IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2019, accepted June 21, 2019, date of publication July 1, 2019, date of current version July 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926127

Service Selection With QoS Correlations in Distributed Service-Based Systems

DONGWEI LI[1], DAYONG YE[2], NAN GAO[3], AND SHULIANG WANG[1]
[1]School of Computer Science, Beijing Institute of Technology, Beijing 100081, China
[2]School of Software, Sydney University of Technology, Sydney, NSW 2006, Australia
[3]School of Science, RMIT University, Melbourne, VIC 3001, Australia

Corresponding author: Shuliang Wang (slwang2011@bit.edu.cn)

This work was supported by the China Scholarship Council (CSC). Authors thank the reviewers for their valuable comments.

ABSTRACT Service selection is an important research problem in distributed service-based systems, which aims to select proper services to meet user requirements. A number of service selection approaches have been proposed in recent years. Most of them, however, overlook quality-of-service (QoS) correlations, which broadly exist in distributed service-based systems. The concept of QoS correlations involves two aspects: 1) QoS correlations among services and 2) QoS correlations of user requirements. The first aspect means that some QoS attributes of service not only depend on the service itself but also have correlations with other services, e.g., buying service 1 and then getting service 2 with half price. The second aspect means the relationships among QoS attributes of user requirements, e.g., a user can accept a service with fast response time and high service cost or the user can also accept a service with slow response time and low service cost (Fig. 1). These correlations significantly affect user selection of services. Currently, only a few existing approaches have considered QoS correlations among services, i.e., the first aspect, but they still overlook QoS correlations of user requirements, i.e., the second aspect, which are also very important in distributed service-based systems. In this paper, a novel service selection approach is proposed, which not only considers QoS correlations of services but also accounts for QoS correlations of user requirements. This approach, to the best of our knowledge, is the first one which considers QoS correlations of user requirements. Also, this approach is decentralized which can avoid the single point of failure. The experimental results demonstrate the effectiveness of the proposed approach.

INDEX TERMS Distributed service-based systems, service selection, QoS correlations, user requirements.

I. INTRODUCTION

Nowadays, there are thousands of Web services available in large distributed service-based systems (SBSs) [1]–[3]. Some of the services may provide equivalent functions but with different quality-of-service (QoS) values, e.g., cost, response time, throughput, reliability and so on. Thus, selecting proper services to meet user requirements becomes an important research problem in distributed SBSs [4]. User requirements include two perspectives: functional and non-functional requirements [5]. Service discovery aims to discover Web services which can meet user basic functional requirements, whereas service selection aims to select services to satisfy user QoS requirements. In the cases where users require a composite and complex service, a workflow of Web services needs to be formed through service composition which aims to satisfy end-to-end global QoS constraints [6]–[9].

Currently, many QoS-aware service selection and composition approaches have been proposed [5], [10]–[12]. These works assume that the QoS values of services are predetermined and fixed. Then, for each task, they use greedy-like algorithms to select one service from each set of candidate services to optimize the system quality. These works, however, overlook the QoS correlations which indicate relationships among QoS attributes of both providers' services and user requirements. QoS correlations are very important, as they widely exist in distributed SBSs. For example, on the one hand, a provider has two services, s_1 and s_2. Then, to attract users, the provider may set that "buy s_1, get s_2 with half price". Moreover, if both of the two services, s_1 and s_2, are employed, the response time may be reduced, because the

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

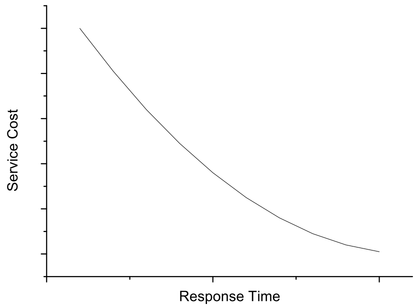


FIGURE 1. User QoS requirement correlation.

two services are executed on one server and the time for transmitting data between them can be saved, given that the server is not overloaded. Such pricing strategy and response time reduction can significantly affect user selection of services. On the other hand, different users usually have their own tastes and preferences due to various factors such as budget, socio-economic status and even personality, which will then affect their daily behaviors [13] (e.g., decision making for choosing favorite services with different qualities). For example, a user may accept a service with longer response time but a lower price, or the user may also accept a service with a higher price but a shorter response time (Fig. 1). In addition, a user may accept a service with faster processing speed but lower accuracy, or the user may also accept a service with slower processing speed but higher accuracy. Such QoS correlations of user requirements are very important as they broadly exist among users and demonstrate user flexibility. If such QoS correlations are overlooked, some users may lose the opportunity to get the required services.

QoS correlations of providers' services have been considered in [14]. However, QoS correlations of user requirements are still overlooked by the existing research. In this paper, we propose a novel service selection approach which, to the best of our knowledge, is the first one that takes QoS correlations of user requirements into consideration. The proposed approach is based on multi-agent negotiation, where a user is modeled as a buyer agent and a provider is modeled as a seller agent. The buyer agent negotiates with the seller agent over the services required by the user. Here, an intelligent agent is an entity which can make rational decisions autonomously in a dynamic environment. An intelligent agent blends pro-activeness and re-activeness, shows rational commitments to decision making and exhibits flexibility when facing an uncertain and changing environment. There are two advantages of the proposed approach.

1) The proposed approach does not only take into account providers' service QoS correlations but also user requirement QoS correlations, so it suits real environments better than most of the existing approaches which overlook user requirement QoS correlations.
2) The proposed approach is based on individual negotiation and thus does not need a central controller. Hence, the approach is decentralized in nature and thus, can avoid the single point of failure which exists in centralized approaches [6], [14]–[16].

Negotiation has been broadly used in service selection and composition for building distributed SBSs [17]–[21]. Most of these works focus on the study of service level agreement, i.e., studying the features and performance of different negotiation strategies while overlooking QoS correlations. In contrast, this paper does not study specific negotiation strategies but employs negotiation as a tool to realize the proposed approach. Also, this paper takes QoS correlations of both providers' services and user requirements into consideration.

The rest of the paper is organized as follows. Section II reviews current related studies. Section III provides a motivating example. Then, Section IV presents the details of the proposed approach which is evaluated in Section V. Finally, Section VI concludes the paper and points out future research.

II. RELATED WORK

Currently, a large number of approaches has been proposed for service selection and composition. Typically, QoS-aware service composition is modeled as a combinatorial optimization problem which can be solved by integer programming [6], [22], i.e., given a set of QoS attributes and a set of constraints on the values of the QoS attributes, then maximizing (or minimizing) an objective function. In this section, a number of existing approaches which are closely related to ours are reviewed.

Zeng *et al.* [6] proposed a middleware platform for Web service selection for composition. In their platform, two selection approaches are used, where one is based on the local selection of services and the other is based on global allocation of tasks to services using integer programming. By using the two approaches, user satisfaction can be maximized, which is expressed as utility functions over QoS attributes under user-given constraints.

Ardagna and Pernici [15] formalized the service composition problem as a mixed integer linear programming problem. They used loops peeling in the optimization and took constraints posed by stateful services into consideration. In their approach, there is a centralized *Concretisator* module which selects the best concrete services to be invoked from a service registry for each task of the composed service according to user's requirements and the optimization objective.

Trummer *et al.* [16] introduced a new category of PQDSS (Pareto Quality-Driven Service Selection) algorithms for multi-objective quality-driven service selection. Their algorithms combine polynomial runtime in a number of services and workflow tasks with precision guarantees according to the additive ϵ-metric. By tuning a precision parameter, their algorithms can adapt to various scenarios, e.g., precision must be guaranteed or high efficiency must be met.

Moustafa *et al.* [23] proposed a stigmergy-based approach to model service interactions and handle service composition. In their approach, interactions among service agents are indirect by leaving and sensing artificial pheromone. Such pheromone encodes specific information which is used to achieve service composition. Also, they developed a trust model for service selection based on the balance between the trust ranks of the concrete services and the trust ranks of the whole workflow.

Moustafa and Zhang [24] proposed a reinforcement learning approach for multi-objective service composition and adaptation in dynamic uncertain environments. They used the multi-objective Markov decision process to model the service composition problem. Then, two algorithms were devised to handle the single policy and multiple policy multi-objective service composition based on user preferences. The solution of each of the two algorithms is a procedure that indicates how an agent selects a service in each state. By using the two algorithms, an agent can find a set of optimal workflows.

Deng *et al.* [25] proposed a mobility-aware service selection approach. Their approach takes the mobility of users into account, where the mobile network's signal strength may vary with the movement of users. The approach is based on a swarm intelligence optimization algorithm: teaching-learning-based optimization (TLBO). TLBO consists of two phases: teacher phase and student phase. The teacher phase means learning from the teacher, while the student phase means learning through interactions between learners. Each learner means a feasible service composition and the teacher is the best solution obtained so far.

The above studies do not consider QoS correlations on either the provider side or user side. Moreover, the approaches developed in [6], [15], [16] are centralized which has the potential of the single point of failure. QoS correlations and dependencies have been addressed in some studies.

Basu *et al.* [26] presented an approach to analyze service execution data to discover dependencies among services. Their approach, which consists of four steps, is based on the messages exchanged between services. The approach first infers causal dependencies within messages. Then, based on the causal dependencies, the approach creates and prunes a dependency graph. Finally, based on the dependency graph, the approach identifies frequent paths which represent traces of dependent message exchanges among the managed services. Similarly, in [27], Romano *et al.* also presented an approach to extract dependencies among services. Their approach is based on the concept of vector clocks which generate a partial ordering of events. The service dependencies then can be inferred from the ordering of events.

