A LOCATION-AWARE SERVICE SELECTION MODEL

Ying Li, Zhiling Luo, Jianwei Yin
Zhejiang University
cnliying@zju.edu.cn, luozhiling@zju.edu.cn, zjuyjw@zju.edu.cn

Abstract

This paper introduces a service selection model with the service location considered. The location of a service represents its position in the network, which determines the transmission cost of calling this service in the composite service. The more concentrated the invoking services are, the less transmission time the composite service costs. On the other hand, the more and more popular big data processing services, which need to transfer mass data as input, make the effect much more obvious than ever before. Therefore, it is necessary to introduce service location as a basic feature in service selection. The definition and membership functions of service location are presented in this paper. After that, the optimal service selection problem is represented as an optimization problem under some reasonable assumptions. A shortest-path based algorithm is proposed to solve this optimization problem. At last, the case of railway detection is studied for better understanding of our model.

Keywords: service selection; service composition; service location; shortest path; big data

1. INTRODUCTION

In the Service Oriented Architecture (SOA), complex services can be easily composed using individual services form various service providers. Service selection, one of the critical problems of service composition, is generating considerable interests in recent years in several computer science communities. Service selection determines the non-functional qualities of composite service in varieties directions, such as cost, reliability and reputation. A large number of methods for service selection are presented by different researchers and communities. So there is a considerable amount of service selection approaches have been proposed by different researchers and communities. Some of these approaches have even already been used in practice.

However, the era of big data brings new challenges to service selection. The concept of big data comes from the research on Database, which represents that the size of data involved in computing is huge (Gigabyte, Terabyte, even Petabyte). Unlike being ignored or even considered useless before, in the last decades, the big data generating from producing and trading is considered as a treasure with of the development of data mining technology. As a result, the big data services which processes the big data are also brought to the fore now. An example of big data service is the customer mining service, which finds out those potential customers by analyzing the customers' purchasing behavior.

The data accessed by big data services are typically very large, which brings a big challenge to classical service selection methods. In the above example of customer mining service, all the known customer records should be read at the input, which can be as large as a few TB. When invoking these big data services in a composite service, there are huge data should be transmitted in the network. The time cost in transmission process is the transmission cost, which is not fully taken into account in classical service selection approaches. Most classical service selection approaches use the response time to identify the transmission cost, which is not suitable in this context now. Because the response time represents the time recorded from user calls until the service responses. It is usually much smaller than the transmission time; especially the size of data is further huge.

In order to select the optimal services from candidate services with transmission time of big data considered, the service location is introduced in this paper. The location of a service is determined by the area the service belongs to. The area is an abstract concept which is defined as a set of services with high speed communication connected. That is to say, a sub network, a cloud, even a computer is suitable to be considered as an area. The transmission cost among two services in a same area is small enough to be ignored when comparing with those deployed in two different areas. By considering these definitions and assumptions, the optimal service selection turns out to be an optimization problem that how to select the most concentrated services.

It is quite difficult to solve this optimization problem. However, it is considerable to reduce the original problem to a shortest path selecting problem. An algorithm based on shortest path selection is proposed to solve the original optimization problem.

In order to give a better understanding of our model and algorithm, we discussed a fully service selection example about railway disease detection. The railway disease detection is a daily work to check the status of each part of the railway to find out the potential damage and broken (called disease). We have studied the data of JGX (Beijing-Guangzhou railway line, China) which is as huge as 200MB per day. Three steps of analysis is needed for the collected raw data, namely Noise Reduction, Disease Detection and Classification. Lots of service can be used to handle each analysis step. Our job is to select the optimal services with less transmission cost.

The contribution of this paper can be stated as follows:

- *A location-aware service selection model*. Differing from classical service selection approaches, the service location is introduced as a critical feature to identify the transmission cost for big data services and the optimal selection problem is converted into an optimization problem.
- *An optimal service selection algorithm*. An algorithm based on the shortest path selection for location aware service selection is proposed. The algorithm has a linear time complexity and easy to implement.

The rest of the paper is organized as follows. Section 2 reviews the classical service selection approaches. Section 3 proposes the system model with the location definition and a few considerable assumptions. Section 4 discusses the problem of location aware service selection and introducing the service selection algorithm. The case study is presented in section 5. The performance study is presented in section 6. Finally, section 7 concludes this paper and an outlook on possible continuations of our work.

2. RELATED WORK

Service composition is a classical problem and there are amounts of papers in this field. Zeng, L. (2003, 2004) proposes the composition method with the quality of service (QoS) considered. The idea caused lots of attentions. Alrifai, M. (2009, 2010a) converts the service composition problem to a decision problem on which component services should be selected such that user's end-to-end QoS requirements (e.g. Availability, response time) and preference (e.g. Price) are satisfied. It is the source of selecting by solving the optimization problem. Mixed integer programming (MIP) is used which is quite widely used in this model. Zhang, M. (2010) takes the service environment into consideration. The black-box analysis method of optimizing composite service is adopted. Klein, A. (2010) considers repeated executions of services in the long-term. The modified problem is modeled with linear programming. And it is solved optimally in polynomial time. A distributed heuristic approach is proposed in Jing, L. (2010). Bakhshi, M. (2010) proposes an approach using fuzzy logic to infer based on quality measures ranked by user. Lecue, F. (2011) takes the semantic dimension into consideration. Jin, J. (2011) proposes a heuristic service composition method, named LOEM-T (Local Optimization and Enumeration Method with Solution Tendency Estimation). Wagner, F. (2011) utilizes a data structure which arranges functionally similar services in clusters and computes the QoS of each cluster. This idea is quite interesting and it benefits a lot in composition. There are more composition methods including Ardagna, D. (2007), Funk, C. (2007), Li, H (2011), Bo, Y. (2011), Babamir, S. (2011) and Wang, Pengwei (2011).

