Abstract

In QoS-aware service selection, a service requester seeks to maximize his utility by selecting service providers that charge the lowest service prices while meeting the requester's global QoS requirements. In existing selection approaches, a service requester focuses on finding providers based on their QoS and thereby ignores their service prices that could change with their QoS. High QoS may provide more benefits, but may require a high service price. As a result, the highest QoS may not produce the maximum utility. A service requester and service providers have a conflicting interest over service prices. We model the process of QoS-aware service selection as a price competition game. In the case where the providers' minimum prices are public knowledge, we propose an approach to compute a $\epsilon$-Nash equilibrium that maximizes the service requester's utility. However, in general, a service provider would not reveal its minimum acceptable price. Thus, it is important for a service requester to predict the minimum price for a service that meets its QoS requirements. We propose a collaborative approach to predicting a provider's minimum price for a desired QoS based on prior usage experiences. Based on the prediction results of the service providers' minimum prices, we propose a multi-round mechanism where the service requester chooses a service for each task by using a Vickrey auction with reserve price. The experimental results show our approach can find the optimal service providers efficiently and effectively.

Keywords: Service Selection, QoS, Price Prediction, Multi-round Vickrey Auction

1. INTRODUCTION

An important motivation for service-oriented computing (SOC) is to compose elementary services to realize a complex service [1]. We model a complex service as an abstract process that contains a set of tasks. In QoS-aware service selection, a service requester has global QoS requirements, such as response time and throughput, on the composite service. The service requester seeks to find the optimal service providers to achieve an optimization goal, such as the maximization of utility, while satisfying the global QoS requirements.

The service requester and the service providers are rational and self-interested [1]. They seek to maximize their respective utilities. A service requester's utility from a task is the difference between the price it pays and the benefit it receives once the service provider completes the task. We define a provider's utility from a task as the difference between its cost of fulfilling the task and the price it receives. Clearly, although a service requester and service providers have competing interests with respect to the task performance and QoS to be delivered and the price to be paid, they have a common interest in coming to an agreement. There is a game between the service requester and the service providers.

Existing approaches [2][3] focus on designing reconfigurable service providers that offer different QoS such that a service requester can select its desired QoS. The reconfigurable providers increase their selection space thereby assisting a service requester to select the best possible QoS to meet his QoS requirements. Not surprisingly, different QoS impose different service costs on providers [4]. Thus, the objective of a service requester is to assess such QoS space and obtain an optimal solution. In general, a service requester can achieve that objective only when he has complete information about service providers. That is, the service requester must know which service providers provide the minimum service price for the QoS requested. However, in general, a service provider would not expose his minimum acceptable service price to the service requester and other service providers. Thus, a service requester might have incomplete information about service providers, thereby suggesting the need for predicting the minimum service prices of service providers.

We propose to predict minimum service prices using a collaborative approach. The approach has been used in several works to predict QoS based on past usage experiences with service providers [5][6][7][8]. A limitation of the existing works is that they do not consider the situation where a service price is negotiable and dependent upon the QoS. However, in general, the service price has a relationship with QoS. Thus, finding an optimal service provider based on only QoS may not be useful to maximize a requester's utility.

Moreover, some approaches have been proposed to estimate the service price at the provider side [9], assuming the provider is an expert in evaluating its service cost and determining what price it must charge. The approach may not be helpful to the requester since the requester may not
be an expert in evaluating the provider’s service cost. Thus, we need an effective and efficient approach for a service requester to predict providers’ minimum service prices given its QoS requirements, thereby finding the optimal service providers. Wang and Du [10] proposed an approach to estimate the minimum service price from a lower value to a higher value for each task, until the service requester and one of the service providers agree about the QoS desired by the requester and the service price demanded by the service provider. Adopting Wang and Du’s approach, we can obtain the past usage experience records including QoS, the predictions of providers’ minimum service prices, and so on.

We provide the following contributions:

- First, we model the process of QoS-aware service selection as a price competition game where the service providers compete on the service prices. In the case where the providers’ minimum prices are public information, we propose an approach to compute a (pure) ε-Nash equilibrium that maximizes the service requester’s utility, where no service provider can increase his utility by more than ε > 0 by changing his price.

- Second, in the case where the service requester has incomplete information about the service providers, we propose a collaborative approach to predict the minimum acceptable prices of providers. A service requester predicts the service providers’ minimum prices for its desired QoS using the past usage experiences of other requesters with the same providers, thereby finding optimal service providers. A service price could change over time. We assume that a recent prediction of the minimum service price is more accurate than an old prediction. We thus propose a time-aware collaborative algorithm for a service price prediction.

- Third, we propose a mechanism called multi-round Vickrey auction based on the prediction results of the service providers’ minimum prices. In each round, the service requester chooses a service for each task by using a Vickrey auction with reserve price. We prove that it is a dominant strategy for a service provider to submit his actual minimum price if and only if his minimum price is less than the reserve price. Overall, the experimental results show that our approach is more efficient and effective than Wang and Du’s approach.

**Organization** The rest of the paper is organized as follows. Section 2 provides backgrounds of the QoS-aware service selection problem. Section 3 describes a collaborative approach to predict the minimum service prices of candidate service providers by using past usage experience. Section 4 describes the multi-round Vickrey auction. Section 5 empirically evaluates our prediction-based approach. Section 6 describes the related work. Finally, Section 7 concludes the paper with a discussion of future work.

## 2. BACKGROUND

### 2.1 Game-Theoretic QoS-Aware Service Selection

We specify a complex service requested by a service requester as an abstract process that contains a set of tasks \( T = \{t_1, \ldots, t_n\} \) to be executed by concrete service providers. There is a set of service providers \( A = \{a_1, \ldots, a_m\} \), which provide a service matrix \( S = (s_{ij})_{mn} \) for performing the tasks. Each entry \( s_{ij} \) represents a service that a service provider \( a_j \) provides for a task \( t_i \). We regard \( s_{ij} = null \) as that the service provider \( a_j \) cannot complete the task \( t_i \).

The service requester has a global QoS requirement for the requested service. We write a **global QoS requirement** as a vector \( q_G = (q_{G,1}, \ldots, q_{G,k}) \), where \( q_{G,k} \) represents the admissible value of the \( k \)-th QoS attribute. For example, a global QoS requirement is that the response time must be less than or equal to 2.5 (seconds) and the throughput must be less than or equal to 3.5 (kbps). This can be specified as \( q_G = (q_{G,1}, q_{G,2}) = (2.5, 3.5) \), where \( q_{G,1} \) is the admissible value of the overall response time and \( q_{G,2} \) is the admissible value of the overall throughput. The service requester also has a **budget Budget**, that is, the amount of money that it is willing to spend for the tasks. A **service request** \( (T, q_G, Budget) \) refers to a set of tasks \( T \), a global QoS requirement \( q_G \) and a budget **Budget**. To realize the service request, the service requester needs to find exactly one service for each task while the aggregated QoS meets the global QoS requirement and the total service price does not exceed the budget.

The objective of the service requester is to find a set of service providers that can realize the service request to achieve an optimization goal, such as the maximization of utility. The utility that the service requester from a task is the difference between the service price it pays and his valuation of the task. We observe that, given the services provided by the service providers, the service requester seeks to purchase the services that can maximize his utility according to his valuation and their service prices (the valuation minus the total service price). Naturally, the service providers strive to maximize their own utilities. Every service provider would adjust his service price to react to his competitors (other service providers).

In this paper, we model the process of QoS-aware service selection as a **price competition game**.

First, the service price has a relationship with the QoS. Providing better QoS may require higher price. The best QoS thus may not be the optimal choice for the service requester. The service requester decides a set of local QoS requirements for the tasks while ensuring that the satisfaction of the local QoS requirements guarantees the
satisfaction of the global QoS requirement \( q_G \). We write a **local QoS requirement** for a task \( t_i \in T \) as a vector \( q_i = (q_{i1},...,q_{iK}) \). For example, suppose that the requested service is composed of three sequential tasks: \( t_1, t_2 \) and \( t_3 \). Considering two QoS attributes: response time and throughput, given a global QoS requirement \( q_G = (2.5, 3.5) \), there are three local QoS requirements \( q_1 = (0.5, 4.0), q_2 = (0.3, 4.5) \) and \( q_3 = (1.0, 5.0) \). We can compute the aggregated QoS by using the aggregation functions introduced in [11]. If satisfying the local QoS requirements, the overall response time is less than or equal to 0.5+0.3+1.0 = 1.8<2.5 (seconds) and the overall throughput is greater than or equal to as \( min(4.0,4.5,5.0) = 4.0 > 3.5 \) (kbps). It can be inferred that if these local QoS requirements are satisfied, the global QoS requirement can be satisfied. Consider the local QoS requirements \( \{q_i\}_{i=1}^n \) for the tasks \( T = \{t_1, ..., t_n\} \), we denote the aggregated QoS by \( agg_{i \in T} q_i \), and then denote the constraint that guarantees the satisfaction of the global QoS requirement as \( agg_{i \in T} q_i \geq q_G \). There exist some approaches [14] for verifying service providers' QoS that they claim. The QoS verification is not interesting from our perspective. In the paper, we assume that the service providers would report their true QoS (except their service prices).

