

Available from: http://dx.doi.org/10.1109/SCC.2012.38.

Copyright © 2012 IEEE.

This is the author’s version of the work, posted here with the permission of the publisher for your personal use. No further distribution is permitted. You may also be able to access the published version from your library. The definitive version is available at http://ieeexplore.ieee.org/.
INSC: An Iterative Negotiation Approach for Service Compositions

Qiang He and Yun Yang
Faculty of Information and Communication Technologies
Swinburne University of Technology
Melbourne, Australia 3122
{qhe, yyang}@swin.edu.au

Jun Yan
School of Information Systems and Technology
University of Wollongong, Wollongong, Australia
jyan@uow.edu.au

Hai Jin
Services Computing Technology and System Lab
Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China 430074
hjin@hust.edu.cn

Abstract—The service-oriented paradigm offers support for engineering service-based systems (SBSs) based on service compositions. The selection of services with the aim to fulfil the quality constraints for SBSs and to achieve the optimisation goals is a critical and challenging issue. In particular, when the quality-of-service (QoS) constraints for a SBS are severe, it is often difficult to find an optimal solution for the SBS. Exploiting the competition among service providers can help SBS developers obtain favourable QoS offers for the component services of SBSs and increase the possibility of finding optimal solutions for the SBSs. In this paper, we present a novel joint optimisation and negotiation approach named Iterative Negotiation for Service Composition (INSC) that supports effective and efficient QoS-aware service selection for SBSs. We evaluate INSC experimentally using example SBSs that are synthetically generated based on a real-world Web service dataset. The experimental results show that INSC can significantly and efficiently increase the possibility of finding optimal solutions in severe service composition scenarios.

Keywords-component; Service-based system; service composition; QoS; negotiation

1. INTRODUCTION

The service-oriented paradigm is emerging as a new way to engineer software systems that are composed of and exposed as services. A great advantage of the service-oriented paradigm is its support for service compositions. Through service compositions, SBS providers can compose existing services to construct new Service-based Systems (SBSs) [1, 2]. In service-oriented environments, the candidate services available for composing SBSs often differ in their QoS values. To compose a SBS, the developer needs to select appropriate services that achieve the SBS provider’s optimisation goal, while fulfilling all quality constraints for the SBS [1-4].

When the quality constraints imposed on a SBS are severe, e.g., limited budget and stringent response time limit, it is often difficult to find an optimal solution to the QoS-aware service selection problem for the SBS. Figure 1 presents some example sets of quality constraints on different difficulty levels. Given a target SBS and a set of available candidate services, “severe” quality constraints are apparently harder to fulfil than “medium” and “simple” ones. Negotiation has been employed by many researchers as a means to increase the possibility of finding optimal solutions when an optimal solution for the SBS cannot be found based on the original QoS levels of the candidate services [1, 4-8]. Through bilateral (i.e., one-on-one) negotiations, SBS developers can ask specific service providers to improve the quality of their services so that the quality constraints for the SBSs can be fulfilled. However, in large-scale service composition scenarios, such a process is too difficult to manage. For example, given 1,000 service providers, the SBS developer needs to maintain up to 1,000 bilateral negotiations.

In addition, exiting negotiation approaches for QoS-aware service selection for SBSs have not exploited the competition among service providers properly. With the development and popularity of e-business, e-commerce, especially the pay-per-use business model promoted by Cloud computing [9], there are more and more functionally-equivalent services available at different quality levels. According to [10], there has been a more than 130% increase in the number of published Web services from October 2006 to October 2007. The statistics published by webservices.seekda.com, a Web service search engine, also indicate an exponential growth in the number of published Web services in the past 64 months. Driven by the widespread of Cloud computing, the service-oriented environment is moving rapidly toward a
perfect competition environment [11] in which service providers compete for service contracts [12], i.e., the contracts for provision of services. The fiercer the competition is for a service contract, the more likely the SBS developer will be able to obtain a satisfactory offer for the price and quality of the service. Exploiting the competition among service providers can increase the success rate of finding optimal solutions for SBSs, especially when the quality constraints are imposed on the SBSs are severe. Thus, a SBS developer should negotiate with multiple service providers over each component service of the SBS and then select the best QoS offers. Bilateral negotiations cannot properly exploit the competition among service providers. A service provider that is unaware of its position in the competition is not a well-motivated competitor. Thus, information regarding the current status of the competition should be provided to service providers so that they can analyse their positions in the competition and improve their offers accordingly.

