

**SERVICE SELECTION IN A MARKETPLACE:
A MULTI-PERSPECTIVE SOLUTION**

by

Dipak Pudasaini

M.Sc. in Computer Science and IT, Tribhuvan University, Nepal, 2005

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2017

©Dipak Pudasaini 2017

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Dipak Pudasaini

SERVICE SELECTION IN A MARKETPLACE: A MULTI-PERSPECTIVE SOLUTION

Dipak Pudasaini
Master of Science, Computer Science, 2017
Ryerson University

ABSTRACT

Most of the current research work on web service selection only considered the selection problem from the perspective of one party – service consumers. A service marketplace serves many parties including service consumers and providers. Thus, it is important to consider multiple parties. In this thesis, we propose a service selection model considering the benefits of multiple parties: consumers, providers and the marketplace. The model ranks services based on not only how much these services satisfy the user requirements but also how much the requests can be distributed to different providers and the revenue gain in the marketplace. We design different objective functions, then combine into a QoS-Plus-PF objective function. The results show that proposed model could achieve a high degree of satisfaction of user requests (i.e., 0.61% to 5.26% worse than the optimal score), and meanwhile have the capability of promoting more diversified set of services (i.e., 48.95% promotion percentage).

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Dr. Cherie Ding for her valuable support and guidance in helping me to go through all the difficulties in my work. Her precious suggestions and guidance have greatly enhanced my knowledge and skills in research and have significantly contributed to the completion of this thesis.

In addition, I would like to thank Dr. Kosta Derpanis, Dr. Alireza Sadeghian, and Dr. Isaac Woungang, who have reviewed my thesis and have given me valuable suggestions which have enabled me to improve my thesis. Also, I would like to acknowledge the support of the Computer Science Department of Ryerson University and my fellow students.

Finally, I would like to express my deep appreciation to my family, relatives, and friends who have motivated and supported me during these years of study.

TABLE OF CONTENTS

AUTHOR’S DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
LIST OF APPENDICES.....	ix
LIST OF ACRONYMS.....	x
CHAPTER 1.....	1
INTRODUCTION.....	1
1.1. Background and the Problem Statement.....	1
1.1.1. Background.....	1
1.1.2. Problem Statement.....	4
1.2. Motivation and Objective.....	6
1.3. Proposed Methodology.....	8
1.4. Outline of Thesis.....	10
CHAPTER 2.....	11
RELATED WORKS.....	11
2.1. Introduction.....	11
2.2. QoS Based Web Service Selection Models.....	11
2.3. Marketplace, Brokerage and Bidding models.....	17
CHAPTER 3.....	22
METHODOLOGY.....	22
3.1. Introduction.....	22
3.2. A Motivating Example.....	22

3.3. QoS Attributes	24
3.4. Architecture of Service Marketplace	25
3.5. Selection Algorithm	28
3.5.1. Overview	28
3.5.2. Selection Process	29
3.5.3. Selection Approach by QoS-only and its Objective Function.....	33
3.5.4. QoS-Plus-FP Objective Function	34
3.5.4. Illustrating the Ranking Model by an Examlpe	38
3.6. Summary	43
CHAPTER 4	44
EXPERIMENTS	44
4.1. Experiment Design, Simulated Dataset and Implementation	44
4.2. Results and Analyses.....	48
4.2.1. Analysis of Promotion of Paid Services and Providers.....	49
4.2.2. Analysis of Rank Change of Free Services	54
4.2.3. Analysis of Impact on Free and Paid Providers and Users	56
4.2.4. Analysis of Fairness and Promotion Score.....	58
4.3. Summary	59
CHAPTER 5	61
CONCLUSIONS AND FUTURE WORKS.....	61
5.1. Conclusions	61
5.2. Future Works	62
APPENDICES	63
REFERENCES	70

LIST OF TABLES

Table 3.1- Ranking of services by QoS-only.....	23
Table 3.2- Ranking list of the services by QoS-only objective function	39
Table 3.3- List of the services in the candidate list.....	39
Table 3.4- Normalized score difference between the optimal and current service.....	40
Table 3.5- Initialized fairness and promotion score.....	40
Table 3.6- Fairness score and promotion score to promote S4.....	41
Table 3.7- Fairness score and promotion score to promote S5.....	41
Table 3.8- Fairness score and promotion score to promote S9.....	42
Table 3.9- Ranking of services by QoS-Plus-FP objective function	43
Table 4.1- A portion of QWS dataset	45
Table 4.2- Request dataset	46
Table 4.3- Ratio between free and paid users	47
Table 4.4- User dataset	47
Table 4.5- Relation between free and paid providers	47
Table 4.6- Provider dataset	48

LIST OF FIGURES

Figure 1.1- Web service model.....	3
Figure 3.1- Architecture of service marketplace.....	26
Figure 3.2- Selection and ranking process	31
Figure 4.1- Percentage of service promotion when the free/paid user ratio changes (free/paid provider ratio: 50:50)	49
Figure 4.2- Percentage of service promotion when all ratios of providers	50
Figure 4.3- Percentage of unique service promotion when the free/paid user ratio changes (free/paid provider ratio: 50:50)	51
Figure 4.4- Percentage of unique service promotion when all ratios of providers	52
Figure 4.5- Percentage of unique provider promotion when the free/paid user ratio changes (free/paid provider ratio: 50:50)	53
Figure 4.6- Percentage of unique provider promotion when all ratios of providers	54
Figure 4.7- Percentage of rank change of free services when the free/paid user ratio changes (free/paid provider ratio: 50:50)	55
Figure 4.8- Percentage of rank change of free services when all ratios of providers	56
Figure 4.9- Percentage of QoS difference between the optimal and promoted services	57
Figure 4.10- Average fairness score.....	58
Figure 4.11- Average promotion score.....	58

LIST OF APPENDICES

Table A.1- Measurement of QoS-only and our Algorithm on free/paid provider 10:90.....	63
Table A.2- Measurement of QoS-only and our Algorithm on free/paid provider 30:70.....	63
Table A.3- Measurement of QoS-only and our Algorithm on free/paid provider 50:50.....	64
Table A.4- Measurement of QoS-only and our Algorithm on free/paid provider 70:30.....	64
Table A.5- Measurement of QoS-only and our Algorithm on free/paid provider 90:10.....	65
Table A.6- Promotion of paid services and providers on free/paid provider ratio 10:90.....	65
Table A.7- Promotion of paid services and providers on free/paid provider ratio 30:70.....	65
Table A.8- Promotion of paid services and providers on free/paid provider ratio 50:50.....	66
Table A.9- Promotion of paid services and providers on free/paid provider ratio 70:30.....	66
Table A.10- Promotion of paid services and providers on free/paid provider ratio 90:10.....	66
Table A.11- Rank change of free services on free/paid provider ratio 10:90	67
Table A.12- Rank change of free services on free/paid provider ratio 30:70	67
Table A.13- Rank change of free services on free/paid provider ratio 50:50	67
Table A.14- Rank change of free services on free/paid provider ratio 70:30	67
Table A.15- Rank change of free services on free/paid provider ratio 90:10	68
Table A.16- Percentage of score difference between optimal and promoted services.....	68
Table A.17- Average fairness and promotion score	68

LIST OF ACRONYMS

AHP: Analytical Hierarchy Process

CMDP: Constrained Markov Decision Process

CP: Constraint Programming

CSB: Cloud Service Brokerage

CSP: Constraint Satisfaction Problem

HTTP: Hyper Text Transfer Protocol

MCDM: Multi Criteria Decision Making

MIP: Mixed Integer Programming

QoS: Quality of Service

REST: Representation State Transfer

SLA: Service Level Agreement

SMTP: Simple Mail Transfer Protocol

SOAP: Simple Object Access Protocol

UDDI: Universal Description Discovery and Integration

URI: Uniform Resource Identifier

VCB: Virtual Cloud Bank

WSDL: Web Service Description Language

WSM: Web Service Model

WSRF: Web Service Relevancy Function

XML: Extended Markup Language

CHAPTER 1

INTRODUCTION

1.1 Background and the Problem Statement

1.1.1 Background

Web service is a programmable module that provides universal accessibility through communication protocols. Web services can be implemented in various programming languages and run on various platforms. They are available over the internet and use standard Extended Markup Language (XML). Web service can communicate with another application over the network by using standard technologies: Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description Discovery and Integration (UDDI).

WSDL is an XML based language that describes the functionalities of web services. The purpose of defining WSDL is to create public interface of a web service. WSDL documents define the message formats and protocol bindings to interact with web services. When interfaces are set up, communication with other services or software occurs through SOAP.

SOAP is a simple and lightweight protocol to transfer structured-type information among web services. It is used to access web services through the internet. It binds with existing protocols such as Hyper Text Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP). Universal Description Discovery and Integration (UDDI) is a specification that provides a Meta service for publishing and locating web services by enabling robust queries against Meta data. UDDI helps for service request such as who, what, where, and how. A combination of WSDL, SOAP and UDDI helps the process of locating and running a web service.

Besides the SOAP-based web services, web services based on the Representation State Transfer (REST) protocol are called RESTful services and follow the REST architecture. Resource is the basic unit in this architecture and RESTful web services are highly scalable, maintainable and light-weight. It does not contain message layer and focuses on design rules for creating stateless services. The unique Uniform Resource Identifier (URI) is used to access resource by client. Some standard operations are used to transfer state for each new resource such as GET, PUT, DELETE, POST etc.

Cloud is the best platform for hosting and running web services. It is a type of computing platform for sharing resources [1]. Nowadays, most of the companies offer their services through the cloud platform such that resource utilization can be minimized. It maximizes benefits for both customers and providers.

A marketplace is a kind of business site where different parties are integrated and communicated for business operations. It consists of providers (or sellers), the products or services offered by these providers, consumers (or buyers), broker (or the marketplace), infrastructure and other services supporting the running of the marketplace. Examples of some e-commerce marketplaces are Amazon, eBay and Alibaba. The main functions of such marketplace are matching buyers and sellers, exchanging goods, services and payments associated with market transactions, and providing the infrastructure such as legal and regulatory framework for efficient functions of the market [2]. The consumers or users request for services in a marketplace according to their requirements. The providers offer their services in a marketplace according to their products. The matchmaking broker is an intermediary between consumers and providers that helps consumers to choose the best service or product according to

their requirements and help providers to provide their services according to the needs of the users [2]. This matchmaking broker is simply called broker in the rest of the thesis.

The Web Service Model (WSM) is divided into three components: service registry, service provider and service consumer. Figure 1.1 shows a traditional web service model.

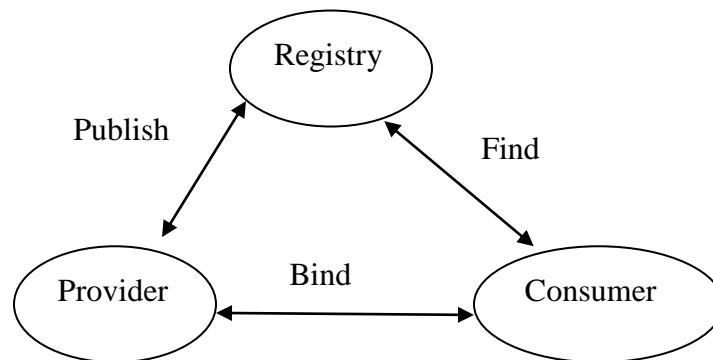


Figure 1.1: Web service model [3]

The selection of web services generally follows two steps. The first step is matching the services based on functional requirements and the second step is ranking the services based on the non-functional requirements. The web services with similar functionalities are measured and ranked based on values of their non-functional properties. The functional requirements are easy to determine by using common tools; however non-functional requirements of web service such as performance, execution time are difficult to analyze and predict [4]. Nowadays, a number of web services with similar functionalities are increasingly available on the internet such that it becomes difficult to select the appropriate web service to fulfill the user requirements [5].

Quality of Service (QoS) represents an important part of non-functional service properties, which usually include various performance-related measurements such as response time, reliability, availability, and throughput [6]. The web service providers are trying to provide services with high QoS values [5]. By doing so, providers can also achieve their goals of gaining the highest profit by having more customers.

In a typical QoS-based service selection process, first a user inputs the QoS requirements into the system. Then, these requirements are processed and a ranking score is calculated for each service by using a utility function. Finally, a list of ranked services is returned from the system. The overall performance of the service selection system depends upon the type of utility function used.

1.1.2 Problem Statement

A service marketplace is a place in which services from different providers are published and become available to subscribers or users of the marketplace [7]. A number of web services that satisfy similar functionalities are increasingly available on the internet, even in the same marketplace. In such scenario, users may find it difficult to choose the best service to fulfill their requirements, and providers may also find it difficult to compete with other similar services published in the same marketplace to attract more customers and gain more profits. Therefore, how to rank services for the benefits of both users and providers is becoming a challenging task. The service selection algorithm in the marketplace should be a balanced solution. It should be a multi-perspective solution considering the benefits for multiple parties.

Most of the current research on the service selection algorithm in a marketplace or in general only considers one party, which is from the service consumer's perspective on how to find a service that best matches consumer's functional and non-functional requirements. Some research related to scheduling or resource optimization may be considered from the provider's perspective. From the perspective of a service marketplace, both service consumers and service providers are considered as its customers. And thus the benefits of both consumers and providers

should be taken care of. However, in the current research work, they are not treated equally when selecting web services in the marketplace. Oftentimes, the providers' benefits are ignored.

There are various types of profit models that can be used in the marketplace [8] [9]. Some models charge a fee to users or providers or both. Some models are profit sharing models in which profits are shared in the marketplace. Some models are advertising models that promote services through advertisements.

In this work, we propose a marketplace design that considers the profit model and a selection algorithm that considers the benefits of service consumers, service providers, as well as the marketplace itself. In the marketplace, from the consumer's perspective, the requirements of service selection should be satisfied and optimal services should be recommended. On the other hand, from the provider's perspective, services of all providers should at least have the chances to be considered, otherwise, they may eventually leave the marketplace. Therefore, the main challenge is formulating a service selection algorithm which considers the benefits of both parties: consumers and providers. By providing such a fair and competitive environment, the marketplace can retain its current customers and attract new customers. In the mean-time, the marketplace also has an opportunity to maximize its own profit. To the best of our knowledge, such scenarios haven't been considered in the current research work on service selection. It is our purpose to address these issues in this work for the satisfaction of users and providers in order to gain more benefits for the marketplace.

1.2 Motivation and Objective

The existing service selection algorithms are usually based on only the QoS values of the web services. The service with the highest-ranking score always takes in the first position in the result list.

The main motivation of our work is that, if the selection algorithm only considers QoS values for the selection and ranking of web services, the ranking order will be the same for the same type of requests that are on the same QoS attributes. Therefore, the lower-ranked services will never have a chance to be selected because usually users only look at the first K services. K is usually small, e.g. any number between 1 and 10. However, sometimes, the difference between services' overall QoS values may not be very large even though the difference between their ranking orders could be large. For instance, all the services in the top 30 positions have very similar QoS values. The difference between the QoS score of the first and the 30th service is less than 5% and they all satisfy a particular QoS requirement. However, the services ranked below the tenth position (11th to 30th) may not have an opportunity to be selected because users will not check them at all.