Second, given a local QoS requirement \( q_i \) for the task \( t_i \), each service provider \( \forall \alpha_j \in A \) sets a price, denoted by \( p_{i,j} \), for his service \( s_{i,j} \) while satisfying this local QoS requirement. Note that inherently high price (i.e., \( p_{i,j} = +\infty \)) is regarded as that the service provider cannot complete the task (i.e., \( s_{i,j} = \text{null} \)) or that the service provider cannot satisfy the local QoS requirement. Given a set of local QoS requirements \( \{q_i\}_{i=1}^n \), there is a \( m \times n \) price matrix \( p = (p_1, ..., p_n) \), where \( p_i \) is a price vector \( (p_{i,1}, ..., p_{i,m}) \) of the services \( S_i \) (\( S_i \) represents \( i \)-th column vector in \( S \)) for performing the task \( t_i \). While facing a price matrix \( p \), the service requester will choose a service for each task according to a decision map \( X \) that associated each price matrix \( p \) to a set of services.

\[
X(p) = \arg \max_{R \in S, \infty \times S} \sum_{s_{i,j} \in R} (o_i(q_i) \cdot p_{i,j}) 

\text{subject to } \sum_{s_{i,j} \in R} p_{i,j} \leq \text{Budget}
\]

where \( o_i(q_i) \) is the service requester's valuation function of the task \( t_i \in T \). We call \( X(p) \) the outcome of the price competition game. The service requester will pay \( p_{i,j} \) for a service \( s_{i,j} \) if the service requester chooses the service for completing task \( t_i \), i.e., \( s_{i,j} \in X(p) \). The utility of the service requester is:

\[
v(p) = \sum_{s_{i,j} \in X(p)} o_i(q_i) \cdot p_{i,j}
\]

For example, given the local QoS requirements \( q_1 = (0.5, 4.0), q_2 = (0.3, 4.5) \) and \( q_3 = (1.0, 5.0) \) for the tasks \( T = \{t_1, t_2, t_3\} \). Suppose that there are three service providers \( A = \{\alpha_1, \alpha_2, \alpha_3\} \) and their service prices are described as a price matrix.

\[
p = \begin{pmatrix} 213.5 & 231.8 & 189.3 \\ 253.2 & +\infty & 148.7 \\ 234.6 & 211.6 & 170.1 \end{pmatrix}
\]

According to the decision map (1), the service requester would choose a service that charges the lowest price for each task while the total price is less than or equal to the budget. Hence, if the budget is greater than or equal to \( 573.82 \), there is \( X(p) = \{s_{1,1}, s_{2,3}, s_{3,2}\} \). This means that the service requester chooses Service \( s_{1,1} \) to perform Task \( t_1 \) and pays 213.5 for the service, chooses Service \( s_{2,3} \) to perform Task \( t_2 \) and pays 211.6 for the service and chooses Service \( s_{3,2} \) to perform Task \( t_3 \) and pays 148.7 for the service. However, if the budget is less than \( 573.82 \), the outcome is empty.

### 2.2 Problem Analysis

Each service provider has minimum price of providing a service, presumably his service cost. His service price would be greater than or equal to his minimum price.

In general, a service provider has different minimum prices for satisfying different QoS requirements and different service providers have different minimum prices for completing a task. Wang and Du [10] proposed a two-step method for estimating the minimum price for a desired QoS. First, they introduce the **price-performance index** to describe a service provider’s capability to offer the minimum price for a desired QoS. They define the price-performance index, \( \theta_{i,j} \), for a service provider \( \alpha_j \in A \) for completing a task \( t_i \in T \). Second, they estimate a **price function** to describe the relationship between the QoS, the price-performance index and the minimum price. They then estimate the minimum price in terms of the price function, the price-performance index and the desired QoS. They denote the minimum price of a service provider \( \alpha_j \) for completing a task \( t_i \) by \( c_i(\theta_{i,j}, q_i) \), where \( c_i(\bullet, \bullet) \) is the price function of the task \( t_i \), \( \theta_{i,j} \) is the price-performance index of the service provider and \( q_i \) is the local QoS requirement. We assume that, given a local QoS
requirement, the service provider that has higher price-performance index has lower minimum price, i.e., \( \hat{\theta}_j(\theta_i,j,q_i) / \theta_{i,j} \leq 0 \).

If the service requester chooses a provider’s service, the utility of the service is the difference between the minimum price (presumably the service cost) and the service price that the service requester pays. The utility of a service \( s_{i,j} \) is:

\[
u_{i,j}(p) = (p_{i,j} \cdot c_i(\theta_{i,j}, q_i)) \cdot 1_{X(p)}(s_{i,j})
\]

(2)

where \( 1_{X(p)}(s_{i,j}) \) is the indicator function: \( 1_{X(p)}(s_{i,j}) = 1 \) if \( s_{i,j} \in X(p) \), i.e., the service requester chooses the service \( s_{i,j} \) for performing the task \( t_i \); \( 1_{X(p)}(s_{i,j}) = 0 \) if \( s_{i,j} \not\in X(p) \), i.e., the service requester does not choose the service.

The objective of each service provider is to set price for his service that can maximize his utility. Naturally, \( p_{i,j} \geq c_i(\theta_{i,j}, q_i) \), so that the utility of the service is non-negative.

### 2.1.1 Complete Information Game

If the service providers’ minimum prices and the service requester’s valuation of the tasks are public information, there is a complete information game for QoS-aware service selection.

For a task \( t_i \in T \), the price-performance indexes of the service providers are \( \{\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,m}\} \). We sort the price-performance indexes from high value to low value, and then obtain an ordered vector as \( \{\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,m}\} \), where \( \theta_{i,1} \) represents the highest price-performance index, \( \theta_{i,2} \) represents the second-highest price-performance index and \( \theta_{i,m} \) represents the lowest price-performance index. A service provider that has higher price-performance index as lower minimum price. Accordingly, \( c_i(\theta_{i,1}, q_i) \) represents the lowest price and \( c_i(\theta_{i,2}, q_i) \) represents the second-lowest price for completing the task \( t_i \).

Since the service providers’ minimum prices are public knowledge, given a set of local QoS requirements as \( q_1, \ldots, q_n \), the service requester can create a price matrix \( p \) as: for each service \( \forall s_{i,j} \in S_i \), if choosing the service to perform the task \( t_i \), the service requester sets the price as:

\[
p_{i,j} = \begin{cases}  
c_i(\theta_{i,1}, q_i) - \varepsilon, & \theta_{i,j} = \theta_{i,1} 
c_i(\theta_{i,j}, q_i), & \theta_{i,j} \not= \theta_{i,1}, \theta_{i,j} \geq \theta_{i,2} 
\end{cases}
\]

(3)

The price matrix means that if the service has the lowest of the minimum prices for completing a task, the service requester sets the price at being slightly less than the second lowest of the minimum prices and if not, the service requester sets the price at the minimum price of the service.

**Theorem 1** Given a set of local QoS requirements \( \{q_1, \ldots, q_n\} \), if \( \sum_{t_i \in T} c_i(\theta_{i,2}, q_i) \leq Budget \), the price matrix \( p \) is a (pure) \( \varepsilon \)-Nash equilibrium.

**Proof:** A (pure) \( \varepsilon \)-Nash equilibrium is a price matrix where no service provider can increase his utility by more than \( \varepsilon \) by changing his price.

According to the decision map (1), the service requester would choose a service that charges the lowest price for each task while the total price does not exceed the budget. In the price matrix \( p \), the lowest price for each task \( \forall t_i \in T \) is \( c_i(\theta_{i,2}, q_i) \). When facing the price matrix \( p \), if \( \sum_{t_i \in T} c_i(\theta_{i,2}, q_i) \leq Budget \), according to decision map (1) the service requester would choose a service that charges for performing \( \sum_{t_i \in T} c_i(\theta_{i,2}, q_i) - \varepsilon \) for performing each task \( \forall t_i \in T \).

Consider a Provider \( a_j \) that provides Service \( s_{i,j} \) for performing Task \( t_i \).

1) If \( \theta_{i,j} = \theta_{i,1} \), then \( p_{i,j} = c_i(\theta_{i,2}, q_i) - \varepsilon \). According to the decision map (1), there is \( s_{i,j} \in X(p) \). According to the utility function (2), the provider will obtain a positive utility, i.e., \( u_{i,j}(p) \geq 0 \). By decreasing his price, he can only decrease his utility. In order to increase his utility by more than \( \varepsilon \), the service provider needs to deviate to a price \( p'_{i,j} > p_{i,j} + \varepsilon \). However, the second lowest price will be less than the price, i.e., \( c_i(\theta_{i,2}, q_i) < p'_{i,j} \). Then for the new price matrix \( p' \) (by substituting \( p'_{i,j} \) for \( p_{i,j} \) in \( p \)), \( s_{i,j} \not\in X(p') \). So, \( u_{i,j}(p') = 0 \). Hence, he can only decrease his utility by more than \( \varepsilon \) by increasing his price.

