

Collaboration Reputation for Trustworthy Web Service Selection in Social Networks

Shanguang Wang¹, Lin Huang¹, Ching-Hsien Hsu², Fangchun Yang¹

¹*State Key Laboratory of Networking and Switching Technology*

¹*Beijing University of Posts and Telecommunications, Beijing, China*

²*Department of Computer Science and Information Engineering,*

²*Chung Hua University*

¹*Beijing 100876, China*

²*Hsinchu 707, Taiwan*

sgwang@bupt.edu.cn; linhuang.bupt@gmail.com; chh@chu.edu.tw; fcyang@bupt.edu.cn

Abstract: Traditional trustworthy service selection approaches focus the overall reputation maximization of all selected services in social networks. However, the selected services barely interact with each other in history, which leads to the trustworthiness among services very low. Hence, to enhance the trustworthiness of Web service selection, a novel concept, collaboration reputation is proposed in this paper. The collaboration reputation is built on a Web service collaboration network consists of two metrics. One metric, invoking reputation can be calculated according to other service's recommendation. The other metric, invoked reputation can be assessed by the interaction frequency among Web services. Finally, based on the collaboration reputation, we present a trustworthy Web service selection method to not only solve the simple Web service selection but also the complex selection. Experimental results shown that compared with other methods, the efficiency of our method and the solution's trustworthiness are both superior increased.

Keywords: Web service; Service selection; Trustworthy; Reputation; Collaboration reputation

1 Introduction

Social networks have recently received much attention on the mobile Internet. For example, YouTube, MySpace and Facebook are among the most popular social network sites, and continue to experience explosive growth both in terms of the number of communities and the overall population [1]. The social networks so constructed provide a powerful means for users to share, organize and locate interesting services such as mobile APP, open API, Web services.

As a large number of functionally equivalent (or similar) Web services have been built and deployed, customers face a difficult task in choosing the best service to build their composite service which satisfies their personalized Quality-of-Services (QoS) requirements. Thus, the efficiency associated with selecting a Web service with a QoS guarantee has increasingly become a critical issue in the Web service selection process [2].

One of major problems in the Web service selection process is that QoS cannot reflect the real situation of Web services because the dynamic environment imposes a stochastic nature on Web services. Some enhanced QoS measurement algorithms [3] have been proposed to eliminate the

uncertainty. However, in actual practice, some Web service providers may intentionally exaggerate their QoS values, and QoS measurement cannot reveal this sort of malicious deception. To address this problem, trustworthy Web service selection is needed, which assigns high level reputation values to different Web services. Then the performance of selected Web services can be guaranteed based on each reputation value (score).

The reputation represents a collective perception of the users in the social network about a Web service. The reputation of a invoked Web service is a collective feedback rating of the users that have interacted with or used the service in the past [4]. Accurate reputation measurement about Web services in social networks plays an important role in identifying good nodes and connections. Hence, the ability to obtain an accurate reputation score of each Web service within a large social network structures is also important [5].

Unfortunately, we found most Web service methods only rely on the reputation value of individual Web service. In such case, the total reputation of all selected Web services which composite a new value-added service (i.e. composite service), is maximized, but the total trustworthiness among services is very low. Why? For a service, if its reputation value is very high, it will be selected with higher probability than other services with low reputation. However, in this composite service, there is a selected service that had little or no interaction with other Web services. Then when one service invokes the service for a new task or instance, the trustworthiness between the two services is less than the average reputation of the two services. If each of the two services has a high reputation value, but they barely interact with each other, which shorten the total trustworthiness of the composite service because of unknown interact risks.

Hence, in such cases, the trustworthiness of Web service selection may not be the highest, i.e., we let the best selection scheme slip away. What do we do?

The answer may be collaboration, which in this paper denotes the invocation collaborative relationship of Web services, including invoking and invoked relationships. The invoking relationship means that one service invoked other services, and the invoked relationship means that one service is invoked by other services in this paper. Actually, we find that the collaboration among Web services can provide a good prospect for trustworthy Web service selection, and it should be taken into consideration actively. Moreover, the collaboration does not only consider the reputation of individual service, but also pay more attention to the intimate relationship of between multiple services. Therefore, we think that an ideal trustworthy Web service selection approach should be able to exclude the Web services with low reputation by collaborating with other services and provide the trustworthy Web service selection for composite services.

In this paper, based on our previous work [6], we aim to propose a trustworthy Web service selection approach that does not only consider individual Web service reputation but also the collaboration reputation of Web services. The main contributions of our work include:

To support collaboration reputation, we first construct a Web service collaboration network (WSCN) to eliminate the Web services with low reputation from the WSCN using our proposed neighbor update strategy and then divide normal Web services into different Web service community using community detection.

To avoid subjective reputation measurement, based on WSCN, we propose a novel concept, collaboration reputation, which is evaluated by invoking reputation and invoked reputation. Invoking reputation is used to evaluate the performance experience of a Web service, can be calculated according to other services' recommendation in the Web service community. Invoked reputation is used to

evaluated the performance importance of a Web service, can be obtained according to the interaction frequency between the invoking Web service (which means it invoked other services) and invoked Web services (which means it was invoked by other services) in the Web service community.

Finally, based on the collaboration reputation, we present a trustworthy Web service selection method. This method is to not only solve the simple Web service selection but also the complex selection. We conduct extensive experiments to evaluate the effectiveness of our approach. The experimental results reveal that our approach not only increases the trustworthiness of Web service selection but also improve the efficiency.

The paper is organized as follows. To begin with, it presents related work on Web service selection in Section 2, and then introduces a framework for trustworthy Web service selection in Section 3 and the WSCN is constructed in Sections 4. Furthermore, we propose a novel concept about collaboration reputation for Web service selection in Section 5. Based on the collaboration reputation, we show the trustworthy Web service selection approach in Section 6. Finally we conduct experiments in Section 7 and conclude the paper in Section 8.

2 Related Work

A number of researchers have recognized the importance of reputation in Web service selection, and many states of art solution have been proposed. They adopted different techniques in different aspects to establish the trustworthiness of Web services or service selections.

Wang and Vassileva [7] discussed trust QoS as applied to web service selection and presented further research directions. Vu et al. [8] collected users' reports on QoS to rank and select Web services based on past QoS data predictions. Yau et al. [9] identified the deviation between the QoS provided by their service providers and the QoS values determined by monitors and service user feedbacks to improve the trustworthiness of the QoS information. The method might result in a false rating when the user's feedback is taken into account. In addition, fuzzy theory has been applied to enhance the QoS trust. Manchala [10] proposed a fuzzy matrix that is defined based on the transaction history to establish transaction trust. Nepal et al. [11] added a query model and underlying data to the fuzzy trust management framework, which represents and queries customer perception. Alfaro LD et al. [12] concluded on a note of optimism concerning the role of reputation systems in mediating online collaboration, and given important references of design and optimization of reputation systems. McNally K et al. [13] proposed a good approach to modeling user and item reputation in social recommender systems and it is more efficient than other approaches.

In addition to fuzzy theory, Bayesian networks are often introduced into the trust evaluation process. Wu et al. [14] used Bayesian networks to model a consumer's assessment of a service QoS. Their approach allows consumers to combine different QoS attributes. Hang et al. [15] developed a Bayesian network that can punish and reward services in terms of QoS property accurately with incomplete observations, so that consumers can prevent themselves from interacting services with unsatisfying QoS. In general, all these mentioned methods merely evaluate Web service individually and not in collaboration with other services. To address the collaboration trust, Elnaffar and Khosravifar et al. [16; 17] proposed a framework aiming to select trusted Web service in a community, which is collection of Web service with a common functionality. The collaboration definition in our paper is different than these previous studies. We focus on the invocation collaborative relationship, which includes invoking and invoked relationships.