Furthermore, most existing negotiation approaches assume that the local quality constraints for each component service of a SBS can be obtained by decomposing the global quality constraints for the SBS. In the negotiation, they focus only on the component services for which the local quality constraints have not been fulfilled yet. The fact is not considered that even though the local quality constraints for certain component services are fulfilled, the QoS capacities of the service providers for those component services may still be exploitable. This is a major limitation of most existing negotiation approaches.

To address the above issues, this paper presents a novel joint optimisation and negotiation approach called Iterative Negotiation for Service Composition (INSC) to support QoS-aware service selection for SBSs. INSC is a structured iterative one-to-many negotiation process. In each round of INSC, service providers propose their QoS offers. If an optimal solution is found based on the received QoS offers, the SBS developer awards the service contracts to the winning service providers. If an optimal solution cannot be found, i.e., the current QoS offers cannot fulfil the quality constraints for the SBS, the negotiation can iterate, allowing the service providers to improve their QoS offers. In each round, the current winning offers and counter-offers are sent to service providers as guidance on analysing their positions in the competition and improving their QoS offers. A novel approach for the generation of counter-offers (i.e., the QoS offers asked by the SBS developer) is proposed to guarantee that the negotiation proceeds in a right direction towards an optimal solution.

The experimental results demonstrate that INSC can significantly increase the success rate of finding optimal solutions for SBSs in situations where the quality constraints for SBSs are severe. The experimental results also show that the efficiency and scalability of INSC are satisfactory.

The rest of the paper is organised as follows. The next section reviews related work. Section III introduces the procedure of INSC, followed by supporting mechanisms introduced in Section IV. Section V presents experimental evaluation. Finally, Section VI summarises the major contributions of this paper and outlines future work.

II. RELATED WORK

One major benefit of service-oriented paradigm is its support for engineering SBSs through composition of services. In a QoS-aware service composition process, SBS developers need to select services that fulfill quality constraints and achieve SBS providers’ optimisation goals for the SBSs. The research area of QoS-aware service selection for SBSs has been attracting tremendous research attention in recent years [1-6, 8-24]. However, these approaches suffer high failure rates in scenarios where severe quality constraints are imposed on the SBSs.

Negotiation has been adopted in some approaches to better guarantee the quality of SBSs. Authors in [8] propose to introduce a layer between QoS and service consumers’ requirements in order to provide service consumers the QoS control of service systems. Aiming at optimising the overall system utility, they adopt a negotiation approach to help find a proper configuration for a pool of services managed by a single service provider. The limitation of this approach is that it cannot be applied in the scenarios where the component services of a SBS are provided by distributed service providers. In [5], the authors propose a framework for automating Web service contract specification and establishment. Within the framework, functionally equivalent services are ranked according to their abilities to fulfil the quality constraints within the target budget. Services are selected heuristically according to their ranks for target SBSs. A negotiation approach is also provided to maximise the overall system utility. However, this negotiation approach only supports bilateral negotiations and hence cannot properly exploit the competition among service providers in service composition scenarios. In [1], authors formulate the QoS-aware service selection problems as MILP problems and adopt loops peeling for optimisation. When a feasible solution does not exist, a QoS negotiation algorithm is suggested to enlarge the solution space of the optimisation problem. The limitation of the proposed negotiation approach is, similarly to [5], that the negotiation process is implemented as a set of parallel bilateral negotiations and hence the competition among service providers cannot be properly exploited. In [6], the authors propose an agent-based SLA negotiation framework for QoS-aware service composition. In the framework, a SBS developer is represented by a set of agents who negotiate with SLAs with service providers over individual component services of a service composition. Such a negotiation is also essentially a combination of multiple bilateral negotiations. The authors in [4] propose an approach based on the notion of skyline to reduce the complexity of the QoS-aware service composition problem. They introduce an algorithm for assisting service providers in making their services part of the skyline. However, they do not provide
experimental evaluation regarding the effectiveness and efficiency the proposed algorithm.