2) If \( \theta_{i,j} \geq \theta_{i,2} \), then \( p_{i,j} = c_i(\theta_{i,j}, q_i) \). According to the decision map (1), there is \( s_{i,j} \not\in X(p) \). According to the utility function (2), the utility of the provider is zero, i.e., \( u_{i,j}(p) = 0 \). By changing his price, he can only decrease his utility.

Therefore, the price matrix \( p \) is a (pure) \( \varepsilon \)-Nash equilibrium.

We propose an approach for the service requester in the complete information game to maximize his utility as below:

First, according to the price matrix \( p \), the service requester would choose a service with the lowest price \( c_i(\theta_{i,2}, q_i) - \varepsilon \) for each task \( \forall t_i \in T \). The utility of the
service requester is \( \sum_{i \in T} (o_i(q_i) - c_i(\theta_{i,(2)}, q_i)) \) and the total service price is \( \sum_{i \in T} c_i(\theta_{i,(2)}, q_i) \) (where \( q_i \) represents a variable of the local QoS requirement for task \( t_i \)). We find that the local QoS requirements have a relationship with the utility of the service requester and the total service price. Hence, taking the limit as \( \varepsilon \to 0 \), the service requester determines the local QoS requirements that can maximize the utility of the service requester, while satisfying the global QoS requirement and the budget restriction.

\[
q^*_1, ..., q^*_n = \arg \max_{q_1, ..., q_n} \sum_{i \in T} (o_i(q_i) - c_i(\theta_{i,(2)}, q_i)) \quad (4)
\]

subject to \( \sum_{i \in T} c_i(\theta_{i,(2)}, q_i) \leq \text{Budget} \)

Solving the constrained optimization problem generates a set of local QoS requirements \( \{q^*_1, ..., q^*_n\} \). The total price is \( \sum_{i \in T} c_i(\theta_{i,(2)}, q^*_i) \), that does not exceed the budget \( \text{Budget} \).

Second, according to the local QoS requirements \( \{q^*_1, ..., q^*_n\} \), the service requester can compute the price matrix \( p \) by substituting \( q^*_i \) for \( q_i \). Since \( \sum_{i \in T} c_i(\theta_{i,(2)}, q_i) \leq \text{Budget} \), according to Theorem 1, the price matrix is a (pure) \( \varepsilon \)-Nash equilibrium. This means that no service provider can increase his utility by more than \( \varepsilon \) by changing the price that the service provider offers. Thus, no service provider is willing to change the price matrix.

According to the price matrix, the service requester chooses the service whose provider has the highest price-performance index, \( \theta_{i,(1)} \), for each task \( \forall t_i \in T \) as:

\[
X(p) = \{s_{i,j} \mid \forall t_i \in T, \theta_{i,j} = \theta_{i,(1)}\}
\]

Taking the limit as \( \varepsilon \to 0 \), the service requester pays \( c_i(\theta_{i,(2)}, q^*_i) \) for each selected service \( \forall s_{i,j} \in X(p) \). The utility of the service requester is \( \sum_{i \in T} (o_i(q^*_i) - c_i(\theta_{i,(2)}, q^*_i)) \), the total price is \( \sum_{i \in T} c_i(\theta_{i,(2)}, q^*_i) \leq \text{Budget} \), and the sum of the utilities of the service providers is \( \sum_{i \in T} c_i(\theta_{i,(2)}, q^*_i) - c_i(\theta_{i,(1)}, q^*_i) \).

For example, Table 1 shows the service requester’s valuation of the tasks and the service providers’ price-performance indexes and minimum prices.

<table>
<thead>
<tr>
<th>( T )</th>
<th>QoS</th>
<th>Valuation</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>(0.25,4.5)</td>
<td>310.9</td>
<td>251.3</td>
<td>281.4</td>
<td>269.7</td>
</tr>
<tr>
<td></td>
<td>(0.5,4.0)</td>
<td>293.4</td>
<td>213.5</td>
<td>253.2</td>
<td>234.6</td>
</tr>
<tr>
<td></td>
<td>(0.75,3.5)</td>
<td>249.3</td>
<td>199.3</td>
<td>219.9</td>
<td>208.2</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>(0.3,4.5)</td>
<td>274.6</td>
<td>231.8</td>
<td>( \theta_{2}=null )</td>
<td>211.6</td>
</tr>
<tr>
<td></td>
<td>(0.5,4.0)</td>
<td>219.7</td>
<td>183.1</td>
<td>( \theta_{2}=null )</td>
<td>170.9</td>
</tr>
<tr>
<td></td>
<td>(0.75,3.5)</td>
<td>164.8</td>
<td>135.4</td>
<td>( \theta_{2}=null )</td>
<td>118.3</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>(0.75,6.0)</td>
<td>237.2</td>
<td>261.4</td>
<td>189.3</td>
<td>249.6</td>
</tr>
<tr>
<td></td>
<td>(1.0,5.0)</td>
<td>204.6</td>
<td>189.3</td>
<td>148.7</td>
<td>170.1</td>
</tr>
<tr>
<td></td>
<td>(1.25,4.0)</td>
<td>166.1</td>
<td>157.8</td>
<td>127.0</td>
<td>144.2</td>
</tr>
</tbody>
</table>

As shown in the table, the price-performance index of Provider \( \alpha_2 \) for Task \( t_2 \) is \( \theta_{2,3} = 11.41 \). If satisfying the QoS requirement \((0.3, 4.5)\), Provider \( \alpha_3 \)’s minimum price is \( c_2(11.41, (0.3, 4.5)) = 211.6 \) and the service requester’s valuation of Task \( t_2 \) is \( o_2((0.3, 4.5)) = 274.6 \). There are two Providers \( \alpha_1 \) and \( \alpha_2 \), whose price-performance indexes are \( \theta_{2,1} = 9.65 \) and \( \theta_{2,3} = 11.41 \), for Task \( t_2 \). We have \( \theta_{2,1} = \theta_{2,3} = 11.41 \) and \( \theta_{2,2} = 9.65 \). Thus, the service requester sets the service price of Provider \( \alpha_3 \) as \( (0.23, 9.65, q_2) \) and sets the service price of Provider \( \alpha_1 \) as \( (0.23, 9.65, q_2) \) (\( q_2 \) is a local QoS requirement for Task \( t_2 \)).

Suppose that the global QoS requirement is \( q_G = (2.5, 3.5) \) and the budget is 620.0. Solving the constrained optimization problem (4) generates the local QoS requirements for the tasks \( t_1, t_2 \) and \( t_3 \) as \( q^*_1 = (0.5, 4.0), q^*_2 = (0.3, 4.5) \) and \( q^*_3 = (1.0, 5.0) \) that can maximize the utility of the service requester, while satisfying the global QoS requirement and the budget restriction.

Third, according to the local QoS requirements and the equation (3), the service requester create a price matrix \( p \) as (let \( \varepsilon = 0.01 \)):

\[
p = \begin{pmatrix}
234.59 & 211.59 & 189.3 \\
253.2 & +\infty & 170.09 \\
234.6 & 211.6 & 170.1 
\end{pmatrix}
\]

The price matrix is (pure) \( \varepsilon \)-Nash equilibrium. According to the price matrix, Provider \( \alpha_3 \) whose service price for Task \( t_2 \) is 211.59 would obtain a positive utility as 20.2. In order to increase his utility by more than 0.01, the provider needs to increase the service price to 211.61. However, the service requester would choose Provider \( \alpha_3 \)’s service whose price is 211.6, instead of Provider \( \alpha_3 \)’s service. Accordingly, Provider \( \alpha_3 \)’s utility would decrease.
to zero. Thus, Provider $\alpha_3$ would accept the price that the service requester offers. The service requester then chooses Provider $\alpha_3$ to provide his service $s_{2,3}$ for performing Task $t_2$ and pays 211.59 to Provider $\alpha_3$.

Finally, the service requester obtains an outcome as $X(p) = \{s_{1,1}, s_{2,3}, s_{3,2}\}$. The overall QoS is $(1.8, 4.0) \geq (2.5, 3.5)$ and the total service price is 616.27 < 620.0.

### 2.2.2 Incomplete Information Game

Naturally, each service provider would not tell his minimum price to the service requester and other service providers. Hence, there is an incomplete information game for QoS-aware service selection, thereby suggesting the need for predicting the minimum service prices of service providers.

Wang and Du [10] proposed an iterative approach based on the predictions of the minimum prices of the service providers. Figure 1 describes their overall approach.

For example, there are three service providers $A = \{\alpha_1, \alpha_2, \alpha_3\}$ that are able to provide services for performing the tasks $T = \{t_1, t_2, t_3\}$. The game process is described as a decision tree in Figure 2.