In fact, the invoking relationship can be considered as collaboration among Web services combined to provide output parameters. To assess the invoking reputation, it depends on community structure in the WSCN. The vertex's importance analysis has been studied in community structure [18]. In addition, the community structure has been applied in the Web service field. Kil [19] used a real-world dataset to analyze the topological landscape of Web service's networks and concluded that the network exhibited small world network and power-law-like structure to some extent. Zhang et al. [20] analyzed the logs of an execution engine to elucidate the Web service community and combined the closely interactive Web services using a composition process. Ji et al. [21] presented a novel Web service management method based on collaboration networks, where the network is undirected and has a weighted edge. In addition, the researchers introduced some metrics to reflect the Web service properties. As distinguished from the aforementioned WSCN construction methods, the community detected in our WSCN can effectively reflect the collaborative relationship. Our community detection strategy is more suitable for collaboration reputation evaluation for trustworthy Web service selection.

3 Trustworthy Web Service Selection Framework

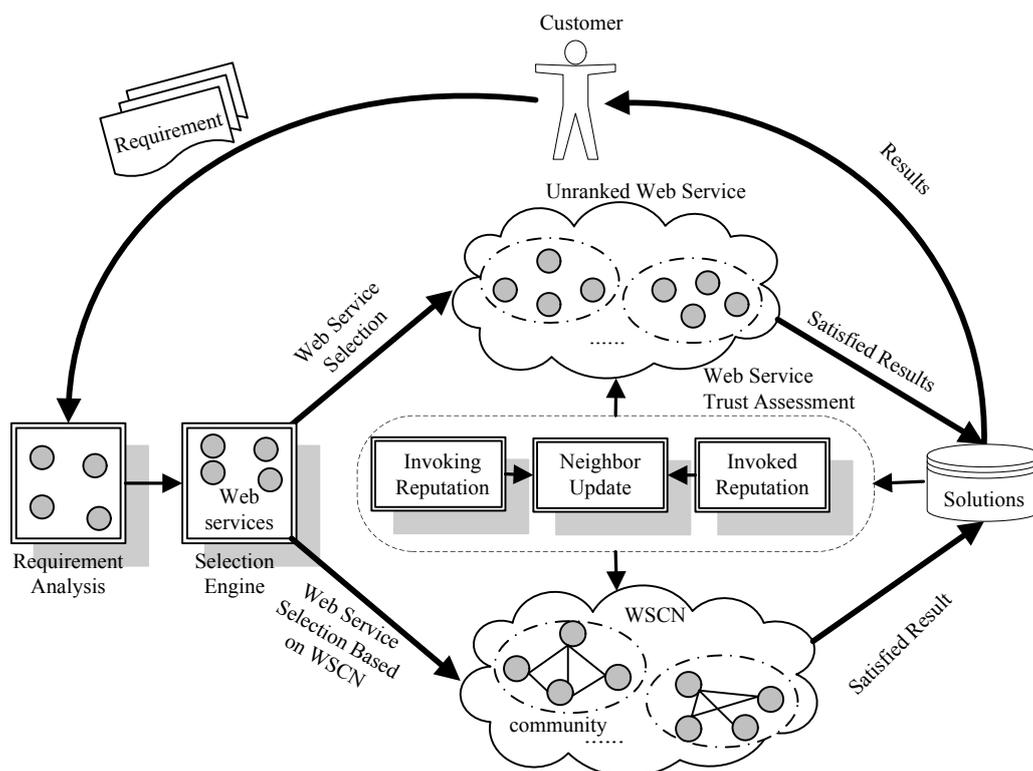


Figure 1 The framework of the proposed trustworthy Web service selection

As shown in Figure 1, we present a framework which is designed to maintain the WSCN and Web services' invoking and invoked relationships. To obtain the individual and collaboration reputation, the framework also provides a component to evaluate the reputation metric and execute the Web service selection process based on collaboration reputation in the WSCN. The framework contains four components. The details of this framework are as follows:

1) **Requirement Analysis.** In this component, the input, output and QoS requirements can be extracted from user requirements. In addition, the Web service that is specified by customers is also

picked out.

2) **Web Service Management.** In this component, there are two types of Web service registration. One type manages the newly arrived Web services, which have no invoking and invoked reputation, and the eliminated neighboring Web services (the Web service with low reputation will be eliminated). The other registration maintains the Web services that appear in the WSCN. The Web services are in a collaborative relationship, including the invoking relationship, invoked relationship and neighbor relationship.

3) **Web Service Selection Engine.** In this component, if a Web service is registered in the WSCN, it will be selected until it becomes unavailable in the WSCN. However, if a Web service is suspected to be low reputation, it will be eliminated from among normal Web services' neighbors. If a Web service is a new arrival and not registered in the WSCN, it will wait for the chance to be selected. It is possible that the new arrival can replace the Web service registering the WSCN because it is deemed as a bad service.

4) **Web Service Trust Assessment.** This component analyzes the execution log in a period of time $[t_1, t_n]$, computes the collaboration reputation and eliminates Web services with low reputation and Web services' neighbors with low reputation. In addition, communities in each time interval $[t_i, t_{i+1}] \in [t_1, t_n]$ are detected, and the trust recommender Web service is selected to compute the invoking reputation.

4 Web Service Collaboration Network (WSCN)

In this section, for understanding WSCN, we will introduce some WSCN definitions.

Definition 1 (Vertex). A vertex v in WSCN represents a Web service ws . Then a fully connected (Web service) graph $v = (C_1, \dots, C_n)$ can be abstracted where each v has 2-tuples, i.e., $\{N, R\}$.

$N = \{Neib_i, Neib_o, Neib_c\}$ is the set of neighbors in the WSCN, including three types. $Neib_i = \{ws_{i1}, ws_{i2}, \dots, ws_{in}\}$ denotes the input neighbors. $Neib_o = \{ws_{o1}, ws_{o2}, \dots, ws_{on}\}$ denotes the output neighbors, and $Neib_c = \{ws_{c1}, ws_{c2}, \dots, ws_{cn}\}$ denotes the invoking neighbors, which are combined to provide the output parameters for satisfying $ws \in Neib_o$.

Definition 2 (Invoking reputation). If $R = \{R_r, R_i\}$ is the set of reputation metrics, and then R_r denotes invoking reputation that can be obtained from the recommendation where the recommendation depends on the trust recommendation vertex (TRV) in the community structure of WSCN.

Definition 3 (Invoked reputation). If $R = \{R_r, R_i\}$ is the set of reputation metrics, and then R_i denotes invoked reputation, which can be evaluated by the invocation frequency between $ws \in Neib_c$ and $ws \in Neib_o$.

Definition 4 (Collaboration reputation). If $R = \{R_r, R_i\}$ is the set of reputation metrics, and then the collaboration reputation (CR) can be evaluated by R_i and R_r .

Definition 5 (Edge). An edge in the WSCN describes the Web service collaboration relationship.

The collaboration relationship is similar to that in a scientific collaboration for a co-authored paper. A Web service in a composite Web service is similar to a paper author. Web services working together for a composite service can be considered similar to authors working together to produce a co-authored paper. Thus, they should be connected. The common composite Web service includes four types: sequence, concurrency, conditional and loop. Hence, a basic WSCN is depicted as shown in Fig. 2.

