Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences

Hongbing Wang, *Member, IEEE*, Chao Yu, Lei Wang, and Qi Yu, *Member, IEEE*

Abstract—As the number of Cloud services is growing at a tremendous speed, there is an increasing number of service providers offering similar functionalities. Selecting services with user desired non-functional properties (NFPs) becomes of significant importance but triggers a number of Big Data related research issues. First, the selection decision should deal with a large volume of service NFPs data. Second, service selection needs to reflect diverse user preferences, including both qualitative and quantitative ones. Third, the uncertainty of the network and service load leads to high variability in NFPs. Fourth, as the trust values of service NFPs are collected via historic user's feedbacks,it brings the veracity dimension to the NFPs of services. Fifth, multiple and sometimes conflicting decision objectives for optimal service selection should be balanced. An effective service selection mechanism is in demand that can tackle all the above Big Data challenges in an integrated way to handle the highly diverse QoS with significant variability along with the trust related issues giving rise to data veracity. Existing investigations focus on either users' QoS preferences or their trust concerns but fail to provide a systematic solution to integrate both criteria in the selection process. In this paper, we tackle heterogeneous preference- and trust-based service selection by developing a novel multi-objective optimization approach to make trade-off decision between service's trust value and user's QoS preference to rank candidate Cloud services based on their match degrees with users' requirements. We conduct extensive experiments to evaluate the effectiveness and efficiency of the proposed approach.

Index Terms—Cloud service selection, trust, QoS preference, big data, multi-objective optimization, qualitative, quantitative

1 Introduction

With the advent of Cloud computing, rapidly increasing internet-based application programming interfaces (APIs) play an significant role in the forthcoming big data web environment [1]. More and more business applications are migrated to the Cloud inspired by the booming cloud computing [2]. As an example, in the service industry, *programmableweb.com*[1] (PW) has collected a repository containing over 12k APIs up to now, and the number has been nearly doubled year by year. Within the 467 API categories of PW, there are 2,096 APIs for Search, 589 for Database, and so on. Most of the APIs are hosted by well-known Cloud platforms (e.g., Google Cloud Bigtable and Microsoft Azure).

The Cloud services have a wide variety of functions in the API economy. Even for the Cloud services holding similar (or even identical) functionalities, the Quality of Service (QoS) is quite different and varying with time. The authenticity of QoS declared by service providers is also quite hard to be identified [3]. The Cloud service application desires for guaranteeing the long-term QoS by service selection [4]. Consequently, a number of typical Big Data challenges arise due to the high variety, low veracity, and fast-growing service data, which make it is a difficult and complex task for a user (e.g., service consumer, a service-based application developer, etc.) to select a most appropriate service under the big service space [5], [6], [7].

When a user asks for a Cloud service, the requirements are usually diverse [8]. After functional requirements are satisfied, a user may pay more attention to NFPs related requirements. Specifically, the NFPs of a Cloud service mainly contain QoS and user feedback for service trust. In different service selection scenarios, some users pay more attention to their QoS preferences while others care more about service's trust. More often, most users put the same emphasis on QoS preferences and trust properties in service selection. To select the most appropriate Cloud services for a service user, functionality-based service selection problem has been intensively investigated [9], [10]. Nonetheless, the rapidly increase of services competing to offer similar functionalities poses fundamental Big Data challenges for service selection which need to tackle various user requirements with large-scale highly various QoS data, while the veracity of trust is hard to be guaranteed [11], [12], [13], [14], [15], [16], [17].

Existing NFPs-oriented service selection approaches mainly focus on: (1) user's preference for QoS and (2) service's trust properties, separately. These approaches seldom simultaneously take the above two types of NFPs into consideration, which is desired by most service users [18], [19], [20]. Specially, preference-based service selection approaches mainly support user's qualitative or quantitative QoS preferences [21], [22], [23], [24]. However, qualitative and quantitative QoS properties are usually modeled and considered separately. Due to limited amount of QoS properties with similar measurement methods under consideration,

1. http://www.programmableweb.com/category

• *H. Wang, C. Yu, and L. Wang are with the School of Computer Science and Engineering and Key Laboratory of Computer Network and Information Integration, Southeast University, SIPAILOU 2, NanJing 210096, China. E-mail: {hbw, leiwang}@seu.edu.cn, 823797837@QQ.com.*
• *Q. Yu is with the College of Computing and Information Sciences, Rochester Institute of Technology, 102 Lomb Memorial Drive, Rochester, NY 14623-5608. E-mail: qi.yu@rit.edu.*

Manuscript received 28 Jan. 2015; revised 24 Aug. 2015; accepted 16 Nov. 2015. Date of publication 22 Sept. 2015; date of current version 3 Aug. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSC.2015.2480393

the requirement models, QoS metrics, as well as the service selection approaches will fall short when a large variety of QoS attributes been considered. On the other hand, trust-based service selection [25], [26], [27], [28], [29], [30], as another hot topic widely investigated by service computing researchers, has developed some metrics for establishing trust, reputation, and referral between trading partners. However, trust is usually considered independently from user's QoS preference when handling their requests.

The Big Data Cloud service environment makes effective service selection more challenging. First, a user may prefer a service holding a high trust value, but s/he may dislike the service as its price may go beyond his/her budget or other QoS properties (e.g., response time or platform style) could not fulfill the user's preference. Second, although a service fulfills the user's QoS preference, it may not be selected by a user if it is not trustworthy. Third, due to the high variability of QoS and uncertain veracity of trust, only by collectively considering the two dimensions of QoS and trust, can we guarantee the global optimization of a Cloud service selection decision. As a result, a viable service selection solution should take both QoS preference and service trust into account. We present the following Cloud service selection scenario to motivate the proposed work.

1.1 Motivating Scenario

Assume that a web application developer, say Nick, may choose one of the Cloud services for a Database application. There are massive candidate Cloud services in Internet that provide database service. Nick has specific requirements with respect to the type of application he is developing. First, an On-Line Analytical Processing (OLAP) application features a long running time of each service invocation (inv) and demands a large data storage space, despite that they are not requested frequently. Consequently, Nick prefers Cloud services concerning the NFPs of price (P), memory size (MS), and response time (RT) as

- $P \geq \$20/1{,}000\_inv/month$, $MS \geq 1TB$, and $RT \leq 1{,}000\,ms$

for OLAP application development. In contrast, when Nick is developing an On-Line Transaction Processing (OLTP) application, the application is invocation-intensive and requires a real-time service response. Consequently, the RT for each invocation should be lower than the OLAP application. We assume that the Cloud services are charged for invocation frequency per month. To reduce cost, Nick would prefer cheaper services, and the required MS could be smaller than the OLAP application. In sum, Nick's preference may be

- $P < \$20/1{,}000\_inv/month$, $MS < 1TB$, and $RT \leq 50\,ms$.

Second, besides the above quantitative preference over NFPs, conditional qualitative preferences may also held by Nick. For the OLAP application, he wants to integrate other related data for effective data analysis and prefers a mashup enabled Database Cloud service. To achieve this, Nick will choose a REST enabled Database Cloud Service, which can be easily integrated with the existing web applications. Meanwhile, Nick prefers a Database with higher level of security plug-ins for the application development of OLTP than OLAP. At last, Nick also has a high requirement for the reputation of service to guarantee the authenticity of QoS reported by service providers. This factor can be treated as the trust of the service. In short, we should consider both of the user's qualitative and quantitative preferences along with the trust of services in the service selection process as only via this way can we choose the best service to meet user requirements.

Cloud service selection when integrating service trust with user QoS preference faces the following Big Data challenges that existing approaches fail to address:

1) *Volume*: The quantity of competing functionally similar Cloud services is fast increasing. In addition, more complex user requirements on various aspect of NFPs of services will further expand the size of data to be processed and lead to a high computational complexity for service selection;
2) *Variety*: The diverse qualitative/quantitative QoS properties and service trust demand a uniform metric to aggregate various NFPs requirements;
3) *Variability/Veracity*: The variability of QoS and uncertain veracity of trust values make it challenging to decide what data to use, which further complicates the service selection process. There may also be some conflicts among multiple user requirements in QoS properties and services' trust;
4) *Multi-objective Decision*: The above-mentioned "4V" features make it difficult to be addressed via single objective decision for Cloud service selection. Trade-off decisions have to be made to balance the multiple and sometimes conflicting objectives. It will result in a high computational overhead.

To deal with the highlighted Big Data challenges, we propose a novel approach that integrates trust with QoS preference (involving both qualitative and quantitative ones) via multi-objective optimization for effective service selection. We identify the following contributions.

- We propose a systematic method to process heterogeneous data of QoS preferences and service trust. Different NFPs can be evaluated by "QoS match degree" using a uniform metric to judge how each property matches user requirements. Specifically, our method can deal with qualitative conditional preferences.
- We propose a multi-objective constrained method for Cloud service selection, which can balance multiple decision objectives of preference-, and trust-based service selection and deal with conflicts among the diverse user requirements and decision objectives.
- We perform experiments on both synthetic and real data sets to evaluate the effectiveness and efficiency of the proposed approach.

This paper was first presented in a shortened form as a conference paper in [31]. We further extend the theoretical analysis and present a comprehensive framework for effective service selection under Big Data environment. The remainder of the paper is organized as follows. We give the preliminary knowledge in Section 2. We introduce the proposed service selection approach in Section 3. We present

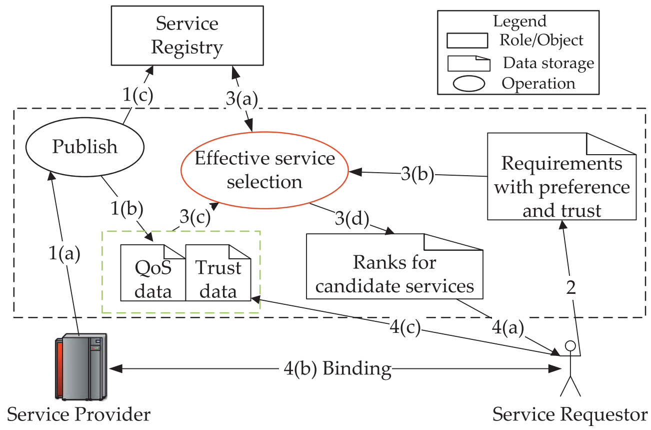


Fig. 1. Framework of cloud service selection procedure.

our experimental studies in Section 4. In Section 5, we review the related investigations. In Section 6, we conclude and layout some important future works.