In Round 1, the service requester knows that the price-performance indexes of all service providers are in a particular distribution over $(0, +\infty)$. Using the distribution, he predicts the highest price-performance index for Task $t_1$ as 16.37, for Task $t_2$ as 15.41, and for Task $t_3$ as 10.18, respectively. For Task $t_3$, based on the prediction of the highest price-performance index, the service requester decides the local QoS requirement as $q_t = (1.25, 4.0)$ (the response time is less than or equal to 1.25 second and the throughput is greater than or equal to 4.0 kbps) and estimates the minimum price as 164.7. And then, the service requester offers a contract, $\omega_1$, for Task $t_3$, where the local QoS requirement is $(1.0, 4.5)$ and the service price is 164.7. Similarly, the service requester offers Contracts $\omega_1$ and $\omega_2$ for Tasks $t_1$ and $t_2$, respectively.

Given each contract, suppose that there exists a service provider whose minimum price is less than or equal to the service price, the service provider would have non-negative utility if he accepts the contract and his utility would be zero if he rejects the contract. The service provider decide whether to accept or reject the contract. As shown in the leftmost branch, the service requester and Provider $\alpha_4$ reach the contract $\omega_3$ for Task $t_3$, denoted by $(\alpha_4, \omega_3)$. But, Contracts $\omega_1$ and $\omega_2$ are not accepted by any service provider. It can be inferred that the service prices of the service providers are greater than the requester’s predictions of the service prices for Tasks $t_1$ and $t_2$. Since a lower price-performance index means a higher price, we infer the actual price-performance indexes of the service providers are less than the predicted indexes. Thus, the service requester knows the price-performance indexes of the
service providers for Task $t_1$ are within (0, 16.37) and the price-performance indexes of the service providers for Task $t_2$ are within (0, 15.41).

In Round 2, consider the service requester predicts the highest price-performance index for Task $t_1$ as 13.10, i.e., within (0, 16.37) and for Task $t_2$ as 10.52, i.e., within (0, 15.41). Based on the predictions, it offers new contracts $o_1$ and $o_2$ for Tasks $t_1$ and $t_2$ based on the predictions. As shown in rightmost branch, the service requester and Providers $a_1$ and $a_3$ reach contracts for Tasks $t_1$ and $t_2$, respectively, i.e., $(a_1, o_1)$ and $(a_3, o_2)$. Thus, the game ends. If contracts still exist that are not accepted yet, the process is iterated till the requester and the providers reach contracts for all tasks.

We find that it is very important to predict the minimum prices of the service providers in the price competition with incomplete information. Wang and Du's approach assumes that the service requesters know that the service providers' price-performance indexes follow a particular kind of distribution and the price functions of the tasks. However, they do not discuss how to acquire the knowledge.

In Section 3, we propose a collaborative approach to predicting the service providers' minimum prices by using the past usage experiences. Moreover, Wang and Du's approach assumes the response strategy of the service providers is myopic. That means that when offered a service price greater than or equal to his minimum price, a service provider would accept the contract and would neglect that he may obtain higher utility in next round if no service provider accepts the contract in this round. In Section 4, based on the predictions of the minimum prices of the service providers, we propose an multi-round Vickrey mechanism where the strategy of the service providers is not myopic.

### 3. A Collaborative Prediction Approach

In this section, we propose a prediction approach by using the past usage experiences of the service requesters about the service providers. We assume that the service requesters will provide their past usage experiences about service providers and tasks. Table 2 shows examples of usage experience records, including response time, and throughput as well as predictions of price-performance indexes and minimum prices of service providers. For example, in the first row of the table with timestamp 201408 (August 2014), Requester $r_1$ created a contract with Provider $a_3$ for Task $t_3$. The QoS offered by Provider $a_3$ is represented as the response time being less than or equal to 0.3 seconds and throughput greater than or equal to 2.0 kbps. The price-performance index and the minimum price of Provider $a_3$ were predicted by Requester $r_1$ as 13.41 and 192.49, respectively.

To predict a provider's minimum price for fulfilling a task, first, we predict the provider's price-performance index, and second, we estimate a function to describe the relationship between the price-performance index, QoS requirement, and the minimum price.

#### 3.1 Predicting Provider's Price-Performance Index

Given Provider $a_i \in A$ and Task $t_i \in T$, we need to predict the price-performance index $\theta_{i,j}$ of Provider $a_j$ for fulfilling Task $t_i$.  

**3.1.1 Direct Usage Experience**

Based on the past usage experiences shown in Table 2, we create a three-dimensional matrix $R = l \times n \times m$, where $l$ represents the number of requesters, $n$ represents the number of tasks, and $m$ represents the number of providers. Each entry $R_{i,j,k}$ ($k \in [I,I]$, $i \in [I,n]$, $j \in [I,m]$) in the matrix represents a prediction of Provider $a_j$'s index $\theta_{i,j}$ for fulfilling Task $t_i$ by Requester $r_k$. Consider an example in Figure 3, where Requester $r_1$ predicted the index $\theta_{3,3}$ of Provider $a_3$ for fulfilling Task $t_3$ as 13.41. We represent this observation from the matrix as $R_{1,3,3} = 13.41$. $R_{1,4,3} = null$ means Provider $a_3$ did not fulfill Task $t_4$ or Requester $r_1$ has no experience about the index $\theta_{4,3}$ of Provider $a_3$ for fulfilling Task $t_4$.  

---

**Table 2. Examples of usage experience records**

<table>
<thead>
<tr>
<th>Requester</th>
<th>Task</th>
<th>Provider</th>
<th>Time Stamp</th>
<th>Response Time (second)</th>
<th>Throughput (kbps)</th>
<th>Prediction of Index</th>
<th>Prediction of Minimum Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>$t_3$</td>
<td>$a_3$</td>
<td>201408</td>
<td>0.3</td>
<td>2.0</td>
<td>13.41</td>
<td>192.49</td>
</tr>
<tr>
<td>$r_3$</td>
<td>$t_3$</td>
<td>$a_3$</td>
<td>201412</td>
<td>0.5</td>
<td>4.5</td>
<td>10.17</td>
<td>257.40</td>
</tr>
<tr>
<td>$r_1$</td>
<td>$t_4$</td>
<td>$a_2$</td>
<td>201408</td>
<td>0.35</td>
<td>3.3</td>
<td>12.85</td>
<td>226.35</td>
</tr>
</tbody>
</table>
We define \( E(\theta_{i,j}) \) containing several predictions \( R_{i,j,k} \) from different requesters about the provider.

A provider’s price-performance index could change over time. We posit that in a long period of time the price-performance index can change greatly, but would not change much in a short period of time. We set a month as the duration over which a significant change must occur. The index predictions can be made at different times. To record the time of such predictions, each non-empty entry \( R_{i,j,k} \) in the matrix \( R \) has a timestamp shown in Table 2.

For example, \( R_{3,3,3} \) was predicted on August 2014. It means that \( R_{3,3,3} \) was predicted on August 2014. We posit that if a prediction \( R_{3,3,3} \) was made more than one month ago, there may be a deviation between the prediction and the possible current index.

For a longer timeline, we divide the timeline into a set of discrete time intervals considering a month as a unit time. For example, suppose the current date is February 2015. As shown in Figure 4, we divide the timeline from August 2014 till January 2015 into six discrete time intervals. We represent the discrete time intervals as \( \Psi = (\psi_1, \ldots, \psi_c) \), where \( \psi_1 \) is the latest time interval and \( \psi_c \) is the oldest time interval. We mark each non-empty entry \( R_{i,j,k} \) with a time interval \( \psi_c \) denoted by \( \psi(R_{i,j,k}) = \psi_c \in \Psi_c \).

To handle the change of a price-performance index over time, we posit that an index is closer to a recent prediction than to a prior prediction. Therefore, we associate a higher weight to a recent prediction than to an old prediction. To calculate the weight \( \omega(R_{i,j,k}) \) of a prediction \( R_{i,j,k} \) of the index \( \theta_{i,j} \), we apply a decay model as given below.

\[
\omega(R_{k,i,j}) = \begin{cases} 1.0, & x = 1 \\ exp(1 - 1.5 \times ln(x)), & x \geq 2 \end{cases}
\]

In the above, \( x \) denotes the index of a time interval \( \psi_c \) of a prediction \( R_{i,j,k} \). According to the decay model, more recent is the time interval of a prediction, more is the weight associated with the prediction.

![Figure 3. The requester-task-provider matrix with predictions](image)

![Figure 4. An example of direct usage experience about Task \( t_3 \) and Provider \( \alpha_3 \)](image)

For example, in Figure 4, suppose current time is February 2015 and a prediction \( R_{3,3,3} \) was made in the month of November 2014, i.e., three months ago. We denote the time interval for \( R_{3,3,3} \) as \( \psi_3 \) and obtain the weight of the time interval as 0.5231. Similarly, suppose another prediction \( R_{5,3,3} \) was made six months ago. We represent the time interval as \( \psi_6 \) and obtain the weight as 0.158. Thus, the weight of the prediction \( R_{5,3,3} \) is less than the prediction \( R_{3,3,3} \). As mentioned above, the price-performance index \( \theta_{3,3} \) is likely closer to the prediction \( R_{3,3,3} \) than the prediction \( R_{5,3,3} \).