The QoS impacts the composition in two directions: service discovery (including recommendation) and service selection.

Ran, S (2003) proposes a model for discovery in which functional and non-functional requirements are considered to evaluate QoS metrics. A metadata model on the basis of extended UDDI is proposed, where quality information data is used to describe the QoS of registered services including quality name, type, value, units, etc. Al-Masri, E. (2007) introduces a solution for controlling the discovery process across accessible services and Web Service Relevancy Function (WSRF) is used for measuring the relevancy ranking of a particular web service based on QoS metrics and client preferences. Mohana, R. (2011a, 2011b) presents an algorithm for building a rule-based model for ranking the service based on QoS using fuzzy clustering and particle swam optimization (PSO). Paularj, D. (2012) introduces OWL-S as a web ontology language for service discovery and composition. A service recommendation based on collaborative filtering is proposed in Tang, M. (2012).

There are lots of research achievements in service selection. Maximilien, E. (2003) proposes an approach in which agents assisting an application in selecting implementations that best match the quality criteria. Liu, Y. (2004) mentions that the QoS values cannot solely be collected from the service provider. Since this is subject to manipulation by the providers. In their framework, the QoS model is extensible and QoS information can be computed based on execution monitoring by users, or via requesters feedback. Soydan, B. (2004) proposes a framework combined an ontology of attributes with evaluation data. Ardagna, D. (2005) extends the mixed linear programming model to include local constraints and global constraints. Vu, L. (2005) presents a new QoS-based semantic web service selection and ranking solution with the application of a trust and reputation management method. Yu, T. (2005a, 2005b) models the end-to-end delay constraint as the multiple choice knapsack problem (MCKP) and provided efficient solutions. Cardellini (2007) proposes a method considering a group of request. A selection is carried out per group of requests rather than per-request. Kritikos (2009) has developed an extensible and rich ontology language for QoS-based WS description. Sun, Q. (2010a) proposes a quick service selection approach (QSSA) which adopts particle swam optimization and fuzzy logic control to support fast and dynamic service selection. Kun, Z. (2010) introduces a composite agent service selection algorithm for non-functional attributes based on simulated annealing. Sun, Q. (2010b) proposes a new approach based on the notion of the skyline (SWS). Alrifai, M. (2010b) proposes a method quite similar to Sun, Q. (2010b). Selecting service from a set of functionally equivalent services is a multi-criteria decision making problem. Wang, Ping. (2011) takes the opinion that different consumers invariably hold differing views of the service contents and it is necessary to estimate the degree of consumer trust in a particular service based on the consumers' direct experiment and indirect recommendation of the service. Suleiman, B. (2011) classifies consumers into groups/classes and optimized

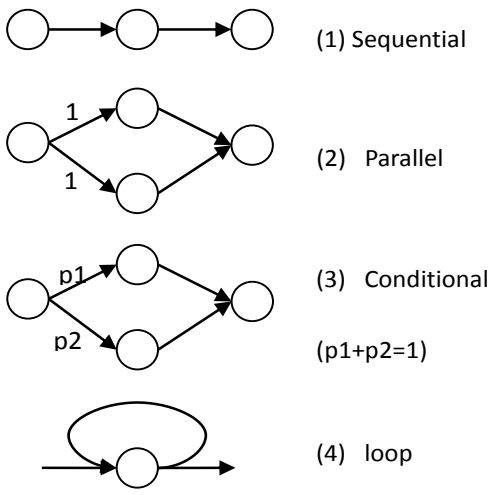

Figure 2: Different control process

multiple QoS criteria based on the customer classes. Lim, E. (2011) takes the satisfaction of user, web service and community (3-way) into consideration. The selection is based on this three satisfaction factors. Kang, G. (2011) uses the Euclidean distance with weights to measure the degree of matching of service based on QoS and provided a global optimal service selection when the service request conflicted. More service selection methods including Ma, S. (2010), Bo, Y. (2011), Lin, D. (2012) and Benouaret, K. (2012a, 2012b).

These QoS-based selection methods solve the problem of selecting the optimal services for composition. However, these methods cannot solve the problem of location aware selection because the location has two important characteristics differing with the qualities mentioned in classical selection methods.

1. *Location is not quantitative*. A location of a service is not directly computable. The quantitative value extends from the location is the distance. The distance of any two services is determined when the locations of the two services are confirmed. It is better to understand distance as a relative value. Unlike location, other qualities such as reputation are really quantitative and it is an absolute value.
2. *Distance is context-sensitive*. There is not an optimal choice without considering the services selected before and after. The lowest transmission cost happens only with all nearby services have the smallest distance. Other qualities are not context-sensitive; it is possible to find an optimal service without considering the service before and after. Take the reputation as an example, the optimal service is the one with largest reputation.

Because of these different features, it is not possible to use the classical methods to solve the location aware selection problem. Furthermore, modeling of location-aware service selection was never formulated nor studied..

3. MODEL

3.1 SERVICE SELECTION

3.1.1 BASIC SERVICE SELECTION

Service selection is a critical part of service composition. Generally speaking, the process of service composition can be divided into three steps.