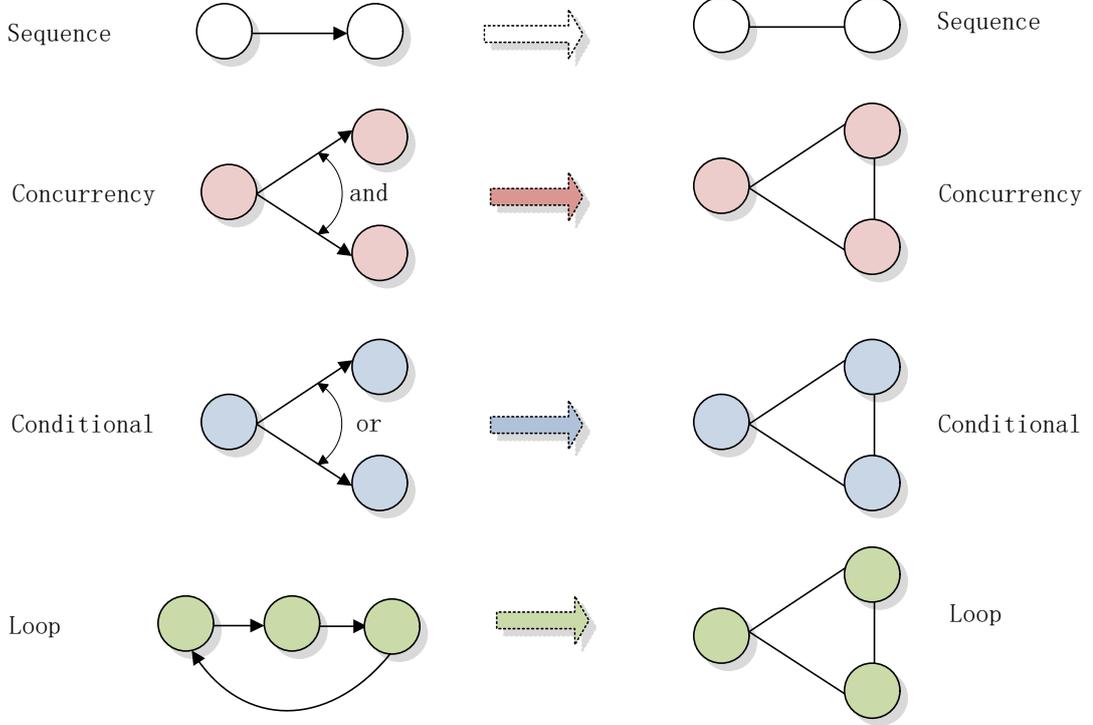


Figure 2 The common structure in a WSCN

Definition 6 (WSCN). WSCN is an undirected graph and its construct will change in the time interval $[t_1, t_2], \dots, [t_{n-1}, t_n]$. WSCN contains the collaboration relationships among Web services in the specific time interval $[t_i, t_{i+1}]$.

4.1 Neighbor Update Strategy in the WSCN

To avoid a Web service with high reputation (we called it as good service in this paper) interacting with a service with low reputation (we called it as bad service in this paper) in the WSCN, any Web service with low reputation will be eliminated from among the Web service's $\{Neib_i, Neib_o, Neib_c\}$. The Web service with low reputation can be detected with the proposed update strategy. The update strategy is based on the following two conditions:

- **Condition 1.** The invocation frequency of a Web service (ws_b) dramatically decreases in the k sequential time interval.
- **Condition 2.** $R_r^{[t_i, t_j]}(ws_b)$ is less than $R_r^{[t_i, t_j]}(ws_n)$, where ws_n and ws_b represent the normal and bad Web service, respectively.

Although R_i contributes to deciding bad Web services, a malicious rater could repeatedly submit

the same composite Web service to improve the R_i level. Thus, when $R_r^{[t_i, t_j]}(ws_b)$ is less than $R_r^{[t_i, t_j]}(ws_n)$, this reveals that ws_b receives fewer recommendation values R_r than that of ws_n in $[t_k, t_{k+1}]$ after k sequential time interval. In addition, the same functional Web services ws_n and ws_b appearing in the WSCN indirectly reflect that ws_b is most likely replaced by ws_n . Then, combining with the above two conditions, we can eliminate the fake Web services from $Neib_i, Neib_o, Neib_c$ in a specific time.

To eliminate the bad service ws_b from the WSCN, we add a gossip algorithm [22] in the WSCN to inform the Web service that has similar functional properties to ws_n to remove ws_b from its neighbor set. Why not use a flooding algorithm? In this case, a flooding algorithm cannot be adopted, as it will increase the system load. To ensure system stability, we adopt a gossip algorithm to spread the message. Gossip algorithms are also called epidemic algorithms. A series of studies have demonstrated that epidemics will spread throughout the network under certain conditions. Therefore, many works [22; 23] adopt such a mechanism for network information dissemination and collection.

To avoid blindly spreading a message, the message forwarding probability in our gossip mechanism is related to weight w_i , where w_i determines the number of neighbors that receive message as follows:

$$w_i = \frac{1}{K} \sum_{i=1}^{|K|} \deg(ws_i) \quad (1)$$

where ws_i is the Web service that receives a message from ws_j ($ws_j \in Neib_i$); $\deg(ws_i)$ denotes the matched number of ws_i output interfaces and ws_j input interfaces; K denotes the number of Web services that match interfaces with ws_j . In our update strategy, if the matched interface number of ws_i is no less than w_i , then there will be a chance to forward messages.

Then the condition that ws_h ($ws_h \in Neib_o$) will forward a message from ws_i is that the matched interface between ws_i and ws_h is similar to that of ws_i and ws_j . Any Web service that forwards message from ws_i must obey this condition. Thus, the number of Web services that forward a message is no greater than the number of Web services that satisfies the minimum $\deg(ws_i)$ in the WSCN, which does not increase the system load.

4.2 Web Service Community Detection

According to the WSCN, the invoking reputation R_r can be computed by some trusted Web service. However, blindly choosing some Web service as recommendations in the WSCN might add some risk to the assessment's correctness. Moreover, this will impose additional load for sending or receiving messages. To overcome the above problem, we adopt the community detection algorithm (CNM) [23] to detect community. The CNM is a new algorithm for inferring community structure from network topology which works by greedily optimizing the modularity. It runs in time $O(md \log n)$ for a network with n vertices and m edges where d is the depth of the dendrogram. If the network is hierarchical, there are communities at many scales and the dendrogram is roughly balanced, it has $d \sim \log n$. If the network is also sparse, $m \sim n$, then the running time is essentially linear, $O(n \log^2 n)$. The CNM is

considerably faster than most previous general algorithms, and allows users to extend community structure analysis to networks that had been considered too large to be tractable. Hence, the community detection strategy can be applied to find the cooperating Web service group, and the invoking reputation of a Web service can be evaluated from TRV, which denotes a vertex having intra-connection to other communities.

The CNM is a condensation algorithm, and its basic idea is to combine communities until Q reaches a maximum. The modularity Q is defined by the following equation:

$$Q = \sum_i (e_{ii} - a_i^2) \quad (2)$$

Where e_{ij} the fraction of edges that join vertices in community i ; a_i is the fraction of ends of edges that are attached vertices in community i .

The operation of the algorithm involves finding the changes in Q that would result from the amalgamation of each pair of communities, choosing the largest of them. The changes in Q , i.e., an incremental matrix ΔQ_{ij} ensures Eq. 2 has fast convergence to the maximum Q . The initial ΔQ_{ij} can be obtained with the following equation:

$$\Delta Q_{ij} = \begin{cases} \frac{1}{2m} - \frac{k_i k_j}{2m^2} & \text{if } i, j \text{ are connected,} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

Where k_i and k_j denotes the degree of community i and j , respectively; m is the edge number of a whole WSCN.