**Predicting the price-performance index** Based on the direct usage experience \( E(\theta_{i,j}) = \{R_{i,j,k} \mid k \in [1,L]\} \) and their time-aware weights, we predict the price-performance index \( \tilde{\theta}_{i,j} \) by using a weighted mean equation as below.

\[
\tilde{\theta}_{i,j} = \frac{\sum_{k=1}^{L} \omega(R_{i,j,k})}{\sum_{k=1}^{L} \omega(R_{i,j,k})}
\]

Consider an example where we want to predict the price-performance index \( \tilde{\theta}_{3,3} \) of Provider \( \alpha_3 \) for fulfilling Task \( t_3 \) on February 2015. In Figure 4, we can find that Requesters \( r_5 \), \( r_9 \), \( r_7 \) and \( r_1 \) predicted index \( \theta_{3,3} \) for Provider \( \alpha_3 \) in different time intervals. We treat this information as a set of direct usage experiences. Specifically, \( E(\theta_{3,3}) = \{R_{5,3,3}, R_{9,3,3}, R_{7,3,3}, R_{3,3,3}\} = \{13.36, 20.54, 21.60, 13.41\} \) of Requesters \( r_5 \), \( r_9 \), \( r_7 \) and \( r_1 \), respectively. The direct usage experiences are marked by the time intervals...
Applying Equation (5), we obtain the weights of the time intervals as: $\omega(13.36) = 0.1580$, $\omega(20.54) = \omega(21.60) = 0.3398$, and $\omega(13.41) = 0.5321$, respectively. Applying Equation (6), we predict the index $\tilde{y}_{3,3}$ as 17.436.

### 3.1.2 Improving the Prediction Accuracy

The approach given above to predict a provider’s price-performance index has a limitation. First, if the requesters have no direct usage experiences about providers (e.g., Provider $a_3$ and Task $t_3$), we cannot use the approach to predict the service provider’s price-performance index. Second, the requesters might have direct experiences about Provider $a_3$ and Task $t_3$, but the recent prediction $R_{1,3,3}$ for Provider $a_3$ was made three months ago, as shown in Figure 4. This means there is a deviation of three months between the most recent prediction and the possible current index. Assume that in this period of three months there are no requesters that have direct usage experiences with Provider $a_3$. Thus, there might be a problem in accurately predicting the provider's current index on the basis of the prediction made three months ago since the index can change over time. To handle the limitations, we look for providers similar to Provider $a_3$ that fulfilled Task $t_3$. We check if there are requesters present in matrix $R$ in Figure 4 that have direct usage experiences with these providers for Task $t_j$ in the recent three months. Using their direct usage experiences, we can predict Provider $a_3$'s index more precisely.

In this paper, we say that two service providers are similar if they have probably have similar price-performance indexes. We use Pearson correlation coefficient (PCC) [13] to compute the similarity between two candidate providers $a_x$ and $a_y$ based on the past usage experiences with the providers. We represent their similarity $sim(a_x, a_y)$ as below.

$$sim(a_x, a_y) = \frac{\sum_{k \in K, i \in I}(R_{k,i,x} - \bar{R}_x)(R_{k,i,y} - \bar{R}_y)}{\sqrt{\sum_{k \in K, i \in I}(R_{k,i,x} - \bar{R}_x)^2} \sqrt{\sum_{k \in K, i \in I}(R_{k,i,y} - \bar{R}_y)^2}}$$ (7)

In the above equation, $I$ represents identifiers for a set of tasks fulfilled by Providers $a_x$ and $a_y$. $K$ represents identifiers for a set of requesters. The requesters in $K$ predicted the price-performance indexes of Providers $a_x$ and $a_y$ for fulfilling the tasks in $I$ in the same time intervals. $\bar{R}_x$ and $\bar{R}_y$ are the mean values of the predicted indexes of Providers $a_x$ and $a_y$ for fulfilling the tasks in $I$ by the requesters in $K$.

<table>
<thead>
<tr>
<th>time intervals</th>
<th>$\psi_0$</th>
<th>$\psi_1$</th>
<th>$\psi_2$</th>
<th>$\psi_3$</th>
<th>$\psi_4$</th>
<th>$\psi_5$</th>
<th>$\psi_6$</th>
<th>$\psi_7$</th>
<th>$\psi_8$</th>
<th>$\psi_9$</th>
<th>current:</th>
</tr>
</thead>
<tbody>
<tr>
<td>service provider $a_3$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
</tr>
<tr>
<td>service provider $a_3$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
<td>$R_{3,3,3}$</td>
</tr>
</tbody>
</table>

Figure 5. The similarity computation between any two providers

For example, suppose the current time is February 2015. We want to predict Provider $a_3$’s price-performance index for fulfilling Task $t_3$. As shown in Figure 5, the most recent prediction $R_{1,3,3}$ of Provider $a_3$’s index for fulfilling Task $t_3$ was made in time interval $\psi_3$, i.e., in November 2014. Since, the most recent index prediction was made three months ago, there might be a large deviation between the index predicted in November 2014 and the possible current index. Therefore, we look for providers similar to Provider $a_3$ with more recent index predictions than the index predicted in November 2014. Suppose another Provider $a_8$ has also fulfilled Task $t_3$. The most recent index prediction $R_{4,3,8}$ was made only a month ago. We found that Requesters $r_5, r_9, r_7$ and $r_1$ predicted the index for Providers $a_3$ and $a_8$ for Task $t_4$ within the time intervals $\psi_6, \psi_4$ and $\psi_3$. From the above, we have $I = \{t_4\}$ (Task $t_j$) and $K = \{5, 9, 7, 1\}$ (Requesters $r_5, r_9, r_7$ and $r_1$).

Then, we calculate the similarity between Providers $a_3$ and $a_8$ according to Equation (7).

For a provider, once its similar providers are found, we further compute a subset of the similar providers having more recent index predictions in their past direct usage experiences. The reason is that a price-performance index is likely closer to a recent prediction than an old prediction. For fulfilling Task $t_4$, a subset of similar providers, denoted by $\Gamma(a_j)$, of Provider $a_j$ is represented as below.

$$\Gamma(a_j) = \{a_\beta | sim(a_j, a_\beta) \geq v_{sim} \wedge a_\beta \neq a_j, \wedge \kappa(t_4, a_j) \}$$ (8)

In the above equation, $v_{sim} \geq 0$ is a threshold for the similarity calculation and $\kappa(t_4, a_j)$ is the time interval index of the most recent prediction about Provider $a_j$’s index for fulfilling the task $t_4$. If there is no direct usage experiences about Provider $a_j$ for Task $t_4$, we set $\kappa(t_4, a_j)$ to $+\infty$. 
Consider an example where Provider $a_3$ is similar to Provider $a_8$ so that $\text{sim}(a_3, a_8) \geq \epsilon_{\text{sim}}$. In other words, Providers $a_3$ and $a_8$ have similar price-performance indexes of fulfilling Task $t_1$. In Figure 5, the most recent prediction for Provider $a_3$’s index for fulfilling Task $t_3$ was made in the time interval $y_{t_3}$. Thus, we know $\kappa(t_3, a_3) = 3$. For Provider $a_8$, its most recent prediction $R_{4,3,8}$ was made in the time interval $y_{t_1}$, i.e., in January 2015. Thus, we have the time interval index of the most recent prediction $R_{4,3,8}$ as 1, i.e., $\kappa(t_3, a_8) = 1$. Since, $\kappa(t_3, a_8) < \kappa(t_3, a_3)$, it indicates that there are more recent predictions about Provider $a_3$’s index for fulfilling Task $t_3$. According to Equation (8), Provider $a_8$ is added to the set $\Gamma(a_3)$.

Using the predictions of similar service providers’ indexes, we predict Provider $a_j$’s index $\hat{\theta}_{i,j}$ for Task $t_i$ as given below.

$$
\hat{\theta}_{i,j} = \frac{\sum_{a_j \in \Gamma(a_j)} \text{sim}(a_j, a_y) \times \hat{\theta}_{i,y}}{\sum_{a_j \in \Gamma(a_j)} \text{sim}(a_j, a_y)} \quad (9)
$$

In the above equation, $\Gamma(a_j)$ refers to a set of similar providers with more recent predictions than Provider $a_j$ and $\text{sim}(a_j, a_y)$ is the similarity score between the Providers $a_j$ and $a_y$. Based on the approach given above, we can obtain a prediction $\hat{\theta}_{i,y}$ of a similar provider $a_y$’s index $\theta_{i,y}$ for fulfilling Task $t_i$ using the Equation (6) based on their past direct usage experiences.