Kang *et al.* [28] proposed a PCA based web service selection framework which takes the user's preference priority into account. To analyze the correlations among QoS attributes, they computed the web service candidates based on the overall QoS and recommended services with top QoS values to users.

Kang *et al.* [29] proposed an optimal composition method for web service selection. They transformed the original Web service selection problem into a multi-objective services composition optimization problem with global QoS constraints. Then, the intelligent optimization of PSO is used to produce the optimal services composition process with QoS constraints swarm optimization.

Tao *et al.* [30] proposed a QoS description mode to support the presentation of resource service correlations. Then, they developed a particle swarm optimization method to address the multi-objective manufacturing grid service composition problem by taking into account resource service correlations.

Barakat *et al.* [31] presented a correlation-aware service composition approach, where quality dependencies among services are considered. Their approach incorporates pruning techniques to reduce search space by eliminating uninteresting service compositions.

In [32], Jin *et al.* first proposed a correlation-aware manufacturing cloud service description model which characterizes the QoS dependencies of service on other related services. Then, they developed a service correlation mapping model to acquire the correlation of QoS values among services. Based on the two models, they used a genetic algorithm to efficiently select services with the consideration of QoS correlations.

Deng *et al.* [14] developed a QoS-aware service composition approach which takes QoS correlations among services into consideration. Their approach incorporates a correlation-aware pruning method which prunes redundant services and reserves the services with QoS correlations. Then, they considered two cases: QoS correlations in adjacent tasks and QoS correlations in non-adjacent tasks. Against each case, they developed a service selection algorithm which can generate optimal solutions.

These works [14], [30]–[32] have considered the QoS correlations of providers' services. They, however, still overlook the important user requirement QoS correlations which are taken into account in this paper. Also, most of these works are centralized in nature which have the potential of the single point of failure. Some other studies consider user requirement QoS correlations, but they focus on different research problems from ours. For example, Ye *et al.* [33] developed a multivariate QoS prediction method based on end user long-term QoS requirements and their correlations, while our work focuses on service selection.

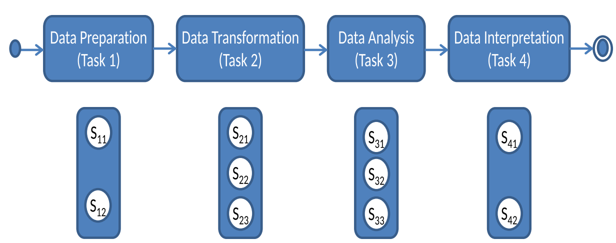


FIGURE 2. **A motivating example.**

III. A MOTIVATING EXAMPLE

This section provides an example of service selection for data processing. In Fig. 2, a user has terabytes of data to be processed. For example, a company has terabytes of data about its customers to be processed in order to reveal meaningful information for the company's marketing strategy design. As the amount of data is extremely large, it is infeasible for the user to process the data using in-house processing tools. Thus, the user has to use services provided by other providers through a platform, e.g., distributed SBSs. The data processing consists of four tasks: data preparation, data transformation, data analysis and data interpretation.

Usually, for each task, there could be multiple available services with different QoS values, e.g., cost and response time. Thus, the user has to select a set of services which can meet the user's requirements. However, the user's requirements may be flexible (Fig. 1), e.g., the user can accept a service with longer response time but less cost, or the user can also accept another service with a higher cost but shorter response time. For example, there are two services for Task 1: service s_{11} and service s_{12}. The cost and response time of service s_{11} are 10 and 100*ms*, respectively. The cost and response time of service s_{12} are 5 and 200*ms*, respectively. The idea of most existing approaches is that the user sets a maximum acceptable response time and then selects a service with the lowest cost, or vice versa, the user sets a maximum acceptable cost and then selects a service with the shortest response time. Thus, if the user's maximum acceptable response time is 200*ms*, service s_{12} should be selected as its cost is lower than service s_{11}. However, in reality, some user requirements may be flexible: the user's maximum acceptable response time is 200*ms*, but the user also accepts a service with shorter response time even though the service may have a higher cost. Thus, in this situation, service s_{11} may also be acceptable to the user. Such QoS correlations of user requirements, which have not been considered in existing literature, do exist in the real world and have to be taken into account in service selection. If such QoS correlations are overlooked, some users may not get the required services. For example, suppose there is another user in the system whose maximum acceptable cost is 5. Then, only service s_{12} is acceptable to this user. However, as described above, if QoS correlations are overlooked, service s_{12} has been taken by the former user. Therefore, the latter user cannot get a satisfactory service. Nonetheless, if QoS correlations are considered, the former user may take service s_{11} and then the latter user can get service s_{12}. Thus, both of them get a satisfactory service.

IV. PROBLEM DESCRIPTION AND APPROACH DESIGN

In this section, we first formalize the problem and then introduce the details of our approach. Afterwards, the theoretical analysis of our approach is given.

A. PROBLEM DESCRIPTION

The service composition problem with QoS correlations is modeled as a multi-agent negotiation problem. At time $t = 0$, there are n users and m providers, where each user is represented as a buyer agent and each provider is represented as a seller agent. Seller agents sell services on behalf of providers and buyer agents buy the required services on behalf of users. Then, seller agents and buyer agents reach agreements on the prices of required services through negotiation. In the following contents, we use the terms, user and buyer, interchangeably and use the terms, provider and seller, interchangeably.

Formally, the negotiation between a buyer agent and a seller agent is a discrete time bilateral alternating-offer bargaining. A buyer and a seller can act at time $t \in \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$, and they make offers in an alternating manner. The possible actions of an agent at time $t > 0$ are: 1) making an *offer*, where an agent proposes a price for the required service; 2) *accept*, which means that the buyer agent and the seller agent reach an agreement; 3) *confirm*, which means that an agent confirms the agreement; 4) *cancel*, which means that an agent cancels an agreement; 5) *exit*, which means the negotiation fails. The action, *exit*, is used in the situation that the deadline of either a buyer agent or a seller agent has expired and no agreement is reached.

Each seller agent, s, has a utility function U_s, which represents the gain over the bargaining outcomes. U_s depends on the seller's reserve price $RP_s > 0$, temporal discount factor $\zeta_s^t \in (0, 1]$ which represents the bargaining cost, and negotiation deadline $T_s \in \mathbb{N}$, $T_s > 0$. Formally, we have

$$U_s = (pr - RP_s) \cdot \zeta_s^t, \tag{1}$$

where pr is the proposed price and $\zeta_s^t = k_s \cdot \zeta_s^{t-1}$. The coefficient, $k_s \in (0, 1)$, is sellers' temporal discount rate.

For each buyer agent, b, because a service has multiple attributes, a buyer may have a preference for some of the attributes. Thus, for each attribute i, the buyer has a specific utility function U_b^i with a specific reservation value R_b^i and a specific discount factor $\zeta_b^{t(i)} \in (0, 1]$. However, because a service, as a whole, cannot be divided, there is only one deadline $T_b \in \mathbb{N}$, $T_b > 0$ for all the attributes. Formally, we have

$$U_b^i = \begin{cases} (x_b^i - R_b^i) \cdot \zeta_b^{t(i)}, & \text{if } i \text{ is a positive attribute,} \\ (R_b^i - x_b^i) \cdot \zeta_b^{t(i)}, & \text{if } i \text{ is a negative attribute,} \end{cases} \tag{2}$$

where x_b^i is the agreed value of attribute i and $\zeta_b^{t(i)} = k_b \cdot \zeta_b^{(t-1)(i)}$. The coefficient, $k_b \in (0, 1)$, is buyers' temporal discount rate. A positive attribute means that the larger the attribute's value, the more satisfied the buyer is, e.g., bandwidth and reliability. A negative attribute means that the smaller the attribute's value, the more satisfied the buyer is, e.g., response time and price. Then, the overall utility of the buyer is the sum of single-attribute utility functions:

$$U_b = \sum_{1 \le i \le k} U_b^i, \tag{3}$$

where k is the number of attributes. This kind of multi-attribute utility functions is called additive multi-attribute utility functions and is broadly used in multi-agent negotiation research [34]–[36]. Without loss of generality, if there is no agreement between a buyer agent and a seller agent when the negotiation finishes, the utilities of both the buyer and the seller are 0, $U_s(NoAgreement) = U_b(NoAgreement) = 0$.

Moreover, if $t \geq T_s$, $U_s = -1$; and if $t \geq T_b$, $U_b = -1$. Such settings ensure the rationality of agents with the presence of deadlines: once the deadline of an agent has expired, the agent prefers to take action *exit* and gets 0 utility with no agreement, rather than making any agreement beyond the deadline with -1 utility. The details of the proposed approach will be presented in the following sub-section.

B. APPROACH DESIGN

In this research, we proposed a novel service selection approach, which consists of two parts: a candidate service pruning algorithm and a service negotiation protocol. The candidate service pruning algorithm is used to remove the services which are not good enough. The service negotiation protocol is then used for bargaining between buyer agents and seller agents.

1) CANDIDATE SERVICE PRUNING

After buyer b sends service requests to sellers, b may receive a number of responses from sellers which can provide b with the required services. For example, in Fig. 2, for each task, there is a set of services which can meet the requirements of the task.

In these responses, there may be many redundant services which have the same functionality but have different qualities and prices. In order to reduce the complexity of service selection, **Algorithm 1** is used for buyers to remove some redundant services from candidate service sets.