2 Preliminaries

2.1 Definitions

In this section, we first present a standard process for Cloud service selection based on the traditional Service-oriented Architecture (or SOA) standards. As illustrated in Fig. 1, the process consists of the following four steps:

1) When Service Providers publish their Cloud services, we will collect and store QoS attribute values in QoS Data and then publish the functional detail into the Service Registry (see Fig. 1 the line 3(a)-(c)).
2) When Service Requestor needs a Cloud service, he can specify the Cloud service with his qualitative/quantitative preference as well as trust requirements (Fig. 1 line 2).
3) When we perform selection, we will first find candidate Cloud services in Service Registry according to function property (Fig. 1 line 3(a)). Second, we rank every candidate service and then select the best service to Service Requestor according to service's NFPs which contains QoS attribute values and trust information (Fig. 1 line 3(b)-(d)).
4) After Service Requestor finishes one calling process, we will store the user feedbacks for service's QoS in Trust Data (Fig. 1 line 4(c)). By this approach, the part in the dashed rectangle acts as a mediator between the client and provider.

Now, we will first give some formal definitions.

Definition 1 (Preference Independent and Mutually Preference Independent). *Let X, Y be two quantitative QoS attributes of Cloud services, $\forall\, x_1, x_2 \in X$, a weak order operator $\succeq$ is defined to represent a Preference Order (e.g., $x_1 \succeq x_2$ states that, the user more prefers the QoS attribute of x_1 than x_2). If $\exists\, x_1, x_2 \in X$, s.t., any $y \in Y$ satisfies $(x_1, y) \succeq (x_2, y)$, then we can say the attribute X is independent from Y (c.f. [32]). Similarly, if $\exists\, y_1, y_2 \in Y$, s.t., any $x \in X$ satisfies $(y_1, x) \succeq (y_2, x)$, then we can say the attribute Y is independent from X. Moreover, if X is independent from Y and Y is independent from X, then we say X and Y are mutually preference independent.*

Definition 2 (Utility Function). *The utility function(c.f. [33]) maps a Cloud service attribute value (the attributes are usually used to describe preference or trust) to a real number with the range between 0 and 1.*

Definition 3 (Additive Utility Function). *Given other conditions the same, the additive utility function of two attributes x, y can be defined as*

$$F(x, y) = F_1(x) + F_2(y), \tag{1}$$

where X and Y are quantitative attributes sets, and $x \in X$, $y \in Y$, $F_1(x)$ and $F_2(y)$ are the utility functions for attribute x and y, separately.

Note that additive utility function can guarantee different QoS attributes to be consistent and comparable. In this paper, to deal with the Big Data challenge of data variety, we will normalize inconsistent quantitative QoS properties using an additive utility function. The qualitative QoS properties are not mutually preference independent, and will be specially modeled by TCP-net (see Section 3).

Definition 4 (Candidate Services Class). *We define candidate Cloud services class $S = \{s_1, s_2 ... s_m\}$, which represents a collection of candidate Cloud services and achieving the same functionality but behave differently in QoS attributes.*

Definition 5 (Multi-dimensional QoS Model). *As the Quality of Service parameters describe the quality of a product or service. It serves as a benchmark to differentiate the services and the service providers. Given a candidate service j ($j \in S$), the Multi-dimensional QoS Model for j is a vector which can be expanded as $Q_j = \{q_1, q_2, \ldots, q_k, q_{k+1}, \ldots, q_n\}$, where $q_1, \ldots, q_k$ are quantitative properties and $q_{k+1}, \ldots, q_n$ are qualitative properties.*

Trust mechanism for service (or goods) selection is widely investigated in existing literatures [25], [30], [34]. In sum, the trustworthiness of a service should be estimated based on both direct and indirect experience. Direct experience means the quality of service evaluation received from the service user, whereas indirect experience comes from referrals by peers [15]. Previous works (e.g., [27]) established ways to model trust from indirect experience but this is beyond the scope of our paper. In this work, we follow the existing works [16], [35] that the service trust could be directly collected via service consumers' feedbacks into a central QoS registry. Note that similar trust evaluation mechanism has been used in real life E-commerce reputation systems, e.g., eBay, Sporas, Histos, Epinions, etc [30]. These provide evidence that this is a promising application for Cloud service marketplace. We give the following definition of trust for a candidate Cloud service.

Definition 6 (Multi-dimensional Trust of QoS). *We assume that each QoS attribute has an evaluation of trust value according to its historical invocation and reported by service users, we define Multi-dimensional Trust of QoS for candidate service j as $T_j = \{t_1, t_2, \ldots, t_n\}$, each trust value can be calculated as:*

$$t_i = \frac{1}{m}\sum_{k=1}^{m} t_{ik}, \tag{2}$$

TABLE 1
Relative Importance Settings by Integer Numbers (1-7)

Relative Importance	Definition	Details
1	Same Important	The importance of two targets are the same
3	A Little Important	One target is a little more important than the other
5	Important	One target is important than the other
7	Absolute Important	One target is very important than the other
2, 4, 6	Middle Important	The middle of two adjacent Definitions

where m is the total number of evaluators for candidate service j, t_{ik} $(1 \leq k \leq n)$ represents the kth *evaluation for the QoS property q_i to candidate service j.*

According to Definition 1 and Definition 2, we can get the Decision Matrix. We define it as:

Definition 7 (Decision Matrix). *The Decision Matrix [13] is a matrix that include all candidate services' QoS values, i.e.,*

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1j} & \cdots & y_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{i1} & \cdots & y_{ij} & \cdots & y_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{m1} & \cdots & y_{mj} & \cdots & y_{mn} \end{bmatrix} \tag{3}$$

where y_{ij} denotes the QoS property $q_i^{\prime}s$ value of candidate service j.

In this paper, user preferences contain three aspects: (1) the constraints on quantitative QoS property, (2) the conditional preference on qualitative QoS property, and (3) the relative importance about one QoS property to another.

First, for quantitative QoS property, user often has constraint on each QoS property, such as response time [100 ms, 200 ms]. We assume the user's constraint on q_i for candidate service j as $[c_{ij}^{'}, c_{ij}^{''}]$, which means *constraints on quantitative QoS property*.

Second, for qualitative QoS property, user's preferences over these properties are often conditional and cannot be directly expressed in numerical value. For example, a data storage service has two properties, e.g., Platform and Location. Suppose a Platform is a file system, the user may prefer the service to be located in USA. Furthermore, if the Platform is a database, the user may prefer it to be located in China. Such preference means *conditional preference on qualitative QoS property*.

Third, when performing service selection, the user may consider every QoS property not the same importance level. For example, the user may consider the q_1 is three times as q_2, and q_2 is two times as q_3. Such preference means *relative importance about one QoS property to another*. We refer to Saaty's 1-to-9 scale for AHP and ANP preferences [36] and map the relative importance to integer numbers by Table 1. Consequently, we can get the relative importance between every two QoS attributes for service j. After n(n + 1)/2 comparisons we can obtain a relative importance matrix, i.e.,

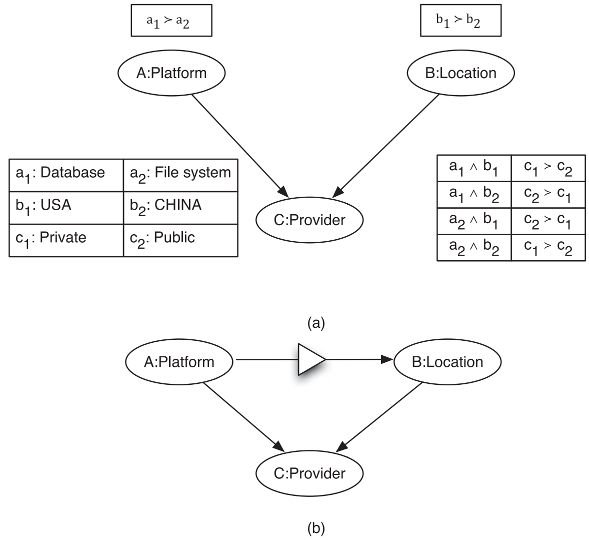


Fig. 2. Examples of: (a) CP-net; (b) TCP-net.

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{\hat{i}1} & \cdots & a_{\hat{i}i} & \cdots & a_{\hat{i}n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix}, \tag{4}$$

where $a_{\hat{i}i}$ represents the relative importance between QoS properties $q_{\hat{i}}$ and q_i.

Let $1 \leq \hat{i}, i, k \leq n$ and we have the following: (1) $a_{ii} = 1$; (2) $a_{i\hat{i}} = 1/a_{\hat{i}i}$. Due to the subjective expression of relative importance by users, conflicts may appear among different attributes. Hence, we do not consider the transitivity of preference in this paper.

2.2 TCP-Net

To model the conditional preference on qualitative QoS property, Boutilier et al. in [37] first proposed a Conditional Preference Networks (CP-net) model. In [38], we have successfully introduced a CP-net-based model to represent the qualitative preferences for service selection.

We first show an example to illustrate CP-net and explain its limitation. Suppose we consider a Cloud service selection issue and a data storage service has some qualitative attributes, for example, Platform (a_1: Database, a_2: File System), Location (b_1: USA, b_2: CHINA), and Provider (c_1: a private company, or c_2: a public organization).