Another motivation is that we want to come up with a selection algorithm which could generate revenue for the marketplace. Some users may not mind paying a fee in order to guarantee that the best service (in terms of their QoS values) is always recommended by the selection algorithm. Others may not want to pay the fee, but do not mind if the selection algorithm promotes services that are not originally in the top positions as long as their QoS values are good enough to satisfy all their requirements. Some providers may not mind to pay a fee in order to have their services promoted so that their not-so-strongly-ranked services could get a chance to be used by the consumers. Others may not want to pay a fee, but do not mind

their services might be demoted if the marketplace wants to promote other services, as long as it is not happening all the time and their services are still treated fairly in terms of the rank calculation. The revenue could also be generated through advertisements on a search result page. Most of the existing web service selection algorithms did not consider these scenarios at all and did not have a revenue generation model for the marketplace.

With these two motivations, the main objective of our work is to provide a ranking mechanism that can be used in a service marketplace and considers the benefits of all three parties: consumers, providers and the marketplace.

Consumers' benefits will be considered because only services that satisfy their requirements and have high overall QoS values will be ranked in high positions. Providers' benefits will be considered because their services will have a chance to be promoted to higher positions as long as their overall QoS values are in the acceptable range. The benefit of the marketplace will be considered because more consumers and providers can be attracted and retained in the marketplace and the whole marketplace will be healthier due to multiple alternative available services and providers instead of a few dominating services and providers.

This approach provides the satisfaction to consumers, providers and the marketplace for the selection and ranking of web services; however, in some cases this ranking mechanism may have the following limitations.

- A service which is the best in terms of its QoS values may not be used as much as it is supposed to be when a normal ranking algorithm is used.
- A user may not have the best service recommended for some requests.
- In real scenarios, most of the users may not want to pay a fee so that the implementation could be difficult.

This approach provides a good platform for consumers, providers and the marketplace considering their respective benefits. The marketplace offers a healthy competition mechanism because the existence of multiple competing providers pushes for better services, best services have the highest chance to be selected, good-but-not-the-best services have more opportunities to be used and even poorly performing services have a chance to improve their quality. When multiple providers are attracted and remain in the marketplace, users will also be attracted and stay because they will have more options. As long as all their requirements are satisfied, users usually do not care about whether the recommended service is the top one service. With more providers and users, with the implementation of different profit models, the marketplace could maximize its profit.

1.3 Proposed Methodology

Nowadays, some large companies such as Amazon, Google, etc. have special sites for registered users and general sites for common users. Some companies offer free access to their services for the new customers for a certain period of time and then they must pay a fee to use the services. The market is more competitive. Some users are willing to pay for the use of their services and others are not. Some providers are willing to pay incentives for the promotion of their services and others are not. The marketplace wants to maximize their benefits by interacting with as many providers and users as possible. This research aims to cover all these scenarios.

The main purpose of this work is to develop a service selection algorithm that can be used by the service marketplace, and provides the benefits for consumers, providers and the marketplace. In order to achieve these goals, we consider a marketplace that includes both free and paid users and free and paid providers. For paid providers, if their services satisfy the

minimum user requirements, they will have the opportunity to be promoted. For free providers, their services might be demoted due to the promotion of other services but a certain level of fairness will be maintained. For paid users, the selection algorithm will only consider the user requirements and services are ranked based on QoS values only. For free users, the selection algorithm will consider user requirements as well as the provider's benefit, and some services from the paid providers may be promoted.

The objective function that calculates the overall score of each web service by using only the QoS values are called QoS-only objective function. Most of the previous work used QoS-only objective function to rank the services. In this work, besides the QoS values, we also add the fairness score and the promotion score into the objective function, which we can call QoS-Plus-FP objective function. Fairness score is to measure how fair our selection algorithm treats the free services by counting how many times top-ranked free service is demoted. The free service is only allowed to be demoted to a lower position if the fairness measurement is in an acceptable range. Promotion score is to measure how much promotion power our selection algorithm gives to the paid services by counting how many times the paid service is promoted. The final QoS-Plus-FP objective function we use in our selection system is the combination of the QoS score of the service (i.e. the difference between the optimal score and the score of the service based on the QoS-only objective function), the promotion score of the current service (if it is a paid service), the average fairness score of all free services and the average promotion score of all paid services.

In this proposed selection system, first the minimum requirements of the users are input to the selection system. Then QoS-based objective function ranks the services, and lists the services satisfying the user's requirements. If a paid user requests for a service, this ranking list

is the final list of services. If a free user requests for a service and the first service ranked by QoS-only objective function is from a paid provider, again, this ranking list is the final list of services. If a free user requests for a service and the first service ranked by QoS-only objective function is from a free provider, then for the paid services whose QoS values satisfy the user requirement and are considered in an acceptable range, we calculate their scores using the QoS-Plus-FP objective function. These services are then ranked based on these QoS-Plus-FP objective scores. Then, the re-ranked list of services will be returned to the user. There will also be advertisements on the search result page so that providers may choose to pay a fee to promote their services. It is up to the users whether they want to select the ads.

1.4 Outline of thesis

The remaining part of the thesis is organized as follows:

Chapter 2 briefly reviews different QoS-based service selection models and auction models for web service selection. Then, it reviews works closely related to our work on the satisfaction of users, providers and brokers for the selection of web services.

Chapter 3 describes the architecture model of our system, and the selection and ranking process. Then, it explains the QoS-only selection approach and its objective function, and our selection approach using QoS-Plus-FP objective function.

Chapter 4 defines the experiment used to evaluate our proposed system. It describes the details about dataset generation, design and implementation, and result and analysis.

Finally, in Chapter 5, we conclude our thesis with a summary and provide directions for our future work

CHAPTER 2

RELATED WORKS

2.1 Introduction

Web services are the fundamental parts of any organizations nowadays and they allow various applications to share data and services, and communicate among them. They use standardized protocols to communicate and provide cost effective solutions on the Internet. They are becoming more powerful for implementing business applications and provide appropriate solutions to the customers.

In this chapter, we will review some research related to our work including QoS based web service selection and ranking, marketplace models, brokerage models and auction models.

2.2 QoS Based Web Service Selection Models

QoS-based web service selection and ranking is the most important method for web service users to choose the required service among a large number of similar types of services. There are various approaches for the selection and ranking of web services such as vector based approach, utility based approach, constraint programming approach, Multi Criteria Decision Making (MCDM) approach, skyline approach etc.

Yan and Piao [10] described a vector-based model for representing service consumer's QoS requirements and service provider's QoS advertisement. They also developed an algorithm for matching and ranking non-functional requirements for consumers. The distance between the requested vector and published vector was calculated to prepare the final ranking list.

AI-Masri and Mahmoud [4] developed a web service ranking mechanism. This method used Web Service Relevancy Function (WSRF) to measure the ranking of web services. When the clients submitted their requests, WSRF values were calculated for matching the services. In this method, first, the QoS values of services were arranged in a matrix form, then these values were normalized, and finally, the values in each row were used to calculate the WSRF scores for each web service. They concluded that the higher WSRF values were more relevant and desirable than lower values.

Liu et al. [11] extended the QoS computational model for web service selection and implemented a QoS registry in hypothetical phone service provisioning marketplace. This work mainly focused on generic quality criteria: execution price, execution duration, and reputation, and business criteria: transaction, compensation rate and penalty rate. The QoS value of each web service was calculated by normalization methods. A matrix was used to represent QoS values of offered web services and QoS criteria in the service request. The first normalization was used to provide uniform index to represent service quantities of the providers and second normalization was used to represent uniform quality criteria and set threshold to a group of quality criteria. QoS registry used this normalized score to rank the services. This method provided more flexibility for Service providers to query the QoS computed values and update their services.

Menasce and Dubey [12] explained the selection of web service provider based on utility function under cost constraints. Utility function was used for QoS broker to perform service provider selection. Utility function allowed stake holder to ascribe a value to the usefulness of a system as the function of several attributes such as response time, throughput etc. The utility function was monotonically decreasing for response time and increasing for throughputs.

Lamparter et al. [13] described standard representation of web service configurations and preferences of users to meet the requirements using utility function policies. The utility function was used to model web service configuration and preferences. This work allowed for developing an algorithm for optimal service selection by using multi attribute decision theory methods. The utility of service configuration was given by quasi-linear function that represented the difference between requestor preference score and its price.

Qu et al. [14] provided context-aware cloud service selection model based on comparison and aggregation of user subjective assessment and objective performance assessment. The objective and subjective performances were aggregated to find overall performance. This method used utility function to measure the context similarity of the web services.

Constraint programming is the study of computational systems based on constraints. The main concept of constraint programming is for solving the problems by stating constraints which must be satisfied by the solution. Constraint Satisfaction Problem (CSP) is a problem that consists of finite set of variables. The function maps every variable to a finite domain and a finite set of constraints. Ruiz-Cortés et al. [15] explained how to automate the procurement of mapping service demands and service offer onto the CSP. The checking for consistency of an offer was to find internal contradiction and checking for conformance was to conform a demand that allows to check whether demands of one party satisfy the requirements of another party and vice versa. The architecture of the two-way matchmaker contains ST-matchmaker, translator, pre-processor and solver where the process is automated. In case of optimal selection experiment, parameter value offers showed linear nature but non-parameter value offer showed polynomial function. So when demands and offers increased, performance was also increased. In this case, non-

parameter value offer was better than parameter value offer. It showed strong impact on performance of finding the minimum value of a function.

Kritikos and Dimitris [16] described the Mixed Integer Programming (MIP) technique to solve the problem of web service selection and matchmaking. The CP was used to solve non-linear constraints but MIP was used to solve linear constraints. MIP was similar to Constraint Programming (CP) but MIP variable could be integer or real value, whereas CP could only be integer. The experimental result showed that in case of linear constraints, MIP outperformed the CP. CP was more efficient to solve constraints with small domain involving integers only, whereas the size of domain does not play an important role in solving problem using MIP.

Ma et al. [17] developed a semantic QoS-aware framework for web services by combining constraint programming and semantic matchmaking. This method included three layers: semantic matchmaking layer, constraint programming layer and QoS selection layer. They also defined QoS ontology for web service selection.

Analytic Hierarchy Process (AHP) is a decision-making process in which a problem is divided into smaller units. In this process, first the problem is defined then the decision hierarchy is structured from top to bottom with decision making goals. After this, a set of pair-wise comparison matrices are developed. Then each element in upper level to lower level is compared. Priorities are made by using weight until the final priorities of the alternatives in the bottom-most level are obtained. Tran et al. [18] proposed a model for designing and developing QoS ontology for web services. The proposed QoS ontology has five parts. The first part has QoS role, description, level and group, and other part has QoS property, relationship, metric and core QoS properties. They used AHP model for developing dynamic ranking algorithm. Pair-wise comparison was made in AHP that determined the preference of QoS property in terms of

ranking the web services. For a service request, first, AHP hierarchy was developed then weights of QoS properties were calculated. Finally, the weighted sum of all QoS values was calculated as the final ranking score.

Greg et al. [19] provided the framework and mechanism for prioritizing and assessing the quality of cloud services. It followed the MCDM technique to rank the cloud services and consists of three phases: building a hierarchical structure for cloud services based on SMI KPI's, computing the relative weight of each QoS property, and ranking cloud services. Herssens et al. [20] tested the selection of web services according to priorities by using a MCDM approach. It used out-ranking method to define global priority constraint for the ranking of web services. Rehman et al. [21] provided cloud service selection methodology that utilizes past history data and performs MCDM analysis to rank all cloud services. The MCDM used in the pre-interaction phase was to capture the variations in QoS over time and then identify the criterion that was not important for decision making. The QoS data available for different time slots formed a decision matrix using the MCDM technique. Finally, the values obtained in different time slots were aggregated. Two approaches TOPSIS method and ELECTRE method were used to find the top ranked services in each time slot.

The quality of service that is provided by service provider is uncertain. It may change over time. Most approaches use predefined function to solve the problem. If multiple quality criteria are to be considered, then users are required to express their request over different quality attributes as weighted numeric value. To deal with the uncertain quality of web services, Yu and Bouguettaya [22] purposed p -dominant service skyline for the selection of web services. This method addressed some shortcomings of the existing web service selection and ranking approaches. It computed the p -dominant skyline using P-R-Tree. They developed two

algorithms-the first one computes the dual pruning and the second one computes dominate probability.

Skoutas et al. [23] used the skyline approach to define the dominance relations between services and developed an algorithm to retrieve top k-most dominant web services. It proposed three algorithms for matching web service descriptions with service request according to the criteria, including TKDD, TKDG and TKM. The TKDD computes top-k web services according to dominated score criteria. The main objective of this was to quickly find each object in which other objects are dominating it. TKDG computes top-k dominant web services. It retrieved k-match objects they were dominating from large numbers of other objects. The TKM also computed top-k matches with respect to some criteria. It is derived from TKDG.

Zheng et al. [24] developed a framework for QoS ranking of cloud services. They described the system architecture of QoS ranking prediction for cloud services and defined the technique to predict similar users. They proposed mainly two algorithms called CloudRank1 for finding an approximately optimal ranking and CloudRank2 for more accurate ranking prediction by defining confidence values of different preference values.

Karthiban [25] proposed QoS-aware selection of web services based on clustering. It proposed a technique to mine WSDL documents then clustered them into the group of similar types of web services.

In [26], selection of web services was related to satisfying three parties: consumers, providers and brokers. User satisfaction was measured on the requests sent to the whole community. They considered three QoS attributes: availability, successibility and response time. The provider satisfaction was measured by the overall participation of all providers in the community in which participation was calculated by the ratio between workload and capacity.

Broker satisfaction was measured by the total revenue. This approach is community-based and considers the satisfaction of three parties in the community.

Most of these service selection approaches discussed above only considered the selection and ranking of web services based on user's perspective but our approach is based on three parties' perspectives: user's, provider's and broker's. The work reported in [26] is the closest to ours. One of the major differences is that they consider that all the providers form a community and thus any member from the community can provide the required service to the user, whereas in our work, we consider each provider is an independent entity and providers are competing with each other to attract users. Another major difference lies on our objective function. Compared to the objective function used in [26], it considers different criteria and it is built upon the unique profit model we have proposed for the marketplace.

2.3 Marketplace, Brokerage and Bidding Models

A marketplace is a kind of business site where product and service information is provided by multiple third parties whereas the transactions are processed by the marketplace operator. The brokerage models are generally used to match more than one party for their demands and supply. Bidding models are used to create the point where transactions are occurred among the parties. In our proposed model, brokerage is used to match the requirements of the consumers and services of the providers for the selection of web services, and generate more revenue inside a marketplace.

Zhang et al. [27] developed auction based approach to cloud service differentiation. This research described the interaction with users by auction mechanism that allows users to specify their priorities using budget and job characteristics defined by a utility function. The proposed

model considers multiple users and multiple providers and is a cost-effective process for cloud providers and users. Map-Reduce scheduling technique was used for resource allocation and utility function was used for virtual partitions of job.

Lampe et al. [28] provided an optimal approach for maximizing cloud provider's profit using the mechanism of equilibrium price auctions. It described the utility function for optimal allocation between multiple providers and multiple users, and it also defined the algorithm for VM prices and VM distribution using the heuristic allocation approach. The proposed model was mainly addressing concerns for cloud providers and it used the concurrent pricing and distribution of virtual machines across physical machines based on equilibrium price auction.

Tang et al. [29] provided bidding strategies to minimize the cost and volatility of resources provisioning by using the Constrained Markov Decision Process (CMDP). This research considered multiple users and multiple providers and mainly focused on costs and resources. The proposed model defines optimal bidding strategy based on CMDP model and also policies.

Leslie et al. [30] proposed resource allocation and job scheduling framework. This model considered multiple users and multiple providers and was based on resources, execution costs for cost approximation.