Algorithm 1: Candidate Service Pruning

1 * Let b be a buyer and CS_b be the candidate service set of b * \
2 buyer b sends service requests to potential sellers;
3 Sellers respond b by listing the available services;
4 buyer b classifies the services for each task, $task_i$, and saves the services to each candidate service set, $CS(task_i)$, respectively;
5 **for** *each candidate service set* $CS(task_i)$ **do**
6 $\quad CS'(task_i) \leftarrow \emptyset$;
7 $\quad$ **for** *each service* $se \in CS(task_i)$ **do**
8 $\quad\quad$ **if** *se is free of correlations* **then**
9 $\quad\quad\quad CS(task_i) \leftarrow CS(task_i) - \{se\}$;
10 $\quad\quad\quad CS'(task_i) \leftarrow CS'(task_i) \cup \{se\}$;
11 $\quad$ **for** *each service* $se' \in CS'(task_i)$ **do**
12 $\quad\quad$ **for** *each service* $se \in CS(task_i)$ **do**
13 $\quad\quad\quad$ **if** $U_b(se') < U_b(se)$ **then**
14 $\quad\quad\quad\quad CS'(task_i) \leftarrow CS'(task_i) - \{se'\}$;
15 $\quad\quad\quad\quad$ **break**;
16 $\quad CS(task_i) \leftarrow CS(task_i) \cup CS'(task_i)$;
17 $CS_b \leftarrow \bigcup CS(task_i)$;

In Lines 2-4, buyer b sends service requests to sellers and receives responses from sellers. The service request indicates which services are required by the buyer. For example, in Fig. 2, the user requires four types of services: data preparation, data transformation, data analysis and data interpretation. Then, seller s responds b which services can be provided. Seller s also presents b the attribute values of these services, including the prices and the QoS correlations of these services. For example, in Fig. 2, suppose that seller s is a provider and it has services s_{11} and s_{21}. Seller s responds buyer b that it has services s_{11} and s_{21}. Moreover, s also shows b the attribute values of the two services and the QoS correlations of the two services, e.g., if b buys the two services together, b can get the second service s_{21} with half price. Buyer b then classifies the services for each task, $task_i$, and saves the services to each candidate service set, $CS(task_i)$, respectively.

In Lines 5-10, for each candidate service set $CS(task_i)$, buyer b extracts the services, which do not have correlations with other services, and saves these correlation-free services into a set $CS'(task_i)$.

In Lines 11-15, buyer b calculates the utility of each service. If the utility of a correlation-free service is less than the utility of a non-correlation-free service, then the correlation-free service is removed from set $CS'(task_i)$. This pruning mechanism encourages sellers to propose prices to buyers as real as possible, because otherwise sellers' services may be directly removed from buyers' candidate service sets without further negotiation. Also, this removal mechanism keeps all the services which have correlations with other services. This is because the services, which have QoS correlations among them, are usually provided by one seller. Thus, negotiation with one seller for a set of services can save buyers time and communication overhead compared to negotiation with a number of different sellers for the same set of services (referring to Property 3 in Sub-section IV-C).

In Lines 16 and 17, buyer b unites the candidate service sets of all the tasks together and forms a final candidate service set CS_b which will be used in service negotiation.

2) SERVICE NEGOTIATION

After candidate service pruning, buyer b starts negotiating with sellers of the services in candidate service set CS_b. Our service negotiation protocol extends Rubinstein's alternating-offers protocol [37], where an agent can make multiple agreements for service with other agents and cancel agreements without paying penalty before confirmation is made. Rubinstein's alternating-offers protocol is powerful, which captures the essence of negotiation. It is easy to implement and has been widely used for bilateral bargaining [38], [39]. The proposed negotiation protocol is shown in **Algorithm 2**.

In Lines 2 and 3, for each service *se* in set CS_b, buyer b starts negotiation with the seller of each service. Here, if service *se* has QoS correlations with other services, buyer b will negotiate with the seller over the bundle of the services. For example, in Fig. 2, seller s can provide services s_{11} and s_{21}, and s_{21} can be half price if buyer b uses s_{11}. In this

Algorithm 2: Service Negotiation

```
1  \* Let s be a seller, b be a buyer and CS_b be the
   candidate service set of b * \
2  for each service se ∈ CS_b do
3      Let s be the seller of service se;
4      while t < T_s and t < T_b do
5          if b accepts service se then
6              TA(b) ← TA(b) ∪ {se};
7              TA(s) ← TA(s) ∪ {se};
8              break;
9          else
10             b generates an offer to s;
11             if s accepts the offer then
12                 TA(b) ← TA(b) ∪ {se};
13                 TA(s) ← TA(s) ∪ {se};
14                 break;
15             else
16                 s generates a counter-offer to b;
```

situation, buyer b negotiates with seller s over the bundle of services s_{11} and s_{21}.

In Lines 4-10: For the pair of a seller, say s, and buyer b, before their deadlines, they negotiate the price of a service or a bundle of services. Buyer b and seller s alternately propose offers until an agreement is reached. In Lines 5-8, b evaluates the service (or the bundle of services) and if b is happy with it, b accepts the service (or the bundle of services) and forms a temporary agreement with s. The negotiation then finishes. Otherwise, b will generate an offer to s (Line 10). We use an example to illustrate this process. Suppose seller s can provide service s_{11}. Then, b calculates the utility $U_b(s_{11})$ of using s_{11} by employing Equation 3. On the one hand, if $U_b(s_{11}) \geq RU_b(task_1)$, where $RU_b(task_1)$ is b's reserved utility of task 1, then s_{11} is acceptable and a temporary agreement on s_{11} is formed between buyer b and seller s (Lines 6 and 7). As described in Section I, user requirements are flexible and correlated. Thus, we use reserved utility for calculation rather than reserved price, as users can accept a service with a higher price and a better quality or a lower price with poorer quality. On the other hand, if $U_b(s_{11}) < RU_b(task_1)$, b will generate an offer to s (Line 10). The offer generation is based on: 1) the price offered by seller s, Pr_s, 2) the utility that buyer b can get based on the current offer, U_b, 3) b's reserved utility RU_b, 4) the temporary agreements $\mathcal{TA}(b)$ that b has made with other sellers for the same type of service, i.e., services with the same function but different qualities and prices, 5) the remaining time to the deadline T_b, and 6) the market competition. Here, the market competition is modeled using agents' bargaining position BP which can be calculated using Equation 4 [40].

$$BP_b(t) = avg_i\left(\frac{\delta_i(t)}{\frac{\Delta_i(t)}{t}}\right) \tag{4}$$

In Equation 4, $BP_b(t)$ is buyer b's bargaining position at time t, which is obtained by averaging the ratio of $\delta_i(t)$ and $\frac{\Delta_i(t)}{t}$. Here, $\delta_i(t)$ is the amount of reduced price made by seller i at time t, and $\frac{\Delta_i(t)}{t}$ is the average amount of reduced price made by seller i in the previous t rounds. Buyer b concurrently negotiates with multiple sellers and i represents one of these sellers. If $BP_b(t) \ll 1$, it is implied that many sellers make small concessions, i.e., reduce a small amount of prices, at time t, thus buyer b is very likely to be in a competition intensive market. Similarly, if $BP_b(t) \gg 1$, buyer b is very likely to be in a competitive free market. Then, the offer, i.e., the proposed price Pr_b, can be derived using Equation 5.

$$Pr_b = \begin{cases} Pr_s \cdot \dfrac{U_b}{RU_b} \cdot \dfrac{t}{T_b} \cdot \dfrac{1}{1+BP_b(t)}, & \text{if } \mathcal{TA}(b) = \emptyset \\ Pr_s \cdot \dfrac{|\mathcal{TA}(b)|}{1+|\mathcal{TA}(b)|} \cdot \dfrac{U_b}{RU_b} \cdot \dfrac{t}{T_b} \cdot \dfrac{1}{1+BP_b(t)}, & \text{otherwise} \end{cases} \tag{5}$$

where $|\mathcal{TA}(b)|$ means the number of elements in set $\mathcal{TA}(b)$. When seller s receives buyer b's offer, s calculates utility of the offer $U_s(Pr_b)$ using Equation 1. If $U_s(Pr_b) \geq RU_s(se)$, where $RU_s(se)$ is the reserved utility of seller s on service se, this offer is acceptable (see Lines 11-16).

Then, a temporary agreement is formed and the negotiation terminates (Lines 11-14). A seller's utility is based not only on its reserved price and the proposed price, but also on the current time. Therefore, we use reserved utility for calculation instead of reserved price. If $U_s(Pr_b) < RU_s(se)$, s will generate a counter-offer to b (Line 16) and the negotiation continues to the next round. Similar to the buyer, the proposed price Pr_s from the seller, can be obtained using Equation 6.

$$Pr_s = \begin{cases} Pr_b \cdot \dfrac{U_s}{RU_s} \cdot \dfrac{T_s}{t} \cdot (1+BP_s(t)), & \text{if } \mathcal{TA}(s) = \emptyset \\ Pr_b \cdot \dfrac{1+|\mathcal{TA}(s)|}{|\mathcal{TA}(s)|} \cdot \dfrac{U_s}{RU_s} \cdot \dfrac{T_s}{t} \cdot (1+BP_s(t)), & \text{otherwise} \end{cases} \tag{6}$$

where $|\mathcal{TA}(s)|$ means the number of elements in set $\mathcal{TA}(s)$. The offer generation of seller Pr_s is based on: 1) the price offered by buyer b, Pr_b, 2) the utility that seller s can get based on the current offer, U_s, 3) s's reserved utility RU_s, 4) the temporary agreements $\mathcal{TA}(s)$ that s has made with other buyers for its service se, 5) the remaining time to the deadline T_b, and 6) the market competition.