We suppose a user Nick's preference statements are modeled by a CP-net. As shown in Fig. 2a, a CP-net is consists of a graphical model (i.e., the directed graph) and logical representation (i.e., the preference statements as Conditional Preference Tables) (c.f., [38]). From the CP-net we can see, Nick prefers database to file system (i.e., $a_1 \succ a_2$) and USA to CHINA (i.e., $b_1 \succ b_2$). Then Nick's preference on provider depends on the platform and the location of service. If the service is based on database in USA (i.e., $a_1 \wedge b_1$), Nick prefers a private company to a public organization (i.e., $c_1 \succ c_2$). Otherwise, if the service is based on database in CHINA (i.e., $a_1 \wedge b_2$), Nick prefers a public organization to a private company (i.e., $c_2 \succ c_1$). If the service is based on file system in USA (i.e, $a_2 \wedge b_1$), Nick prefers

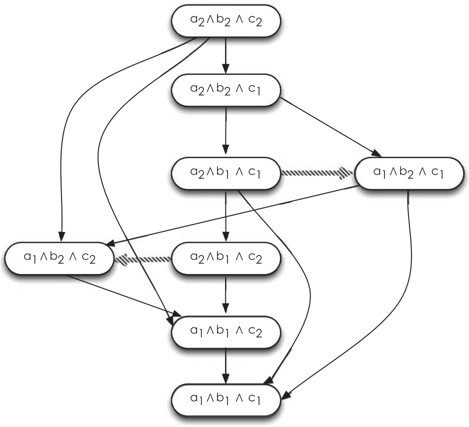


Fig. 3. Induced preference graph.

a private company to a public organization (i.e., $c_2 \succ c_1$). If the service is based on file system in CHINA (i.e, $a_2 \wedge b_2$), Nick prefers a public organization to a private company (i.e., $c_1 \succ c_2$).

With the CP-net model, we can infer the user's preference under some attribute sets. For example, if we compare the attribute sets $a_1b_1c_1$ VS $a_1b_1c_2$, since a_1b_1 is the same, and the user more prefers c_1 than c_2, then we can identify $a_1b_1c_1 \succ a_1b_1c_2$. But for some other attributes composition patterns, for example, $a_1b_2c_1$ VS $a_2b_1c_1$, we cannot identify which set does the user more prefers.

To enhance the expressive power of CP-net, Brafman et al. proposed TCP-net by introducing relative importance [39]. In this paper, we adopt TCP-net (c.f., Fig. 2b) to express the user's qualitative preferences, which can support relative importance preferences of qualitative QoS properties for Cloud service selection. Formally, the TCP-net is defined as follows.

Definition 8 (TCP-net). *Given a candidate service j, for $j \in S$, let $X = X_1, X_2, \ldots, X_n$ be a set of attributes for service j. A TCP-net over X is a directed graph G over $X_1, X_2, \ldots, X_n$, modeling the qualitative conditional preferences. There are two types of arcs in TCP-nets. The directed arc stands for conditional preference. The directed arc with a triangle stands for relative importance.*

As shown in Fig. 2b, the user will preferentially consider the platform of service at selection time by introducing the relative importance. As illustrated in Fig. 3, we can obtain a complete Induced Preference Graph (c.f., [39]) based on the TCP-net preference statement.

From Fig. 3, we can immediately conclude that $a_1b_2c_1 \succ a_2b_1c_1$, and $a_1b_2c_2 \succ a_2b_1c_2$ (see the dashed arrows in Fig. 3). Note that in the induced preference graph, the top and the bottom elements are the worst and the best outcomes, respectively. And the arcs points to the more preferred outcomes.

3 The Integrated Service Selection Approach

The *integrated service selection approach* is proposed to address the Big Data challenges (see Section 1). To deal with large volume and high variety of the NFPs data, we extend the method to process the heterogeneous information. As a result, the match degree of user requirement for each QoS property can be calculated and compared using a uniformed metric. For the challenges of variability and veracity, the match degree is calculated by collectively considering the QoS preferences and service trust. To deal with the conflicts among multiple decision objectives, the service selection decision will be made via a multi-objective optimization model.

In sum, the proposed Cloud service selection approach consists of the following three phases. First, we handle inconsistent quantitative QoS properties by dimensionless additive utility functions (i.e., with a range of [0,1]). Second, we calculate the QoS match degree for each quantitative/qualitative QoS property of the candidate services by integrating trust with user constraints. Specially, a qualitative QoS property is not a numerical value. It is in dependence of other qualitative QoS properties when representing user's preference. We can not simply normalize qualitative QoS properties. These properties would be specially processed when calculating their QoS match degrees (see Section 3.2). Third, we define a linear function via a Multi-objective Constrained Model to calculate the ranking value for each candidate service.

3.1 Normalization for Quantitative QoS Properties

In this section, we normalize the quantitative QoS properties in the decision matrix (see Definition 7) uniformly. This aims to address the data variety issue resulted from the various quantitative QoS properties.

First, some QoS property values are the larger the better, such as reliability, availability, which can be attributed to *benefit-oriented QoS indexes*. Some QoS property values are the less the better, such as response time and renting cost, which can be attributed to *cost-oriented QoS indexes*. The diversity of QoS values blocks the comparison for service selection under a uniform metric. Therefore we need to preprocess the decision matrix to ensure that larger QoS values can indicate better service quality.

Second, different QoS property values have different dimensions. Even for the same property, using different dimensions may result in different numerical values, for example, 10,000 ms vs 10 s. Therefore we need to eliminate the effect of different dimensions, so-called removing dimensionless.

We perform the min-max transformation (c.f. [24]) for the decision matrix. Let $z_{ij} \in [0, 1]$ be the handled dimensionless value for a QoS property q_i of service j. For benefit-oriented QoS indexes, the greater the original value is, the better the quality of service is. Hence,

$$z_{ij} = \begin{cases} \frac{y_{ij} - y_j^{\min}}{y_j^{\max} - y_j^{\min}}, & y_j^{\max} - y_j^{\min} \neq 0 \\ 1, & y_j^{\max} - y_j^{\min} = 0. \end{cases} \quad (5)$$

where y_{ij} represents ith QoS property of service j, $y_j^{max} = max\{y_{ij}\}, y_j^{min} = min\{y_{ij}\}$, for $i \in [1, k]$ and $j \in [1, m]$, k represents the number of quantitative QoS properties and m is the number of candidate services, respectively.

For cost-oriented QoS indexes, the less of the original value is, the better the quality of service is. We will have

$$z_{ij} = \begin{cases} \frac{y_j^{\max} - y_{ij}}{y_j^{\max} - y_j^{\min}}, & y_j^{\max} - y_j^{\min} \neq 0 \\ 1, & y_j^{\max} - y_j^{\min} = 0. \end{cases} \quad (6)$$

To facilitate the expansion for neither benefit-oriented nor cost-oriented QoS properties, we also take another preprocessing method for quantitative QoS properties. We assume the optimal value range for a service j be $[y_j^0, y_j^*]$, and $y_j^{'}$ is defined as the lower limit which user can tolerate, $y_j^{''}$ be the upper limit which users may demand. In practice, the optimal QoS value internal should be set according to the specific application. Then

$$z_{ij} = \begin{cases} 1 - \frac{y_j^0 - y_{ij}}{y_j^0 - y_j^{'}}, & y_j^{'} < y_{ij} < y_j^0 \\ 1, & y_j^0 \leq y_{ij} \leq y_j^* \\ 1 - \frac{y_{ij} - y_j^*}{y_j^{''} - y_j^*}, & y_j^* < y_{ij} < y_j^{''} \\ 0, & otherwise. \end{cases} \tag{7}$$

Similarly, the trust value for each dimension of a Cloud service's QoS property can be preprocessed in the same way. As trust is obviously a type of benefit-oriented index, we take the first method (i.e., (5)) to normalize it.

After normalization, the user's preference for any quantitative QoS properties would be expressed under an unified metric. As a result, we can assume the user's constraint on quantitative QoS properties q_i be $[p_j^{'}, p_j^{''}]$, for $0 \leq p_j^{'}, p_j^{''} \leq 1$, i.e., only when the QoS property value z_{ij} falls between $p_j^{'}$ and $p_j^{''}$, will we say the property q_i of service j meets user's requirement.

3.2 QoS Match Degree for Each QoS Property

We define *QoS Match Degree* $f_i(j)$ as how much one QoS property q_i of candidate service j satisfies the user's requirements. For the Big Data challenge 3 (see Section 1), to ensure the effectiveness of service selection, we will also take the trust value for QoS into account when calculating the QoS match degree for each QoS property.

For quantitative QoS property, let $i = 1, 2, .., k$, $f_i(j)$ is defined as follows:

$$f_i(j) = \begin{cases} \frac{z_{ij} - p_j^{'}}{(p_j^{'} + p_j^{''})/2} * t_{ij}, & 0 \leq z_{ij} < (p_j^{'} + p_j^{''})/2 \\ t_{ij}, & z_{ij} = (p_j^{'} + p_j^{''})/2 \\ \frac{p_j^{''} - z_{ij}}{(p_j^{'} + p_j^{''})/2} * t_{ij}, & (p_j^{'} + p_j^{''})/2 < z_{ij} \leq 1, \end{cases} \tag{8}$$

where t_{ij} represents the user evaluated trust value to service j's ith QoS property, $[p_j^{'}, p_j^{''}]$ represents the user constraint for service j's QoS property q_i.

From the above equation, we can say that if a QoS value violates user preference, the QoS Match Degree is a negative number, and when the absolute value of the violation is larger, the service's QoS value will more deviate from the user's requirement. Moreover, no matter the QoS match degree is negative or positive, the larger the trust of QoS the larger the absolute value of the violation.

However, the above method is not applicable for qualitative QoS properties, since a qualitative QoS property is not a numerical value and it often contains conditional attribute (the formation of an attribute group).

TCP-net [40] is popular adopted as the formal model for representing and reasoning with qualitative preferences (retrieve Section 2.2). In this paper, we assume the user's qualitative preference statements are modeled by TCP-net. We assume user U_i and user U_j have expressed their qualitative preferences over same service attributes, separately, and the preferences have been modeled by TCP-net graph model and logical representation. Suppose there are a pair of nodes A, B in the above TCP-net models, which is consisted by a series of service attribute values collection. If the preference order for attribute values collections A, and B is $A \succ B$ in U_i's TCP-net model, and $A \succ B$ also holds in U_j's TCP-net model. Then the qualitative preference similarity between U_i and U_j on A and B could be induced, i.e., their preferences on the pair of these two attribute sets are the same.