In each iteration, the CNM algorithm combines two maximum ΔQ of community C_i and C_j , and the ΔQ_{ij} is computed after merging C_i and C_j via the following equation:

$$\Delta Q_{ij} = \begin{cases} \Delta Q_{ik} + \Delta Q_{jk} & k \text{ connect to both } i \text{ and } j \\ \Delta Q_{ik} - 2a_j a_k & k \text{ connect } i \text{ but not to } j \\ \Delta Q_{jk} - 2a_i a_k & k \text{ connect } j \text{ but not to } i \end{cases} \quad (4)$$

With

$$a_i = \frac{k_i}{2m}$$

Where after C_i is merged to C_j , $a_i = 0$ while $a_j = a_i + a_j$; when the ΔQ_{ij} value becomes negative, the iteration combination process stops.

As shown in Algorithm 1, the CNM algorithm can perform community detection in the WSCN.

Algorithm 1 Community Detection

Input: $WSCN, ws_1, ws_2, \dots, ws_n$ **Output:** *communities*

1. **Foreach** ws in $WSCN$
 2. set a_i ;
 3. **End For**
 4. **While** ($\Delta Q > 0$)
 5. select the largest ΔQ_{ij} in matrix ΔQ ;
 6. newComm_j = Merge(i, j);
 7. update the j th row and the j th column in matrix ΔQ ;
 8. update a_i and a_j ;
 9. delete the elements of both the i th row and the i th column;
 10. return(i, j);
 11. **End While**
-

By using the above community detection, we can identify a collaboration community among Web services. In Algorithm 1, the operation cost is Θm , and m denotes the iteration number, which is a constant. Thus, the algorithm's time complexity is still $\Theta(n \log^2 n)$.

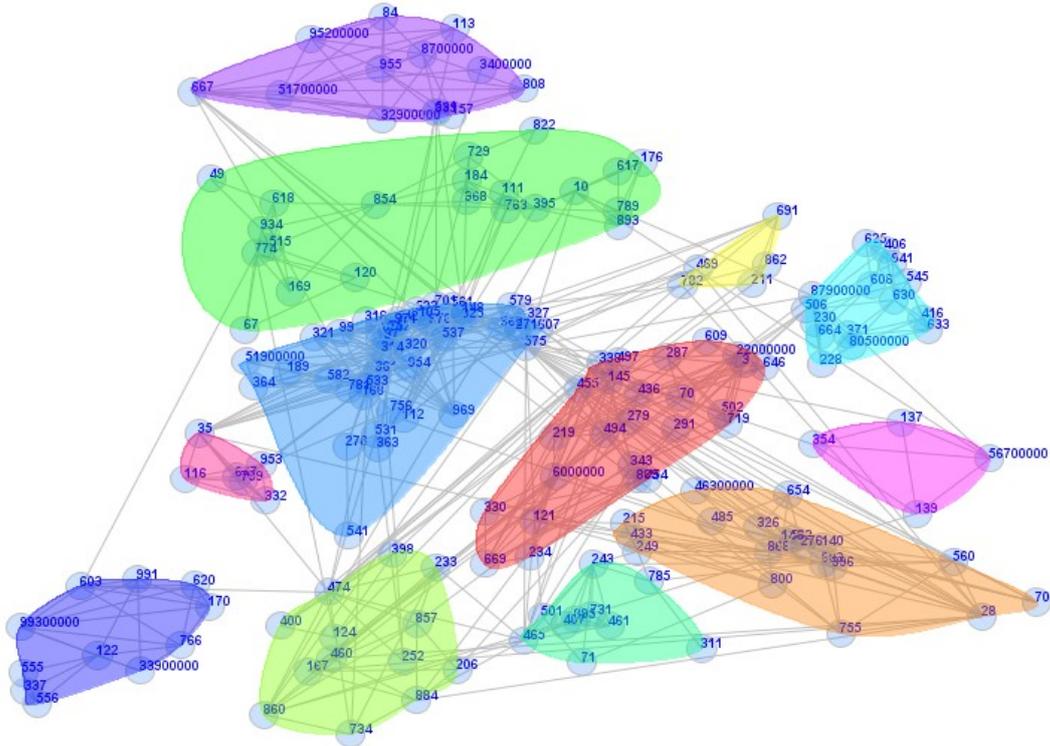


Figure 3 An example of Web service collaboration network

As shown in Fig. 3, in this paper, the Web service collaboration community can be generated from a WSCN in $[t_i, t_{i+1}]$, and the setting and determination of other Web service-related parameters will be illustrated in the evaluation section. In Fig. 3, the Web services are generated by WSBen [24], which is inspired by extensive studies on real Web services to support various Web service network topologies and distributions. We use NeSVA [25] to present the community structure in WSCN. As shown in Fig. 3, a vertex with 00000 appended at the end of its tag denotes a deceptive Web service that publishes fake QoS information. After a time interval $[t_i, t_{i+1}]$, some bad Web services have a few connections to

other Web services, though some are still divided in the community with many good Web services. In this figure, the outbound degree of the good Web service is twice as high as that of the bad Web service. The final result of the bad Web service will be divided into a community with few and new Web services. In our reputation measurement, a Web service's invoking reputation is related to the number of TRV. Thus, the recommender number will result in the distinction between bad and good services after a time interval.

5 Collaboration Reputation

Web services are located in open, distributed environments, and there is an underlying collaborative relationship among them. Hence, the collaboration reputation based on the WSCN provides us a novel method to measure the reputation of Web service for trustworthy Web service selection, including two types: invoking reputation and invoked reputation.

5.1 Invoking Reputation

As we mentioned above, repeatedly submitting the same composite Web service can improve the invoked reputation level, but the recommenders simply limit their submission to specific services. A Web service that joins various compositions will result in more recommenders. Apparently, there are shortcomings in reputation measurements that merely depend on the invoked reputation. Thus, the invoking reputation plays an important role in the reputation measurement and act as a significant metric to evaluate how important a Web service is. In this section, we will examine the invoking reputation in terms of the collaboration community in the WSCN.

The community is characterized by dense connections within the community but sparse connections among communities. A vertex that has more connections to a vertex located in other communities will have higher trustworthiness. Therefore, a vertex in community C , which is more closely related to the TRV in the same community C has a higher invoking reputation. Let $R_r^{[t_1, t_n]}$ be the invoking reputation value during the time interval $[t_1, t_n]$, and then it is computed by the following equation:

$$R_r^{[t_1, t_n]} = \sum_{i=1}^n (\Delta T_i \sum_{k \in TRV_C^{[t_i, t_{i+1}]}} \frac{Norm_Q(k)}{Dist_k + 1}) \quad (5)$$

with

$$\Delta T_i = \frac{\frac{1}{T_i}}{\sum_{i=1}^n \frac{1}{T_i}}$$

Where $Dist_k$ is the shortest path from Web service ws_i to ws_k ($ws_k \in TRV_C^{[t_i, t_{i+1}]}$ and ws_i, ws_k belongs to same community C); ΔT_i ($\Delta T_i < \Delta T_{i+1}$ because more recent trust recommendations are more persuasive) is the time weight of $T_i = [t_i, t_{i+1}]$ ($[T_1, T_2, \dots, T_n]$ are time intervals in $[t_1, t_n]$); $Norm_Q(k)$ is the importance evaluation for a ws and is defined by the following equation:

$$Norm_Q = \frac{\sum_{i=1}^n QoS_i(ws)}{n} \quad (6)$$

Where $QoS_i(ws)$ denotes the value of the common QoS attributes of web services, such as reliability and availability.