At the end of a negotiation, as buyer b may negotiate with multiple sellers simultaneously for a single task, buyer b may have achieved a set of temporary agreements for the task. In these temporary agreements, buyer b selects the temporary agreement, which can bring b the highest utility, and confirms with the seller. The temporary agreement now becomes a final agreement. Buyer b then cancels other temporary agreements. Similarly, seller s may also have a set of temporary agreements for a single service. In these temporary agreements, seller s selects the temporary agreement which has the highest price, and confirms with the corresponding buyer.

Seller s then cancels other temporary agreements. Only when the selection of temporary agreement for buyer b and seller s are the same, the service selection can be considered successful. If buyer b confirms the temporary agreement while seller s cancels this agreement, the service selection for buyer b will be failed, and vice versa.

C. ANALYSIS OF THE APPROACH

As described in Sub-section IV-B, the proposed approach consists of two parts: a candidate service pruning algorithm and a service negotiation protocol. The two parts are analyzed as follows.

1) CANDIDATE SERVICE PRUNING

The candidate service pruning algorithm (**Algorithm 1**) is used to prune the services which are not worth taking further consideration.

Property 1: Suppose that a buyer has m tasks. For an individual task, $task_i$, suppose there are n_i services in total and n'_i correlation-free services, $n_i \geq n'_i$. Thus, the computation complexity to prune services for a buyer is $O(mn'^*k^*)$, where $n'^* = MAX\{n'_1, \ldots, n'_m\}$ and $k^* = MAX\{(n_1 - n'_1), \ldots, (n_m - n'_m)\}$.

Analysis: **Algorithm 1** has two steps. The first step is to distinguish correlation-free and non-correlation-free services, whose complexity is $O(\sum_{1 \leq i \leq m} n_i)$. The second step is to remove those correlation-free services which have less utilities than non-correlation-free services, whose complexity is $O(\sum_{1 \leq i \leq m}(n'_i(n_i - n'_i)))$. Therefore, the overall complexity is $O(\sum_{1 \leq i \leq m}(n_i + n'_i(n_i - n'_i)))$. As $\sum_{1 \leq i \leq m}(n'_i(n_i - n'_i))$ has included $\sum_{1 \leq i \leq m} n_i$, $O(\sum_{1 \leq i \leq m}(n_i + n'_i(n_i - n'_i))) = O(\sum_{1 \leq i \leq m}(n'_i(n_i - n'_i)))$. Then, by setting $n'^* = MAX\{n'_1, \ldots, n'_m\}$ and $k^* = MAX\{(n_1 - n'_1), \ldots, (n_m - n'_m)\}$, the complexity becomes $O(mn'^*k^*)$. Moreover, there are two extreme cases: 1) all the services are correlation-free, i.e., for each i, where $1 \leq i \leq m$, $n_i = n'_i$, and 2) all the services have correlations with other services, i.e., for each i, where $1 \leq i \leq m$, $n'_i = 0$. In the two extreme cases, all the candidate services have to be taken into account during the service negotiation phase.

2) SERVICE NEGOTIATION

The service negotiation between a buyer agent and a seller agent is carried out in an alternating manner through both parties giving concessions (**Algorithm 2**).

Property 2: The proposed negotiation protocol is not monotonic.

Analysis: During the negotiation process, the offers of a buyer agent and a seller agent are generated using Equations 5 and 6, respectively. In Equation 5, the price offered by the buyer in round t, i.e., $Pr_b(t)$, is based on the seller offered price in round $t-1$, i.e., $Pr_s(t-1)$, multiplying a set of items. Each of these items is less than or equal to 1. Thus, $Pr_b(t) \leq Pr_s(t-1)$. Here, $Pr_s(t-1)$ is created using Equation 6 based on $Pr_b(t-2)$ and a set of items which are larger than or equal to 1. Thus, $Pr_s(t-1) \geq Pr_b(t-2)$. It can be seen that both $Pr_b(t)$ and $Pr_b(t-2)$ are less than or equal to $Pr_s(t-1)$. $Pr_b(t)$ and $Pr_b(t-2)$ are determined dynamically. Hence, it is unable to compare their values. Generally, during negotiation, agents should monotonically increase/decrease or insist on the prices of their previous offers until an agreement is reached or the deadline of any party is due [38]. However, in the proposed negotiation protocol, dynamic situations are taken into consideration. The price of an offer generated by an agent to a trading partner is based on the negotiation outcomes of this agent with other trading partners. Typically, when there are a few competitors and a number of trading partners, concessions can be small, but when there are a number of competitors and a few trading partners, concessions may have to be large so as to secure at least one agreement. In a dynamic environment, the competition changes dynamically. Thus, the proposed protocol is not monotonic. For example, there are one buyer and one seller who are bargaining on a service. The seller asks price 100 while the buyer wants 80. After several negotiation rounds, they reach at 90, where both parties offer price 90. If the environment is static, the two parties may make a deal on 90. However, if the environment is dynamic, a new seller may join in and offer price 85 for the same service. Then, the buyer may request the former seller to match up price 85, as otherwise, the buyer will reach an agreement with the latter seller.

Property 3: Let $T = Min(T_b, T_s)$, where T_b is the deadline of a buyer and T_s is the deadline of a seller. The communication complexity of the negotiation protocol is $O(T)$.

Analysis: In each negotiation round, in the worst case, the buyer rejects the seller's offer and generates a new offer to the seller. However, the seller also rejects the buyer's offer and sends a counter-offer to the buyer. This process continues until one of their deadline is reached, i.e., $MIN\{T_b, T_s\}$. Thus, in the worst case, the number of communication messages created during the negotiation process is $2T$. Then, the complexity is $O(T)$. Moreover, if a buyer is negotiating with multiple sellers simultaneously, e.g., n sellers, the complexity becomes $O(nT)$. Hence, negotiation should be taken against as fewer partners as possible. This analysis explains the statement in Sub-section IV-B, where buyers should select the services, which have QoS correlations, in priority so as to save communication overhead.

V. EXPERIMENT AND ANALYSIS

In this section, the proposed service selection approach, named as *Nego.-based*, is evaluated in comparison with three other approaches which are described as follows.

1) The correlation-aware service pruning (*CASP*) method [14]. This approach first prunes some redundant services and then selects optimal services in a centralized manner. This approach is centralized which has the potential of the single point of failure. Moreover, this approach considers QoS correlations of provider services without QoS correlations of user requirements.
2) A random approach, named as *Random*. This approach, created by us, randomly selects a service for each task.

3) A simplified version of our approach, named as *Nego.-based-simplified*. This approach is similar to the proposed approach but is simplified. The simplified approach, which is also decentralized, considers QoS correlations of providers' services but does not consider QoS correlations of user requirements. In this approach, each user has a reserved price instead of a reserved utility, because QoS correlations of user requirements are not considered. Thus, after negotiation, each user just selects the service with the lowest price.

In this experiment, the performance of the four approaches is evaluated by measuring three quantitative metrics: success rate, average agreed price of services, and average utility of users (i.e., buyers).

1) Success rate is the percentage of successful service selection. It is a ratio between the number of successful service selection and the total number of attempted service selection.
2) Average agreed price of services is calculated using the total price obtained by summing the agreed price of each sold service to divide the number of sold services.
3) Average utility of users represents user satisfaction level which is calculated using the total utilities obtained by all the users to divide the number of users.

In the following sub-sections, we first describe the experimental setup and then present the experimental results. After that, we will discuss and summarize the experiments.

A. EXPERIMENTAL SETUP

The experiment was conducted in five different scenarios. In the first scenario, it is measured that how the performance of the four approaches changes with the variation of user-to-provider ratio. In the second scenario, it is measured that how the performance of the approaches changes with the variation of the scale of the environment. In the third scenario, it is measured that how the performance of the approaches changes with the variation of the percentage of services which has QoS correlations. In the fourth scenario, it is measured that how the performance of the four approaches changes with different kinds of user requirement QoS correlations. In the fifth scenario, it is measured that how the performance of the four approaches changes with the variation of the probability of users and providers leaving and joining.

This experiment is simulated by using a general programming language, Java. Threads (Java objects) are used to simulate each agent in the programming. The total number of task types in the experiment is set to 10 and each buyer randomly has 2 to 10 tasks. For each type of task, there are totally 8 services which can be used to carry out this type of task, although these services have different QoS values. Each seller randomly has 0 to 8 such services. Each service has three attributes: response time, reliability and price. Similar to [6], [41], [42], we have defined the value of response time as a random integer in $[10, 50]$, and the value of reliability as a random integer in $[1, 10]$. The price is then based on the values of both response time and reliability. Generally, a fast response time and a high reliability mean a high price. Thus, the price is set to: $x \cdot respTime + y \cdot reli$, where $respTime$ is response time and $reli$ is reliability while x and y are set to -1 and 60, respectively. The setup of x and y is based on the setup of response time and reliability to ensure the value of price to be in the same magnitude as response time and reliability. Certainly, the three attributes can be presented in other ways. For example, reliability can be presented in a probability form. In this experiment, we simplify the presentation of the three attributes by using integers. This presentation, however, does not affect the relative performance of the four approaches.

Each buyer's requirements for response time and reliability are random integers in $[10, 50]$ and $[1, 10]$, respectively. For example, a buyer's requirements for response time and reliability are 30 and 5, respectively. This means that the buyer considers the services whose response time is less than or equal to 30 and reliability is larger than or equal to 5. In addition, the temporal discount factors, ζ_s^0 and ζ_b^0, are set to 1, while the temporal discount rates, k_s and k_b, are set to 0.98 (referring to Equations 1 and 2). The values of temporal discount factors and temporal discount rates are experimentally chosen to yield the best results. The experiment was run on an Intel i5-2450m 2.5GHz Windows 10 PC with 10GB RAM.