Shang et al. [31] proposed the dynamic pricing scheme to meet different requirements. They determined the price of cloud resources using learning algorithms based on historical trading information. It considered multiple users and multiple providers and proposed knowledge based double auction model. The objective function used only price to find the overall profit of buyers and sellers through the trading.

Shang et al. [32] defined auction Bayesian game based model for cloud market and developed optimal pricing strategy for this model. They considered multiple users and multiple providers, and objective function used for pricing strategy. They developed a simple pricing solution between cloud resource provider and resource buyers. In experiment, it analyzed the pricing strategy by using the simplified model. Finally, from the analysis, it developed a simple equation of pricing strategy for providers and consumers.

Fujiwara et al. [33] proposed a market mechanism that enabled users to order a combination of services for workflows and co-allocations and to reserve future and current services in a forward and spot market. They considered multiple providers and multiple users. The objective function used price and resources. It developed a simulator system called W-mart to explore market behavior. It found the interaction between forward market and spot market where forward price was expected to be forecast of the spot price.

Murali et al. [34] described a simulation model for real time bidding which was used to find optimal budget allocation strategies. This model was used to optimize performance and budget. The stochastic dynamic programming approach was a simulation model used to determine the budget in each time slot for this model.

Chen and Zhang [35] proposed a bidding and auction mechanism in multi-project organizations to help decision making. This model solved the resources allocating mechanism in multi-projects. They described two levels in which upper level represented the benefits by allocating global resources and lower level represented the benefits of specific projects.

Li et al. [36] proposed feedback control-based model for decision making in bidding for cloud spot services. They minimized the errors from previous similar work in bidding and used Amazon's historical spot price to validate this model.

Barreto et al. [37] developed a conceptual model in the cloud for brokering and authentication. They addressed the issue of sharing different resources with users and they also checked authentication information to determine whether they are proper users. This was implemented in different cloud domains.

He et al. [38] proposed a model for service selection that was based on combinatorial auction. They proposed an iterative process in which first, service providers bid for combination of services and provide offers for the multi-dimensional quality of the service, and then according to these received bids, found the solution to fulfill the quality constraints. If the solution was not found, the auction iterates such that service providers improve their bids to improve the winning chance. They defined to create new services by combining existing services and to provide more competition among providers.

Park et al. [39] proposed a Virtual Cloud Bank (VCB) to provide cloud services by using Cloud Service Brokerage (CSB). They explained three models: tenant analysis model, service analysis model and cloud service query model for collecting and analyzing user's requirements, analyzing cloud services and query the cloud services respectively.

Mohabey et al. [40] described an intelligent procurement marketplace for mixing up web services. They developed a combinatorial auction model and focused on QoS and SLA of multiple providers who provide web services. The mathematical functions and interfaces were designed for the solution.

These marketplace, brokerage and bidding models mainly focused on trading of web services in a marketplace. Some of this research focused on resources utilization in clouds. Many models were developed for providers, or from provider's point of view. Our work is mainly focused on the selection of web services and its purpose is to satisfy both consumers and

providers. In addition to this, it proposed a profit model that can generates more revenue for the broker.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This work proposes a new approach for the selection and ranking of web services. Users basically have functional and non-functional requirements for the selection of web services. Functional requirements are related to the working function of the web services, and non-functional requirements are related to the quality of services. This work focuses on the ranking of web services based on their non-functional requirements to satisfy users, providers and the broker.

3.2 A Motivating Example

The marketplace consists of many providers offering a variety of services. Providers publish their services while the users search the relevant services from the marketplace. The current service selection algorithm ranks the services based on the QoS-only objective function. This ranking mechanism ranks and recommends the services in the same way, for the same types of requests for every user. Therefore, the lower ranked services may never get a chance to be selected.

Let us consider web services for airline reservation. Suppose there are three services: *A*, *B* and *C*. Their QoS values on the three attributes availability, reliability and response time are listed in Table 3.1.

Table 3.1: Ranking of services by QoS-only

Service	Availability (%)	Reliability (%)	Response time (ms)
A	80	90	5
B	80	89.8	5.1
C	65	60	10

By using a ranking mechanism which considers only the QoS information in its objective function, if a user has a request as follows: $\text{availability} \geq 65$, $\text{reliability} \geq 60$ and $\text{response time} \leq 10$, service *A* is selected as the best service, followed by service *B* and then *C*. Although the overall score of *A* and *B* are not very different, service *A* is always ranked at the top. Service *B* is never ranked first in any circumstances. If a system only returns the top ranked service, or a user only chooses the top ranked service, the provider of service *B* may eventually consider withdrawing from the marketplace. If the provider of service *B* is allowed to provide some incentive to make it appear in the first position, some users may choose to use service *B* and their satisfaction degree should not be much different than that on service *A*, considering that their QoS values are very close to each other. However, the current selection algorithms will never give this opportunity to service *B*.

The main purpose of our selection algorithm is to promote services such as *B* if their providers would like to pay a fee to have their services to be promoted, in the condition that all the requirements of users can still be satisfied, and there is not much difference between their ranking scores with the score of the top-ranked service. The overall score of service *C* is more different than those of *A* and *B*. Therefore, it has no opportunity to be promoted even if its provider wants to pay a fee. Users may not care about the exact ranking order of the services as long as their requirements are satisfied. Our algorithm aims to provide an opportunity for more providers and users to be included and stay in the marketplace. Thus, the broker has a chance to attract more users and providers to allow maximum revenue to be generated.

3.3 QoS Attributes

QoS attributes help to determine the list of the best web services based on user requirements. They are divided into different categories. For positive attributes such as throughput, availability, higher values are considered better, and for negative attributes such as response time, cost, lower values are considered better [20]. Some QoS attributes for the selection and ranking of web services are explained below [6] [12 [20].

- **Response Time:**

This is the time taken to send the request and receive the response. The response completion time is the time when all the data for response arrives to a user whereas the user request time is the time when the user sends a request. It is generally measured in milliseconds.

- **Throughput**

This is defined as the maximum number of requests that the web services can process for a unit time. It can be used as a performance index to evaluate a Web Services Provider (WSP).

- **Availability**

Availability is defined as the percentage of time that a customer can access a web service. It is measured in percentage.

- **Successibility**

This measures the requests that have been successfully completed. It is the ratio of response message to request message and is measured in percentage.

- **Reliability**

It is defined as the ability of a system to work as expected for the specific time period. It is measured in percentage.

- **Compliance**

This is defined as the extent to which a WSDL document follows WSDL specification. It is measured in percentage.

- **Best practice**

This is defined as the extent to which a web service follows Web Service Interoperability (WS-I) basic profile. It is measured in percentage.

- **Latency**

This is defined as the time taken to process a request by a server. It is generally measured in milliseconds or microseconds.

- **Documentation**

This is used to measure how much documentation is completed. For example, how many description tags are completed in WSDL?

Our selection system has used the above 9 QoS attributes due to the dataset we used in the experiment. The selection approach itself is flexible so that we can easily add or remove QoS attributes.

3.4 Architecture of our Service Marketplace

A service marketplace is a place where providers can publish their services and users can search and request services. In our design of the marketplace, we consider it is mainly for publishing, selecting and requesting (or invoking) web services. And therefore, its basic components include publication UI, selection UI, result UI, invocation UI, invocation proxy, and the matchmaking broker. Its major data repositories include service repository and monitored

QoS repository. The architecture model of the marketplace is shown in Figure 3.1. The service selection algorithm is implemented inside the marketplace.

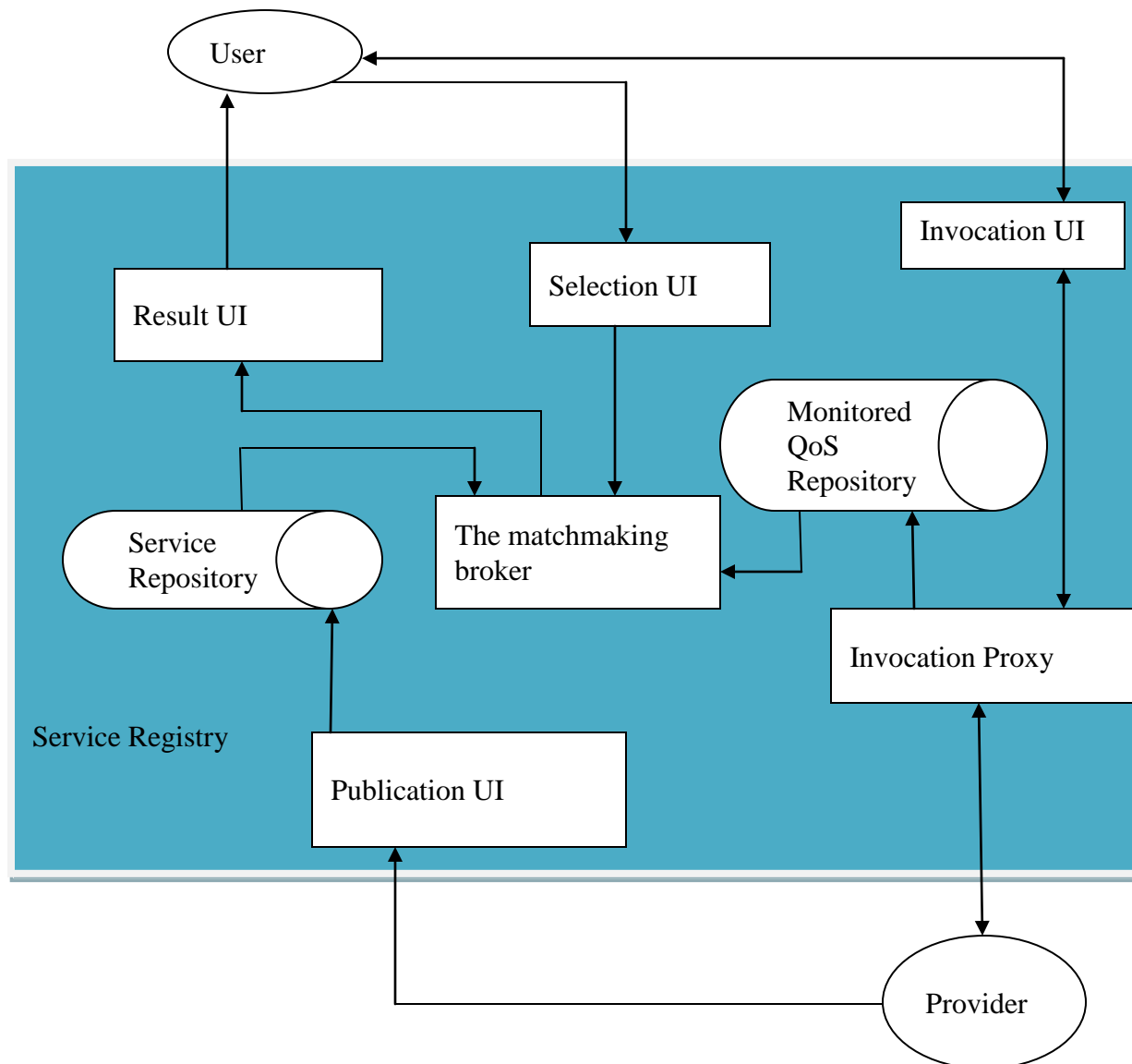


Figure 3.1: Architecture of service marketplace

Users play a main role inside the marketplace. They send requests to the marketplace according to their requirements, and they receive the results generated by the ranking algorithm used by the broker. Providers also play a main role inside the marketplace. They provide services for the users. There may be more competition among the providers inside the marketplace. Selection UI and result UI are those components through which users can submit their requests

and receive the results respectively. A set of services of the providers are stored in the service repository. Users' invocation histories are recorded in the invocation proxy. The actual QoS data are saved in the monitored QoS repository. The matchmaking broker is to match the service with the user request using our proposed selection and ranking algorithm.

The workflow of the selection system is explained as follows. The service providers may have a number of services which are published into a service repository. When the users want to search the services, they provide functional and non-functional requirements through a selection UI. All the requirements that the user provided are then passed to the broker. The broker first identifies those services that match the functional requirements from a service repository, and then ranks the services based on non-functional requirements. The resulting services are then given to the users.

To record the user's selection pattern, the invocation request of the service is passed to invocation UI to the invocation proxy then sent to the service provider. And the requested service is passed through the invocation proxy to invocation UI to the user. In this way, the proxy can record the invocation history of the user, and also monitor the actual QoS values of the invocation. The monitored QoS Repository is used to save the monitored QoS data.

In our proposed marketplace, there are two types of users and providers – free and paid. The ranking of the services is different for different types of users and providers according to our selection and ranking algorithm. If a paid user requests for a service in the marketplace, QoS-only objective function is used to rank the services no matter the providers are paid or free. If a free user requests for a service and the top-ranked service based on the QoS-only objective function is from a free provider, the services are re-ranked based on our proposed QoS-Plus-FP objective function. The details of the selection algorithm are discussed in Section 3.5.

3.5 Selection Algorithm

3.5.1 Overview

The marketplace is a platform for users, providers and the marketplace. The services of the providers are provided through the marketplace to the users. A marketplace has a matchmaking algorithm to match the requirements of the users, with the services of the providers. By using this algorithm, users are satisfied because all their requirements are satisfied, providers are satisfied because more of their services are selected compared to the case when normal ranking mechanisms are used, and the marketplace is satisfied because more users and providers are in the marketplace and more revenue can be generated through charged fees to the providers and users.

Let us consider a marketplace which has a number of web services with similar functionalities from different providers. There are a number of users requesting the services. A user may have a number of requests with different requirements.

Let the users be:

$$u_1, u_2, u_3, u_4, \dots, u_m$$

where m is the total number of users.

Each user may be requesting the service many times. For each user u_j , the number of requests is l_j and a list of requests is:

$$req_1^j, req_2^j, req_3^j, req_4^j, \dots, req_{l_j}^j,$$

All the users and their requests are listed below.

Users

Requests

$$u_1 \quad req_1^1, req_2^1, req_3^1, req_4^1, \dots, req_{l_1}^1$$

$$u_2 \quad req_1^2, req_2^2, req_3^2, req_4^2, \dots, req_{l_2}^2$$

$$\begin{array}{ccc}
u_3 & & req_1^3, req_2^3, req_3^3, req_4^3, \dots \dots \dots, req_{l_3}^3 \\
\dots & & \dots\dots\dots \\
u_m & & req_1^m, req_2^m, req_3^m, req_4^m, \dots \dots \dots, req_{l_m}^m
\end{array}$$

There are many services available which users can select according to their requirements.

Let the services be:

$$S_1, S_2, S_3, S_4, \dots \dots \dots, S_n$$

where n is the number of services.

3.5.2 Selection Process

This approach defines the marketplace by considering three parties: users, providers and the marketplace itself. There are two types of users who would request services in the marketplace including free and paid users. Similarly, there are two types of providers who would offer their services in the marketplace including free and paid providers. The users and the providers have different policies inside the marketplace. Different objective functions are used to rank the services according to the types of users and providers.

In our marketplace, since there are two types of users and providers, based on their combinations, we have the following selection process.

- **Free user:** If a free user requests for a service and the first service ranked by QoS-only objective function is from a paid provider, a list of services ranked by QoS-only objective function is the final ranking list.

If a free user requests for a service and all providers are free, again, QoS-only objective function is used to rank the services. If a free user requests for a service and there is a mix of paid and free services, when the top service ranked by the QoS-only objective function

is from a free provider, services will be re-ranked based on QoS-Plus-FP objective function to decide whether and which paid service will be promoted.

- **Paid user:** If a paid user requests for a service, QoS-only objective function is used to rank the services no matter the providers are paid or free.