According to $R_r^{[t_1, t_n]}$, the Web service that has fewer chances to connect to $TRV_C^{[t_1, t_n]}$ will have a lower invoking reputation level. Thus, a Web service that hardly appears in the invocation logs will not achieve a higher invoking reputation.

5.2 Invoked Reputation

Although invoked reputation results can be misleading, the metric can be used to assess the interaction frequency between invoking and invoked web services. In this paper, we measure invoked reputation by the PageRank algorithm [26] which is the classic algorithm used to assess a page's importance. The algorithm propagates the importance from one Web page to others until the iterative process converges. The importance of the page p is defined by the following equation:

$$PR(p) = \frac{1-d}{N} + d \sum_{j \in B} \frac{PR(j)}{F_p} \quad (7)$$

Where $d(d \in [0,1])$ is a scale factor, which determines ratio between the self-importance of a page and importance obtained from the other linking pages; N is the Web page set; F_p denotes the degree of outbound linking to that page p , and B are the pages connecting to p .

In the above equation, $PR(p)$, the value can converge after several iterations, whereas the importance of a Web service subsequently transmits at the end. Thus, this process results in the invoked reputation of a frontal Web service to be less than that of a Web service in the back of a composite Web service. For simplicity, we consider the importance propagation to its directed neighbor. Thus, the invoked reputation can be obtained as follows:

$$R_i^{[t_1, t_n]} = (1-d)Norm_Q + \sum_{i=1}^n (d \sum_{j=1}^n W_{ij} \Delta T_i Norm_Q) \quad (8)$$

Where d is a scale factor, which is the same as the original PageRank algorithm; $Norm_Q$ is the assessment of Web the service's inherent importance, which is mentioned in Eq. 6; W_{ij} denotes the ratio the matched interface number and the whole interface number in the invoked service (ws_i -invoking- ws_j often involves a matched interface, and thus, the importance propagation between ws_i and ws_j is related to the matched interface number in once invocation); ΔT_i ($\Delta T_i < \Delta T_{i+1}$ because more recent trust recommendations are more persuasive) is the time weight of $T_i = [t_i, t_{i+1}]$ ($[T_1, T_2, \dots, T_n]$ are time intervals in $[t_1, t_n]$); $R_i = \{R_{ij1}, R_{ij2}, \dots, R_{ijn}\}$ represents the invoked reputation R_i that accumulates from Web service $ws_{j1}, ws_{j2}, \dots, ws_{jn}$ during T_n , and in each R_{ij} , we limit the

maximum invocation number of ws_i -invoking- ws_j for computing R_{ij}

5.3 Collaboration Reputation Computation

In this section, the collaboration reputation can be computed using these two metrics: invoked reputation and invoking reputation. Therefore, the collaboration reputation of ws_k can be computed with the following equation:

$$CR^{[t_i, t_n]}(ws_k) = \alpha \frac{R_i^{[t_i, t_n]}(ws_k)}{\max(R_i^{[t_i, t_n]}(ws))} + \beta \frac{R_r^{[t_i, t_n]}(ws_k)}{\max(R_r^{[t_i, t_n]}(ws))} (\alpha + \beta = 1) \quad (9)$$

Where α , β are the weight values; $\max()$ is to normalize the collaboration reputation for Web service selection in Section 5. The Algorithm 2 details the collaboration reputation calculation algorithm based on the above equations.

Algorithm 2 Calculate collaboration reputation of Web services

Input: $Set_{\Delta T_1}(ws), Set_{\Delta T_2}(ws), \dots, Set_{\Delta T_n}(ws)$
Output: R

for $set_{\Delta T_i}(ws) \in set_{\Delta T}(ws)$ **do**
 for $ws \in set_{\Delta T_i}(ws)$ **do**
 if ws invokes ws_j **then**
 put(ws , matched interfaces);
 end if
 end for
 $com_{\Delta T} = \text{divideWSCN2Comm}(set_{\Delta T_i}(ws))$
 for $ws_i \in set_{\Delta T_i}(ws)$ **do**
 $R_{i\Delta T}(ws_i) = \text{getInvokedRep}(ws_i.invokeList());$
 $R_{r\Delta T}(ws_i) = \text{getInvokingRep}();$
 end for
end for
for $T_i \in T$ **do**
 $ws_T = \text{sum}(R_{iT_i, rT_i}());$
end for

In Algorithm 2, in each time interval T_i , we compute the R_i and R_r for each ws . In the last loop, we sum $R_{iT_i}(ws_i)$, and $R_{rT_i}(ws_i)$ is evaluated to obtain the collaboration reputation values (CR) in T . The greatest time costs emerge for the invoking relationship collection and community detection. The maximum time complexity is $\Theta(mn + n^2 + mn \log^2 n)$, where m denotes the number of T_i , and n is the Web service number.

6 Trustworthy Service Selection based on Collaboration Reputation

Given a service requirement Re , the Web service selection process in the WSCN is to select a service ws , i.e., $I_{Re} \supseteq I_{ws}; O_{Re} \subseteq O_{ws}; Q_{Re} \leq Q_{ws}$ and $\max(CR)$, where the 3-tuple $Re = \{I_{Re}, O_{Re}, Q_{Re}\}$ represents the customer's functional and non-functional (QoS) requirements, and $\max(CR)$ is used to ensure the Web service selected has the optimal collaboration reputation.

There are two types of service selection algorithms, i.e., simple and complex Web service selection. Thus, CR computation can vary. In following section, we will introduce simple and complex service selection based on CR computation, respectively.

6.1 Simple Web Service Selection

Different from existing methods to select a Web service with best reputation by matching interfaces, the simple Web service selection based on the WSCN can depend on the Web service's neighbors. The best trustworthy selection is defined by the following equation:

$$wss = \bigcap_{neib=i}^n NeiS\left(\bigcup_{int f=j}^k ws_j\right) \quad (10)$$

Where the union operator facilitates finding the Web service ws_j , which satisfies the customer's interface requirement $int f$, and $\bigcup_{int f=j}^k = I_{Re} \cup O_{Re}$; the final solutions wss are obtained by intersecting all ws 's neighbors $Neib^i$ (if $int f \supseteq O_{Re}$) or $Neib^o$ (if $int f \subseteq I_{Re}$). In other words, the final solution's neighbors satisfy the entire functional requirement with $max(CR)$.

6.2 Complex Web Service Selection

Complex Web service selection involves finding a composite Web service with a higher guaranteed CR . The search process can be considered as a multistage graph as shown in Fig. 4.