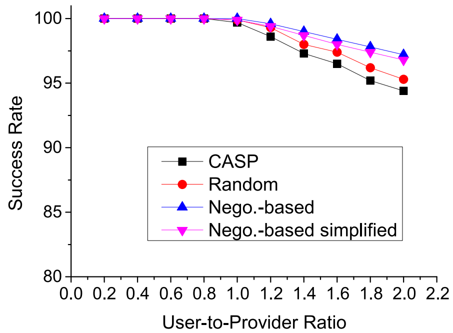
B. EXPERIMENTAL RESULTS

1) THE FIRST SCENARIO

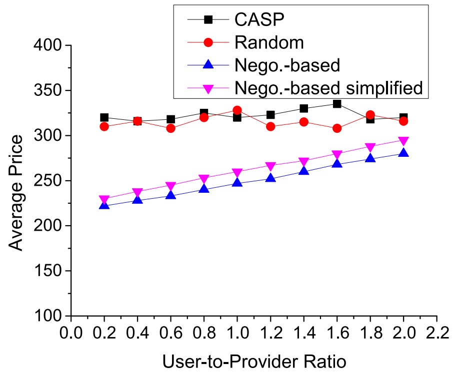
Fig. 3 shows the performance of the four approaches with the variation of user-to-provider ratio. In this scenario, the number of users and providers altogether is fixed at 500, and the percentage of services which has QoS correlations is fixed at 25%.

In Fig. 3(a), the success rates of the four approaches decrease with the increase of user-to-provider ratio. A large user-to-provider ratio means that there are many users but only a few providers in the environment. Thus, in this situation, the service competition becomes intense for users and a number of users is difficult to obtain required services. Specifically, the success rates of *Neog.-based* and *Nego.-based-simplified* approaches are higher than *CASP* and *Random* approaches, where the average difference is about 2.5%. *CASP* and *Random* approaches directly pick up qualified services for users without bargaining, and thus the service selection can be successful as long as there are qualified services. However, *CASP* and *Random* approaches consider only the services whose utilities can meet users' reserved utilities. In comparison, *Neog.-based* and *Nego.-based-simplified* approaches consider almost all the services[1] whose quality can meet user requirements irrespective of the prices, because the prices can be negotiated later.

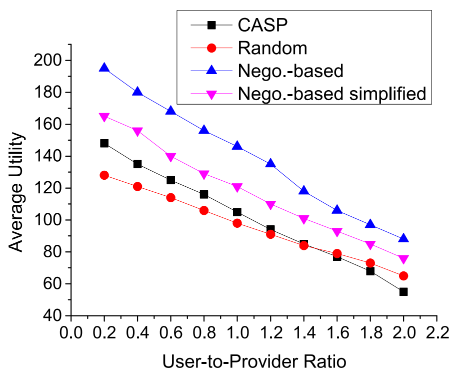
[1] Some services with very low utilities have been removed in **Algorithm 1** in order to save users' time and communication overhead.



(a) Success Rate



(b) Average Price



(c) Average Utility

FIGURE 3. Performance of the four approaches with different user-to-provider ratios.

Therefore, the candidate service sets in *Neog.-based* and *Nego.-based-simplified* approaches are usually larger than the candidate service sets in *CASP* and *Random* approaches. In general, more candidate services can lead to a higher success rate. It should also be noted that the success rate of *Random* approach is larger than that of *CASP* approach when the user-to-provider ratio is larger than 1.2. As the user-to-provider ratio is high, the number of services may not be enough for all the users. For example, there are two users, b_1 and b_2, and two services, se_1 and se_2. Suppose that 1) both se_1 and se_2 can meet b_1's requirement; 2) only se_1 can meet b_2's requirement; 3) utility of se_1 is larger than se_2 for b_1: $U_{b_1}(se_1) > U_{b_1}(se_2)$. If *CASP* approach is used, user b_1 will select se_1, as $U_{b_1}(se_1) > U_{b_1}(se_2)$. This selection, however, leads to user b_2 having no service to use, as only se_1 can meet b_2's requirement. If *Random* approach is used, user b_1 may select se_2, as b_1 selects each service with the same probability. Then, b_2 can take service se_1. Thus, *Random* approach can achieve a greater success rate than *CASP* approach especially when the number of services is relatively small compared to the number of users.

In Fig. 3(b), with the increase of user-to-provider ratio, the average agreed price under *CASP* and *Random* approaches remains relatively steady and rises gradually under *Neog.-based* and *Nego.-based-simplified* approaches. Moreover, it can be found that *Neog.-based* and *Nego.-based-simplified* approaches enable users to have services with lower prices than *CASP* and *Random* approaches, where the average difference is about 17%. In *CASP* and *Random* approaches, services are directly selected without bargaining and the selection is not based on price but based on utility (in *CASP* approach) or probability (in *Random* approach). Also, only the successfully selected services are counted to calculate the average agreed price. Thus, although the change of user-to-provider ratio affects the success rates under *CASP* and *Random* approaches, it does not affect much on the average agreed price under *CASP* and *Random* approaches. In *Neog.-based* and *Nego.-based-simplified* approaches, users are allowed to negotiate with providers. Thus, prices can be reduced during the negotiation process. When the user-to-provider ratio is low, users have strong bargain power and thus the prices can be reduced much. When the user-to-provider ratio is high, providers have strong bargain power and thus the prices are firm.

In Fig. 3(c), with the increase of user-to-provider ratio, the average utility of users decreases in all the four approaches, about 45% $\sim$ 50%. The reason is similar to the situation in Fig. 3(a), where the increase of user-to-provider ratio makes a number of users difficult to obtain required services. Since the users who do not obtain required services get 0 utility, with the increase of the number of such users, the average utility of users decreases. An interesting phenomenon is that when the user-to-provider ratio is less than 1.4, the average utility in *CASP* approach is greater than that in *Random* approach, but when the user-to-provider ratio is larger than 1.4, the situation reverses. When the user-to-provider ratio is small, the number of services is enough for all the users in the environment. As *CASP* approach selects the optimal services, i.e., the services with the highest utilities, while *Random* approach selects services randomly, the average utility in *CASP* approach is greater than that in *Random* approach. However, when the user-to-provider ratio is large, the number of services is not enough for all the users. In this situation, in *CASP* approach, the selection of services with the highest utilities may result in some users having no qualified services to use. We still use the above two-user and two-service example to explain this phenomenon. There are two users, b_1 and b_2, and two services, se_1 and se_2, where

1) both se_1 and se_2 can meet b_1's requirement; 2) only se_1 can meet b_2's requirement; 3) utility of se_1 is larger than se_2 for b_1: $U_{b_1}(se_1) > U_{b_1}(se_2)$. If *CASP* approach is used, user b_1 will select se_1 while user b_2 has no service to use. If *Random* approach is used, both b_1 and b_2 may have services to use. Thus, in *Random* approach, both b_1 and b_2 can obtain utilities, while in *CASP* approach, only b_1 obtains the utility whereas b_2 gets only 0 utility. Hence, the average utility of users in *Random* approach is greater than that in *CASP* approach.

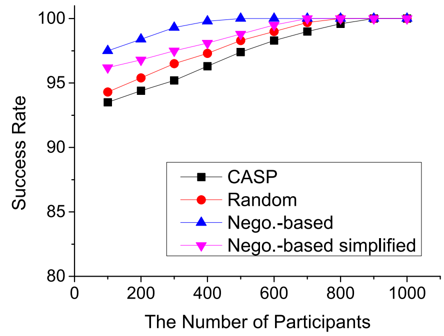
Moreover, in Fig. 3, it can be seen that the performance of *Neog.-based* approach is better than *Nego.-based-simplified* approach, i.e., higher success rate, lower average price and more average utility. This is mainly because service selection in *Neog.-based* approach is based on utility of users while service selection in *Nego.-based-simplified* approach is based on the price of services. In *Nego.-based-simplified* approach, as all the users focus on low price services, the low price services will become very popular and the prices of these services are hard to reduce, which negatively affects success rate, average agreed price and average utility. It should be noted that low price services do not mean high utility services (Equations 2 and 3), as each user's utility depends on the user's own reservation value on each attribute of a service. In *Neog.-based* approach, users select services based on the utility that a service can bring to the user. Therefore, some users may favor high-quality services which usually have high prices. As a result, users in *Neog.-based* approach do not always stare at low price services but almost all the services in the environment, which positively affects success rate, average agreed price and average utility.

2) THE SECOND SCENARIO

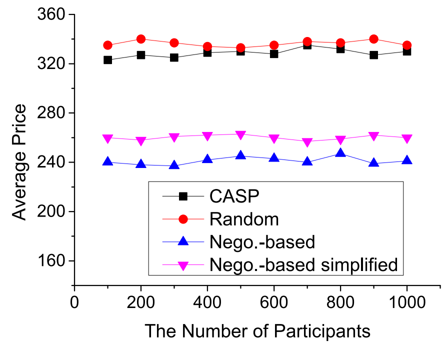
Fig. 4 demonstrates the performance of the four approaches with the variation of the scale of the environment, i.e., the variation of the number of participants. In this scenario, the user-to-provider ratio is fixed at 1, and the percentage of providers whose services have QoS correlations is fixed at 25%.

In Fig. 4(a), with the increase of the scale, the success rates of the four approaches increase gradually. With the increase of the number of participants, as the user-to-provider ratio is fixed, based on the experimental setup, the service-to-user ratio increases. This implies that each user may have more candidate services. Therefore, the service selection is more possible to be successful. This reason can also apply to the situation in Fig. 4(c), where with the increase of the scale, the average utility of the four approaches rises steadily. More candidate services enable each user to have a larger selection space and ensure each user to obtain required services. Thus, the average utility of users can increase.

