.2 Research Problem: Composite Web Services Selection

As briefly introduced in the introduction chapter, section 1.1, composite web service selection refers to the problem of selecting an optimal set of web services that can collectively achieve a specific complex functionality, from the pool of available web services. The collaboration and execution of this set of web services forms a *composite web service* and the user who needs the composite service to achieve the complex functionality is known as the *composite service requester*. To achieve the complex functionality, web services need to be executed in a specific sequence. The specification of the right sequence can be defined at a high level, as an abstract business process (BP).

As discussed in subsection 2.2.3, the business process comprises a set of tasks, each with clear functionality, along with the control and data flow among them. To create the concrete composite service, the service requester needs to find (at least) one concrete web service to execute each task.

Moreover, the selected web services need to satisfy the *preferences* and *constraints* of the composite service requester about the quality of service (QoS) and cost of the composition. The quality of a service is defined by a set of quality of service attributes (such as the response time, availability and reputation) that are important and relevant to that web service (to be discussed in more details in subsection 3.4).

These preferences and constraints may need to be addressed at two levels: at the level of a single service and at the level of the composite service. For example, the requester may require that the response time of a specific service does not last longer than a maximum threshold for the input data of the service to remain valid. Imagine a web service which receives the geographic location of a moving vehicle and displays its location on a map. If the execution of this service takes long, the displayed position will not be valid. At the same time, the requester may need the end-to-end response time of the composite service not to take longer than a specific time for the composite service to be useful.

As discussed in subsection 2.2.3 (WSC Lifecycle), composite service selection is a critical stage in the web service composition process (WSC). However, it is worth mentioning that composite service selection might be a problem in areas other than WSC. For example, in many scientific workflows, it is required that a combination of different types of services be selected to realize the workflow, such as a combination of web services and legacy software applications. In this case, the composite web service selection is meaningful for that part of workflow where web services are required.

Regardless of the context of the composite service selection problem, its objective is to find the "optimal" web services based on the service requester's preferences and constraints over different criteria such as quality of service and price.

3.3 Web Service Selection Challenges

Web service selection is considered to be a complex problem. The complexity partly arises from different challenges that researchers in this area have to confront with. Some of these challenges have already been recognized and received the attention of the research community. These are:

- 1. The NP-hardness of the service selection problem and scalability,
- 2. The need to distinguish the abstract business process from its possible set of execution paths,
- 3. Defining the aggregation functions for the QoS attributes to measure the end-toend quality of the composite service,
- 4. Elicitation of the importance of the different QoS attributes from the service requester's perspective (Moghaddam and Davis 2014).

However, some important issues have not received the attention they deserve from the web service research community, including:

- 5. Existing dependencies between constituent services of a composition,
- 6. Price determination for web services,
- 7. The presence of multiple requests for composite services.

In this section, we discuss these challenges and how they have been addressed by the current literature on composite web service selection.

NP-hardness and Scalability:

Composite service selection can be modelled as a Multi-dimensional Multi-choice Knapsack Problem (MMKP) (Yu et al. 2007), which is known to be an NP-hard problem in the strong sense (Parra-Hernandez and Dimopoulos 2005). This means that there is no polynomial time algorithm to find the optimal solution. Such a limitation can especially complicate finding an optimal solution for large problem instances.

As a consequence, there is a need for heuristic approaches when the problem size is too large to be solved by optimal solution procedures (Parra-Hernandez and Dimopoulos 2005). A number of heuristic algorithms have been proposed in the literature to find near-optimal solutions. Good exemplars include (Yu et al. 2007; Berbner et al. 2006; Menascé et al. 2010).

Some researchers have proposed a Genetic Algorithm approach to solve the scalability problem (Canfora et al. 2005; Jaeger and Muehl 2007; Ma and Zhang 2008). An alternative proposal to reduce the computational time of the service selection search algorithm is to shrink the search space. For instance, Alrifai et al. (2010) has proposed pruning the service candidates that are not likely to be part of the optimal solution, by computing the service skyline for each service class.¹⁰

From Business Process to Execution Path:

In the current approaches to composite service selection, the assumption is that the required composite service is described as a high-level business process (Ardagna and Pernici 2007). Different languages and models have been used for describing the business process, such as UML activity diagram (Ardagna and Pernici 2007), statechart (Zeng et al. 2004), extended BPEL (Agarwal and Jalote 2010), or YAWL (El Haddad et al. 2010).

Regardless of the modelling notation, different control-flow constructs are allowed in the existing process modelling languages such as *sequence*, *loop*, *parallel execution* and *conditional branching*. For some of these structures, such as loop and conditional branching, the runtime structure is different from the abstract structure. For instance, only one of the tasks in the conditional branching would be selected for execution. This means that a BP might be executed along different paths, based on the control-flow at runtime.

Each possible path of BP execution is called an *execution path* or *execution flow*. During service selection, an *execution plan* is created by assigning web services to the tasks of

¹⁰ For a set of d-dimensional data points, the skyline is a subset of the points where no point in it is dominated by any other member. If $\vec{p}(p_1, ..., p_d)$ and $\vec{q}(q_1, ..., q_d)$ are two points in the d-dimensional data set, p dominates q iff $\forall i \in [1, d], p_i \geq q_i$ and $\exists j \in [1, d], p_j > q_j$ (Yu and Bouguettaya 2012). The notation \geq is defined as being *better than or equal*, and \geq as *better than*. In the service domain, a *service skyline* is the set of providers where no provider is dominated by any other, in terms of the offered values for QoS attributes.

an execution path. There is a need for special consideration to these constructs and how they affect the execution path of a BP at run time.

In composite service selection, it is essential to distinguish the abstract BP and its possible set of execution paths. Researchers have used different techniques to translate a BP to its corresponding execution paths, such as *loop peeling* (Ardagna and Pernici 2007), or *loop unfolding* (Yu et al. 2007; Zeng et al. 2004) to treat loop structures. In the former approach, every loop is annotated with the expected maximum number of its iterations, considering a probability distribution for the number of loop iterations. In the latter case, the loop is unfolded by cloning the functions in the loop for a number of times such as the maximal loop count, which can be obtained from process execution history or the process designer.

Aggregation Functions:

A critical challenge in service selection is how to measure the end-to-end quality of the composite service. The aggregated value of a QoS attribute is calculated based on:

- 1. The QoS attribute value of the individual services participating in the composition,
- 2. The business process structure.

For example, the overall price of a composite service is defined as the sum of the prices of the participating services. However, the execution time of the composite service needs a more complex aggregation function, for example one that returns the maximum execution time among the parallel services, adds up the execution times of sequential services, and combines these two values if there are both parallel and sequential structures in the BP.

In Jaeger et al. (2004) and its extension Jaeger et al. (2005), the authors have proposed aggregation functions for some QoS attributes such as execution time, price and throughput, supporting a comprehensive set of structural patterns that can be found in workflows. Zeng et al. (2004) has proposed aggregation functions for attributes such as execution price, execution duration, and reputation. To define the aggregation function for non-sequential structures of some of the attributes, they rely on the Critical Path Algorithm from scheduling algorithms and project planning (Pinedo 1995, p.115). In a project network, a critical path is the set of connected tasks that together will take the

longest to complete. Based on this definition, in a business process which has multiple execution paths, the path with longest execution time is the critical path. Aggregation of some quality attributes such as execution time, availability and successful execution rate are calculated with respect to the critical path of the BP. Similar aggregation functions have been developed by (Canfora et al. 2005; Ardagna and Pernici 2007; Richter et al. 2012; He et al. 2014).

The approach based on defining a critical path for the composite service execution helps to define linear aggregation functions or functions that can be easily linearized, which is required for some optimization techniques such as linear programing. We'll discuss this requirement in more detail in subsection 3.5.2.1.

Defining the Weights of QoS Attributes:

In the composite service selection literature, it is generally assumed that the service requester has a clear idea of the importance of a QoS attribute with respect to other attributes and the potential trade-off in achieving them. Such understanding lets the requester to assign a scalar weight to each QoS criterion. However, such an assumption may not be realistic, especially as the number of QoS attributes involved in the selection criteria increases.

Some researchers have tried to address this problem. Wang (2009) has proposed a resolution process for determining the linguistic weights of QoS criteria based on a group of participants' preferences. Yu and Bouguettaya (2012) has proposed two algorithms for calculating the service skyline. In general, determining the skyline of a set of data requires pair-wise comparison of all the members of the data set which can be very expensive in terms of computational time and memory usage. However, the algorithms proposed by Yu and Bouguettaya (2012) exploit the indices of the service operations to compute the skyline more efficiently. The computed skylines guarantee to include the best user desired service providers without any user intervention, that is, no need to define weights for quality attributes.

Dependencies between Constituent Web Services of a Composition:

In the web services domain, services combined to form a composition are dependent on a variety of different factors. These dependencies include: Execution time dependency: Participant services in a composition need to be executed in a specific sequence to achieve the high level goal of the composite service. The time dependencies are specified in the abstract business process, through the control flow specification.

There is a lower level execution time dependency that might exist between the operations of the same service where an operation might need to be executed before another one. This is referred to as the behavior of a service and is a type of intra-service dependency (Yu and Bouguettaya 2008). However our focus here is on the different types of dependencies between different web services or what is known as the inter-service dependency.

- Data dependency: This dependency exists when (part of) the output of one service is consumed as (part of) the input of another service (Milanovic and Malek 2004). This dependency is also known as input/output dependency (Omer and Schill 2009) or message dependency (Yang and Papazoglou 2002).
- Dependencies driven by different types of constraints: such as technical constraints (Ai and Tang 2008), technological constraints (Aggarwal et al. 2004), business constraints (Aggarwal et al. 2004), domain related dependencies (Verma et al. 2004), and user constraints (Omer and Schill 2009).

As an example of the technical constraint dependency, consider a composite service where a document is encrypted using a specific encryption algorithm at one step. Then, it might become necessary to ensure that the document can be decrypted by a compatible service in subsequent steps (Verma et al. 2004). Here, the two encrypting and decrypting web services are dependent based on a technical constraint.

The identification, automatic discovery and modelling of the inter-service dependencies have been important research problems for the composite web service community due to their applications in areas such as:

 Dynamic selection of a web service for a process, mostly considering the message dependency among partner services as discussed in works such as (Korhonen et al. 2003; Aggarwal et al. 2004; Verma et al. 2005; Yang and Papazoglou 2002)

- Monitoring the composite service with the objective of failure diagnosis and recovery as discussed in the papers by Wassermann and Emmerich (2011) and Bodenstaff et al. (2008)
- SLA management tasks including the creation, negotiation and handling of the SLA violations, as in the work by Winkler et al. (2010)

While these applications mostly incorporate the point of view of composite service requesters (or end-users which may be different to the requester) and their satisfaction from the composite service execution, the inter-service dependencies can also affect the providers' preferences about offering web services.

A provider who can offer services for a set of consecutive tasks in the business process might be able to offer a discount if the service requester buys these services as one bundle. Bundling can help the service provider internalize some of the costs related to interface compatibility required for data exchange between offered web services, which leads to the possibility of decreasing the cost of service provisioning. Bundling may also allow service providers to improve the quality of bundled services with a competitive price (He et al. 2014). For example, when bundled services are executed on the same machine, the provider can guarantee a lower execution time for the set of offered services. The competitive price offers for bundles, in turn, can improve the provider competitive power in the web services market. Increasing the consumer loyalty is another advantage of bundling for providers (Herrmann et al. 1997).

As discussed earlier in subsection 1.2.1, the dominant assumption in composite service selection approaches is that web services are offered as independent entities with no dependencies. Ignoring these dependencies has led to somewhat unrealistic formulation of the composite service selection problem in current approaches. Moreover, such a formulation cannot achieve a solution that is in the best interest of the service requester or the providers without allowing them to express their preferences and constraints over combinations of dependent web services forming a composite service.

Price Determination for Web Services:

As we discussed in subsection 1.2.2, the price determination models followed in the current service selection approaches can cause serious limitations for web service providers and requesters.

The dominant pricing model is called the fixed pricing model which leaves the complexity of price determination of web services entirely to service providers. Moreover, this model forces the service requesters to have an approach of take-it or leave-it toward a service offer. The second pricing model which is based on automated negotiation between providers and requester is often too complex to be practical. Furthermore, the complexity involved in this model makes it very hard for the service selection approaches to find globally optimum solution.

In our study, the pricing model, or more generally, the model that a service selection approach follows to determine the QoS profile of web services forms the basis for categorization of the current composite service selection approaches. This basis and the limitation of the fixed pricing and negotiation-based pricing models are discussed in more details in section 3.4.

Web Services' Market: Presence of Multiple Requests for Composite Services

In the extant literature on the web services' marketplaces, composite services are considered an essential part of these markets traded along the single web services (Papazoglou 2003; Yarom et al. 2004; Weinhardt et al. 2011b). In these markets, aggregation of web services supply is recognized as one of the potential value-added services to be offered to the market participants (Papazoglou 2003). As an essential part of the aggregation, composite service selection can enhance the exchange of composite services by: (1) matching composite service requests and web service offers based on the preferences and constraints of the service requesters and providers, and (2) facilitating the price determination of composite services.

However, very limited research has been performed to study how these markets affect composite service selection approaches. Most of the current composite service selection approaches have considered the setting with a single request for a composite service. One may reason that these approaches can also be applied to multiple requests by solving the service selection problem for the set of requests, one by one. We argue that the presence of multiple requests needs special consideration that simply solving the problem for the requests one at a time will not address it.

The consideration is in regard to the service providers' resource limitations. If providers' resources were unlimited, there would be no competition among service requesters to

procure web services. Therefore, a solution that optimizes the composite service selection for a single request could be extended to many requests without losing its effectiveness. In reality, web service providers are bounded by their resource limitations, such as limited computational power of their servers, and the requesters compete for these resources. Thus, new approaches are required to solve the composite service selection problem for the collective set of the requests, rather than solving the problem for each request separately.

3.4 Basis of the Categorization

To present the extant literature on composite web service selection, we have classified the existing approaches into three main categories: *optimization-based approaches*, *negotiation-based approaches* and *auction-based approaches*. The composite service selection approaches are categorized based on two aspects of web services' quality of service (QoS) profile, Fig 3.1:

- 1. *Dynamicity*: the underlying assumption of these approaches regarding how fixed or flexible is the offered QoS profile of a web service,
- 2. *Determination complexity*: how complex is the process to determine values for the quality profile attributes.

The QoS profile of a web service consists of a set of values offered for the different quality attributes of the web service by the service provider. QoS attributes or non-functional properties of a web service are the constraints defined over its functionality (O'Sullivan et al. 2002). They can be categorized as:

- Technical domain-independent attributes, such as: response time, availability, reliability, robustness (the ability of the service to continue its work in the presence of invalid, incomplete or conflicting inputs), capacity (the limit of concurrent requests a service can support for the guaranteed performance),
- Non-technical domain-independent attributes, such as: execution price¹¹, penalties, discounts, reputation,

¹¹ Based on this categorization, the price of a web service is considered as an element in the QoS profile of the service. However, throughout this research we may refer to this set as "the quality of service and price", with price as a separate element.

• Domain-dependent attributes which are only meaningful in a specific application domain, such as: refresh time for a traffic monitoring service (Comuzzi and Pernici 2009).

The web service QoS profile plays a central role in web service selection research. Different providers may offer the same service at different levels of quality to maintain their competitive advantage over each other (Medjahed and Atif 2007). As well, a single provider might offer the same functionality with ranging quality levels to cover a wider range of customers, i.e. service requesters with different preferences and constraints. Moreover, at the composite service level, the QoS of the final composite service is the key factor to ensure service requester's satisfaction (Zeng et al. 2004).



Fig 3.1. Classification of composite service selection approaches based on QoS Profile

In the current literature, there are two extremes regarding the assumption about the dynamicity of the QoS profile: (1) being pre-determined and not-customizable, or (2) being flexible and negotiable. Corresponding to the two extremes are the two important trends in the service selection literature: *Optimization-based approaches* which typically assume a predetermined QoS profiles and, *Negotiation-based approaches* which permit QoS profiles to be flexible and negotiable. However, regardless of the offered QoS being pre-determined or negotiable, the process to decide the best set of values for a service QoS profile is far from trivial for both optimization-based and negotiation-based approaches.

In general, a provider needs to consider different factors when choosing the appropriate values for different set of quality attributes. For example, the technical domainindependent attributes are likely to be bounded by the provider's limitations over their provisioning capacity; for example the specification of the servers executing web services. However, they still need to choose the set of values that are likely to generate the highest profit. As we already discussed it in subsections 1.2.2 (Price Determination for Web Services) and 2.3.1 (Dynamic Pricing), the decision regarding price determination of web services is considered even more complex, being a function of a complex set of variables such as the offered technical quality for the service, the production cost, existing supply and demand for that service and the competitor pricing.

In both optimization-based and negotiation-based approaches, the complexity of QoS profile determination is completely left to the providers. In optimization-based approaches, they have to constantly collect information from other providers and requesters to set the price at the most profitable level. We have already discussed the problems related to such an approach in subsections 1.2.2, 2.3.1 and 3.3, which in summary, make the assumed price determination model very unlikely to be profitable due to the specific characteristics of web services and their execution environment, Internet. Such an issue can make serious business implications if these approaches are to be applied in real applications.

In negotiation-based approaches, the price and quality are set through automated negotiation: providers and requester exchange their offers and counter-offers to reach to a mutual agreement over the values of quality attributes. In this approach, the relaxation of the assumption regarding static QoS profiles is an improvement over the less flexible optimization-based approaches. However, negotiation is known to be a very complex process, incorporating many different theories such as artificial intelligence, social psychology and game theory (Jennings et al. 2001). Negotiators need to decide on the best strategy for preparing their offers, taking into accounts many factors such as the opponent's strategy space, time and available resources.

Automated negotiation approaches for service selection have drawn extensively on the more general agent-based negotiation literature for concepts, functions, and frameworks. In order to achieve an automated negotiation process, they have incorporated many simplifying assumptions regarding the strategy space of both requester and providers.

This has led to a formulation of composite service selection which is not very realistic. Moreover, to reduce the negotiation complexity, the negotiation is performed for each service in composition, separately and independently. This, at best, achieves a locally optimum solution.

There is a third category of approaches emerging to solve composite service selection problem, based on economic theories. We have called this category "the auction-based approaches" as the composite service selection problem is mostly formulated based on auction models. Generally in this model, the service requester is the auctioneer and the providers are bidders to sell their web services.

With regard to the dynamicity of the QoS profile, in auction-based approaches, the price is dynamically determined by the market. Dynamic pricing in markets was discussed in subsection 2.3.1. This is an improvement over the completely pre-determined QoS profiles in optimization-based approaches. If required, it is also possible to achieve the customizable QoS profile, similar to negotiation-based approaches, through designing the appropriate auction model.

Price determination in particular and QoS determination in general are less complex in auction-based approaches, compared to the other two approaches. Instead of fully leaving the complexity of this process to the service providers, the market facilitates price determination and QoS determination by constantly sending the appropriate signals about the status of supply and demand to the service providers.

A service provider attending enough number of auctions receives feedback from the market: losing in too many auctions indicates that the service provider needs to reduce the price or improve the quality to be more competitive in the market, while winning in many auctions shows that the provider can demand a higher price for their services. Beside the providers, the market signals the service requester about their budget or quality constraints too: if the requester cannot procure the required composite service by attending in sufficient number of auctions, there is a need to increase the budget or ask for less severe quality for the composite service.

3.5 Optimization-based Approaches

The most natural approach to solve the composite service selection problem is to map it to an optimization problem. Optimization can be performed at two levels: *local optimization* for an individual task, approaches such as (Agarwal and Jalote 2010; Mukhija et al. 2007; El Haddad et al. 2010) and *global optimization* for the whole composite service, followed by for example (Zeng et al. 2004; Yu et al. 2007; Ardagna and Pernici 2007).

3.5.1 Local Optimization

Local optimization approaches choose the best service for the tasks in the BP, one at a time. For each task, services are ranked based on some criteria, such as the QoS attributes. The dominant technique to rank services is to assign a score to each web service, using utility theory.

In utility theory (from microeconomics), the service requester or provider preferences can be mapped to values of utility, where higher utility means greater preferences (Wilkes 2009). To avoid the complexities of multi-dimensional utility function elicitation, each QoS attribute and the price have an independent utility function, based on assuming the independence of the outcomes of utility functions originating from *Multi-attribute Utility Theory (MAUT)* (Keeney and Raïffa 1993).

Using a single attribute linear utility function, denoted as U_j in equation (3-1), the offered value for the *j*-th QoS attribute, q_j ($j \in J$: set of QoS attributes), by web service *s*, is mapped to a value between 0 and 1. In this equation, q_j^{max} and q_j^{min} are the maximum and minimum values offered for q_j by all the candidate web services of the same functionality class.

$$U_{j}(q_{j}) = \begin{cases} \frac{q_{j} - q_{j}^{min}}{q_{j}^{max} - q_{j}^{min}} & \text{if larger } q_{j} \text{ more desirable} \\ \frac{q_{j}^{max} - q_{j}}{q_{j}^{max} - q_{j}^{min}} & \text{if smaller } q_{j} \text{ more desirable} \end{cases}$$
(3-1)

The aggregated utility of all the QoS attributes offered by service *s*, denoted as U(s) in equation (3-2) below, is calculated as the weighted sum of the individual utility functions. Service requester assigns a normalized weight (w_j) to each QoS attribute to specify its importance. The sum of the normalized weights of all attributes involved should add up to 1, equation (3-3).

$$U(s) = \sum_{j \in J} w_j \cdot U_j (q_j)$$
 (3-2) $\sum_{j \in J} w_j = 1$ (3-3)

The limitation of local optimization approaches is that the service for each task is selected regardless of the dependencies that exist between the services participating in the composition. Therefore, it may not lead to a global optimality for the end-to-end QoS of the composite service. Moreover, it is not possible to consider the end-to-end quality requirements of the composite service while selecting services locally.

3.5.2 Global Optimization

To overcome the limitations of the local optimization service selection, global optimization approaches have been proposed. In one such approach, optimization is carried out for the overall BP, and the end-to-end requirements and constraints can be defined for the overall BP too. Nevertheless, service requester can still set local QoS selection criteria for each task. This can be achieved by applying the local QoS constraints as filters to the list of the candidate services returned by service registry.

Generally speaking, any optimization problem has three key elements: the objective function, the set of decision variables and the set of constraints. The solution of the optimization problem is the set of values for decision variables which maximizes (or minimizes) the objective function, while no constraint is violated. Existing global optimization approaches have addressed each of these elements as follows:

Objective Function:

In the current literature, the objective function is generally defined as: maximizing service requester's *satisfaction* from the execution of the composite service. To measure such satisfaction, researchers have referred to the utility theory: the objective function is formulated as the weighted sum of the end-to-end QoS attributes' utility functions. As usually there are multiple, probably conflicting, objectives to be optimized

simultaneously, this formulation of the service selection is a multi-objective (also known as multi-attribute or multi-criteria) optimization problem.

An example objective function is presented in equation (3-4) below, taken from (Zeng et al. 2004). In this formulation, service requester wants to minimize the cost (j_1 QoS attributes) and execution time (j_2) of the composite service, at the same time, maximizing the availability, success rate and reputation (j_3 to j_5).

$$Max\left(\sum_{j=1}^{2} \left(\frac{Q_{j}^{max} - Q_{i,j}}{Q_{j}^{max} - Q_{j}^{min}} * w_{j}\right) + \sum_{j=3}^{5} \left(\frac{Q_{i,j} - Q_{j}^{min}}{Q_{j}^{max} - Q_{j}^{min}} * w_{j}\right)\right)$$
(3-4)

 $Q_{i,j}$ is the aggregated value for *j*-th QoS attribute, offered by *i*-th execution plan and w_j is the normalized weight for *j*-th QoS attribute. Q_j^{max} and Q_j^{min} are the maximum and minimum aggregated values, offered for *j*-th QoS attribute by all the possible execution plans of the BP.

Constraints:

While optimizing the objective function, it is possible to specify constraints on different aspects of the selected solution. In composite service selection, one important set of such constraints have been defined over the end-to-end quality of the composite service. Ideally, service requester should be able to specify an upper bound or a lower bound for any quality attribute. For example, it should be possible to define a maximum budget to procure the composite service, or the minimum acceptable availability of the composite service.

Decision Variables:

The choice of what will represent the decision variables determines the type of optimization problem. The dominant approaches are modelling the problem as *Integer Linear Programming (ILP), Genetic Algorithm (GA), Constraint Satisfaction* and *Stochastic Programming*.

3.5.2.1 Integer Linear Programming (ILP)

Modelling the composite service selection as an integer linear programming problem (ILP) is one of the dominant approaches among the optimization-based researchers. Examples include (Zeng et al. 2004; Ardagna and Pernici 2007). In this approach, an

integer binary decision variable is assigned to each service which value specifies if that service is selected to be part of the composition or not.

More formally, given a BP with *J* number of tasks, there will be *J* classes of candidate services where all the m_j candidate services in the *j*-th class, can execute *j*-th task $(j \in J)$. Then, the decision variable x_{ij} is defined to indicate whether the candidate web service s_i , in service class *j*, is assigned to execute task *j* (equal to 1) or not (equal to 0). Generally, an extra constraint is defined to make sure that only one service is selected from each class; equation (3-5) below:

$$\sum_{i=1}^{m_j} x_{ij} = 1, \quad x_{ij} \in \{0,1\}$$
(3-5)

The advantage of this approach is the availability of many open source and commercial ILP solvers. Many of these solvers are very accurate and considerably fast in finding the optimal solution for not very large problem instances. However, an increase in the number of candidate web services leads to the increase of the number of decision variables, which in turn results in the explosion of the search space, and the number of the conditions to be checked. Therefore, the ILP approach is limited by how large the BP (number of the tasks) and the candidate services' space are.

Another limitation of this approach is that the objective function and the set of constraints should be *linear*. As the aggregation functions for quality attributes depend on the structure of the business process, it may not be always possible to define linear aggregation functions for all structures and all quality attributes.

3.5.2.2 Genetic Algorithm (GA)

Another optimization approach to solve service selection has been the application of genetic algorithm, followed by researchers such as (Canfora et al. 2005; Jaeger and Muehl 2007; Ma and Zhang 2008). The GA objective is to evolve a population of candidate solutions of an objective function toward better solutions.

Modelling a problem based on GA generally starts with encoding the candidate solution of an optimization problem in a computer processable manner, called *genome*. Then, the first *generation* is created, including a *population* of usually random solutions (or *individuals*). Each individual of this generation needs to be evaluated according to a

fitness function which shows how "good" it is. After the evaluation, the best individuals are selected for reproduction. The new generation is created using the best individuals of the previous generation by applying *genetic operators* such as *mutation* and *crossover*.

These steps are iterated till some conditions are met; such as a specific number of generations, or a time deadline is reached. The motive to the reproduction is that the new generation will contain better individuals than the old one and the average fitness function for the newly produced population will be higher. Thus, near optimal solutions can be found by repeating these steps.

In the service selection domain, Canfora et al. (2005) and Jaeger and Muehl (2007) applied a simple one-dimensional coding schema for the problem representation, while others, including Ma and Zhang (2008), have used more complex representations such as a relation matrix coding schema. In the former case, each *individual* represents the assignment of the candidates to the tasks. In the latter one, the matrix can represent all the execution paths of the BP at the same time.

Similar to integer linear programming approaches, GA approaches define the objective function as to maximize some QoS attributes and minimize some others. The objective function is defined through the GA's fitness function. However unlike ILP approaches, GA does not need to define a linear objective function. Being a heuristic search algorithm to find near-optimal solutions, the GA based service selection approaches can more efficiently handle larger problem instances compared to ILP approaches.

Being an unconstrained search technique (Fonseca and Fleming 1998), one limitation of GA is that it is not possible to define additional constraints while searching for the best solution. Therefore, service requester cannot directly specify quality or budget constraints for the composite service. Some researchers have tried to integrate the constraints indirectly into the search process. One such technique is called *the additive penalty method* where a penalty cost that is proportional to the total violation of each of the constraints is added to the fitness function (Hilton and Culver 2000). There are other techniques to incorporate constraints into the GA search, such as the ones proposed by Carlson (1995) and Michalewicz (1995). However, these techniques have not found any application in the service selection literature.
3.5.2.3 Constraint Satisfaction

A group of researchers argue that the objective of the composite service selection approach needs to be defined based on the time required to create a composite service, rather than achieving an optimal composition. The argument is based on a vision of the future semantic web being populated with millions of services which will be available globally. Therefore, the composite service selection approach needs to be scalable in terms of efficiently searching in the pool of available services, with respect to time.

Following such an approach, Lecue and Mehandjiev (2009) have proposed a fast selection approach which might not lead to an optimal composition. In their work, the service selection is modelled as a constraint satisfaction problem, where a stochastic search method (more precisely, a hill-climbing algorithm) finds the first set of services that satisfy the set of defined constrains (both in terms of functional and non-functional requirements).

A generalization of constraint satisfaction approach is presented by Rosenberg et al. (2009), modelling the service selection problem as a *Constraint Optimization Problem* (*COP*). In this approach, constraints are weighted and the goal is to find a solution maximizing a function of weighted constraints. They argue that, in contrast to the assumption of the constraint satisfaction approaches, not all constraints are *hard*; meaning that the solution strictly needs to satisfy them. Rather, it is possible to categorize some of the constraints as *soft* which are optional and it would be "nice" to have them.

The proposed CO algorithm does not find the best solution that exists in the search space (in terms of the utility gained by the service requester). Rather it searches for the best solution within the boundaries of the service requester's constraints. They add up all soft constraints to form an objective function, trying to maximize it. However, as they have mentioned, this approach has problems regarding scalability which makes it not applicable for large problem instances.

3.5.2.4 Stochastic Programming

Focusing on the uncertainty that exist about the values offered for QoS attributes, some researchers have applied Stochastic Programming to solve the composite service selection problem. For example, Wiesemann et al. (2008) argues that the nature of some QoS

attributes such as response time and price is *non-deterministic*, and hence the WSC should be treated as a decision problem under uncertainty.

In their approach, the service requester needs to quantify the risks associated with time and cost uncertainties through a particular quantile-based risk measure called the *average value-at-risk (AVaR)*. The associated AVaR measures for execution time and cost are then used to build *the worst-case risk functions* for the two attributes. These risk functions form the objective function that aims to minimize the AVaR of the random variables defined for the service response time and invocation cost.

In their experiment, they have compared their risk-aware formulation of WSC in terms of the execution time and the cost of the resulted composite services with those of the deterministic formulation of the problem. According to their findings, for every deterministic composite service, there exists a risk-aware composition with smaller cost and execution time.

3.6 Negotiation-based Approaches

Negotiation is a process to reach an agreement that is beneficial to the involved parties through information exchange and compromises (Kim et al. 2003). Generally having different preferences over the negotiation issues, negotiating parties seek to reconcile these differences through negotiation.

In computer science terminology, negotiation is defined as a distributed search through the space of potential agreements (Jennings et al. 2001). It has been used for many years to solve a variety of problems such as resource allocation in grid computing and getting agents to cooperate or compete over a common goal in multi-agent systems. In the context of computer science research, we should make it clear that what we mean by negotiation is *an automated process* where negotiation is performed automatically by a piece of software such as an agent, a web service, or a third-party broker system. The automated negotiator replaces the human negotiator and performs negotiation on the negotiator's behalf.

In the web service domain, negotiation is mainly employed for (semi-)automatic creation of *Service Level Agreement (SLA)*, also known as contract, policy or license. In general terms, SLA is the agreement between service consumer and provider. In service oriented infrastructure, SLA is an automatically processable contract between a service and its client, where the client can be an organization, a person, or another service (Ul Haq et al. 2010). During *SLA negotiation*, service providers and requesters negotiate over SLA terms such as QoS attributes, rewards, penalties and deliverables to create a formal SLA at the end of the process (Zulkernine et al. 2009). The formal SLA is meant to satisfy both sides' requirements.

SLA negotiation solutions are divided by the assumption that the service provider is predetermined before the negotiation or not. The two corresponding approaches are called *pre-contractual SLA negotiation* and *dynamic provider selection*. Although the former type of negotiation does not aim at service selection, we briefly explain it here to make the distinction between the two cases of negotiation:

• Pre-contractual SLA Negotiation (Grimm 2007)

In this approach, the negotiation is performed after service selection, with a predetermined provider. The objective of this negotiation is to set the values for the service parameters in order to define the concrete service which will be carried out. This is a "one-to-one" negotiation process between service requester and the selected service provider. Proposals in this area include, but not limited to (Zulkernine and Martin 2011; Comuzzi and Pernici 2005; Gimpel et al. 2003).

• Dynamic Provider Selection

Here, negotiation is performed after service discovery and as a service selection mechanism, aiming at dynamically selecting the service provider that best matches the service requester's non-functional requirements. This is a "one-to-many" negotiation process between the service requester and the candidate service providers. A successful negotiation output can be used for contract specification.

Basically in the dynamic provider selection approach, a high-level negotiation process is conceptualized that negotiates for the business process. This high level process consists of multiple negotiation sub-processes, each associated with one task in the BP. Each negotiation sub-process in turn, may include multiple negotiation threads, one thread for each candidate provider, to choose the best service for the specific task. In general, when building an automated negotiation solution, several key components comprising *the general negotiation framework* (Mueller 1996; Faratin et al. 1998) should be addressed. These critical components are:

- 1. Negotiation Object: the set of issues that the parties negotiate to reach an agreement over their values,
- 2. Negotiation Protocol: the communication and message exchange rules among negotiation parties,
- 3. Decision-making Model: the rules that the interacting parties follow to decide when to start negotiation, how to prepare an offer, acceptable agreement range, and the time to abandon negotiation.

However, the general negotiation framework does not provide all the required elements to design a negotiation solution for dynamic selection of service providers. Particularly, a further management layer is required to deal with the end-to-end QoS requirements and ensure an overall successful negotiation process. This management layer is referred to in the literature as coordination (Chhetri et al. 2006).

In Fig 3.2, we present the WSC negotiation framework. This extends the general negotiation framework with an additional component, the coordination model, which includes the required aspects of coordination strategy and architecture as explained below:

- **Coordination Strategy** involves decisions on: (1) Time to initiate negotiation processes for each task: All parallel? Sequential? With what priority? (2) The type of information to collect from ongoing negotiation processes and/or finished ones to improve the negotiation result, and (3) Actions to take for improving the negotiation result or prevent its failure, based on the collected information.
- **Coordination Architecture** involves how many and what type of negotiators are involved in negotiation (agents, web services, broker systems), and the required number of coordination layers and their configuration.



Fig 3.2. WSC negotiation framework adapted from the general negotiation framework (Mueller 1996; Faratin et al. 1998)

We discuss below the realizations of the key elements of the framework in the current literature. A summary of this discussion is included in Fig 3.3.