In order to guarantee or improve the overall quality of the SBS, the above negotiation approaches seek to improve the quality of the component services for which the local QoS constraints are not met via bilateral negotiations. Such negotiation processes become too difficult to manage as the service composition scenarios scale up. Also, those approaches cannot help SBS developers properly exploit the competition among service providers. A SBS developer should be able to negotiate with multiple candidate service providers over each component service of the SBS in a structured one-to-many negotiation process. In addition, in order to properly exploit the competition among service providers, proper guidance should be provided during the negotiation to service providers on analysing their positions in the competition and improving their QoS offers.

To address the above issues, this paper presents Iterative Negotiation for Service Composition (INSC), a novel joint optimisation and negotiation approach that supports effective and efficient QoS-aware service selection for SBSs. In INSC, SBS developers and service providers exchange QoS offers and counter-offers on a round-by-round basis. By exploiting the competition among service providers, INSC can significantly increase the success rate of finding optimal solutions in scenarios where severe quality constraints are imposed on the SBSs.

III. NEGOTIATION PROCEDURE

INSC is a structured iterative negotiation process, which allows service providers to improve their QoS offers round by round. This section presents the procedure of INSC.

To compose a SBS, the SBS developer needs to look up candidate services for each component service of the SBS. This can be achieved by querying service resources, e.g., public UDDI registries, search engines and service portals. The original QoS offers for the candidate services can be obtained by querying the service sources or the service providers. Given a set of candidate services and their original QoS levels, the iterative negotiation can start. The procedure of INSC is depicted in Figure 2, as explained below.

1) Dynamic Minimum Decrement

If an optimal solution cannot be found, the negotiation iterates, allowing the service providers to improve their QoS offers. In order to guarantee that the negotiation proceeds in the right direction towards an optimal solution, the SBS developer generates and sends counter-offers to the service providers as guidance on analysing their positions in competition and improving their QoS offers. Counter-offers specify the QoS offers asked by the SBS developer. Corresponding to a QoS offer asked by the SBS developer, each counter-offer is specified by \((S, CO)\), where \(S\) represents the service that the QoS offer is placed on and \(CO\) represents the base QoS offer asked by the SBS developer from the service provider for \(S\). Based on received counter-offers, service providers improve their QoS offers to increase their chances of winning, i.e., being part of the final solution.

In INSC, the QoS offers for the next round must not be worse than the counter-offers. For example, given the QoS offer \(QoS_i\) proposed by a service provider \(sp_i\) in the current round \((\{s_i\}, \{$110.00\})\), suppose the counter-offer corresponding to that QoS offer is \((\{s_i\}, \{$100.00\})\), the price that \(sp_i\) proposes for \(s_i\) in the next round must not exceed \$100.00. Otherwise, \(sp_i\) will be removed from the negotiation.

Steps 1 to 4 are repeated until one of the stop criteria is met: 1) an optimal solution is found; 2) any service contracts are left unallocated - the QoS capacities (i.e., the best QoS they can offer) of the service providers that compete for those service contracts are exhausted; or 3) the time limit for the negotiation is violated. Sometimes a final deal-sealing round will be triggered to determine the final winning service providers for certain services. The details are provided in Section IV.A.2.

IV. SUPPORTING MECHANISMS

In this section, we present the mechanisms that support INSC, including negotiation constraints, winner determination and counter-offer generation.