Finally, we have the Equations (6) and (9) to predict the index $\hat{\theta}_{i,j}$ of a candidate provider $a_j \in A$ for fulfilling Task $t_i \in T$ given below.

$$
\hat{\theta}_{i,j} = \begin{cases} 
\mu \cdot \hat{\theta}_{i,j} + (1 - \mu) \cdot \hat{\theta}_{i,j}, & E(\hat{\theta}_{i,j}) \neq \phi, \Gamma(a_j) \neq \phi \\
\hat{\theta}_{i,j}, & E(\hat{\theta}_{i,j}) \neq \phi, \Gamma(a_j) = \phi \\
\hat{\theta}_{i,j}, & E(\hat{\theta}_{i,j}) = \phi, \Gamma(a_j) \neq \phi
\end{cases} \quad (10)
$$

In the above equation, the parameter $\mu \in [0,1]$ determines a proportion, $E(\hat{\theta}_{i,j})$ represents the direct usage experience about the price-performance index $\theta_{i,j}$, and $\Gamma(a_j)$ represents the set of providers similar to Provider $a_j$. There are more recent predictions about $\Gamma(a_j)$ than the predictions about Provider $a_j$.

### 3.1 Estimating Price Function

A service provider can adjust its QoS to meet the QoS requirement of a task. In this section, we estimate the function of a service provider, which describes the relationship between the QoS requirement, price-performance index, and the minimum price. For a task $t$ and a candidate service provider $\alpha$, we estimate the price function as $c(\beta, q)$, where $q$ is the QoS requirement and $\beta$ is the price-performance index of Provider $\alpha$ for fulfilling Task $t$. We choose a general non-linear parametric function as below.

$$
c(\beta, q) = \frac{a \times \beta^b}{\beta^d} \quad (11)
$$

From the usage experience records shown in Table 2, we further obtain a subset of usage experience records including the response time, throughput, price-performance index, and the minimum price of a service provider. For example, the usage experiences about Provider $a_3$ and Task $t_3$ are listed in Table 3. In the first row of the table, for fulfilling Task $t_3$, while the response time less than or equal to 0.3 second and the throughput greater than or equal to 2.0 kbps, the minimum price of Provider $a_3$, whose index is 13.41, was predicted as 192.49.

Table 3. An example of usage experience records about Provider $a_3$ and Task $t_3$.

<table>
<thead>
<tr>
<th>Response Time</th>
<th>Throughput</th>
<th>Prediction of Index</th>
<th>Prediction of Minimum Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>2.0</td>
<td>13.41</td>
<td>192.49</td>
</tr>
<tr>
<td>0.5</td>
<td>4.5</td>
<td>10.17</td>
<td>257.40</td>
</tr>
<tr>
<td>0.25</td>
<td>3.2</td>
<td>12.97</td>
<td>236.87</td>
</tr>
</tbody>
</table>

We estimate a provider’s minimum service price on the basis of the provider’s QoS and price-performance index. Using the historical data about the QoS, the price-performance index, and the minimum service price, we can estimate a relationship between the three. For the estimation, we use the non-linear regression algorithm [15] provided by the Matlab toolkit. Using the algorithm, we compute the parameters $a$, $b$ and $d$ of the Equation (11) based on the historical records given above. Using the parameters, we predict the price function of a service provider for fulfilling a task.

In summary, for a task $t_i \in T$, first, we predict a service provider’s price-performance index as $\hat{\theta}_{i,j}$ and estimate its price function. Second, we estimate the provider’s minimum price $c_i(\hat{\theta}_{i,j}, q_i)$ based on the local QoS requirement $q_i$, the prediction of the price-performance index and the price function.

### 4. Multi-Round Vickrey Auction

For a task $t_i \in T$, we predict its price function as $c_i(\cdot, \cdot)$ and predict the price-performance indexes of the service
providers as \( \{\tilde{\theta}_{i,1}, \ldots, \tilde{\theta}_{i,m}\} \) by using the collaborative prediction approach. We sort the predicted price-performance indexes from high value to low value, and then obtain an ordered vector as \( (\tilde{\theta}_{i,1}, \tilde{\theta}_{i,2}, \ldots, \tilde{\theta}_{i,m}) \), where \( \tilde{\theta}_{i,1} \) represents the highest predicted price-performance index, \( \tilde{\theta}_{i,2} \) represents the second-highest predicted price-performance index and \( \tilde{\theta}_{i,m} \) represents the lowest predicted price-performance index.

We assume that the service requesters will provide their usage experiences about the service providers and the tasks. Accordingly, each service provider can predict the price-performance indexes of other service providers by using the collaborative prediction approach. Thus, we assume that the predictions of the price-performance indexes are public knowledge for the service providers.

As discussed in Section 2.2.1, for each task \( \forall t_i \in T \), the service requester would choose the service whose price is predicted to be the lowest price, \( c_i(\tilde{\theta}_{i,1}, q_i) \), and the price that the service requester pays for the service would be slightly less than the predicted second-lowest price, \( c_i(\tilde{\theta}_{i,2}, q_i) + \epsilon \). However, in general, there may be a deviation between the predicted and the actual value. The deviation may cause the chosen service provider not to accept the price that the service requester offers. For example, for a task \( t_2 \), the price-performance indexes of the service providers \( a_1 \) and \( a_2 \) are \( \tilde{\theta}_{2,1} = 9.65 \) and \( \tilde{\theta}_{2,2} = 11.41 \), respectively. Suppose that the predictions of their price-performance indexes are \( \tilde{\theta}_{2,1} = 12.11 \) and \( \tilde{\theta}_{2,3} = 11.08 \), respectively. In the case, the service requester would choose Provider \( a_1 \) and would pay \( c_2(11.08, q_2) + \epsilon \). However, whatever the local QoS requirement \( q_2 \) is, the value of \( c_2(11.08, q_2) + \epsilon \) would be less than the actual minimum price \( c_2(9.65, q_2) \) of Provider \( a_1 \). Thus, Provider \( a_3 \) would not accept the price. The service requester needs a mechanism for verifying the predictions and obtaining an outcome based on the verified predictions.

Based on the predictions of the service providers’ price-performance indexes and the price functions, we propose a mechanism called multi-round Vickrey auction as below: given a global QoS requirement \( q_G \) and a budget \( \text{Budget} \).

In round \( k \), the tasks \( T \) could be divided into two sets: the one is \( T^a \). For each task \( \forall t_i \in T^a \), the service requester has chosen exactly one provider’s service \( s_{i,j} \) for performing the task in the previous rounds. The local QoS requirement is denoted by \( q_i \) and the price is denoted by \( p_{i,j} \); the other one is \( T^u \). For each task \( \forall t_i \in T^u \), no service provider bids yet (\( T^u \cap T^a = \emptyset \)) and \( T^u \cup T^a = T \).

In round 1, \( T^u = T \) and \( T^a = \emptyset \).

First, for each task \( \forall t_i \in T^u \), the service requester sets the reserve price equal to the prediction of the \((k+1)\)-th lowest minimum price, denoted by \( \lambda_i = c_i(\tilde{\theta}_{i,(k+1)}, q_i) \). The reserve price is the highest price that the service requester is willing to pay in this round.

Second, the service requester decides the local QoS requirements for the tasks \( T^u \) as below:

\[
\begin{align*}
\max & \sum_{t_i \in T^u} (q_i(q_i) - c_i(\tilde{\theta}_{i,(k+1)}, q_i)) \\
\text{subject to} & \forall s_{i,j} \in T^u \implies p_{i,j} \geq q_i
\end{align*}
\]

\[
\sum_{t_i \in T^u} p_{i,j} + \sum_{t_i \in T^u} c_i(\tilde{\theta}_{i,(k+1)}, q_i) \leq \text{Budget}
\]

Solving the problem generates a local QoS requirement \( q_i^* \) for each task \( \forall t_i \in T^u \). Accordingly, the service requester sets a reserve price at \( \lambda_i^* = c_i(\tilde{\theta}_{i,(k+1)}, q_i^*) \) for each task \( \forall t_i \in T^u \).

Third, for each task \( \forall t_i \in T^u \), the service requester chooses a provider’s service that can satisfy the local QoS requirement \( q_i^* \) by using a Vickrey auction with the reserve price \( \lambda_i^* = c_i(\tilde{\theta}_{i,(k+1)}, q_i^*) \). In a Vickrey auction with reserve price, each service provider bids (below the reserve price) or keeps silent without knowing the bids of the other service providers. For a task \( t_i \in T^u \), if receiving bids, the service requester chooses the provider with the lowest bid but the price, called Vickrey price, is the second-lowest bid, and moves the task from the set \( T^u \) into the set \( T^a \). If only one service provider bids, the price is the reserve price.

If there exist some tasks for which no service provider submits bid in this round, i.e., \( T^u = \emptyset \), the service requester goes to round \( k+1 \).

In round \( k+1 \), the service requester revises the local QoS requirement \( q_i \) and the reserve price \( \lambda_i = c_i(\tilde{\theta}_{i,(k+2)}, q_i) \) for each task \( \forall t_i \in T^u \) while satisfying the global QoS requirement and the budget restriction. And then, the service requester chooses a provider’s service for each task \( \forall t_i \in T^u \) using a Vickrey auction with the reserve price \( \lambda_i \) again.