3.6.1 Negotiation Object

In service selection, the negotiation object is the set of QoS attributes that service requester and providers choose to negotiate over their values. In general, QoS attributes can be negotiable or non-negotiable. Negotiable attributes are those whose values can be determined at run-time, during service invocation (Comuzzi and Pernici 2009). The non-negotiable attributes have pre-determined values that cannot be changed. For example, a domain-dependent attribute specifying the service encryption method may be considered non-negotiable by service requester and/or provider.

For each negotiable attribute, service requester and provider each has a minimum and a maximum admissible value. Negotiation is performed over the range of admissible values for each attribute. Price, availability, and response time are the more commonly included terms in recent service selection experimental investigations (Zulkernine and Martin 2011; Richter et al. 2012; Yan et al. 2006).

When the negotiation object includes more than one issue (or attribute), the negotiator needs to know the relative importance of each issue. This is usually realized through a normalized weight for each negotiation issue.

3.6.2 Negotiation Protocol

To discuss the negotiation protocol for composite service selection, we need to notice that the high level negotiation process for the composite service is only a conceptual illustration of the negotiation process. The actual negotiation that ultimately occurs is a bilateral negotiation between service requester's representative for each task and the representative of the candidate provider.

Some researchers have used a general bilateral protocol (Richter, Chhetri, et al. 2010; Ardagna and Pernici 2007; Comuzzi and Pernici 2005; Gimpel et al. 2003), also called the *bilateral message exchange* or *bargaining*. This general protocol consists of a series of message exchanges between the two parties in terms of *offers* and *counter-offers*, until one of them accepts an offer or withdraws from the negotiation due to reaching to a stop criterion such as the maximum negotiation time.

Some researchers, including Zulkernine and Martin (2011); Yan et al. (2007); Chhetri et al. (2006), have followed a standard protocol such as *FIPA Iterated Contract Net Interaction Protocol(ICN IP)* (Foundation for Intelligent Physical Agents 2000). This protocol allows multi-round bidding, supporting one-to-many negotiation. Under this protocol, the negotiation initiator issues the initial *call for proposals (CFP)* (Foundation for Intelligent Physical Agents 2000). The other side of the negotiation (contractors) answers by sending an offer or by refusing to participate in the negotiation. The initiator may accept or reject an offer or reply with a revised CFP. The negotiation terminates when the initiator accepts one or more offers, or refuses all the bids without issuing a new bid, or if all the contractors refuse to bid.

Some researchers have proposed *generic* negotiation protocols. The idea is not to bind the negotiation solution to a particular protocol at design time. Rather, delaying the determination of the suitable negotiation protocol until the actual execution of the negotiation process to make a flexible solution. For example, Hudert et al. (2009) have extended the WS-Agreement specification, originally being developed by Andrieux et al. (2007), to define a separate stage for protocol determination. During this stage, the negotiation parties agree on a common negotiation protocol before the actual negotiation process starts.



Fig 3.3. The realization of the WSC negotiation framework based on the current literature

3.6.3 Decision Making Model

The decision making model in a negotiation-based solution has to make decisions regarding two important aspects of negotiation:

- How to evaluate a received proposal as to whether accept it or not (*utility function*),
- How to prepare a counter-proposal (*tactic/strategy*).

The first aspect is studied through the concept and theories related to the utility function of the negotiator, and the second aspect is covered by discussions related to a negotiator's high level strategy and more specific, low level tactics.

3.6.3.1 Utility Function

To evaluate the received proposals, negotiators need to have clear understanding of their preferences about the negotiation object. These preferences guide their decisions during negotiation. In service selection, the dominant approach to express the preferences is

based on utility theory; the same approach that we already explained for optimizationbased service selection in section 3.5.

Several researchers addressing SLA negotiation have used *single attribute linear* utility function to evaluate the value of an individual issue (Zulkernine and Martin 2011; Yan et al. 2007; Ardagna and Pernici 2007). This utility function is similar to the equation (3-1) mentioned before (subsection 3.5.1). In the negotiation context, q_j^{max} and q_j^{min} are defined as the maximum and minimum admissible values for *j*-th QoS attribute according to the negotiator's preferences and constraints. The *parametric* single attribute utility function (Comuzzi and Pernici 2005), and *multi-attribute* utility function representing the relative preference with respect to each pair of attributes (Gimpel et al. 2003) have also been discussed in the literature.

As a QoS profile typically involves more than one attribute, the negotiator needs to aggregate the preferences over all the attributes involved to make the decision regarding the acceptance or rejection of a received offer. The more commonly used technique to measure the utility of a profile with multiple attributes is to assign a normalized weight to the utility of each attribute and then calculate the overall utility using a weighted linear additive function; similar to the aforementioned equations (3-2) and (3-3).

3.6.3.2 Negotiation Tactics

To generate an offer during the negotiation process, two main approaches are discussed in the literature: *concession* and *trade-off*. The main difference between the two approaches is in the utility of the offer for the negotiator.

A negotiator with the concession approach concedes to the other side of negotiation (opponent) while preparing every new offer. The concession is made by preparing an offer that has a lower utility value for the negotiator itself, and apparently, a higher utility value for the opponent. In order to make the concession, the negotiator needs to make decisions regarding: firstly, the pace of offering concessions throughout the negotiation process, and secondly, the amount of concession in each offer. The offer is then prepared with respect to the decided concession.

Faratin et al. (1998; 2002) and Comuzzi et al. (Comuzzi et al. 2005; Comuzzi and Pernici 2005) have proposed heuristic approaches to define the offer in bilateral negotiations. In

Faratin's proposal, three families of tactics are presented to prepare a concessionary offer: time-dependent, resource-dependent, and behavior-dependent (or imitative). The concession made in each tactic depends on the tactic's influential factor which is: the negotiation time, negotiator's available recourses and the opponent's behavior. Faratin's heuristic functions (1998) are widely adopted by researchers, such as Zulkernine and Martin (2011); Richter et al. (2010); Ardagna and Pernici (2007), due to the clear distinction of tactic families (based on time, resource, and opponent behavior), the clear mathematical representation and the analysis of negotiation convergence for different parameters of the model.

The tactics in Faratin's proposal are based on the generalized influential factors for any bilateral negotiation context. However, negotiation for composite service selection has specific characteristics that can be used to introduce additional influential factors, and consequently, new tactics. For example, negotiating for a task in the composite service consists of multiple negotiation threads with different service providers for that task. Hence, the requester's negotiator receives multiple offers at the same time. Considering this characteristic, (Jiuxin et al. 2010) has proposed a new influential factor called the Global Negotiation States factor. This factor can reduce the need for unnecessary negotiations in one-to-many negotiations. In this proposal, the received offers are compared to each other, and then, if all the offers are far from the negotiator's offer, the negotiator should be ready to make some (big) concessions. Otherwise, if any offer is more desirable than the negotiator's own offer, negotiator will raise its expectations and prepares the next offer based on the value of the best received offer.

In contrast to the concession approach, a negotiator with the trade-off approach tries to keep its utility value stable at a desirable level (the *aspiration* level) throughout the negotiation, while generating an offer that has more utility value for the opponent. This can be achieved by trading-off between the values of different issues (Faratin et al. 2002), that is lowering the values of some QoS attributes while demanding more on some others. Such a strategy maximizes the chance of the offer being accepted.

In the trade-off approach, as the negotiator usually has no information about the opponent preferences and utility function, the main challenge is how to determine which offer increases the opponent's utility value. The trade-off strategy proposed by Faratin et al. (2002) uses the concept of "fuzzy similarity" (Zadeh 1971) to approximate the

preferences of the opponent. Assuming that the opponent's last offer reflects its preferences, the negotiator uses it as a reference point and prepares a counter-offer that is most similar to it. In the Yan et al. (2007) proposal, the authors take advantage of the one-to-many negotiations occurring for a composite service. The utility value of all the received offers is calculated, and the one with best utility is used as a reference point for preparing the counter-offer.

3.6.3.3 Negotiation Strategy

Negotiation strategy is another part of the decision model. Conceptualized at a higher level of abstraction than the negotiation tactics, it aims to maximize the utility function of the negotiator for a contract (Faratin et al. 1998), by determining when to use which tactic to prepare the offer, or what combination of tactics to use. More precisely, strategy can be thought of as the pattern of change in the weight of different tactics over time (Zulkernine et al. 2009). Taking it one step further, Di Nitto et al. (2007) states that strategy is not just about how to weight different tactics over time, but it can also address the following factors:

- Changing the importance of negotiation issues over time, such as preferring availability over the response time if the latter cannot be improved so far,
- Changing the severity of the constraint, such as relaxing some constraints on the values of some negotiation issues to reflect more concession when the negotiation time is about to expire.

Deciding on the best strategy for a negotiator involves the challenges addressed mostly in game-theory, microeconomics, and multi-agent systems and has not been the focus of composite service selection community.

3.6.4 Coordination Model

To avoid the complexity of dependent negotiation processes, researchers including Richter et al. (2011); Jiuxin et al. (2010); Yan et al. (2007), have assumed that the multiple negotiations are independent and concurrent. For the same reason (avoiding complexity), no information is collected during an ongoing negotiation process to improve the result of another negotiation process.

As proposed by Yan et al. (2007), the coordinator takes part only at the end of the process to either confirm or reject the negotiation result. Extending (Yan et al. 2007), Richter et al. (2012; 2010) attempted to make the coordinator more actively involved in negotiation. Thus, the coordinator does not wait for all the negotiation processes to finish. Rather, when a negotiation process finishes successfully, the surplus of the negotiation issue is calculated. Surplus is the difference between the actual agreed value and the least desired value (that is maximum payable price from the service requester point of view) of the negotiation attribute. Subsequently, it is distributed over failed or unfinished negotiation processes of those tasks which are dependent to the task producing the surplus. The dependency is determined based on the QoS attribute under negotiation, and the task's position in the process, and is maintained in a tree-format. However, redistributing surplus may prevent the failure of the negotiation process when service requester has severe QoS requirements. It is not helpful in situations where negotiation fails due to the limited negotiation time of either side of negotiation.

3.7 Hybrid Approaches

There are service selection approaches which are not based on pure optimization or negotiation. In this section, we summarize two of the more important contributions.

3.7.1 Optimization + Configuration Approach

One attempt to proceed from a totally predetermined QoS profile to a more flexible one is the work by Comuzzi and Pernici (2009). In their approach, rather than providing a single value for each QoS attribute, the service provider publishes the set of values that they can offer for each QoS attribute. For example, a provider offering a Traffic Monitoring Service can publish the offered quality for the two QoS attributes of $Q_{refresh time}$ and $Q_{execution time}$ this way:

$$Q_{refresh time} = \{2h, 1.5h, 1h, 0.5h\}, Q_{execution time} = \{[2s, 3s), [1s, 2s)\}$$

This means that $Q_{refresh time}$ (the time interval between the updates of the traffic information) can be offered with any of the intervals of 0.5, 1, 1.5 or 2 hours, and the service execution time, $Q_{execution time}$, can be preserved within any of the two specified intervals.

Additionally, instead of assuming a single value for the service price, they have proposed a pricing model. In this model, the provider publishes a set of pricing functions for the service's QoS attributes. Each attribute's pricing function determines how much it will cost for the service requester to select a specific level of quality. The web service total price is calculated as the sum of its constituting pricing functions.

The proposed service selection technique is based on local optimization, minimizing the price for requester. For each task, the web service with the lowest price for the *minimum quality profile* is selected. Minimum quality profile consists of the lowest level of quality for each QoS attribute which still satisfies service requester's quality demand. When service selection is completed, a subsequent *agreement configuration* step is performed. During this step, the difference between the price of the low quality profile of the selected service and the service requester's budget is used to improve upon the offered service quality for requester.

In this research, the assumption about the QoS profile of a service is different from the optimization-based approaches in that they do not assume predetermined QoS profile for neither the service offer nor the service request. Instead, the service provider is able to publish the quality profile in the form of different quality levels that they support. Besides, a higher level of flexibility is supported for the service price offering with the proposed pricing model. Thus, service requester can receive a personally-configured service, based on their preferences and constraints.

However, the flexibility of the QoS profile in negotiation-based approach does not exist here, as no negotiation (exchanging of offers and counter-offers) is actually taking place between the two sides. Rather, a configuration process tailors the service quality based on the requester's preferences and budget.

3.7.2 Optimization + Negotiation Approach

The research by Ardagna and Pernici (2007) is another attempt to relax the assumption about a fully pre-determined QoS profile to a more flexible one, by combining optimization with negotiation. In their proposal, service selection starts as a mixed integer linear programming (MILP) optimization problem. If the optimization process fails to find a feasible solution, due to sever QoS constraints for example, a negotiation process will initiate. At the beginning of the negotiation process, the execution plan that satisfies the maximum number of constraints is identified. Then, negotiation starts with any service provider that contributes to violating the global constraints of this execution plan. After the negotiation is completed, the providers who have agreed to improve their offered quality of service, in return for a higher price, will be added to the optimization space. In other words, negotiation is used to find new QoS attribute values for web service invocations by expanding the optimization solution domain. As the last step of service selection, optimization is repeated with the new solution pool.

As they have mentioned in their work, identifying the maximum number of constraints that can be satisfied is an NP-hard problem. Thus, they have assumed the global constraints are limited which allows to find the maximum number of violated constraints through an exhaustive search. Comparing their approach with pure negotiation-based approaches, coordination is not required here (Fig 3.4). In fact, provider selection is performed through optimization, and not negotiation. However, in contrast to optimization-based approaches, the providers have a chance to improve their offered quality if their existing offers do not satisfy the service requester's requirements.



Fig 3.4. Different perspectives on applying negotiation for service selection during WSC

3.8 Auction-based Approaches

More recently, a third approach based on dynamic market mechanisms such as auction models has emerged for the composite service selection problem. Unlike the previous two approaches, there is a limited research on auction-based approaches for composite service selection, especially in terms of concrete experiments for evaluating the proposals.

In this section, we first introduce the auction models applied in the current literature for composite service selection. Then, the research in this area is studied based on the design

elements of an auction: the bidding language, the allocation rule and the pricing scheme. A summary of the analysis is provided at the end of this section, in Table 3.1.

3.8.1 Auction Models

The proposed auction models can be studied based on the following attributes:

- 1. Being direct auction or a reverse auction,
- 2. Being combinatorial or non-combinatorial,
- 3. Being a single-shot or an iterative auction,
- 4. Auction for a single composite service request or multiple ones.

3.8.1.1 Direct / Reverse Auction Model

Many of the researchers have modelled service selection as a reverse auction (Esmaeilsabzali and Larson 2005; Mohabey et al. 2007a; Mohabey et al. 2007b; Prashanth and Narahari 2008; Blau et al. 2010; Watanabe et al. 2012; He et al. 2014). In this approach, service requester or an independent third-party take the role of the auctioneer and service providers bid to sell their services.

Contrary to this trend is the direct auction model proposed in Lamparter (2007). In this auction model, service providers offer their services in bundles. Service requesters can bid to buy these bundles. For each bundle, the requester with the highest bid wins the bundle.

The general problem with modelling composite service selection as a direct auction is that a requester who needs multiple services to create a composite service may have to attend multiple auctions to win all the required services. To have a fully operational composite service, the requester needs to win all the related bundles. With no guarantee for winning all the bundles, the requester might end up winning some services and losing some others. Such situation may incur undesirable cost for the requester.

However, the direct auction model proposed by Lamparter does not have this problem. As the auction is modelled as a combinatorial auction, a composite service requester can prepare a bid for a bundle of services to make sure that they will win the auction only if they get all the required services. As this study is the first to consider service selection for multiple composite service requests, we will discuss it in more details in subsection 3.8.1.4 (A Single Request / Multiple Requests).

3.8.1.2 Combinatorial / Non-combinatorial Auction Model

A number of researchers have chosen to model the composite service selection problem based on non-combinatorial auctions, such as (Esmaeilsabzali and Larson 2005; Blau et al. 2010; Watanabe et al. 2012). In this model, a provider can only bid to offer a single service. We have already discussed the implications of such an assumption in subsection 3.3 (Dependencies between Constituent Web Services of a Composition).

Combinatorial auction model has been a popular approach among many researchers from different disciplines. As already discussed in subsection 2.3.3, combinatorial auctions offer many advantages such as increased economic efficiency, increased revenue or in case of a reverse auction, cost savings (Cramton et al. 2006), time efficiency and impacting the market structure (Bichler et al. 2006). At the same time, the complementarity effects that exist between the tasks of a composite service and the dependencies between web services forming a composition have been a major motivation for researchers to consider combinatorial auction models for service selection. Examples include: (Mohabey et al. 2007a; Mohabey et al. 2007b; Lamparter 2007; Prashanth and Narahari 2008; He et al. 2014).

As mentioned in Blau et al. (2010), the main problem with combinatorial auction is the complexity of the winner determination problem. Combinatorial auctions are proved to be NP-complete (Sandholm 2002) and therefore a solution based on these auction models is not scalable to settings with large numbers of bidders and services involved. Therefore, one of the concerns in this area is to evaluate the proposed model in terms of scalability.

3.8.1.3 One-shot / Iterative Auction

Some researchers have modelled the auction for composite service selection as a one-shot auction: the bidders submit the offers and the auctioneer determines the winners based on submitted bids. However, some researchers such as Watanabe et al. (2012) and He et al. (2014) have chosen more complex auction models.

The proposed non-combinatorial auction model in Watanabe et al. (2012) has two steps. In the first step, for each task in the composite service, the first few providers (number is varied in the experiment) with the best quality maximizing offers are selected (quality includes the price). So far, this approach is similar to the local optimization discussed in subsection 3.5.1. However, in order to address the limitation of local optimization approaches in not being able to support the end-to-end quality constraints for composite services, a second step is proposed. In this step, the global quality constraints are investigated and if violated by the current offers, the providers will be asked to improve their quality while allowing them to have a trade-off tactic. As discussed in subsection 3.6.3.2, a trade-off tactic allows the participants to improve on some quality attributes while decreasing the desirability of some others. The proposed solution does not guarantee to find the optimal (quality maximizing) utility for the composite service. However, the end-to-end quality constraints are satisfied in the second step if negotiation with providers is successful.

An iterative combinatorial auction model is proposed in He et al. (2014) to solve the service selection problem. A number of stop criteria has been defined and if none of the criterion is met at the end of a round of bid submission, the auction proceeds to the next round. One such stop criteria is the quality requirements of the service requester being satisfied by the available service offers. If there is no set of bids to fulfil the quality requirements, the auctioneer sends an Ask-QoS to more competitive providers and asks them to improve their quality and price offers. Application of iterative auctions for composite web service selection implies that service providers are willing to spend enough time attending multiple rounds of auction for the same composite service. This approach may not lead to profitable trades for service providers, for all types of composite service requests, for example composite services which prices are not expected to be high. We will discuss the limitation of the application of iterative auctions for web service selection in more detail in subsection 4.3.1.2.

3.8.1.4 A Single Request / Multiple Requests

Most of the research in this area aims to solve the service selection for one composite service, including (Esmaeilsabzali and Larson 2005; Mohabey et al. 2007a; Mohabey et al. 2007b; Prashanth and Narahari 2008; Blau et al. 2010; Watanabe et al. 2012; He et al. 2014).

Papazoglou (2003) is one of the first to discuss the presence of multiple composite services in the web services marketplaces. He states that the purpose of these markets is to create the opportunity for service requesters and providers to meet and conduct business as well as fostering the possibility of offering value-added services such as aggregation of the web service supply/demand.

Other researchers have studied the web services markets from a variety of aspects. For example, if they need to be open (Papazoglou 2003) or established privately (Petrie and Bussler 2008); be centralized or decentralized (Yarom et al. 2004); their fundamental structure and players (Geng et al. 2003; Legner 2009; Weinhardt et al. 2011b) and the best strategies of the players (Tang 2004; Gunther et al. 2007); trust establishment (Brehm and Golinska 2009); and the semantic aspect of web services in such marketplaces (Lamparter 2007; Schulte 2010). In these literatures, all researchers agree on composite services being an essential offering in the web services' markets. However, composite services are considered as already-existing entities that can be traded in the market along single web services. Very limited research exists on how these markets can facilitate the different aspect of creating value-added composite services, including composite service selection or the price determination of a composite service.

Tang (2004) is one of the first to consider the impact of multiple requests for composite service selection. Taking the service provider's perspective, this study investigates the optimal strategies for offering web services, taking into account the integration cost. Tang has analyzed a setting where two service vendors sell two distinct but functionally complementary services to different groups of potential buyers. Based on the performed analysis, service providers benefit from offering composite services, either through forming strategic alliance with other providers or selling composite services in a marketplace. With the main focus being on providers' best strategies in offering functionally complementary services, there is no discussion of specific auction models for allocating services to requesters.

In a more recent study, Lamparter (2007) studied the use of semantic technologies to automate contracting in a market for web services. This study includes a conceptual market model to match web service offers to multiple composite service requests. In this direct auction, the service providers offer their services in bundles and service requesters

bid to buy these bundles. However, the proposed model has limitations in addressing the composite service selection problem.

Firstly, it is assumed that the composite service requester is limited to have only one winning bundle. This means that the service requester should only bid for the bundles that include all the services required to create the composite service. In other words, the bundle is not dividable. With such an assumption, the requests for complex composite services are not very likely to find service bundles which include all the required services.

The second problem is that even if this constraint, requester having one winning bid, is relaxed to allow requesters have multiple winning bids, the service requester faces a challenging problem in bid preparation: they need to decide how to divide the required composition with respect to existing offered bundles so that they can optimally achieve their quality and price requirements.

To reduce the bid preparation complexity for service requesters, it is possible to design a different auction model that instead of the requester checking how to prepare the bids, the auction-based model finds the optimal bundles based on the requester's requirements. However, extra constraints are required to be added to the matching algorithm to make sure that: (1) the combination of the bundles includes all the required services for the composition, and (2) the requester will not be assigned more services than they require.

Thirdly, the model aims at solving the composite service selection problem for all the requests simultaneously. This means even with one request being unsuccessful in finding all its constituent services, the whole auction will fail; that is, no other requests would be assigned any services. The success rate of such a model in allocating services to requests is likely to be low, especially if the requests are for complex composite services.

Finally, no evaluation has been done on the performance of the proposed model. Therefore, it is not possible to have estimates about the success rate of the auction-based model, the final cost of the composite services or the time required by the model to find service allocation for composite requests.

To the best of our knowledge, our study is the first to consider the impact of presence of multiple requests on the composite service selection approach, while addressing the limitations discussed before, and perform a thorough evaluation of the performance of the different possible service selection approaches in such a setting.

3.8.2 Bidding Language

The existing literature has mainly chosen a multi-dimensional bidding language where bidders can include other aspects of the item in addition to the price. Considering the fact that web services are specified not just by the performed functionality, but also by their non-functional or quality attributes, one important dimension to include is the set of values for the quality of the service attributes.

Lamparter (2007) has proposed a bidding language that includes the quality of service and price. However instead of including an explicit price, the bid includes a pricing function that maps the quality of service to a monetary value for that service. Each bid contains a configuration set and a pricing policy: $\{C, U(c_i)\}$ where each configuration, $c_i \in C$, includes a set of values for the quality attributes. The pricing policy maps the configuration to a real number which is the price to be paid (to be requested) for that configuration.¹² For bundled services, the price of the bundle is calculated as the sum of the prices of the bundled services. This is very similar to the hybrid approach proposed by (Comuzzi and Pernici 2009).

There are some studies which have focused on other aspects of web services rather than the quality of service. For example, Prashanth & Narahari (2008) have proposed a combinatorial auction model for the setting that requesters are interested in procuring multiple instances of the composition. The proposed bidding language is multi-unit where service providers submit bids in the form of a discount curve (Fig 3.5). The discount curve specifies the provider's volume discount based on the number of procured executions of the bundle. To achieve a polynomial time allocation rule, there are restrictions over the structure of the composite service (either sequential or tree-like structure) and the possible combinations for bundling services (sequential or a path in the tree structure).

¹² In their study, both web service offers and requests are called bids in general. However, the formulation of the service selection problem only takes into account the service requests. Therefore, the model is equivalent to a direct auction model where service providers sell their services and service requesters bid to buy the services.



Fig 3.5. An example bid in the form of volume discount curve

In this regard, another example is the work by Mohabey et al. (2007b). They have considered the interface of the web services (input and output) to be included in the bids in addition to the quality of service. They have added an additional allocation constraint to check the interface matching of sequential services.

As discussed in subsection 2.4.2.1, when specifying the bidding language, it should be also clear what combination of bids bidders can have, that is, the choice of OR, XOR or OR*. Most researchers did not include any specific discussion regarding this aspect. This implies that a bidder can have as many bids as they want, and have as many winning bids as possible.

This setting, at first, seems to be according to the OR language. However, most formulations of the problem have included a constraint to limit the number of winning services for each task in the business process to be exactly one. This constraint helps the bidders to have the expressivity of the XOR language by adding dummy items to the bids they want to XOR. This makes the bidding language according OR^{*}, which is as simple as the OR combination, yet more expressive.

An exception to the unlimited winning bids for providers is the bidding language applied in He et al. (2014). They have used XOR to combine the bids of the same provider to restrict them to have one winning bid only. As their proposed mechanism is based on iterative auctions, the choice of XOR for combining bids simplifies the information feedback problem for the auctioneer at the end of each auction round.

3.8.3 Allocation Rules

As discussed in subsection 2.4.2.2, the allocation rule has two main components: the objective function and the set of allocation constraints. In the current literature on auction-based approaches for composite service selection, there are two dominant formulations of the objective function:

1) Utility maximization:

This formulation includes all quality attributes and the price in the objective function. The aim is to maximize the utility (of the composite service execution) for the requester as in proposals by (Esmaeilsabzali and Larson 2005; Mohabey et al. 2007a; Watanabe et al. 2012) or maximize the utility across all participants including the requester and providers, as in the proposal by Blau et al. (2010).

2) Cost minimization (profit maximization):

This approach formulates the objective function only based on the price. Some proposals aim to minimize the cost for requester, such as works by Mohabey et al. (2007b), while some others maximize the profit for providers (more specifically, the willingness-to-pay of the service requesters) such as the proposal by Lamparter (2007).

The proposals following the first formulation mostly specify an objective function similar to the ILP-based global optimization approaches: maximizing the sum of the utilities of the end-to-end quality attributes (including the price) while taking into account the importance of each quality attribute for the requester through assigning a weight to the attributes, as discussed in subsection 3.5.2.

There are exceptions to this approach, such as the works by Watanabe et al. (2012) and Blau et al. (2010). Watanabe et al. (2012) has proposed a two-step approach. At the first step, a set of quality maximizing providers are selected for each task. As this selection is performed locally for each task, the second step aims to check and resolve the potential end-to-end quality constraint violations, through a round of negotiation with the set of selected service providers. Blau et al. (2010) has proposed an objective function that maximizes the utility across all participants, bidders and the bid-taker. The objective function is defined as:

$$argmax_{f\in F} (\alpha, \varphi(\mathcal{A}_f) - \mathcal{P}_f)$$

where $f \in F$ is the set of services to be selected for a composite service request; \mathcal{A}_f is the aggregated quality profile of the set of services f that includes all quality attributes except for price; φ is a function that maps quality profile of f to a score; α is the requester's maximum willingness to pay for the composite service; and \mathcal{P}_f is the total price asked for the set of services in f.

To be selected, the set f needs to maximize the difference between the price the requester is willing to pay for a composite service based on its quality profile and the price asked by providers to provision the composite service. The proposed objective function is different from other proposals in two aspects:

- The objective function considers both sides; maximizes the willingness to pay of the requester for providers and minimizes the procurement cost for the requester;
- The utility function is Simple Additive with respect to all quality attributes, except for price. This gives a score for the quality profile of the set of services which is multiplied by the requester's maximum willing to pay.

As discussed in subsection 3.3, the main challenge for the utility maximizing formulation is elicitation of the weights for different quality attributes from the service requester. To address this problem, some researchers have formulated the objective function only based on price.

The price-based formulation excludes the need to specify the weights for quality attributes, and at the same time, provides enough support for service requester to specify their end-to-end quality requirements through adding extra constraints to the allocation rule. The price-based formulation assumes that service requesters generally have a clear understanding of what level of quality is acceptable for them, for example what should be the maximum response time of the composite service or its minimum availability. This assumption is less restrictive than the assumption about the weights for quality attributes, especially when more than two or three quality attributes are involved.

Nevertheless, each formulation is suitable for a different group of service requesters. If a requester is interested in maximizing the quality of the requested service and they have clear understanding of the trade-off between quality attributes, the utility maximization formulation can be adopted. Whereas if the requester's main concern is the cost of the procurement or they do not have specific preferences toward the priority of different quality attributes, the cost minimization formulation is suggested. Adopting such a perspective, He et al. (2014) has discussed both formulations, although the experiments only cover the utility maximization formulation.

As the set of allocation constraints to be included in the auction's set of allocation rules, researchers have generally considered the requester's constraints regarding the end-toend quality of the composite service; for example constraints over the response time, availability or budget of the composition.

In addition to this set, Mohabey et al. (2007b) has proposed a constraint over the interface of the sequential services in a composition to ensure interface matching. However, their model assumes that providers have specified the interface of each service (with all its complexity regarding the data structures) by a global identifier which allows the model to match the interfaces against one another. There is no discussion on if such an assumption can be supported by the existing web service specification standards such as WSDL.

3.8.4 Payment Rule

The question of specific payment rule has not been considered in most of the auctionbased proposals for service selection. This is mainly due to the fact that mechanism designers in general and auction designers in particular design the payment rule to achieve specific properties in the auction model, such as incentive compatibility or allocative efficiency, as discussed in section 2.4 (Mechanism Design).

Mainly focusing on the application of auctions to solve the composite service selection problem as a resource allocation problem, researchers have mostly not addressed the mechanism design perspectives; neither to design an auction with specific properties, nor a post-design analysis of the properties of the proposed auction models. Moreover, composite web service selection is known to be a very complex problem. This complexity makes the analysis of the incentives of service requesters and providers far from being trivial, especially if the problem is modelled as a combinatorial auction. The existing well-known incentive-compatible mechanisms are not directly applicable to this problem. We will discuss this issue in more details in subsection 7.2.2 (Economic Efficiency and Incentive Compatibility).

Two of the researchers who have adopted the mechanism design perspective in designing their auction models are Blau et al. (2010) and Watanabe et al. (2012). Both proposals are based on non-combinatorial auction models.

The payment rule proposed in Blau et al. (2010) is an extension of the VCG payment and it is said to be incentive-compatible for providers across all aspect of their bids (price and quality). However, the proposed formulation of the objective function includes elements from the providers' bids as well as the composite service request. This means that a requester can increase their utility by strategically changing their request, either in terms of their declared willingness to pay (monetary) or the scoring function for the composite service quality profile. In other words, the mechanism is not incentive-compatible for requesters.

The Vickrey auction proposed by Watanabe et al. (2012) uses an extension of the Vickrey payment: the provider with the best quality wins the auction, but only offers the quality level of the second best offer. As the authors have already discussed in the paper, extending the original Vickrey auction where the bidders only bid for the price, to a multi-attribute auction does not necessarily holds the truthful property of the original mechanism. The abstract discussion included in the paper on the incentive-compatibility of the proposed mechanism is based on unclear assumptions and lacks theoretical foundations.

In general, when no specific payment rule is discussed, we can assume that the payment rule is the same as the first-price auction model where the bidder with the highest bid wins the auction and pays the price they have mentioned in their bids (Parsons et al. 2011).

| Research | Auction Model | Bidding Language | Allocation Rule | Payment Rule |
|--|--|--|--|------------------------------------|
| (Tang 2004) | No specific auction model; analyzing service providers' best strategies | - | - | - |
| (Esmaeilsabzali and Larson 2005) | Reverse; Non-combinatorial | Multi-dimensional | maximize quality subject to budget constraints | - |
| (Lamparter and Bjorn Schnizler 2006) | Conceptual architecture | - | - | - |
| (Mohabey et al. 2007a) | Reverse; Combinatorial | Multi-dimensional | maximize quality (no price), subject to budget constraints | - |
| (Mohabey et al. 2007b) | Reverse; Combinatorial | Multi-dimensional (quality and interface) | minimize cost, subject to quality and interface matching constraints | - |
| (Lamparter 2007) | Direct; Combinatorial | Multi-dimensional (quality and pricing function) | maximize profit for providers | - |
| (Vilenica et al. 2011) | Conceptual framework | | - | - |
| (Prashanth and Narahari 2008) | Reverse; Combinatorial | Multi-unit, QoS not considered | minimize cost for k execution of the composite service | - |
| (Blau et al. 2010) | Double; Non-combinatorial | Multi-dimensional | maximize utility across all participants | An extension of VCG payment |
| (Vilenica et al. 2011) | Conceptual framework | - | - | - |
| (Watanabe et al. 2012) | Reverse; Non-combinatorial; followed by a round of QoS adjustment | Multi-dimensional | maximize utility locally, followed by a global constraint resolution phase | Extension of Vickrey payment |
| (He et al. 2014) | Multi-round; Reverse; Combinatorial | Multi-dimensional, XOR combination | maximize quality subject to budget constraints; or minimize cost subject to quality constraints | - |

3.9 Conclusion and Discussion

In this chapter, we presented an extensive review on the current approaches for composite service selection: optimization-based, negotiation-based and auction-based. The service selection approaches are categorized based on their assumptions about two aspects of the quality profile of the web services: dynamicity of the profile and the complexity involved in quality determination.

Optimization-based approaches are limited by the assumption of the web service offers quality and price to be pre-determined. One implication of this assumption is that the complexity of the price determination is completely left to service providers. Such a pricing strategy may not be profitable for service providers considering the specific characteristics of web services and their execution environment. Negotiation-based approaches are an improvement over the completely pre-determined profile assumption in optimization-based approaches. However, to avoid the complexity of an automated negotiation process, these approaches are based on simplifying assumptions about the strategy space of the automated negotiators of service providers and requesters. This makes the application of these approaches somewhat unrealistic, at least for the near future.

Auction-based approaches are based on theories and models adopted from economics and auction theory. Application of auction models facilitates the price determination for service providers and composite service requesters through providing feedback to the auction participants. Moreover, combinatorial auctions allow the service providers and requesters to express their preferences for bundle of services more fully by allowing them bid for a combination of services. This is specifically important for composite service selection as the web services constituent a composite service are dependent on each other based on a number of factors. These dependencies create complementarity effects among these services, which in turn, affects the service providers and requesters preferences for bundles of services.

Some important issues have not been adequately addressed in the current literature of composite service selection. Firstly, the impact of bundling (more specifically the bundle size) on the cost of composite services is not studied in the current literature. Secondly, while many of the current approaches have considered the requester's need to specify

end-to-end quality constraints for quality attributes such as response time, availability and budget, important requirements regarding the dependability of the composite service to the service providers has not been addressed. Finally, the impact of the presence of multiple requests on the composite service selection approach has not been studied in the literature. This is an important issue to consider in composite service selection, especially in the context of the design and maintenance of web services marketplaces where service providers and requesters attend the market to trade single and composite services.