A. Negotiation Constraints

1) Dynamic Minimum Decrement

If an optimal solution cannot be found, the negotiation iterates, allowing the service providers to improve their QoS offers. In order to guarantee that the negotiation proceeds in the right direction towards an optimal solution, the SBS developer generates and sends counter-offers to the service providers as guidance on analysing their positions in competition and improving their QoS offers. Counter-offers specify the QoS offers asked by the SBS developer. Corresponding to a QoS offer asked by the SBS developer, each counter-offer is specified by \((S, CO)\), where \(S\) represents the service that the QoS offer is placed on and \(CO\) represents the base QoS offer asked by the SBS developer from the service provider for \(S\). Based on received counter-offers, service providers improve their QoS offers to increase their chances of winning, i.e., being part of the final solution. In INSC, the QoS offers for the next round must not be worse than the counter-offers. For example, given the QoS offer \(QoS_i\) proposed by a service provider \(sp_i\) in the current round \((\{s_i\}, \{$110.00\})\), suppose the counter-offer corresponding to that QoS offer is \((\{s_i\}, \{$100.00\})\), the price that \(sp_i\) proposes for \(s_i\) in the next round must not exceed \$100.00. Otherwise, \(sp_i\) will be removed from the negotiation.
When generating counter-offers for service providers based on their current QoS offers, minimum decrements (minimum increments in the case of positive QoS parameters, such as throughput and availability) measured by percentage are applied, which represent the minimum concessions that service providers have to make to remain in the next round of the negotiation. In a large-scale scenario where the number of service provider is huge, large minimum decrements are usually preferable as they efficiently filter out uncompetitive service providers. On the other hand, small minimum decrements are more frequently in small-scale scenarios because overly large minimum decrements may filter out too many (sometimes all) remaining service providers, decreasing the success rate of finding an optimal solution for the SBS. In a negotiation that lasts for several rounds, the number of remaining service providers usually decreases as the negotiation proceeds. The minimum decrement should be reduced accordingly. INSC adopts a novel mechanism, called dynamic minimum decrement (DMD) to facilitate ask-QoS generation. At the early stage of the negotiation, higher minimum decrements are applied to generate counter-offers. By doing this, DMD efficiently filter uncompetitive service providers, reducing the complexity of the winner determination problem (WDP). As the negotiation proceeds, the minimum decrement decreases to effectively exploit the QoS capacities of the remaining service providers. By doing so, DMD can improve the efficiency of the negotiation without sacrificing the effectiveness. The selection of DMD model for count-offer generation is domain-specific and is determined by the SBS developer. Figure 3 illustrates three example DMD models. As presented, the minimum decrements generated based on model #2 decreases faster over time than that based on models #1 and #3.

2) Final Deal-Sealing Round

A final deal-sealing round is a round for some service providers to have a last chance to improve their QoS offers for certain services. The result from a final deal-sealing round will determine the final winning service providers for those services. If the QoS offers proposed for a service in the final deal-sealing round are still estimated equivalent, the final winning offer for the service will be selected randomly from the those QoS offers. There are two situations where a final deal-sealing round can be held: 1) when a time limit has been set for the negotiation and it is violated; 2) when all the service providers competing for certain services are removed from the negotiation in the same round because they cannot accept the counter-offers, leaving some service contracts unallocated.

B. Winner Determination

Based on received QoS offers, the current winners, i.e., the winning service providers in the current round, are determined. This is a multi-attribute decision making problem, which in the context of this research is referred to as the WDP. Suppose the business process of a SBS S consists of $r$ ($r \geq 1$) component services $K=\{s_1, ..., s_r\}$. Given $r$ ($r \geq 2$) groups of QoS offers (corresponding to $r$ component services) proposed by the service providers, each contains $m$ received QoS offers. Each QoS offer, denoted by a two-tuple $(s_i, Q_i)$, specifies a service $s_i$ and a set of QoS values $Q_i$ offered for $s_i$. The WDP is a constraint optimisation problem (COP) that aims at selecting one QoS offer from each QoS offer group so that the selected QoS offers achieve the SBS developer’s optimisation goal $\text{objective}(S)$, while fulfilling all quality constraints for the SBS $S$ $C=\{c_1, ..., c_l\}$.