- Building the composite process with abstract services. Unlike the concrete service, the abstract service is the symbol representing a group of services with similar functions and interfaces. The abstract services are composed together by some control statements (such as assignment, switch and loop).
- Finding some suitable services (namely candidate services) for each abstract service. These candidate services have the same functions and interfaces as the abstract service. This step is known as service discovery. The discovery process is mainly searching first k candidate services for each abstract service in the service library with some functional constraints. Service recommendation is also similar, which selects the most suitable services and recommends to users.
- Selecting an optimal service from each candidate service group. The abstract services should be replaced with selected concrete ones in the composite process. This step is known as service selection. How to select the optimal services with location considered is the

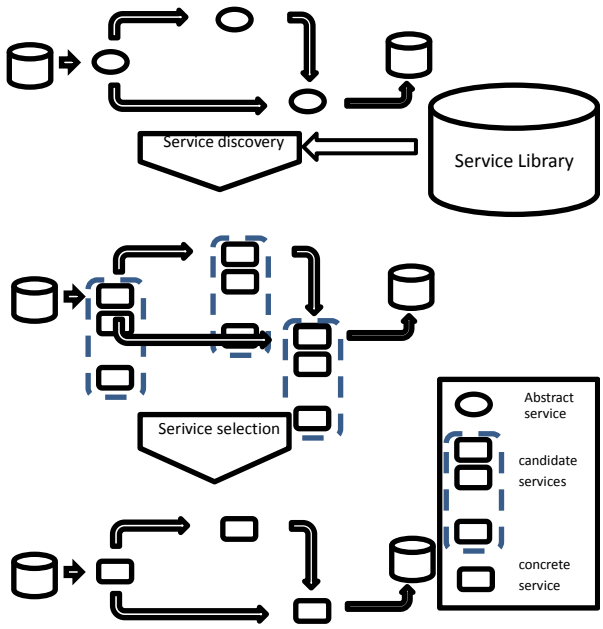

Figure 1: Process of service discovery and selection

major problem discussed in this paper.

Fig. 1 presents the process of service discovery and service selection. In this figure, the composite service process is presented at the top, which fetches data from a database, calls three abstract services (present as the ellipses) and writes the result into another database. After the step of service discovery, alternative services are picked up from service library and constructed as candidate service groups. The last step is selecting the optimal concrete services from each candidate service group.

In most composite services, there are some control process patterns, as sequential process, parallel process, conditional process and loop process. Fig.2 represents the different control processes. (1) is the sequential process, (2) is parallel process, (3) is conditional process and (4) is loop process. The sample of Fig. 1 contains a parallel process. However, the hybrid control processes make it too complex to analyses the optimal selection. To simplify the problem, the sequential process is majority discussed in the rest, and the analysis of parallel process, conditional process and loop process will be extended in the future work. An example of sequential process is present in Fig. 3. In order to describe the following analysis precisely, some math symbols are introduced as follows.

A symbol S represents the whole services used in selection, which is constructed by M groups of candidate services. $S = \{S_1, S_2, ..., S_M\}$. Each S_i not only represents an abstract service but also represents a group of candidate services having the same function as the abstract one. $S_i = (S_{i1}, S_{i2}, ..., S_{iN})$. N is the number of candidate services of each group. S can also be represented by a matrix.

$$S = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{pmatrix} \quad (1)$$

There are two special services, namely S_0 and S_0'. They are the services providing the reading and writing operation of the database. S_0 and S_0' should be confirmed before selecting.

3.1.2 TRANSMISSION COST

The transmission cost is an important indicator of composite service. In the sequential process, the transmission cost happens since it spends some time transport the data from one service to another (the next one). A symbol $C_{ij} = Tr_{ij} * Da_{ij}$ is used to represent the transmission cost from service s_i to service s_j. Tr_{ij} is the transmission speed from service s_i to service s_j and Da_{ij} is the amount of data. While Tr_{ij} is determined by the network condition and the Da_{ij} is determined by composite process. The whole transmission cost is $C = \sum_i^{M-1} C_{ii+1}$. It is easy to find out that Da_{ij} cannot be changed by the selection of service. The only way to minimize Da_{ij} is to modify the service process which is out of discussion of this paper. However, the different selection really determines the different Tr_{ij}. So the purpose of our model is to find out a group of optimal service selection minimize Tr_{ij}.

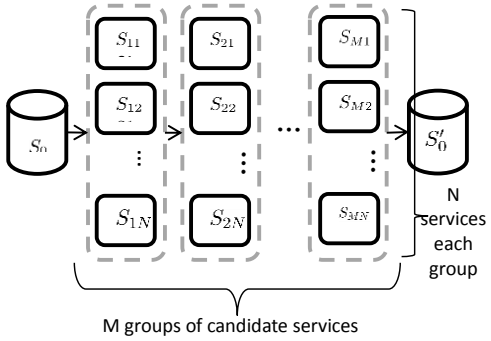

Figure 3: Sequence service process

Fig. 4 represents a sample composite service with S_1, S_2 and S_3. The network condition determines Tr_{12} and Tr_{23} . The whole transmission cost $C = C_{12} + C23 = Tr_{12} * Da_{12} + Tr_{23} * Da_{23}$.

3.2 SERVICE LOCATION

3.2.1 DISTRIBUTION ASSUMPTION

The location of a service is the feature representing its network environment. Generally speaking, the distribution of services is not random in the network. A reasonable assumption of the service distribution is proposed.