When the user's qualitative preference is modeled by a TCP-net. We use the topological sorting method to obtain a topological order from the TCP-net's induced preference graph. The algorithm is shown as Algorithm 1.

Algorithm 1. Qualitative TCP-net Preferences Sorting Algorithm

Input: The Induced Preference Graph of user-described TCP-net model, G;
Output: The user preference set sequence, V;

```
Stack < attribute set > S;
Vector < attribute set > V;
FindInDegree(G, in-degree);
InitStack(S);
while The number of nodes in G > 0 do
    Anonymously select a Node A in G whose in-degree is 0;
    S.push(A);
    Decrement one in-degree for each adjacency nodes of A;
    Delete Node A and its induced arcs from G;
end while
while ! S.empty() do
    V.push_back(S.top());
    S.pop();
end while
return V;
```

As can be seen from Algorithm 1, we choose one node A whose in-degree is zero. We add A to the result set, and decrement the in-degrees of all nodes related to A, then repeatedly until the TCP-net has no node. As an example, Fig. 4 shows the first three steps' preferences sorting results for the TCP-net illustrated in Fig. 3.

As shown in Table 2, we append an ID for each node of TCP-net's induced preference graph. In finally, the sequence of the nodes by our preferences sorting algorithm is: $1 \succ 2 \succ 4 \succ 6 \succ 3 \succ 5 \succ 7 \succ 8$. The sequence (the more front the better) determines how each node (i.e., attribute set for qualitative preference) matches the user's requirement.

Let the maximal ID be CPN, $\forall$ an attribute group sp and its topological sorting sequence number $od(sp)$, the QoS Match Degree for sp will be

$$f_{sp} = \frac{CPN - od(sp)}{CPN}. \tag{9}$$

Similar to the quantitative QoS property, we also consider the trust value for each qualitative QoS property when calculating the QoS match degree. Let $i = k + 1, \ldots, n$, and

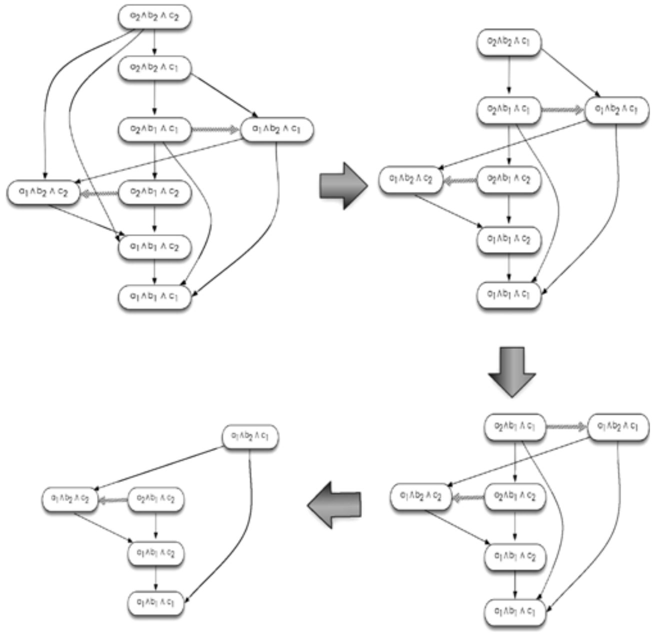


Fig. 4. A concrete example for TCP-net preferences sorting.

$od(sp_j)$ be the topological sorting sequence number for a service j's qualitative QoS attribute set sp_j, we decompose the QoS Match Degree of sp_j to each qualitative QoS property, and will get the QoS Match Degree for each qualitative QoS property:

$$f_i(j) = f_{sp_j} = \frac{CPN - od(sp_j)}{CPN} * t_{ij}. \tag{10}$$

The above-mentioned Equation (8) together with (10) consist the complete QoS Match Degree for all of the QoS properties, which is used to evaluate how much each QoS attribute of a service j satisfies the user's quantitative and qualitative requirements.

3.3 Multi-Objective Constrained Model for Service Ranking

We define a linear function to calculate the ranking value of each candidate service. We assume w_i is the weighting-value of QoS property q_i, for $1 \leq i \leq n$, the additive utility function of q_i for candidate service j can be represented as $w_i * f_i(j)$. The overall QoS ranking value V_j for candidate service j can be obtained from a weighted sum of the normalized single QoS property value, i.e.,

$$V_j = \sum_{i=1}^{n} w_i * f_i(j). \tag{11}$$

What follows, we will present two different approaches to solve the problem of estimating w_i. First, we estimate w_i based on the relative importance among QoS properties given by the user. We then integrate the service trust value.

3.3.1 Objective Estimation: According to Preference

Let $a_{\hat{i}i}$ be the user described relative importance for QoS properties between $q_{\hat{i}}$ and q_i, for $1 \leq \hat{i}, i \leq n$. If the user can

TABLE 2
ID Setting for TCP-Net Nodes

Attribute set	$a_1b_1c_1$	$a_1b_1c_2$	$a_1b_2c_1$	$a_1b_2c_2$	$a_2b_1c_1$	$a_2b_1c_2$	$a_2b_2c_1$	$a_2b_2c_2$
ID	1	2	3	4	5	6	7	8

exactly estimate each $a_{\hat{i}i}$ with none conflicts, then $a_{\hat{i}i} = w_{\hat{i}}/w_i$, we can obtain the following equation:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{\hat{i}1} & \cdots & a_{\hat{i}i} & \cdots & a_{\hat{i}n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} w_1/w_1 & \cdots & w_1/w_i & \cdots & w_1/w_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{\hat{i}}/w_1 & \cdots & w_{\hat{i}}/w_i & \cdots & w_{\hat{i}}/w_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_n/w_1 & \cdots & w_n/w_i & \cdots & w_n/w_n \end{bmatrix}. \tag{12}$$

From the equation above, we can deduce

$$\sum_{\hat{i}=1}^{n} a_{\hat{i}i} = \frac{\sum_{\hat{i}=1}^{n} w_{\hat{i}}}{w_i}. \tag{13}$$

And if we assume

$$\sum_{\hat{i}=1}^{n} w_{\hat{i}} = 1. \tag{14}$$

Then we can get w_i:

$$w_i = \frac{1}{\sum_{\hat{i}=1}^{n} a_{\hat{i}i}}. \tag{15}$$

The user may however not always estimate every relative importance $a_{\hat{i}i}$ exactly, e.g., inconsistent or even conflict may exist in the subjectively estimation of different QoS properties. To solve this problem, we will use Weighted Least Squares method to optimize the estimation of w_i:

Problem 1: **Minimize:**

$$f_1 = \sum_{\hat{i}=1}^{n} \sum_{i=1}^{n} \left(a_{\hat{i}i} w_i - w_{\hat{i}}\right)^2.$$

Subject to:

- $\sum_{i=1}^{n} w_i = 1$, and
- $0 < w_i < 1$, for $i = 1, 2, \ldots, n$.

3.3.2 Objective Estimation: Combined with Trust

For the variability nature of service QoS, we can also estimate the weighting-value of w_i according to the service's trust values. The trust values are collected from the historical users' real experience by invoking the Cloud services, and can objectively reflect the service's QoS performance.

If all the trust evaluation of users are honest, the objective evaluation according to service's trust value during different time period can better reflect the service's real average performance.

Let $t_{k\hat{i}}$ and t_{ki} represent the trust values for QoS property $q_{\hat{i}}$ and q_i, respectively, evaluated by user k, for $1 \le k \le m$, $w_{\hat{i}}$ and w_i represent the weighting-value for $q_{\hat{i}}$ and q_i, separately. Like subjective preference on relative importance, we use *Weighted Least Squares* method to deal with conflicts among different QoS properties and estimate the weighting-value of w_i for service j according to service trust:

Problem 2: **Minimize:**

$f_2 = \sum_{k=1}^m \sum_{\hat{i}=1}^n \sum_{i=1}^n \left(\frac{t_{k\hat{i}}}{t_{ki}} * w_i - w_{\hat{i}}\right)^2.$

Subject to:

- $\sum_{i=1}^n w_i = 1$, and
- $0 < w_i < 1$, for $i = 1, 2, \ldots, n$.

Service selection according to user described relative importance (i.e., Problem 1) can effectively capture user preference but not the service's actual performance. It falls short in dealing with the variability challenge. In Problem 2, selection decision is made according to service trust. Trust data is collected over the time and can better reflect the average performance of a service. It helps select services with higher global QoS performance but cannot reflect user preference and suffers from the data veracity issue. To balance the challenges of variability and veracity, we will integrate these two factors instead of solving each problem individually. Constraints combining QoS preference and service trust can be represented as follows.

Problem 3: **Minimize:**

- $f_1 = \sum_{\hat{i}=1}^n \sum_{i=1}^n \left(a_{\hat{i}i} w_i - w_{\hat{i}}\right)^2$
- $f_2 = \sum_{k=1}^m \sum_{\hat{i}=1}^n \sum_{i=1}^n \left(\frac{t_{k\hat{i}}}{t_{ki}} * w_i - w_{\hat{i}}\right)^2.$

Subject to:

- $\sum_{i=1}^n w_i = 1$, and
- $0 < w_i < 1$, for $i = 1, 2, \ldots, n$.

To solve this problem, we can construct a multi-objective planning:

Multi-objective Planning: **Minimize:**

$$f = f_1 + f_2 = \sum_{\hat{i}=1}^{n} \sum_{i=1}^{n} \left(a_{\hat{i}i} w_i - w_{\hat{i}}\right)^2 + \sum_{k=1}^{m} \sum_{\hat{i}=1}^{n} \sum_{i=1}^{n} \left(\frac{t_{k\hat{i}}}{t_{ki}} * w_i - w_{\hat{i}}\right)^2.$$

Subject to:

- $\sum_{i=1}^n w_i = 1$, and
- $0 < w_i < 1$, for $i = 1, 2, \ldots, n$.