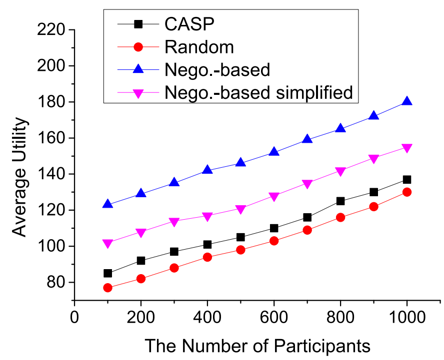
In Fig. 4(b), with the increase of the scale, the average agreed price keeps relatively stable under all the four approaches. As described above, there are more available services in the environment with the increase of the scale. However, as the description about Fig. 3(b), because service



(a) Success Rate



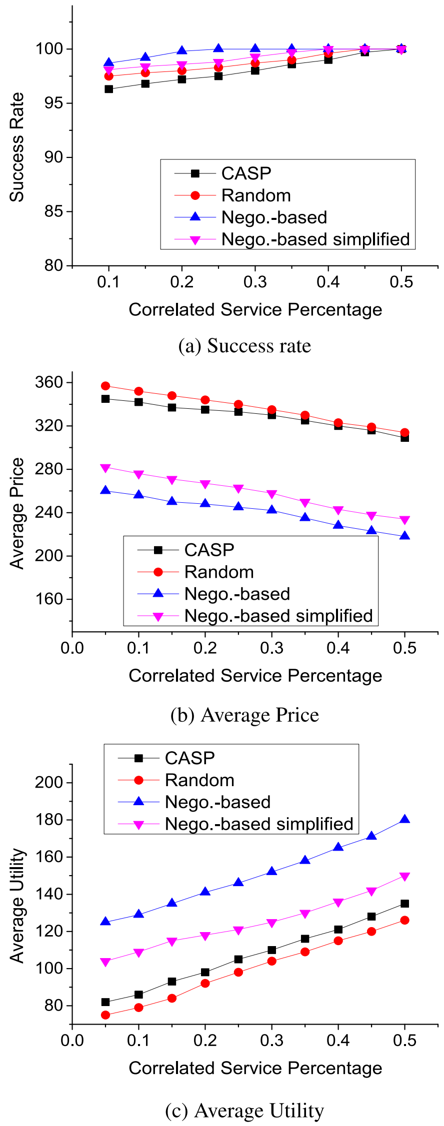
(b) Average Price



(c) Average Utility

FIGURE 4. Performance of the four approaches with different number of participants.

selection in *CASP* and *Random* approaches is not based on the prices of services, the change of the number of services does not have much impact on the average agreed price under *CASP* and *Random* approaches. In *Neog.-based* and *Nego.-based-simplified* approaches, services are selected based on negotiation. As the user-to-provider ratio is fixed at 1, with the increase of the scale, the number of users who negotiate with each provider keeps relatively steady. One of the factors which affect a provider's proposed price is the number of temporary agreements that the provider has achieved (Equation 6). Since the number of users who negotiate with each provider keeps relatively steady, the number of temporary agreements that each provider may achieve also keeps relatively steady. Thus, the increase of the scale does not affect the agreed price much in *Neog.-based* and *Nego.-based-simplified* approaches.

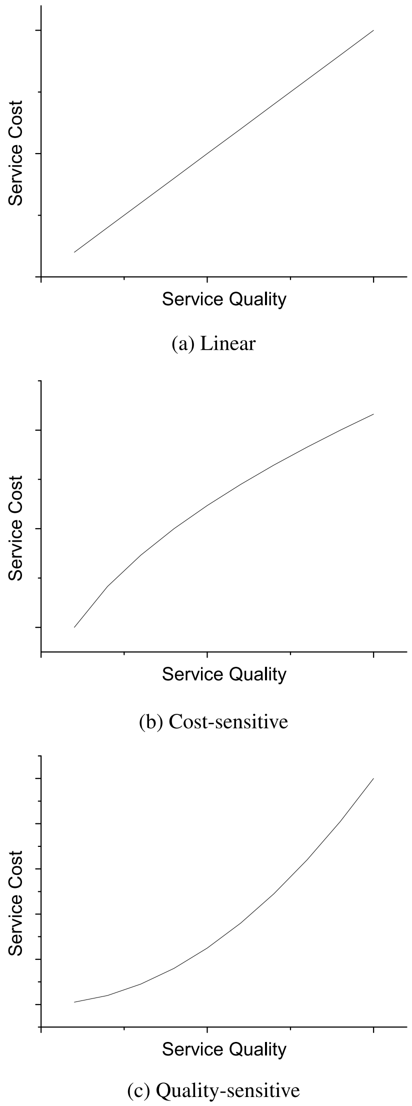


(a) Success rate

(b) Average Price

(c) Average Utility

FIGURE 5. Performance of the four approaches with different correlated service percentages.



(a) Linear

(b) Cost-sensitive

(c) Quality-sensitive

FIGURE 6. Different kinds of user requirement QoS correlations.

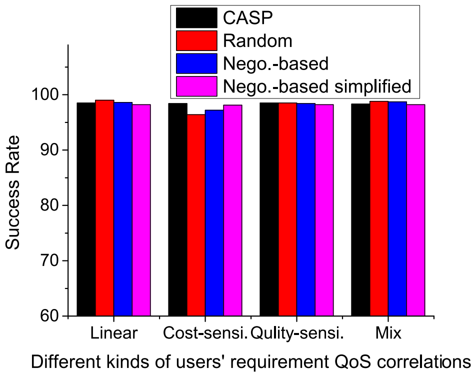
3) THE THIRD SCENARIO

Fig. 5 displays the performance of the four approaches with the variation of correlated service percentage, i.e., the percentage of services which has QoS correlations. In this scenario, the user-to-provider ratio is fixed at 1, and the number of participants is fixed at 500.

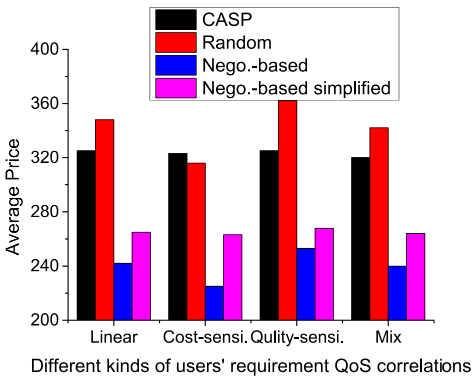
In Fig. 5, it can be seen that with the increase of correlated service percentage, the success rate and the average utility of the four approaches rise, while the average agreed price under the four approaches decreases. The increase of correlated service percentage implies that providers either reduce the price of services or improve the quality of services. Thus, the average price can reduce while the average utility can increase. For the success rate, as providers reduce the price or improve the quality of services, some services, which are previously not qualified for specific users, are now acceptable to these users. Therefore, the success rate also increases.

4) THE FOURTH SCENARIO

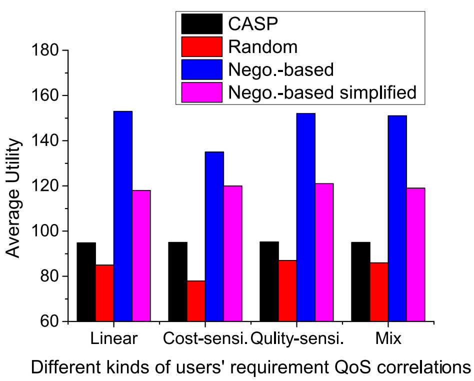
This scenario is to evaluate the performance of the four approaches when the kinds of user requirement QoS correlations change. Similar to [41]–[43], we have implemented three different kinds of user requirement QoS correlations in Fig.6. The first kind, Fig. 6(a), represents *linear* users who are sensitive to both cost and quality. These users care about the cost-performance ratio of services. The second kind, Fig. 6(b), represents *cost-sensitive* users. These users usually have budget constraints and thus, care about the cost more than the quality of services. The third kind, Fig. 6(c), represents *quality-sensitive* users. These users usually have special requirements for quality of services and have sufficient budget. Thus, they care about the quality more than the cost of services. In this scenario, the user-to-provider ratio is fixed at 1, the number of users and providers altogether is fixed at 500, and the percentage of providers whose services have QoS correlations is fixed at 25%.



(a) Success Rate



(b) Average Price



(c) Average Utility

FIGURE 7. Performance of the four approaches with different kinds of user requirement QoS correlations.