Chapter 4

4 Theory and Methodology

4.1 Introduction

In this chapter, we discuss the theoretical foundations of our research and the research methodology that we developed to conduct our study. First in section 4.2, we discuss the four pillars of our research: the web service technology, auction theory, mechanism design and mathematical optimization.

Then in section 4.3, we introduce the research methodology that we followed to conduct our research. This methodology consists of three steps: (1) designing an auction-based approach to composite service selection problem, (2) develop a model of the proposed auction-based approach, and (3) perform evaluation on the proposed approach.

4.2 Theoretical Foundations

Resource allocation in modern distributed computing paradigms such as grid computing, pervasive computing and, more recently, cloud computing is characterized by its complexity and the inadequacy of classical approaches in handling these paradigms. The complexity arises from two factors: the *large number* of users, applications and machines involved, and the *heterogeneity* of these entities (Ferguson et al. 1996).

The complications of resource allocation in these environments have led researchers to look for new methods based on economic models. This direction comes from the fact that these models have already proven to be successful allocation mechanisms in human economies, the complexity of which exceeds that of any computing environment. In this regard, auction theory is the leading approach into which researchers have delved to solve a variety of problems in computing. It has been employed in areas such as packet routing, load balancing, data replication and migration, and coordination of robot navigation. More recently, leading companies have used auction models to improve their services. For instance, Amazon has used auctions to allocate resources in the cloud (Amazon 2014), and Google and Yahoo assign advertising spaces based on auctions over search keywords (Varian 2007).

The use of auction-based mechanisms to solve complex computing problems has been elevated by the recent advancements in the field of *mechanism design*. Being a sub-field of microeconomics and game theory, the mechanism design focus is on solving problems which involve multiple, rational, self-interested participants with private information about their preferences. The objective of mechanism design is to "design" a mechanism to achieve a "given" outcome. The basic question in mechanism design is how to design an economic system so that the selfish behavior of the participants leads to desirable properties such as efficiency, optimality and individual rationality. These properties were discussed in subsection 2.4.1 (Auction Properties).

With the help of the theoretical accomplishments in this area about what is possible and what is not possible to achieve in an auction, auction designers can more systematically analyze and predict the behavior of participants. Subsequent to the analysis, it is possible to design auctions with appropriate incentives for participants to behave in a certain way, so that the auction achieves its designated desirable properties.

Viewing composite service selection as a resource allocation problem, in this thesis we investigate the use of auction theory to solve this problem. Our work addresses the complex issues in composite service selection by combining and extending theories and technologies from various fields of research in a novel way. The present study is at the intersection of web service technology, auction theory, mechanism design and mathematical optimization:

4.2.1 Web Service Technology

Web services have emerged as the potential silver bullet to realize the service-oriented computing architecture (Oh et al. 2008). The specific characteristics of web services, in terms of their interface and communication technology, have led researchers to the

interesting idea of composing them together to build more complex composite services, a process known as web service composition (WSC). An essential stage in WSC is the service selection. The objective is to choose the set of web services that can collectively deliver the required composite service, and satisfy the service requester's preferences and constraints the most.

This thesis studies composite web service selection, focusing on important issues which are yet to receive the attention of the research community, namely, the dependencies that exist between web services that form a composition, the need to more dynamic price determination strategies for web services and the presence of multiple, simultaneous, requests for composite services.

4.2.2 Auction Theory

Auction theory is an applied branch of economics that studies the behavior of an auction's participants who are aiming to buy or sell some products or services. Auctions have been proven to be efficient mechanisms for allocating products and services in complex everyday settings. The complexity involved in the modern distributed systems has led researchers to employ auction theory to solve a variety of resource allocation problems in these environments.

We have explored the application of auctions to solve the composite service selection problem. The dynamic pricing strategy in auctions is a response to the challenge of price determination for web services. We have modelled composite service selection as a combinatorial procurement auction where the service requester is the auctioneer (or, more precisely, bid-taker) and service providers are the bidders. They place their bids to offer their services for composite services. The combinatorial aspect of the auction allows service providers and requesters to explore and exploit the dependencies between constituent web services to meet their preferences and interests.

4.2.3 Mechanism Design

As a sub-field of economics and game theory, mechanism design studies the problems with multiple self-interested participants. The objective is to design a mechanism that can achieve the desirable properties decided by the mechanism designer, despite the selfish behavior of the participants. Mechanism design provides researchers with the theoretical analysis of bidders' behaviors and their incentives. Firstly, such analysis helps in understanding what is possible to achieve in an auction and what is not possible. Secondly, it gives insight into the required incentives for mechanism participants to behave in a certain way. Altogether, the mechanism designer will be able to design a mechanism that can achieve its desirable properties through installing the right incentives. The theoretical analysis offered by mechanism design is specifically valuable for researchers whose expertise resides outside microeconomics, who are interested in applying economic models in their problem domain.

In this thesis, we have studied the composite service selection problem through the lens of mechanism design. Initially, we identified the specific requirements of this problem to an auction-based solution. These requirements differentiate the auction for composite service selection from existing auction models in other domains. Based on these requirements, we have designed auction models that solve the composite service selection problem and its extension, that is, the multiple composite service selection problem. We have applied mechanism design to analyze our proposed auctions in terms of their achievable set of desirable properties as well as what is not possible to achieve.

In this regard, we will discuss achievable and desirable properties of the proposed auction models in subsection 4.3.1.3 (Payment Rules). The limitations of the proposed models in terms of the desirable properties are discussed in section 7.2.2 (Economic Efficiency and Incentive Compatibility).

4.2.4 Mathematical Optimization

We have mapped our proposed auction-based mechanism for composite service selection to an integer linear programing (ILP) optimization problem. In the ILP problem, the objective is defined based on the problem domain requirements to minimize the cost of composite service provisioning. The ILP formulation allows us to incorporate allocation constraints and the requester's preferences into the optimization problem as search constraints. Moreover, many commercial and open source ILP optimization software packages are available that can solve very complex ILP problems efficiently. We used a commercial solver application, known as CPLEX¹³. CPLEX, the same as any other solver, requires two inputs to perform the search for an optimal solution of an optimization problem: (1) a specification of the optimization problem, (2) the data to be searched for the optimal solution.

We have implemented the optimization problem, that is the ILP formulation of the proposed auction-based model to composite service selection, in a modelling language called AMPL¹⁴ (Fourer et al. 1990). AMPL is an algebraic modelling language for describing and solving complex optimization problems. The simulation data of the web service offers and composite service requests are also specified based on AMPL standard for data representation. CPLEX takes the ILP formulation of the auction-based composite service selection and, based on this model, searches for the optimal set of web services to be allocated to the composite service requests.

4.3 Research Methodology

To address the proposed research objective discussed in section 1.3, we employed a research methodology with three steps: (1) designing the auction-based mechanisms that solve the (multiple) composite service selection problem, (2) modelling the proposed auction-based mechanisms, and (3) evaluating the proposed models through simulations.

4.3.1 Designing the Auction-based Mechanism

The design of an auction-based mechanism for the composite service selection problem requires answering the following two questions (Fig 4.1):

- 1. What does an auction-based approach mean? What are the elements that build up an auction model?
- 2. There are already a variety of auction models, standard and arbitrary, that have been applied in other domains such as transportation, communication networks, resource scheduling. How an auction-based mechanism in

¹³ < <u>http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/</u> >

¹⁴ A Modelling Language for Mathematical Programming <<u>http://www.ampl.com/</u>>

composite service selection is different to other existing auction-based mechanisms?

To answer the first question, we studied a variety of auction models designed for different domains, in addition to the auction theory literature. The study helped us to identify the auction design elements which were discussed in subsection 2.4.2 (Auction Design Elements). The elements are: the bidding language (the auction protocol), allocation rules (the winner determination problem) and the payment rule (pricing scheme).



Fig 4.1. Designing an auction-based mechanism for composite service selection

The second question needs to be answered based on the specific characteristics and technologies associated with web services and the current approaches for the web service composition and the composite service selection problems. These specific characteristics differentiate an auction model for composite service selection from other existing models. They help us establish the specific requirements of our problem domain to an auction-based solution. In other words, they are the "requirements of the composite service selection problem" to an auction-based solution. The requirements are presented here, categorized based on the design elements of an auction model.

4.3.1.1 Bidding Language

Req 1. The bidding language must support multi-attribute bidding.

An important aspect of web services is the non-functional properties or quality of service attributes (QoS). These attributes are the constraints exhibit over the service functionality (O'Sullivan et al. 2002). Two providers that offer the same service functionality may have different values for the QoS attributes of their services. These attributes model the competitive advantage that providers may have over each other (Medjahed and Atif, 2007). Therefore, the design of the bidding language needs to support more than the traditional price-only bids. In addition to the price, bids specify the values offered for other quality attributes such as response time, availability and reputation.

Req 2. The bidding language must support combinatorial bidding.

As we discussed before in subsections 1.2.1 and 3.3, an important issue in composite service selection is the need to consider the dependency between services constituent a composition. The providers need to be able to bid for a combination of services to fully express their preferences. Thus, the bidding language needs to support multi-item bidding. Moreover, each provider should be able to submit multiple bids and there is no restriction on the number of winning bids of a provider.

4.3.1.2 Allocation Rules

Req 3. The auction model is a procurement auction (one buyer, multiple sellers).

Our design is based on the reverse or procurement auction models rather than a direct auction. The reason is that in composite service selection, it is the service requester who requires a set of different services to achieve a specific goal and it is very likely that these services need to be procured from different providers. Therefore, if the auction is designed as a direct auction, with service providers as bid-takers and service requesters as bidders, the service requesters may need to attend different auctions to procure all their required web services and, more importantly, win in all these auctions to be able to create the composition. Even if they win in all the auctions except for one, the composite service cannot be realized and the service requester has to withdraw from all other auctions. In most auction settings, withdrawing from an auction after winning it is not allowed or incurs a withdrawing cost.

Therefore, the design to suit our problem domain is the procurement model where the service requester is the bid-taker (auctioneer) and the service providers bid to offer their services. The auction is considered successful only if there are web services available for all the tasks of the composite service satisfying the requesters' preferences and constraints. Consequently, the requester can commit to the result of a successful auction without concern for unwanted costs.

Req 4. Free-disposal does not exist.

To create the composite service, the service requester needs all the tasks to be successfully auctioned and find service providers to provision them. In auction theory, this is referred to as an auction *without free disposal*: the auctioneer has to sell (procure) all the items and the bidders cannot accept more than what they had bid for (Sandholm et al. 2002). Lack of free disposal makes it difficult to apply approximation methods for reducing the complexity of the problem which will be discussed in subsection 7.2.1, the time limitations of the proposed approaches.

Req 5. The auction model is a combinatorial auction.

The proposed design is based on combinatorial auction models. As discussed in subsection 2.3.3 (Combinatorial Auctions), in this model multiple items can be auctioned simultaneously and bidders can bid for combination or bundle of items. This auction model is important when there are dependencies between the items under auction: either they complement each other or can be substitute for each other.

As discussed in subsections 1.2.1 and 3.3, web services constituting a composite service are dependent on each other based on factors such as the sequence of execution time, resources consumed, input/output message or data, and user-specified constraints. These dependencies can create complementarity effect among the services which makes it attractive for service providers to offer them in bundles. As an example of the complementarity effect, consider a service provider who is interested in providing services for a set of consecutive services exchanging data. By provisioning for these dependent services and bidding for them as in one bundle, the provider can internalize some of the costs of interface compatibility required for data exchange. This can decrease

the cost of service provisioning. Consequently, the discount in the bundle's price can result in the provider's increased competitiveness in the market for web services. Moreover, they can offer better qualities for the bundled services by having more control over the execution environment of the adjacent services in the composition.

Req 6. The auction's objective is a single attribute optimization problem, based on the price.

Req 7. Quality of service constraints need to be supported in the allocation rules.

The objective function of the auction is designed to include only the price, rather than all the quality of service attributes. The price-based design leads to a cost-minimization objective function rather than a utility-maximization one, as discussed in subsection 3.8.3. The service requester's requirements on other quality of service attributes are considered as allocation constraints to be taken into account while searching for the optimal solution.

It seems natural to assume that service requesters are mainly concerned about quality attributes meeting some criteria. In other words, service requesters can easily state their desired level of quality in terms of their minimum expectation from the quality of the service, rather than having a clear and perfect utility function that specifies the weight of different quality attributes toward each other. As discussed in subsections 3.3 and 3.8.3, eliciting these weights has been one of the challenges for the utility-maximizing approaches to composite service selection. An objective function aiming at maximizing the utility of user regarding the different quality attributes usually forces the researchers to include unrealistic assumptions on the model; such as the weights being known for the requesters and the quality attributes not being correlated.

Therefore, it is easier and more realistic to assume that instead of specifying weights for quality attributes, the service requester is interested in specifying the concerns they have regarding the quality level; such as what is the maximum response time acceptable or minimum availability required. In such a setting, the multi-attribute characteristics of web services have been taken into account, without having to deal with complexities of a utility-maximization objective function.

At the same time, if a requester cares about quality and aims to maximize the quality at the cost of more expensive services, they can achieve this objective even with the cost-
minimizing formulation by specifying very high expectations on the quality of service levels.

Req 8. The auction model is single-shot and not iterative.

Having an iterative auction for a composite service means that the providers have to submit their bids, wait for the result of the first round of the auction, based on the provided information about the results of the first round revise their bids and re-submit the bids. They need to continue to do so until the final round of the auction. However, in the auction for service selection, the items under auction are web services which mostly offer small, limited, functionality at a relatively low price. Therefore, the service providers would likely prefer to attend more auctions for different composite services rather than spending more time (for evaluating their bids and improving their strategic behavior based on the result of the previous round) in a multi-round auction for the same composite service.

4.3.1.3 Payment Rules

Req 9. The pricing scheme of the proposed model is similar to a first price auction; the winners receive the amount they have bid.

Auction designers use the pricing scheme to install properties such as incentive compatibility in the mechanism. As discussed in subsection 2.4.2.3, the well-known incentive compatible mechanism for multiple items is called the Vickrey Clark Grove (VCG) mechanism. The payment rule in a VCG mechanism is so that any winner's payment is independent from their own valuations for the items (their bids).

However, the VCG mechanism has serious drawbacks that make its application rather impractical, including: making bidding very complex for bidders, needs the bidders to reveal many information about their valuations, possibility of very low revenue outcome, highly susceptible to collusion, and most importantly not being budget-balanced which means that the mechanism need to be subsidized from outside.

Therefore, although in theory it is possible to adopt the VCG payment to achieve an incentive-compatible mechanism, it will not suit practical applications. As a result, we decided to follow the first price auction model for the payment which, in our case, means that the service providers will be paid the amount they have bid for if they win the auction, and zero otherwise.

As discussed in subsection 2.4.1.7, the impossibility theorem in mechanism design (Myerson & Satterthwaite 1983) states that it is impossible to design an exchange mechanism which is incentive compatible, (interim) individually rational and budgetbalanced that achieves efficiency in equilibrium. In this regard, the first price payment rule leads to an auction model which has the individual rationality and budget- balanced properties, but not the economic efficiency and incentive compatibility.

Individual rationality and budget-balanced are both very important in designing a mechanism with practical application. An auction with individual rationality does not leave any of the participants worse off, than had they not participated in the mechanism. An auction which is budget-balanced does not need subsidy or fund from outside. We will discuss the limitation of the proposed approach on incentive compatibility and economic efficiency later in subsection 7.2.2.

4.3.2 Modelling the Proposed Auction-based Mechanism

We have mapped the proposed auction-based mechanism to an integer linear programming (ILP) optimization problem. Such an approach allows us to look for the optimal solution based on the objective function (minimize the cost of procuring the composite service for the composite service requester) and the allocation constraints (the quality of service requirements of the composite service). Beside the quality of service requirements, the service requester can incorporate any other preferences about the provisioning of the composite service into the ILP model by defining appropriate allocation constraints. Moreover, there are many, commercial and open source, optimization solvers available that can solve the ILP problems efficiently up to a reasonable size of the problem.

Nevertheless, scalability remains as an issue of such an approach which looks for the optimal solution especially with our problem domain being an NP-complete problem. This limitation is later discussed in subsection 7.2.1 (Solve Time).

4.3.3 The Simulation-based Evaluation Process

With the absence of publicly available data sets about web services' prices, bundling of web services and requests for composite web services, we based our evaluation on conducting simulations. The proposed simulation-based evaluation process is presented in Fig 4.2.



Fig 4.2. The evaluation process

The objective of the simulation is to analyze the performance of the proposed auctionbased composite service selection mechanisms (*single* and *simultaneous*) in allocating web service offers to one or more composite service requests. The performance metrics have been defined based on the important criteria in this problem domain which are: (1) the success rate of the mechanism in allocating service offers to requests, (2) the final cost of composite service provisioning, and (3) the time to find the optimal allocations for the requests.

However, due to the limited empirical research on composite service selection with bundling or composite service selection in the presence of multiple requests, the design of the simulation, in terms of the data generation model and seeding of the parameters, was a challenging for our study.

We initially designed the experiments to evaluate the proposed "single auction" mechanism (to be discussed in Chapter 5). Based on the experience from these experiments, we revised the design of the experiments and improved it with more realistic scenarios and seeding data for the evaluation of the proposed "simultaneous auction" mechanism (to be discussed in Chapter 6).

In our experiments, the two essential elements of the simulations are the data generation model and the seeding of the simulation's parameters. For the data generation model, we initially used an existing package, CATS suite (Leyton-Brown et al. 2000), that generates combinatorial bids for combinatorial auctions. However due to the limitations of this package, later we decided to design our own data generation model to have full control over the data set to be able to accommodate the specific requirements of our problem domain, that is, the (multiple) composite service selection problem. Nevertheless, our designed data generation model is based on the data distributions applied in combinatorial auctions literature. The final data generation model is discussed in subsection 6.4.4.

For the seeding of the data generation model, we initially referred to the existing experiments in the service selection literature. However, the limited studies and experiments on service selection with bundling of web services and multiple requests for composite services led us to look for more realistic data on the Internet.

In this regard, we studied a number of web services' communities on the Internet where service requesters need composite services. Based on the information we collected from these directories, we decided to perform the experiments of the simultaneous auction mechanism in specific market sections of web services rather than a general market. Without focusing on specific market sections, it is rather difficult to estimate the number of service providers, requesters and the type of the web service offers and requests in a generic market for web services.

The market sections are designed based on two factors dividing the web services marketplaces into the following categories:

- The market economy size, that is the magnitude of the number of participants in a market and categorizes the markets into small economy and large economy (Tang 2004),
- (2) The composite service complexity, which divides the markets to market for simple composite services and market for complex composite services (Weinhardt et al. 2011a).

The resulting four market sections are presented in Table 4.1 below. The simplest section is the market section with small economy (limited number of service providers and requesters) and mainly trading simple composite services. An example of such a market

can be a newly-formed market for mobile applications. The most complex section is the market where many service providers and requesters participate to trade complex composite services. The example for this section can be the rather mature web service community around Life Science with the web service registry called BioCatalogue¹⁵. This directory currently has more than 790 active members who are mainly interested in complex scientific workflows. More details about these market sections and motivational scenarios are presented in subsection 6.4.3 (Scenarios to Investigate: Market Sections).

 Table 4.1. The four market sections

| Composite Service Complexity Economy Size | SIIMPLE | COMPLEX |
|---|--------------|---------------|
| SMALL | small-simple | small-complex |
| LGRGE | large-simple | large-complex |

These market sections and the number of web service providers and requesters in the existing web service communities helped us establish the seeding data for the simulation parameters in the data generation model.

In Table 4.2, we have presented a summary of the implementation detail of the evaluation process design element; which are the performance metrics, the baseline for comparison, the scenarios to be investigated (market sections), the simulation's data generation model, and the seeding of the simulation parameters. More details are presented for the single auction approach in section 5.2 and for the simultaneous auction approach in section 6.4.

The proposed evaluation process in general, and the data generation model and the four market sections in particular provide a useful framework for the service selection community. The designed evaluation process improves the transparency of the experiments in the composite web service selection domain by identifying a number of elements that need to be supported and their implementation need to be clarified when conducting empirical evaluations.

¹⁵ < <u>https://www.biocatalogue.org/</u>>

Table 4.2. The implementation of the evaluation process

for the proposed auction-based mechanisms

| Evaluation | Single Auction | Simultaneous Auction |
|---|--|--|
| Element | (Chapter 5) | (Chapter 6) |
| Performance Metrics | Success rate of the mechanism Cost of procuring a composite service Solve time | Success rate of the mechanism Cost of procuring a composite service Cost of procuring a task Cost homogeneity across different requests Solve time |
| Baseline | A non-combinatorial auction (single-item bids) | The single auction mechanism when applied to a set of requests one at a time A fixed-price mechanism where the requesters determine the price to be paid for the composite service |
| Scenarios to Investigate (Market sections) | Generic | Four market section based on two factors: The complexity of the requests (the number of tasks in a request) The economy size (the number of requesters and providers attending the market section) |
| Data Generation Model | CATS arbitrary distribution: number of and choice of services in a bundle IPV and CM discounted pricing function: pricing of services and the bundles Decay distribution: number of and choice of services in a bundle | IPV discounted pricing function: pricing of services and the bundles Decay distribution: number of and choice of services in a bundle Uniform: number of tasks in a request |
| Seeding | Similar experiments in the literature | Similar experiments in the literature Data available from the web services' communities on the Internet |

The data generation model and the experimental scenarios establish the basis for more realistic data sets by considering the service selection requirements in the creation of the data set and relying on the data obtained from existing web service communities for seeding the simulation.

4.4 Conclusion

In this chapter, we first discussed the theoretical foundations of our research which are: web service technology, auction theory, mechanism design and mathematical optimization. Then, we introduced our research methodology which comprises three steps: design, modelling and evaluation. The design includes two parts: (1) identifying the design elements of an auction model, and (2) differentiating an auction model for composite service selection from other auction models by studying the web services characteristics in general and the requirements of the composite service selection problem to an auction-based solution in particular. Then, we discussed our modelling approach which is based on mathematical optimization, or more specifically integer linear programming (ILP). Finally, we presented the evaluation process which we designed to evaluate our proposed auction-based mechanisms for composite service selection. Due to lack of publicly available data sets of web services' prices, bundling or composite service requests, we based our experiments on simulations.

Chapter 5

5 An Auction-based Mechanism for Composite Service Selection

5.1 Introduction

In this chapter, we present an auction-based mechanism to solve the composite service selection problem for a single composite service. We refer to this mechanism as the *"single auction"* mechanism. The proposed mechanism is based on multi-attribute combinatorial procurement auctions and is formulated as an integer linear programming (ILP) problem.

In section 5.2, we discuss the requirement of considering the dependencies between the constituent services in a composition and the impact of these dependencies on the service providers and requesters' preferences and constraints. We also introduce and develop two important constraints regarding: (1) the cohesion of the composite service, and (2) the configuration of the providers participating in the service composition.

The design of the proposed auction-based model is presented in section 5.3. The design includes the three elements of an auction model: the multi-attribute bid specification, winner determination problem which is mapped to an integer linear programing problem and a discussion of the payment rule.

The proposed model is evaluated through simulation-based experiments and the results are presented in section 5.4. The objective of the experiments is to study the performance of the proposed mechanism in terms of its success rate, the provisioning cost of the

composite service and the running time of the proposed mechanism to find the optimal solution.

Parts of this chapter have been previously published in (Moghaddam et al. 2013). The theory and design discussions have been revised and updated, and a new experiment has been added which will be discussed in subsection 5.3.2.3 (Stage 3).

5.2 A Mechanism based on Combinatorial Auctions

Auctions, as market-based mechanisms, allow for dynamic pricing which is critical for products such as web services that are characterized by dynamic execution environments (in terms of the provider's available resources), and users with different and changing demands.

In combinatorial auctions, multiple distinct items are auctioned simultaneously and the bidders can bid for a combination of items, or bundles. Bundling enables the bidders to express their preferences for the items more fully, which leads to economic efficiency and greater auction revenue (Cramton et al. 2006). The possibility of bundling is particularly important when bidders have preferences not just for specific items but for bundles due to the complementarities or substitutability effects that exist among the items (de Vries and Vohra 2003). The dependencies can make the utility of a bundle greater (when items are complements) or smaller (when they can be substitutes) than the sum of the utilities of the individual items.

As discussed in subsection 1.2.1 and also 3.3, an issue with current QoS-based composite service selection approaches is that they mostly assume that each provider offers a single service. Even if they offer more than one service, the offers are considered to be independent, that is, there is no possibility of offering bundles of services. Such an assumption ignores the dependencies between web services forming a composition.

Furthermore, most service selection approaches allow the service requester to specify their constraints over a specific set of quality attributes of the composite service, including response time, availability, reputation or budget. While an important point to consider about a composite service is that it is very likely to have several service providers being involved in the provisioning of the composition, and therefore, the service requester may have complex preferences over the number and the configuration of the providers involved in the provisioning of the composite service.

To overcome these problems, we developed *an auction mechanism* for composite service selection based on combinatorial auctions. The auction mechanism views the composite service requester as the auctioneer and the service providers as the bidders who bid to offer their services for the composite service. The proposed mechanism allows the providers to offer their services as single services or as bundles of related services. We have also enabled the service requester to specify preferences about *the cohesion of the composite service* and *configuration of the providers* involved in the provisioning of the composite service through two allocation constraints.

5.2.1 Cohesion in a Composite Service

In general, cohesion can be defined as "a measure of the *bindings* of the *elements* within a single *module*" (Eder et al. 1994). Cohesion has been defined more specifically for: procedural systems (Stevens et al. 1974), the object-oriented paradigm (Yourdon and Constantine 1979; Briand et al. 1998) and service-oriented systems (Papazoglou and Van Den Heuvel 2006; Perepletchikov et al. 2007). To the best of our knowledge, no study has considered the composite service requester's need to manage the cohesion of the composition.

In defining cohesion for service-oriented systems, a service is the main design construct to apply encapsulation and abstraction principles. In a composite service where services are provided by different providers, another abstraction level is the offered bundle of services. We characterized the cohesion for a composite service based on the bundles as the abstraction level. We defined the bindings between services based on their direct data dependency as it is considered to be one of the most important types of inter-service dependencies. Its importance is due to the fact that ultimately, at the lowest level, the connection between services is through mapping the input and output messages between the partner services' ports (Milanovic and Malek 2004).

Following this, we define the cohesion of the composite service based on *direct data dependencies between the services offered in a bundle*. Based on this definition, the scope of a module is a bundle of services offered by a provider. The elements of the module are

the offered web services and the binding of services is measured based on their direct data communication.

Being able to adjust the cohesion level of the composite service is of significant importance for the service requester. Cohesion, as defined here, can directly influence the composite service maintainability, reliability and (provider-) dependability. However, managing the cohesion of a composite service is two-fold: the service requester may be interested in having more data-cohesive modules or less cohesive ones based on the composite service structure, knowledge about the web service's market or user-specific constraints.

Based on "design rule theory" (Baldwin and Clark 2000, p.46), when designing complex systems, it is preferred that the tasks strongly dependent on each other be performed by the same doer who understands the dependency very well. This is known as modular design and its aim is to achieve desirable features such as change manageability and maintainability. For the same reason, the service requester may prefer the strongly dependent services to be offered by the same provider. Based on our definition of cohesion for composite services, this means that such a service requester is interested in increasing the cohesion in the composite service. In this context, maximum cohesion is achieved if all services are provided by the same provider.

However, web services' execution environment is the Internet where communication channels and service providers are not always reliable. Therefore, the service requester may not want to be too heavily dependent on any particular provider to increase the reliability of the composite service. This can be achieved by assigning the dependent tasks to different providers. In other words, the requester may need to reduce the cohesion of the composite service. In this context, the minimum cohesion of the composite service is achieved if no two dependent services are executed by the same provider.

It worth mentioning that choosing the appropriate number of suppliers is an important consideration for any buyer in a procurement process. As discussed in Bichler & Kalagnanam (2006), the buyer needs to determine the minimum number of winning suppliers to avoid depending too heavily on just a few of them. As well, the maximum number of suppliers may need to be kept low to avoid the administrative overhead of managing too many suppliers. In the context of our problem, the number of providers is not a direct measure for provider-dependability due to:

- The inter-service dependencies between constituent services of a composition,
- The involvement of a provider in service composition through providing multiple services.

Therefore, we enabled the service requester to control the dependability of the composite service to the providers by developing an intermediate concept, that is, the cohesion. The management of the level of cohesion is performed by adding a resource allocation constraint to the winner determination problem of the auction mechanism. Our developed cohesion constraint enables the service requester to define a lower and an upper bound for the cohesion of the composite service, relative to the maximum cohesion attainable for the composition (all services are procured from the same provider).

We measure the cohesion of a composite service as the sum of the cohesion of the bundles winning the auction to execute the composition. To measure the cohesion of a bundle, we need to refer to the abstract structure of the composite service, or more precisely, the business process (BP) representing it. Considering the BP structure, two web services have direct data dependency if their corresponding tasks in the business process exchange data.

Therefore, we initially need to identify the data dependencies among the tasks of the BP. In order to do so, we define the business process's *dependency matrix*. For a business process with M number of tasks, the dependency matrix is a $M \times M$ matrix where each of its elements represents the data dependency between two tasks in the BP: if the two tasks need to exchange data, the corresponding element is 1 and 0 otherwise. Clearly, the dependency matrix is a symmetric matrix. This matrix is then used to identify and measure the inter-dependencies between the services executing these tasks.

We define the *data cohesion factor* (*DCF*) to calibrate the cohesion of a bundle. Since in our model, the providers can have more than one winning (non-overlapping) bundles, this means that a provider may have a winning bundle which includes smaller bundles. Therefore, we also need to consider the data dependency between the services of different bundles of the *same* provider.

We have called the cohesion factor related to the dependencies of the services of the same bundle as the *local cohesion factor (LCF)*, and the cohesion between services of different bundles (of the same provider) as the *interactive cohesion factor (ICF)*. Both types of cohesions are calibrated based on the direct data connections between the services according to the dependency matrix. That is, if no two services in the bundle exchange data (based on the dependency matrix), the bundle's data cohesion factor is zero. Otherwise, for any direct data communication, the bundles' cohesion factor is increased by one unit.



Fig 5.1. An example business process with five tasks

| Task in the BP | А | В | С | D | E |
|-------------------|---|---|---|---|---|
| А | 0 | 1 | 1 | 0 | 0 |
| В | 1 | 0 | 0 | 1 | 0 |
| C | 1 | 0 | 0 | 1 | 0 |
| D | 0 | 1 | 1 | 0 | 1 |
| Е | 0 | 0 | 0 | 1 | 0 |

Fig 5.2 The dependency matrix of the example business process

An example business process is presented in Fig 5.1 which has five tasks, named A, B, C, D and E. The BP's dependency matrix is illustrated in Fig 5.2. There are two providers who are bidding to offer their services.

 Table 5.1. Calibrating the local and interactive cohesion factors (LCF and ICF) of the offered bundles based on the BP structure

| Provider ID | Bid ID | Offered Bundle | LCF | ICF _{ij} |
|-------------|--------|----------------|-----|---------------------------|
| 1 | 1 | {A,B,C} | 2 | $ICF_{12} = ICF_{21} = 2$ |
| 1 | 2 | {D,E} | 1 | $ICF_{13} = ICF_{31} = 0$ |
| 1 | 3 | {E} | 0 | $ICF_{23} = ICF_{32} = 2$ |
| 2 | 4 | {A,D,E} | 1 | $ICF_{45} = ICF_{54} = 2$ |
| 2 | 5 | {B} | 0 | |

The offered bundles are depicted in Table 5.1. The local and interactive cohesion factors of these bundles (LCF and ICF_{ij}) are measured and also specified in this table. ICF_{ij} represents the cohesion between bundle *i* and bundle *j* which are offered by the same provider.

5.2.2 Configuration of the Composite Service Provisioning

With the possibility of the involvement of many providers in provisioning of a composite service, a requester may have specific constraints regarding the configuration of the involved providers. We have identified two of such considerations as:

- Some specific services need to be provided by the "same" provider,
- Some specific services need to be provided by "different" providers.

The ability to manage these configurations can be critical for satisfying service requester's concerns regarding security and privacy of the composite service. The two motivating scenarios discussed below explain more clearly the need to consider such constraints:

- Security concern: In a composite service, some of the constituent services need to support an encryption algorithm. In order to minimize the number of providers who have access to the encryption key, the requester may decide to procure all the services with this requirement from the same provider.
- Privacy concern: In a composite service, the collective data provided to two constituent services can reveal a person's true identity despite data anonymization. To preserve privacy, the requester may decide that these services are required to be procured from different providers.

The proposed ILP formulation enables the service requester to place two constraints on the configuration of composite service provisioning in regards to selecting the same or different providers for a set of tasks in the business process.

5.3 The Design of the Combinatorial Auction Mechanism

We have mapped the auction-based mechanism for composite service selection to an integer linear programming problem. In the auction mechanism (more precisely "the

single auction mechanism"), the objective function is to minimize the cost for the service requester, subject to quality and data-cohesion constraints and service provisioning preferences. In our proposed model, we allow the service providers to offer services in bundles. We also enable the service requesters to specify the type of bundles they prefer based on the degree of cohesion required for the composite service and their preferences for the configuration of the composite service provisioning.

The composite service is defined as an abstract business process, comprising a set of tasks. The service providers bid to procure services for these tasks. For simplicity, we have assumed that the BP only includes sequential structures. The existence of other structures (parallel, loop and conditional) in the BP only affects our model in terms of the aggregation functions for quality attributes such as response time and availability. As we have already discussed in subsection 3.3, the problem of mapping the BP's complex structure to an execution path has already been extensively discussed in the literature. Ultimately, all the proposed techniques (which need linear aggregation functions for quality attributes) try to have a sequential execution path. Nevertheless, it is possible to extend the model to these structures following techniques such as the one suggested by Zeng et al. (2004).

5.3.1 Multi-attribute Bid Specification

Let *B* be the set of all received bids from all providers, with an arbitrary member denoted as b_i where the total number of all received bids is *N*. Let *T* be denoted as the set of all tasks in the business process, with an arbitrary member defined as t_j where the total number of tasks in the BP is *M*. Let also *K* be the total number of bidding providers for the tasks in the business process.