To capture the quality constraints for $S$, we first model the WDP as a constraint satisfaction problem (CSP), which consists of a finite set of variables $X=\{X_{ij}, ..., X_{ij}\}$ with respective domains $D=\{D_1, ..., D_s\}$ listing the possible values for each variable, and a set of constraints $C=\{c_1, ..., c_l\}$ over $X$. A solution to a CSP is an assignment of a value to each variable from its domain such that every constraint is satisfied. The CSP model of the WDP can be formally expressed as follows.

For $r$ component services, there are $m \times r$ 0-1 variables $X_{ij}$ ($i=1, ..., m; j=1, ..., r$) and $D(X_{ij})=\{0, 1\}$, $X_{ij}$ being 1 if the $j^{th}$ QoS offer in the $i^{th}$ QoS offer group is selected as part of the current winning QoS offers, 0 otherwise. The constraints for the CSP model are:

Coverage constraints: $\sum_{i=1}^{m} X_{ij} = 1 \quad \forall j \in [1, r]$ \hspace{1cm} (1)

Quality constraints: $q_p(S) < c_p \quad \forall p \in [1, l]$ \hspace{1cm} (2)

where $q_p(S)$ is the $p^{th}$ quality parameter of $S$ and can be calculated by applying the quality aggregation functions presented in Table I.

Constraints family (1) guarantees full coverage of $K$, i.e., each component service is included in exactly one QoS offer. Constraints family (2) ensures that all quality constraints for the SBS are fulfilled.

Solving the above CSP can generate a solution that fulfils all quality constraints for the SBS. Such a solution

<table>
<thead>
<tr>
<th>Quality Parameter</th>
<th>Aggregation Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$q_{cost}(S) = \sum_{s \in S} q_{price}(s_j)$</td>
</tr>
<tr>
<td>Response Time</td>
<td>$q_{rt}(c_{st}) = \max_{d_{ij}} (\sum_{s \in D_{ij}} q_{rt}(s_j))$</td>
</tr>
<tr>
<td>Availability</td>
<td>$q_{av}(c_{st}) = \prod_{s \in D_{ij}} q_{av}(s_j)$</td>
</tr>
<tr>
<td>Throughput</td>
<td>$q_{th}(c_{st}) = \min_{d_{ij}} (\min_{s \in D_{ij}} (q_{th}(s_j)))$</td>
</tr>
</tbody>
</table>

Figure 3. Example models for DMD generation.
is called a feasible solution. Very often, there are many feasible solutions. As an example, Figure 4 presents several feasible solutions for a SBS that yield different overall system utility at different overall costs. Now we seek to achieve the SBS provider’s optimisation goal. Given an objective function that represents the SBS provider’s optimisation goal, the CSP is turned into a constraint optimisation problem (COP). In a COP, each feasible solution generated by solving the CSP is associated with a ranking value for the objective function. The solution with the optimal ranking value is the solution to the COP, i.e., the optimal solution for the SBS.

Most existing approaches in the area of QoS-aware service selection aims at maximising the overall system quality (or utility) [1-4]. However, SBS providers’ optimisation goals can be various (and often conflicting to each other), e.g., to minimise the overall cost of service usage for the system or the overall response time of the system. INSC allows SBS developers to specify optimisation goals flexibly according to their needs, which in COP models are represented using different objective functions. In this research, we use the objective functions for two typical optimisation goals as examples: 1) to maximise the overall system utility; and 2) to minimise the overall cost of service usage.

1. Maximising the overall system utility. The solution to this optimisation goal is the set of QoS offers that fulfills all the quality constraints for S and meanwhile, maximises the overall utility of the SBS, e.g., $S_{MaxUtility}$ in Figure 4. The objective function that captures this optimisation goal is expressed as follows:

$$\text{objective}(S): \text{maximise} \left( \sum_{i=1}^{n} \sum_{j=1}^{m} u(s_{ij}) \times X_{ij} \right)$$  \hspace{1cm} (3)

where $s_{ij}$ is the service on which the $f^{th}$ QoS offer in the $i^{th}$ QoS offer group is placed on.