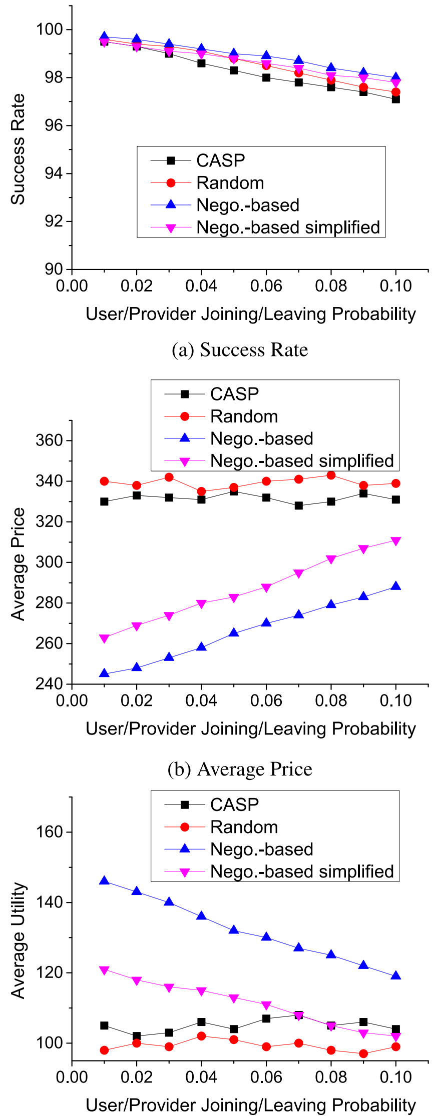
Fig. 7 demonstrates the performance of the four approaches with the change of the kinds of user requirement QoS correlations. In addition to the above-mentioned three kinds, the fourth kind, *Mix*, means that each user is randomly set to one of the three kinds: *linear*, *cost-sensitive* and *quality-sensitive*. As *CASP* and *Nego.-based-simplified* approaches do not consider QoS correlations of user requirements, their performance keeps relatively stable in all the cases of this scenario. In Fig. 7(a), it can be seen that the success rates of *Random* and *Nego.-based* approaches are lower in the *cost-sensitive* case than in the three other cases. As described above, the cost-sensitive users usually imply that the users have a limited budget. When all the users are cost-sensitive, all of them favor relatively low price services. However, the number of low price services may not be large enough to supply all the users. Thus, some users may not be able to obtain their required services due to their budget limitation. Unlike the *cost-sensitive* case, when all the users are quality-sensitive, all of them prefer high-quality services. Although the number of high-quality services may not be large enough to offer all the users, some users can still select relatively low-quality services and this does not harm the success rate. In the *linear* case, users do not focus solely on the price or quality of services but they care about the cost-performance ratio of services. These users can accept both low price and high price services as long as the quality of the services can match up to their prices. Therefore, the success rate, in this case, is also higher than the *cost-sensitive* case. In the *Mix* case, all the three types of users exist together. Then, the cost-sensitive users select low price services, the quality-sensitive users pick up high-quality services, and the linear users choose the services with reasonable quality and prices. As the selection spaces of the three types of users do not overlap in a large extent, the success rate, in this case, is higher than the *cost-sensitive* case as well. Moreover, the relatively low success rate in the *cost-sensitive* case yields a relatively low average utility in the same case as shown in Fig. 7(c), since unsuccessful users get 0 utility and all the users are taken into account when calculating average utility. In Fig. 7(b), the average price is also the lowest in the *cost-sensitive* case. The cost-sensitive users focus on low price services only due to their budget limitation, whereas the users in other cases do not solely care about the price. Thus, users in the *cost-sensitive* case can achieve the lowest price among the four cases.

5) THE FIFTH SCENARIO

This scenario is to evaluate how the four approaches work in a dynamic environment, where users and providers may join or leave. In this scenario, the user-to-provider ratio is fixed at 1, the number of users and providers altogether is fixed at 500, and the percentage of providers whose services have QoS correlations is fixed at 25%.

In Fig. 8(a), it can be found that with the increase of joining/leaving probability, the success rate of the four approaches decreases. This is because the leaving of providers may result in that users cannot find proper services before their deadlines. Although new providers may join the environment, new users may also join the environment and compete with existing users for services.

In Figs. 8(b) and 8(c), with the increase of joining/leaving probability, the average price and utility keep relatively stable under *CASP* and *Random* approaches. However, the average price increases while the average utility decreases under *Neog.-based* and *Nego.-based-simplified* approaches. In *Neog.-based* and *Nego.-based-simplified* approaches, users negotiate with providers about the price of services. When some providers leave, users who are negotiating with these providers have to close the negotiation unexpectedly.



(a) Success Rate

(b) Average Price

(c) Average Utility

FIGURE 8. Performance of the four approaches with different user/provider joining/leaving probabilities.

Although new providers may join, the users have to start a new negotiation with these new providers. However, as time has progressed, the users' deadline is approaching. Then, in order to achieve agreements, users may have to accept relatively high price services or low utility services. This situation does not exist in *CASP* and *Random* approaches, as users in these two approaches do not bargain with providers. Hence, the average price and utility are not affected by the change of joining/leaving probability under *CASP* and *Random* approaches.

C. DISCUSSION OF THE EXPERIMENT

In this section, four aspects of the experiment are discussed.

1) CHOICE OF THE COMPARED APPROACHES

As this paper is the first one which considers QoS correlations of user requirements, there are not existing closely related approaches for comparison. We, thus, employ a relatively related approach, i.e., *CASP*, and create two approaches ourselves, i.e., *Random* and *Nego.-based-simplified*. Among existing approaches, *CASP* approach is the most related approach to ours, because it also considers QoS correlations and directly addresses the service selection problem. *Random* approach is a common approach, as it reflects the fact that some people are indecisive when they have multiple choices and then randomly pick up one. *Nego.-based-simplified* approach is a simplified version of the proposed *Nego.-based* approach. Users in *Nego.-based-simplified* approach consider only the price of services. This approach reflects the fact that some people are sensitive to price. They bargain with providers and then among all the services, which can meet their requirements, they pick up the cheapest one. Certainly, with proper modification, existing service selection approaches, e.g., mixed integer programming [15], could also be used for comparison in the experiment. We leave this as one of our future studies.

2) SETUP OF EXPERIMENTAL SCENARIOS

Five scenarios are used in the experiment: the change of user-to-provider ratio, the change of the number of participants in the environment, the change of the percentage of services which has QoS correlations, the different kinds of user requirement QoS correlations, and the change of user/provider leaving and joining the environment. The first four scenarios are used to evaluate the four approaches in a static environment, whereas the last scenario is used to evaluate them in a dynamic environment. Specifically, the change of user-to-provider ratio is used to simulate different levels of service competition so as to evaluate the suitability of the four approaches. The change of the number of participants is used to simulate different scales of the environment so as to evaluate the scalability of the four approaches. The change of the percentage of services which has QoS correlations is used to simulate different percentages of providers who have promotions so as to evaluate the flexibility of the four approaches. The different kinds of user requirement QoS correlations are used to simulate different kinds of users: linear, cost-sensitive, and quality-sensitive, so as to evaluate the applicability of the four approaches. The change of user/provider leaving and joining the environment is used to simulate different levels of dynamism of the environment so as to evaluate the robustness of the four approaches.

Other scenarios, certainly, can also be set up in the experiments, e.g., the change of the number of services and tasks in the environment. Actually, against this scenario, we set that each provider randomly has 0 to 8 services for each type of task and each user randomly has 2 to 10 tasks. Thus, this scenario has been covered in the experiment to some extent.

3) COMPREHENSIVENESS OF THE EXPERIMENTS

Three quantitative metrics are measured in the experiment: success rate, average agreed price of services, and average utility of users. Other metrics, certainly, can also be included in the experiment, e.g., average utility of providers. This metric, average utility of providers, is opposite to the metric, average utility of users. If users' average utility increases, then providers' average utility decreases and vice versa. Thus, the metric, average utility of providers, has been covered in the experiment to some extent.

4) RESULTS OF THE EXPERIMENTS

The results obtained in the experiment are averaged by running each approach 100 times in each scenario. Specifically, in the first three scenarios, once each user finishes service selection, the experiment is over. In the fourth scenario, as new users and providers may join and existing users and providers may leave, the experiment keeps running for 500 time steps. At each time step, each user or provider may leave with a specific probability and a new user or provider may join with a specific probability as well. The length of a time step is set to 200ms in the experiment.

D. SUMMARY

Overall, according to the experimental results, the proposed *Nego.-based* approach outperforms the three other approaches in various settings. Specifically, the proposed *Nego.-based* approach can achieve about 2%, 3% and 4% more success rate than *Nego.-based-simplified* approach, *Random* approach and *CASP* approach, respectively, on average. Also, *Nego.-based* approach reaches about 5%, 48% and 50% less price than *Nego.-based-simplified* approach, *Random* approach and *CASP* approach, respectively, on average. Meanwhile, *Nego.-based* approach gains about 20%, 25% and 28% more utility than *Nego.-based-simplified* approach, *CASP* approach and *Random* approach, respectively, on average. Based on the fact that *Nego.-based* approach outperforms *Nego.-based-simplified* approach, it can be concluded that flexible users are easier to be successful and can achieve more utility than those users who focus only on prices.

VI. CONCLUSION AND FUTURE WORK

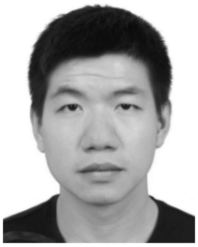
This paper proposes a novel service selection approach which not only considers QoS correlations of services but also accounts for QoS correlations of user requirements. This approach, to the best of our knowledge, is the first service selection approach which considers QoS correlations of user requirements. Moreover, this approach is decentralized which can avoid the single point of failure. Compared to other approaches, the proposed approach can (1) achieve a higher success rate; (2) reach a much lower price for users; (3) gain much more utility for users.

In future research, we will make several improvements: (1) We intend to extend our approach to the multi-tenant environment. Currently, although this approach is developed in a multi-user environment, it does not consider multi-tenancy. Multi-tenancy is an important feature in current large scale distributed SBSs [44], and should be taken into account in our approach in the next step; (2) Once the above extension is finished, we will evaluate our approach in real environments; (3) During the evaluation, we will involve more existing approaches for comparison, because, with proper modification, some existing approaches may be able to handle user requirement QoS correlations.