In any case, the service promoted through the advertisements can always be part of the selection result page.

The service selection and ranking process is shown in Figure 3.2.

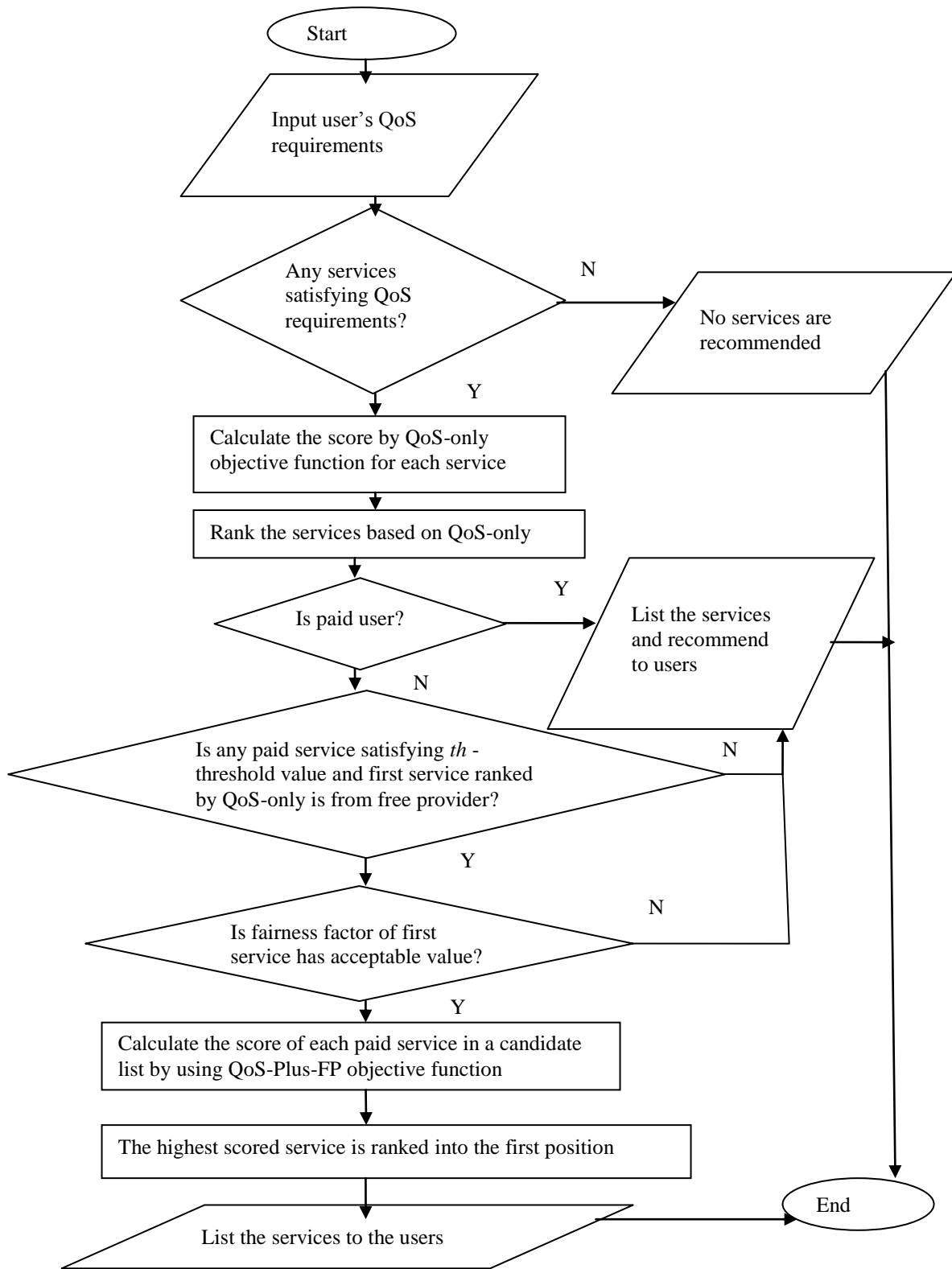


Figure 3.2: Selection and ranking process

Figure 3.2 described the whole selection and ranking process. The symbol th represents a pre-defined threshold value which is the maximally allowed percentage of the score difference between the optimal score among all services and the score of the current service. In this work, value of th is set to 10%. The fairness measurement factor is the measurement of fairness of free services, which will be defined later. If the free service has the rank changed to a lower position, the fairness score is reduced. In this work, the fairness measurement factor is set to $1/10$, which means the fairness score should always be greater than or equal to $1/10$.

First, the users' requirements are input then all the services satisfying these requirements are listed. Then, the overall score of each service is calculated by using QoS-only objective function. After that, services are ranked according to these scores.

If a paid user requests a service, the matching services are ranked based on the QoS-only objective function. Therefore, no further processing is required.

If a free user requests a service, then the first service in the list is checked. If the first service is from a paid provider, those services are listed to a user. If the first service is from a free account provider, then other services in the list are checked. If any other paid listed services satisfy the th -threshold value, promotion score of each of these paid services is calculated, and fairness scores of all free services whose scores by the QoS-only objective function are greater than the score of the potentially promoted paid service are calculated by assuming these services are demoted to lower positions. The fairness measurement factor of the first ranked service is checked. If its value is not in an acceptable range, then the same ranking list is returned to the user; otherwise, the ranking scores of all paid services based on QoS-Plus-FP objective function are calculated. After that, the paid service with the highest QoS-Plus-FP score is promoted to the first position. Then, the ranked list of services is returned to the user.

3.5.3 Selection Approach by QoS-only and its Objective Function

The traditional approach used QoS-only objective function to rank the services. In this method, objective function is designed to rank the services based on the QoS values only.

This approach includes two steps: Data normalization and ranking based on the QoS-only objective function.

- **Data Normalization**

The values for different QoS attributes vary. There is no compatibility among different QoS attributes. The larger values are better for some QoS attributes and smaller values are better for other QoS attributes. The scales of the values are also quite different. It is meaningless to calculate the weighted sum of the raw values of QoS attributes. Therefore, we should normalize the values of the QoS attributes first.

For each QoS attribute, let us assume that Q is the original QoS value from a service, and Q_{max} and Q_{min} are the maximum and minimum values of this QoS attribute among all services.

For a positive monotonic attribute, the QoS values are normalized using (1).

$$NQ = \frac{Q - Q_{min}}{Q_{max} - Q_{min}}, \quad (1)$$

where $Q_{max} - Q_{min} \neq 0$.

For a negative monotonic attribute, the QoS values are normalized using (2).

$$NQ = \frac{Q_{max} - Q}{Q_{max} - Q_{min}}, \quad (2)$$

where $Q_{max} - Q_{min} \neq 0$.

- **QoS-only objective function**

The QoS-only objective function is defined as the weighted sum of the normalized QoS values and the weight of each QoS attribute is decided by the user preference [41]. The minimum requirements of the user for selecting the web services should be satisfied. Each service has different QoS values, and each QoS attribute is weighted differently based on every user request.

The score of the matching degree between user u_j 's request req_k^j and service s_i is given below [41].

$$\text{Score}_k^j(S_i) = \sum_{h=1}^q w_h Q_{ih} \quad , \quad (3)$$

under the constraints

$$Q_{ih} \leq Q_{rh} \text{ if smaller values are considered better}$$

and

$$Q_{ih} \geq Q_{rh} \text{ if larger values are considered better}$$

where q is the total number of QoS attributes, w_h is the weight of the user request on the h -th attribute, Q_{ih} represents the normalized QoS values of the i -th service on the h -th QoS attribute and Q_{rh} is the minimum QoS value required for each request.

3.5.4 QoS-Plus-FP Objective Function

This objective function is used to rank the web services to benefit multiple parties. First, this approach designs the individual objective functions measuring the difference between optimal and current service score, average fairness score, average promotion score and promotion score of each paid service. Then the weighted sum of these individual objective functions would be the QoS-Plus-FP objective function used in the ranking process. According to the score of the QoS-Plus-FP objective function, services are ranked.

- **Objective function measuring the difference between optimal and current QoS score**

The satisfaction level depends on the score of the QoS matching degree. The score of the services closer to the optimal score is more satisfactory than the score from far. Therefore, the difference between the score of a service based on QoS-only objective function and the optimal score plays an important role for the satisfaction of a service user. The smaller the difference, the more satisfied the user.

The optimal score from the QoS based objective function is defined in (4),

$$OScore_k^j = \max_{i \in [1, n]} (Score_k^j(S_i)), \quad (4)$$

where $Score_k^j(S_i)$ is score of services S_i for the k -th request of user u_j and the optimal score is the maximum values among all matching services (S_1 to S_n).

The difference between the optimal score and the score of the current service is defined in (5).

$$DifS_k^j(S_i) = OScore_k^j - Score_k^j(S_i), \quad (5)$$

The smaller value of the difference is the better. Therefore, the normalized difference score is calculated as in (6).

$$FDifS_k^j(S_i) = \frac{Max_Score - DifS_k^j(S_i)}{Max_Score - Min_Score}, \quad (6)$$

where Max_Score is the maximum score among all difference scores and Min_Score is the minimum score among all difference scores.

- **Average Fairness Score**

Fairness score is the measurement of how fair our algorithm is for free services. We calculate the fairness score considering the case when a free service is demoted to a lower position. If fairness measurement factor $fs(s_i)$ is not in an acceptable range, it is kept as first ranked service and fairness score is reset to 1, otherwise we demote a free service and promote a paid service by using QoS-Plus-FP objective function on all satisfying paid services.

The fairness score is only changed when the rank of a free service is changed into a lower position. In this case, we need to calculate the fairness score for this service, and also update the average fairness score for all the free services in the marketplace. Initially, the fairness score of a free service is set to 1, then it is calculated using (7).

$$fs(S_i) = \left(f_i - \frac{1}{Z_i}\right), \quad (7)$$

where f_i is the fairness score of the i -th service before the demotion and Z_i is the change factor of fairness score. The value of Z_i can be set to any value. In this work the value of Z_i is set to 10 if $i \leq 3$, otherwise 20. The reason for selecting these values for Z_i is that the demotion is less fair for the top ranked services than other lower ranked services and thus we set this value as 10 for top three services and 20 for other services.

For each paid service that potentially can be promoted, its promotion would cause the demotion of other higher-ranked free services. So we want to calculate the average fairness score to measure the impact of promoting each such service to affected free services. For all paid services, if their scores ranked by QoS-only objective function are greater than or equal to th -threshold value and all the requirements of the users are satisfied, a list of these services is called candidate list or promotion list. In this work the threshold value th is set to 10%.

The average fairness score is calculated for those services which are in the promotion list. For each paid service, if it is promoted, the average fairness score is calculated using (8).

$$Avg_{fs}(S_i) = \frac{\sum_{i=1}^{n-p} fs(S_i)}{n-p}, \quad (8)$$

where p is number of paid services. The service with higher average fairness score has more chance to be promoted.

- **Promotion score for paid services**

The promotion score is only considered for paid services. It depends on how many times the service is included in the promotion list. If the service satisfies the threshold value th from QoS only objective function, these services are in the promotion list. The service with a higher value of promotion score has more opportunity to be promoted.

Initially, the promotion score of each paid service is set to 0. If the services are in the promotion list, the promotion score of each paid service is calculated using (9). If the service is promoted by using QoS-Plus-FP objective function, the promotion score is reset to 0. This process is continued for all the requests if promotion of the service is taking place.

$$ps(S_i) = \frac{Nwp_i}{\max_{i \in [1,n]}(Nwp_i)}, \quad (9)$$

where Nwp_i is the number of times S_i is in the promotion list but not being promoted and $\max_{i \in [1,n]}(Nwp_i)$ is the maximum number of times a service in the candidate list without being promoted among all paid services.

- **Average Promotion Score**

The average promotion score of each paid service is calculated if they are in the promotion list. First the promotion score of each paid service is calculated by (9). Then we

suppose that each service is promoted and set the promotion score to 0. Finally, the average promotion score of all paid service is calculated using (10).

$$\text{Avg}_{\text{ps}}(S_i) = \frac{\sum_{i=1}^p \text{ps}(S_i)}{p}, \quad (10)$$

When the service is promoted by QoS-Plus-FP objective function, promotion score of the promoted service is set to 0.

- **Overall QoS-Plus-FP objective function**

The QoS-Plus-FP objective function is the combination of objective functions defined in (6), (8), (9) and (10). This function is the final ranking function. It is defined in (11).

$$\text{QoSPlusFPObjFun}_k^j(S_i) = W1\text{FDif}S_k^j(S_i) + W2 \text{Avg}_{\text{fs}}(S_i) + W3 \text{ps}(S_i) + W4 \text{Avg}_{\text{ps}}(S_i), \quad (11)$$

where $W1$, $W2$, $W3$ and $W4$ are the weights on each individual objective function. In the experiment, all the weights are set to equal values and $W1+W2+W3 + W4=1$.

3.5.5 Illustrating the Ranking Model by an Example

We use a simple example to illustrate the ranking algorithm. We considered 12 services in this example in which free and paid status of these services are defined in Table 3.2. The steps to rank the services are given below.

- The ranking list of the services by QoS-only objective function is shown in Table 3.2.

Table 3.2: Ranking list of the services by QoS-only objective function

Service Name	QoS-only Score	Free/Paid
S1	0.9989	Free
S2	0.9987	Free
S3	0.9986	Free
S4	0.9985	Paid
S5	0.9983	Paid
S6	0.9982	Free
S7	0.9980	Free
S8	0.9977	Free
S9	0.9000	Paid
S10	0.7231	Paid
S11	0.7001	Free
S12	0.6991	Paid

- The list of services that satisfies user requirements and threshold value th are shown in Table 3.3. This list is called the candidate list or the promotion list.

Table 3.3: List of the services in the candidate list

Service Name	QoS-only Score	Free/Paid
S1	0.9989	Free
S2	0.9987	Free
S3	0.9986	Free
S4	0.9985	Paid
S5	0.9983	Paid
S6	0.9982	Free
S7	0.9980	Free
S8	0.9977	Free
S9	0.9000	Paid

The paid services in the candidate list are {S4, S5, S9}.

- We find the normalized values of the score difference between optimal service and current service which is shown in Table 3.4.

Table 3.4: Normalized score difference between the optimal and current service

Service Name	QoS-only Score	Normalized score difference between the optimal and current service	Free/Paid
S1	0.9989	1.0000	Free
S2	0.9987	0.9993	Free
S3	0.9986	0.9990	Free
S4	0.9985	0.9987	Paid
S5	0.9983	0.9980	Paid
S6	0.9982	0.9977	Free
S7	0.9980	0.9970	Free
S8	0.9977	0.9960	Free
S9	0.9000	0.6701	Paid
S10	0.7231	0.0801	Paid
S11	0.7001	0.0033	Free
S12	0.6991	0.0000	Paid

- We initialize fairness score for free services and promotion score for paid services as shown in Table 3.5.

Table 3.5: Initialized fairness and promotion score

Service Name	QoS-only Score	Normalized score difference between the optimal and current service	Free/Paid	Fairness Score	Promotion Score
S1	0.9989	1.0000	Free	1	
S2	0.9987	0.9993	Free	1	
S3	0.9986	0.9990	Free	1	
S4	0.9985	0.9987	Paid		0
S5	0.9983	0.9980	Paid		0
S6	0.9982	0.9977	Free	1	
S7	0.9980	0.9970	Free	1	
S8	0.9977	0.9960	Free	1	
S9	0.9000	0.6701	Paid		0
S10	0.7231	0.0801	Paid		0
S11	0.7001	0.0033	Free	1	
S12	0.6991	0.0000	Paid		0

- We assume that service S4 is promoted. We change the fairness and promotion score as shown in Table 3.6 and then calculate its score using QoS-Plus-FP objective function.