Each bid b_i is defined as $b_i = (T_i, Q_i, c_i)$, where c_i is the cost of providing service(s) for the task(s) in the set T_i ($T_i \subseteq T$) and Q_i is the set of the offered quality values for those service(s). Each member of Q_i is a tuple, including the quality attributes' values of the tasks in T_i . For the current model, we considered two quality attributes in the quality tuple: availability and response time denoted as v_i and r_i respectively. v_i and r_i are defined as functions from the set of tasks to a positive number. Thus, we have:

$$Q_i = \left\{ \left(v_i(t), r_i(t) \right) \middle| t \in T_i \right\}.$$

5.3.2 The Winner Determination Problem (WDP)

In WDP (or allocation rule) of the proposed single auction mechanism, the objective function is defined as minimizing the cost for the service requester as specified in function (5-1). The decision variable is denoted as z_i to be 1 if b_i is a winning bid and 0 otherwise. Constraint (5-2) ensures that each task is assigned to no more than one provider. It also implies that any number of non-overlapping bids from the same provider can simultaneously win the auction. This implies that the bidding language is OR. However, providers can have XOR language by adding dummy items to the bids they need to combine them in a mutually exclusive way. This means that that the specified bidding language is in fact OR* types (Nisan 2000). To get the unique assignment, we defined matrix $A_{N\times M}$ with an arbitrary element of a_{ij} which is 1 if T_i (in b_i) includes t_j and 0 otherwise.

The quality constraints over availability and response time are defined in $(5-3)^{16}$ and (5-4), where *V* and *R* are the service requester's acceptable minimum availability and maximum response time for the composite service. The requester's budget constraint is specified in (5-5) where *B* is the requester's available budget for the composite service.

$$Minimize \qquad \qquad \sum_{i=1}^{N} c_i * z_i \tag{5-1}$$

Subject to:

Allocation
constraint
$$\forall j \in \{1..M\}$$
 $\sum_{i=1}^{N} a_{ij} * z_i = 1$ (5-2)

Availability
constraint
$$\sum_{i=1}^{N} \sum_{t \in T_{i}} \ln(v_{i}(t)) * z_{i} \ge ln(V)$$
(5-3)

Response
time
constraint
$$\sum_{i=1}^{N} \sum_{t \in T_i} r_i(t) * z_i \le R$$
(5-4)

¹⁶ The aggregation function of availability is linearized using a logarithm function (Zeng et al. 2004).

Budget
constraint
$$\sum_{i=1}^{N} c_i * z_i \le B$$
(5-5)

Cohesion
constraint
$$MAXC * LC \le \sum_{i=1}^{N} \sum_{j=i}^{N} h_{ij} * z_i * z_j \le MAXC * UC$$
 (5-6)

Provisioning constraint

constraint
(the same
$$\exists k \in \{1..K\}$$
 $\sum_{i=1}^{N} \sum_{j \in X} a_{ij} * d_{ik} * z_i = |X|$ (5-7)
provider)

Provisioning
constraint
(different
providers)
$$\forall k \in \{1..K\} \qquad \sum_{i=1}^{N} \sum_{j \in Y} a_{ij} * d_{ik} * z_i \le 1$$
(5-8)

The service requester's preference for data cohesion is defined in (5-6). The interactive cohesion factor is defined by h_{ij} when $i \neq j$ and both bids *i* and *j* belong to the same provider. When i = j, h_{ij} defines the local cohesion for bid *i*. It will be 0 otherwise. The lower and upper bounds of the required cohesion are denoted as *LC* and *UC* which can be defined in terms of the percentage of the maximum possible cohesion in a BP, *MAXC*. For example, *LC*=100% means the service requester wants the maximum cohesion for the BP, which indicates that they prefer all tasks to be provided by the same provider.

Equation (5-7) ensures that all the tasks in *X* will be provided by the same provider where $X \subseteq T$. Constraint (5-8) ensures the opposite for the tasks in the set $Y \subseteq T$, that is, the tasks in set *Y* will be provisioned by different providers.

In the above formulation, constraint (5-6) is not linear, due to the expression $z_i * z_j$. To linearize the constraint, we replaced this expression with a new decision variable, x_{ij} . This replacement requires the following set of constraints, (5-9), to be added to the model:

Set of constraints
$$\forall i \in \{1..N\}$$

to ensure: $x_{ij} = z_i * z_j$ $\forall j \in \{1..M\}$
 $x_{ij} \le z_j$ (5-9)
 $x_{ij} \ge z_i + z_j - 1$

Another constraint requiring linearization is the constraint (5-7), the provisioning of a set of tasks by the same provider, which is not linear due to the use of the existential quantifier. To make it linear, we have introduced a new decision variable, y_k , which is 1 if provider $k \in \{1..K\}$ is selected to provide services for the tasks in set X, and 0 otherwise.

The linearized form of (5-7) is defined in constraints (5-10), (5-11) and (5-12). In this set, M is defined as a number sufficiently large to guarantee that (5-10) and (5-11) are satisfied when the introduced decision variable y_k is 0. Using such a number is referred to as the Big-M method in linear programming (Padberg 1999, p.54). In these constraints, d_{ik} is an arbitrary member of the matrix $D_{N\times K}$, which defines the mapping of the bids to the providers. d_{ik} is 1 if bid b_i comes from provider k and 0 otherwise. Constraint (5-12) specifies that there should be at least one winning provider that provides services for the set of tasks in X. ¹⁷

The Linearized form of constraint (5-7), the same provider for the set X

$$\forall k \in \{1..K\}$$
 $|X| - M * (1 - y_k) \le \sum_{i=1}^{N} \sum_{j \in X} a_{ij} * d_{ik} * z_i$ (5-10)

$$\forall k \in \{1..K\} \qquad \sum_{i=1}^{N} \sum_{j \in X} a_{ij} * d_{ik} * z_i \le |X| + M * (1 - y_k)$$
(5-11)

$$\sum_{k=1}^{K} y_k \ge 1 \tag{5-12}$$

5.1.1 The Payment Rule

As already discussed in subsection 2.4.2.3, the auction designers employ the payment rule as a measure to install their desired properties in the mechanism. An important property is to achieve economic efficiency which can be achieved if the auction designer knows

¹⁷ As we already specified in the allocation constraint (5-2), there will be exactly one winning provider for each task in the set of solutions.

the true valuation of the bidders for the items under auction, that is, if the mechanism is incentive-compatible.

The Vickrey-Clarke-Groves (VCG) mechanism (Vickrey 1961; Clarke 1971; Groves 1973) is the most notable way to achieve incentive-compatibility and economic efficiency where many items are auctioned simultaneously, such as in combinatorial auctions. Therefore, for the pricing scheme, we draw on the VCG model.

The VCG payment for the single auction mechanism can be defined as in equation (5-13) below:

$$\forall k \in \{1..K\} \qquad p_k = \sum_{j \in B \setminus B_k} c_i z_i^{* \exists k} - \sum_{j \in B \setminus B_k} c_i z_i^* \qquad (5-13)$$

In the above equation, z_i^* are the decision variable values for the optimal solution (z_i^* is 1 for the winning bid, and 0 for others) and $z_i^{* \restriction k}$ are the variable values of the optimal assignment if we remove the bids of provider k from the set of bids. Let $B_k = \{b_i | d_{ik} = 1\}$, that is the set of bids from provider k. We define $B \setminus B_k$ as the set of all bids without the bids of provider k. The price p_k to be paid to the winning bidder k is the result of subtracting the cost of all other winning bids in the optimal allocation, z_i^* , from the sum of the cost of bids in $z_i^{*\restriction k}$, the optimal allocation in absence of bidder k.

With this formulation, the VCG payment to each bidder is independent of the bidder's offered price. In a direct auction, the VCG payment grants a discount to a bidder on their payment to the mechanism (Schnizler et al. 2005). In the reverse auction, this will be a bonus on top of the requested price for the items or services. With such a payment, the bidders have no incentive to strategically manipulate their bids as it cannot improve their gained utility.

With the VCG payment and incentive compatible bidders, the objective function of minimizing the cost of the composite service (based on the bids information) becomes equivalent to maximizing the economic efficiency. The proposed mechanism is also individually rational for participants with quasi-linear utility functions. In the context of composite service selection, a quasi-linear utility function means that the valuation of service providers and requester for the web services do not change if the participants were

poorer or richer, that is, their valuations do not depend on their wealth (Rasmusen 2001, p.202).

Moreover, if the bidders valuation for the items is based on independent private information, and the auction objective is economic efficiency, then the VCG scheme maximizes the revenue to the seller in a direct auction, or minimizes the cost for the buyer in a reverse auction (de Vries and Vohra 2003).

However, we should note that the assumption about having a quasi-linear utility function requires that bidders should not have any budget constraint. In the context of a reverse auction, it means that the buyer should not have any overall limit on the cost of procurement (Ausubel and Milgrom 2006). The existence of such constraint affects the truthfulness of the participants, whether the constraints are known to others or not (Rothkopf 2007). Therefore, the design of a truthful auction for composite service selection by applying the VCG payment needs the service requester not to have any budget constraint, constraint (5-5). Otherwise, the mechanism will not be necessarily truthful.

Despite its interesting properties in theory, the VCG payment has serious limitations that have made its application in real world auctions to be somewhat rare (Ausubel and Milgrom 2006; Rothkopf 2007). We have discussed the problems related to designing an incentive compatible mechanism for our problem domain and the limitations of the VCG payment in this domain more specifically in the concluding chapter in subsection 7.2.2. Two important limitations relate to: (1) the reluctance of web service providers to truth revelation, and (2) losing the budget-balanced property in the auction model.

In the absence of the VCG payment, the common practice is to ask the winners to pay their bids, which is also the common practice in the auction-based approaches to the composite service selection as discussed in subsection 3.8.4. However, one needs to note that without the VCG payment, the mechanism's objective becomes equivalent to minimizing the cost of composite service procurement based on the received bids, which may or may not be truthful.

5.2 Experiment Design

In order to evaluate the proposed single auction mechanism, we performed a number of simulation-based experiments. The objective of the experiment was to study the impact of two important aspects of the single auction mechanism, cohesion constraint and bundle crowdedness, on appropriate performance metrics. Three performance metrics were defined which are:

- The cost of procuring the composite service: calculated as the sum of the prices of the winning bids.
- The success rate: defined as the number of times the auction is successful in finding an optimal allocation of services for the business process over the total number of running the experiment for each combination of simulation parameters.
- The solve time to find the optimal allocation.

The ILP formulation of the single auction mechanism was implemented using AMPL. Then, the ILP model was given to the solver CPLEX 10.0 along with the simulation data (problem instances). The experiments were performed on a computer with 16 processors, each 1600 MHz, and total memory of 24 GB RAM.

The simulation data (problem instances) has two main parts: (1) the data of the business process (the request for the composite service), and (2) the data of the bids, offering services for that process. The business process was generated based on a specific number of tasks (a parameter of the experiment). The combinatorial bids for the experiments were generated by the CATS suite (Leyton-Brown et al. 2000). CATS is a suite of distribution families used for generating combinatorial bids for five real-world domains of combinatorial auctions.

To generate the combinatorial bids, CATS distribution needs two inputs: (1) the number of items under auction, and (2) the dependency distribution. In our domain, the number of items under auction is equal to the number of tasks in the composite service and the dependency distribution is arbitrary. The arbitrary distribution is used to generate bids for domains in which there are arbitrary dependencies between the items being auctioned, as explained in subsection 2.3.3 (Combinatorial Auctions).

Apart from these two input parameters, CATS arbitrary distribution uses a number of parameters to generate the data which have a default value but can be changed based on the problem domain. One such parameter is called "additional_good" which affects the number of items in the bids. The default value for this parameter is 0.9 which mostly generates very large bundles, close to the total number of items in the auction. To make more realistic bids for our problem domain, we experimented with different values for this parameter and we fixed it on 0.75. In the following, we have referred to this parameter as *"the bundle crowdedness"* which is denoted as α .

The pricing function in CATS arbitrary distribution is by default super-additive, that is, the price of a bid is greater than or equal to the sum of the prices of the individual items in the bid. Therefore, in some of the experiments, we implemented a new pricing to accommodate the providers need to offer discount over the bundles' price.

To prepare the experiment data, we developed a java program that takes the combinatorial bid data set generated by CATS, then changes the data if required (for example the price of the bundles), adds other required data (such as the information about the requested composite service), and lastly writes the data (problem instances) to a file based on the AMPL specification for data presentation. Overall, 6,300 problem instances were solved in these experiments.

5.3 Results

The results of the experiments to evaluate the proposed single auction mechanism are presented below in three parts: (1) the impact of the cohesion constraint on the performance metrics, (2) the impact of bundling on the cost of the composite service, and (3) the study of the solve time of the single auction mechanism.

5.3.1 The Impact of the Cohesion Constraint on Cost and Success Rate

The need to manage the cohesion of the composite service has been addressed for the first time in the current study. Adding this constraint to the WDP imposes extra limitation on the auction mechanism. Therefore, it is important to study its impact on the performance of the single auction mechanism. In order to do so, we considered three cases for comparison:

- The service requester is interested in high cohesion for the composite service. In this case, LC, the cohesion's lower bound, is at least 50% or 75% of the MAXC (maximum possible cohesion of the composite service), and no constraint exists for the cohesion's upper bound (UC=1),
- The requester is interested in low cohesion. In this case, UC is at most 25% or 50% of MAXC while no constraint exists for the lower bound (LC=0).
- A baseline with no constraint for the cohesion (UC=1, LC=0).

The configuration of the simulation parameters is summarized in Table 5.2. For each combination of parameters, 30 problem instances were generated and solved. The collected results were averaged across the 30 instances.

| Parameter | Values |
|------------------------------------|----------------------|
| Number of Bids | 100,150,, 300 |
| High Cohesion Constraint | LC={0.5,0.75}, UC=1 |
| Low Cohesion Constraint | LC=0, UC={0.25, 0.5} |
| Number of Tasks in BP | 20 |
| Number of Providers | 50 |
| Bundle Crowdedness (α) | 0.75 |
| Bid Generation Distribution | CATS arbitrary |
| Pricing Function of the Bids | super-additive |

 Table 5.2. Configuration of the simulation parameters for the experiment on the cohesion constraint

The impact of cohesion constraint on success rate is illustrated in Fig 5.3. The results indicated that the success rate of the auction is less than 10% when requester specifies a cohesion constraint as high as LC=0.75. This is due to the allocation constraint that requires the selection of exactly one service for each task. This constraint prevents any two bundles with overlap to win the auction, even if they are offered by the same provider. This leads to the low success rate of the mechanism when high cohesion is required.



Fig 5.3. Success Rate of the auction mechanism for different cohesion constraints

The impact of the cohesion constraint on the cost of composite service is analyzed next. To get an accurate average cost, the unsuccessful instances with zero cost were removed from the results. The results are depicted in in Fig 5.4.



Fig 5.4. Impact of different levels of cohesion constraint on cost

The first observation is that, as expected, the lowest cost is achieved by the baseline where no cohesion constraint exists. It also indicates that the cost of the composite service does not significantly increase when the objective is to achieve low cohesion. In other words, if the service requester needs to reduce provider-dependability, this is achievable at almost no extra cost. In contrast, reaching a high cohesion (LC at least 50% or 75% of MAXC) increases the cost of the composite service on average by 20%. This implies that

having maintainability as a non-functional requirement increases the cost of procuring the composite service.

To delve deeper into the influence of cohesion on cost, we repeated the experiment changing the levels of cohesion for the upper bound and lower bound according to the range [0.2, 0.4, 0.6, 0.8]. We have two groups of data corresponding to: (1) the upper bound (UC) is changing and the lower bound is fixed at a neutral value (LC=0), (2) the lower bound (LC) is changing and the upper bound is kept neutral (UC=1). These two groups correspond to the two lines in Fig 5.5. We have also included a baseline with no cohesion constraint (UC=1 and LC=0) which corresponds to the two points in Fig 5.5.

In this experiment, the number of bids and tasks are fixed at 300 and 10 respectively. For each level of cohesion, 50 independent problem instances were generated and the collected results were averaged.

The result as illustrated by Fig 5.5 confirmed our previous findings: a tight constraint on the cohesion's upper bound (UC=0.2) and lower bound (LC=0.8) increases the cost of the composite service. Compared to the baseline, the increase in the cost is six times for LC=0.8, versus an increase of 1.5 times for UC=0.2. The cost to increase the cohesion dramatically increases at LC=0.6. These findings can help the service requester to set the appropriate level of cohesion considering the trade-off between the cohesion, the cost of the composite service and the success rate.



Fig 5.5. Cost as a function of cohesion

5.3.2 The Impact of Bundling on Cost

An important aspect of the proposed model is that service providers are able to offer their services in bundles. Therefore, we needed to investigate the impact of bundling on the performance of the proposed auction mechanism. The performance metric was defined as the cost of the composite service. The experiments were performed during three stages and the results obtained from each stage directed the design of the next stage.

5.3.2.1 Stage 1

To perform this analysis, we designed an experiment by varying the bundle crowdedness parameter. The configuration of the simulation's parameters is summarized in Table 5.3. For each combination of parameters, 30 problem instances were generated and solved. The collected results were averaged across the 30 instances.

| Parameter | Values |
|------------------------------------|---------------------|
| Number of Bids | 100,150,, 300 |
| Number of Tasks in BP | 20 |
| Number of Providers | 50 |
| Bundle Crowdedness (α) | 0.25, 0.5, 0.75 |
| Bid Generation Distribution | CATS arbitrary |
| Pricing Function of the Bids | CATS super-additive |

 Table 5.3. Configuration of the simulation parameters for the experiment

 on bundle crowdedness, Stage 1

The average cost of the composite service and the cost ratio are plotted Fig 5.6 (a) and Fig 5.6 (b) respectively. The cost ratio is calculated as the ratio of the cost of the composite service to the cost of the first data point which is the cost with 100 bids. The results show that firstly, the cost of the composite service decreases with having more bids, regardless of α . This is due to the fact that with more bids there will be more variety in the service offers which can lead to the procurement cost reduction. Secondly, the cost of the reduction in cost with having more bids is slower when the bundles are more crowded: with the least crowded bundles (α =0.25), increasing the number of bids from 100 to 300 decreases the cost by over 30%, while this decrease for most crowded bundles (α =0.75) is around 15%. To investigate the reason of the second and third observations, we designed the second stage of the experiment.



Fig 5.6. Impact of bundle crowdedness on (a) cost and (b) cost ratio (CATS arbitrary distribution)

5.3.2.2 Stage 2

As we speculated that these observations may have been the result of the specific pricing function implemented by CATS, rather than the impact of the bundle crowdedness, we performed experiments with different pricing functions.

The pricing function provided by CATS is by default super-additive, that is, no discount is considered for the bundle price. Therefore, we developed new pricing functions to consider the impact of having more crowded bundles when providers consider a discount for the bundle. Following the auction theory literature, the pricing functions are developed based on two types of valuation models that providers may have when pricing the individual services:

- The independent private values (IPV) model: in this model, each provider has a private value for their offered services which is not known to the other providers and is also independent from the valuations of other providers for these services.
- The common values (CV) model: in this model, the value of offering a specific service is more or less the same for all the providers, but the estimate of each provider for how much they can charge to sell the service in the market is different between providers (Parsons et al. 2011).

In our developed pricing function, first a (random) value is assigned to each service in the bundle for each provider. Then, the price of the bundle is calculated as the sum of the prices of the services in the bundle minus a random discount specific to that provider. We initially considered the discount to be up to a maximum of 3% of the price of the bundle. The configuration of the simulation parameters for this round of experiment is summarized in Table 5.4. For each combination of parameters, 30 problem instances were generated and solved. The collected results were averaged across the 30 instances.

| Parameter | Values |
|------------------------------|-------------------------|
| Number of Bids | 100,150,, 300 |
| Number of Tasks in BP | 20 |
| Number of Providers | 50 |
| Bundle Crowdedness (α) | 0.25, 0.5, 0.75 |
| Bid Generation Distribution | CATS Arbitrary |
| Pricing Function of the Bids | Sub-additive (discount) |
| Max Discount | 3% |

Table 5.4. Configuration of the simulation parameters for the experimenton bundle crowdedness, Stage 2

To develop a baseline for comparing prices, we generated a set of problem instances with *single-item bids (non-combinatorial)* where each bid offers a single web service. The baseline is established after the current optimization-based and negotiation-based service selection approaches where bundling is not considered.

The results of the experiments with the sub-additive (discounted) pricing functions (the ones that we developed based on the two valuation models) did not lead to a different result from that of the CATS distribution with the super-additive pricing function.

Surprisingly, even when considering a discount for the price of bundles, the cost of the composite service appeared to be higher when providers offered more crowded bundles. The results of the IPV model are depicted in Fig 5.7.¹⁸



Fig 5.7. Impact of bundle crowdedness on cost (discounted pricing function, IPV providers)

However, an important result regarding the comparison of the cost of the combinatorial auction with the non-combinatorial baseline indicated that the price of the composite service is much higher when providers offer single-service bids, compared to when we have bundling with discounted pricing function for the bundles.

Moreover, as illustrated in Fig 5.8, the decrease in the cost of the composite service due to increase in supply (available bids) can reach up to 50% for the combinatorial auctions (α =0.25). While this decrease in cost for the non-combinatorial baseline is around 20%.

We proceeded to the third stage of the experiment to investigate the reason behind the increase in the cost with increasing the size of the bundles.

¹⁸ The CV model demonstrated similar patterns which were presented in (Moghaddam et al. 2013).



Fig 5.8. Impact of bundle crowdedness on cost ratio (discounted pricing function, IPV providers)

5.3.2.3 Stage 3

We initially expected that with increasing the size of the bundles and having a discount over the price of the bundles, the cost of the composite service decreases with increasing the size of the bundles, that is, the cost decreases as α increases. However, our initial experiments did not support this proposition.

To further investigate this matter, we designed another experiment with full control over the generated data set by generating the complete data set rather than using CATS distribution. To prepare a combinatorial bid, there are three elements to consider:

- (1) the number of items in each bid
- (2) the items to be included in the bid
- (3) the price of the bid

We applied a well-known distribution to choose the number and the items in a bid which is called the *decay distribution* (Sandholm 2002). In this distribution, a bid starts with one random item. Then, repeatedly a new random item is added to the bid with probability α (the bundle crowdedness) until an item is not added to the bid or the bid size reaches the maximum number of auctioned items.

Changing the distribution of the combinatorial bids should not affect the behavior of the auction model in finding the optimal allocation. However, different distributions, as listed in (Leyton-Brown et al. 2000), generate different combinatorial instances in terms of the difficulty to be solved, that is, the computational time to find the optimal solution. The

decay distribution is known to be generating some of the hardest instances of combinatorial bids, while CATS distributions are considered to be easy (Sandholm et al. 2005).

The price of the bundle is calculated using the price function developed in the previous stage, assuming IPV providers. We increased the maximum discount to 25% on the bundle price, to eliminate the possibility that the observation under investigation is resulted from the discount being very small.¹⁹ The configuration of the simulation parameters for this round of simulation is summarized in Table 5.5. For each combination of parameters, 30 problem instances were generated and solved. The collected results were averaged.

 Table 5.5. Configuration of the simulation parameters for the experiment

 on bundle crowdedness, Stage 3

| Parameter | Values |
|------------------------------|-------------------------|
| Number of Bids | 100,150,, 500 |
| Number of Tasks in BP | 20 |
| Number of Providers | 5 |
| Bundle Crowdedness (a) | 0,0.1,0.2,,0.9 |
| Bid Generation Distribution | Decay distribution |
| Pricing Function of the Bids | Sub-additive (discount) |
| Max Discount | 25% |

After running the simulation, the number of winning bids with 1 service, the number of winning bids with 2 services and so on was counted. The result is illustrated in Fig 5.9 where the sum of winning bids with a specific number of services in them (one service, up to 20 services) is plotted against bundle crowdedness. The results indicate that most of the winning bids have one service only (almost 92% of all the winners).

To get a better visibility in the diagram, we divided the data sets to three groups (Fig 5.10): first group is the winning bids with one service, second group is the winning bids with two services and the third group collectively represents all other winning bids which have more than two services.

¹⁹ We had repeated the second stage of the experiment with 25% discount which did not change the results.



Fig 5.9. Number of winning bids with 1 service, 2 services, up to 20 services versus alpha

In Fig 5.10, we can see that initially the number of single-service bids winning the auction decreases with increasing α . However, when α increases beyond 60%, the number of single-service winners increases again. This can be explained based on the allocation rule of the single auction mechanism where only one service can be selected for each task in the business process. With this constraint, when bundle crowdedness increases, the bids are more likely to have overlaps in services they offer. Therefore, the chance of single-service bids in winning the auction increases.



Fig 5.10. Number of winning bids with 1 service, 2 services, and more than 2 services versus alpha

We have plotted the cost of the composite service against α in Fig 5.11. The interesting observation is that the cost of the composite service reduces with having more crowded bundles in the auction, up to α =60%. After this point, the trend is reversed and the cost increases.



Fig 5.11. The cost of the composite service versus alpha

The results obtained from these two diagrams, Fig 5.10 and Fig 5.11, collectively suggest that the increase in the cost of the composite service that incurs along with increasing the bundle size is the result of more single-service bids winning the auction. These bids do not include any discount for their services which leads to a higher cost of provisioning the composite service.

5.3.3 Analysis of the Solve Time

We evaluated the solve time of the single auction mechanism using the configuration setting in Table 5.6. The solve time included the CPU time of both AMPL (model generation) and CPLEX (solving time).²⁰ This was averaged across the 30 independent problem instances for each combination of simulation parameters. Fig 5.12 clearly demonstrates the exponential time complexity of the composite service selection problem. With 300 bids and a business process with 30 tasks, the maximum time is around

 $^{^{20}}$ The "CPU clock" time (second) has been measured by adding the values of the parameters _ampl_time and _solve_time.

20 minutes which can be acceptable for requesters who need a large composite service and have no time criticality.

| Parameter | Values |
|------------------------------|---------------------|
| Number of Bids | 100,150,, 300 |
| Number of Tasks in BP | 10,20,30 |
| Number of Providers | 50 |
| Bundle Crowdedness (α) | 0.75 |
| Bid Generation Distribution | CATS arbitrary |
| Pricing Function of the Bids | CATS super-additive |

 Table 5.6. Configuration of the simulation parameters to evaluate the solve time



Fig 5.12. Solve time of the auction mechanism

5.4 Conclusion

In this chapter, we proposed an auction mechanism to solve the composite service selection problem. The single auction mechanism considers some of the important issues in composite service selection which were not fully addressed by the research community. The first issue concerns the assumption of web services being offered independently. Such an approach ignores the dependencies that exist between the constituent services of a composition. Our proposed mechanism is designed based on combinatorial auctions which allow more than one item to be auctioned simultaneously and the bidders to submit their preferences for the items as in bundles. With allowing bundling in the composite service selection, service providers are able to more fully express their preferences by

offering bundles of dependent services. This enables the service providers to improve the quality of service and to reduce the cost for the bundled services. These, in turn, can enhance the providers' competitive power in the market as well as the consumer's loyalty.

Moreover, we addressed the need of the service requester to control the dependability of the composite service to the providers and the maintainability of the composition. This requirement was realized through defining a mediate concept, that is, the cohesion of the composite service. Additionally, we addressed the service requester's requirement to manage the configuration of providers' involvement in the composition through recognizing two patterns: the need for a set of tasks to be provided by the same provider or to be provided by different providers.

The proposed single auction mechanism was formulated as an ILP problem. The identified requirements of the service requester, regarding the cohesion and configurations of providers in provisioning the composite service, were added to the ILP formulation as constraints to be checked while searching for the optimal solution.

We performed extensive experiments through simulation to evaluate the single auction mechanism. The objective of the experiment was to study the impact of two important aspects of the proposed mechanism, the cohesion constraint and bundling, on the performance metrics.

Regarding the cohesion constraint, the results show that with combinatorial bids, it is more expensive to achieve a composite service with high cohesion than one with low cohesion. This is resulted from the allocation constraint which requires exactly one service to be chosen for each task. Therefore, the requester needs to choose the right level of cohesion considering the trade-off between the cost and the cohesion level.

Regarding the impact of bundling on cost of the composite service, our experiments indicated that: firstly, the cost of the composite service is lower when bundling is allowed with discounted bundle prices compared to having non-combinatorial bids (no bundling). Secondly, increasing the number of services in bundles can reduce the cost of composite service up to a threshold. When the bundle crowdedness goes beyond this threshold, the cost begins to increase. We identified the reason to be in relation to the number of single-service bids winning the auction. After the bundle crowdedness passes the threshold, it

becomes too costly to find non-overlapping bundles. This leads to more single-service bids win the auction, and consequently, the cost of the composite service to increase.
Chapter 6

6 A Market for Composite Services

6.1 Introduction

In this chapter, we propose a "*simultaneous auction mechanism*" for the multiple composite service selection problem. The objective of our simultaneous mechanism is to solve the service selection problem for multiple composite services, rather than a single composite service. The "*multiple composite service selection*" problem is an extension of the composite service selection problem toward the vision of a marketplace for web services. In the web service selection literature, this study is the first to consider, investigate and propose a solution to the problem of multiple composite web service selection.

In section 6.2, we discuss the main functions of a market and the significance of a marketplace for web services. Next, the elements of designing such a marketplace are described: the specification of the bidding language which includes the web service offers and the requests for composite services, and the allocation rule.

The details of each element's design are presented in section 6.3. First, the mathematical specification of the offers and requests is presented. Next, we have proposed two allocation mechanisms to match web service offers to composite service requests in a market, named as *Full-Matching* and *Partial-Matching* mechanisms. The objective of the Full-Matching mechanism is to find service providers for "all the requests" in the market at the lowest price, while maintaining the end-to-end quality of the composite services at the requested level. On the other hand, the Partial-Matching mechanism aims to find

providers for "as many requests" as possible, and then to minimize the procurement cost for the set of feasible requests subject to the end-to-end quality constraints. The allocation mechanisms are mapped to Integer Linear Programming (ILP) problems.

In section 6.4, the design of the evaluation process is discussed. The design steps comprise: firstly, establishing a baseline for comparing the results; secondly, defining the performance metrics; thirdly, discussing the motivating scenarios for a composite services' marketplace which leads to the introduction of four major market sections; and, lastly, presenting the stochastic model to generate the simulation data.

The execution details of the experiment are presented in section 6.5. The formulation of the problem chosen for the experiment, the hardware and configuration of the experiment, and the objectives of the experiment are discussed here. The objective of the evaluation is defined as: comparing the performance of the proposed simultaneous auction mechanism to two other mechanisms. The two mechanisms are (1) the single auction mechanism proposed in Chapter 5 when applied to a set of requests for composite services, one at a time, and (2) a fixed-price mechanism where service requesters fix the price to be paid for a composite service.

The results are presented in section 6.6 in three main parts, along the performance metrics: success rate, cost of composite service procurement and the solve time. For each performance metric, the comparison is performed along two directions: first, the mechanism type (between three mechanisms), and next, between the four market sections. Sensitivity analysis is also performed to examine the impact of simulation parameters on each performance metric. Finally, we performed statistical analysis to determine whether the results are statistically significant or not.

6.2 A Market for Web Services

A market is generally defined as the physical or virtual meeting point where buyers and sellers set the prices and exchange products or services. One of the main functions of a market is to create opportunities for matching of buyers and sellers that provides the following benefits (Bakos 1998):

- 1. The reduction in the cost of search to find potential buyers or sellers
- 2. The reduction in the cost of collecting information on demand and supply

- The reduction in the complexity of price determination of the products or services with the help of the market feedback after participating in a sufficient number of trades
- 4. Depending on the type of items or services to be exchanged, the reduction in the cost of physically transferring the product from sellers to buyers.

While the reduction in the cost of transferring the product is only relevant to physical markets, electronic markets are more significant in terms of reducing the consumer's search cost in obtaining information about prices and product offerings, and the supplier's cost in communicating information about their offerings (Bakos 1991). The reduction in these costs has led to emergence of new opportunities in electronic markets such as aggregation of services and products that traditionally are provided by separate industries (Bakos 1998).

Since the emergence of web services' technologies, the service-oriented computing community have been interested in establishing markets for web services. A web services' market is an electronic online market that offers all the above-mentioned benefits except for the last one. In this market, web services' buyers and sellers can meet and conduct business electronically (Papazoglou 2003). It also fosters opportunities for aggregation of web services' supply and demand by offering added-value composite services and grouping buying power (Papazoglou 2003).

Exiting web service directories, where information about web services is published on the Internet, provide the meeting point for web service sellers and buyers to some extent. However, they provide no further support for the actual exchange to happen. To provide service requesters and providers with the possibility of exchanging composite services in a market, two aspects need to be considered by the designers of such a market:

- Offer and request specification for composite services: the market needs to support service providers in offering bundles of services, and support service requesters in asking for composite services and their constraints and preferences for the composite service,
- 2. Allocation mechanism: the market needs to support the offering of allocation mechanisms that can perform composite service selection for multiple requests.

In the previous chapter, we introduced a composite service selection mechanism based on combinatorial auctions with the focus being on finding providers for a single composite service. Moving toward the vision of a marketplace for web services, in this chapter we investigate the problem of composite service selection for multiple requests or "*multiple composite service selection*".

6.3 The Design of a Marketplace

The market of web services includes three main parties: the market maker, service requesters and service providers. The market maker is an independent party who creates and maintains the market and ensures that the market is open for business (Papazoglou 2003). The market maker keeps a directory of the tasks which describe the functionality of the web services exchanged in the marketplace: service providers offer their services to execute these tasks and service requesters require web services to execute these tasks to achieve their goal. The market may be a dedicated market, focusing on a specific domain such as geographic information system (GIS)-related services, or can be designed to be more generic.

We have assumed that a composite service is defined at a high level as an abstract business process (BP) which comprises a set of tasks. For simplicity, we have assumed that the BP only includes sequential structures. The existence of other structures (parallel, loop and conditional) in the BP only affects our model in terms of the aggregation functions for quality attributes such as response time and availability. As we have already discussed in subsection 3.3, the problem of mapping the BP's complex structure to an execution path has already been extensively discussed in the literature. Ultimately, the proposed techniques which require linear aggregation functions for quality attributes create a sequential execution path. With such an approach, our assumption of having a sequential BP does not limit the proposed model in terms of covering different structures.

Service requesters enter the market and register the required composite (or single) services, that is, *the requests*. The specification of each request includes all the tasks in its business process, the sequence of the tasks in the business process, the local (for individual services) and the end-to-end (for the composite service) QoS constraints and the budget constraint.

Service providers also register their offered services in the market, that is, *the bids* or *offers*. They can register single services as well as bundles of services. Similar to the requests, the offers are multi-dimensional, including information about the quality and the price of the offered services. If the offer includes a bundle of services, the QoS profile of each service is presented separately, while the price is specified as an aggregated value requested for the bundle as a whole.

When a new provider or requester enters the market, they initially need to check the directory to decide whether or not to attend that market. For example, if a service requester finds no commonality between the tasks in the market directory and the tasks needed in their request, they might not find it useful to attend that market. Nevertheless, providers and requesters can register new tasks in the directory and then submit their offers or requests to the market maker. Many requests for a specific new task signals the web service providers the market's need for new services.