We take Lagrange multipliers on this planning and the Lagrange function is defined as:

$$L = \sum_{\hat{i}=1}^{n} \sum_{i=1}^{n} \left(a_{\hat{i}i} w_i - w_{\hat{i}}\right)^2 + \sum_{k=1}^{m} \sum_{\hat{i}=1}^{n} \sum_{i=1}^{n} \left(\frac{t_{k\hat{i}}}{t_{ki}} * w_i - w_{\hat{i}}\right)^2 + 2\lambda\left(\sum_{i=1}^{n} w_i - 1\right).$$

Then we make the partial derivative of w_l on L, and make the results of partial derivative to zero. Along with $\sum_{l=1}^n w_l = 1$, we will get n equations as:

$$\begin{cases} \frac{\partial L}{\partial w_1} = \sum_{\hat{i}=1}^n \left(a_{\hat{i}1} w_1 - w_{\hat{i}}\right) a_{\hat{i}l} - \sum_{i=1}^n \left(a_{1i} w_i - w_1\right) + \\ \sum_{k=1}^m \sum_{\hat{i}=1}^n \left(\frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{k1}} w_1 - w_{\hat{i}}\right) \frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{kl}} - \sum_{k=1}^m \sum_{i=1}^n \left(\frac{\mathrm{t}_{k1}}{\mathrm{t}_{ki}} w_i - w_1\right) + \lambda = 0, \\ \cdots \\ \frac{\partial L}{\partial w_l} = \sum_{\hat{i}=1}^n \left(a_{\hat{i}l} w_l - w_{\hat{i}}\right) a_{\hat{i}l} - \sum_{i=1}^n \left(a_{li} w_i - w_l\right) + \\ \sum_{k=1}^m \sum_{\hat{i}=1}^n \left(\frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{kl}} w_l - w_{\hat{i}}\right) \frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{kl}} - \sum_{k=1}^m \sum_{i=1}^n \left(\frac{\mathrm{t}_{kl}}{\mathrm{t}_{ki}} w_i - w_l\right) + \lambda = 0, \\ \cdots \\ \frac{\partial L}{\partial w_n} = \sum_{\hat{i}=1}^n \left(a_{\hat{i}n} w_{\mathrm{n}} - w_{\hat{i}}\right) a_{\hat{i}n} - \sum_{i=1}^n \left(a_{ni} w_i - w_n\right) + \\ \sum_{k=1}^m \sum_{\hat{i}=1}^n \left(\frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{kn}} w_{\mathrm{n}} - w_{\hat{i}}\right) \frac{\mathrm{t}_{k\hat{i}}}{\mathrm{t}_{kn}} - \sum_{k=1}^m \sum_{i=1}^n \left(\frac{\mathrm{t}_{kn}}{\mathrm{t}_{ki}} w_{\mathrm{i}} - w_n\right) + \lambda = 0, \\ \sum_{l=1}^n w_l = 1. \end{cases}$$

Since there are $n + 1$ variables and $n + 1$ equations, the weighting vector $W = [w_1, w_2 \ldots w_n]^T$ will be easily calculated. Finally, we will apply the Ranking Function (i.e., (11)) to calculate the ranking value for each candidate service $S = \{s_1, s_2 \ldots s_m\}$. The service obtained the highest ranking value will be selected to the desired user.

4 Experimental Evaluation

In this section, we conducted some experiments to evaluate the service selection approach proposed in this paper. To investigate the applicability of our approach in Big Data Cloud service-based applications, the experiments aim at answering the following two research questions (RQ).

- RQ 1: How about the effectiveness of the proposed approach in terms of user satisfaction, especially when compared with competitive methods?
- RQ 2: How about the computational complexity of such an integrated service selection approach?

All the experiments were implemented using java 1.6, on an IBM server with 8 CPUs of 2.13 Ghz and 16 GB RAM. The operating system is Windows Server 2003.

4.1 Data Set

The publicly available QWS dataset[2] is used in our experiments. QWS contains 5,000 real services with QoS data. For better simulating the real-life application of BigData-space service selection and assure that the experimental results are not biased, we also generated the second dataset with 10,000 services, which are assigned with arbitrary QoS values. The settings of QoS attributes for our synthetical dataset are shown in Table 3. Although the service trust data can be directly or indirectly collected [15], due to the lack of publicly available real service trust data [35], exiting research works mainly use artificial synthetic data for service selection experiments [3], [15], [17], [19], [35]. Meanwhile, the synthetic data can avoid bias and guarantee the objectivity of the experiment, if the artificial data is approximately consistent with the real data. In this paper, similar to [15], [17], [35], the trust values for the services in QWS and synthetic dataset are randomly generated in a normal distribution. For both QWS and synthetic dataset, we also randomly generate user's requirements with QoS prosperities and relative importance for the QoS attributes.

2. http://www.uoguelph.ca/ qmahmoud/qws/

TABLE 3
QoS Attributes Settings for Our Synthetical Dataset

Attribute	Value of Attribute
cost($)	(0,2,000)
response time (ms)	(0,400)
reputation	(1,10)
success rate	(0,100)
reliability	(0,100)
location	{Shanghai, Beijing, London}
privacy	{visible to everyone, visible to the same network, not visible}
number of concurrent	(0,1,000)
availability	(0,100)

Each group of experiment was performed for 50 times repeatedly, and the average performance was calculated.

4.2 Effectiveness

To investigate RQ 1, we first compare the effectiveness of our proposed approaches by exploring relevance, confidence, as well as availability degrees between the desired and selected services for different approaches.

4.2.1 Approaches under Comparison

To analyze the effectiveness on integrating user preference with service trust, we implement the following three different approaches for Cloud service selection based on our proposed method. Each approach will be applied to the two datasets for comparison.

1) *Preference-Only*: When calculating the ranking value for each candidate service, we only according to the user's preference over quantitative, qualitative QoS properties and relative importance for different QoS properties. To achieve this goal, we remove all trust information in the decision matrix. The QoS match degree by Preference-Only (p-o) for each quantitative QoS property is defined as:

$$f_i^{p-o}(j) = \begin{cases} \frac{z_{ij}-p_j^{'}}{(p_j^{'}+p_j^{''})/2}, & 0 \leq z_{ij} < (p_j^{'}+p_j^{''})/2 \\ 1, & z_{ij} = (p_j^{'}+p_j^{''})/2 \\ \frac{p_j^{''}-z_{ij}}{(p_j^{'}+p_j^{''})/2}, & (p_j^{'}+p_j^{''})/2 < z_{ij} \leq 1, \end{cases} \tag{16}$$

where $z_{ij} \in [0,1]$ represents the handled dimensionless value for QoS property q_i of candidate service j, for $1 \leq i \leq k$, and $[p_j^{'}, p_j^{''}]$ represents the user constraints for QoS property q_i.

The QoS match degree for each qualitative QoS property is defined as:

$$f_i^{p-o}(j) = \frac{CPN - od(sp_j)}{CPN}. \tag{17}$$

where $k+1 \leq i \leq n$, CPN represents the maximal ID for the nodes in the induced preference graph, sp_j represents candidate service j's qualitative QoS preference attribute set, and $od(sp_j)$ represents the topological sorting sequence number for sp_j.

In the same way, we only take the objective estimation of Problem 1 (see Section 3.3.1) to get the weighting value of w_i for each QoS property q_i.

2) *Trust-Only*: When calculating the ranking value for each candidate service, we only consider the trust values for each QoS property. All constraints on QoS preferences are removed in the service selection decision process. Consequently, the QoS match degree by Trust-Only (t-o) for each QoS property is defined as:

$$f_i^{t-o}(j) = t_{ij}. \tag{18}$$

where t_{ij} represents the trust value of service j́s ith QoS property.

Similarly, we only take the objective estimation of Problem 2 (see Section 3.3.2) to get the weighting value of w_i for each QoS property q_i.

3) *Integrated* approach: We combine QoS preference with trust by the service selection approach proposed in this paper, and take the multi-objective optimization approach of Problem 3 to get the weighting value of w_i for each QoS property q_i.

4.2.2 Metrics

We use the following three metrics to evaluate the performance of the above-mentioned threefold service selection approaches, i.e., relevance degree (RVD), confidence degree (CFD), and availability degree (AAD).

Let Φ be the set of selected services under several times of service selections by a specific service selection approach, and $\Psi = \{q_i\}$ be the set of QoS properties for each service, for $i \in [1, n]$.

First, RVD is defined to evaluate how much the selected services' QoS properties set match the user's preferences, i.e.,

$$RVD = \frac{1}{\kappa \times n} \sum_{i \in [1,n]} \sum_{\mu \in \Phi} f_i^{p-o}(\mu), \tag{19}$$

where $\kappa = |\Phi|$ represents the number of selected services, n represents the number of QoS properties for each service, i represents ith QoS property, μ represents μth selected service, and $f_i^{p-o}(\mu)$ represents the QoS match degree by Preference-Only for ith QoS property of service μ.

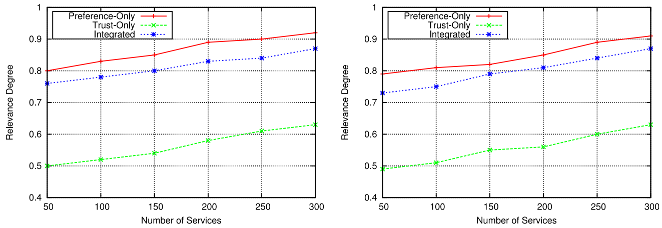
Second, CFD is defined to evaluate the match degree between the selected services' QoS values and the services' real performance (i.e., the historic user evaluation as the trust values), i.e.,

$$CFD = \frac{1}{\kappa \times n} \sum_{i \in [1,n]} \sum_{\mu \in \Phi} f_i^{t-o}(\mu), \tag{20}$$

where $f_i^{t-o}(\mu)$ represents QoS match degree by Trust-Only for ith QoS property of service μ.