Table 3.6: Fairness score and promotion score to promote S4

Service Name	QoS-only Score	Normalized score difference between the optimal and current service	Free/Paid	Fairness Score	Promotion Score
S1	0.9989	1.0000	Free	0.9	
S2	0.9987	0.9993	Free	0.9	
S3	0.9986	0.9990	Free	0.9	
S4	0.9985	0.9987	Paid		1/1
S5	0.9983	0.9980	Paid		1/1
S6	0.9982	0.9977	Free	1	
S7	0.9980	0.9970	Free	1	
S8	0.9977	0.9960	Free	1	
S9	0.9000	0.6701	Paid		1/1
S10	0.7231	0.0801	Paid		0
S11	0.7001	0.0033	Free	1	
S12	0.6991	0.0000	Paid		0

Normalized score between optimal and current service=0.9987

Promotion score=1

Average fairness score= $(0.9+0.9+0.9+1+1+1+1)/7=0.957$

Average promotion score= $(0+1+1+0+0)/5=0.4$

QoS-Plus-FP score=0.9987+1+0.957+0.4=3.3557

- We assume that service S5 is promoted. We change the fairness and promotion score as shown in Table 3.7 and then calculate its score using QoS-Plus-FP objective function.

Table 3.7: Fairness score and promotion score to promote S5

Service Name	QoS-only Score	Normalized score difference between the optimal and current service	Free/Paid	Fairness Score	Promotion Score
S1	0.9989	1.0000	Free	0.9	
S2	0.9987	0.9993	Free	0.9	
S3	0.9986	0.9990	Free	0.9	
S4	0.9985	0.9987	Paid		1/1
S5	0.9983	0.9980	Paid		1/1
S6	0.9982	0.9977	Free	1	
S7	0.9980	0.9970	Free	1	
S8	0.9977	0.9960	Free	1	
S9	0.9000	0.6701	Paid		1/1
S10	0.7231	0.0801	Paid		0
S11	0.7001	0.0033	Free	1	
S12	0.6991	0.0000	Paid		0

Normalized score between optimal and current service=0.9980

Promotion score=1

Average fairness score= $(0.9+0.9+0.9+1+1+1+1)/7=0.957$

Average promotion score= $(0+1+1+0+0)/5=0.4$

QoS-Plus-FP score= $0.9980+1+0.957+0.4=3.3550$

- We assume that service S9 is promoted. We change the fairness and promotion score as shown in Table 3.8 and then calculate its score using QoS-Plus-FP objective function.

Table 3.8: Fairness score and promotion score to promote S9

Service Name	QoS-only Score	Normalized score difference between the optimal and current service	Free/Paid	Fairness Score	Promotion Score
S1	0.9989	1.0000	Free	0.9	
S2	0.9987	0.9993	Free	0.9	
S3	0.9986	0.9990	Free	0.9	
S4	0.9985	0.9987	Paid		1/1
S5	0.9983	0.9980	Paid		1/1
S6	0.9982	0.9977	Free	0.95	
S7	0.9980	0.9970	Free	0.95	
S8	0.9977	0.9960	Free	0.95	
S9	0.9000	0.6701	Paid		1/1
S10	0.7231	0.0801	Paid		0
S11	0.7001	0.0033	Free	1	
S12	0.6991	0.0000	Paid		0

Normalized score between optimal and current service=0.6701

Promotion score=1

Average fairness score= $(0.9+0.9+0.9+0.95+0.95+0.95+1)/7=0.935$

Average promotion score= $(0+1+1+0+0)/5=0.4$

QoS-Plus-FP score= $0.6701+1+0.935+0.4=3.0051$

- We compare the QoS-Plus-FP score of S4, S5 and S9. We can see that S4 has greatest QoS-Plus-FP score so that S4 promoted to the first position. We then update fairness scores and promotion scores of all services affected by this service as shown is Table 3.9.

Table 3.9: Ranking of services by QoS-Plus-FP objective function

Service Name	QoS-only Score	Normalize value between the difference of optimal and current service	Free/Paid	Fairness Score	Promotion Score
S4	0.9985	0.9987	Paid		0
S1	0.9989	1.0000	Free	0.9	
S2	0.9987	0.9993	Free	0.9	
S3	0.9986	0.9990	Free	0.9	
S5	0.9983	0.9980	Paid		1/1
S6	0.9982	0.9977	Free	1	
S7	0.9980	0.9970	Free	1	
S8	0.9977	0.9960	Free	1	
S9	0.9000	0.6701	Paid		1/1
S10	0.7231	0.0801	Paid		0
S11	0.7001	0.0033	Free	1	
S12	0.6991	0.0000	Paid		0

This ranking list is for one request. We continue the same process for other requests for the users.

3.6 Summary

In this chapter, we have described the architecture of our marketplace, the service selection process by the QoS-only objective function, and the service selection process by our selection algorithm. We have also described different selection cases, and defined our QoS-Plus-FP objective function. In the next chapter, we will focus on the evaluation of our new algorithm.

CHAPTER 4

EXPERIMENTS

In this chapter, we first describe our experiment design and the implementation, and how we generate the simulated dataset. After that, we showed and analyzed various experimental results.

4.1 Experiment Design, Simulated Dataset and Implementation

In our experiment, we assume that the services have already been discovered according to the functional requirements by an existing method. Therefore, we only need to rank the services based on non-functional requirements. In this experiment, we mainly focused on ranking of web services based on QoS only approach (our baseline) and our approach.

In the experiment, to validate our approach, we would like to show: 1) how many services are promoted according to our objective function compared with the existing QoS-only objective function; 2) how the promotion of services changes in different scenarios (i.e., when the ratio between free and paid providers or between free and paid users changes); 3) how the ranking order of the free service changes in different scenarios; 4) what the impacts are on fairness of free services and promotion capabilities of paid services by our algorithm; 5) how the fairness and promotion scores are distributed.

In this work, we have proposed a selection algorithm which can be used in a service marketplace and we have used different objective functions in the selection process for different types of users (free or paid) and providers (free or paid). Our selection approach is different from

previous selection approaches. Therefore, we cannot use the standard QoS dataset such as QWS directly in the experiment. Here, we have decided to use a simulation program to generate data set and use the simulated datasets to evaluate our algorithm.

First, we used QWS dataset [4] – a standard service QoS dataset as our service QoS repository. It consists of 2507 services and each service has a name, values of 9 QoS attributes and the URL of its WSDL files.

A sample of the QWS dataset is given in Table 4.1

Table 4.1: A portion of QWS dataset [4]

Response Time	Successibility	Availability	Throughput	Reliability	Compactness	Best practice	Latency	Documentation
302.75	90	89	7.1	73	78	80	187.75	32
482	95	85	16	73	100	84	1	2
3321.4	96	89	1.4	73	78	80	2.6	96
126.17	100	98	12	67	78	82	22.77	89
107	95	87	1.9	73	89	62	58.33	93

We simulated a user dataset by setting the ratio between free and paid users at a few fixed values. Similarly, we simulated a provider dataset by setting the ratio between free and paid providers at a few fixed values. Each provider is assigned with a number of services from the QWS dataset.

Based on the value ranges (minimum and maximum values) of each QoS attribute in the QWS dataset, we simulated the user requests (non-functional requirements on one or multiple of these 9 QoS attributes). Each request was generated randomly and the value lied within the range of these maximum and minimum values. This dataset generation process is flexible because we can easily generate any number of requests. In each request, we can have any number of QoS attributes with any values (in the range of the values found in the QWS dataset). Any number of

users and providers can be generated following a pre-set ratio between free and paid ones. Also, each user can submit any number of requests.

In our current implementation, we set up the experiment in the following ways.

- We considered nine QoS attributes described in [4]. These attributes are response time, throughput, availability, successibility, reliability, latency, compliance, documentation and best practice. There were altogether 100 requests in the simulated request dataset. Each request contains the requirements on one to four QoS attributes. The reason we chose a small number of attributes is to make sure there are a reasonable number of matching services. If a request contains requirements on all 9 QoS attributes, there may be very few results and in some cases, we may end up with an empty result set. A sample request dataset is shown in Table 4.2. In the table, the requested values should be greater than or equal to the specified values for positive attributes, including successibility, availability, throughput, reliability, compactness, best practice and documentation. And the requested values should be less than or equal to the specified values for negative attributes, including latency and response time.

Table 4.2: Request dataset

Request Id	Response Time	Successibility	Availability	Throughput	Reliability	Compactness	Best Practice	Latency	Documentation
1					50.50				21.77
2	2943.83		73.33						46.95
3					58.40	47.54			78.05
4			74.65	11.96			71.44	2590.98	
5	3492.86							2702.07	

- We generated 100 users in our user dataset and the ratios between free and paid users we considered are listed in Table 4.3. Since there are 100 users, if the ratio is 10:90, it means there are 10 free users and 90 paid users.

Table 4.3: Ratio between free and paid users

Free User	0	10	30	50	70	90	100
Paid User	100	90	70	50	30	10	0

A sample of user dataset is shown in Table 4.4. The free users are represented as 1 and paid users are represented as 0.

Table 4.4: User dataset

UserId	IsFreeUser
1	1
2	0
3	0
4	1
5	1

- We generated 100 providers in our provider dataset and the ratios between free and paid providers we considered are listed in Table 4.5. We randomly assigned the 2507 services from the QWS dataset [4] in equal portion to all providers. Among all of the services, 25 were assigned to each of the first 99 providers and the last provider was assigned with 32 services.

Table 4.5: Ratio between free and paid providers

Free Provider	0	10	30	50	70	90	100
Paid Provider	100	90	70	50	30	10	0

A sample of provider dataset is Table 4.6.

Table 4.6: Provider dataset

Prov Id	Serv Id	Response Time	Successibility	Availability	Throughput	Reliability	Compactness	Best Practice	Latency	Documentation	Free/Paid Provider Ratio
											70:30
1	1	302.75	89	7.1	90	73	78	80	187.75	32	Free
22	539	287	89	12	96	67	100	82	1	35	Free
48	1194	210.25	78	8.8	78	67	89	72	54.75	8	Free
80	1992	329.13	85	0.4	85	53	89	66	139.74	8	Paid
90	2226	145	85	6.8	95	73	78	84	9	9	Paid

- In each run, there were 100 users, and each user picked 1 to 5 requests randomly from the request dataset.

We run the experiment in the following steps. First the numbers of users and providers with free and paid ratios are set up. There are altogether seven settings for the users and seven setting for the providers as defined in Table 4.3 and Table 4.5. Therefore, the total number of settings is 49. Then, each user can pick one to five requests from the request dataset. There are 100 users to pick requests. After that, each request is processed. Finally, the ranking lists of services are returned to the users.

Our system was implemented using Java Programming Language with Eclipse and SQLite database. This experiment was run on a PC with Intel Pentium Central Processing Unit (CPU) B940, 2.00GHZ clock speed, 32-bit operating system, 2GB Random Access Memory (RAM). The operating system used in this experiment was Windows 7.

4.2 Results and Analyses

We ran each setting of the experiment 10 times then final result was the average of these 10 runs (shown in Appendix Table A.1-A.5). These experimental results were used to analyze

the results. The analysis is focused on the results of the difference between QoS-only objective function and QoS_Plus-FP objective function for free and paid- users and providers. It is also focused on promotion of paid services, and impacts on free and paid- users and providers by our QoS-Plus-FP objective function.

4.2.1 Analysis of Promotion of Paid Services and Providers

In this section, we analyze the promotion percentages of paid services, unique paid services and unique paid providers in different scenarios. The detailed results are reported in Appendix Table A.6-A.10.

The promotion percentage of paid services is calculated as the number of services promoted divided by the total number of requests. Figure 4.1 shows the change in the percentage values when the ratio between the free/paid users is changing, and here the ratio between free/paid providers is 50:50).

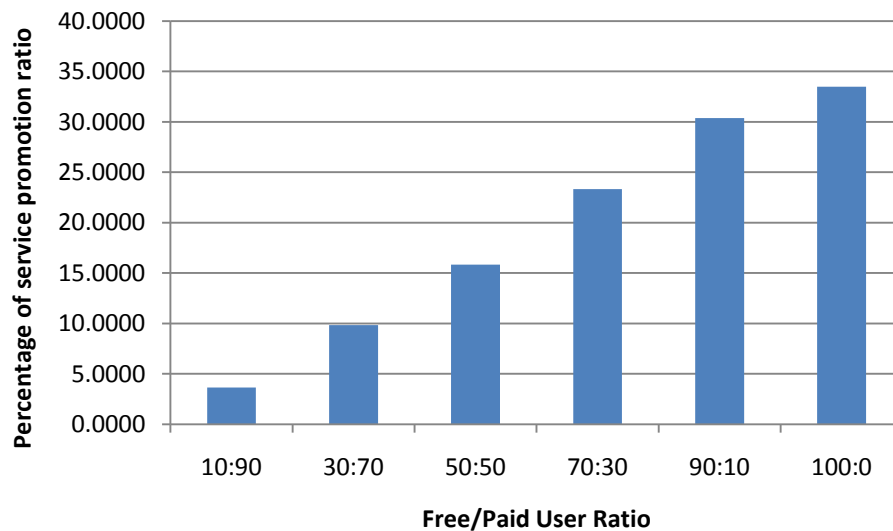


Figure 4.1: Percentage of service promotion when the free/paid user ratio changes (free/paid provider ratio: 50:50)

As we can see in the figure, when the ratio between free and paid providers is fixed and the ratio between free and paid user increases, the percentage of service promotion is also increasing. This is because in our system, the promotion only happens on requests from free users. If more free users request the service, the probability of promoting services is also higher.

The percentage values for all free/paid provider ratios are shown in Figure 4.2.

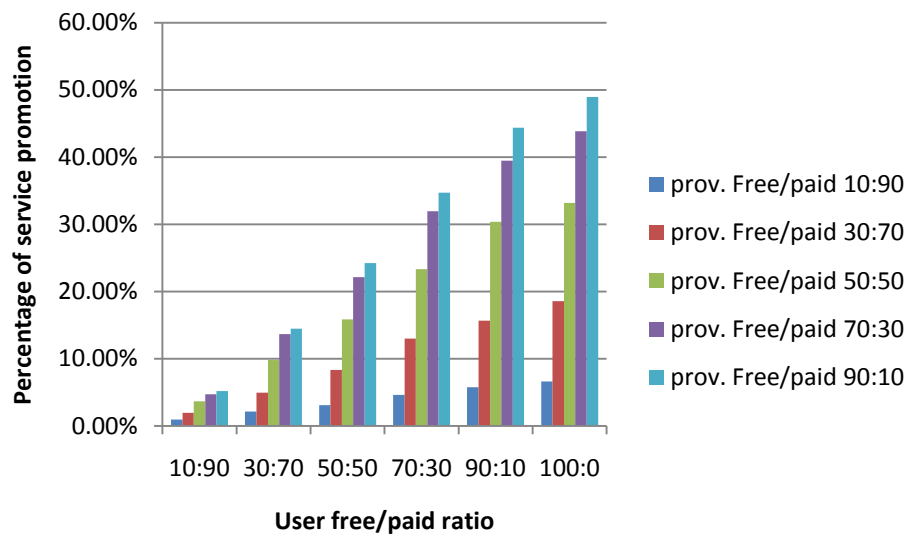


Figure 4.2: Percentage of service promotion with all ratios of providers

As we can see in the figure, with all provider ratios, when there are more free users, the promotion percentage is increasing, which is consistent with the 50:50 case. Also, for each fixed user ratio, when there are more free providers, the promotion percentage is increasing. This is because if there are more free providers, there are less paid providers, and consequently the chance of promoting their services becomes higher.