The market maker receives offers and requests and matches the offers to requests at specific intervals. The intervals can be based on time limits or the number of active participants in the market. In other words, the matching can be performed if the number of offers and requests get to a specific number, even if the time limit has not yet been reached. After each matching round, the market maker notifies the successful providers and requesters and charges them a fee based on the market's business model. For example, the successful participants' payments to the market maker can be based on the number of transactions, or the final price of the composite service. The results are also announced to all participants to improve their future decision making. For example, the unsuccessful participants might decide to modify their offers and requests based on the market feedback or decide to continue with their current setting for more rounds.

The market maker uses the procurement combinatorial auction model as the basis for the allocation mechanism. As the allocation mechanism considers multiple composite service requests at the same time, we refer to it as the "simultaneous auction" mechanism to distinguish it from "the single auction" mechanism introduced in Chapter 5 that solves the service selection problem for a single request.

The simultaneous auction mechanism matches the offers and requests that exist in the market, based on the defined objective function. As discussed in subsection 3.8.3, currently two alternative objectives of procurement auctions for web services have been

considered in the literature: (1) maximizing the overall utility for the service requester in terms of quality and price of the composite service, and (2) minimizing the cost of service procurement subject to quality constraints.

The market maker decides what objective to implement based on the requirements of that market's participants. If requesters have a clear understanding of the trade-off between different quality attributes and it is important for them to maximize the achieved quality, maximizing the utility is a more appealing objective function and attracts more requesters to the market. In other cases, the objective to minimize the cost can engage more requesters as it requires less complex specification of the service request (there is no need to define weights for quality attributes). Moreover, in a market with this objective, even requesters with a focus on quality can define tighter constraints to achieve their desired level of quality. In the following formulation, we have focused only on the cost-minimizing objective function.

6.3.1 Specification of Offers and Requests

Let T denote the set of all the tasks registered in the market's directory where t is a member of the set T and M is the total number of tasks in the market.

Let *B* denote the set of all received offers (bids) from all web service providers, with a member of the set defined as *b* and *N* is the total number of all received bids. Each bid *b* is defined as $b = (T_b, Q_b, c_b)$, where $(T_i \subseteq Task)$ is the set of tasks that this bid is offering web services that can execute them, Q_b is the set of QoS profiles of these web services and c_b is the provider's requested price to execute these services.

While c_b is the price requested for the whole bundle, Q_b is a set including all the QoS profiles of the services offered in *b*, that is, $Q_b = \{P_{bt} \mid t \in T_b\}$ where P_{bt} is the QoS profile of one service.

Let *L* denote the number of quality of service attributes in a QoS profile of a typical service in this market, with *l* being the *l*-th quality attribute in the profile. Then, P_{bt} is defined as $P_{bt} = \{q_{lbt} | l \in L\}$ where q_{lbt} is the offered value for the *l*-th quality attribute of the service in *b*, executing *t*.

As an example, consider the bid *b* illustrated in Fig 6.1. This bid includes services for three tasks: (1) checking the syntax of an email address, (2) sending an email to a specified email address, and (3) preparing an invoice for sending the email. The QoS profile includes two quality attributes, $L = \{v, x\}$, where *v* represents the availability and *x* represents the response time. The price requested for this bundle is \$5.



Fig 6.1. Example of a bid for a bundle of services

The three tasks are indexed based on the market directory as t_1 , t_2 and t_{10} . Then we have:

$$b = (T_b: \{t_1, t_2, t_{10}\}, Q_b: \{P_{bt_1}, P_{bt_2}, P_{bt_{10}}\}, c_b: \$5),$$

$$P_{bt_1} = \{q_{xbt_1}: 2 \text{ sec }, q_{vbt_1}: 99\%\},$$

$$P_{bt_2} = \{q_{xbt_2}: 10 \text{ sec }, q_{vbt_2}: 95\%\},$$

$$P_{bt_{10}} = \{q_{xbt_{10}}: 10 \text{ msec }, q_{vbt_{10}}: 99\%\}.$$

Let *R* be the set of all received requests for composite services from all service requesters, with a member of the set denoted as *r* and *W* is the total number of all received requests. Each request *r* is defined as $r = (T_r, P_r, B_r)$, where $T_r \subseteq T$ is the set of tasks requested in *r*, P_r is the QoS profile required for the end-to-end quality of the requested composite service, and B_r is the budget constraint to procure services for *r*.

The QoS profile P_r is a tuple including the quality attributes' values requested for r, that is, $P_r = \{ q_{lr} \mid l \in L \}$, where q_{lr} is the minimum or maximum (depending on the type of quality attribute) acceptable value for the *l*-th end-to-end quality attribute of the composite service, r.

As an example, consider the composite service request r as illustrated in Fig 6.2. This request needs services to execute five tasks. The end-to-end QoS requirements of this request specify that the service availability should be over 90% and the response time should not exceed 25 msec. The budget available to procure all the required web services is \$15.



Fig 6.2. Example of a request for a composite service

The required tasks are indexed based on the market directory to be t_1 , t_2 , t_8 , t_9 and t_{10} . Then we have:

$$r = (T_r: \{t_1, t_2, t_8, t_9, t_{10}\}, P_r: \{q_{xr}: 25 \text{ msec}, q_{vr}: 90\%\}, B_r: \$15)$$

6.3.2 Full-Matching of Requests and Offers

In this section, we explain the first simultaneous auction mechanism proposed for the multiple composite service selection problem; called the *Full-Matching* mechanism. This allocation mechanism targets all the requests for composite services and tries to find their best matching offers. With multiple requests in the market, the objective needs to minimize the cost for all requests collectively. This means that each service requester might not get the lowest price compared to what they could get if they could have an auction with all the bids just for themselves. However, the prices are Pareto optimal for the service requesters jointly: no individual requester can achieve a lower price without making another requester worse off.

The objective function is defined in function (6-1) as to minimize the cost for all the requests. The decision variable is denoted by z_{br} ($b \in B, r \in R$) to be 1 if offer *b* is selected for request *r* and 0 otherwise.

Constraint (6-2) ensures that for each task in each request, there is exactly one winning bid. In this constraint, a_{bt} is an arbitrary member of the matrix $A_{N\times M}$. This matrix provides the mapping of the bids to the tasks in the market, where a_{bt} is 1 if T_b includes task *t* and 0 otherwise. The budget constraint for each service request is specified in constraint (6-3). It states that the total cost of procuring a composite service should not exceed the requester's specified budget.

The constraints over other quality attributes are presented in a general form in (6-4) and (6-5), depending on the type of quality attribute. We define a quality attribute to be a "positive" attribute if higher values are more desirable for it, including attributes such as availability and reputation. Similarly, a "negative" quality attribute is one for which lower values are more desirable, such as response time and recovery from failure time. Constraint (6-4) is used for positive quality attributes by ensuring that the minimum desirable value of that quality attribute is met. Constraint (6-5) is applied for negative quality attributes by setting a maximum acceptable value.

In these constraints, each quality attribute l has a specific aggregation function, G_S^l , that calculates the aggregation of the quality attributes' values of the services executing the set of tasks T_r . The aggregated value of the *l*-th quality attribute value needs to be larger or smaller than the requested end-to-end quality attribute's value for the request *r*. The aggregation function for quality attributes can be similar to the ones presented in the literature, as discussed in subsection 3.3. Examples of aggregation functions for availability and response time were defined in subsection 5.3.2.

An interesting difference between the formulation of budget constraint and of other quality attributes is that the former needs to be aggregated only over the set of winning bids while the latter needs two nested aggregations: over the set of winning bids as well as over the set of tasks of a request. The reason is that in the specification of an offer for a bundle of services, the price of the bundle is defined as a single value for the whole bundle, whereas the other quality attributes are defined separately for each service in the bundle. As there will be only one winning bid for each request's task, constraint (6-2), the aggregation function over the set of bids, is a simple sum. The overall end-to-end aggregation of each quality attribute depends on the specific aggregation function of that quality attribute, defined by the function G^{l} . The function G^{l} takes the quality attribute

values of all the tasks in a request (grouped by the union operator) and calculates the endto-end value of that quality attribute.

$$\sum_{r \in R} \sum_{b \in B} c_b * z_{br} \tag{6-1}$$

Subject to:

Minimize

Full allocation
$$\forall r \in R,$$

constraint $\sum_{b \in B} a_{bt} * z_{br} = 1$ (6-2)

Budget constraint $\forall r \in R$ $\sum_{b \in B} c_b * z_{br} \le B_r$ (6-3)

Quality
Quality
constraints

$$\begin{array}{l} \forall r \in R, \\ \forall l \text{ which is a} \\ \text{positive quality} \\ \text{attribute} \end{array} \quad G^{l} \left(\bigcup_{t \in T_{r}} \sum_{b \in B} (q_{lbt} * a_{bt} * z_{br}) \right) \geq q_{lr} \quad (6-4)$$

 $\forall r \in R$, $\forall l$ which is a negative quality attribute

$$G^{l}\left(\bigcup_{t\in T_{r}}\sum_{b\in B}(q_{lbt}*a_{bt}*z_{br})\right) \leq q_{lr}$$
(6-5)

The bidding language presented by this formulation is the OR* language, which has the expressive power of both OR and XOR languages: on one hand, each provider can have any number of winning bids as in an OR language, on the other hand, as each task of a BP should have exactly one service executing it, providers can simulate the XOR combinations by adding a common dummy task to the bids that need to be XOR with each other. However, to have the OR* language, the market maker needs to support the introduction and registration of dummy tasks in the market directory.

6.3.3 Partial-Matching of Requests and Offers

The main limitation of the Full-Matching mechanism is that even if one request is not feasible due to budget or quality constraints, the whole auction fails and the mechanism fails to match any of the requests. To overcome this limitation, we propose the Partial-Matching mechanism which relaxes the Full-Matching mechanism requirement to find providers for all the requests. The Partial-Matching mechanism aims to find the optimal set of providers for as many requests as possible. To achieve this, we need to change both the objective function (6-1) and the allocation constraint (6-2).

We defined a new objective function, equation (6-6), which consists of two parts (objectives). The first objective is to minimize the cost for all the requests (the same as before) and the second objective aims to maximize the number of feasible requests. The feasibility of a request is indicated by a new decision variable, y_r which is 1 if the request r is feasible and 0 otherwise.

To construct one linear objective function from the two parts, we have used the Big-M method (Padberg 1999, p.54). This method is based on using a sufficiently large number, M, in linear optimization problems. In our case, applying the Big-M technique to our minimization problem leads the optimization software (solver) to first maximize the part which includes the negative M, and then minimize the other part of the objective function.

The relaxation of the allocation constraint (6-2) is achieved by defining a set of new constraints, (6-7) to (6-11), and a new decision variable, x_{tr} , associated to each task of each request. The decision variable x_{tr} is equal to 1 if task t in request r is feasible and 0 otherwise. A request's task is called "feasible" if a bid is selected to provision it. Based on this definition, if the request r does not contain task t, then x_{tr} needs to be 0, which is forced by constraint (6-7).

The set of constraints, (6-8) to (6-11), defines the relationship of a feasible request, a feasible request's task and a winning bid. Constraint (6-8) states that if a request is feasible, then "all" of its tasks should feasible. In this constraint, D is a $M \times W$ matrix that defines the mapping of the tasks in the market to the requests, with an element d_{tr} that is equal to 1 if r includes t and 0 otherwise. Using this matrix, the set of tasks in a request r are defined as $T_r = \{t \mid b_{tr} = 1\}$. Constraint (6-9) indicates that if a request is not feasible, "none" of its tasks should be feasible. Moreover, a request being infeasible also means that no bids should be assigned to it, specified by constraint (6-10). Constraint (6-11) ensures that if a task in a request is feasible, then there is one winning bid containing that request's task.

Constraints (6-12), (6-13) and (6-14) present the budget constraint and the quality requirements for positive and negative quality attributes for the feasible requests. The

notations presented above to formulate the Full-Matching and Partial-Matching mechanisms are summarized in Table 6.1.

Minimize

 $\forall r$

$$\sum_{r \in R} \sum_{b \in B} c_b * z_{br} - M * \sum_{r \in R} y_r \tag{6-6}$$

Subject to: $\forall r \in R$ if $y_r = 1$ then $\sum_{t \in T_r} x_{tr} * d_{tr} = \sum_{t \in T_r} d_{tr}$ (6-8)

$$\forall r \in R \qquad if \ y_r = 0 \ then \qquad \sum_{t \in T_r} x_{tr} * d_{tr} = 0 \tag{6-9}$$

$$\in R \qquad if \ y_r = 0 \ then \qquad \qquad \sum_{b \in B} z_{br} = 0 \tag{6-10}$$

Budget
$$\forall r \in R$$
 if $y_r = 1$ then $\sum_{b \in B} c_b * z_{br} \le B_r$ (6-12)

Positive
quality
constraints
$$\forall r \in R$$
 if $y_r = 1$ then $G^l \left(\bigcup_{t \in T_r} \sum_{b \in B} (q_{lbt} * a_{bt} * z_{br}) \right) \ge q_{lr}$ (6-13)

Negative
quality
constraints
$$\forall r \in R$$
 if $y_r = 1$ then $G^l \left(\bigcup_{t \in T_r} \sum_{b \in B} (q_{lbt} * a_{bt} * z_{br}) \right) \le q_{lr}$ (6-14)

In the above formulation, we had used "if ... then" statements which are not linear. To map the problem to an Integer Linear Programming (ILP) problem, we need to linearize these constraints by removing the "if ... then" statements.

Revising the non-linear constraints (6-9) and (6-10), we can see that they collectively lead to the conclusion that $\forall r \in R, \forall t \in T_r, x_{tr} = y_r$. The linear form is presented in constraint (6-15). The linearized version of constraints (6-10) and (6-11) are presented below in constraints (6-16) and (6-17), respectively. The linearization of the constraints (6-12), (6-13) and (6-14) for budget and quality attributes are presented in constraints (6-18), (6-19) and (6-20), respectively:

$$\forall r \in R, \\ \forall t \in T_r \qquad \qquad \sum_{b \in B} a_{bt} * z_{br} = x_{tr}$$
 (6-17)

$$\forall r \in R, \qquad \qquad B_r - \sum_{b \in B} c_b * z_{br} \ge y_r - 1 \qquad (6-18)$$

$$\forall r \in R, \\ l: a \text{ positive } \\ quality attribute \qquad \qquad G^l \left(\bigcup_{t \in T_r} \sum_{b \in B} (q_{lbt} * a_{bt} * z_{br}) \right) \ge q_{lr} * y_r$$
 (6-19)

$$\forall r \in R, \\ l: a negative \\ quality attribute \qquad \qquad G^l \left(\bigcup_{t \in T_r} \sum_{b \in B} (q_{lbt} * a_{bt} * z_{br}) \right) \le q_{lr} * y_r$$
 (6-20)

| Notation | Definition |
|------------------|---|
| Т | the set of all tasks registered in the market's registry |
| М | the total number of tasks in the market |
| t | an element of the set <i>T</i> |
| L | the set of quality attributes in a QoS profile |
| l | an element of the set L |
| В | the set of all received offers (bids) |
| Ν | the total number of all received bids |
| b | an element of the set <i>B</i> |
| T_b | the set of tasks that <i>b</i> offers web services to execute $(T_b \subseteq T)$ |
| c _b | the cost of providing service(s) for the task(s) in T_b |
| Q_b | the set of the offered QoS profiles for the services executing T_b |
| P_{bt} | the QoS profile of the service executing task t in b b _i = (T _i , c _i , Q _i) |
| q_{lbt} | the value of l -th quality attribute function offered for the task t in bid b |
| R | the set of all received requests from all service requesters |
| W | the total number of all received requests |
| r | an element of the set <i>R</i> |
| T_r | the set of tasks requested in $r(T_r \subseteq T)$ |
| Br | the budget constraint to procure T_r |
| P_r | the QoS profile requested for the end-to-end quality of the composite service |
| | specified by request r |
| q_{lr} | the value of the l -th quality attribute, requested for r |
| Z _{br} | the decision variable to be 1 if offer b is selected for request r |
| $A_{N \times M}$ | the matrix mapping the bids to the tasks in the market |
| a_{bt} | an element of the matrix $A_{N \times M}$ |
| y_r | the decision variable to be 1 if the request r is feasible, 0 otherwise |
| x _{tr} | the decision variable to be 1 if t in r finds a provider, 0 otherwise |
| $B_{M \times W}$ | the matrix, mapping the set of tasks in the market to the requests |
| b _{tr} | an element of matrix $B_{M \times W}$ |

Table 6.1. Notation used for the formulation of Full-Matching and Partial-Matching mechanisms

6.3.4 Providers' Resource Limitation

In the formulation presented so far to respond to the multiple composite service selection problem, each bid can be selected for multiple requests. This means that this formulation does not impose any restrictions on the providers' capacity to offer their services. Such a formulation can be useful in markets where providers claim unlimited resources.

However, when providers have limit resources, we need to add an appropriate constraint to represent resource limitation. The simplest case would be to limit each bid to be assigned to at the most one request, as depicted in constraint (6-21). If a provider is willing to offer a bid more than once, they can simply replicate the bid with different identifications. Alternatively, it is possible to define different resource limitation

constraints for different providers, changing "one" at the right-hand side of the constraint to the number of times a bid can be selected:

Resource limitation constraint
$$\forall b \in B, \sum_{r \in R} z_{br} \le 1$$
 (6-21)

6.4 Experiment Design

To evaluate the proposed simultaneous auction mechanism, we have conducted extensive experiments through simulation. The objective of the experiments is to compare the performance of the proposed Partial-Matching simultaneous auction mechanism with appropriate baseline mechanism(s).²¹ To achieve this, we need to define the performance metrics and the baseline for comparison.

The design of the experiment is in line with our research methodology, previously explained in Chapter 4. Based on this methodology, the evaluation process consists of the following main elements:

- Define the performance metrics to be measured,
- Establish a baseline,
- Identify the scenarios to be investigated, that is, the market sections to be studied,
- Design the stochastic model to generate the input data,
- Determine the seeding for the simulation.

6.4.1 Performance Metrics

We have identified a set of important performance criteria to assess the simultaneous auction mechanism based on the literature for composite service selection and auction theory. The performance metrics comprise:²²

²¹ From this point, whenever we refer to the simultaneous auction mechanism, it is in fact the Partial-Matching formulation.

²² Considering the objective function which is to minimize the procurement cost, we did not find it useful to define performance metrics to measure utility. Therefore, the design of the experiment is for the setting where offers and requests include a monetary element (cost or budget), and there is no need to other quality attributes.

- 1. Success Rate (SR): the number of feasible requests over the total number of requests in each simulation round (with specific number of bids and requests),
- 2. Cost per composite service (CPC): the average cost of procuring a composite service in each simulation round,
- 3. Solve time: the time taken by the optimization software to find the set of successful requests in each simulation round.

A simulation round starts with a specific number of requests and offers to be matched. The simultaneous auction performs service selection for all the requests in one matching round. Therefore, for the simultaneous auction the simulation round is the same as the matching round. However, the single auction and the fixed-price mechanisms perform the matching of the requests, one at a time. This means that a simulation round for these two mechanisms includes multiple matching rounds, to be exact, equal to the number of requests in that simulation round.

6.4.2 Establish a Baseline

The proposed simultaneous auction mechanism solves the composite service selection problem by considering all existing requests at the same time. We compared this mechanism with two other mechanisms which solve the service selection problem for multiple requests, one at a time: (1) *the single auction* mechanism introduced in Chapter 5 when applied to more than one request, and (2) *a fixed-price* mechanism where the service requester fixes the price to be paid for a composite service.

While the simultaneous auction demonstrates a long-term strategy for service allocation (by waiting for a specific number of requests to arrive the market or the time to reach to a specific interval), the baseline mechanisms have a greedy short-term strategy by solving the problem for each request.

6.4.2.1 Single Auction Mechanism

We compare the performance of the proposed simultaneous auction mechanism to that of the single auction mechanism discussed in Chapter 5. The single auction solves the composite service selection problem for a single request without considering others. The comparison between the two mechanisms is in fact comparing the short-term greedy strategy of the single auction with the long-term strategy of the simultaneous auction mechanism. This comparison is interesting as all other current approaches to solve the composite service selection problem, discussed in Chapter 3, have considered one request only. They can only be extended to a setting with multiple requests by solving the problem for each request separately. We expect that the long-term liberal strategy of the simultaneous auction achieves better performance compared to the short-term selfish strategy of the single auction.

In our experiment design, the single auction mechanism solves the multiple composite service selection problem by considering the requests one by one, based on the first-come, first-served (FCFS) policy. For each request, the auction aims to find the lowest procurement cost while satisfying all other allocation constraints. If the auction is successful, the winning bids are removed from the pool of offers and the auction starts over for the next request with the remaining available offers.

6.4.2.2 Fixed-price Mechanism

Auctions and their dynamic pricing strategy are seen as efficient alternatives to the fixed pricing policy, in electronic markets on the Internet in general (Strauss et al. 2009; Lee and Szymanski 2007; Wurman 2001) and in many domains such as cloud computing in particular (Zaman and Grosu 2013; Mihailescu and Teo 2010). Therefore, in auction literature, a fixed-price mechanism is usually used as the baseline against an auction model which allows dynamic pricing.

The fixed pricing policy can be implemented by either the buyer or the seller depending on who is setting up the trade. In settings where there are items for sale, a seller with the fixed pricing strategy determines a static price for the items and the buyers can take it or leave it. While, in a procurement setting where there is a buyer with a fixed pricing policy, they fix the price to be paid for obtaining the products or services and the buyers can either take the offer or leave it. In both these settings, the trader with the fixed pricing policy does not have the possibility of choosing among all the offers. Rather, the fixedprice mechanism aims to find the first business partner who agrees to trade at the predetermined price.

As we model the composite service selection as a reverse or procurement auction, in the baseline fixed-price mechanism, it is the service requester (buyer) who declares a fixed price that they are willing to pay for the composite service. This price is also known as

the *willingness to pay*. For each request, the fixed-price mechanism looks for the first feasible solution (set of service offers) which its total price is below the declared willingness to pay of the requester and satisfies all the other allocation constraints.

The fixed-price mechanism considers requests and offers based on the first-come, firstserved (FCFS) policy. After a successful allocation of web services to a request, the set of allocated services are removed from the pool of available offers and the search continues for the next request with the remaining services.

6.4.3 Scenarios to Investigate: Market Sections

Based on the review of the literature on the markets for web services, we decided to perform our experiment in four important market sections for web services, rather than in a general market. These sections are formed based on two factors, as depicted in Table 6.2:

- 1. The market economy size which categorizes the markets into small economy and large economy, adopted from Tang (2004).
- The composite service complexity which divides the markets into markets for simple composite services and markets for complex composite services, based on Weinhardt et al. (2011).

| Composite Service Complexity | SIMPLE | COMPLEX |
|---------------------------------|--------------|---------------|
| Economy Size | | |
| SMALL | small-simple | small-complex |
| LARGE | large-simple | large-complex |

 Table 6.2. The four market sections in the experiments

6.4.3.1 Economy Size

The economy size of a market is determined by the number of participants in that market which, in our case, consists of the number of service providers and the number of service requesters. Clearly, the more participants attending the market, the larger the economy size of that market is. The economy size of a market for a particular domain of web services is affected by the maturity of the web service users in that domain. In newly formed communities or the ones with less maturity regarding the web service's technologies, the number of service offers and requests are limited. In more mature communities, the number and variety of the available services and requests for single and composite service are considerably larger.

A more mature community benefits from having more participants: with the number of providers increasing, the number and variety of single and bundled services expand and the competition to offer better quality with lower prices increases. This encourages more requesters to attend the market, which in turn, attracts a larger number of providers and persuades them to develop new web services for the market.

6.4.3.2 The Complexity of Composite Services

In Weinhardt et al. (2011a, p.31), two factors are suggested as the basis for dividing the web services' market into four sections (Fig 6.3):

- The degree of cross-organizational interaction which divides the interaction into either no interaction (one provider offers one or more services) or existence of interaction (multiple providers offer aggregated services),
- The degree of composition complexity which divides services into single (or, more accurately, simple) and complex services.

Regarding the first aspect on having one provider or multiple providers offering aggregated services, composite web service selection mostly involves more than one provider in the provisioning of a composite service. As web services offer simple atomic functionalities, it is more likely to procure different services from different providers especially if the required composite service is rather large and complex. Therefore, in our study we are not dealing with market sections where one provider offers all the aggregated services.



Fig 6.3. Web services' market sections (Weinhardt et al. 2011a)

Regarding the second aspect on simple or complex services, composite web services can range from being simple to complex. The two extremes, simple and complex, are motivated by the current literature focusing on two application domains for composite services:

- 1. Mobile computing (Lamparter 2007)
- Scientific workflows (Tan and Zhou 2013; Néron et al. 2009; Ludäscher et al. 2006; Oinn et al. 2004)

In the following section, a description of each application domain is provided and a motivating scenario for that domain is presented.

6.4.3.2.1 Mobile Applications

With recent advances in technologies for mobile devices, such as increased computational power and increased Internet bandwidth, it has become possible for mobile users to carry out more sophisticated operations. However, due to the limited resources of mobile devices, parts of these operations are performed on remote computers with the help of web services. In mobile applications, the composite services are usually not very complex and large due the resource limitations of mobile devices which prevent their users from defining very complex queries and workflows. In this environment, the service users are generally individual end-users. They usually do not need mission critical services (availability), but response time can be very important.

Motivating Scenario in Mobile Applications: Mia is a mobile user in Australia. She often forgets to fill up her car's petrol tank. Therefore, she found it helpful to subscribe to a mobile application called FuelBuddy which gives her the location of the cheapest fuel station, closest to her current location. FuelBuddy combines several services to prepare the result for Mia and, in the end, only sends her mobile the route to the first couple of options. FuelBuddy outsources these services from various service providers over the Internet. The business process of this composite service is depicted in Fig 6.4.



Fig 6.4. A mobile computing scenario: FuelBuddy

6.4.3.2.2 Scientific Workflows

Information technology (IT) has revolutionized the way science is conducted in many fields such as bioinformatics, biodiversity, life science and astronomy. In a typical experiment in these areas, many researchers from different research groups are involved, needing to collaborate and experiment on a large amount of distributed data, using distributed resources. Therefore, the complexity of manually conducting experiments has been a challenge for researchers in these fields.

Scientific workflows have emerged to tackle the complexity of conducting scientific experiments. A scientific workflow consists of a set of computational or data manipulation steps to process, transform and carry out the data in a distributed environment (Gil et al. 2007). Each step specifies a computation to be executed by a software program such as a web service or a legacy system. These steps are chained based on the specification of the workflow structure. The requested workflows are usually very

complex and may include tens of tasks. The requesters have constraints regarding the accuracy of the services (certified by third parties), and preferences about the privacy and security of the data.

Motivating Scenario in Scientific Workflows: Protein sequence analysis or protein sequencing refers to the techniques used to determine the protein's amino acid and its conformation. With the help of these techniques, researchers can identify the similarities between novel sequences and well-characterized database sequences to answer questions like: What is the protein under study? To what family does it belong? What is its function? How can we explain its function in structural terms?²³

The analysis of protein sequence data was previously restricted to those with access to mainframes or expensive desktop computer programs. It is a very time-consuming process which can take up to two days. Nowadays, with online tools and services available, these restrictions are largely eliminated. Many web services have been developed to search the current protein databases, and many other web services now exist for performing different analyses of the data.

The workflow illustrated in Fig 6.5 performs a generic protein sequence analysis, adopted from the website, myExperiment.²⁴ This site is a social networking site for scientists where they can share scientific workflows they have created. The workflows can be developed using any of the several tools available such as Taverna²⁵ (Oinn et al. 2004), Kepler²⁶, rapidminer²⁷ and LONI Pipeline.²⁸ Some of these tools, such as Taverna and Kepler, allow researchers to incorporate already existing web services in their workflow model as well as executing the workflow.

²³ <<u>http://www.biochem.ucl.ac.uk/bsm/dbbrowser/jj/info4frm.html></u>

²⁴ <<u>http://www.myexperiment.org/</u>>

²⁵ <<u>http://www.taverna.org.uk/></u>

²⁶ <<u>https://kepler-project.org/></u>

²⁷ <<u>http://rapidminer.com/></u>

²⁸ <<u>http://pipeline.bmap.ucla.edu/></u>



Fig 6.5. Scientific workflows scenario: protein sequence analysis (M.B. Monteiro 2008)

6.4.4 The Stochastic Model to Generate Data

While there are some data for real individual web services' quality attributes such as the Quality of Web Service (QWS) data set by Al-Masri and Mahmoud (2007), there is no public information on web services' pricing or bundling. Therefore, we designed a stochastic data generation model based on the literature of combinatorial auctions. The design of the model includes: firstly, a parametric model to generate data so that it is as

close as possible to real-world web services' offers and requests and, secondly, establishing a careful starting configuration for seeding the parameters used in simulation.

6.4.4.1 Data Generation Model

In the evaluation of the simultaneous auction mechanism, we need to have the data for bids as well as the requests. The data generation model follows three steps to generate the data:

Step 1. Generating the tasks in the market:

The model starts by generating a fixed number of tasks for the market, based on the given value for the simulation parameter **Tasks_Number**. This is the set of tasks that are mostly relevant to a particular business domain, and therefore, there is a community of service providers and requesters interested in these tasks.

Step 2. Generating the set of service providers:

In the second step, a fixed number of service providers will be created, based on the given value for the simulation parameter **Providers_Number**. Each provider will have an independent valuation for each task, drawn from a uniform distribution between [1,100]. This is referred to as Independent Private Valuation (IPV) model in auction theory (Kagel and Levin 1993) and is similar to the CATS approach in generating the prices for the items under auction. Participants with an IPV model only know their own valuation and they do not care about others' valuations for the items being auctioned. The IPV model is the common assumption in the auction and mechanism design literature.²⁹

Subsequently, each provider will be assigned a discount factor which is drawn from a uniform distribution between [0, Max_Discount], where Max_Discount is the maximum discount given by providers in that market.

²⁹ As discussed in subsection 5.3.2.2 (stage 2), another valuation function in auction theory is called "the common value" where the value of offering the service for a task is more or less the same for all providers, but each provider's estimate for how much they can charge to sell the service in the market is different. Although this approach seems to be more realistic than the IPV, it is not used very often in research and experiments. One reason might be the complexity of analysis related to games based on this model. For example, the winners of a game with this valuation model will always suffer from the winner's curse. Moreover, in the previous chapter, the results of the experiments with the two valuation functions did not lead to significant difference in the cost or success rate. Therefore, we decided to base the simulation for the simultaneous auction on the more commonly used independent private valuation (IPV) model.

Step 3. Generating the offers (bids):

In the third step, a fixed number of combinatorial offers (bids) is generated for one simulation round based on the given value for the simulation parameter **Offers_Number**. Offers are then randomly assigned to providers. This means that each provider might have one or more bids active in the market. Each bid has the following elements to be considered during bid generation:

- 1. Number of services to be offered in the bid
- 2. The services to be offered
- 3. The price for the bundle.

To decide the number and the services in each bid, we have two options:

- Uniform distribution: *m*, the number of services in each bid, is drawn from a uniform distribution between [1, M], where M is the number of tasks in the market. Then, choose *m* random tasks from the set of M tasks, and add the equivalent services to the bid.
- 2. Decay distribution: Consider a new parameter, $\alpha < 100$, which determines the bid (bundle)'s crowdedness. α can be set at different values for different providers or can have the same value across all providers in the market. Randomly, choose the first service to be included in the bid. Then, draw a random number between [0,100]. If this number is smaller than α , choose another service randomly to be added to the bundle and start over, otherwise stop.

We chose the decay distribution option as it is said to generate harder instances of combinatorial bids for solvers (Sandholm et al. 2002). To set α , we initially experimented with setting a different α for different providers. As the average α across all providers had been close to 50%, the average size of the bids would be less than having three services. This would create simple combinatorial bid instances. To have more variety of simple and complex bids, we decided to keep α fixed for the whole market on 75%.

The price requested for the services offered in a bid is calculated as the sum of the provider's valuations for the services in the bid, minus a discount which is calculated based on the provider's discount factor.

Step 4. Generating the requests for composite services

In the final step, a fixed number of requests is generated in the market based on the given value for the simulation parameter **Requests_Number**. To specify a composite service request, we needed similar elements to those of a bid:

- 1. The size of the composite service (CS)
- 2. The tasks requested in the CS
- 3. The requester's budget constraint to procure the CS.

Based on the market sections considered, the composite services are generated in two sizes: small for a market with simple CSs or large for a market with complex CSs. The tasks in a request are randomly chosen from the set of tasks in the market. To calculate the budget constraint, firstly, the requester's valuation for each task is generated based on the IPV model. Then, the budget is calculated as the sum of the requester's valuations for the tasks in that request.

Note that, as the data generation model does not guarantee the existence of at least one bid for each task in the market directory, there might be some service requests including tasks for which no service is being offered. Such a scenario is very likely to happen in a real-world web service market. This means that requesters are interested in a specific service that is not currently offered and it signals the providers the need for developing such a service. However, apparently all the requests including such tasks would be infeasible, regardless of the allocation mechanism.

6.4.4.2 Seeding the Parameters

Due to the absence of publicly available real data for web services' offers and requests, and lack of similar experiments on web services' markets in the current literature, seeding of the simulation parameters was a challenging problem for this study. The main difficulty was related to visualizing a marketplace for web services: What would be the likely number of service bids or requests in such a market? Or what would be the range of numbers? What is the likely size of composite services requested in that marketplace?

To answer these questions, we referred to the existing communities around web services on the Internet. These communities currently do not have the functionalities of a market, that is, price discovery or facilitating the exchange. Nevertheless, they can be a representation of marketplaces in the future.

We believe that future markets for web services will focus on specific domains, rather than being a generic marketplace. Similar to the existing communities of web services around different domains, there will be markets focusing on different areas, such as GISrelated web services to retrieve, store, display and analyze GIS-enabled information. General web services might attend domain-specific markets as well as more generic ones.

Our belief is supported by the studies that we performed over some of the real-world examples of web services' markets. Companies interested in providing a marketplace for web services have moved toward offering more specialized web services in a specific domain, rather than offering an environment for exchanging all sorts of web services. For example, StrikeIron³⁰ and Xignite³¹ are two start-up companies which initially appeared as general web services' markets (Blau et al. 2010). However, they later focused on offering services in specific domains: data quality validation and verification web services in the case of StrikeIron, and financial services in the case of Xignite. This trend pictures the web services' markets.

In our studies, we reviewed active communities of web service providers and requesters focusing on different domains, including but not limited to: biodiversity research, GIS-related web services, life science research, astronomy and eLearning web services. The website, ProgrammableWeb, also provides information on web services and their users, categorized in more than 60 domains.