Assumption I. (The gathering): *Services are not randomly distributed in the network. A certain number of services are naturally gathered as a service set. The services in one set are high speed connected. This services set is denoted as an area.*

The Assumption I can be explained as follows. On one hand, the services provided by the same provider are deployed in the same server or some servers in the same subnet. In this situation, the server and the subnet is an area. On the other hand, a Cloud is an area with lots of services deployed on it by different providers. By our definition, a server is an area, a subnet is an area and even a Cloud is also an area. The only constraint is that the services in the same area are high speed connected. Fig. 5 presents the relation of areas and services. In Fig. 5, S1, S2, S3, S4, S5 represent services. Area1, Area2 and Area3 represent areas. S1, S2, S3 belong to Area1, S4 belongs to Area2 and S5 belongs to Area3. By considering the definition of area, S1 S2 are high speed connect, namely the cost of transmitting data between them is quite low.

3.2.2 LOCATION

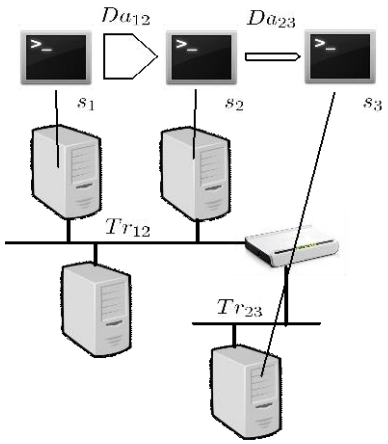

Figure 4: The transmission cost

Considering a context with O areas, $A = \{A_1, A_2, ..., A_O\}$ is used to represent the all areas. The map from the services S to the area A is confirmed before selection. In other words, the relationship of belonging is confirmed for any services. This belonging relationship represents the location of the service. A function (denotes $\mathcal{I}$) is defined to identify the index of the area which the service belongs to.

$$\mathcal{I}(s) = \sum_{i=0}^{O} i \times \{s \in A_i\} \tag{2}$$

A special index function is used above, whose value equals 1 when the proposition in the braces is TRUE, and otherwise it equals 0:

$$\{p : boolean\} = \begin{cases} 1 & p \text{ is True} \\ 0 & p \text{ is False} \end{cases} \tag{3}$$

The location of a service is represented by the area it belongs to, in other words, the index of its area, namely $\mathcal{I}(s)$.

3.3 Service Distance

3.3.1 Distance of Areas

The distance of the two areas is the measurement of the speed of transmission data between the two areas. Since all the areas in the context are confirmed, the distances of each two of them are also confirmed. A matrix D is defined to represent these distances. D_{ij} represents the distance between A_i and A_j.

Table 1 is an example of D representing the distances of areas in Fig. 5. In fact D is just the Distance matrix of the graph which constructed by the areas. Generally speaking, $D_{ij} \neq D_{ji}$. Because the network transmission sometimes (differing from types of network) spends different time in uploading and downloading. The different types of network is not the majority of this paper, so it is considerable to generally use D to represent the distances rather than ensure D is a symmetrical matrix. While in order to simplify our computing, $D_{ij} = D_{ji}$ is assumed in the following samples. For the convenience of writing, a function $\mathcal{D}(i, j)$ is defined instead of using D_{ij} directly.

Table 1: An example of D

D	A1	A2	A3
A1	0	D1	D2
A2	D1	0	D3
A3	D2	D3	0

3.3.2 Distance of Services

In order to give a better formalization we do not use the Tr_{ij} to represent the basic indicator of services, service distance is defined as the mean transmission speed of two services. We use d_{ij} to denote the distance between service s_i and service s_j. d_{ij} is the mean of Tr_{ij} through a period of time. Tr_{ij} changes in different time. Since in the rush-hour, Tr_{ij} is much smaller than that in other time. Here is a proposition about service distance.

Proposition II (Service Distance): *For any service s, d_i represents the distance between service s and service s_i, which is in the same area as s and d_j represents the distances between service s and service s_j, which is in another area, we have $d_i < d_j, \forall i, j$.*

Proof. *Let's assume that existing a $\hat{j}$ and a $\hat{i}$, such that $d_{\hat{i}} > d_{\hat{j}}$. It contradicts with the definition of area. Because the Assumption 1 guarantees that the services in the same area have a high speed connection. If $d_{\hat{i}} > d_{\hat{j}}$, it is the service s_j that in the same area with s instead of service s_i. It contradicts the condition above.*

If exist a $\hat{j}$ and a $\hat{i}$, such that $d_{\hat{i}} = d_{\hat{j}}$, s_j and s_i should either both in the same area with s or both not in. It contradicts assumptions that service s_j is in another area. **Q.E.D.**

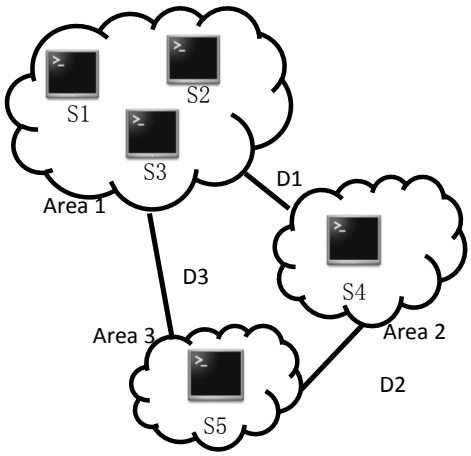

Figure 5: The relation with areas and services

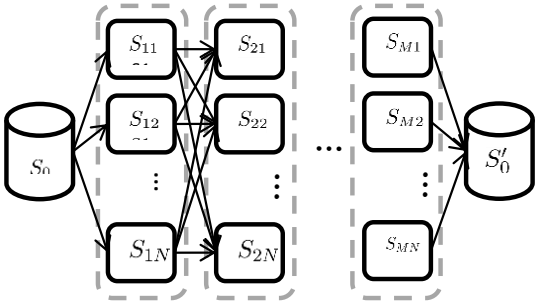

Figure 6: An example of the graph of selection problem

Proposition II guarantees $d_i < d_j$, which is a weak condition. Following assumption provides a strong condition.