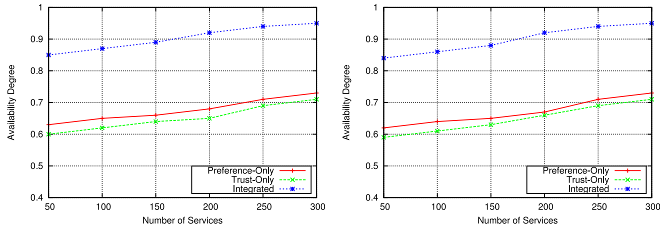
Third, we combine QoS preference with trust and define AAD as the metric to evaluate how much the selected services' real QoS performance match the user's requirements, i.e.,

$$AAD = \frac{1}{\kappa \times n} \sum_{i \in [1,n]} \sum_{\mu \in \Phi} f_i(\mu), \tag{21}$$



(a) QWS Dataset (b) Synthetic Dataset

Fig. 5. Comparison on relevance degree (RVD).



(a) QWS Dataset (b) Synthetic Dataset

Fig. 7. Comparison on availability degree (AAD).

where $f_i(\mu)$ represents the QoS match degree for ith QoS property of service μ, and can be calculated by (8) and (10) for quantitative and qualitative QoS properties and combined with trust.

4.2.3 Performance Evaluation

To investigate the effectiveness of our approach, we select all QoS properties for the services in our dataset, and vary the number of services for from 50 to 300 with a step value of 50. We use Preference-only, Trust-Only, and the Integrated approach for service selection, separately, and compare the RVD, CFD, and AAD for the results of service selection via the above threefold approaches, respectively.

As can be seen from Figs. 5, 6, and 7 with the increment of service number, we will get higher RVD, CFD, and AAD for all of the service selection approaches. It is because a larger scale of candidate service pool can provide more choices to select the most appropriate service for user. We also identify some interesting findings based on the experimental results under different metrics.

First, the RVD for Preference-only is higher than other two approaches, and the RVD value for Integrated approach is higher than Trust-only (see Fig. 5). This observation indicates that the selection results by Preference-only is more fit for user's preference. The Trust-only approach puts more weighting value on the services having higher trust values. This is consistent with the actual situation, because the user's subjectively preferred QoS properties may hold a lower trust value, due to the high variability of QoS and uncertain veracity of service trust evaluators.

Second, we obtained highest CFD values by Trust-only. The CFD values by Preference-only are lower than the Integrated approach (see Fig. 6). It is because the trust value for each service is based on the historic user's evaluation for the services. When calculating the ranking value for candidate services, the Trust-only approach gives higher weight value for the services holding a higher trust value. This observation indicates that, the selected services' QoS values by Trust-only service selection approach are closer to the historic users' evaluation.

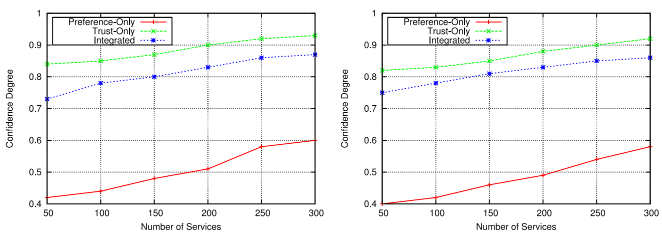
Third, as can be seen from Fig. 7, the AAD values for the Integrated service selection approach are higher than Preference-only and Trust-only, when selecting the same number of services. This indicates the Integrated approach a valuable tradeoff between user preference for QoS and the services' trust value. Meanwhile, the AAD values by Preference-only and Trust-only are very close to each other. This observation indicates that, the Preference-only and Trust-only show similar performance when comprehensively consider the user preference and service trust.

In sum, the effectiveness experimental results of different approaches conducted on the same dataset show larger difference. To satisfy a user preference, and select services with good QoS, the integrated service selection approach demonstrates better performance than the other two approaches. On the other hand, the difference for the same approaches conducted on different data set is smaller. This indicates that the synthetic data set approximate the real data well and the experiments based on the synthetic data set is informative. It's important to note that we can also use different approaches (e.g., Preference-only or Trust-only) for different applications. We should analyze the specific application requirement to select the most suitable approach, and the strength and weakness of each approach should be considered, respectively.

4.3 Precision

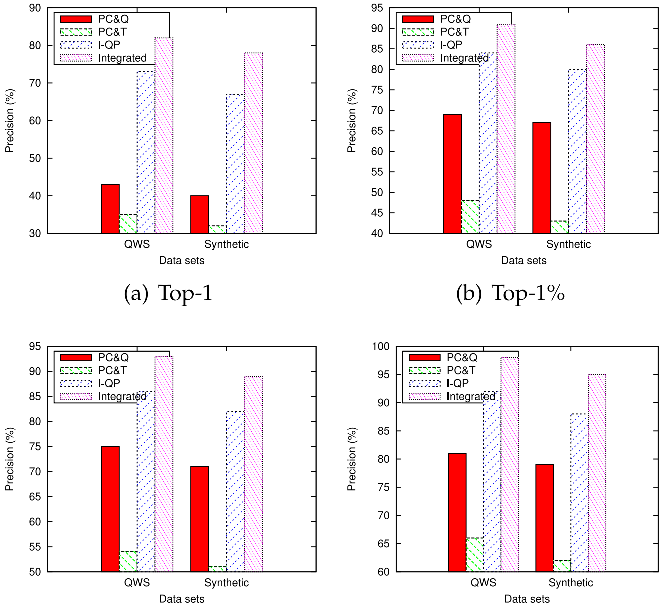
To compare the precision of our method with other existing service selection approaches, we artificially appoint the service selection result for each randomly selected 100 user requirements and obtain the *Baseline* in QWS and Synthetic data set, separately. The fundamental principle for selecting the Baseline services is to choose the optimal (or near-optimal) services with respect to both QoS preference and service trust on the basis of meeting the requirements of users. We compare our approaches with the popular pairwise comparison method [41]. Specifically, the relative importance matrix for quantitative QoS attributes in our datasets is used as the ratings for pairwise comparison. The *Pairwise_comparison* algorithm [41] is implemented to calculate the weights (i.e., Priority weight) for each QoS attribute. Therefore, two approaches for the precision comparison are designed, i.e., Pairwise Comparison and QoS (PC&Q), and Pairwise Comparison and Trust (PC&T). Note that the priority weight is calculated based on the subjective judgement on the relative importance of QoS properties by users.

We calculate the weighting vector W for all candidate services. As for PC&Q, the weighting value for each service



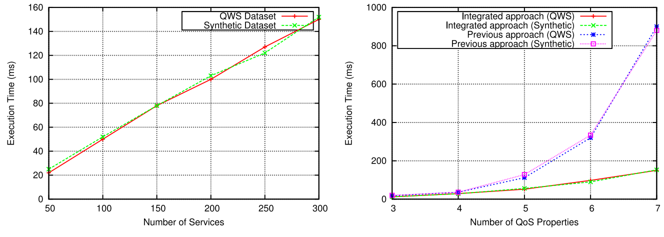
(a) QWS Dataset (b) Synthetic Dataset

Fig. 6. Comparison on confidence degree (CFD).



(a) Top-1 (b) Top-1% (c) Top-3% (d) Top-5%

Fig. 8. Precision comparison with relative existing approaches.



(a) $n = 7$ (b) $m = 300$

Fig. 9. Execution time comparison.

is the cumulative sum of each QoS value multiplied by the priority weight of the corresponding QoS attribute. In PC&T, each service's weighting value is the cumulated sum of the trust value for each QoS attribute multiplied by the priority weight of the corresponding QoS attribute. For each of the selected 100 user's requirements, we first exclude the services not matching the user's QoS constraints. We select top ranked candidates from the remaining services as the service selection result for the user, according to the weighting vector W calculated by PC&Q and PC&T, respectively.

For each of the service selection requirements, we separately select the Top-1, -1, -3, and -5 percent candidate services. We conduct service selection for each of the 100 requirements via PC&Q, PC&T, the proposed integrated method in Section 4.2.1, and the integrated method without the consideration of Qualitative Preference (I-QP), separately. The service selection results by different approaches will be compared with the Baseline. We employ precision to evaluate the user satisfaction of different approaches. The precision is calculated as follows.

$$Precision = (N_{contains\ Baseline}/N) \times 100\%, \tag{22}$$

where $N_{contains\ Baseline}$ represents the number of service selections whose result set contain a Baseline service, and $N = 100$ represents the total time of service selections.

As can been seen from Fig. 8, (1) the precision by the integrated method is slightly higher than I-QP, and it is much higher than PC&Q and PC&T. This is because when the evaluation decision is made for services, although PC&Q and PC&T take the user's subjective preference (i.e., relative importance) and service's objective performance (i.e., QoS and historic users' evaluation for the performance as trust values) together into consideration, the integrated method via multi-objective constrained model considers the above multiple factors, and can also make trade-offs for the conflicts caused by the users' subjective judgments and multiple decision objectives. Moreover, the integrated method can also support the user's qualitative preferences; (2) With the increase of the number of selected services in the service selection result set, the precision increases gradually. Especially when Top-5 percent is included in the result set, the precision by the integrated method approaches 100 percent. This observation implies that, we can also recommend the Top-5 percent services to the required user, and the final selection decision should be manually made by the user. This could provide a promising solution in the service selection application for Cloud industry in the era of Big Data.

4.4 Computational Complexity

To investigate RQ 2, we compared the computational complexity of the proposed Integrated service selection approach with our previous work [42]. The approach in [42] can only support quantitative and qualitative QoS preferences, and the qualitative preference is modeled based on CP-net [37]. The experiments were conducted as follows.

First, we fix the number of QoS properties as $n = 7$ and vary the number of services for from 50 to 300 with a step value of 50. As shown in Fig. 9a, both QWS dataset and Synthetic dataset demonstrate that with the increment of service number, the execution time shows linear growth. Because our proposed Integrated service selection approach need to iterate through all the fixed candidate service set. With the increment of service number, the linear growth of computational complexity demonstrates the good performance. Specially, the execution time of service selection for 300 services under the two types of data sets are all lower than 160 ms. It can indicate the ability of the realtime service selection response by our proposed approach.