Our search showed that, depending on the maturity level of the community of web service users and providers, they form small or large communities (the economy size of the communities). Moreover, the services offered in these communities can be composed together to create composite services ranging from simple to complex. Simple composite services do not include many single web services (less than five), while complex composite services may incorporate more than 20 services to achieve their goal.

³⁰ <<u>http://www.strikeiron.com</u>>, active since 2002

³¹ <<u>http://splice.xignite.com/</u>>, active since 2003

| Parameter | Values of the parameter for | Levels |
|------------------------------|--|--------|
| Tasks_Number (in the market) | [25,50] | 2 |
| Providers_Number | [5,50] | 2 |
| Offers_Number (Bids_Number) | In small economy: [50,100] In large economy: [300,500] | 4 |
| α in the market | 75% | 1 |
| Max_Discount | 25% | 1 |
| Requests_Number | In small economy: [4,12] In large economy: [20,28] | 4 |
| Requests_Size | Small, #tasks between [3,4,5] Large, #tasks between [15,16,,20] | 2 |
| Total combinations | | 128 |

Table 6.3. Seeding the experiment's input parameters

For seeding the parameters, we relied on these observations to gain insight on the number of service providers, service offers, service requesters and size of requests for composite services. Based on the market sections discussed before, Table 6.2, we focused the experiments on two extremes of composite service complexity: marketplaces with simple requests (such as mobile applications), and marketplaces with complex ones (such as scientific workflows). The final seeding of the simulation parameters is depicted in Table 6.3.

6.5 Experiment Execution

We proposed two variations of the simultaneous auction, the Full-Matching and the Partial-Matching mechanisms. We performed the evaluation on the Partial-Matching mechanism as it can achieve higher success rate compared to the Full-Matching mechanism. In terms of the cost, both mechanisms would achieve the same average cost of procurement for problem instances where all the requests are successful in procuring web services. In other instances, where not all requests are successful, the Full-Matching mechanism would fail and the cost is zero.

The formulation of the simultaneous auction presented in section 6.3.3 restricts the number of service providers selected for each task in a request to be exactly one. With some consideration, this constraint can be relaxed to improve the success rate of the

mechanism and achieves lower procurement costs for composite services.³² The relaxation would allow each task to have "at least" one winning bid, rather than "exactly" one.

We argue that as the mechanism objective function is defined to minimize the cost, there is no need to restrict the winning bids for each task. In other words, there is no problem if more than one service is selected for a task if such an allocation leads to a less expensive provisioning of the composite service. However, the requester needs to decide which of the winners will ultimately execute the task. The selection might be random or based on some criteria such as choosing the one who is already providing more (or fewer) services for other tasks in the composition.

The intended relaxation is applied to the allocation constraint (6-17) where the *equals* sign is changed to *bigger than or equal* to relax the exactly one winning provider assumption. The final ILP formulation of the simultaneous auction mechanism to be evaluated is presented here:

$$\begin{array}{ll} \mbox{Minimize} & \sum_{r \in R} \sum_{b \in B} c_b * z_{br} - BIGM * \sum_{r \in R} y_r \\ \mbox{Subject to:} & & \\ & \forall r \in R, \forall t \notin T_r & & x_{tr} = 0 \\ \mbox{Partial} & & \\ \mbox{allocation} & \forall r \in R, \forall t \in T_r & & x_{tr} = y_r \\ \mbox{constraints} & & \\ & \forall r \in R, \forall b \in B & & z_{br} \leq y_r \end{array}$$

³² The relaxation can be applied in a market where the offers and requests include only a monetary element (cost and budget). It cannot be used when other quality of service constraints are required as allowing more than one provider for each task complicates the service selection problem: with the possibility of multiple providers, the requester needs to make a later decision regarding the selection of only one of the providers for actually executing the task, even if the decision is as easy as making a random choice. This means that the mechanism is not aware of the choice of the actual service at the service selection time. Therefore, it cannot determine which of the selected services' quality profiles to consider when evaluating the satisfaction of the end-to-end quality of service constraints. While it is possible to consider an aggregation of the QoS of all the selected services in the formulation of the problem, the quality constraints will be much tighter than what the requester needs. However, as our evaluation process is focused on performance metrics regarding success rate, cost and solve time of the mechanism, the relaxation of the having exclusively one provider for each task will not cause any problem.

$$\begin{array}{ll} \forall r \in R, \forall t \in T_r & \sum_{b \in B} a_{bt} * z_{br} \geq x_{tr} \\ \hline Budget \\ constraint & \forall r \in R, & B_r - \sum_{b \in B} c_b * z_{br} \geq y_r - 1 \\ \hline Resource \\ limitation \\ constraint & \forall b \in B & \sum_{r \in R} z_{br} \leq 1 \end{array}$$

Based on the seeding of the simulation parameters, as presented in Table 6.3, there are 128 combinations of variables' values for evaluation. For each combination, 30 instances are generated to be able to have meaningful statistical analysis. The performance metrics are averaged over the 30 instances.

All three mechanisms (fixed-price, single auction and simultaneous auction) were implemented in the AMPL language. The related ILP problems were solved by a package of AMPL 2014 and CPLEX 12.6 as the solver, using a server computer with 64 AMD Opteron processors each 1400 MHz and a total memory of 132 GB RAM. In the simultaneous auction model, a time-out option of 60 seconds was set for the solver, due to the time complexity of the mechanism. A similar time-out option was set for the solver for solving the two other mechanisms to maintain consistency across the three mechanisms.

Now we can define the objective of the experiments more specifically. The objective of the experiments is to compare the three mechanisms (fixed-price, single auction and simultaneous auction) based on the defined performance metrics (success rate [SR], cost per composite service [CPC] and solve time) in the four market sections, as depicted in Table 6.4. More specifically, the experiment aimed to answer the following questions:

Vertical Comparison: How do the performance metrics of the simultaneous auction mechanism compare to the two other mechanisms?

Horizontal Comparison: In the simultaneous auction mechanism, how are the performance metrics different in a market section with complex requests and a market with simple requests? How do the mechanism's performance metrics differ in a market with large economy compared to a market with a small economy?

Table 6.4. The evaluation space

| Composite Service Selection Mechanism Market Section (Economy size-CS complexity) | Fixed-Pricing | Single Auction | Simultaneous Auction |
|--|---------------|----------------|----------------------|
| Small-Simple | | | |
| Small-Complex | | | |
| Large-Simple | | | |
| Large-Complex | | | |

6.6 Results

The results are presented below based on the performance metrics: SR, CPC and solve time.

6.6.1 Success Rate (SR)

The success rate (SR) demonstrates how successful the mechanism is in finding web services for the requests. It is defined as the number of feasible requests to the total number of requests in each instance of multiple composite service selection problem. We have studied the success rate in two directions: firstly, along the three service selection mechanisms and, secondly, for each mechanism along the four market sections.

6.6.1.1 Comparison between the Mechanisms

The SRs of the three mechanisms are plotted against the four market sections in Fig 6.6. The results show that, among the three mechanisms, the best SR is achieved by "the simultaneous auction" in all four market sections. The superiority of the success rate of the simultaneous auction is due to its long-term allocation strategy versus the greedy short-term strategy of the other two mechanisms. Single auction and fixed-price mechanisms allocate web services to requests one request at a time, based on the order of the requests' arrival to the market. With such a greedy allocation strategy, the requests that arrive later and have tighter budget constraints might not find the required set of bids in the market in the remaining pool of web services' offers. However, the simultaneous auction mechanism has a long-term allocation strategy, and in each simulation round, it considers all requests and their needs simultaneously. Therefore, it has the possibility to

match the requests and offers more effectively and increase the number of successful requests.



Fig 6.6. SR achieved by the three mechanisms in the four market sections

The results also state that the similarly greedy allocation strategies of the single auction and the fixed-price mechanisms have resulted in very close SRs for these two mechanisms.

Fig 6.6 also denotes that the difference between the SR of the simultaneous auction and that of the two other mechanisms is most substantial in the (large-complex) market section with the simultaneous auction's SR being 16% higher than the two other mechanisms. This trend is followed by the (large-simple) section with the simultaneous auction's SR being 15% higher; the (small-complex) section with the SR being 10% higher, and lastly, the (small-simple) section where the SR is 3% higher.

This trend in the difference of the SRs of these three mechanisms suggests that the simultaneous auction's lead in achieving a better SR is more notable in more complicated settings. In other words, the simultaneous auction's long-term allocation strategy demonstrates higher efficiency in more complex settings. In simpler settings, the long-term and short-term strategies might not lead to very different outcomes.

6.6.1.2 Comparison between Market Sections

Studying each mechanism's SR in the four market sections leads to several interesting results. The results show that firstly, all three mechanisms achieve a higher SR with complex requests compared to simple requests, in both economy sizes. For the simultaneous auction mechanism, the difference is 5% in the large economy and 14% in the small economy. For the two other mechanisms, the difference is approximately 4% in the large economy and 7% in the small one.

Secondly, all mechanisms have a higher success rate in the large economy compared to the small economy, regardless of the complexity of the requests. For single auction and fixed-price mechanisms, the SR in the large economy is 47% higher than the small economy for simple requests (69%–22%) and 44% higher for complex ones (73%–29%). The simultaneous auction's SR in the large economy is 59% higher than the small economy for simple requests (84%–25%) and 50% higher for complex requests (89–39%).

In general, the SR of all the mechanisms is nearly over 70% for the large economy, while it is below 40% for the small one. This means that, success rate-wise, in small economies, using a fixed-price mechanism is as good as having a short-term allocation strategy such as the single auction, and marginally worse than setting up a simultaneous auction with a long-term allocation strategy. The low SR might prevent service requesters and providers from attending a service selection mechanism with a business model based on subscription fees, while in a business model based on applying fees to only successful transactions, it might not be as discouraging.

6.6.1.3 Sensitivity Analysis

In this section, the impact of different values of the simulation parameters on the SR of the simultaneous auction mechanism is analyzed. These parameters include the number of tasks, number of providers, number of bids and number of requests.



Fig 6.7. Impact of number of tasks on the simultaneous auction's SR for (a) simple and (b) complex requests

In Fig 6.7, the impact of the number of tasks registered in the market's directory on the SR of the simultaneous auction mechanism is studied for (a) simple requests and (b) complex ones. The diagrams indicate that the simultaneous auction achieves a higher SR in a market with a smaller number of tasks (25 tasks) compared to having a larger number (50 tasks) in its directory regardless of the requests complexity. The reason is that when we generate a fixed number of bids over a larger set of tasks, there will be fewer bids including each task compared to when generating the same number of bids for a smaller set of tasks. Having more bids for each task increases the probability of the tasks in the smaller set to find a web service.

The impact of the number of providers on the simultaneous auction's SR is examined in Fig 6.8 for (a) simple requests and (b) complex ones. Our initial expectation was that increasing the number of providers who are generating a fixed number of bids leads to more variety in the valuations of the tasks, and a subsequently higher variety in the

generated bids' prices. We expected that the higher variation in bids eventually leads to a higher SR and lower procurement cost. However, we can see that this parameter does not have a significant impact on the SR. This might be related to the fact that the number of providers is not directly involved in the ILP formulation of the simultaneous auction mechanism.



Fig 6.8. Impact of number of providers on simultaneous auction's SR for (a) simple and (b) complex requests

With regard to the impact of the number of bids and number of requests on SR, both Fig 6.7 and Fig 6.8 demonstrate that having a higher number of bids increases the SR as a result of the increase in supply, and having a higher number of requests in the market decreases the SR due to the increase in demand.

6.6.1.4 Statistical Analysis

We performed the Kruskal-Wallis test on the success rate (SR) of the three mechanisms to determine if the results are statistically significant. The Kruskal-Wallis test (Kruskal and Wallis 1952) is used to determine whether there are significant differences between

two or more groups of an independent variable on a continuous or ordinal dependent variable. The null hypothesis of the Kruskal-Wallis test is that all groups come from the same distribution against the alternative hypothesis that is there are at least two groups of the dependent variable which are statistically significantly different.

This test is also known as the "one-way ANOVA on ranks" and is the nonparametric alternative to the one-way ANOVA, which unlike ANOVA does not assume a normal distribution of the residuals of the dependent variable.

If more than two groups are involved, the non-parametric tests require that post-hoc analysis be performed to detect which groups are significantly different from each other usually by pairwise comparisons of the groups. A common procedure used with Kruskal-Wallis test is the Conover-Iman method (1981) which is a distribution-free rank transformation method that replaces data by its rank (Conover and Iman 1982).

We performed the Kruskal-Wallis test on the success rate (SR) of the three mechanisms as the SRs' residuals did not follow a normal distribution. Firstly, we considered only one independent variable, the type of service selection mechanism, which has three values: the simultaneous auction, the single auction and the fixed-price mechanism.

As depicted in Table 6.5, Kruskal-Wallis test shows that the computed p-value is lower than the significance level alpha=0.05. Therefore, we should reject the null hypothesis and accept the alternative hypothesis that the SRs of at least two mechanisms are significantly different.

| K (Observed value) | 153.730 |
|----------------------|----------|
| K (Critical value) | 5.991 |
| DF | 2 |
| p-value (Two-tailed) | < 0.0001 |
| alpha | 0.05 |

 Table 6.5. The Kruskal-Wallis test

 (dependent variable: SR, independent variable: type of mechanism)

As specified in Table 6.6, the multiple pairwise comparisons using the Conover-Iman procedure/Two-tailed test shows that the SR of the simultaneous auction is significantly different from the SR of the other two mechanisms, while the SR of the single auction and that of the fixed-price do not differ significantly.
| Sample | Frequency | Sum of ranks | Mean of ranks | Gro | oups |
|-----------------|-----------|--------------|---------------|-----|------|
| Single SR | 3840 | 21077732.000 | 5488.993 | Α | |
| Fixed SR | 3840 | 21087877.500 | 5491.635 | Α | |
| Simultaneous SR | 3840 | 24195350.500 | 6300.873 | | В |

Table 6.6. The Conover-Iman procedure (dependent variable: SR, independent variable: type of mechanism)

Single: Single Auction, Fixed: Fixed-price, Simultaneous: Simultaneous auction

We also want to study the statistical significant of the impact of the economy size and the request complexity at the same time as the impact of the service selection mechanism on the SR, that is, to consider three independent variables simultaneously. However, the Kruskal-Wallis test takes into account only one independent variable. Therefore, we defined a dummy independent variable that represents the combination of the three actual independent variables forming 12 groups.

Table 6.7. The Kruskal-Wallis test (dependent variable: SR, independent variable: type of mechanism, size of economy and requests' complexity)

| K (Observed value) | 3034.562 |
|----------------------|----------|
| K (Critical value) | 19.675 |
| DF | 11 |
| p-value (Two-tailed) | < 0.0001 |
| Alpha | 0.05 |

As depicted by Table 6.7, the Kruskal-Wallis test with three independent variables shows that the computed p-value is lower than the significance level alpha=0.05. This means the null hypothesis is rejected and the alternative hypothesis is accepted. To understand which groups are different from each other, we performed the Conover-Iman procedure as depicted by Table 6.8.

The results show that firstly, the SR of the simultaneous auction is statistically significantly different to that of the single auction and the fixed-price mechanisms in all market sections, except for the (small-simple) section. This means that when the market size is small and the requests are for simple composite services, the choice of the service selection mechanism does not significantly affect the success rate. Secondly, in all market sections, there is no statistical difference between the success rate of a fixed-price and a single auction mechanism.

| Sample | Frequency | Sum of | Mean of | Groups | | | | | | |
|-------------------------------------|-----------|------------|----------|--------|---|---|---|-----------|---|---|
| FR Fixed- Small-Simple | 480 | 711429.50 | 1482.145 | A | | | | <u>ps</u> | | |
| FR Single - Small-Simple | 480 | 711701.50 | 1482.711 | A | | | | | | |
| FR Simultaneous- Small-Simple | 480 | 771509.50 | 1607.311 | A | В | | | | | |
| FR Fixed- Small-Complex | 480 | 853577.50 | 1778.286 | | В | | | | | |
| FR Single- Small-Complex | 480 | 854901.50 | 1781.045 | | В | | | | | |
| FR Simultaneous- Small-Complex | 480 | 1094138.50 | 2279.455 | | | С | | | | |
| FR Single- Large-Simple | 480 | 1729005.00 | 3602.094 | | | | D | | | |
| FR Fixed- Large-Simple | 480 | 1733783.00 | 3612.048 | | | | D | | | |
| FR Fixed- Large-Complex | 480 | 1842155.50 | 3837.824 | | | | D | Е | | |
| FR Single- Large-Complex | 480 | 1856887.50 | 3868.516 | | | | | Е | | |
| FR Simultaneous- Large-Simple | 480 | 2127205.00 | 4431.677 | | | | | | F | |
| FR Simultaneous- Large-Complex | 480 | 2305386.00 | 4802.888 | | | | | | | G |

 Table 6.8. The Conover-Iman procedure (dependent variable: SR,

 independent variable: type of mechanism, size of economy and requests' complexity)

6.6.2 Cost per Composite Service (CPC)

To compare the performance of the proposed simultaneous auction mechanism in terms of the cost of procuring composite services, we defined two units of measurement for the cost:

- Cost per Composite Service (CPC): the average cost of procuring a single composite service. It is calculated by dividing the total cost of procurement achieved by the mechanism by the number of feasible requests.
- Cost per Task (CPT): the average cost of procuring a task. The CPT is determined by dividing the CPC by the average number of tasks in a request, depending on the request's complexity.

Note that as the instances whose SR is equal to zero have an unreal procurement cost of zero, we have removed these instances from the results to get a meaningful cost analysis.

6.6.2.1 Comparison between the Mechanisms

The average cost per composite service (CPC) of each mechanism is depicted in Fig 6.9 based on the four market sections. The results show that firstly, the cost of procuring a composite service is much higher if the requester attends the fixed-price mechanism, compared to the single auction or the simultaneous auction mechanisms. This is according to our expectations from the dynamic pricing strategy in auctions: the price discovery of auction mechanisms can lead to considerably lower prices compared to the requesters' predetermined prices.



Fig 6.9. CPC achieved by the three mechanisms in the four market sections

Secondly, the CPC of the simultaneous auction mechanism does not have a significant difference from that of the single auction. This may not seem intuitive as the greedy strategy of the single auction mechanism aims to find the best providers for each request regardless of other existing requests. Therefore, it is expected to lead to lower procurement cost for each request compared to the simultaneous auction which aims to find the best providers for the collective set of requests, while simultaneously, tries to maximize the number of feasible requests.

The very close CPC of these two mechanisms is likely to be the result of averaging the cost across all successful requests in a simulation round regardless of their order of being considered for service selection. In each simulation round, the single auction mechanism might be able to find very good deals for the early requests. However, with more service offers being matched with the requests, the remaining available services would be the more expensive ones, leading to higher costs for the late arriving requests. Therefore, the average cost of the composite service by the single auction is not very different from that of the simultaneous auction.

To investigate this possibility, we studied the impact of a request's order of being considered for service selection on its cost of service procurement. We expect that in the single auction mechanism the cost of the requests in one simulation round is affected by the requests' order of arrival to the market which is their order of being considered for service selection. In contrast, the cost achieved by the simultaneous auction should not be affected by the requests' order of arrival.

To examine this proposition, we need new performance metrics. The reason is that as the CPC and CPT are calculated by averaging the cost of successful requests in one round of simulation, they do not provide useful information on how the requests' order of arrival in the market affects the procurement cost. Therefore, we separately measured the cost of the first feasible CS and the last feasible one in each simulation round and then averaged them across all simulation rounds.

The results are presented in Fig 6.10, for (a) simple and (b) complex requests where the costs of procuring the first and last requests are compared between the single and simultaneous auction mechanisms. The results indicate that, firstly, the first request attending the single auction mechanism has the lowest procurement cost. At the same time, the last request attending the single auction has the highest cost. Secondly, the first and last requests which attended the simultaneous auction mechanism are procured at very close costs. The costs achieved by the simultaneous auction are positioned between the lowest and highest costs obtained by the single auction mechanism. These results apply to both simple and complex requests.



(a) Simple Requests



(b) Complex Requests



Thirdly, the cost of procuring the first request by the single auction mechanism does not change substantially with having more requests in the market, while the cost of the last request dramatically increases with having more requests. This can be explained based on the single auction strategy which aims to find the best deal for each request, where the first request benefits from having the most offers available, while the last request is most affected by the number of requests that need to be served before it. To summarize, although the auction and simultaneous auction mechanisms achieve close outcomes in terms of the average cost for the requests, the simultaneous auction attains more "homogenous" costs for the requests compared to the single auction, where the order of being considered for service selection affects the final cost. The single auction finds the best deals for the requests that are first to arrive, while the last requests will be procured with the highest cost compared to the requests already been served. The impact of order of arrival on the procurement cost can significantly affect service requesters' decision regarding which service selection mechanism to attend.

6.6.2.2 Comparison between Market Sections

To have a meaningful cost analysis between different market sections, we use the average cost per task (CPT) performance metric. CPT achieved by each mechanism is depicted in Fig 6.11 which highlights several interesting results. Firstly, the cost per task (CPT) in a large economy is considerably lower than in a small economy. This is not a surprise as in larger economies the variety of offers is expected to result in lower cost of procurement for consumers, regardless of the allocation mechanism.

Secondly, in each economy size, the CPT of complex requests is lower than that of simple requests. This might not be very intuitive as generally it is expected that a product with higher complexity in its lifecycle imposes a higher procurement cost on its consumers. This result is related to our formulation of the composite service selection problem. In this formulation, a bid can win the execution of a request's task as far as it offers a web service for that task and it is part of the minimum cost allocation, even if the bid includes services that are irrelevant to that request. Therefore, some of the selected bids for a request may include service offers which are not required by the requester at all, that is, "useless" services in the winning bids is likely to be lower than in a simple request. Ultimately, the higher number of useless services increases the average CPT in simple requests.



Fig 6.11. CPT achieved by the three mechanisms in the four market sections

However, it is worth mentioning that restricting the offers on their number of useless services to allow them to be part of the winning allocation for a request will not necessarily reduce the cost. On the contrary, adding any restriction to the current formulation of the composite service selection problem, including restrictions on the winning offers' configuration, is expected to increase the average cost of procurement.

To summarize, the results presented in Fig 6.9 and Fig 6.11 show that in small economies, a fixed-price mechanism attains a procurement cost marginally worse than the cost achievable through the single auction or the simultaneous auction. However, in large economies, the allocation strategy based on auction models (either single or simultaneous) significantly decreases the procurement cost for the composite service requesters, compared to the fixed-price mechanism.

6.6.2.3 Sensitivity Analysis

In this section, the impact of different values of simulation parameters on the cost of a CS is studied in the simultaneous auction mechanism. The set of parameters includes the number of tasks, number providers, number of bids and number of requests.

The impact of the number of tasks in the market's directory on the average cost of procuring service(s) for a single task (CPT) is depicted in Fig 6.12 for (a) simple requests and (b) complex requests. We can see that in a market with 25 tasks to bid for, the CPT is lower than in a market with 50 tasks, with both simple and complex requests. As was

discussed in subsection 6.6.1.3 (Sensitivity Analysis of the SR), the reason is that with a fixed number of bids, in a market with 25 tasks there are more service offers for each task compared to in a market with 50 tasks which leads to the reduction of cost.

The impact of the number of providers on the market's CPT is illustrated by Fig 6.13, for (a) simple requests and (b) complex requests. The diagram indicates that the number of providers does not have a significant impact on the CTP, which is due to the fact that this parameter is not directly involved in the ILP formulation of the composite service selection problem.



Fig 6.12. Impact of number of tasks on the simultaneous auction's CPT for (a) simple and (b) complex requests

Finally, as illustrated by both Fig 6.12 and Fig 6.13, the impact of the number of bids and number of requests on the CPT is not a surprise: the increase in the number of bids reduces the cost due to a higher supply in the market, while increasing the number of requests results in higher costs caused by a higher demand.



Fig 6.13. Impact of number of providers on the simultaneous auction's CPT for (a) simple and (b) complex requests

6.6.2.4 Statistical Analysis

For statistical analysis, we performed the Kruskal-Wallis test on the cost per composite service (CPC) of the three mechanisms which residuals did not follow a normal distribution. At the first step, we considered the type of service selection mechanism as the independent variable. The null hypothesis is that the CPCs of the three mechanisms come from the same distribution and they are not statically significantly different.

As depicted in Table 6.9, Kruskal-Wallis test shows that the computed p-value is lower than the significance level alpha=0.05. This means that we should reject the null hypothesis and accept the alternative hypothesis that the CPCs of at least two mechanisms are significantly different.

| K (Observed value) | 550.410 |
|----------------------|----------|
| K (Critical value) | 5.991 |
| DF | 2 |
| p-value (Two-tailed) | < 0.0001 |
| Alpha | 0.05 |

Table 6.9. The Kruskal-Wallis test (CPC)(dependent variable: CPC, independent variable: type of mechanism)

As indicated by Table 6.10, the multiple pairwise comparisons based on the Conover-Iman procedure/Two-tailed test shows that firstly, the CPC of the fixed-price is significantly different from the CPC of the other two mechanisms. Secondly, the CPC of the single auction and the simultaneous auction do not differ significantly.

Table 6.10. The Conover-Iman procedure (dependent variable: CPC, independent variable: type of mechanism)

| Sample | Frequency Sum of ranks | | Mean of ranks | Groups | | |
|--------------------|------------------------|--------------|---------------|--------|---|--|
| CPC Single | 3418 | 15701800.000 | 4593.856 | Α | | |
| CPC Simultaneous | 3418 | 16040375.000 | 4692.912 | Α | | |
| CPC Fixed | 3418 | 20835210.000 | 6095.731 | | В | |

Single: Single Auction, Simultaneous: Simultaneous auction, Fixed: Fixed-price

At the second step, we studied the statistical significant of the impact of the economy size and the request complexity at the same time as the impact of the service selection mechanism on the CPC. We perfumed the Kruskal-Wallis test again, this time with three independent variables. Based on the results presented in Table 6.11, the computed p-value is lower than the significance level alpha=0.05 which means that the null hypothesis is rejected and the alternative hypothesis is accepted. This states that the CPC of at least two groups are significantly different from each other. To identify how these groups are different from each other, we performed the Conover-Iman procedure as depicted by Table 6.12.

Table 6.11. The Kruskal-Wallis test (dependent variable: CPC, independent variable: type of mechanism, size of economy and requests' complexity)

| 4252.894 | K (Observed value) |
|----------|----------------------|
| 19.675 | K (Critical value) |
| 11 | DF |
| < 0.0001 | p-value (Two-tailed) |
| 0.05 | alpha |
| | alpha |

| Sample | Frequency | Sum of ranks | Mean of ranks | Groups | | | | | | | |
|--------------------------------------|-----------|-----------------|---------------|--------|---|---|---|---|---|---|---|
| CPC Single- Large-Simple | 480 | 324706.000 | 676.471 | Α | | | | | | | |
| CPC Simultaneous- Large-Simple | 480 | 378696.000 | 788.950 | Α | | | | | | | |
| CPC Single- Small-Simple | 347 | 508269.000 | 1464.752 | | В | | | | | | |
| CPC Simultaneous- Small-Simple | 347 | 511853.500 | 1475.082 | | В | | | | | | |
| CPC Fixed- Large-Simple | 480 | 782395.000 | 1629.990 | | | С | | | | | |
| CPC Fixed- Small-Simple | 347 | 573001.500 | 1651.301 | | | С | | | | | |
| CPC Single- Large-Complex | 480 | 1476848.500 | 3076.768 | | | | D | | | | |
| CPC Simultaneous- Large-Complex | 480 | 1505194.500 | 3135.822 | | | | D | | | | |
| CPC Single- Small-Complex | 342 | 1281796.500 | 3747.943 | | | | | Е | | | |
| CPC Simultaneous- Small-Complex | 342 | 1330697.000 | 3890.927 | | | | | | F | | |
| CPC Fixed- Large-Complex | 480 | 2053802.000 | 4278.754 | | | | | | | G | |
| CPC Fixed- Small-Complex | 342 | 1511618.500 | 4419.937 | | | | | | | | Н |

 Table 6.12. The Conover-Iman procedure (dependent variable: CPC,

 independent variable: type of mechanism, size of economy and requests' complexity)

The results show that firstly, the CPC of the fixed-price mechanism is significantly different to those of the auction-based mechanisms. In other words, the dynamic pricing in auction-based mechanisms achieves lower costs for composite services regardless of the size of the market or the complexity of the requests.

Secondly, the CPC of the simultaneous auction is not statistically significantly different to that of the single auction in any of the market sections, except in the (small-complex) section. As been discussed in subsection 6.6.2.1, the reason for the close costs achieved by the simultaneous auction and the single auction is that the cost is averaged over all the successful requests in one simulation round.

However, the statistical analysis indicates that in small markets with complex requests in demand, the greedy strategy of the single auction achieves lower costs for composite services compared to the simultaneous auction. This means that in this market section, the small number of requests prevents the simultaneous auction to demonstrate its efficiency in terms of the cost.

6.6.3 Solve Time

For any allocation mechanism, an important performance metric is the time taken to find the allocation based on the objective defined for the mechanism. For our specific problem domain and the proposed allocation mechanisms, the solve time is even more important. There are many discussions in both the combinatorial auction and composite service selection literature that as these problems are NP-hard, it is not possible to solve them in polynomial time. As a result, an ILP formulation of the problem is not scalable. Therefore, it was critical to our research to perform an analysis of the "solve time" of our proposed mechanism.

We set a time limit of 60 seconds³³ to prevent the solver (CPLEX 12.6) from being trapped by the complexity of the problem. In a market for complex composite services, such as scientific workflows, 60 seconds is considered a relatively short time to allow the matching mechanism to finish its job. However, in a market for simple requests, such as mobile applications, the time taken to find an allocation can be a determining factor for the requesters and greatly influences their decision about which service selection mechanism to attend.

To analyze the solve time of the three mechanisms, we first studied the number of instances in which the solver reached the time limit, that is, the solve time is greater than or equal to 60 seconds. This study shows that firstly, the fixed-price and single auction mechanisms do not reach the time limit in any of the problem instances. Secondly, the simultaneous mechanism does not reach the time limit when the requests are simple regardless of the size of the economy. However, with complex requests, it reaches to the time limited in 137 instances in the large economy and 20 instances in the small economy, out of the 480 instances in each market section, that is, 28% of instances in the (large-complex) market section and 4% in the (small-complex) section.

We also studied the average solve time of the three mechanisms in each market section. The solve time is plotted in the logarithm scale in Fig 6.14 which is averaged across all the instances of a market section. The results show that the fixed-price mechanism has the shortest solve time in all market sections except (small-simple). This could be

³³ This is the "wall clock" time (second) and is measured through the parameter _solve_elapsed_time.

anticipated as the fixed-price mechanism has a constraint satisfaction approach by searching for the first set of offers which are below the price specified by the requester, rather than searching the whole solution space to find an optimal allocation.

The fixed-price mechanism is then followed by the single auction, again in all market sections except (small-simple) where the simultaneous auction mechanism is the fastest. The single auction is obviously faster than the simultaneous auction as it solves the service selection problem for a single request.



Fig 6.14. Solve Time of the three mechanisms to find the best allocation for the four market sections

The (small-simple) market section is an exception to these trends. The reason is that the single auction and fixed-price mechanisms have to allocate offers to the requests one by one, while the simultaneous auction mechanism solves the allocation problem in one go. Therefore, when the complexity of the problem instance is not high, such as the instances in the (small-simple) section, the simultaneous auction mechanism achieves a better solve time than the two other mechanisms.

Despite having the longest solve time in three out of four market sections, the solve time of the simultaneous auction mechanism can be considered reasonable considering the market sections: the solve time is around 33 seconds in (large-complex) market section, five seconds in (small-complex), one second in (large-simple) and 70 milliseconds in (small-simple) section.

Comparing the solve time across the four market sections shows that for all three mechanisms, the solve time is shorter with simple requests compared to complex ones in both economy sizes. Moreover, it takes less time to solve the service selection problem in small economies compared to large ones, with either request types.

6.7 Conclusion

In this chapter, we introduced and studied the "multiple composite service selection problem". This problem extends the "composite service selection problem" to the setting with multiple requests for composite services. In the web service selection literature, the current study is the first to consider, investigate and propose a solution to the problem of multiple composite web service selection. The significance of this study is its impact on the design of web services' marketplaces where many service requesters and providers meet to trade single and composite web services. As one of the main functions in markets, the multiple composite service selection approaches can enhance the matchmaking between web service offers and requests by considering multiple requests simultaneously.

We proposed two service selection mechanisms based on combinatorial auctions to solve the multiple composite service selection problem by simultaneously matching the web service offers (single or bundled services) and requests (for composite services) that attend a web service market. The two mechanisms are called the "Full-Matching" and the "Partial-Matching" simultaneous auction mechanisms. The Full-Matching mechanism aims to find services for all the requests in the market, while the Partial-Matching's objective is to solve the service selection problem for as many requests as possible. We presented the mathematical formulation of the proposed approaches including the specification of the offers and requests and the Integer Linear Programming (ILP) formulation of the allocation mechanisms.

We performed extensive experiments to evaluate the proposed Partial-Matching simultaneous auction mechanism through simulation. Being the first to consider multiple requests, the design of the evaluation process was a significant challenge for our study. The design included establishing the baseline, defining the performance metrics, designing specific scenarios for the experiment which are presented in the form of four market sections, developing a stochastic model to generate the data and performing an extensive search on current web services' communities to seed the simulation parameters.

The objective of the evaluation was to compare the performance of three mechanisms in solving the multiple composite service selection problem. The three mechanisms comprise: (1) the Partial-Matching simultaneous auction, (2) the single auction model when applied to a set of requests one at a time, and (3) a fixed-price mechanism where service requesters fix the price to be paid for composite services and service providers can take it or leave it. The performance metrics were defined as the success rate (SR) (the ratio of successful requests to all existing requests), the average cost of procuring a composite service and the time to solve the problem.

The evaluations show that that the success rate (SR) of the simultaneous auction mechanism is statistically significantly higher than the SR of the other two mechanisms in all market section except the (small-simple) section. This means that the long-term strategy of the simultaneous auction allows for more efficient matching of service offers and requests. The exception is the market section with small number of participants (small size of the economy) and the requests for simple composite services where the choice of the service selection mechanism does not have a significant impact on the success rate.

Studying the average cost of procuring a composite service, the evaluations show that the simultaneous and the single auction mechanisms do not achieve statistically significantly different costs in the long run in all market sections except for the (small-complex) section. This indicates that the seemingly greedy strategy of the single auction in solving the problem for each request does not achieve lower cost for the collective set of requests except for when the market is small and the requesters demand complex composite services.

However, the simultaneous auction procures the composite services at more homogenous costs. In other words, the requests' order of arrival to the market does not impact their service procurement cost. Whereas in the single auction mechanism, the requests' order of arrival influences the cost: the first request to arrive gets the best deal with lowest prices and the last request gets more expensive deals compared to the requests served before it.