Assumption III (Strong Service Distance Condition): *For any service s, d_i represents the distance between service s and service s_i, which is in the same area as s and d_j represents the distances between service s and service s_j, which is in another area, we have $d_i \ll d_j, \forall i, j$.*

The Assumption III is easy to understand because the transmission cost between the two services in a same subnet is always much smaller than the cost crossing subnets. The strong condition makes it possible to define the distance of any two services in a simple way.

The distance of any two services is defined by the distance of the areas they each belongs to. For service s_{ij} and s_{kl}, the distance function $dis(s_{ij}, s_{kl})$ is defined as follows:

$$dis(s_{ij}, s_{kl}) = \mathcal{D}(\mathcal{I}(s_{ij}), \mathcal{I}(s_{kl})) \tag{4}$$

Since the strong condition, 0 is used to represent the distance between services in the same area. The distance of S1 and S4 is just D1 in Figure 5, while the distance of S1 and S3 is 0.

4. LOCATION AWARE SELECTION

4.1 EVALUATION FUNCTION

With the definition of distance under the assumptions mentioned in section 3, the problem of selecting optimal services with lowest transmission cost can be converted into a problem of selecting optimal services with the smallest distance between each two services. Considering the sequential process described in Fig. 6 and the candidate services S defined in Formula (1), the vector $\theta = (\theta_1, \theta_2, ..., \theta_M)$ is used to identify a considerable selected service group. For an element θ_i in θ, the service $s_{i\theta_i}$ is selected as the concrete service for the abstract S_i. ω is the weight coefficient, which is also a vector. The sum of the distances of the selected group θ is defined as an evaluation function as follows.

$$\mathcal{F}(\theta) = \sum_{i=1}^{m-1} dis(s_{i\theta_i}, s_{(i+1)\theta_{i+1}})\omega_i \tag{5}$$

Considering the distance between the first service and the database service S_0, and the distance between the last service and the database service S'_0, the complete evaluation function is defined as follows.

$$\begin{aligned} \mathcal{F}'(\theta) \quad = \quad & dis(s_0, s_{1\theta_1})\omega_0 + \\ & \mathcal{F}(\theta) + \\ & dis(s_{m\theta_m}, s'_0)\omega_m \end{aligned} \tag{6}$$

The Simple Additive Weighting (SAW) is used here. The ω is the weight coefficient. The weight coefficient really makes sense because the transmission cost is not only related to the transmission speed but also related to the size of data. Generally speaking, the size of data in different part of composite process is different. An example is that S_1 read the whole data from the database service S_0 as the input, while output the statistic result of the origin data. The second service S_2 reads the statistic result as the input. In this condition, the distance between S_1 and S_0 has more influence on the whole transmission cost than that between S_2 and S_1. So we can initialize the weight parameter with $\omega_1 > \omega_2$.

4.2 OPTIMIZATION PROBLEM

Finding the smallest $\mathcal{F}'$ with a feasible parameter θ is an optimization problem, which can be described as follows:

$$\begin{aligned} \inf \quad & \mathcal{F}'(\theta) \\ \text{subject to} \quad & \theta_i \in [1, N] \\ & \theta_i \in \mathcal{Z} \end{aligned} \tag{7}$$

This formula means the feasible set of parameter θ is constrained by $1 \leq \theta_i \leq N$ and θ_i is an integer. The optimal solution $\hat{\theta}$ satisfies $\hat{\theta}$ belongs to feasible set and $\forall \theta, \mathcal{F}'(\hat{\theta}) \leq \mathcal{F}'(\theta)$. Following theorem presents the relationship with this optimization problem with our original service selection problem.

Theorem IV (Optimization problem): *A location aware service selection problem with service distribution $\mathcal{I}$ and area distance D confirmed, is equivalent to the optimization problem described in (7).*

The proof of this theorem is omitted because it is mentioned in previous sections. More concern is given on the solving of this optimization problem.

Formula (7) is an integer programming (IP) problem, which is a famous NP problem. A problem belonging to NP class means its positive solutions can be found in polynomial time on a non-deterministic machine. Generally speaking, there is not an algorithm with Non-deterministic Polynomial time complexity [21]. A famous IP problem is Traveling Salesman Problem (TSP): Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?[22] However, because of the features of our model, it is possible to convert our problem to a shortest path problem of a special graph and an algorithm of polynomial time complexity is proposed later.

4.3 SHORTEST PATH PROBLEM

A possible way of solving problem (7) is iterating with all the possible combination tried and finding the combination with $\mathcal{F}'(\theta)$ minimized, called the iterating algorithm. Iterating algorithm works in this way: iterating the whole possible combination, which is as much as M^N to get the best selection (minimized $\mathcal{F}$). It is easy to find that iterating algorithm reaches the global optimal solution with time complexity $O(M^N)$, which is exponential and not useful in practice.