2. Minimising the overall cost of service usage. The solution to this optimisation goal is the set of QoS offers that fulfills all quality constraints for S and meanwhile, minimises the total price of the selected services, e.g., $S_{MinCost}$ in Figure 4. The objective function that captures this optimisation goal is expressed as follows:

$$\text{objective}(S): \text{minimise} \left( \sum_{i=1}^{n} \sum_{j=1}^{m} q_{\text{price}}(s_{ij}) \times X_{ij} \right)$$  \hspace{1cm} (4)

This COP can be solved by applying Integer Programming (IP) techniques [2] (or Mixed Integer Programming (MIP) technique [1, 4] if decimal variables are involved). If the solution to the COP can be found, based on the results from solving the COP, the winning service providers can be determined. According to the results, the SBS developer awards the service contracts to the winning service providers and finalises the SLAs for the concrete services. However, sometimes the solution to the COP cannot be found, which indicates that the current QoS offers are not good enough to fulfil all the quality constraints for the SBS. If the stop criterion for the negotiation is not met the negotiation will proceed to the next round, allowing the service providers to improve their QoS offers. In such cases, we remove constraint family (2) from the COP model and find the current best QoS offers, which achieve the SBS provider’s optimisation goal while fulfilling constraints families (1). The SBS developer then distributes the current best QoS offers to the service providers as the current winning QoS offers, based on which they can analyse their positions in the competition - they can compare their current QoS offers to the current winning QoS offers and determine whether and how to improve their QoS offers.

C. Counter-Offer Generation

In INSC, counter-offers are used to guide service providers on improving their QoS offers and filter out uncompetitive service providers. Service providers that cannot accept the counter-offers are removed from the negotiation. Utilising counter-offers, SBS developers can make sure that the negotiation proceeds in the right direction towards an optimal solution by demanding better QoS offers from the service providers until their QoS capacities are exhausted or the negotiation stops. In addition, by removing uncompetitive service providers, the complexity of solving the WDP can be reduced because the size of the search space for the WDP (i.e., the number of QoS offers) is reduced. As described in Section III.A, high minimum decrements can speed up the negotiation process at the early stage of a negotiation by efficiently filtering out uncompetitive service providers, while low minimum decrements can increase the success rate of finding an optimal solution by effectively exploiting the QoS capacities of the remaining service providers at the late stage. The generation of counter-offers in INSC is based on dynamic minimum decrement (DMD), aiming at guaranteeing both the effectiveness and the efficiency of the negotiation. As the negotiation proceeds, the QoS capacities of the service providers are gradually exhausted. Accordingly, the minimum decrements applied to the generation of counter-offers decreases over time as the negotiation proceeds.
V. EXPERIMENTS

This section presents the experimental evaluation of INSC, focusing on the comparison with existing IP based optimisation approaches in terms of effectiveness (measured by the success rate of finding an optimisation solution) and efficiency (measured by negotiation duration and the computation time taken per round for solving the WDP).

A. Prototype Implementation

We have implemented a prototype of INSC in Java using JDK 1.6.0 and Eclipse Java IDE. It implements the mechanisms introduced before. Given as input the functional specification of the business process of a SBS, a set of quality constraints for the SBS and an optimisation objective, an iterative negotiation can be held for the SBS. For solving the COP (i.e., the WDP) introduced in Section III.B, the prototype uses CPLEX v12.2, a commercial solver developed by IBM.