REFERENCES

[1] S. Deng, Z. Xiang, J. Yin, J. Taheri, and A. Y. Zomaya, "Composition-driven IoT service provisioning in distributed edges," *IEEE Access*, vol. 6, pp. 54258–54269, 2018.
[2] M. Vargas-Santiago, L. Morales-Rosales, S. Pomares-Hernández, and K. Drira, "Autonomic Web services enhanced by asynchronous checkpointing," *IEEE Access*, vol. 6, 5538–5547, 2018.
[3] F. Dahan, H. Mathkour, and M. Arafah, "Two-step artificial bee colony algorithm enhancement for QoS-aware Web service selection problem," *IEEE Access*, vol. 7, pp. 21787–21794, 2019.
[4] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu, "Web service recommendation via exploiting location and QoS information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1913–1924, Jul. 2014.
[5] S. Hwang, C. Hsu, and C. Lee, "Service selection for Web services with probabilistic QoS," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 467–480, May/Jun. 2015.
[6] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.
[7] T. Yu, Y. Zhang, and K. J. Lin, "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Trans. Web*, vol. 1, no. 1, p. 6, May 2007.
[8] T. Yu and K.-J. Lin, "Service selection algorithms for Web services with end-to-end QoS constraints," *Inf. Syst. E-Bus. Manage.*, vol. 3, no. 2, pp. 103–126, Jul. 2005.
[9] R. J. R. Raj and T. Sasipraba, "Web service selection based on QoS constraints," in *Proc. Trendz Inf. Sci. Comput. (TISC)*, Dec. 2010, pp. 156–162.
[10] P. Bartalos and M. Bieliková, "QoS aware semantic Web service composition approach considering pre/postconditions," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2010, pp. 345–352.
[11] H. Zheng, W. Zhao, J. Yang, and A. Bouguettaya, "QoS analysis for Web service compositions with complex structures," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 373–386, Jul./Sep. 2013.
[12] G. Zou, Q. Lu, Y. Chen, R. Huang, Y. Xu, and Y. Xiang, "QoS-aware dynamic composition of Web services using numerical temporal planning," *IEEE Trans. Services Comput.*, vol. 7, no. 1, pp. 18–31, Jan./Mar. 2014.
[13] N. Gao, W. Shao, and F. D. Salim, "Predicting personality traits from physical activity intensity," 2019, *arXiv:1906.07864*. [Online]. Available: https://arxiv.org/abs/1906.07864
[14] S. Deng, H. Wu, D. Hu, and J. L. Zhao, "Service selection for composition with QoS correlations," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 291–303, Mar. 2016.
[15] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Trans. Softw. Eng.*, vol. 33, no. 6, pp. 369–384, Jun. 2007.
[16] I. Trummer, B. Faltings, and W. Binder, "Multi-objective quality-driven service selection—A fully polynomial time approximation scheme," *IEEE Trans. Softw. Eng.*, vol. 40, no. 2, pp. 167–191, Feb. 2014.
[17] E. Di Nitto, M. D. Penta, A. Gambi, G. Ripa, and M. L. Villani, "Negotiation of service level agreements: An architecture and a search-based approach," in *Proc. Int. Conf. Service-Oriented Comput. (ICSOC)*, 2007, pp. 295–306.
[18] S. Paurobally, V. Tamma, and M. Wooldridge, "A framework for Web service negotiation," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, no. 4, p. 14, Nov. 2007.
[19] M. G. Buscemi and U. Montanari, "QoS negotiation in service composition," *J. Logic Algebr. Program.*, vol. 80, pp. 13–24, Jan. 2011.
[20] F. H. Zulkernine and P. Martin, "An adaptive and intelligent SLA negotiation system for Web services," *IEEE Trans. Services Comput.*, vol. 4, no. 1, pp. 31–43, Jan./Mar. 2011.
[21] N. Ranaldo and E. Zimeo, "Capacity-driven utility model for service level agreement negotiation of cloud services," *Future Gener. Comput. Syst.*, vol. 55, pp. 186–199, Feb. 2016.
[22] V. Gabrel, M. Manouvrier, K. Moreau, and C. Murat, "QoS-aware automatic syntactic service composition problem: Complexity and resolution," *Future Gener. Comput. Syst.*, vol. 80, pp. 311–321, Mar. 2018.

[23] A. Moustafa, M. Zhang, and Q. Bai, "Trustworthy stigmergic service compositionand adaptation in decentralized environments," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 317–329, Mar./Apr. 2016. doi: 10.1109/TSC.2014.2298873.
[24] A. Moustafa and M. Zhang, "Multi-objective service composition in uncertain environments," *IEEE Trans. Services Comput.*, to be published. doi: 10.1109/TSC.2015.2443785.
[25] S. Deng, L. Huang, D. Hu, J. L. Zhao, and Z. Wu, "Mobility-enabled service selection for composite services," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 394–407, May/Jun. 2016.
[26] S. Basu, F. Casati, and F. Daniel, "Toward Web service dependency discovery for SOA management," in *Proc. IEEE Int. Conf. Services Comput.*, Jul. 2008, pp. 422–429.
[27] D. Romano, M. Pinzger, and E. Bouwers, "Extracting dynamic dependencies between Web services using vector clocks," in *Proc. IEEE Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Dec. 2011, pp. 1–8.
[28] G. Kang, J. Liu, M. Tang, and B. Cao, "Web service selection algorithm based on principal component analysis," *J. Electron.*, vol. 30, no. 2, pp. 204–212, Apr. 2013.
[29] G. Kang, J. Liu, M. Tang, and Y. Xu, "An effective dynamic Web service selection strategy with global optimal QoS based on particle swarm optimization algorithm," in *Proc. IEEE 26th Int. Parallel Distrib. Process. Symp. Workshops PhD Forum*, May 2012, pp. 2280–2285.
[30] F. Tao, D. Zhao, Y. He, and Z. Zhou, "Correlation-aware resource service composition and optimal-selection in manufacturing grid," *Eur. J. Oper. Res.*, vol. 201, no. 1, pp. 129–143, Feb. 2010.
[31] L. Barakat, S. Miles, and M. Luck, "Efficient correlation-aware service selection," in *Proc. IEEE 19th Int. Conf. Web Services*, Jun. 2012, pp. 1–8.
[32] H. Jin, X. Yao, and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," *J. Intell. Manuf.*, vol. 28, no. 8, pp. 1947–1960, Dec. 2017. doi: 10.1007/s10845-015-1080-2.
[33] Z. Ye, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 382–393, May/Jun. 2016.
[34] F. Di Giunta and N. Gatti, "Bargaining over multiple issues in finite horizon alternating-offers protocol," *Ann. Math. Artif. Intell.*, vol. 47, nos. 3–4, pp. 251–271, Aug. 2006.
[35] N. Gatti, F. Di Giunta, and S. Marino, "Alternating-offers bargaining with one-sided uncertain deadlines: An efficient algorithm," *Artif. Intell.*, vol. 172, nos. 8–9, pp. 1119–1157, May 2008.
[36] B. An, N. Gatti, and V. Lesser, "Alternating-offers bargaining in one-to-many and many-to-many settings," *Ann. Math. Artif. Intell.*, vol. 77, nos. 1–2, pp. 67–103, Jun. 2016.
[37] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, Jan. 1982.
[38] B. An, V. Lesser, and K. M. Sim, "Strategic agents for multi-resource negotiation," *Auton. Agents Multi-Agent Syst.*, vol. 23, no. 1, pp. 114–153, Jul. 2011.
[39] B. An, N. Gatti, and V. Lesser, "Bilateral bargaining with one-sided uncertain reserve prices," *Auton. Agents Multi-Agent Syst.*, vol. 26, no. 3, pp. 420–455, May 2013.
[40] K. M. Sim, "Agent-based cloud computing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 4, pp. 564–577, Oct. 2012.
[41] K. K. Fletcher, X. F. Liu, and M. Tang, "Elastic personalized nonfunctional attribute preference and trade-off based service selection," *ACM Trans. Web*, vol. 9, no. 1, p. 1, Jan. 2015.
[42] Y. Wang, Q. He, D. Ye, and Y. Yang, "Service selection based on correlated QoS requirements," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2017, pp. 241–248.
[43] Y. Wang, Q. He, D. Ye, and Y. Yang, "Formulating criticality-based cost-effective fault tolerance strategies for multi-tenant service-based systems," *IEEE Trans. Softw. Eng.*, vol. 44, no. 3, pp. 291–307, Mar. 2017.
[44] Q. He, J. Han, F. Chen, Y. Wang, R. Vasa, Y. Yang, and J. Hai, "QoS-aware service selection for customisable multi-tenant service-based systems: Maturity and approaches," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun./Jul. 2015, pp. 237–244.



DONGWEI LI was born in Wuhan, China, in 1984. He received the M.Sc. degree in software engineering from Wuhan University, China, in 2010. He is currently pursuing the Ph.D. degree with the Beijing Institute of Technology, Beijing, China. His research interests include data mining and cloud computing.

DAYONG YE received the M.Sc. and Ph.D. degrees in computer science from the University of Wollongong, Wollongong, NSW, Australia, in 2009 and 2013, respectively. He is currently a Research Fellow of cybersecurity with the University of Technology Sydney, Ultimo, NSW, Australia. His current research interests include privacy preserving, artificial intelligence, and cloud computing.

NAN GAO received the M.Sc. degree in software engineering from the Beijing Institute of Technology, in 2016. She is currently pursuing the Ph.D. degree in computer science with RMIT University, Melbourne, VIC, Australia. Her current research interests include applied machine learning for smart buildings and wearable sensing.

SHULIANG WANG received the Ph.D. degree from Wuhan University, China, in 2002, and Hongkong Polytechnic University, Hong Kong, in 2003. He is currently a Full Professor with the Beijing Institute of Technology, China. His research interests include spatial data mining, distributed computing, and data field.