In order to fully take advantage of the structural features, the optimization problem (7) is converted into a conditional shortest path selection problem. The transferring process can be described as the following steps.

1 Iterating all candidate services (including database services S_0 and S_0') and doing step 2 and 3.
2 Append service s_{ij} in the vertex set V.
3 Iterating all candidate services in the next group. For each $s_{i+1,k}$, Append edge E_{ijk} which is the directed edge connecting from the service s_{ij} to the service $s_{i+1,k}$, in edge set E. The length of the edge E_{ijk} is $dis(s_{ij}, s_{(i+1)j})\omega_i$.
4 Find the shortest path from S_0 to S_0'.

The shortest path problem is a classical topic which has been studied in many researches. Once we can guarantee that the vertices of the shortest path are the optimal selection.

Lemma V. *The number of vertices of a path from S_0 to S_0' is M+2. The i-th vertex is one of services in S_{i-1}.*

Proof. *The first conclusion is natural because the transferring process described above. If exist a $\hat{i}$, that the $\hat{i}$ -th vertex does not belong to $S_{\hat{i}-1}$, then $(\hat{i}+1)$-th vertex must does not belong to $S_{\hat{i}}$. Because, the edges to vertex in $S_{\hat{i}}$ comes from $S_{\hat{i}-1}$. We can reduce that $(M+2)$-th vertex is not S_0', which contradicts with the first conclusion.* **Q.E.D.**

With the lemma, it comes to a useful theorem which guarantees the optimal selection.

Theorem VI (Shortest path). *The vertices of the shortest path from S_0 to S_0' is the optimal solution of the optimization problem (7).*

Proof. *Sufficiency: with the help of Lemma V, it is easy to find that for the shortest path $(v_1, v_2, ..., v_t)$,there are three important facts:*

1. *The number of vertices of shortest path is equal to the whole number of abstract services and database services, namely $t = M + 2$.*
2. *v_1 is S_0 and v_{M+2} is S_0'.*
3. *For any $i \in [2, M+1]$, v_i is a service belongs to candidate group S_{i-1}.*

These facts guarantee that $(v_1, v_2, ..., v_t)$ represents a services selection $(S_1, S_2, ..., S_M)$ with an evaluation value $\mathcal{F}'$. The selection is also optimal. Assuming that there is another selection $(\hat{S_1}, \hat{S_2}, ..., \hat{S_M})$ with $\hat{\mathcal{F}'}$ and $\mathcal{F}' > \hat{\mathcal{F}'}$. Let $(\hat{v_1}, \hat{v_2}, ..., \hat{v_t})$ be the corresponding vertices of $(\hat{S_1}, \hat{S_2}, ..., \hat{S_M})$. Then the path length of $(\hat{v_1}, \hat{v_2}, ..., \hat{v_t})$ must be smaller than which of $(v_1, v_2, ..., v_t)$ and this is inconsistent with the fact that $(v_1, v_2, ..., v_t)$ is the shortest path. So the assumption is not true and the shortest path guarantees the optimal solution of optimization problem (7).

Necessity: the necessity represents that if the optimal result of optimization problem (7) is known, denotes $(S_1, S_2, ..., S_M)$, its corresponding path is the shortest path. The necessity can be proved by constructing the path with few steps:

1. *Let v_1 be S_0 and v_{M+2} be S_0'.*
2. *Let v_i be S_{i-1}.*

The path of $(v_1, v_2, ..., v_t)$ is the shortest path. **Q.E.D.**

The Theorem guarantees the equation of the optimization problem (7) and the graph shortest path problem.

4.4 ALGORITHM

With the help of Theorem VI, the optimization problem can be convert into a shortest path problem without any generality. The graph shortest path problem is quite classical problem and lots of algorithms have been proposed in last thirty years. Bellman-Ford algorithm solves the single-source problem if edge weights may be negative [23]. A* search algorithm uses heuristics to try to speed up the search [24]. Dijkstra's algorithm [25] is the most widely used algorithm solving the single-source shortest path problems, and it adopts different kinds of situation and easy to carry out. So it is used in our solution.

Our fully algorithm is presented in Fig. 7. The algorithm is named LA-selection (Location Aware selection). In this algorithm, M is the groups of candidate services, N is the number of candidate service each group. S stores not only the candidate services but also the two database services. The members in the first row of S are both S_0, and the members in the last row are both S_0'. D stores the distances of each area and Omega is the weight coefficient.

The first part of the algorithm converts S to a vertex set V. A considerable improvement is for any group contains two or more services in one area, any of them is feasible. So it is better to use only one vertex to represent these services as the code in line 4. The improvement reduces the number of vertices and promote the efficiency.

The second part of the algorithm is the Dijkstra's Algorithm. Once the last vertex (namely S_0') is reached in the iteration, the loop stops.

The last part is generating the optimal selection as theta.