B. Experimental Setup

In the experiments for INSC, service providers compete for randomly selected services. Service providers’ QoS capacities and the QoS values specified in their first QoS offers were generated based on the QoS information provided in QWS - a publicly available Web service dataset that comprises measurements of nine QoS parameters of over 2500 real-world Web services. We randomly partitioned the services in QWS into categories that represent service providers’ historic QoS offers for corresponding component services. In each category, there were at least five services, i.e., five historic QoS offers for each service provider. The QoS offer specified in a service provider’s first QoS offer was generated based on its historic QoS offers. For example, if a service provider’s highest QoS offer for the price of a service is $2,000, the price proposed in its first QoS offer was selected from a randomly generated interval, e.g., $[1,500, $2,500]. This configuration captured the nature of service providers with various QoS capacities. The optimisation goal in the experiments is to maximise the overall system utility which is calculated using the utility evaluation functions introduced in [4]. To generate severe quality constraints for the SBS, we followed the following steps: 1) Group all the historic QoS offers of the service providers that compete for the same component services. For example, suppose there are ten service providers competition for component service \( s_x \), there are 5x10 historic QoS offers in total in the group for \( s_x \). 2) Rank the QoS values in the QoS offers in each group according to different QoS parameters. 3) Randomly selected a value from the top 50% best offers for each QoS parameter in each group. 4) Aggregate the selected values from each group to generate the quality constraints for the SBS.

When a solution to the COP could not be found, counter-offers were generated and the service providers could improve their QoS offers according to the counter-offers. In the experiments, service providers continued to improve their QoS offers until their QoS capacities were exhausted. For counter-offer generation, we adopted model #3 presented in Figure 3.

The experiments were conducted on a machine with AMD Athlon(tm) X4 640 3.00GHz CPU and 8 GB RAM, running Windows 7 x64 Ultimate.

C. Experimental Results

To evaluate the effectiveness of INSC, we implemented the optimisation approach adopted in [2-4] for comparison, which is also based on IP. We refer to this optimisation approach as Static Optimisation which seeks to select the best QoS offers based on their initial QoS values without considering the competition among candidate service providers. Non-negotiable QoS parameters do not influence the success rate obtained by INSC Optimisation and Static Optimisation. Hence, we assumed in the experiments that all the QoS parameters of the SBS were negotiable.

We first compare the effectiveness of INSC Optimisation and Static Optimisation by their success rates. There are three factors that might influence the success rate: 1) the number of quality constraints for the SBS; 2) the number of component services of the SBS; and 3) the number of service providers in the negotiation. Accordingly, in this series of experiments, we have conducted four sets of experiments. The configuration parameters for each set of experiment are presented in Table II. Each time the number of quality constraints, number of component services or number of service providers was changed, 100 instances of experiments were run and the results were averaged.

The results are presented in Figure 5. As illustrated, INSC Optimisation significantly outperforms Static Optimisation. Static Optimisation failed in finding an optimal solution in all scenarios while INSC Optimisation maintained relatively high success rates compared to Static Optimisation. Specifically, INSC Optimisation obtained average success rates of 24.04%, 10.70%, 78.94% and 37.35% in the first, second, third and fourth sets of experiments respectively. As presented in Figure 5a, the success rate of INSC optimisation decreases from 72% to 5% as the number of quality constraints increases from two to 10. The reason is that the increase in the number of quality constraints made it increasingly difficult to satisfy all severe quality constraints imposed on the SBS. In Figure 5b, the success rate decreases.

<table>
<thead>
<tr>
<th>Set of Experiment</th>
<th>Number of Quality Constraints</th>
<th>Number of Component Services</th>
<th>Number of Service Providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>2 - 10</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>#2</td>
<td>2</td>
<td>10 - 100</td>
<td>100</td>
</tr>
<tr>
<td>#3</td>
<td>2</td>
<td>10</td>
<td>100 - 1,000</td>
</tr>
<tr>
<td>#4</td>
<td>2 - 10</td>
<td>10 - 100</td>
<td>100 - 1,000</td>
</tr>
</tbody>
</table>
quickly as the number of component services increases. Compared to the number of quality constraints, the increase in the number of services influences the success rate more significantly. In particular, the success rate drops to zero when the number of component services exceeds 30. One reason is that the competition among service providers is negligible in those cases. For example, in the case of 40 component services, the average number of service providers competing for each service is 2.5. Exploiting the competition in such cases cannot increase the possibility of finding an optimal solution. Another reason is that when the number of candidate service providers for a component service was small, the contract for the component service was often left unallocated as those candidate service providers were easily filtered out by the DMD-based counter-offers. Any allocated service contracts would immediately lead to failure of the service composition. Thus, neither Static Optimisation nor INSC Optimisation could find solutions in those situations. The results presented in Figure 5c illustrate that the increase in the number of service providers does not negatively influence the success rate. Exploiting the competition among service providers increases the success rate remarkably compared. In the fourth set of experiments, we increased the overall scale of the scenarios. The intensity of the competition among the service providers remained on the same level - the average number of service providers competing for each service was always 10. However, as presented in Figure 5d, the increase in the number of quality constraints made it more difficult to find an optimal solution for the SBS. Thus, the average success rate obtained by INSC optimisation decreased from 90% to 2% as the experiment scaled up.