The promotion percentage of unique paid services is calculated as the number of unique services promoted divided by the total number of services promoted. Figure 4.3 shows the

change in the percentage values when the ratio between the free/paid users is changing, and here the ratio between free/paid providers is 50:50).

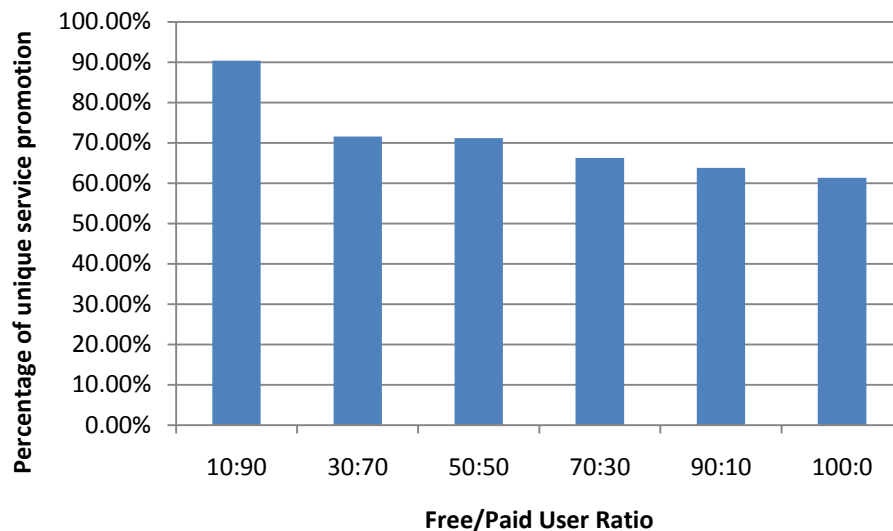


Figure 4.3: Percentage of unique service promotion when the free/paid user ratio changes (free/paid provider ratio: 50:50)

When the ratio of free and paid provides is fixed and the ratio between free and paid users increases, the percentage of unique service promotion is decreasing. The possible explanation is that if more free users request the service, the probability of promoting services is higher, and there is a higher chance that the same service may be promoted multiple times, therefore, smaller percentage of distinct services will be promoted.

The percentage values for all free/paid provider ratios are shown in Figure 4.4.

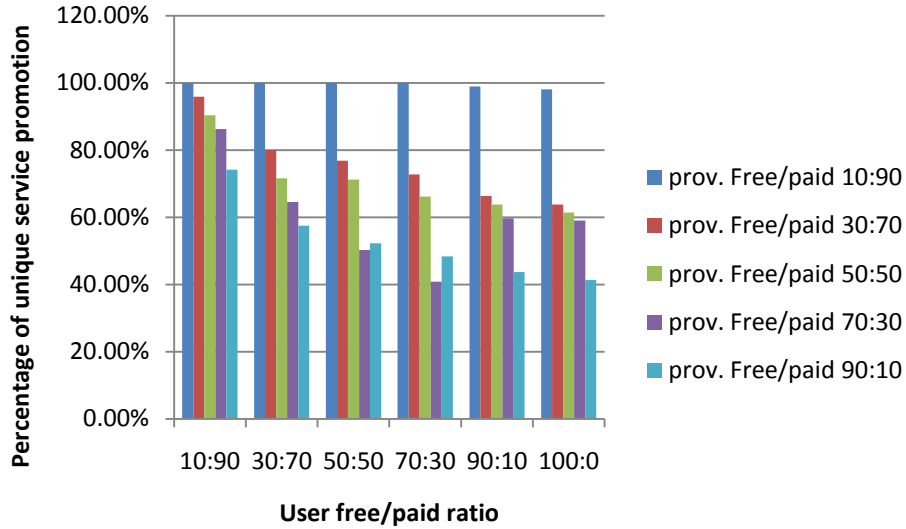


Figure 4.4: Percentage of unique service promotion with all ratios of providers

As we can see in the figure, with almost all provider ratios, when there are more free users, the promotion percentage is decreasing, which is consistent with the 50:50 case. The only exception is when the ratio between free and paid providers is 10:90. In this case, considering that there are more paid providers, which means a smaller chance a free service is originally in the top position, each time when promotion happens, there is a very high probability (close to 100%) a distinct service is promoted. We can also see that for each fixed user ratio, when there are more free providers, the distinct promotion percentage is decreasing. This is because if there are more free providers, there are less paid providers, and consequently the chance of promoting distinct services becomes lower. In some cases, this percentage value is not strictly decreasing, mainly because in our simulated dataset, a user can pick 1 to 5 requests randomly and thus the numbers of requests are not constant.

The promotion percentage of unique paid providers is calculated as the number of unique providers promoted divided by the total number of paid providers. Figure 4.5 shows the change

in the percentage values when the ratio between the free/paid users is changing, and here the ratio between free/paid providers is 50:50).

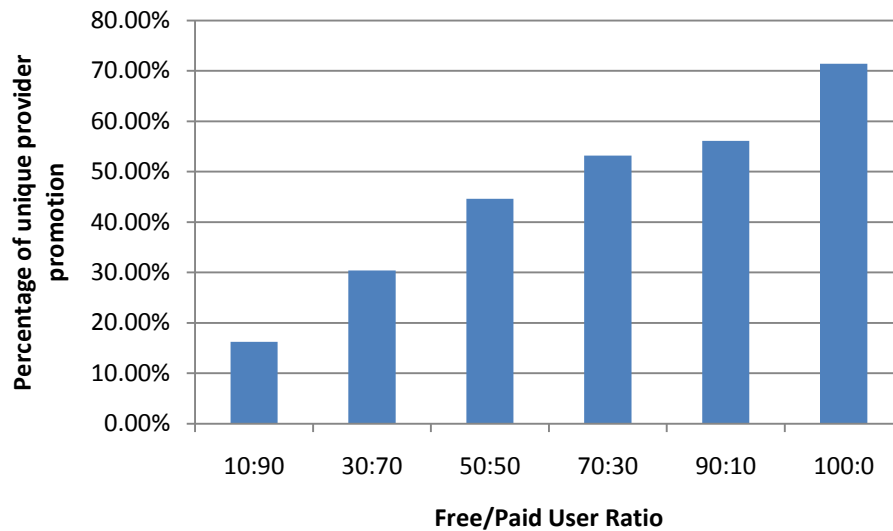


Figure 4.5: Percentage of unique provider promotion when the free/paid user ratio changes
(free/paid provider ratio: 50:50)

When the ratio of free and paid providers is fixed and the ratio between free and paid user increases, the percentage of unique provider promotion is increasing. If more free users request the service, the probability of promoting services from different providers is also higher.

The percentage values for all free/paid provider ratios are shown in Figure 4.6.

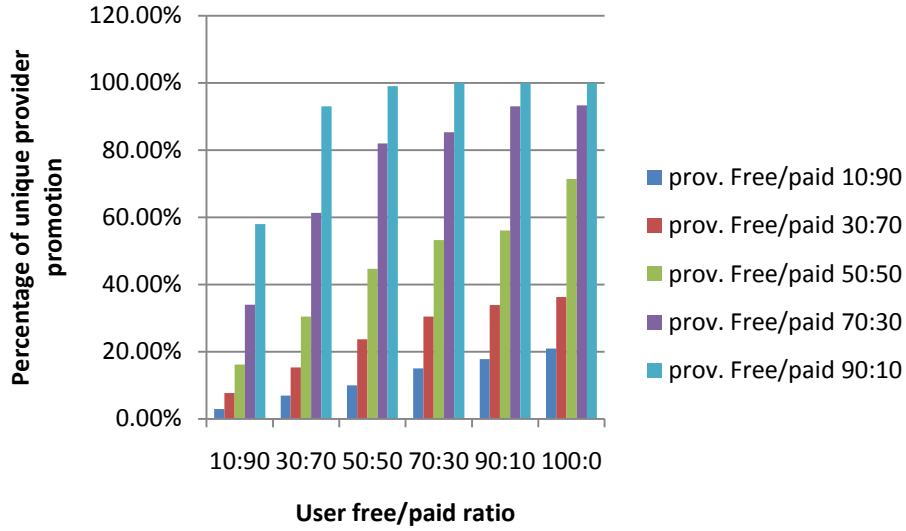


Figure 4.6: Percentage of unique provider promotion with all ratios of providers

As we can see in the figure, with all provider ratios, when there are more free users, the promotion percentage is increasing, which is consistent with the 50:50 case. Also, for each fixed user ratio, when there are more free providers, the promotion percentage is increasing. This is because if there are more free providers, there are less paid providers, and consequently the chance of promoting services from distinct providers becomes higher.

4.2.2 Analysis of Rank Change of Free Services

When the system promotes a paid service, a free service is demoted. To measure how fair our system treats the free services, we define the percentage of rank change of free services as the number of times a top-ranked free service demoted divided by the number of free services on top. Figure 4.7 shows the change in the percentage values when the ratio between the free/paid users is changing, and here the ratio between free/paid providers is 50:50).

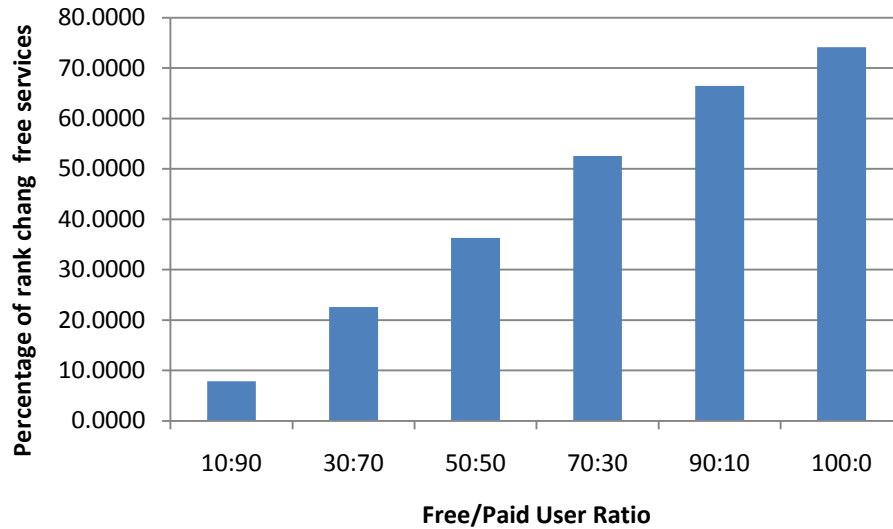


Figure 4.7: Percentage of rank change of free services when the free/paid user ratio changes (free/paid provider ratio: 50:50)

When the ratio of free and paid providers is fixed and the ratio between free and paid user increases, the percentage of rank change of free services is increasing. This is because if more free users request the service, the probability for the original top-ranked service to get demoted is also higher.

The percentage values for all free/paid provider ratios are shown in Figure 4.8.

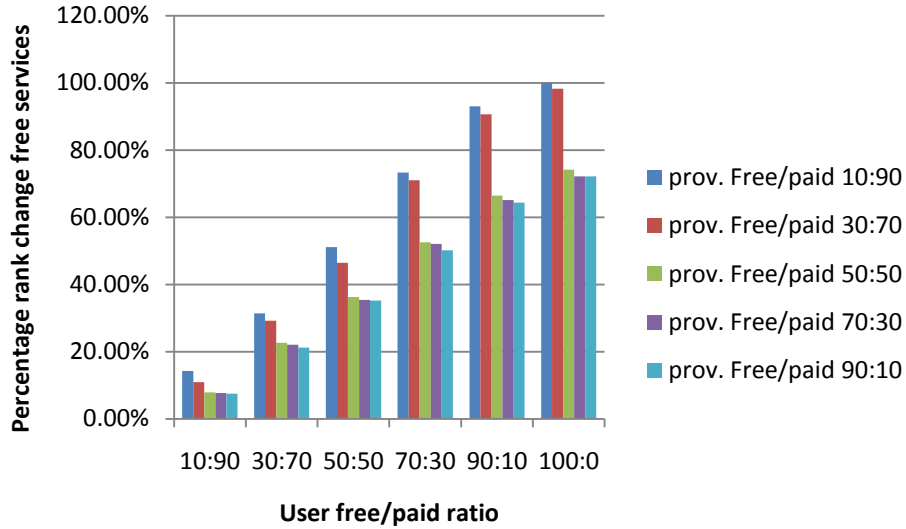


Figure 4.8: Percentage of rank change of free services with all ratios of providers

As we can see in the figure, with all provider ratios, when there are more free users, the rank change percentage is increasing, which is consistent with the 50:50 case. For each fixed user ratio, when there are more free providers, the rank change percentage is decreasing. This is because if there are more free providers, there are less paid providers, and consequently the chance of promoting paid services or the chance of demoting free services becomes smaller. The detailed results are shown in Appendix A.11-A.15.

4.2.3 Analysis of Impact on Free and Paid Providers and Users

In this part, we show the score difference between optimal service and promoted service so that we can see by doing service promotion, how the user satisfaction degree on the recommended service is affected. The bigger the difference, the user will be less satisfied with the selection result. Figure 4.9 shows the percentage of score difference between optimal

services and promoted services. The percentage value is calculated as the absolute difference between the two scores divided by the optimal score.

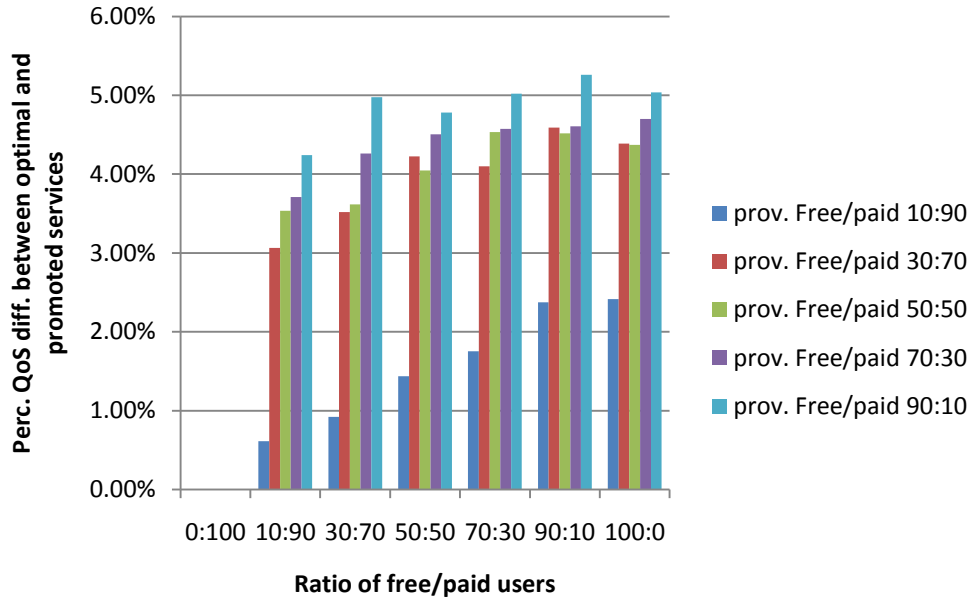


Figure 4.9: Percentage of QoS difference between optimal and promoted services

As we can see, the difference is usually small, between 0.61% and 5.26%. When there are more free users, the percentage value is increasing, because there are more services promoted, which makes QoS difference bigger. And when there are more free providers, the percentage value is generally increasing. This is because if there are more free providers and thus less paid providers, there is a higher chance that the less optimal services will be promoted and therefore the higher difference. In some cases, the percentage of this difference score is slightly decreased because users choose the request randomly and sometimes more free users choose more requests. The detailed results are shown in Appendix Table A.16.