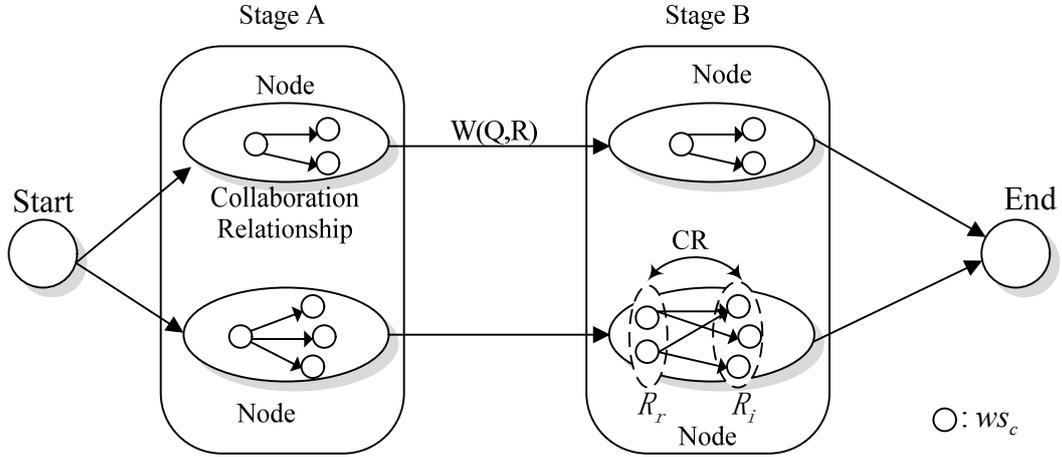


Figure 4. Multistage graph for a composite Web service

In the multistage graph, $ws_c \in Neib_c$ and $ws_o \in Neib_o$ consist of nodes. Thus, the collaboration reputation of a composite service (cs) can be evaluated by transforming Eq. 9 to Eq. 11 as follows:

$$CR^{[t_1, t_n]}(cs) = \alpha \frac{CR_i^{[t_1, t_n]}(t)}{\max(CR_i^{[t_1, t_n]})} + \beta \frac{CR_r^{[t_1, t_n]}(t)}{\max(CR_r^{[t_1, t_n]})} \quad (\alpha + \beta = 1) \quad (11)$$

with

$$CR_i^{[t_1, t_n]}(t) = \frac{\sum_{i=1}^m R_{i_{ws_o}}^{[t_1, t_n]}}{m}$$

$$CR_r^{[t_1, t_n]}(t) = \frac{\sum_{i=1}^m R_{t_{ws_c}}^{[t_1, t_n]}}{m}$$

Where α , β are the weight values; $\max()$ is to normalize the collaboration reputation of cs ; m is the number of cs includes Web services. Therefore, the best trustworthy solution of complex Web service selection can be found according to Algorithm 3 as follows:

Algorithm 3 Complex Web service selection

Input: $\{I_{Re}, O_{Re}, Q_{Re}\}$
Output: Best Satisfied Solution
 $ws = \text{getStartWS}()$;
 $\text{addQueue}(ws, \text{Neib}_c(ws))$
while $\text{queue.size} \neq 0$ **do**
 $ws = \text{queue.pop}()$;
 $ws_{neib_o} = \text{Neib}_c(\text{Neib}_o(ws))$
 if ws_{neib_o} *satisfied* **then**
 $\text{addQueue}(ws_{neib_o})$;
 $\text{addStage}(ws_{neib_o})$
 end if
end while
for $stage_i \in \text{stages}$ **do**
 $\text{compute}(Q_i, CR)$;
end for

In Algorithm 3, we use a breadth-first traversal strategy to combine the matched Web service $\text{Neib}_c(ws)$ and $\text{Neib}_o(ws)$ to construct a service node, and the node is added to each stage by the $\text{addStage}()$. Then the node's collaboration reputation is assessed by CR_{node} . Thus, the CR of the

best trustworthy solution is computed by $\max(\frac{\sum_{i=1}^m CR_{node_m}^{[t_1, t_n]}}{m}, \frac{\sum_{i=1}^m CR_{node_m}^{[t_1, t_n]}}{m})$ after obtaining all stages. Hence,

the time complexity of the selection algorithm is $\Theta(m^2)$.

7 EXPERIMENTS

In this section, we detail the experiments that prove the effectiveness of our Web service selection approach and also compare our approach with other approaches. To generate Web service execution logs, we merely consider the reliability, and the successfully executed composition Web services will be stored in a log.

7.1 Experiment Setup

Suppose that there are three types of service models, good, normal, and bad. The bad service provides unsatisfactory reliability, whereas good services provide satisfactory reliability, and the normal service reliability constantly changes. The good service and bad Web service determination is thus in terms of reliability. To attract customers, the bad Web service publishes the same reliability value as the good Web service. We adopt WSPR approach (it is a famous AI planning-based Web service composition approach) [27] to construct the composite Web service, and the key factor of combining services depends on the three metrics $\langle f, m, p \rangle$, where m denotes the matched interface

number, p the interface's popularity, and f the failure number in the Web service's collaboration history. It has the same meaning as k mentioned above, as when $f > k$, ws_1 will not collaborate with ws_2 . For each ΔT , some requirements are proposed by each customer. Some proposed requirements are also included in Table 1.

Table 1. Parameter settings

Parameters of Service Model	Value
Simulation Cycles	100
Each Cycles ΔT	500
Customer Number	100
Web Service Number	1000
Bad Web Service Number	30%
Good Web Service Number	30%
Normal Web Service Number	40%
Reliability of Good Web Service	[0.9,1]
Reliability of Bad Web Service	[0,0.3]
Reliability of Normal Web Service	[0.3,0.9]
Failure Rate	normal distribution

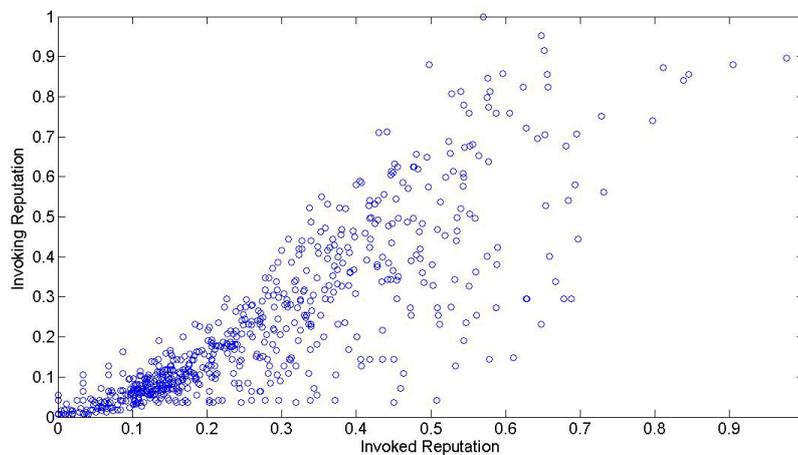


Figure 5 Reputation distribution

7.2 Experimental Results on Reputation Measurement

In this section, we are interested in observing the reputation level of different types of Web services. Fig. 5 shows the invoking and invoked reputation distribution in terms of the above parameter settings. The vertices mainly concentrate in the area [0.05 0.15].

The vertices that are closer to 1 in invoking reputation dimension demonstrate that they have more connections to the TRV . In addition, these results also reflect that these Web services have more inbound and outbound links. Thus, the popular Web service with a guaranteed reputation will have more chance to be assigned for various tasks. The bad Web service mainly concentrates in the area of [0

0.005] because of the fewer recommenders and lower invocation frequency. The reputation value range of a normal Web service is between those two reputation levels but tends to be closer to the bad reputation values because of unstable reliability.

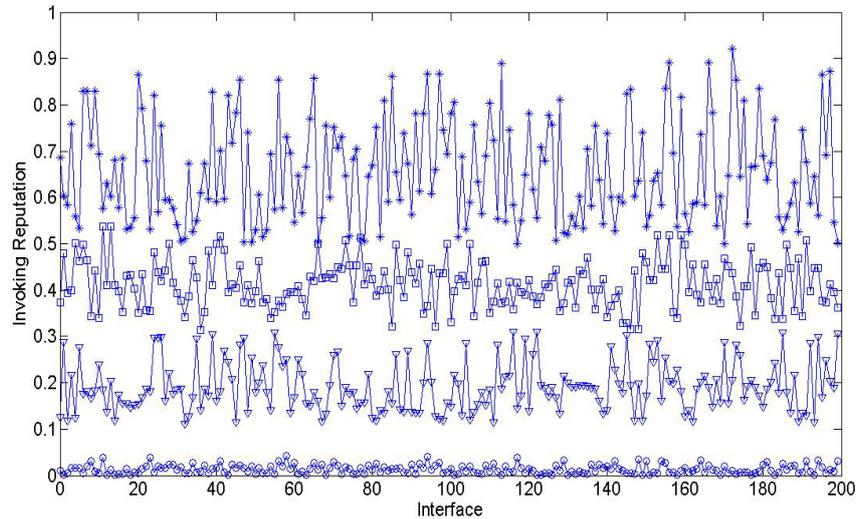


Figure 6 The invoking reputation of with different Web service interfaces.