If \( T^u = \emptyset \), the game process ends and the service requester obtains an outcome. After the auction ends, if the service requester has chosen a service for each task, his requested service can be implemented and he pays the specified service price to each chosen service; Otherwise, the service requester does not need to pay but he fails to implement his requested service.
Theorem 2 In round \( k \) of a multi-round Vickrey auction, for a task \( t_i \in T^a \), according to the prediction results \((\hat{\theta}_{i,1}, \hat{\theta}_{i,2}, ..., \hat{\theta}_{i,m})\), it is a dominant strategy for a service provider to submit his minimum price if and only if his actual price-performance index is at least \( \hat{\theta}_{i,(k+1)} \).

Proof: In round \( k \), for the task \( t_i \in T^a \), the reserve price is \( c_i(\hat{\theta}_{i,(k+1)}, q_i) \). If the service provider's actual price-performance index is less than \( \hat{\theta}_{i,(k+1)} \), his minimum price will be greater than the reserve price \( c_i(\hat{\theta}_{i,(k+1)}, q_i) \). Thus, the service provider will not bid (below the reserve price).

If the service provider's actual price-performance index is at least \( \hat{\theta}_{i,(k+1)} \), his minimum price will be less than or equal to the reserve price. According to the predictions of the price-performance indexes, the service provider knows there may exist at least another service provider whose predicted price-performance index is greater than or equal to \( \hat{\theta}_{i,(k+1)} \). In the multi-round Vickrey auction, when the other service providers decide to bid in this round, there is no chance for the service provider to win in the next round. Thus, the service provider will decide to bid in this round. In a Vickrey auction, truthful bidding is a dominant strategy for bidders. Therefore, the service provider will submit his minimum price.

For a task \( t_i \in T^a \), suppose that there are \( m \) predictions of the service providers' price-performance indexes. If no service providers bid for the task till round \( m \), according to Theorem 2, the actual price-performance indexes of the service providers all are less than the lowest predicted price-performance index \( \hat{\theta}_{i,(m)} \). Hence, there will be no other predictions in the next round. For example, for Task \( t_2 \), suppose that the predictions of the price-performance indexes of Providers \( \alpha_1 \) and \( \alpha_2 \) are 13.71 and 12.49, respectively. As shown in Table 1, the actual price-performance indexes of Providers \( \alpha_1 \) and \( \alpha_3 \) are less than the predictions. In round 1, no service provider bids for the task. In the case, the service requester has two choices: first, the game ends. The service requester fails to realize his service request. Second, the service requester looks for more usage experiences about the service providers and predicts the service providers' price-performance indexes. And then, the game continues.

Consider the example where the global QoS requirement is \((2.5, 3.5)\) and the budget is 620.0. Suppose that the predictions of the price-performance indexes of Providers \( \alpha_1, \alpha_2 \) and \( \alpha_3 \) for Tasks \( t_1, t_2 \) and \( t_3 \) are given as below:

\[
\begin{pmatrix}
\hat{\theta}_{1,1} & \hat{\theta}_{1,2} & \hat{\theta}_{1,3} \\
\hat{\theta}_{2,1} & \hat{\theta}_{2,2} & \hat{\theta}_{2,3} \\
\hat{\theta}_{3,1} & \hat{\theta}_{3,2} & \hat{\theta}_{3,3}
\end{pmatrix} = 
\begin{pmatrix}
15.31 & 12.11 & 12.61 \\
14.84 & 13.73 & 14.31
\end{pmatrix}
\]

(12)

<table>
<thead>
<tr>
<th>Requirement and</th>
<th>Local QoS</th>
<th>Winner</th>
<th>Service Providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>T^a</td>
<td>Winner</td>
<td>Service Providers</td>
<td></td>
</tr>
<tr>
<td>( t_1 )</td>
<td>( q_1 = (0.5, 4.0) )</td>
<td>( \lambda_1 = 201.9 )</td>
<td>( \alpha_1, \alpha_2, \alpha_3 )</td>
</tr>
<tr>
<td>Round 1</td>
<td>( q_2 = (0.3, 4.5) )</td>
<td>( \lambda_2 = 229.3 )</td>
<td>( \alpha_3 )</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>( q_3 = (1.0, 5.0) )</td>
<td>( \lambda_3 = 145.9 )</td>
<td>( \text{null} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement and</th>
<th>Local QoS</th>
<th>Winner</th>
<th>Service Providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>T^a</td>
<td>Winner</td>
<td>Service Providers</td>
<td></td>
</tr>
<tr>
<td>( t_1 )</td>
<td>( q_1 = (0.75, 3.5) )</td>
<td>( \lambda_1 = 217.3 )</td>
<td>( \alpha_1 )</td>
</tr>
<tr>
<td>Round 2</td>
<td>( q_2 = (1.25, 4.0) )</td>
<td>( \lambda_2 = 135.2 )</td>
<td>( \alpha_2, \alpha_3 )</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>( q_3 = (0.5, 4.0) )</td>
<td>( \lambda_3 = 201.9 )</td>
<td>( \alpha_1, \alpha_2, \alpha_3 )</td>
</tr>
</tbody>
</table>

As shown in Figure 6, in round 1, \( T^a = \{t_1, t_2, t_3\} \) and \( T^a = \text{null} \). According to the prediction results (12), \( \hat{\theta}_{1,(2)} = 14.84 \). For Task \( t_1 \), the service requester decides the local QoS requirement as \( q_1 = (0.5, 4.0) \) and the reserve price as \( \lambda_1 = c_1(14.84,(0.5,4.0)) = 201.9 \). As shown in Table 1, the actual price-performance indexes of Providers \( \alpha_1, \alpha_2 \) and \( \alpha_3 \) are all less than 14.84. According to Theorem 2, no service provider bids for the task. According to the prediction results (12), \( \hat{\theta}_{2,(2)} = 13.73 \). For Task \( t_2 \), the service requester decides the local QoS requirement as \( q_2 = (1.0, 5.0) \) and the reserve price as \( \lambda_2 = c_2(13.73,(1.0,5.0)) = 145.9 \). Similarly, no service provider bids for Task \( t_3 \). According to the prediction results (12), \( \hat{\theta}_{3,(2)} = 11.08 \). For Task \( t_2 \), the service requester decides the local QoS requirement as \( q_2 = (0.3, 4.5) \) and the reserve price as \( \lambda_2 = c_2(11.08, (0.3,4.5)) = 229.3 \). As shown in Table 1, the actual price-performance indexes of Providers \( \alpha_1 \) and \( \alpha_3 \) are 9.65 and 11.41, respectively. According to Theorem 2, only Provider \( \alpha_3 \) bids. The service requester chooses Provider \( \alpha_3 \) and the pays the reserve price 229.3. The service requester moves Task \( t_2 \) from the set \( T^a \) to the set \( T^a \).

In round 2, \( T^a = \{t_1, t_3\} \) and \( T^a = \{t_2\} \). According to the prediction results (12), \( \hat{\theta}_{1,(2)} = 13.19 \). For Task \( t_1 \), the service requester revises the local QoS requirement as \( q_1 = (0.75, 3.5) \) and the reserve price as \( \lambda_1 = c_1(13.19,(0.75, 3.5)) = 201.9 \).
does not know other service providers' minimum prices as well price-performance indexes. We use Wang and Du's approach to predict the providers' price-performance indexes and estimate their minimum prices based on the contracts the service providers accept or refuse while assuming that the price functions and the distribution of the service providers' price-performance indexes are public knowledge.

In our experiments, we interpret the data generated by the Wang and Du's approach as the historical usage experiences. Accordingly, we obtain a $142 \times 50 \times 4532$ requester-task-provider matrix of price-performance indexes. The matrix is sparse. The number of entries comes to 485,597. The entries represent the predictions of the service providers' price-performance indexes with different timestamps.

### 5.2 Experimental Results

#### 5.2.1 Prediction Accuracy

We obtain the usage experience records as the examples shown in Table 2 and Table 3. According to the usage records, the service providers' price-performance indexes can be predicted. In Section 3.1, we predict a provider's price-performance index based on the direct usage experiences of the provider. To improve the prediction accuracy, we further find similar providers that have more recent predictions than the provider, and predict the price-performance index based on the usage experiences of the similar providers. For a provider, the parameter $\mu$ in Equation (10) controls how much the prediction relies on the direct usage experience of the provider and the usage experience of its similar providers. If $\mu = 0$, we predict the price-performance index using its direct usage experiences. If $\mu = 1$, we predict the price-performance index based on the usage experiences of the similar providers. Accordingly, a service requester can estimate the service provider's minimum price for a task based on the QoS requirement and the predicted price-performance index.

In the first set of experiments, we evaluate the accuracy of the prediction of minimum price. In the data, each provider's actual minimum price is between 0.0 and 1000.0.

We use the mean absolute error (MAE) to measure the prediction accuracy. MAE is defined as

$$\frac{1}{N}\sum_{i,j} |\bar{\delta}_{i,j} - \hat{\delta}_{i,j}|$$

where $\bar{\delta}_{i,j}$ denotes the actual minimum of Provider $a_j$ for Task $t_j$, $\hat{\delta}_{i,j}$ denotes the predicted price-performance index, and $N$ is the number of predicted values.