4.2.4 Analysis of Fairness and Promotion score

Figure 4.10 and Figure 4.11 show the distribution of average fairness score and average promotion score when the ratio between free and paid providers is 50:50. The result is from one run, x-Axis shows the request number for which a service is promoted, and y-Axis shows the changing average fairness/promotion score when the requests come into the system one by one.

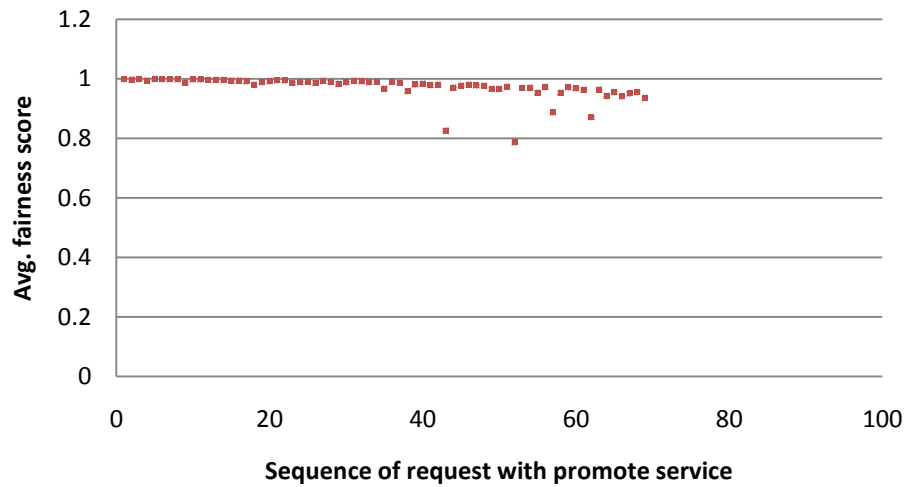


Figure 4.10: Average fairness score

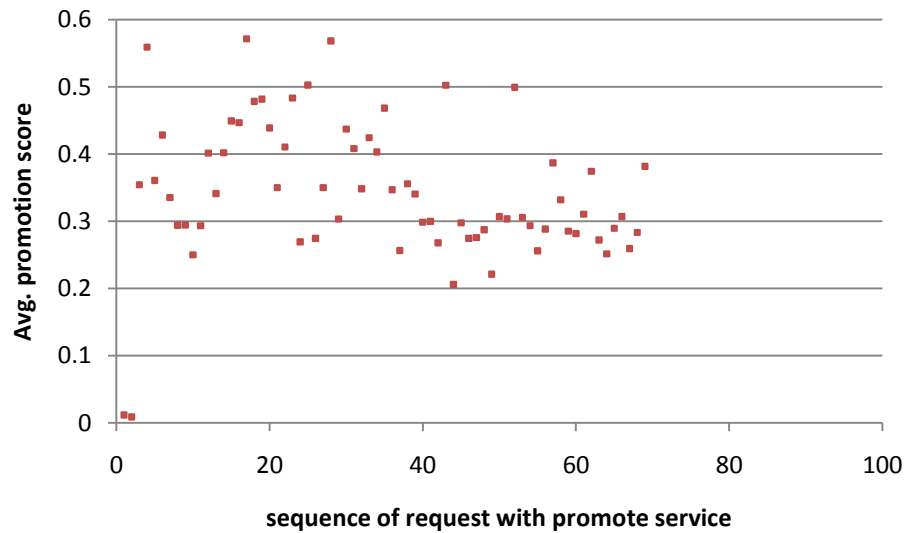


Figure 4.11: Average promotion score

As we can see, the average fairness score ranges from 1 to 0.8, which shows that our algorithm on average maintains a fairly high average score. So, it indicates the algorithm treats the free services in a fair manner and the impact is negligible. As for the average promotion scores, they range between 0.2 and 0.5, and overall the score is stable. A small average score means all the services have got chances to be promoted at a certain point. So our results show that more services are promoted by this process and thus more revenues are generated in the marketplace.

The average execution time to run our algorithm is 82.65ms. Compared to the running time on the baseline QoS-only algorithm which is 1.62ms, it is bigger but in a tolerable range. Since our algorithm could bring more benefits to providers as well as the marketplace, which is not considered in the baseline algorithm, we consider the increased running time is acceptable.

4.3 Summary

In this chapter, we explained our experiment design and the process we followed to generate the simulated datasets. We evaluated and analyzed our algorithm and proved that our algorithm has the capability to promote services so that more services will have chance to be selected and invoked by service consumers. By doing this, revenue can be generated from paid providers. If users want to see the organic search result only, they can choose to become paid users. This option can also generate revenue for the marketplace. The result showed that when there are more paid users, fewer services are promoted such that more revenue is generated from users and less revenue is generated from providers. Similarly, when there are less paid users,

more services are promoted such that more revenue is generated from providers and less revenue is generated from users. In this way, the marketplace can always keep high overall revenue.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

The web services which match with user's functional requirements need to be ranked based on user's non-functional requirements. There are different criteria used as non-functional requirements such as response time, availability, latency, reliability, successibility etc. For the selection and ranking of web services, most of the current research works consider only one party – the service users or consumers. There are also research works on service bidding and auction which actually takes service providers' perspectives and the main goal is to achieve optimized resource allocation and utilization. In a service marketplace where there are multiple parties involved, including the marketplace, providers and consumers, a selection algorithm considering only one party's benefit would not be sufficient and accurate. Therefore, in this work, we have proposed and designed a service selection approach which considers all parties.

Our proposed model has the following contributions:

- We developed a profit model for the service marketplace. The profit can be generated through paid users. If a marketplace is healthy and competitive, and there are many good services hosted, more users can be attracted to the marketplace and more users are willing to pay fees. The profit can also be generated through paid providers. If a marketplace offers mechanisms to promote services for the providers and there are many active users in the market, more providers can be attracted to the marketplace and are willing to pay promotion fees. Advertisements for promoting services can be another source of profit.
- We developed a selection algorithm to consider multiple perspectives: users, providers and the marketplace. We compared the results using QoS-only objective function and

QoS-Plus-FP objective function, and tested the results in different settings. The result showed that our approach provided benefits for all parties because paid users can always get best services in terms of their overall QoS values, free users can still get good services satisfying all their requirements without paying any fee, paid providers can promote their services so that they can get more users using their services, free providers can still attract users as long as their services are good, and finally the marketplace can generate more revenue.

5.2 Future Work

We would like to work on the following directions in the future.

- Firstly, we could implement a fully-functioning service marketplace and collect real data to test our selection model.
- Secondly, the current implementation processes one request at a time. We could try to process more than one request at a time so that any number of requests from any users could be processed in parallel.
- Finally, we would like to improve our algorithm to make it more effective in terms of its promotion power to paid services and fairness to free services.

APPENDICES

APPENDIX A

Table A.1: Measurement of QoS-only and our algorithm on free/paid provider 10:90

	QoS-only		Our algorithm						
Users Free/Paid	No. of Free services on top	No. of Paid services on top	No. of Free services on top	No. of Paid services on top without promotion	No. of Paid services on top by promotion	No. of unique service promoted	No of unique provider promoted	Avg. Score diff. between optimal and promoted service	Avg. Optimal score
0:100	17.7	277.7							
10:90	18.6	269.2	16	269.2	2.6	2.6	2.6	0.00586	0.9562
30:70	19.9	276.5	13.6	276.5	6.3	6.3	6.2	0.00882	0.95553
50:50	18.2	280.2	9	280.2	9.2	9.2	9	0.01359	0.94652
70:30	19	281	5.2	281	13.8	13.8	13.5	0.01691	0.96361
90:10	18.5	282	1.3	282	17.2	17	16	0.02256	0.95057
100:0	20.3	286.3	0	286.3	20.3	19.9	18.8	0.0227	0.939013

Table A.2: Measurement of QoS-only and our algorithm on free/paid provider 30:70

	QoS-only		Our algorithm						
Users Free/Paid	No. of Free services on top	No. of Paid services on top	No. of Free services on top	No. of Paid services on top without promotion	No. of Paid services on top by promotion	No. of unique service promoted	No of unique provider promoted	Avg. Score diff. between optimal and promoted service	Avg. Optimal score
0:100	55.9	240.8							
10:90	53.4	249.7	47.6	249.7	5.8	5.5	5.4	0.0295	0.9626
30:70	49.8	241.3	35.5	241.3	14.3	11.5	10.7	0.0342	0.97162
50:50	52.9	243.1	28.2	242.6	24.7	18.9	16.6	0.0397	0.94007
70:30	55	246.4	15.9	246.4	39.1	28.3	21.3	0.04	0.97608
90:10	53.5	265.1	4.9	265.1	48.5	32.1	23.7	0.0439	0.95616
100:0	57.3	245.6	1	245.6	56.3	35.7	25.4	0.0427	0.97318

Table A.3: Measurement of QoS-only and our algorithm on free/paid provider 50:50

	QoS-only		Our algorithm						
Users Free/Paid	No. of Free services on top	No. of Paid services on top	No. of Free services on top	No. of Paid services on top without promotion	No. of Paid services on top by promotion	No. of unique service promoted	No of unique provider promoted	Avg. Score diff. between optimal and promoted service	Avg. Optimal score
0:100	131.4	167							
10:90	133.9	165.7	123.4	165.7	10.5	9.5	8.1	0.0326	0.92371
30:70	128.1	166	99.2	166	28.9	20.6	15.2	0.0348	0.96203
50:50	132.5	170.3	84.6	170.3	47.9	34.1	22.3	0.0385	0.95117
70:30	129.1	162.5	61.6	162.5	67.8	44.9	26.6	0.0425	0.93783
90:10	138.1	164.7	46.2	164.7	91.9	58.6	33.8	0.0438	0.96931
100:0	138.7	168.4	35.9	168.4	102.8	62.8	35.7	0.0422	0.96551

Table A.4: Measurement of QoS-only and our algorithm on free/paid provider 70:30

	QoS-only		Our algorithm						
Users Free/Paid	No. of Free services on top	No. of Paid services on top	No. of Free services on top	No. of Paid services on top without promotion	No. of Paid services on top by promotion	No. of unique service promoted	No of unique provider promoted	Avg. Score diff. between optimal and promoted service	Avg. Optimal score
0:100	176	121.5							
10:90	184.1	115.7	170	115.7	14.1	12.2	10.2	0.0349	0.94061
30:70	184.7	112.7	144.1	112.7	40.6	26	18.4	0.0404	0.94828
50:50	190.1	114.7	122.8	114.7	67.3	33.7	24.6	0.0419	0.92991
70:30	182.6	115.7	87.4	115.7	95.2	38.4	25.6	0.0445	0.9727
90:10	180.3	117.2	62.9	117.2	117.4	70	27.9	0.0442	0.95911
100:0	182.9	118.2	50.7	118.2	132.2	77.7	28	0.0455	0.96792

Table A.5: Measurement of QoS-only and our algorithm on free/paid provider 90:10

	QoS-only		Our algorithm						
Users Free/Paid	No. of Free services on top	No. of Paid services on top	No. of Free services on top	No. of Paid services on top without promotion	No. of Paid services on top by promotion	No. of unique service promoted	No of unique provider promoted	Avg. Score diff. between optimal and promoted service	Avg. Optimal score
0:100	203.6	90.8							
10:90	210.8	94.3	195	94.3	15.8	11.7	5.8	0.0412	0.971048
30:70	209.9	98	165.3	98	44.6	25.4	9.3	0.0463	0.93034
50:50	200.1	90.4	129.8	90.4	70.3	36.6	9.9	0.0459	0.96023
70:30	201.8	90.2	100.5	90.2	101.3	48.7	10	0.0468	0.93223
90:10	205.3	92.8	73.1	92.8	132.2	57.6	10	0.05	0.95058
100:0	209.6	99.1	58.3	99.1	151.3	62.6	10	0.0481	0.95486

Table A.6: Promotion of paid services and providers on free/paid provider ratio 10:90

Free/Paid User ratio	Service Promotion Percentage		Unique Service Promotion Percentage		Unique Provider Promotion Percentage	
	Percentage	Standard Deviation	Percentage	Standard Deviation	Percentage	Standard Deviation
10:90	0.95%	0.0050	100.00%	0.0000	2.89%	0.0100
30:70	2.14%	0.0078	100.00%	0.0000	6.89%	0.0190
50:50	3.09%	0.0061	100.00%	0.0000	10.00%	0.0160
70:30	4.60%	0.0097	100.00%	0.0000	15.00%	0.0290
90:10	5.74%	0.0100	98.91%	0.0210	17.78%	0.0250
100:0	6.62%	0.0049	98.05%	0.0320	20.89%	0.0184

Table A.7: Promotion of paid services and providers on free/paid provider ratio 30:70

Free/Paid User ratio	Service Promotion Percentage		Unique Service Promotion Percentage		Unique Provider Promotion Percentage	
	Percentage	Standard Deviation	Percentage	Standard Deviation	Percentage	Standard Deviation
10:90	1.91%	0.0119	95.89%	0.0629	7.71%	0.0130
30:70	4.91%	0.0082	79.95%	0.0314	15.29%	0.0326
50:50	8.33%	0.0146	76.80%	0.0934	23.71%	0.0405
70:30	12.96%	0.0165	72.74%	0.0627	30.43%	0.0474
90:10	15.64%	0.0288	66.35%	0.0711	33.86%	0.0326
100:0	18.57%	0.0205	63.75%	0.0487	36.29%	0.0264

Table A.8: Promotion of paid services and providers on free/paid provider ratio 50:50

Free/Paid User ratio	Service Promotion Percentage		Unique Service Promotion Percentage		Unique Provider Promotion Percentage	
	Percentage	Standard Deviation	Percentage	Standard Deviation	Percentage	Standard Deviation
10:90	3.64%	0.0067	90.37%	0.0928	16.20%	0.0426
30:70	9.81%	0.0079	71.57%	0.0478	30.40%	0.0195
50:50	15.82%	0.0085	71.17%	0.0594	44.60%	0.0438
70:30	23.31%	0.0183	66.19%	0.0581	53.20%	0.0560
90:10	30.37%	0.0288	63.81%	0.0851	56.09%	0.0549
100:0	33.48%	0.0218	61.34%	0.0439	71.40%	0.0369

Table A.9: Promotion of paid services and providers on free/paid provider ratio 70:30

Free/Paid User ratio	Service Promotion Percentage		Unique Service Promotion Percentage		Unique Provider Promotion Percentage	
	Percentage	Standard Deviation	Percentage	Standard Deviation	Percentage	Standard Deviation
10:90	4.70%	0.0085	86.29%	0.0734	34.00%	0.0771
30:70	13.65%	0.0146	64.49%	0.0625	61.33%	0.0452
50:50	22.11%	0.0266	50.20%	0.0218	82.00%	0.0400
70:30	31.91%	0.0401	40.83%	0.0448	85.33%	0.0452
90:10	39.46%	0.0136	59.70%	0.0377	93.00%	0.0233
100:0	43.83%	0.0337	59.02%	0.0272	93.33%	0.0149