Fig. 6 shows the invoking reputation of with Web services different interfaces where the Web services have the same input parameters. There are 200 input parameters, and each parameter contains 3 good or normal Web service and 1 bad service. The lower invoking reputation of bad Web services results in them is being gradually eliminated from among the neighbors.

Fig. 7 shows that the success ratios of Web services with different reputation mechanisms such as Reputation, Non-reputation and Collaboration reputation. The success rate of other reputation mechanisms is oscillating. However, our collaboration reputation mechanism is more effective for eliminating bad Web services from the WSCN than other reputation mechanisms. The broadcast message allows the success rate to display an obvious increase in each time interval. Although a bad service can join the Web service composition in the beginning, the neighbor updates prevent such services from collaborating with more Web services.

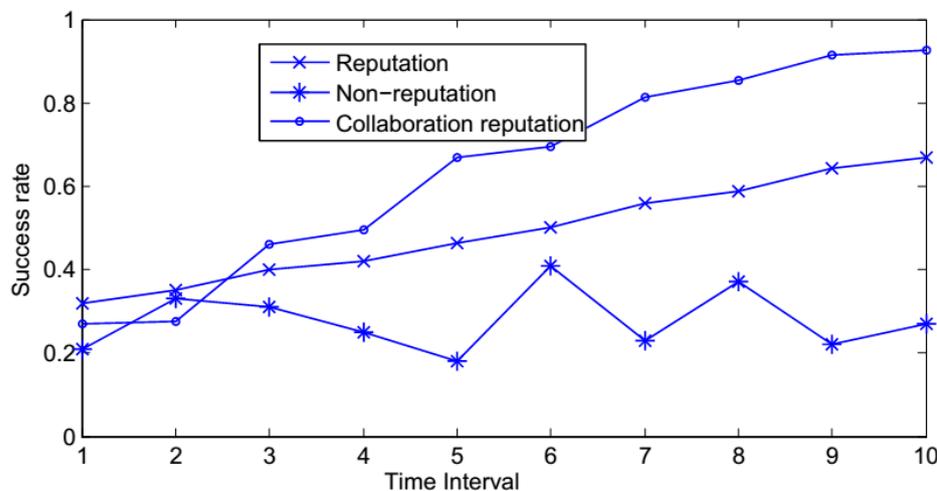


Figure 7 Success rate with respect to different reputation mechanisms

7.3 Experimental Results on Web Service Selection

The following experiments reveal the success rate and time cost of service selection. We tested a different number of Web service in solutions, which has 25 Web services at most. We compare our approach with traditional trustworthy Web service selection (called TTWSS) and Web service selection based on WSPR [27] (called WSSW). TTWSS is a very simple service selection approach and it often selects the services with the highest reputation to assemble a trustworthy composite service. WSSW performs trustworthy Web service selection based on AI planning and it can find a solution in polynomial time, but with possible redundant Web services.

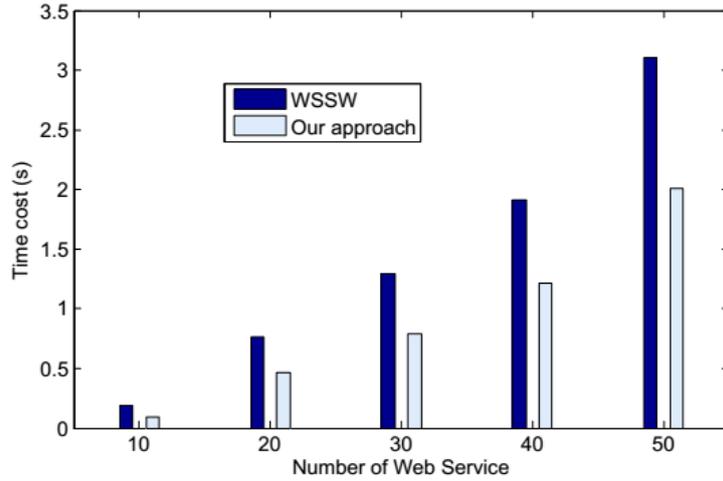


Figure 8. Time cost of Web service selection

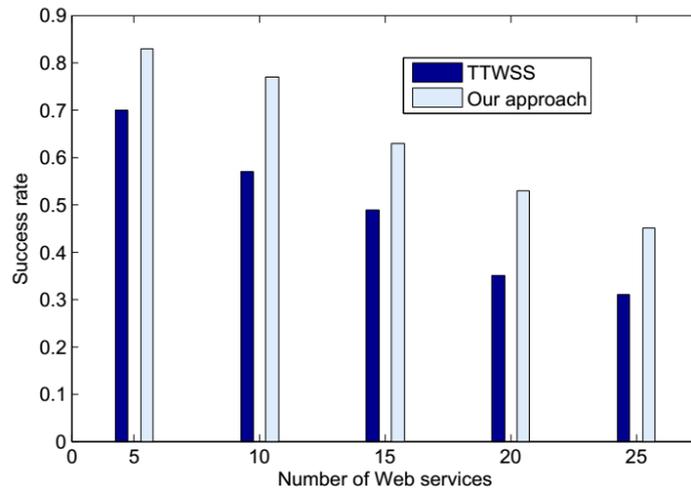


Figure 9 Success ratio of Web service selection

Fig. 8 shows that the computation time of WSSW is half that of our approach as the search process simply involves forward searching. Although the collaboration reputation computation cost some extra time, the search space in our approach is limited to neighbors, whereas the WSSW needs to match all Web services.

As shown in Fig. 9, with the collaboration reputation guaranteed, our approach has a more advantage in the success ratio than TTWSS, and increases by approximately 20%. Why? For TTWSS,

it only combines the Web services with the highest reputations leads to a much greater number of Web services being involved with the composite Web service, because of less consideration of interface optimal matching. As each Web service has a certain failure ratio, TTWSS results in a higher failure rate for service selection when only combining the services with the highest reputations.

8 Conclusions

In this paper, we proposed a trustworthy Web service selection approach based on collaboration reputation by constructing a Web service collaboration network based on social networks. According to the experiment results, the proposed approach can fairly and effectively evaluate the Web service's reputation, and it can especially effectively distinguish Web services with different reputation levels from service selection process. The success rate of Web service selection increases as the interaction round increases, and Web services with low reputation are excluded in the service selection process. Moreover, the efficiency of the Web service selection is also guaranteed.

Of course, our approach has some limitations: 1) the efficiency of our approach will slide down significantly or even be unable to work when the Web service community of the social network is very small; 2) our approach is unfair for new published Web services or the services with few invocation records in a mobile social network [26]; 3) our approach cannot guarantee global QoS constraints for service selection system. Our future work will also investigate the best method to perform this trust relationship evolution for the first limitation.

Acknowledgements

The work presented in this study is supported by the Natural Science Foundation of Beijing under Grant No.4132048, NSFC (61202435), and NSFC (61472047).