Concerning the solve time taken by mechanisms to solve the multiple composite service selection problem, the solve time of the simultaneous auction is much longer than the two other mechanisms. This was expected due to the complexity of matching many requests and offers at the same time. However, the solve time of the simultaneous auction can be

considered reasonable considering the market sections: for requesters who need complex composite services the solve time is around 33 seconds in large economy markets and five seconds in small economy markets when there is a time limit of 60 seconds to find the optimal allocation. For simple requests, this average is around one second in large economy markets and 70 milliseconds in small economies.

Chapter 7

7 Conclusion

This study aimed to advance our understanding of the composite service selection problem. Mainly, we were interested in investigating the reasons why the current approaches were not as successful as speculated by academia in finding practical applications despite the enormous continuing attention of research community to this problem for more than a decade. The work presented in this dissertation is a contribution to identify and elaborate on the limitations of the current approaches and develop novel techniques to address these limitations.

In this chapter, we summarize the main contributions and results in section 7.1. Then, we shed light on the limitations of our research in section 7.2. As an important part of this study, we also analyze the proposed mechanisms through the lens of mechanism design to examine their limitations in terms of achievable desirable properties (subsection 7.2.2). Finally, the theoretical and practical implications of the results of our study for web service research community in general, and web service composition community in particular are presented in section 7.3. We also suggest directions for future research in this section.

7.1 Contributions and Summary of Results

7.1.1 A Combinatorial Auction Mechanism for Composite Service Selection

In Chapter 5, we developed a mechanism based on combinatorial auctions to solve the composite service selection problem. We have referred to the proposed mechanism as

"the single auction" mechanism as it solves the problem for a single composite service. In this mechanism, service providers bid to offer their services to a composite service requester.

A design based on auction theory delivers dynamic pricing for web services, while at the same time, reduces the complexity of price determination for composite services. Moreover, a design based on combinatorial auctions allows the service providers to offer their services in bundles which is proven to enhance economic efficiency (de Vries and Vohra 2003). The single auction mechanism has been also used to establish a comparison basis for evaluating the proposed "simultaneous auction mechanism" in Chapter 6.

7.1.1.1 Studying the Impact of Bundling on the Cost of a Composite Service

Although there are other proposals that have applied combinatorial auctions for service selection, this is the first study to investigate the impact of bundling, in terms of the number of services in bundles, on the cost of the composite service.

To study the impact of bundling on the performance of the single auction mechanism, we developed a baseline with non-combinatorial bids, that is, each bid only offers a single service. The baseline is established based on current optimization-based and negotiation-based service selection approaches where bundling of services is not considered. The performance metric is defined as the cost of the composite service, calculated as the sum of the costs specified by the winning bids. We also generated several sets of problem instances with different probabilities for the bundle size, that is, different bundle crowdedness.

The results show that firstly, the single auction mechanism achieves lower cost compared to the baseline. More specifically, the cost of the composite service is much higher when providers offer their services in non-combinatorial bids, compared to when we have bundling with discounted price for bundles.

Secondly, having more crowded bundles does not necessarily lead to lower cost for the composite service. Rather, increasing the bundle crowdedness reduces the cost up to a threshold. When bundles' size grows beyond this threshold, the cost starts to increase rather than to decrease. More investigation revealed that the increase in cost is closely related to the number of single-service bids winning the auction, compared to the number of bids with more services winning the auction. This is due to a resource allocation

constraint in the auction mechanism which requires that no more than one service should be selected to execute each task of the composite service. When bundle crowdedness increases, the bids are more likely to have overlaps in services they offer, and consequently, more single-service bids win the auction which leads to increase in the composite service cost.

7.1.1.2 Studying Dynamic Pricing against Fixed Pricing

In Chapter 6, we evaluated the impact of the dynamic pricing in the single auction mechanism on the cost of the composite services. We established a fixed-price mechanism as the baseline. In this baseline, the service requester determines a fixed price for the composite service to be paid to providers. The fixed-price mechanism looks for the first set of providers who collectively offer their services below the pre-determined price. The experiments show that the single auction mechanism achieves significantly lower procurement cost for composite services compared to the fixed-price mechanism.

7.1.1.3 Introducing and Measuring the Cohesion of a Composite Service

In Chapter 5, we introduced and defined the concept of cohesion for composite services. This concept is used to enable service requester manage important quality requirements such as maintainability, reliability and (provider-) dependability. The current study is the first to define the cohesion of the composite service and propose a technique to measure it based on the cohesion of the bundles of services forming the composition.

We defined cohesion for the composite service based on "the direct data dependencies between the participating services offered in a bundle". It is measured as the sum of the cohesion of the bundles winning the auction to execute the composition. A resource allocation constraint was developed to enable the service requester define a lower and an upper bound for the cohesion of the composite service, relative to the maximum cohesion attainable when all services are procured from the same provider. This constraint is added to the winner determination problem of the single auction mechanism.

Adding this constraint imposes extra limitation on the auction mechanism. Therefore, it is important to study its impact on the performance of the mechanism. The comparison was performed among three cases:

- The service requester is interested in high cohesion; that is to include relatively few providers in provisioning to address concerns regarding maintainability of the composite service,
- 2) The service requester is interested in low cohesion; that is to include many providers to prevent high dependability on any particular provider,
- 3) A baseline with no cohesion constraint.

The results show that firstly, the success rate of the auction is considerably low when requester specifies a cohesion constraint as high as the lower bound being at least equal to 75% of the maximum achievable cohesion in the composition. This is due to the allocation constraint in the auction mechanism that requires each task to be procured from exactly one provider. As a result, with the increase in the requester's need of having fewer providers for dependent tasks, the probability of finding non-overlapping bundles from the same provider decreases. This, in turn, leads to a very low success rate for mechanism.

Secondly, the cost of the composite service does not significantly increase when the objective is to achieve low cohesion. In other words, if the service requester decides to be independent of any specific provider by setting a low cohesion constraint, this is achievable at almost no extra cost. In contrast, reaching a high cohesion increases the cost of the composite service on average by 20%. These findings can help the service requesters to set the appropriate level of cohesion considering the trade-off between the cohesion level, the cost of the composite service and the success rate.

7.1.1.4 Identify the Need and Develop Constraints to Manage the Configuration of Composite Service Provisioning

We identified the service requester's need to manage the configuration of service providers in the execution of a composite service. More specifically, we identified two important patterns of service providers' involvement in the composition: a set of tasks need to be executed by the "same" provider or by "different" providers. These patterns are very important in the context of service requester's security and privacy concerns.

We developed two resource allocation constraints corresponding to each of the patterns, which were added to the ILP formulation of the winner determination problem of the auction mechanism.

7.1.2 A Simultaneous Auction Mechanism for Trading Composite Services

7.1.2.1 Identifying and Investigating the "Multiple Composite Service Selection" Problem

In Chapter 7, we introduced the problem of "multiple composite service selection" by extending the composite service selection problem to include multiple requests. Composite services have been recognized as an essential part of a marketplace for web services, where service providers and requesters meet to trade single and composite web service. However, very limited study has been done to examine how such a marketplace impacts the composite service selection process. To the best of our knowledge, this is the first study to investigate composite service selection in the presence of multiple requests for composite services. All other approaches solve the problem for a single request and no discussion exists about solving the problem for multiple requests, neither simultaneously nor one by one.

7.1.2.2 Proposing Two Simultaneous Auction Mechanisms to Solve Multiple Composite Service Selection

We developed "a simultaneous auction mechanism" that solves the composite service selection problem for multiple requests. The simultaneous auction mechanism is based on combinatorial auctions. A design based on auction models in general, and combinatorial auctions in particular achieves these desirable features: (1) enhances dynamic pricing for composite services compared to a fixed pricing strategy, (2) facilitates price determination of single and composite services by sending constant feedback about the status of supply and demand obtained from the information revealed after each auction, and (3) accommodates the need for bundling web services due to the inter-service dependencies between constituent services of a composition.

The simultaneous auction mechanism aims to procure web services for the requests at the minimum price, subject to resource allocation constraints of service requesters. It comes in two variations: the Full-Matching and the Partial-Matching mechanisms. The Full-Matching mechanism aims to allocate services to all the requests. Therefore, in the case that some requests are not feasible due to their allocation constraints, the whole auction fails and no request, even the feasible one, will be assigned any services. The Partial-Matching mechanism relaxes the need to solve the problem for all requests; rather, it aims to solve the problem for the largest set of feasible requests.

objective of the Partial-Matching mechanism is to maximize the number of feasible requests, and then, minimize the cost for this set.

The proposed Partial-Matching mechanism is backed up with evaluation through extensive experiments based on simulations. The objective of the evaluation was to compare the performance of the Partial-Matching simultaneous auction with two other mechanisms which solve the problem for multiple requests one at a time: (1) the single auction mechanism when applied to a set of requests one at a time, (2) a fixed-price mechanism where the service requesters determine a fixed price to be paid for the composite services. The performance metrics were defined as the success rate (the number of feasible requests to all the requests), the average cost of a composite service achieved by the mechanism and the time to solve the problem for a specific set of requests. Moreover, we defined four market sections for our study based on: (1) the complexity of the composite service requests, calibrated based on the number of services in a composition, and (2) the number of providers and requesters attending the market; referred to as, the size of the economy.

7.1.2.3 Studying the Impact of the Choice of Mechanism on the Service Selection's Success Rate

The evaluations show the following results. Firstly, simultaneous auction achieves the highest success rate, regardless of the requests' complexity or the size of economy. This result is statistically significant for all market sections except the (small-simple) section where the simultaneous auction mechanism is not significantly better than the other two mechanisms. Secondly, comparing the single auction's success rate to that of the fixed-price mechanism shows that the success rate achieved by the two mechanisms does not have a significant difference as they follow a similar greedy service selection strategy.

To summarize, the difference between the success rates of the simultaneous auction to that of the other two mechanisms grows as the market attracts more participants (size of economy changes from small to large) and as the requesters demand more complex composite services. However, in the simplest market section (small-simple) the difference is not significance. This means that when market is small and requesters are after simple composite services, the choice of the mechanism does not significantly change the success rate of composite service selection. However, in a small market whose service requesters need complex composite services and in large markets regardless of the requests' complexity, applying a simultaneous auction can significantly improve the success rate of composite service selection.

7.1.2.4 Studying the Impact of the Choice of Mechanism on the Cost of the Composite Service

The evaluations show the following results. Firstly, the difference between the costs of a composite service achieved by the simultaneous auction to that achieved by the single auction is not statistically significant in three out of four market sections. This may not seem very intuitive as we expected the greedy strategy of the single auction to achieve lower costs. However, as the cost per composite service (CPC) is calculated by averaging the costs of all the requests in a simulation round, the simultaneous auction can keep up with the single auction in the long run and achieves prices as low as the single auction. However, in small markets with complex requests, the greedy strategy of the single auction in terms of achieving lower prices.

Secondly, while the order of considering the requests for service selection does not affect the simultaneous auction, it considerably affects the cost achieved by the single auction due to its first-come, first-served policy. This speculation was backed up by repeating the measurement of cost, this time comparing the two mechanisms' costs of the first request and that of the last request, being averaged across all the simulation rounds. The investigation indicates that the simultaneous auction achieves more homogenous costs for the requests, while the cost by the single auction is considerably affected by the order of requests: the first request gets the best deals with lowest prices and the last request gests the more expensive deals. This impact can be a determining factor for a requester when choosing which mechanism to attend for service selection. Thirdly, comparing the cost achieved by the two auction-based mechanisms to that of the fixed-price mechanism shows that the cost by the auctioned-based mechanisms is lower than the cost by the fixedprice mechanism. The difference in cost is statistically significant in all market sections.

To summarize, the cost of procuring a composite service through the simultaneous auction is not significantly different to that of the single auction in all market sections except for (small-complex), where the single auction can procure composite services at lower prices. However, the simultaneous auction achieves more homogenous costs for the requests in one simulation round, compared to the single auction where the cost is affected by the order of considering the requests for service selection. This can be a determining factor for requesters to consider before choosing a service selection mechanism. Finally, in all market sections, the dynamic pricing in auction-based mechanisms leads to lower prices compared to the fixed-price mechanism.

7.1.2.5 Studying the Impact of the Choice of Mechanism on the Service Selection's Time

In the experiments to evaluate the time of the proposed simultaneous auction mechanism, we set a time limit of 60 seconds ("wall clock" time) to prevent the mechanism from being trapped by hard instances of the problem. Similar time limit was set for the single auction and fixed-price mechanisms. In a market for complex composite services, such as scientific workflows, 60 seconds is considered relatively a very short time to allow the matching mechanism to finish its job. However, in a market for simple requests, such as mobile applications, the time taken to find an allocation can be a determining factor for requesters in choosing what service selection mechanism to attend. The evaluations show the following results.

Firstly, the simultaneous auction mechanism does not reach the time limit when the requests are simple. For complex requests, the simultaneous mechanism reaches the time limit in 28% of the instances in a large market and 4% of the instances in a small market. This means that for this set of instances, we cannot be sure if the simultaneous auction mechanism has found the optimal allocation or not. However, based on the following results from studying the cost, we conclude that it is very likely that the simultaneous auction finds the optimal allocation but the optimality cannot be proven within the time limit: (1) the cost of the simultaneous auction was not significantly different to that of the single auction, (2) the single auction does find the optimal allocation as the time limit is never reached.

Secondly, the single auction and fixed-price mechanisms never reach the time limit as they solve the service selection problem for requests one by one. Thirdly, the fixed-price mechanism has the shortest solve time, followed by firstly, the single auction mechanism, and secondly, by the simultaneous auction mechanism in all market sections except (small-simple) where the simultaneous auction mechanism is the fastest. The reason is that the single auction and fixed-price mechanisms have to allocate offers to the requests one by one, while the simultaneous auction mechanism solves the allocation problem in one go. Therefore, when the complexity of the problem instances is not high, such as the instances in the (small-simple) section, the simultaneous auction mechanism achieves a better solve time than the two other mechanisms.

To summarize, although it takes longer for the simultaneous auction to solve the multiple composite service selection problem in three of the market sections (all except the simple-small section) compared to the other two mechanisms, the solve time can be considered reasonable with respect to the market sections. In the (large-complex) market section, this time is 31.23 seconds excluding the solve time of instances which reached the time limit, and 33.64 seconds for all instances. In the (small-complex) section, the average solve time is 5 seconds. These times can be reasonable for requesters who need complex composite services that are not time sensitive. For requesters who require a simple composite service, the average solve time is below one second in both economy sizes.

7.1.3 A Comprehensive Evaluation Process

The evaluation process was presented in Chapter 4. This process was designed to support a comprehensive evaluation of the proposed auction-based approaches through simulations. The evaluation process includes the following elements:

- 1. The performance metrics
- 2. The baseline for comparison
- 3. The scenarios to be investigated
- 4. The data generation model
- 5. The seeding of the simulation

After the first set of experiments for the proposed single auction mechanism, we revised and enhanced the implementation of the evaluation process's elements.

A challenge for this study was the absence of publicly available data sets of web services' prices, bundling of web services and composite web services. Moreover, the limited empirical research on composite service selection with bundling or in the presence of multiple requests further exacerbated the problem of data generation and seeding.

To address the problem of data for bundled service offers, we mainly referred to the combinatorial auction literature for the required data distributions. We initially started with data sets generated by CATS (Leyton-Brown et al. 2000) which includes a family of

distributions to generate data for combinatorial auctions. However, the data generated by CATS has some problems regarding: (1) the variety and number of parameters involved in the data generation that their impact on the output is not very clear; (2) it is rather an old test suit which is not supported anymore. Therefore, we later moved to generate the complete data set based on a well-known distribution for combinatorial auctions called decay distribution (Sandholm 2002). For the pricing of services, we referred to the IPV (independent private valuation) model to generate services' prices. In auction theory, IPV is the most common valuation model used to define the valuation of the bidders for the items under auction.

To address the problem for multiple requests, we studied a number of web service communities on the Internet where service requesters need composite services. However, these communities were very different in terms of their number of registered web services and composite services and the complexity of the composite services. Therefore, we decided to limit our study to specific market sections, rather than a generic market. As discussed in subsection 7.1.2.2, the market sections were designed based on two factors: (1) the complexity of the requests for composite services, and (2) the size of economy. Based on these factors, the web services' market was divided to four sections.

The proposed evaluation process in general, and the data generation model and seeding of the simulation parameter in particular provide a useful framework for the service selection community. It improves the clarity of the experiments in this area and establishes the basis for more realistic data sets. This is achieved by identifying and defining a set of elements for evaluations performed in this area and relying on the data obtained from existing web service communities for seeding the simulations.

7.1.4 Summary Remarks

In summary, we have provided empirical evidence that considering bundling and multiple requests improves the performance of the composite service selection approach, specifically in terms of the cost of procuring a composite service and the success rate of the approach.

The proposed single auction mechanism achieves lower costs for composite services by allowing providers to bundle their services with a discounted price when compared to the baseline with services being offered independently. However, increasing the size of the bundles reduces the cost up to a threshold, beyond which, the cost starts to increase rather than to decrease.

The proposed simultaneous auction achieves significantly higher success rate in allocating services to requests compared to the single auction and fixed-price mechanisms, both of which follow a greedy strategy by assigning services to requests one at a time. The average cost of a composite service achieved by the simultaneous auction is not significantly different to that of the single auction mechanism despite the greedy approach of the single auction. However, the simultaneous auction mechanism achieves more homogenous costs for different requests while the cost achieved by the single auction is affected by the order the requests are being considered for service selection. This can be a determining factor for service requesters when considering which service selection mechanism to attend. Finally, in both single auction and the simultaneous auction models leads to lower prices for the requesters compared to the fixed-price mechanism.

7.2 Limitations

7.2.1 Solve Time

Composite service selection problem is proven to be NP-complete (Yu et al. 2007). This means that it is not possible to guarantee to find an optimal solution within polynomial time, as discussed in subsection 3.3. Combinatorial auctions are also proven to be NP-complete (Sandholm 2002). This indicates that the complexity of a composite service selection approach based on combinatorial auctions is at least as hard as any of these two problems.

In the proposed auction-based approaches, the winner determination and matching problems are mapped to integer linear programming problems. The ILP problems are then implemented using AMPL, a mathematical language, and are solved by CPLEX, the best known and most widely used optimization solver. To avoid the complexity of the problem, we had to set a time limit of 60 seconds for the solver. The results of the experiments on the simultaneous auction presented here are based on using the latest version of CPLEX (CPLEX 12.0, 2014). Initially, we ran the experiments with an older

version (CPLEX 10, 2005) where the success rate of the simultaneous auction was remarkably lower than what was achieved by the new version of the CPLEX.

Despite the acceptable solve time of the new solver in solving the very hard instances of the problem (market section large-complex, average time 33.64 seconds), we need to notice that this time was achieved by having set the time limit and within the generated data sets. It is not possible to comment on the scalability of the proposed mechanism beyond these two limitations.

One possibility to reduce the complexity of the problem is to aim for finding near optimal solutions using heuristic approaches, rather than optimal solutions. Many heuristics have already been proposed to solve composite service selection (Yu, Zhang, and Lin 2007, Berbner et al. 2006, Menasce et al. 2010) or for combinatorial auctions (Sandholm 2002). However, it's not possible to apply any of these heuristics directly to a composite service selection problem based on combinatorial auctions. The heuristics in composite service selection do not support bundling of services and the heuristics of combinatorial auctions assume free disposal (Sandholm et al. 2002). As discussed in subsection 4.3.1.2, free disposal does not exist in composite service selection as the composite service requester needs to find services for all the tasks in the business process, abstracting the composite service. Therefore, there is a need to new heuristic approaches considering the properties of both composite service selection and combinatorial auction problems.

7.2.2 Economic Efficiency and Incentive Compatibility

Economic efficiency is an important objective to achieve for auction designers. Meanwhile, the necessary condition for designing an efficient auction is to incentivize the bidders to truthfully reveal their valuation for the items under auction. In other words, the mechanism needs to be incentive-compatible. However, it is not always easy to achieve incentive-compatibility due to a number of reasons, as discussed below:

7.2.2.1 Computational Cost

Achieving incentive compatibility can be expensive in computational terms, especially when multiple items are auctioned simultaneously as in combinatorial auctions. The Vickrey-Clarke-Groves (VCG) mechanism (Vickrey 1961; Clarke 1971; Groves 1973) is

the most notable mechanism to achieve incentive-compatibility and economic efficiency in such a setting.

However, the VCG payment requires extra optimization problems to be solved to calculate the payment (to or from) each winning bidder: first, the winner determination problem, and then for each winning bidder, the WDP when taking out the bids of the winning bidder. Considering the complexity of the winner determination problem in composite service selection, calculating the Vickrey payments may not be scalable specially when the economy size of the web service market grows or requests need more complex composite services.

7.2.2.2 Bidders Reluctant to Truthful Revelation

Bidders might be reluctant to be truthful due to behavioral reasons as well as economic reasons (Rothkopf et al. 1990). From the behavioral aspect, many people with experience in conducting business are reluctant to reveal their true cost or value and they strongly prefer to keep such information confidential (Rothkopf et al. 1990).

Economic reasons are related to the subsequent transactions after the auction. Most of the discussions about incentive compatibility have assumed that an auction is an isolated event. However, in many situations an auction might be followed by subsequent transactions with the involvement of the bidders, such as: other auctions for similar or identical items, secondary markets to re-sale the auctioned items (Haile 2000) or when further negotiation is required to seal the contract (Rothkopf et al. 1990).

Moreover, a truth revealing strategy may inform competitors of valuable information. In our problem domain, a service provider can sell a web service over and over, based on their capacity, to different requesters. In this setting, providers might be concerned about attending a mechanism which reveals their true value (cost) of the offered services to all the participants. This information can be used to their disadvantage, in the subsequent auctions, secondary markets or negotiations.

This is an important factor to consider when designing auctions. For example in the case of auction for spectrum frequency, the auction designers as well as the bidders know that the next auction in this domain will be held not in the near future, and therefore, the economics of the problem is very likely to be completely different at the time of the next auction. As a result, the bidders may not be concerned about the revelation of their true values. This situation is very different to web service domain where auctions may be apart only in minutes and the revealed information can be used to the disadvantage of providers.

7.2.2.3 The Existence of Secondary Markets for Composite Services

Secondary markets refer to a market for items or services which had been sold in a previous auction. The resale opportunity has fundamental effect on the bidding strategies, seller behavior and the interpretation of the bidding data (Haile 2000).

In composite web service domain, there are many discussions around the role of intermediary entities, such as in (Tang 2004), which are interested in composing web services together and sell the composite services to the end-users. Having these intermediaries as second markets adds to the complexity of designing incentive compatible and economic efficient auctions for composite services.

7.2.2.4 Truthfulness for all Market Participants

The auction model proposed for a single composite service is incentive-compatible with the use of the VCG payment. It is truthful for both sides; providers and the requester. Providers are incentivized through the VCG payment. Requesters, as well, has no incentive to declare an untruthful budget constraint, as the winners are the providers with the lowest prices in the market. In other words, the requester cannot improve their utility by shading their budget.

However, the situation is different for the simultaneous auction mechanism proposed for multiple requests. Let us assume that the auction designer accepts the computational cost of a VCG mechanism, and the providers are not concerned about the truthful revelation. The VCG payment makes the mechanism truthful for service providers. In this setting, the requesters need to pay for the VCG payment; in other words they accept to pay the price of making providers truthful. As a result of this situation, the requesters find enough incentive to shade their budgets.

Let us discuss this problem more specifically for the two allocation mechanisms: the Full-Matching and the Partial-Matching mechanisms with providers which have limited resources.³⁴

In the Full-Matching mechanism, a service requester knows that the mechanism will be successful only if it can find solutions for all the requests. This requirement forces the mechanism to prioritize the requests based on their budget constraints when allocating services to them. Therefore, if there are two requests requiring similar services but they have different budget constraints, the request with a lower budget constraint will end up receiving lower prices, and the one with a higher budget constraint has to procure the same composite service at a higher price. This incentivizes all requesters to shade their budgets to receive less expensive services.

As an example, consider a scenario where there are four requests in the marker, r_1 to r_4 , and the Full-Matching mechanism finds two feasible solutions, S_1 and S_2 , for the requests with equal total costs (14 for both). S_1 assigns the combination of (5,4,3,2) as the provisioning cost to the requests, while S_2 assigns the combination of (6,3,3,2), as depicted in Table 7.1.

As both solutions are feasible considering the budget constraints of the requesters and they achieve equal minimum cost, the mechanism needs to break the tie randomly to be considered fair by the participants. However, if r_2 shades the budget down to 3, the mechanism has to choose S₂, instead of randomly breaking the ties. This means that r_2 can improve its utility by manipulating the mechanism.

| | r 1 | r 2 | r 3 | r 4 | total minimum cost |
|----------------|------------|------------|------------|------------|--------------------|
| Real Budget | 8 | 4 | 5 | 3 | |
| Shaded Budget | 8 | 3 | 5 | 3 | |
| S ₁ | 5 | 4 | 3 | 2 | 14 |
| S2 | 6 | 3 | 3 | 2 | 14 |

Table 7.1. An example scenario for the full matching mechanism

³⁴ If providers do not have limitation over their resources (each bid can be assigned to as many requests as required), the requesters are not competing against each other and therefore, a VCG payment can make both sides truthful.

The Partial-Matching mechanism aims to minimize the cost for as many requests as possible. Therefore, a requester that has a higher budget will be assigned more expensive services (up to the budget constraint, of course) and the request with lower budget will get the less expensive providers, for the mechanism to be able to maximize the number of feasible requests. The example depicted in Table 7.2 shows that if the requesters were all truthful, the mechanism could have chosen S₁ which achieves the minimum cost. While if r_2 strategically shade its budget, the mechanism needs to choose S₂ to primarily maximize the number of feasible requests, and secondarily, minimizes the cost for this set.

| | r 1 | r ₂ | r 3 | ľ 4 | total minimum cost |
|-----------------------|------------|-----------------------|------------|------------|--------------------|
| Real Budget | 8 | 7 | 5 | 3 | |
| Shaded Budget | 8 | 5 | 5 | 3 | |
| S ₁ | 5 | 6 | 3 | 2 | 16 |
| S ₂ | 8 | 4 | 3 | 2 | 17 |

Table 7.2. An example scenario for the partial matching mechanism

7.2.2.5 Budget-balanced Property

If we decide to design a mechanism which is incentive compatible for requesters as well as providers, we will need a payment rule which sets the price of each composite service lower than the truthful budget of the requester, and calculates a payment for each winning provider higher than the truthful cost of the service. However, as discussed in subsection 2.4.1.7 (Myerson–Satterthwaite Impossibility Theorem), such a payment rule makes the mechanism not to be budget-balanced. This means that we need an entity from outside the market to subsidize the market by paying for the cost of truthfulness. Such a mechanism may be practical for some domains, but not for the composite service selection problem.

To summarize, there is a need to new models of the composite service selection problem that consider the impact of revealing the information of one auction to the next ones. These models also need to study the trade-off between the four desirable properties, economic efficiency, incentive compatibility, individual rationality and budget balance, to come up with new approaches for composite service selection problem to achieve economic efficiency.

7.2.3 Data

As we discussed in subsection 7.1.3, the public available data for real web services is very limited. The existing data sets such as QWS Dataset (Al-Masri and Mahmoud 2007) do not have any information about web services' prices. Therefore, generating appropriate datasets for evaluation purposes have been a challenge for web service research communities.

We used common data distributions in combinatorial auction research to generate our data. We conducted experiments with two distributions, CATS arbitrary and Decay, being well-known distributions in combinatorial auction literature.

However, more studies are required to test the appropriateness of other distributions for composite service selection and also compare the results of different distributions.

7.3 Implications and Future Work

The findings of this research have significant theoretical and practical implications. Theoretically, this research extends the current research on composite service selection problem, building on auction theory and mechanism design. By building on auction theory, the proposed mechanisms have in their core the means for "dynamic pricing", which we demonstrated that reduces the cost of procuring composite services compared to a fixed pricing strategy. Another key feature of an auction-based approach is to facilitate the price determination of the single and composite services for service providers and requesters. This is achieved by constantly signaling the status of supply and demand obtained from the information revealed after each auction.

In the proposed mechanisms, we addressed two important aspects of composite service selection: bundling of web services by service providers and the presence of multiple composite service requests. Bundling is proven to increase the economic efficiency due to allowing the bidders more fully express their preferences for combination of items (de Vries and Vohra 2003). In composite service selection, bundling allows providers express more complex preferences in offering web services:

1. They can offer discount for the price of bundled services by internalizing some of the cost of service provisioning,

2. The quality offer for the bundle of services can also be improved by being procured from a single provider, rather than many.

Composite services are considered to be an import element in the web service marketplaces (Papazoglou 2003; Yarom et al. 2004; Legner 2009; Weinhardt et al. 2011b). However, there is a very limited study on how these markets impact composite service selection. In other words, the problem of composite service selection in the presence of multiple composite services has not received the deserving attention of the research community. This study opens up a new research direction in the research on web service composition and composite service selection. The results of the current study show that the current approaches of service selection do not efficiently extend to the setting with multiple composite services, in terms of the success rate of the mechanism. If the service selection approach does not consider the presence of multiple requests at the time of looking for the optimal service allocation, it fails to accommodate the competition of service requesters for the limited resources of service providers, which leads to many requests remain un-executable (no web services for them).

By addressing the two aspects of bundling and multiple requests, this work also makes a practical contribution. It enhances the practicality of service selection approaches by moving the models of web service offers and composite services to more realistic settings: service offers can be single or bundle of services and there can be more than one request for composite services.

Furthermore, the proposed simultaneous mechanism contributes to the development of web service marketplaces. The results of our experiments can provide useful guidelines for web service market makers in choosing the appropriate service matching mechanism based on the complexity of the requests for composite services and the size of the economy in the target market.

Future work involves extending the simultaneous auction mechanism to solve the abovementioned problems, including the time complexity and the need for economic efficiency. Moreover, an interesting extension to the current study would be supporting the application of the simultaneous auction mechanism in other areas of interest, such as cloud computing. With the increasing interest to the delivery model of pay-as-you-go and providing more high level services in the cloud following the Software as a Service (SaaS) model, more web service providers are expected to move to the cloud. The cloud environment can serve as the management layer for web service marketplaces, which was originally seen as the role of grid computing (Papazoglou 2003). In this context, service selection can be offered by the cloud or by third-parties who offer value- added services in the cloud environments to the customers.

8 References

- Agarwal, V. & Jalote, P., 2010. From Specification to Adaptation: An Integrated QoSdriven Approach for Dynamic Adaptation of Web Service Compositions. In *IEEE International Conference on Web Services (ICWS)*. pp. 275–282. Available at: 10.1109/ICWS.2010.39.
- Aggarwal, G., Goel, G. & Mehta, A., 2009. Efficiency of (revenue-)optimal mechanisms. In *Proceedings of the tenth ACM conference on Electronic commerce (EC '09)*. New York, USA: ACM Press, pp. 235–242. Available at: http://dl.acm.org/citation.cfm?id=1566374.1566408 [Accessed November 3, 2014].
- Aggarwal, R., Kunal, V., Miller, J. & Milnor, W., 2004. Constraint driven Web service composition in METEOR-S. In *Proceedings*. 2004 IEEE International Conference on Services Computing, 2004. (SCC 2004). pp. 23–30.
- Ai, L. & Tang, M., 2008. QoS-Based Web Service Composition Accommodating Interservice Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm. In *IEEE Fourth International Conference on eScience (eScience '08)*. pp. 119–126.
- Al-Masri, E. & Mahmoud, Q.H., 2007. QoS-based Discovery and Ranking of Web Services. In Proceedings of 16th International Conference on Computer Communications and Networks, 2007. ICCCN 2007. pp. 529–534.
- Alonso, G., Casati, F., Kuno, H. & Machiraju, V., 2004. Web services: concepts, architectures and applications, Springer Verlag.
- Alrifai, M., Skoutas, D. & Risse, T., 2010. Selecting skyline services for QoS-based web service composition. *Proceedings of the 19th international conference on World wide web*, pp.11–20.
- Amazon, 2014. Amazon EC2 Spot Instances. Amazon Web Services, Inc. Available at: http://aws.amazon.com/ec2/purchasing-options/spot-instances/ [Accessed April 14, 2014].
- Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M. & (2004–2007), O.G.F. (OGF), 2007. Web services agreement specification (WS-Agreement), Open Grid Forum (OGF) (2004–2007). Available at: http://ogsa.gridforum.org/Public_Comment_Docs/Documents/Oct-2006/WS-AgreementSpecificationDraftFinal_sp_tn_jpver_v2.pdf.
- Ardagna, D. & Pernici, B., 2007. Adaptive Service Composition in Flexible Processes. *IEEE Trans. on Software Eng.*, 33(6), pp.369–384. Available at: 10.1109/TSE.2007.1011.
- Ausubel, L.M., 2003. Auction theory for the new economy. In Derek C. Jones, ed. New Economy Handbook. San Diego, California, pp. 123–162.
- Ausubel, L.M. & Cramton, P., 1999. *The optimality of being efficient*, Unpublished paper. Available at: http://www.ausubel.com/auction-papers/99wp-optimality-of-beingefficient.pdf.
- Ausubel, L.M. & Milgrom, P., 2006. The Lovely but Lonely Vickrey Auction. In P. C. Cramton, Y. Shoham, & R. Steinberg, eds. *Combinatorial Auctions*. Cambridge, Mass.; London: MIT Press.
- Bakos, Y., 1991. A strategic analysis of electronic marketplaces. *MIS quarterly*, 15(3), pp.295–310.
- Bakos, Y., 1998. The emerging role of electronic marketplaces on the Internet. *Communications of the ACM*, 41(8), pp.35–42.

- Baldwin, C.Y. & Clark, K.B., 2000. *Design Rules, Vol. 1:The Power of Modularity*, The MIT Press.
- Berbner, R., Spahn, M., Repp, N., Heckmann, O. & Steinmetz, R., 2006. Heuristics for QoS-aware Web Service Composition. In *International Conference on Web Services* (ICWS '06) . pp. 72–82.
- Bergen, M., Ritson, M., Dutta, S., Levy, D. & Zbaracki, M., 2003. Shattering the Myth of Costless Price Changes. *European Management Journal*, 21(6), pp.663–669. Available at: http://www.sciencedirect.com/science/article/pii/S0263237303001312 [Accessed February 23, 2014].
- Bichler, M., 2001. Electronic commerce and electronic marketplaces . In *The future of eMarkets: multi-dimensional market mechanisms*. Cambridge University Press. Available at: http://books.google.com/books?id=-lhLmmSM--4C.
- Bichler, M., Davenport, A., Hohner, G. & Kalagnanam, J., 2006. Industrial procurement auctions. In P. C. Cramton, Y. Shoham, & R. Steinberg, eds. *Combinatorial auctions*. Cambridge, Mass.; London: MIT Press Cambridge, MA, pp. 593–612.
- Bichler, M. & Kalagnanam, J.R., 2006. Software frameworks for advanced procurement auction markets. *Communications of the ACM*, 49(12), pp.104–108.
- Blau, B., Conte, T. & Dinther, C. van, 2010. A multidimensional procurement auction for trading composite services. *Electronic Commerce Research and Applications*, 9(5), pp.460–472.
- Blumrosen, L. & Nisan, N., 2007. Combinatorial Auctions. In *Algorithmic Game Theory*. Cambridge University Press, pp. 267–299.
- Bodenstaff, L., Wombacher, A., Reichert, M. & Jaeger, M.C., 2008. Monitoring Dependencies for SLAs: The MoDe4SLA Approach. In *IEEE International Conference on Services Computing*. IEEE, pp. 21–29.