Second, we fix the number of services as $m = 300$ and vary the number of QoS properties for from 3 to 7, with a step value of 1. As shown in Fig. 9b, the experimental results based on both QWS dataset and Synthetic dataset indicate that with the increment of QoS property number, the execution time by the previous approach (c.f. [42]) almost shows exponential growth, and the Integrated approach proposed in this paper shows polynomial growth. This observation can prove that, the computational complexity of the Integrated method is relatively lower than the previous approach. It is because we transform the quantitative QoS attributes into qualitative properties by the previous approach. However, the number of service's quantitative QoS properties is always far more than qualitative QoS attributes. The integrated service selection approach proposed in this paper directly handles the quantitative QoS properties. It can effectively reduce the computational complexity. Therefore, the integrated method performs better to deal with the challenge of big volume of service data and is more suitable for BigData-oriented service selection applications.

5 Related Work and Discussion

In this section, we briefly overview several works on service selection that significantly influenced our paper.

Due to the large-scale of service space and diverse user requirement, service selection is an open issue for service computing. Some service selection architectures have been proposed to guarantee effective service selection for Service-based application design and service composition. Ran [43] first extended the traditional SOA architectures to support QoS by proposing a services discovery model. Hadad et al. [23] proposed a QoS-Broker architecture to find the best Web service, which deals with Web service client's query along with QoS requirements to find the best Web service. Benouaret et al. [44] proposed WS-Sky framework for service selection. The framework leverages two variants of the notion of skyline to effectively and efficiently select Web services that can better fit for the user demands.

To support dynamic service composition, service selection with global QoS optimization is a critical issue and widely investigated in the previous works. Gao et al. [19] proposed a trust-oriented QoS-aware service selection approach based on genetic algorithms to optimize the global QoS of service composition. To support effective and efficient service selection for Service-based Systems based on combinatorial auction, the authors in [45] proposed a Combinatorial Auction approach (CASS) for Service Selection.

In practice, multiple users may together desire for services holding the similar function. The work in [46] focus on solving the problem and developing a global optimal service selection method for multiple competitive service requesters.

To select the functionally similar services, the non-functional criteria of atomic services such as QoS are always taken into account. In the last five years, many exiting works have focused on heterogeneous QoS preferences for atomic service selection, for example, QoS-Aware service selection [23], [41], qualitative preferences oriented service selection [47], and the combination of qualitative/quantitative preference for service selection [48].

Specially, Santhanam et al. [47] adopted TCP-net to model qualitative preferences. Pierre Châtel et al. [49] proposed a LCP-net to model linguistic NFPs preference statements for service selection. Schröpfer et al. [48] use UCP-net to model user preferences over the attribute values quantitatively, and they use TCP-net to assign relative importance and conditional relative importance between NFPs qualitatively.

To identify higher trustworthy services, many works have investigated trust and reputation based service selection [34]. The works in [18], [27], [28] focus on defining and developing trust-, reputation-, and referral-based models to perform service selection. For the issue of the selection services blindness, Ye et al. [20] and Vu et al. [50] proposed to use the trust value for QoS to evaluate the real QoS of services and rank the candidate services for service selection. Cui et al. [51] presented a service selection method based on credible user's recommendation on QoS.

To deal with the trust issue for Web service selection under the open environment, more recent and advanced approaches have been proposed [25]. To provide accurate reputation assessment in open environments in which Web services are selfish and utility maximizers, Bentahar et al. [26] investigated the payoffs of different scenarios by focusing on discouraging Web services to act maliciously. In [12], probability theories are adopted to estimate the trustworthiness of Web services by exploiting Dirichlet and generalized Dirichlet distributions. An algorithm is also proposed to deal with malicious feedback and various strategic behaviors which are commonly performed by Web services. Probabilistic models and machine learning techniques are used in [52] to predict which quality class each Web service belongs to. Abedinzadeh and Sadaoui [53] proposed a novel generic Agent Trust Management (ATM) framework ScubAA based on the theory of Human Plausible Reasoning (HPR) for managing trust in an open system. To overcome feedback subjectivity issues, an automated rating model based on expectancy-disconfirmation theory from market science is proposed [11]. Two approaches for trust learning of single and composed services including Bayesian Networks, and Mixture of Multinomial Dirichlet Distributions (MMDD) have been proposed for service selection [54].

Different from the existing works, in this paper, we focus on atomic Cloud service selection which can support user's quantitative and qualitative QoS preference and can guarantee the authenticity of the QoS. In this context, we aims to find a most appreciate Cloud service for a service-based application developer or a self-adaptive service composition, under Big Data service space.

Kim and Doh [55] proposed mechanisms which can select services according to QoS and trust. Specially, the trust values are collected from historic user's feedbacks. Nevertheless, these works do not take qualitative preference and service trust into account. In [27], a two-layered preference-based service selection framework that integrates trust and preference is proposed. Their work is most similar with our work, but we combine trust and preference together instead of dividing service selection into two layers, so that we can minimize computational complexity and can make a combined tradeoff for different influencing factors for effective service selection.

6 Conclusion and Future Work

Cloud service selection which consider both user preference and service poses a number of Big Data challenges that existing approaches fail to address. To this end, this paper proposes a solution that takes into account both qualitative and quantitative user QoS preference with the service trust. To achieve the integrated service selection approach and deal with the heterogeneous information, first, we propose an effective approach to resolve inconsistencies of QoS properties; second, we detail QoS Match Degree in both qualitative and quantitative preferences; last, we define a linear weighting function to rank how each service matches the user's requirements and we estimate the variables of this function through a Multi-objective Constrained Model. We have also carried out an experimental evaluation to understand the impact of the proposed solution on the execution of the application. The experimental results demonstrate that they are more effective than competitive ones.

This work can be extended in the following future directions:

- First, in real life, serval service providers may hire some malicious users to provide very low ratings to other services that provide similar functionalities, while providing very high ratings to their own services. The approaches used for preventing malicious feedbacks in trust and reputation based service selection (e.g., [11], [12], [52]) should be integrated with our method.
- Second, the user profile (similar users) and usage profile (invoked services) should also be taken into account. Some related collaborative filtering approaches can be considered to integrate with our service selection approach.
- Third, our work is under the hypothesis that the users would provide enough information that contains user preference and evaluation of services (the formation of trust information) to help make decisions. An important question is how to promote users to give feedback about services they invoked and how to obtain these information in service selection conveniently and effectively.

ACKNOWLEDGMENTS

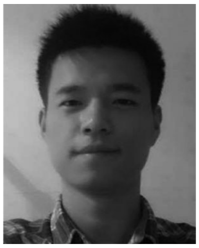
This work was partially supported by National Key Research and Development Plan (No. 2018YFB1003800) and NSFC Projects (Nos. 61672152, 61532013), Collaborative Innovation Centers of Novel Software Technology and Industrialization and Wireless Communications Technology.