*Negotiation duration*, measured by the number of negotiation rounds, is a relevant efficiency concern of INSC because long negotiation duration implies low efficiency and high computational overhead - the WDP has to be solved many times in the entire negotiation process. In addition, increased negotiation duration may reduce SBS developers’ profits from the negotiation because usually a certain amount of administrative cost for maintaining the negotiation applies in each round. When the negotiation proceeds to later stages, the marginal benefit for SBS developers from the competition among service providers may decrease dramatically which, in the worst-case scenario, is not even worth the additional administrative cost. However, a long negotiation has potentials to achieve the SBS provider’s optimisation goal because it gives SBS developers the necessary time to exploit high-value service providers’ QoS capacities fully. Apparently, there is a tradeoff between the effectiveness and the efficiency of the INSC negotiation. Table III presents the average negotiation duration in different sets of experiments. On average, it took approximately four rounds for an INSC negotiation to find an optimal solution in the experiments.

Besides the number of negotiation rounds, the time consumption for each round in solving the WDP is also a very important issue regarding efficiency because the WDP is a NP-complete problem. Figure 6 presents the average time consumption per round for solving WDPs (when an optimal solution was found) in different sets of experiments. As illustrated, the time consumption per round for solving WDP is not very large. In the largest-
scale scenario, solving the WDP took approximately two seconds per round on average.

The experimental results presented above demonstrate that INSC significantly increases the success rate of finding optimal solutions for SBSs by exploiting the competition among service providers. Those benefits come at a reasonable price - INSC may require several rounds to complete. However, the time consumption for solving the WDP in each round is satisfactory, especially given the significant increase in success rate. Thus, we believe the efficiency of INSC is satisfactory in most real-world applications.

VI. CONCLUSIONS AND FUTURE WORK

Quality of Service (QoS) aware service selection for service-based systems (SBSs) is a critical issue in service-oriented environments. This paper has proposed Iterative Negotiation for Service Compositions (INSC), a novel joint optimisation and negotiation approach that supports effective and efficient QoS-aware service selection for SBSs based on iterative negotiation. Using INSC, SBS developers can properly exploit the competition among service providers. When an optimal solution cannot be found, INSC iterates, giving service providers chances to update their QoS offers. Counter-offers and current winning offers are sent to service providers as guidance on how to improve their QoS offers, giving them the incentives to compete. The experimental evaluation shows that, by exploiting the competition among service providers, INSC significantly improves the success rate of finding optimal solutions in scenarios where severe quality constraints are imposed on SBSs. INSC also demonstrates satisfactory negotiation duration and high efficiency in solving WDPs.

As discussed in Section IV.C, when the number of service providers competing for a service is small, the contract for the service is often left unallocated, resulting in failure of the service composition. In those scenarios, the service providers’ QoS capacities are easily exhausted. In future work, we will investigate this issue and improve the success rate in such situations.

ACKNOWLEDGMENT

This work is partly funded by the Australian Research Council Discovery Project Scheme under grant No. DP0663841, National Science Foundation of China under grant No. 90412010 and China Grid project from Ministry of Education of China.

REFERENCES