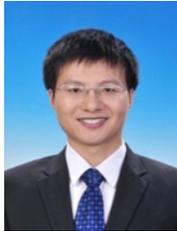
References

- [1] L. Eng Keong, C. Ruichuan, and C. Zhuhua, Social Trust and Reputation in Online Social Networks, Proceedings of IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS 2011) , 2011, pp. 811-816.
- [2] S.G. Wang, Q.B. Sun, and F.C. Yang, Towards Web Service selection based on QoS estimation. International Journal of Web and Grid Services 6 (2010) 424-443.
- [3] W. Ping, C. Kuo-Ming, L. Chi-Chun, H. Chun-Lung, and L. Yinsheng, A Fuzzy Model for Selection of QoS-Aware Web Services, Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2006), 2006., pp. 585-593.
- [4] S. Wang, Z. Zheng, Z. Wu, F. Yang, and M.R. Lyu., Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems. IEEE Transactions on Services Computing, 2014, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6953216&tag=1.
- [5] L. JooYoung, and J.C. Oh, A model for recursive propagations of reputations in social networks, Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), 2013, pp. 666-670.
- [6] Wang S, Zhu X, Zhang H., "Web service selection in trustworthy collaboration network," Proceedings of the 8th IEEE International Conference on e-Business Engineering (ICEBE 2011), pp. 153-160.
- [7] Wang Y., Vassileva J., "A Review on Trust and Reputation for Web Service Selection," Proceedings of the 27th

- International Conference on Distributed Computing Systems Workshops (ICDCS 2007), pp. 25-32, 2007.
- [8] Le-Hung Vu, Manfred Hauswirth and Karl Aberer, "QoS-based Service Selection and Ranking with Trust and Reputation Management," Proceedings of the Cooperative Information System Conference (CoopIS 2005), pp. 466-483, 2005.
- [9] Stephen S. Yau, Jing Huang and Yin Yin, "Improving the Trustworthiness of Service QoS Information in Service-Based Systems," Proceedings of 7th Autonomic and Trusted Computing (ATC 2010), pp. 208-218, 2010.
- [10] D. W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions," Proceedings of the International Conference on Distributed Computing Systems (ICDCS 1998), pp. 312-321, 1998.
- [11] Surya Nepal, Wanita Sherchan, Jonathon Hunklinger, Athman Bouguettaya, "A Fuzzy Trust Management Framework for ServiceWeb," Proceedings of the IEEE International Conference on Web Services (ICWS 2010), pp. 321-328, 2010.
- [12] Wu, G., Wei, J., Qiao, X., Li, L., "A Bayesian network based QoS assessment model for web services," Proceedings of the IEEE International Conference on Services Computing (SCC 2007), pp.498-505, 2007.
- [13] L.D. Alfaro, A. Kulshreshtha, I. Pye, and B.T. Adler, "Reputation systems for open collaboration," Communications of ACM, 2011, 54 (8): 81-87.
- [14] K. McNally, M.P. O'mahony, and B. Smyth, "A comparative study of collaboration-based reputation models for social recommender systems," User Modeling and User-Adapted Interaction, 2014, 24 (3): 219-260.
- [15] Chung-Wei Hang and Munindar P. Singh, "Selecting Trustworthy Service in Service-Oriented Environments," Proceedings of the 12th AAMAS Workshop on Trust in Agent Societies (TRUST 2009), PP. 1-12, 2009.
- [16] S. Elnaffar, Z. Maamar, H. Yahyaoui, J. Bentahar, Ph. Thiran, "Reputation of communities of web services-preliminary investigation," Proceedings of the 22th IEEE International Conference on Advanced Information Networking and Applications (AINA 2008), pp. 1603-1608, 2008.
- [17] B. Khosravifar, J. Bentahar, P. Thiran, Ahmad Moazin, and Adrien Guiot, "An Approach to Incentive based Reputation for Communities of Web Services," Proceedings of the 7th International Conference on Web Services (ICWS 2009), PP. 303-310, 2009.
- [18] Tian Zhu, Bin Wu and Bai Wang, "Social Influence and Role Analysis Based on Community Structure in Social Network," Proceedings of International Conference on Advanced Data Mining and Applications (ADMA 2009), pp.788-795, 2009.
- [19] Hyunyoung Kil, "Graph Theoretic Topological Analysis of Web Service Networks," Journal of World Wide Web, 2009, 12(3):321-343.
- [20] Xizhe Zhang, Ying Yin, Mingwei Zhang, Bin Zhang, "A Composite Web Services Discovery Technique Based on Community Mining," Proceedings of the 4th Asia-Pacific Services Computing Conference (APSCC 2009), pp. 445-450, 2009.
- [21] Guang Ji, Jie Zhang, Zhuofeng Zhao and Yanbo Han, "Service Collaboration Network: A Novel Mechanism for Web Service Management," Proceedings of the 12th Asia-Pacific Web Conference (APweb 2010), pp.85-91, 2010.
- [22] Eugster P, Guerraoui R, Handurukande S, Kermarrec AM, Kouznetsov P, "Lightweight probabilistic broadcast," ACM Transactions On Computer Systems, 2003,21(4):341-374.
- [23] Clauset A, Newman M. E. J., and Moore C., "Finding Community Structure in Very Large Networks," PHYSICAL REVIEW E, 2004, 70: 66-111.
- [24] Seog-Chan Oh, Kil, H. , Lee, D. , Kumara, S.R.T., "WSBen: a Web-services discovery and composition benchmark," Proceedings of the 2006 IEEE International Conference on Web service (ICWS 2006), pp. 239-246, 2006.
- [25] Ye Q, Wu B, Wang B. JSNVA, "A Java Straight-line Drawing Framework for Network Visual Analysis," Proceedings of the 4th International Conference on Advanced Data Mining and Applications (ADMA 2008), pp.667-674, 2008.
- [26] S. Brin, L. Page, "The anatomy of a large scale hyper textual web search engine," Computer Networks and ISDN Systems, 1998, 30(7):107-117.
- [27] Oh, S., Lee, D., & Kumara, S. R., "Web Service Planner (WSPR): An Effective and Scalable Web Service Composition

Algorithm," International Journal of Web Services Research, 2007, 4(1): 1-22.

- [28] Y. Wang, J. Tang, Q. Jin, and J. Ma, "On studying business models in mobile social networks based on two-sided market (TSM)". The Journal of Supercomputing, 2014, 7(3): 1297-1317.



Shanguang Wang is an associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He received his Ph.D. degree in computer science at Beijing University of Posts and Telecommunications of China in 2011. His PhD thesis was awarded as outstanding doctoral dissertation by BUPT in 2012. His research interests include Service Computing, Cloud Services, QoS Management.



Lin Huang received the M.E. degree in computer science and technology from the Institute of Network Technology, Beijing University of Posts and Telecommunications, in 2012. Currently, she is a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include Reputation measurement, Web service selection.



Ching-Hsien Hsu is a professor in department of computer science and information engineering at Chung Hua University, Taiwan. His research includes high performance computing, cloud computing, parallel and distributed systems, ubiquitous/pervasive computing and intelligence. He has been involved in more than 100 conferences and workshops as various chairs and more than 200 conferences/workshops as a program committee member. He is the editor-in-chief of international journal of Grid and High Performance Computing, and serving as editorial board for around 20 international journals.



Fangchun Yang received his PhD degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 1990. He is currently a professor at the Beijing University of Posts and Telecommunication, China. He has published 6 books and more than 80 papers. His current research interests include network intelligence, services computing, communications software, soft switching technology, and network security. He is a fellow of the IET.