- Brehm, N. & Golinska, P., 2009. Engineering web service markets for federated business applications. In *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. Springer, pp. 366–373.
- Briand, L.C., Daly, J.W. & Wüst, J., 1998. A unified framework for cohesion measurement in object-oriented systems. *Empirical Software Engineering*, 3(1), pp.65–117.
- Canfora, G., Penta, M. Di, Esposito, R. & Villani, M.L., 2005. An approach for QoSaware service composition based on genetic algorithms. *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp.1069–1075.
- Cantillon, E. & Pesendorfer, M., 2006. Auctioning bus routes: The London experience. In and R. S. Peter Cramton, Yoav Shoham, ed. *Combinatorial Auctions*. Cambridge, Mass.; London: MIT Press.
- Carlson, S.E., 1995. A general method for handling constraints in genetic algorithms. In Proceedings of the Second Annual Joint Conference on Information Science. pp. 663–667.
- Charfi, A. & Mezini, M., 2004. Hybrid web service composition: business processes meet business rules. *Proceedings of the 2nd international conference on Service oriented computing*, pp.30–38.
- Chen, J., Huang, H. & Kauffman, R.J., 2011. A public procurement combinatorial auction mechanism with quality assignment. *Decision Support Systems*, 51(3), pp.480–492.
- Chhetri, M.B., Lin, J., Goh, S.K., Yan, J., Zhang, J.Y. & Kowalczyk, R., 2006. A coordinated architecture for the agent-based service level agreement negotiation of Web service composition. In *Australian Software Engineering Conference* (ASWEC). p. 10 pp.
- Clarke, E.H., 1971. Multipart pricing of public goods. *Public choice*, 11(1), pp.17–33.
- Comuzzi, M., Francalanci, C. & Giacomazzi, P., 2005. Trade-Off Based Negotiation of Traffic Conditioning and Service Level Agreements in DiffServ Networks. In F.

Chiara & G. Paolo, eds. *International Conference on Advanced Information Networking and Applications*. Taipei, Taiwan, pp. 189–194. Available at: http://doi.ieeecomputersociety.org/10.1109/AINA.2005.330.

- Comuzzi, M. & Pernici, B., 2009. A framework for QoS-based Web service contracting. *ACM Trans. Web*, 3(3), pp.1–52.
- Comuzzi, M. & Pernici, B., 2005. An architecture for flexible Web service QoS negotiation. In *IEEE International Enterprise Computing Conference (EDOC)*. pp. 70–79.
- Conover, W.J. & Iman, R.L., 1982. Analysis of Covariance Using the Rank Transformation. *Biometrics*, 38(3), pp.715–724. Available at: http://www.jstor.org/stable/2530051.
- Conover, W.J. & Iman, R.L., 1981. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician*, 35(3), pp.124–129.
- Cornes, R., 1996. *The theory of externalities, public goods, and club goods*, Cambridge University Press.
- Cramton, P.C., Shoham, Y. & Steinberg, R., 2006. *Combinatorial auctions*, Cambridge, Mass.; London: MIT Press.
- Edelman, B., Ostrovsky, M. & Schwarz, M., 2007. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1), pp.242–259.
- Eder, J., Kappel, G. & Schrefl, M., 1994. Coupling and cohesion in object-oriented systems. *Technical Reprot, University of Klagenfurt, Austria*.
- Erl, T., 2005. *Service-oriented architecture: concepts, technology, and design*, Prentice Hall PTR Upper Saddle River, NJ, USA.
- Esmaeilsabzali, S. & Larson, K., 2005. Service Allocation for Composite Web Services Based on Quality Attributes. In Seventh IEEE International Conference on E-Commerce Technology Workshops. IEEE, pp. 71–82.

- Faratin, P., Sierra, C. & Jennings, N.R., 1998. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4), pp.159–182.
- Faratin, P., Sierra, C. & Jennings, N.R., 2002. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2), pp.205–237.
 Available at: http://www.sciencedirect.com/science/article/B6TYF-472BM4F-7/2/cf86220dc936205556325888e0f4aed0.
- Ferguson, D.F., 1989. *The application of microeconomics to the design of resource allocation and control algorithms*. PhD Thesis, Columbia University.
- Ferguson, D.F., Nikolaou, C., Sairamesh, J. & Yemini, Y., 1996. Economic models for allocating resources in computer systems. In S. H. Clearwater, ed. *Market-Based Control A Paradigm for Distributed Resource Allocation*. World Scientific Publishing Co., Inc., pp. 156–183.
- Fonseca, C.M. & Fleming, P.J., 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms- Part I: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(1), pp.26–37.
- Foundation for Intelligent Physical Agents, 2000. FIPA Contract Net Interaction Protocol. Available at: http://www.fipa.org/specs/fipa00029/SC00029H.pdf.
- Fourer, R., Gay, D.M. & Kernighan, B.W., 1990. A Modeling Language for Mathematical Programming. *Management Science*, 36(5), pp.519–554.
- Gallego, G. & Ryzin, G. van, 1997. A Multiproduct Dynamic Pricing Problem and Its Applications to Network Yield Management. *Operations Research*, 45(1), pp.24–41.
- Garg, D., Narahari, Y. & Gujar, S., 2008. Foundations of mechanism design: A tutorial Part 1-Key concepts and classical results. *Sadhana*, 33(2), pp.83–130.

- Geng, X., Huang, Y. & Whinston, A.B., 2003. Smart marketplaces: a step beyond Web services. *Information Systems and e-Business Management*, 1(1), pp.15–34.
- Gibbard, A., 1973. Manipulation of voting schemes: a general result. *Econometrica: Journal of the Econometric Society*, 41(4), pp.587–601.
- Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny,
 M., Moreau, L. & Myers, J., 2007. Examining the challenges of scientific workflows. *Computer*, 40(12), pp.26–34.
- Gimpel, H., Ludwig, H., Dan, A. & Kearney, B., 2003. PANDA: Specifying Policies for Automated Negotiations of Service Contracts. In M. E. Orlowska et al., eds. *Service-Oriented Computing - ICSOC 2003*. Springer Berlin / Heidelberg, pp. 287–302. Available at: http://dx.doi.org/10.1007/978-3-540-24593-3_20.
- Grimm, S., 2007. Discovery, Identifying Relevant Services. In R. Studer, S. Grimm, &A. Abecker, eds. Semantic web services: concepts, technologies, and applications.Springer Verlag.
- Groves, T., 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society*, 41(4), pp.617–631.
- Gunther, O., Tamm, G. & Leymann, F., 2007. Pricing web services. *International Journal* of Business Process Integration and Management, 2(2), pp.132–140.
- El Haddad, J., Manouvrier, M. & Rukoz, M., 2010. TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. *IEEE Transactions* on Services Computing, 3(1), pp.73–85.
- Haile, P.A., 2000. Partial pooling at the reserve price in auctions with resale opportunities. *Games and Economic Behavior*, 33(2), pp.231–248.
- He, Q., Yan, J., Jin, H. & Yang, Y., 2014. Quality-Aware Service Selection for Service-Based Systems Based on Iterative Multi-Attribute Combinatorial Auction. *IEEE Transactions on Software Engineering*, 40(2), pp.192–215.

- Herrmann, A., Huber, F. & Coulter, R.H., 1997. Product and service bundling decisions and their effects on purchase intention. *Pricing Strategy and Practice*, 5(3), pp.99– 107.
- Hilton, A.B.C. & Culver, T.B., 2000. Constraint Handling for Genetic Algorithms in Optimal Remediation Design. *Journal of Water Resources Planning and Management*, 126(3), pp.128–137. Available at: http://link.aip.org/link/?QWR/126/128/1.
- Huberman, B.A. & Clearwater, S.H., 1995. A Multi-Agent System for Controlling Building Environments. In International Conference on Multiagent Systems (ICMAS). pp. 171–176.
- Hudert, S., Ludwig, H. & Wirtz, G., 2009. Negotiating SLAs-An Approach for a Generic Negotiation Framework for WS-Agreement. *Journal of Grid Computing*, 7(2), pp.225–246. Available at: http://dx.doi.org/10.1007/s10723-009-9118-3.
- Jackson, M.O., 2003. Mechanism theory. In Ulrich Derigs, ed. *Optimization and Operations Research*. Oxford UK, p. 274, in the Encyclopedia of Life Support Systems. Available at: http://www.eolss.net.
- Jaeger, M.C. & Muehl, G., 2007. QoS-based Selection of Services: The Implementation of a Genetic Algorithm. In *Communication in Distributed Systems (KiVS)*, 2007 *ITG-GI Conference*. Bern, Switzerland: VDE, pp. 1–12.
- Jaeger, M.C., Rojec-Goldmann, G. & Muhl, G., 2004. QoS aggregation for Web service composition using workflow patterns. In *IEEE International Enterprise Distributed Object Computing Conference (EDOC)*. pp. 149–159.
- Jaeger, M.C., Rojec-Goldmann, G. & Muhl, G., 2005. QoS aggregation in Web service compositions. In *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '05)*. pp. 181–185.
- Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M.J. & Sierra, C., 2001. Automated negotiation: prospects, methods and challenges. *Group Decision* and Negotiation, 10(2), pp.199–215.

- Jiuxin, C., Yongsheng, L., Junzhou, L. & Bo, M., 2010. Efficient multi-QoS attributes negotiation for service composition in dynamically changeable environments. In *IEEE International Conference on Systems Man and Cybernetics (SMC)*. pp. 3118– 3124.
- Kagel, J.H. & Levin, D., 1993. Independent private value auctions: Bidder behaviour in first-, second- and third-price auctions with varying numbers of bidders. *The Economic Journal*, 103(419), pp.868–879.
- Karlin, A.R. & Peres, Y., 2014. *Game Theory, Alive*, Available at: http://homes.cs.washington.edu/~karlin/GameTheoryBook.pdf.
- Keeney, R.L. & Raïffa, H., 1993. Decisions with multiple objectives: preferences and value tradeoffs, Cambridge University Press. Available at: http://books.google.com.au/books?id=10Ea-BiARWUC.
- Kim, A., Kang, M., Meadows, C., Ioup, E. & Sample, J., 2009. A Framework for Automatic Web Service Composition, Washington, US: Technical Report, Naval Research Laboratory Center for High Assurance Computer Systems.
- Kim, J.B., Segev, A., Patankar, A. & Cho, M.G., 2003. Web services and bpel4ws for dynamic ebusiness negotiation processes. In *Conference on Web Services, ICWS*. pp. 111–117.
- Klemperer, P., 1999. Auction Theory: A Guide to the Literature. *Journal of Economic Surveys*, 13(3), pp.227–286. Available at: http://dx.doi.org/10.1111/1467-6419.00083.
- Korhonen, J., Pajunen, L. & Puustjarvi, J., 2003. Automatic composition of web service workflows using a semantic agent. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03)*. IEEE, pp. 566–569.
- Kruskal, W.H. & Wallis, W.A., 1952. Use of ranks in one-criterion variance analysis. Journal of the American statistical Association, 47(260), pp.583–621.

- Kutanoglu, E. & Wu, S.D., 1999. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Transactions*, 31(9), pp.813–826. Available at: http://dx.doi.org/10.1023/A:1007666414678.
- Lamparter, S., 2007. *Policy-based contracting in semantic web service markets*. PhD Dissertation, Karlsruhe Institute of Technology.
- Lamparter, S. & Bjorn Schnizler, 2006. Trading services in ontology-driven markets. *Proceedings of the 2006 ACM symposium on Applied computing*, pp.1679–1683.
- Lecue, F. & Mehandjiev, N., 2009. Towards Scalability of Quality Driven Semantic Web Service Composition. In *IEEE International Conference on Web Services (ICWS)*. pp. 469–476.
- Lee, J.-S. & Szymanski, B.K., 2007. Auctions as a dynamic pricing mechanism for eservices. In Service Enterprise Integration. Springer, pp. 131–156.
- Legner, C., 2009. Is There a Market for Web Services? In E. Di Nitto & M. Ripeanu, eds. Service-Oriented Computing - ICSOC 2007 Workshops. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 29–42.
- Leyton-Brown, K., Pearson, M. & Shoham, Y., 2000. Towards a universal test suite for combinatorial auction algorithms. *Proceedings of the 2nd ACM conference on Electronic commerce*, pp.66–76.
- Likhodedov, A. & Sandholm, T., 2003. Auction mechanism for optimally trading off revenue and efficiency. In *Proceedings of the 4th ACM conference on Electronic commerce - EC '03*. New York, USA: ACM Press, p. 212.
- Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao,
 J. & Zhao, Y., 2006. Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18(10), pp.1039–1065.
- M.B. Monteiro, 2008. Workflow for Protein Sequence Analysis. Available at: http://www.myexperiment.org/workflows/124.html [Accessed July 1, 2014].

- Ma, Y. & Zhang, C., 2008. Quick convergence of genetic algorithm for QoS-driven web service selection. *Computer Networks*, 52(5), pp.1093–1104. Available at: http://www.sciencedirect.com/science/article/pii/S1389128607003489.
- McAfee, R.P. & McMillan, J., 1987a. Auctions and Bidding. *Journal of economic literature*, 25(2), pp.699–738. Available at: http://www.jstor.org/stable/2726107.
- McAfee, R.P. & McMillan, J., 1987b. Auctions and Bidding. *Journal of Economic Literature*, 25(2), pp.699–738 CR American Economic Association. Available at: http://www.jstor.org/stable/2726107.
- McIlraith, S.A., Son, T.C. & Honglei, Z., 2001. Semantic Web services. *Intelligent Systems, IEEE*, 16(2), pp.46–53. Available at: 10.1109/5254.920599.
- McMillan, J., 1994. Selling spectrum rights. *The Journal of Economic Perspectives*, 8(3), pp.145–162.
- Medjahed, B. & Atif, Y., 2007. Context-based matching for Web service composition. *Distributed and Parallel Databases*, 21(1), pp.5–37. Available at: http://dx.doi.org/10.1007/s10619-006-7003-7.
- Medjahed, B. & Bouguettaya, A., 2005. A multilevel composability model for semantic web services. *IEEE Transactions on Knowledge and Data Engineering*, 17(7), pp.954–968.
- Menascé, D.A., Casalicchio, E. & Dubey, V., 2010. On optimal service selection in Service Oriented Architectures. *Performance Evaluation*, 67(8), pp.659–675.
- Michalewicz, Z., 1995. A survey of constraint handling techniques in evolutionary computation methods. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*. The MIT Press, Cambridge, Massachusetts, pp. 135–155.
- Michlmayr, A., Rosenberg, F., Leitner, P. & Dustdar, S., 2010. End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo. *IEEE Transactions on Services Computing*, 3(3), pp.193–205.

- Mihailescu, M. & Teo, Y.M., 2010. Combinatorial auction-based allocation of virtual machine instances in clouds. In 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid). IEEE, pp. 513–517.
- Milanovic, N. & Malek, M., 2004. Current solutions for web service composition. *Internet Computing, IEEE*, 8(6), pp.51–59. Available at: 10.1109/MIC.2004.58.
- Moghaddam, M., 2011. An Active Negotiation Model for Service Selection in Web Service Composition W. Zimmermann, ed., Halle, Germany: University Halle-Wittenberg, Institute of Computer Science.
- Moghaddam, M. & Davis, J., 2014. Service Selection in Web Service Composition: A Comparative Review of Existing Approaches. In A. Bouguettaya, Q. Z. Sheng, & F. Daniel, eds. *Web Services Foundations*. Springer New York, pp. 321–346. Available at: http://dx.doi.org/10.1007/978-1-4614-7518-7_13.
- Moghaddam, M., Davis, J. & Viglas, T., 2013. A Combinatorial Auction Model for Composite Service Selection Based on Preferences and Constraints. 10th IEEE International Conference on Services Computing (SCC 2013).
- Mohabey, M., Narahari, Y., Mallick, S., Suresh, P. & Subrahmanya, S. V, 2007a. A Combinatorial Procurement Auction for QoS-Aware Web Services Composition. In *IEEE International Conference on Automation Science and Engineering (CASE* 2007). pp. 716–721.
- Mohabey, M., Narahari, Y., Mallick, S., Suresh, P. & Subrahmanya, S. V, 2007b. An Intelligent Procurement Marketplace for Web Services Composition. In *IEEE/WIC/ACM International Conference on Web Intelligence*. pp. 551–554.
- Mueller, H.J., 1996. Negotiation principles. In G. M. P. O'Hare & N. R. Jennings, eds. Foundations of Distributed Artificial Intelligence. New York: John Wiley & Sons, Inc., pp. 211–229.
- Mukhija, A., Dingwall-Smith, A. & Rosenblum, D.S., 2007. QoS-Aware Service Composition in Dino. In *Fifth European Conference on Web Services*. pp. 3–12. Available at: 10.1109/ECOWS.2007.24.

- Mundt, J., 1993. Externalities: Uncalculated Outcomes of Exchange. *Journal of Macromarketing*, 13(2), pp.46–53.
- Myerson, R., 1979. Incentive compatibility and the bargaining problem. *Econometrica*, 47(1), pp.61–73.
- Myerson, R.B., 1981. Optimal auction design. *Mathematics of operations research*, 6(1), pp.58–73.
- Myerson, R.B. & Satterthwaite, M.A., 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2), pp.265–281.
- Néron, B., Ménager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P. & Letondal, C., 2009. Mobyle: a new full web bioinformatics framework. *Bioinformatics*, 25(22), pp.3005–3011. Available at: http://bioinformatics.oxfordjournals.org/content/25/22/3005.abstract.
- Nisan, N., 2000. Bidding and allocation in combinatorial auctions. In *Proceedings of the* 2nd ACM conference on Electronic commerce. Minneapolis, Minnesota, USA: ACM, pp. 1–12.
- Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V. V, 2007. *Algorithmic Game Theory*, Cambridge University Press.
- Di Nitto, E., Di Penta, M., Gambi, A., Ripa, G. & Villani, M., 2007. Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach. In B. Krämer, K.-J. Lin, & P. Narasimhan, eds. *Service-Oriented Computing – ICSOC* 2007. Springer Berlin / Heidelberg, pp. 295–306. Available at: http://dx.doi.org/10.1007/978-3-540-74974-5_24.
- O'Sullivan, J., Edmond, D. & ter Hofstede, A., 2002. What's in a Service? *Distributed and Parallel Databases*, 12(2), pp.117–133. Available at: http://dx.doi.org/10.1023/A:1016547000822.

- Oh, S.C., Lee, D. & Kumara, S.R., 2008. Effective Web Service Composition in Diverse and Large-Scale Service Networks. *IEEE Transactions on Services Computing*, 1(1), pp.15–32.
- Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A. & Li, P., 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17), pp.3045–3054. Available at: http://bioinformatics.oxfordjournals.org/content/20/17/3045.abstract.
- Olivares, M., Weintraub, G.Y., Epstein, R. & Yung, D., 2012. Combinatorial Auctions for Procurement: An Empirical Study of the Chilean School Meals Auction. *Management Science*, 58(8), pp.1458–1481. Available at: http://pubsonline.informs.org/doi/abs/10.1287/mnsc.1110.1496 [Accessed October 28, 2014].
- Omer, A.M. & Schill, A., 2009. Web service composition using input/output dependency matrix. In Proceedings of the 3rd workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing. London, United Kingdom: ACM, pp. 21–26.
- Padberg, M.W., 1999. *Linear Optimization and Extensions*, Springer. Available at: http://books.google.com.au/books?id=1L4YO8vQNCEC.
- Papazoglou, M.P., 2003. Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003)*. IEEE Comput. Soc, pp. 3–12.
- Papazoglou, M.P. & Van Den Heuvel, W.-J., 2006. Service-oriented design and development methodology. *International Journal of Web Engineering and Technology*, 2(4), pp.412–442.
- Pardalos, P., Migdalas, A. & Pitsoulis, L., 2008. Pareto optimality, game theory and equilibria, Springer.

- Parkes, D., 2006. Iterative combinatorial auctions. In P. C. Cramton, Y. Shoham, & R. Steinberg, eds. *Combinatorial auctions*. Cambridge, Mass.; London: MIT Press, pp. 41–77.
- Parkes, D.C., 2001. Iterative combinatorial auctions: Achieving economic and computational efficiency. PhD Thesis, University of Pennsylvania. Available at: http://www.eecs.harvard.edu/~parkes/diss.html.
- Parra-Hernandez, R. & Dimopoulos, N.J., 2005. A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(5), pp.708–717.
- Parsons, S., Rodriguez-Aguilar, J.A. & Klein, M., 2011. Auctions and bidding: A guide for computer scientists. ACM Comput. Surv., 43(2), pp.1–59.
- Perepletchikov, M., Ryan, C. & Frampton, K., 2007. Cohesion Metrics for Predicting Maintainability of Service-Oriented Software. In *Seventh International Conference* on Quality Software QSIC '07. pp. 328–335.
- Petrie, C. & Bussler, C., 2008. The Myth of Open Web Services: The Rise of the Service Parks. *IEEE Internet Computing*, 12(3), pp.95–96.
- Pinedo, M., 1995. Scheduling: Theory, algorithms, and systems, Prentice Hall (Englewood Cliffs, NJ).
- Prashanth, B. & Narahari, Y., 2008. Efficient algorithms for combinatorial auctions with volume discounts arising in web service composition. In 2008 IEEE International Conference on Automation Science and Engineering. IEEE, pp. 995–1000.
- Rasmusen, E., 2001. *Games and Information: An Introduction to Game Theory*, Blackwell. Available at: http://books.google.com.au/books?id=7ylayBG9sa4C.
- Rassenti, S.J., Smith, V.L. & Bulfin, R.L., 1982. A Combinatorial Auction Mechanism for Airport Time Slot Allocation. *The Bell Journal of Economics*, 13(2), pp.402– 417. Available at: http://www.jstor.org/stable/3003463?origin=crossref.

- Richter, J., Baruwal Chhetri, M., Kowalczyk, R. & Bao Vo, Q., 2012. Establishing composite SLAs through concurrent QoS negotiation with surplus redistribution. *Concurrency and Computation: Practice and Experience*, 24(9), pp.938–955.
- Richter, J., Chhetri, M.B., Kowalczyk, R., Bao Quoc, V., Talib, M.A. & Colman, A., 2010. Utility Decomposition and Surplus Redistribution in Composite SLA Negotiation. In *IEEE International Conference on Services Computing (SCC)*. pp. 627–630.
- Richter, J., Klusch, M. & Kowalczyk, R., 2010. On monotonic mixed tactics and strategies for multi-issue negotiation. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, pp.1609–1610.
- Rosenberg, F., Celikovic, P., Michlmayr, A., Leitner, P. & Dustdar, S., 2009. An End-to-End Approach for QoS-Aware Service Composition. In *IEEE International Enterprise Distributed Object Computing Conference (EDOC '09)*. pp. 151–160.
- Ross, M., 2013. Real time bidding for dummies. B&T. Available at: http://www.bandt.com.au/features/real-time-bidding-for-dummies [Accessed March 12, 2014].
- Rothkopf, M.H., 2007. Thirteen reasons why the Vickrey-Clarke-Groves process is not practical. *Operations Research*, 55(2), pp.191–197.
- Rothkopf, M.H., Teisberg, T.J. & Kahn, E.P., 1990. Why Are Vickrey Auctions Rare? *Journal of Political Economy*, 98(1), pp.94–109. Available at: http://www.jstor.org/stable/2937643.
- Sairamesh, J., Ferguson, D.F. & Yemini, Y., 1995. An approach to pricing, optimal allocation and quality of service provisioning in high-speed packet networks. In Proceedings of IEEE Computer and Communications Societies. Bringing Information to People INFOCOM'95. pp. 1111 – 1119.
- Sandholm, T., 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1), pp.1–54.

- Sandholm, T., Suri, S., Gilpin, A. & Levine, D., 2005. CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *Management Science*, 51(3), pp.374–390.
- Sandholm, T., Suri, S., Gilpin, A. & Levine, D., 2002. Winner determination in combinatorial auction generalizations. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. Bologna, Italy: ACM, pp. 69–76.
- Schnizler, B., Neumann, D., Veit, D. & Weinhardt, C., 2005. A multiattribute combinatorial exchange for trading grid resources. In *Proceedings of the Research Symposium on Emerging Electronic Markets (RSEEM)*. Amsterdam, Netherlands.
- Schulte, S., 2010. Web Service Discovery Based on Semantic Information-Query Formulation and Adaptive Matchmaking. PhD Dissertation, Technische Universität Darmstadt.
- Schwind, M., 2007. Dynamic Pricing and Automated Resource Allocation for Complex Information Services, Springer.
- Sheffi, Y., 2004. Combinatorial auctions in the procurement of transportation services. *Interfaces*, 34(4), pp.245–252.
- Shoham, Y. & Leyton-Brown, K., 2009. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*, Cambridge University Press.
- Sierra, C., de Màntaras, R.L. & Busquets, D., 2000. Multiagent Bidding Mechanisms for Robot Qualitative Navigation. In *Lecture Notes in Artificial Intelligence*. pp. 198– 212. Available at: http://dx.doi.org/10.1007/3-540-44631-1_14.
- Smith, A., 1904. An Inquiry into the Nature and Causes of the Wealth of Nations Edwin Cann, ed., London: Methuen and Co., Ltd. Available at: http://econlib.org/library/Topics/Details/invisiblehand.html.

- Srivastava, N.K., Viswanadham, N. & Kameshwaran, S., 2008. Procurement of global logistics services using combinatorial auctions. In *IEEE International Conference* on Automation Science and Engineering. IEEE, pp. 297–302.
- Stein, A., Hawking, P. & Wyld, D.C., 2003. The 20% solution?: a case study on the efficacy of reverse auctions. *Management Research News*, 26(5), pp.1–20.
- Stevens, W.P., Myers, G.J. & Constantine, L.L., 1974. Structured design. IBM Systems Journal, 13(2), pp.115–139.
- Strauss, J., Frost, R. & Ansary, A.I., 2009. Price: The Online Value. In *E-marketing*. Pearson Prentice Hall.
- Tan, W. & Zhou, M., 2013. Scientific Workflows Enabling Web-Scale Collaboration. In Business and Scientific Workflows. John Wiley & Sons, Inc., pp. 161–188. Available at: http://dx.doi.org/10.1002/9781118554609.ch7.
- Tang, Q., 2004. Economics of web service provisioning: optimal market structure and intermediary strategies. Doctoral Dissertation, University of Florida.
- Thakkar, S., Ambite, L. & Knoblock, A., 2005. Composing, optimizing, and executing plans for bioinformatics web services. *The VLDB Journal*, 14(3), pp.330–353.
- The Royal Swedish Academy of Sciences, 2007. The Prize in Economic Sciences 2007 -Press Release. Available at: http://www.nobelprize.org/nobel_prizes/economicsciences/laureates/2007/press.html [Accessed March 20, 2014].
- UDDI Consortium, 2001. UDDI Executive White Paper. Available at: www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf.
- Ul Haq, I., Paschke, A., Schikuta, E. & Boley, H., 2010. Rule-based validation of SLA choreographies. *The Journal of Supercomputing*, pp.1–22. Available at: http://dx.doi.org/10.1007/s11227-010-0492-1.
- Varian, H.R., 2007. Position auctions. *International Journal of Industrial Organization*, 25(6), pp.1163–1178.

- Verma, K., Akkiraju, R., Goodwin, R., Doshi, P. & Lee, J., 2004. On accommodating inter service dependencies in web process flow composition. In AAAI Spring Symposium (on Semantic Web Services). pp. 37–43.
- Verma, K., Gomadam, K., Sheth, A., Miller, J. & Wu, Z., 2005. The METEOR-S Approach for Configuring and Executing Dynamic Web Processes, Technical Report (TR6-24-05).
- Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1), pp.8–37.
- Vilenica, A., Hamann, K., Lamersdorf, W., Sudeikat, J. & Renz, W., 2011. An Extensible Framework for Dynamic Market-Based Service Selection and Business Process Execution Distributed Applications and Interoperable Systems. In P. Felber & R. Rouvoy, eds. *Distributed Applications and Interoperable Systems SE - 12*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 150–164.
- De Vries, S. & Vohra, R. V, 2003. Combinatorial Auctions: A Survey. *INFORMS Journal* on Computing, 15(3), p.284.
- W3C Working Group Note 11, 2004. Web Services Glossary. *The World Wide Web Consortium (W3C)*. Available at: http://www.w3.org/TR/ws-gloss/ [Accessed April 30, 2014].
- Wang, P., 2009. QoS-aware web services selection with intuitionistic fuzzy set under consumer's vague perception. *Expert Systems with Applications*, 36(3, Part 1), pp.4460–4466.
- Wang, R., 1993. Auctions versus Posted-Price Selling. *The American Economic Review*, 83(4), pp.838–851. Available at: http://www.jstor.org/stable/2117581.
- Wassermann, B. & Emmerich, W., 2011. Monere: Monitoring of Service Compositions for Failure Diagnosis. In G. Kappel, Z. Maamar, & H. Motahari-Nezhad, eds. *Service-Oriented Computing (Lecture Notes in Computer Science)*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 344–358.

- Watanabe, A., Ishikawa, F., Fukazawa, Y. & Honiden, S., 2012. Web Service Selection Algorithm Using Vickrey Auction. In *IEEE 19th International Conference on Web Services*. IEEE, pp. 336–342.
- Weide, K., 2013. Real-time bidding in the United Staes and Worldwide, 2010-2017. Available at: http://www.pubmatic.com/reports/IDC-RTB-2013.pdf [Accessed March 13, 2014].
- Weinhardt, C., Blau, B., Conte, T., Filipova-Neumann, L., Meinl, T. & Michalk, W., 2011a. Business Aspects of Web Services, Berlin, Heidelberg: Springer Berlin Heidelberg.
- Weinhardt, C., Blau, B., Conte, T., Filipova-Neumann, L., Meinl, T. & Michalk, W., 2011b. The Vision of Web Service Markets. In *Business Aspects of Web Services*. Springer, pp. 191–194.
- Wiesemann, W., Hochreiter, R. & Kuhn, D., 2008. A Stochastic Programming Approach for QoS-Aware Service Composition. In *IEEE International Symposium on Cluster Computing and the Grid (CCGRID '08)*. pp. 226–233. Available at: 10.1109/CCGRID.2008.40.
- Wikipedia, 2014. Wikipedia, the free encyclopedia. Available at: http://en.wikipedia.org/wiki/Impression_(online_media) [Accessed August 8, 2014].
- Wilkes, J., 2009. Utility Functions, Prices, and Negotiation. In R. B. and K. Bubendorfer, ed. *Market-Oriented Grid and Utility Computing*. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- Winkler, M., Springer, T. & Schill, A., 2010. Automating Composite SLA Management Tasks by Exploiting Service Dependency Information. In *Proceedings of the 2010 Eighth IEEE European Conference on Web Services*. IEEE Computer Society, pp. 59–66.
- Wurman, P.R., 2001. Dynamic pricing in the virtual marketplace. *IEEE Internet Computing*, 5(2), pp.36–42.

- Wurman, P.R., Walsh, W.E. & Wellman, M.P., 1998. Flexible double auctions for electronic commerce: theory and implementation. *Decision Support Systems*, 24(1), pp.17–27.
- Yan, J., Kowalczyk, R., Lin, J., Chhetri, M.B., Goh, S.K. & Zhang, J., 2007. Autonomous service level agreement negotiation for service composition provision. *Future Generation Computer Systems*, 23(6), pp.748–759.
- Yan, J., Zhang, J., Lin, J., Chhetri, M.B., Goh, S.K. & Kowalczyk, R., 2006. Towards Autonomous Service Level Agreement Negotiation for Adaptive Service Composition. In 10th International Conference on Computer Supported Cooperative Work in Design (CSCWD '06). pp. 1–6.
- Yang, J. & Papazoglou, M.P., 2002. Web Component: A Substrate for Web Service Reuse and Composition. Proceedings of the 14th International Conference on Advanced Information Systems Engineering, pp.21–36.
- Yarom, I., Topol, Z. & Rosenschein, J.S., 2004. Decentralized Marketplaces and Web Services: Challenges and Opportunities. In Workshop on Agent Mediated Electronic Commerce VI.
- Yourdon, E. & Constantine, L.L., 1979. *Structured design: fundamentals of a discipline* of computer program and systems design, Prentice Hall. Available at: http://books.google.com.au/books?id=zMQmAAAAMAAJ.
- Yu, Q. & Bouguettaya, A., 2008. Framework for web service query algebra and optimization. *ACM Transactions on the Web (TWEB)*, 2(1), pp.6:1–6:35.
- Yu, Q. & Bouguettaya, A., 2012. Multi-attribute optimization in service selection. World Wide Web, 15(1), pp.1–31. Available at: http://dx.doi.org/10.1007/s11280-011-0121-9.
- Yu, T., Zhang, Y. & Lin, K.J., 2007. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on the Web (TWEB), 1(1), pp.6:1– 6:26.

- Zadeh, L.A., 1971. Similarity relations and fuzzy orderings. *Information Sciences*, 3(2), pp.177–200.
- Zaman, S. & Grosu, D., 2013. Combinatorial auction-based allocation of virtual machine instances in clouds. *Journal of Parallel and Distributed Computing*, 73(4), pp.495– 508.
- Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J. & Sheng, Q.Z., 2003. Quality driven web services composition. *Proceedings of the 12th international conference on World Wide Web*, pp.411–421.
- Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J. & Chang, H., 2004. QoS-Aware Middleware for Web Services Composition. *IEEE Trans. Softw. Eng.*, 30(5), pp.311–327.
- Zulkernine, F., Martin, P., Craddock, C. & Wilson, K., 2009. A Policy-Based Middleware for Web Services SLA Negotiation. In *IEEE International Conference on Web Services ICWS*. pp. 1043–1050.
- Zulkernine, F.H. & Martin, P., 2011. An Adaptive and Intelligent SLA Negotiation System for Web Services. *IEEE Transactions on Services Computing*, 4(1), pp.31 – 43.