4.5 FRAMEWORK

Algorithm 1: LA_Selection(S, N, D, Omega)

Input: M groups candidate services S, N is the number of sevices each group, D is the distance matrix and Omega is the weight coefficient

```
1.  k=0;
2.  for i = 1:M+2
3.      for j = 1:N
4.          if not ExistSameArea( S(i,j))
5.              V[k] = S(i,j);    //initialing the vertex set V
6.              V[k].j = j;
7.              V[k].dist = MAX; //initialing distance
8.              k++;
9.          end
10.     end
11. end
12. V[1].dist = 0;
13. si = FindMinDist(V);
14. while si > 0
15.     if IsLast(V[si])    //if V[si] is the last
16.         F'=V[si].dist;//F' is the shortest destance
17.         Break;
18.     else
19.         for i iterats next group
20.         if V[i].dist > V[si].dist + Omega[si] *
             D(Area(V[si]), Area(V[i]))

21.             V[i].dist = V[si].dist + Omega[si] *
                 D(Area(V[si]), Area(V[i]))

22.             V[i].last=si;//recording the vertex in
    path
23.         end
24.         end
25.     end
26.    si=FindMinDist(V);
27. end
28. for i =m:-1:0 //generating the optimal selection as
    theta.
29.      theta[i]=V[si].j
30.      si=V[si].last;
31. end
```

With all the preparation theorems, we now can propose the location aware service selecting framework.

1 Finding out the areas of each candidate services belong to and the distances between each area. These information can be collected by two ways. The first is the information of service provider. It is natural to find that the services of the same provider may be in the same area. The simple way of division of area is classifying services by their providers. The second is the logs of services. Most logs contains the calling time and response time. It is possible to estimate the distance of areas by these logs. In fact it is quite complex problem to estimate the distance of areas and this is not the majority of this paper.

Table 2: Distribution

Service	Belonging Area	Service	Belonging Area	Service	Belonging Area
s_0	A_1				
s_{11}	A_1	s_{12}	A_2	s_{13}	A_1
s_{21}	A_2	s_{22}	A_3	s_{23}	A_4
s_{31}	A_4	s_{32}	A_4	s_{33}	A_1
s_0'	A_1				

Table 3: distance matrix of areas

Distance	A_1	A_2	A_3	A_4
A_1	0	12	22	200
A_2	12	0	24	400
A_3	22	24	0	432
A_4	200	400	432	0

2 Allocating the value of weight coefficients ω_i. The weight coefficient is proportional to the size of data. $\omega_i \propto Da_{ii+1}$. It is determined by the design of service process which is available before the service selection.

3 Converting original process into a graph. The vertices are corresponding to candidate services. The edges connect all the services from one candidate group to each services in next candidate services. This step has been fully discussed in section 4.3.

4 Finding out the shortest path from S_0 to S_0'. The algorithm 1 in Fig.7 is used in this step.

Because of Theorem VI, the service invoking in the shortest path from S_0 to S_0' is the optimal solution $\hat{\theta}$. The Theorem IV guarantees that the optimal solution $\hat{\theta}$ is the optimal selection with lowest transmission cost. Fig. 8 represents an example of converting service selection problem to the shortest path problem. There are three candidate groups (M=3) in this example and each group have 3 services (N=3). The graph is represent in the top right of Fig. 8. It is exactly a full connect graph for each two candidate service group. And the full graph is the n partite graph. The vertices in the shortest path are bold in the right bottom part of Fig. 8. The corresponding services (namely the optimal selection services) are bold in the left bottom part of Fig. 8.

5. CASE STUDY

5.1 INTRODUCTION

In order to understand our model, an example of railway status detection is studied. The railway status detection is a daily work to check the status of each part of the railway to find out the potential disease (the damage and the broken of the railway segment). Daily railway status detection ensures the safety of the railway

We have studied the data of JGX (Beijing-Guangzhou railway line, China). The detecting train collects data (as a record) 4 times per meter. 24 kinds of data are recorded in a

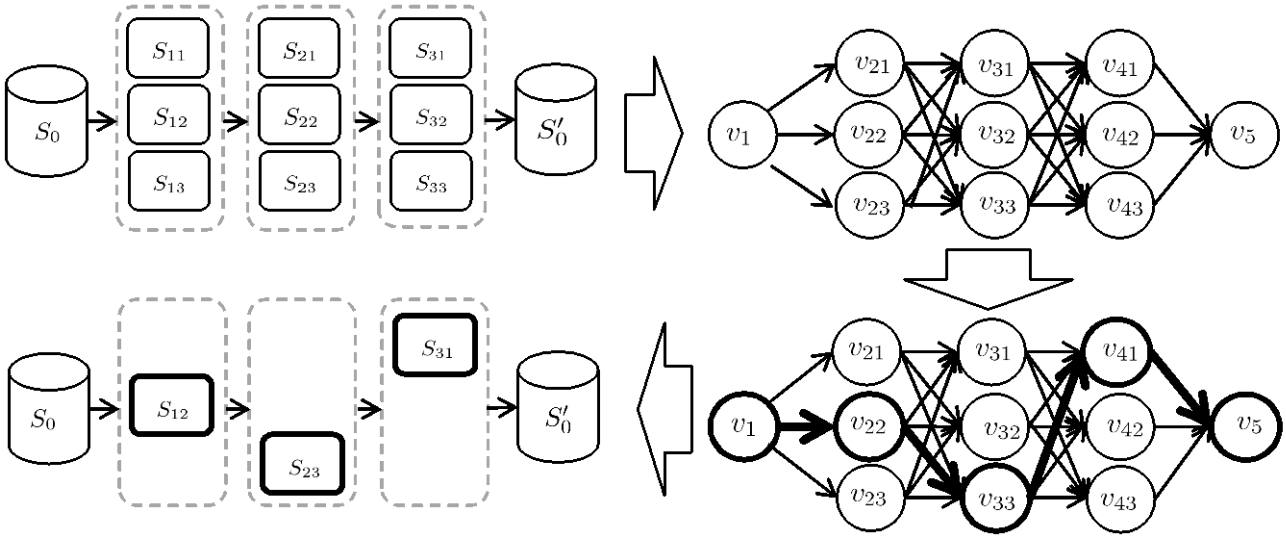

Figure 8: An example of the converting service selection to shortest path problem

record. They are Meters, Flags, Event, Lprf (mm), Rprf (mm), Laln (mm), Raln (mm), Gage (mm), Can't (mm), Xlvl (mm) and etc. The whole length of JGX is 2300 kilometers. There are 2300*1000*4 = 9200000 records are collected in once detection. The size of data in once collection is nearly 200MB. The frequency of detection varies in different seasons. However the size of data collected a month is nearly 12GB. The whole railway detection data are nearly 450GB.