---

1. http://www.wsdream.net/dataset.html
Figure 7 shows the impacts of parameter $\mu$ on the prediction results. The MAE of the prediction based on the direct usage experience does not change whereas the parameter $\mu$ changes. Therefore, it is a horizontal line. We observe that incorporating the usage experience of the similar providers achieves better prediction accuracy than considering the direct usage experience alone. The optimal value of $\mu$ is around 0.5 for MAE.

5.2.2 Effectiveness
In the section, we conducted three sets of experiments. The configuration of each set of experiments is presented in Table 4. In set #2, there are two QoS attributes: response time and throughput. We change the global QoS requirement from (6.0,5.0) to (1.5,45.0) while decreasing the budget from 3000 to 1400. In the experiments, the requirements of the global QoS and budget ((6.0,5.0), 3000) are easily satisfied, but ((1.5,45.0), 1400) are difficult to satisfy. We use them to simulate different difficulty levels of the global QoS requirement and budget. In set #3, we increase the number of tasks within a requested service from 5 to 35 in steps of 5. In set #4, we increase the number of QoS dimensions from 2 to 8 to simulate different types of QoS requirement.

In the sets of experiments, we compare the effectiveness of our approach with that of Wang and Du's approach by the number of rounds taken to obtain a solution; the utilities of the service requester; and the time taken to obtain a solution. We report the data that are averaged over 100 requesters. In Wang and Du’s approach, they assume that the distribution of price-performance indexes and price function are public knowledge but does not discuss how to acquire the knowledge.

Figure 8(a), 9(a) and 10(a) show the number of rounds to obtain the solution of a QoS-aware service selection problem. In the Wang and Du's approach, a service requester knows only the distribution of the service providers' price-performance indexes. Therefore, it takes more rounds than our approach. Using the prediction approach, a service requester could predict the providers' minimum prices. Thus, a service requester finds a solution more quickly than Wang and Du's approach. In general, by using the our approach, the number of rounds taken is stable and it is between 2 and 3. Figure 8(b), 9(b) and 10(b) present the utility that the service requester obtains from a solution. Our approach outperforms Wang and Du's approach. On average, the service requester's utility increases from 703.1 to 906.6. The reason is that the service providers are motivated to submit their minimum prices in the multi-round Vickrey auction.

Figure 8(c), 9(c) and 10(c) present the time spent to obtain the solution of a QoS-aware service selection problem. We adopt a polynomial-time interior-point algorithm [17] to decompose the service requester's global requirements into a set of local QoS requirements. We observe that with increasing number of the tasks within a service request, the time taken to obtain a solution by using our approach is greatly less than the time taken by using Wang and Du's approach.

In summary, in our approach, a service requester is able to obtain a solution of a QoS-aware service more efficiently and effectively.

6. RELATED WORK
Currently, due to the increasing number of Web services, a service requester faces heavy computational burden to explore existing services. Bonatti and Festa [18] have proven that the problem of service selection and composition with global QoS requirement is NP-complete.

To address this issue, various heuristic algorithms have been proposed to find near-optimal solution [19]. Alrifai et al. [20] propose a heuristic algorithm by combining global optimization with local selection techniques. Mixed Integer Programming is first used to find the optimal decomposition of the global QoS requirement into a set of local QoS requirements and then best services are selected using local selection techniques. To reduce the number of services to be considered, they further propose an approach based on the

<table>
<thead>
<tr>
<th>Table 4. Experimental Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor</td>
</tr>
<tr>
<td>(Global QoS requirement, budget)</td>
</tr>
<tr>
<td>Number of Tasks</td>
</tr>
<tr>
<td>Number of QoS dimensions</td>
</tr>
</tbody>
</table>
A genetic algorithm is also a powerful tool for solving the service selection problem. Canfora et al. [22] adopt a genetic algorithm to find a near-optimal solution of the service selection problem. Their simulation results reveal that genetic algorithms outperform linear programming with increasing numbers of candidate services and tasks. Ma and Zhang [23] further propose enhanced initial population policy and evolution policy to improve the effectiveness, convergence and stability of the genetic algorithm for QoS-aware service selection. Klein et al. [24] consider that component services are generally executed repeatedly over long-term periods, and they propose a method for optimizing the QoS of candidate services to reach the long-term QoS requirement by using linear programming. However, these approaches focus on the specification of QoS metrics and efficient selection algorithms. The basic assumption they make is that service providers truthfully report their QoS. In real-world, this assumption might not hold.

Since, service providers do not report their QoS, a lot of work has been done to predict their QoS based on collaborative filtering [25]. Some neighborhood-integrated matrix factorization approaches [7] are proposed for collaborative and personalized QoS predictions. In addition, the Slope One collaborative filtering-based approach [26] is proposed for QoS prediction, where Slope One [27] is considered an effective prediction method due to its simpleness and high performance. QoS prediction approaches that do not incorporate data credibility may suffer a loss in prediction accuracy. Therefore, the reputation-aware QoS prediction approach [28] is proposed to improve the prediction accuracy by calculating service reputations based on their contributions. Different users may receive significantly distinct QoS from the same service provider due to a disparity in their locations, network conditions, and so on. The cluster-based prediction approaches [13] are proposed to predict QoS values based on clustering users and services. These approaches treat services as having static and non-configurable QoS and select the best service providers based on QoS. These approaches focus on only the QoS and ignore the fact that a provider's acceptable service price is negotiable and would change with the QoS.

Recently, the QoS-aware service selection is considered as a game between the service requester and the service provider. Zheng et al. [29] focus on the 1-to-1 bargaining game between a single service provider and a single service consumer where the service price is negotiable. However, in
general, there usually exist multiple service providers and the service requester needs to choose exactly one service provider for each task.

Auctions are widely considered to be an effective mechanism for addressing the pricing problem [30]. An important property of auctions is that service requester requires no information on service agents' pricing polices [31]. Comuzzi et al. [3] propose a multi-attribute and sealed-bid auction to contract with service agents. Mohabey et al. [32] propose a combinatorial auction-based approach to capture service agents' willingness to offer a composite service at a lower price than the total price of stand-alone services that form the composite service, as well as other QoS attributes. However, these approaches do not consider that there may not exist a solution to meet global QoS requirements based on the receiving bids. To reduce the potentially large search space, He et al. [11] propose an approach to filter out uncompetitive bids using historical usage experiences. These approaches use first-price auction to select service provider, where the service provider with the lowest price would be selected.

To motivate the service providers to provide the best QoS or price, Watanabe et al. [33] adopt a second-price Vickrey auction to select the best service agents. In a Vickrey auction, it is a dominate strategy for a service provider to provide the best price or QoS that it can offer. However, if the receiving bids cannot satisfy the global QoS requirement or the budget restriction, the service requester needs another round to ask the service providers to adjust their bids, until a solution to satisfy the global QoS requirement is found. In the case, the property that the service providers would provide the best price or QoS cannot hold any more.

In summary, first, the pricing problem is different from the QoS prediction problem. A provider's service price is negotiable and dependent upon his competitors. We propose the price-performance index to describe a provider's capability to offer a price for a desired QoS. And then, we estimate the provider's price by predicting his competitors' price-performance indexes. Second, we propose a multi-round Vickrey auction with reserve prices, where a service provider is encouraged to submit his minimum price if and only if his minimum price is less than the reserve price.

7. Conclusion

In QoS-aware service selection, a service requester and a service provider have a common interest in coming to an agreement about the QoS and the service price though they have opposite preferences regarding the QoS and the service price. A service provider can have flexible QoS and its service price could change over the QoS. Given a QoS requirement, the optimal candidate provider is the one that is willing to accept the lowest service price.

We posit that the service provider's price-performance index is crucial in finding the optimal service providers. However, a service provider does not directly reveal its minimum service price as well as its price-performance index to any requester. Thus, there is a challenge for the requester: how can it find the optimal service provider under incomplete information? Wang and Du [10] proposed an approach to find the optimal service providers by predicting their service prices based on only the distribution of provider's price-performance indexes. We thus obtained the usage experiences of requesters for various service providers. We predict a service provider's service price for a desired QoS based on these usage experiences. Based on the prediction results, we propose a multi-round Vickrey auction to obtain a solution that satisfy the QoS requirement and the budget restriction. Finally, we conduct a set of evaluations that show that our prediction approach is more effective and efficient than Wang and Du's approach [10].

Our current work generates the price-performance indexes randomly from a gamma distribution and assumes the distribution reflects real-world properties. To the best of our knowledge, there is no dataset that provides the service prices. In future, first, we plan to extract a dataset that provides real-world service prices. Second, we plan to consider the case where the service requester would have a cost to obtain the other service requesters' usage experiences or other service requesters would report false usage experiences to the service requester. Third, we plan to consider the group discount model in the QoS-aware service selection.

8. Acknowledgment

Wang was supported by the National Fundamental Research and Development Program of China (973 Program) under grant 2012CB316205, by the National Natural Science Foundation of China under grants 61003084, 61232007, and 61472428, by the Fundamental Research Funds for the Central Universities, and by the Research Funds of Renmin University of China under grant 14XNLQ06. Singh was partially supported by the US National Science Foundation under grant IIS-0910868.

9. References