REFERENCES

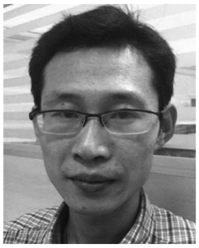
[1] X. Huang, "usageqos: Estimating the QoS of web services through online user communities," *ACM Trans. Web*, vol. 8, no. 1, p. 1, 2013.
[2] Z. Wan, F. J. Meng, J. M. Xu, and P. Wang, "Service composition pattern generation for cloud migration: A graph similarity analysis approach," in *Proc. IEEE Int. Conf. Web Serv.*, 2014, pp. 321–328.
[3] W. Dou, X. Zhang, J. Liu, and J. Chen, "Hiresome-ii: Towards privacy-aware cross-cloud service composition for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 455–466, Feb. 2015.
[4] Z. Ye, A. Bouguettaya, and X. Zhou, "Economic model-driven cloud service composition," *ACM Trans. Internet Technol.*, vol. 14, no. 2-3, pp. 20, 2014.
[5] S. Dustdar, R. Pichler, V. Savenkov, and H.-L. Truong, "Quality-aware service-oriented data integration: Requirements, state of the art and open challenges," *ACM SIGMOD Rec.*, vol. 41, no. 1, pp. 11–19, 2012.
[6] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *J. Netw. Comput. Appl.*, vol. 45, pp. 134–150, 2014.
[7] C. R. Rivero and H. M. Jamil, "Towards a novel model for distributed big data service composition using functional graph matching," in *Proc. IEEE Int. Congress Big Data*, 2014, pp. 794–795.
[8] A. V. Dastjerdi and R. Buyya, "Compatibility-aware cloud service composition under fuzzy preferences of users," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 1–13, 2014.
[9] A. Bouguettaya, S. Nepal, W. Sherchan, X. Zhou, J. Wu, S. Chen, D. Liu, L. Li, H. Wang, and X. Liu, "End-to-end service support for mashups," *IEEE Trans. Serv. Comput.*, vol. 3, no. 3, pp. 250–263, Jul.-Sep. 2010.
[10] Q. Yu, "Sparse functional representation for large scale service clustering," in *Proc. 10th Int. Conf. Serv.-Oriented Comput.*, 2012, pp. 468–483.
[11] N. Limam and R. Boutaba, "Assessing software service quality and trustworthiness at selection time," *IEEE Trans. Softw. Eng.*, vol. 36, no. 4, pp. 559–574, Jul./Aug. 2010.
[12] M. Mehdi, N. Bouguila, and J. Bentahar, "Trust and reputation of web services through qos correlation lens (early access article)," *IEEE Trans. Serv. Comput.*, 2015, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7094317&newsearch=true&queryText=Trust%20and%20Reputation%20of%20Web%20services%20Through%20QoS%20Correlation%20Lens, Doi: 10.1109/TSC.2015.2426185.
[13] Y. Liu, A. H. H. Ngu, and L. Zeng, "Qos computation and policing in dynamic web service selection," in *Proc. 13th Int. Conf. World Wide Web*, 2004, pp. 66–73.
[14] D. A. D'Mello and V. S. Ananthanarayana, "Dynamic selection mechanism for quality of service aware web services," *Enterprise IS*, vol. 4, no. 1, pp. 23–60, 2010.
[15] C.-W. Hang and M. P. Singh, "Trustworthy service selection and composition," *ACM Trans. Autonom. Adaptive Syst.*, vol. 6, no. 1, p. 5, 2011.
[16] L. Li, Y. Wang, et al., "Subjective trust inference in composite services," in *Proc. AAAI*, 2010, pp. 1377–1384.
[17] T. Wu, W. Dou, C. Hu, and J. Chen, "Service mining for trusted service composition in Cross-cloud environment (early access article)," *IEEE Syst. J.*, 2014, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6948332&newsearch=true&queryText=Service%20mining%20for%20trusted%20service%20composition%20in%20Cross-cloud%20environment, Doi : 10.1109/JSYST.2014.2361841.
[18] Z. Noorian, M. Fleming, and S. Marsh, "Preference-oriented Qos-based service discovery with dynamic trust and reputation management," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, 2012, pp. 2014–2021.
[19] H. Gao, J. Yan, and Y. Mu, "Trust-oriented QoS-aware composite service selection based on genetic algorithms," *Concurrency Comput.: Practice Exp.*, vol. 26, no. 2, pp. 500–515, 2014.
[20] B. Ye, A. Pervez, M. Ghavami, and M. Nekovee, "A Trust-based model for quality of web service," in *Proc. 5th Int. Conf. Adv. Serv. Comput.*, 2013, pp. 39–45.
[21] M. Palmonari, M. Comerio, and F. De Paoli, "Effective and flexible Nfp-based ranking of web services," in *Proc. 7th Int. Joint Conf. Serv.-Oriented Comput.*, 2009, pp. 546–560.
[22] S. Lamparter, A. Ankolekar, R. Studer, and S. Grimm, "Preference-based selection of highly configurable web services," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 1013–1022.
[23] J. E. Hadad, M. Manouvrier, and M. Rukoz, "TQOS: Transactional and QoS-aware selection algorithm for automatic web service composition," *IEEE Trans. Serv. Comput.*, vol. 3, no. 1, pp. 73–85, Jan.-Mar. 2010.
[24] L. Zeng, B. Benatallah, A. 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.
[25] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "A survey on trust and reputation models for web services: Single, composite, and communities," *Decision Support Syst.*, vol. 74, pp. 121–134, 2015.
[26] J. Bentahar, B. Khosravifar, M. A. Serhani, and M. Alishahi, "On the analysis of reputation for agent-based web services," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12 438–12 450, 2012.
[27] C. Hang and M. Singh, "Selecting trustworthy service in Service-oriented environments," in *Proc. 12th AAMAS Workshop Trust Agent Societies*, 2009, pp. 1–12.
[28] E. Maximilien and M. Singh, "Toward autonomic web services trust and selection," in *Proc. 2nd Int. Conf. Serv. Oriented Comput.*, 2004, pp. 212–221.
[29] G. Liu, Y. Wang, M. A. Orgun, and E.-P. Lim, "Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks," *IEEE Trans. Serv. Comput.*, vol. 6, no. 2, pp. 152–167, Apr. 2013.
[30] Y. Wang and J. Vassileva, "Toward trust and reputation based web service selection: A survey," *Int. Trans. Syst. Sci. Appl.*, vol. 3, no. 2, pp. 118–132, 2007.
[31] H. Wang, L. Wang, and C. Yu, "Integrating trust with qualitative and quantitative preference for service selection," in *Proc. IEEE Int. Conf. Serv. Comput.*, 2014, pp. 299–306.

[32] R. L. Keeney, "Utility independence and preferences for multiattributed consequences," *Oper. Res.*, vol. 19, no. 4, pp. 875–893, 1971.
[33] P. C. Fishburn, *Utility Theory for Decision Making*. New York, NY, USA: Wiley, 1970.
[34] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.
[35] C. Hang, A. K. Kalia, and M. P. Singh, "Behind the curtain: Service selection via trust in composite services," in *Proc. IEEE 19th Int. Conf. Web Serv.*, 2012, pp. 9–16.
[36] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Serv. Sci.*, vol. 1, no. 1, pp. 83–98, 2008.
[37] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole, "Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements," *J. Artif. Intell. Res.*, vol. 21, pp. 135–191, 2004.
[38] H. Wang, S. Shao, X. Zhou, C. Wan, and A. Bouguettaya, "Web service selection with incomplete or inconsistent user preferences," in *Proc. 7th Int. Joint Conf., ICSOC-ServiceWave*, 2009, pp. 83–98.
[39] R. I. Brafman, C. Domshlak, and S. E. Shimony, "On graphical modeling of preference and importance," *J. Artif. Intell. Res.*, vol. 25, pp. 389–424, 2006.
[40] H. Mukhtar, D. Belaïd, and G. Bernard, "A quantitative model for user preferences based on qualitative specifications," in *Proc. Int. Conf. Pervasive Serv.*, 2009, pp. 179–188.
[41] C. Choi and H. Jeong, "A Broker-based quality evaluation system for service selection according to the QoS preferences of users," *Inf. Sci.*, vol. 277, pp. 553–566, 2014.
[42] H. Wang and W. Liu, "Web service selection with quantitative and qualitative user preferences," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.-Vol. 01*, 2011, pp. 404–411.
[43] S. Ran, "A model for web services discovery with qos," *SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.
[44] K. Benouaret, D. Benslimane, and A. Hadjali, "Ws-sky: An efficient and flexible framework for QoS-aware web service selection," in *Proc. IEEE 9th Int. Conf. Serv. Comput.*, 2012, pp. 146–153.
[45] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction," *IEEE Trans. Softw. Eng.*, vol. 40, no. 2, pp. 192–215, 2014.
[46] G. Kang, J. Liu, M. Tang, X. Liu, and K. K. Fletcher, "Web service selection for resolving conflicting service requests," in *Proc. IEEE Int. Conf. Web Serv.*, 2011, pp. 387–394.
[47] G. R. Santhanam, S. Basu, and V. Honavar, "Representing and reasoning with qualitative preferences for compositional systems," *J. Artif. Intell. Res.*, vol. 42, pp. 211–274, 2011.
[48] C. Schröpfer, M. Binshtok, S. E. Shimony, A. Dayan, R. Brafman, P. Offermann, and O. Holschke, "Introducing preferences over nfps into service selection in SOA," in *Proc. Serv.-Oriented Comput. Workshops*, 2009, pp. 68–79.
[49] P. Châtel, I. Truck, and J. Malenfant, "LCP-nets: A linguistic approach for Non-functional preferences in a semantic SOA environment," *J. UCS*, vol. 16, no. 1, pp. 198–217, 2010.
[50] L.-H. Vu, M. Hauswirth, and K. Aberer, "Qos-based service selection and ranking with trust and reputation management," in *Proc. Confederated Int. Conf. On the Move Meaningful Internet Syst.: CoopIS, DOA, and ODBASE*, 2005, pp. 466–483.
[51] Y. Cui, C. Chen, and Z. Zhao, "Web service selection based on credible user recommended and QoS," in *Proc. IEEE/ACIS 11th Int. Conf. Comput. Inf. Sci.*, 2012, pp. 637–642.
[52] M. Mehdi, N. Bouguila, and J. Bentahar, "Probabilistic approach for QoS-aware recommender system for trustworthy web service selection," *Appl. Intell.*, vol. 41, no. 2, pp. 503–524, 2014.
[53] S. Abedinzadeh and S. Sadaoui, "A Trust-based service suggestion system using human plausible reasoning," *Appl. Intell.*, vol. 41, no. 1, pp. 55–75, 2014.
[54] M. Mehdi, N. Bouguila, and J. Bentahar, "Trustworthy web service selection using probabilistic models," in *Proc. IEEE 19th Int. Conf. Web Serv.*, 2012, pp. 17–24.
[55] Y. Kim and K.-G. Doh, "A trust management model for QoS-based service selection," in *Proc. Inf. Security Appl.*, 2012, pp. 345–357.

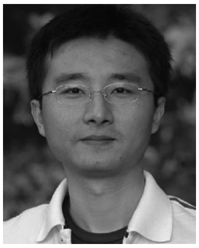
Hongbing Wang received the PhD degree in computer science from Nanjing University, China. He is a full professor from the School of Computer Science and Engineering, Southeast University, China. He served as an visiting scientist at CSIRO ICT Center, Australia, from April 2009 to March 2010. Prior to this, he has visited the Hongkong University and the Waterloo University during 2003 to 2008. His research interests include service computing, cloud computing, and software engineering. He published more than 50 refereed papers in international Journals and conferences, e.g., *Journal of Web Semantics*, *Journal of Systems and Software*, *IEEE Transactions on parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, ICSOC, ICWS, SCC, etc. He is a member of the IEEE.



Chao Yu received the MS degree in the School of Computer Science and Engineering, Southeast University, P.R. China, in 2014. He is currently working as a software engineer in dianping.com at Shanghai, China. His research interest mainly focuses on service selection.



Lei Wang is currently working toward the PhD degree in the School of Computer Science and Engineering, Southeast University, China. He is also a lecturer in Nanjing Forestry University, China. His research interests include service computing, software reliability engineering, and data mining. His publications have appeared in international Journals and popular conferences, e.g., *IEEE Transactions on parallel and Distributed Systems*, ICSOC, ICWS, SCC, etc.



Qi Yu received the PhD degree in computer science from the Virginia Polytechnic Institute and State University (Virginia Tech). He is an associate professor in the College of Computing and Information Sciences at the Rochester Institute of Technology. His current research interests lie in the areas of service computing, databases, and data mining. His publications have mainly appeared in well-known journals (e.g., the *VLDB* Journal, the *ACM Transactions on the Web*, the *World Wide Web Journal*, and the *IEEE Transactions on Services Computing*) and conference proceedings (e.g., ICSOC and ICWS). He is a member of the IEEE.