Table A.10: Promotion of paid services and providers on free/paid provider ratio 90:10

Free/Paid User ratio	Service Promotion Percentage		Unique Service Promotion Percentage		Unique Provider Promotion Percentage	
	Percentage	Standard Deviation	Percentage	Standard Deviation	Percentage	Standard Deviation
10:90	5.17%	0.0098	74.18%	0.0665	58.00%	0.1240
30:70	14.47%	0.0142	57.50%	0.0597	93.00%	0.0640
50:50	24.22%	0.0205	52.24%	0.0413	99.00%	0.0300
70:30	34.68%	0.0267	48.30%	0.0342	100.00%	0.0000
90:10	44.37%	0.0392	43.63%	0.0216	100.00%	0.0000
100:0	48.95%	0.0186	41.33%	0.0241	100.00%	0.0000

Table A.11: Rank change of free services on free/paid provider ratio 10:90

Free/Paid User ratio	Service Rank Change Percentage	
	Percentage	Standard Deviation
10:90	14.27%	0.0536
30:70	31.39%	0.0592
50:50	51.09%	0.1011
70:30	73.30%	0.1152
90:10	92.98%	0.0302
100:0	100.00%	0.0000

Table A.12: Rank change of free services on free/paid provider ratio 30:70

Free/Paid User ratio	Service Rank Change Percentage	
	Percentage	Standard Deviation
10:90	10.87%	0.0205
30:70	29.18%	0.0596
50:50	46.43%	0.0484
70:30	71.00%	0.0380
90:10	90.60%	0.0401
100:0	98.23%	0.0141

Table A.13: Rank change of free services on free/paid provider ratio 50:50

Free/Paid User ratio	Service Rank Change Percentage	
	Percentage	Standard Deviation
10:90	7.85%	0.0155
30:70	22.60%	0.0232
50:50	36.28%	0.0279
70:30	52.57%	0.0282
90:10	66.45%	0.0457
100:0	74.10%	0.0226

Table A.14: Rank change of free services on free/paid provider ratio 70:30

Free/Paid User ratio	Service Rank Change Percentage	
	Percentage	Standard Deviation
10:90	7.66%	0.0131
30:70	22.08%	0.0288
50:50	35.42%	0.0324
70:30	52.01%	0.0512
90:10	65.12%	0.0301

100:0	72.16%	0.0255
-------	--------	--------

Table A.15: Rank change of free services on free/paid provider ratio 90:10

Free/Paid User ratio	Service Rank Change Percentage	
	Percentage	Standard Deviation
10:90	7.48%	0.0136
30:70	21.23%	0.0205
50:50	35.18%	0.0271
70:30	50.19%	0.0357
90:10	64.34%	0.0264
100:0	72.15%	0.0263

Table A.16: Percentage of score difference between optimal and promoted services

Users Free/Paid	Prov. Free/paid 10:90	Prov. Free/paid 30:70	Prov. Free/paid 50:50	Prov. Free/paid 70:30	Prov. Free/paid 90:10
0:100	0.00%	0.00%	0.00%	0.00%	0.00%
10:90	0.61%	3.06%	3.54%	3.71%	4.24%
30:70	0.92%	3.52%	3.62%	4.26%	4.98%
50:50	1.44%	4.22%	4.05%	4.51%	4.78%
70:30	1.75%	4.10%	4.53%	4.57%	5.02%
90:10	2.37%	4.59%	4.52%	4.61%	5.26%
100:0	2.42%	4.39%	4.37%	4.70%	5.04%

Table A.17: Average fairness score and promotion score

S. No.	AvgFairness	AvgPromotion
1	0.99845	0.01163
2	0.99691	0.00872
3	0.99935	0.35410
4	0.99394	0.55882
5	0.99877	0.36060
6	0.99833	0.42821
7	0.99864	0.33510
8	0.99815	0.29382
9	0.98725	0.29412
10	0.99754	0.24979
11	0.99732	0.29317
12	0.99634	0.40129
13	0.99621	0.34112
14	0.99599	0.40169
15	0.99367	0.44900
16	0.99334	0.44640
17	0.99177	0.57121

18	0.97969	0.47798
19	0.98796	0.48141
20	0.99161	0.43860
21	0.99400	0.34972
22	0.99416	0.41030
23	0.98704	0.48306
24	0.98791	0.26900
25	0.99066	0.50236
26	0.98623	0.27442
27	0.99261	0.34985
28	0.98862	0.56800
29	0.98282	0.30308
30	0.98857	0.43695
31	0.99224	0.40786
32	0.99131	0.34825
33	0.98830	0.42410
34	0.98997	0.40288
35	0.96641	0.46822
36	0.99012	0.34663
37	0.98680	0.25624
38	0.95781	0.35550
39	0.98087	0.34024
40	0.98354	0.29840
41	0.97829	0.29960
42	0.98010	0.26756
43	0.82576	0.50222
44	0.96981	0.20563
45	0.97694	0.29747
46	0.97959	0.27438
47	0.97728	0.27548
48	0.97646	0.28733
49	0.96594	0.22093
50	0.96571	0.30675
51	0.97308	0.30328
52	0.78636	0.49909
53	0.96951	0.30545
54	0.96986	0.29317
55	0.95335	0.25567
56	0.97206	0.28802
57	0.88725	0.38670
58	0.95315	0.33170
59	0.97127	0.28505
60	0.96743	0.28137
61	0.96214	0.31027
62	0.87059	0.37415
63	0.96387	0.27189
64	0.94201	0.25118
65	0.95527	0.28925
66	0.94087	0.30670
67	0.95099	0.25923
68	0.95472	0.28319

69	0.93586	0.38138
----	---------	---------

REFERENCES

- [1] S. Kamboj, and N. S. Ghumman, “A Survey on Cloud Computing and its Types”, in Proceedings of the International Conference on Computing for Sustainable for Global Development, New Delhi, India, pp. 2971-2974, 2016
- [2] S. Banerjee Z. Zhou, and R. Johari, “The Importance of Exploration in Online Marketplace”, in Proceedings of the IEEE conference on Decision and Control, California, USA, pp. 3499-3504, 2014
- [3] E. Pejman, Y. Rastegari, P. Majlesi Esfahani, and A. Salajegheh, “Web Service Composition Methods: A Survey”, in Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, pp. 1-5, 2012
- [4] E. Al-Masri, and Q.H. Mahmoud, "QoS-based Discovery and Ranking of Web Services", in Proceedings of the 16th International Conference on Computer Communications and Networks, HI, USA, pp. 529-534, 2007.
- [5] W. Serrai, A. Abdelli, L. Mokdad, and Y. Hammal, "An efficient approach for Web service selection", in Proceedings of the Symposium on Computers and Communication (ISCC), Messina, Italy, pp. 167-172, 2016.
- [6] K. Shade, Awodele, A. Ronke, and O. Samule, “Quality of Service (QoS) Issues in Web Services”, IJCSNS International Journal of Computer Science and Network Security, 12(1), pp. 94-97, 2012

- [7] R. Vigne, W. Mach, and E. Schikuta, "Towards the Smart Web Service Marketplace", in Proceedings of the IEEE International Conference on Business Informatics, Vienna, Austria, pp. 208-215, 2013
- [8] O. L. D. Rugnon, M. A.T. Escuderv, J. L. Rueda, and S. Shanmugalingam, "Towards Dynamic Business Models on Marketplace Environment", in Proceedings of the 14th International Conference on Next Generation Networks", Berlin, Germany, pp. 1-6, 2010.
- [9] B. Yoo, V. Choudhary, and T. Mukhopadhyay "Marketpalces or Web Services? Alternate Business Models for Electronic B2B Transactions", in Proceedings of the 44th Hawaii International Conference on System Sciences, HI, USA, pp. 1-10, 2010
- [10] J. Yan, and J. Piao, "Towards QoS-based Web Service Discovery", in Proceedings of the International Conference on Service Oriented Computing, Sydney, Australia, pp. 200-210, 2008.
- [11] Y.T. Liu, A.H.H. Ngu, L.Z. Zeng, "QoS Computation and Policing in Dynamic Web Service Selection", in Proceedings of the International Conference on World Wide Web, New York, USA, pp. 66-73, 2004.
- [12] D.A. Menasce and V. Dubey, "Utility-based QoS Brokering in Service Oriented Architectures", in Proceedings of the 5th IEEE International Conference on Web Services, UT, USA, pp. 422-430, 2007.
- [13] S. Lamparter, A. Ankolekar, R. Studer, and S. Grimm, "Preference-based Selection of Highly Configurable Web Services", in Proceedings of the 16th International Conference on World Wide Web, Alberta, Canada, pp. 1013-1022, 2007.
- [14] L. Qu, Y. Wang, M.A. Orgun, L. Liu, and A. Bouguettaya, "Context-aware Cloud Service Selection Based on Comparison and Aggregation of User Subjective Assessment and Objective

Performance Assessment”, in Proceedings of the 21th International Conference on Web Services, AK, USA, pp. 81-88, 2014.

[15] A. Ruiz-Cortés, O. Martín-Díaz, A.D. Toro, and M. Toro, "Improving the Automatic Procurement of Web Services Using Constraint Programming", International Journal on Cooperative Information Systems, 14(4), pp. 439-468, 2005.

[16] K. Kritikos, and D. Plexousakis, “Mixed Integer Programming for QoS Based Web Service Matchmaking”, in Proceedings of IEEE Transactions of Service Computing, pp. 122-139, 2009

[17] Q. Ma, H. Wang, Y. Li, G. Xie, and F. Liu, "A Semantic QoS-aware Discovery Framework for Web Services", in Proceedings of the IEEE International Conference on Web Services, HI, USA, pp.129-136, 2008.

[18] V.X. Tran, H. Tsuji, and R. Masuda, "A New QoS Ontology and its QoS-based Ranking Algorithm for Web Services", Simulation Modelling Practice and Theory, 17(8), 1378-1398, 2009.

[19] S. K. Greg, R. Versteeg, and R. Buyya, “SMICloud: a Framework for Computing and Ranking Cloud Services”, in Proceedings of the IEEE International Conference on Utility and Cloud Computing, Melbourne, Australia, pp. 529-534, 2011.

[20] C. Herssens, I.J. Jureta, and S. Faulkner, "Dealing with Quality Tradeoffs during Service Selection", in Proceedings of the International Conference on Autonomic Computing, Chicago, USA, pp. 77-86, 2008.

[21] Z. Rehhman, O.K. Hussain, and F.K. Hussain, “Parallel Cloud Service Selection and Ranking based on QoS History”, International Journal of Parallel Programming, 42(5), pp. 820-852, 2014.

- [22] Q. Yu, and A. Bouguettaya, "Computing Service Skyline from Uncertain QoWS", *IEEE Transactions on Services Computing*, 3(1), 16-29, 2010.
- [23] D. Skoutas, D. Sacharidis, A. Simitsis, V. Kentere, and T. Sellis, "Top-k Dominant Web Services Under Multi-Criteria Matching", in *Proceedings of the 12th International Conference on Extending Database Technology*, New York, USA, pp. 898-909, 2009.
- [24] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services", *IEEE Transactions on Parallel and Distributed Systems*, 24(6): 1213-1222, 2013.
- [25] R. Karthiban, "A QoS-aware Web Service Selection Based on Clustering", *International Journal of Scientific and Research Publications*, 4(2), pp. 1-5, 2014
- [26] E. Lim, P. Thiran, Z. Maamar, and J. Bentahar, "On the Analysis of Satisfaction of Web Services Selection", in *Proceedings of IEEE Ninth International Conference on Service Computing*, HI, USA, pp. 122-129, 2012
- [27] Z. Zhang, R. Ma, J. Ding, and Y. Yang, "ABACUS: An Auction-Based Approach to Cloud Service Differentiation", in *Proceedings of the IEEE International Conference on Cloud Engineering*, CA, USA, pp. 292-301, 2013.
- [28] U. Lampe, M. Siebenhaar, A. Papageorgiou, D. Schuller, and R. Steinmetz, "Maximizing Cloud Provider Profit from Equilibrium Price Auctions", in *Proceedings of the IEEE Fifth International Conference on Cloud Computing*, HI, USA, pp. 83-90, 2012.
- [29] S. Tang, J. Yuan, and X. Li, "Towards Optimal Bidding Strategy for Amazon EC2 Cloud Spot Instance", in *Proceedings of the IEEE Fifth International Conference on Cloud Computing*, HI, USA, pp. 91-98, 2012.

- [30] L. M. Leslie, Y. C. Lee, P. Lu, and A.Y. Zomaya, "Exploiting Performance and Cost Diversity in the Cloud", in Proceedings of the IEEE Sixth International Conference on Cloud Computing, CA, USA, pp. 107-114, 2013.
- [31] S. Shang, J. Jiang, Y. Wu, G. Yang, and W. Zheng, "A Knowledge-based Continuous Double Auction Model for Cloud Market", in Proceedings of the Sixth International Conference on Semantics, Knowledge and Grids, Ningbo, China, pp. 129-134, 2010.
- [32] S. Shang, J. Jiang, Y. Wu, and Z. Huang, "DABGPM: A Double Auction Bayesian Game-Based Pricing Model in Cloud Market", in Proceedings of the IFIP International Federation for Information Processing, pp. 155-164, 2010.
- [33] I. Fujiwara, K. Aida, and I. Ono, "Applying Double-sided Combinational Auctions to Resource Allocation in Cloud Computing", in proceedings of the IEEE 10th Annual International Symposium on Applications and Internet, Seoul, Korea, pp. 7-14, 2010.
- [34] P. Murali, Y. Li, P. Mazzoleni, and R. Vaculin, "Optimal Budget Allocation Strategies for Real Time Bidding in Display Advertising", in Proceedings of the IEEE Winter Simulation Conference, CA, USA, pp. 3146-3147, 2015.
- [35] M. Chen, and Z. Zhang, "A Bid and Auction Mechanism for Resource Management in Project Portfolios", in Proceedings of the IEEE IEEM, Singapore, pp. 1297-1301, 2015.
- [36] Z. Li, M. Kihi, and A. Robertsson, "On a Feedback Control-based Mechanism of Bidding for Cloud Spot Service", in Proceedings of the IEEE 7th International Conference on Cloud Computing Technology and Science, Vancouver, Canada, pp. 290-297, 2015.
- [37] L. Barreto, J. Franga, and F. Siqueira, "Conceptual Model of Brokerage Authentication in Cloud Federations", in Proceedings of the IEEE 4th International Conference on Cloud Networking, Pisa, Italy, pp. 303-308, 2015.

- [38] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-Aware Service Selection for Service-based Systems Based on Iterative Multi-Attribute Combinatorial Auction", *IEEE Transactions on Software Engineering*, DOI 10.1109/TSE.2013.3297911, 2013.
- [39] J. Park, Y. An, and K. Yeom, "Virtual Cloud Bank: Cloud Service Broker for Intermediating Services Based on Semantic Analysis Models", in *Proceedings of the IEEE Conference on Ubiquitous and Computing*, Beijing, China, pp. 1022-1029, 2015.
- [40] M. Mohabey, Y. Nanahari, S. Mallick, P. Suresh, and S. V. Subrahmanya, "An Intelligent Procurement Marketplace for WebServices Composition", in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, CA, USA, pp. 551-554, 2007.
- [41] D. Sachan, S. K. Dixit, and S. Kumar, "QoS Aware Formalized, Model for Semantic Web Service Selection", *International Journal of Web and Semantic Technology*, 5(4), pp. 53-100, 2014.