The data were analyzed manually in the past. It took much human labor. Recently some companies have started to develop professional data analyzing service for the railway department while different service providers are expert in different domain, e.g. some have provided Data filtering services, some have provided Data Encryption services, etc. it is necessary to compose these services together to fit the fully requirement. Railway status detection usually has these steps.

- Collection. Detecting the railway by a special detecting train, collecting variety kind of data, including gauges, warps and etc. this data is as large as a few Gigabytes for a long railway. This step does not belong to data analysis and it can only be done by the railway department itself.
- Noise Reduction. Original data contain many noises caused by the measurement error. It is necessary to reduce these noises before deep analysis. This service is S_1.
- Disease Detection. Analyzing the collected data to find out the potential diseases (the disease means the damage or broken in railway segment), by comparing the data with some special patterns of potential diseases. This service is S_2.
- Classification. Different diseases may cause different problems. Some diseases can be ignored, while some diseases may even cause derailment. So it is necessary to classify these diseases into different levels. This service is S_3.

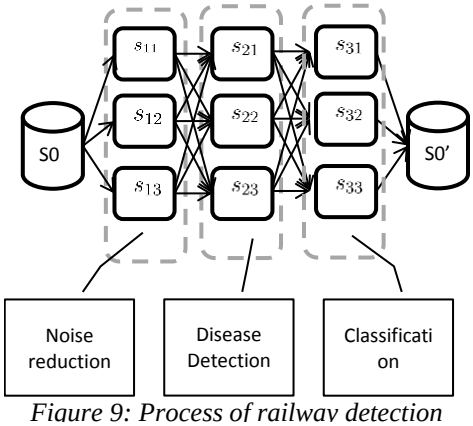

Figure 9: Process of railway detection

In this example, S_0 is the service providing database containing the collected data. S'_0 is the service maintaining database containing different levels of diseases. In reality, S_0 and S'_0 are usually the same. But we still assume that they are different without loss of generality. The whole composite process of railway detection is represented in Fig. 9.

5.2 Preparing Service Locations

There are nine candidate services discovered from 4 service providers. They deployed their service on their server cluster. These server clusters are distributed in different areas. The collected database is in one of the four areas.

Table 4: distance matrix (N stands for unreachable)

	S0	S11	S12	S13	S21	S22	S23	S31	S32	S33	S0’
S0	N	0	120000	0	N	N	N	N	N	N	N
S11	N	N	N	N	120000	220000	2000000	N	N	N	N
S12	N	N	N	N	0	240000	4000000	N	N	N	N
S13	N	N	N	N	120000	220000	2000000	N	N	N	N
S21	N	N	N	N	N	N	N	400	400	12	N
S22	N	N	N	N	N	N	N	432	432	22	N
S23	N	N	N	N	N	N	N	0	0	200	N
S31	N	N	N	N	N	N	N	N	N	N	200
S32	N	N	N	N	N	N	N	N	N	N	200
S33	N	N	N	N	N	N	N	N	N	N	0
S0’	N	N	N	N	N	N	N	N	N	N	N

The distribution of the night candidate services mentioned in Fig. 10 is as follows in Table 2. The distances between each area are listed in Table 3.

5.3 ALLOCATING ω_i

The value of ω_i is decided by the size of data transmitted between services. In the sample of railway detection, data transferring from S_0 to S_1 are the original data. The data transferring from S_1 to S_2 have the same size as the original data. The data used in S_3 and written into S_0' is just the data of disease points, whose size is only near 0.01% of the whole data. A possible allocation of ω_i is ω_0=ω_1=10000, ω_2=ω_3=1.

5.4 CONVERSION

It is needed to convert original sequential process into a graph. The vertices of the graph are the candidate services. The edges are the Connection of service s_{ij} with each candidate service $s_{i+1,k}$ in the next candidate service group. The length of the edge is $dis(s_{ij}, s_{(i+1)k})\omega_i$. There are 24 edges: (s_0, s_{11}), (s_0, s_{12}), (s_0, s_{13}), (s_{11}, s_{21}), (s_{11}, s_{22}), (s_{11}, s_{23}), (s_{12}, s_{21}), (s_{12}, s_{22}), (s_{12}, s_{23}), (s_{13}, s_{21}), (s_{13}, s_{22}), (s_{13}, s_{23}), (s_{21}, s_{31}), (s_{21}, s_{32}), (s_{21}, s_{33}), (s_{22}, s_{31}), (s_{22}, s_{32}), (s_{22}, s_{33}), (s_{23}, s_{31}), (s_{23}, s_{32}), (s_{23}, s_{33}), (s_{31}, s_0'), (s_{32}, s_0'), (s_{33}, s_0'). The distance matrix the graph is described in follows in Table 4, in which, N represents unreachable.

5.5 FINDING SHORTEST PATH

The algorithm is presented in Fig. 7. The M here is 3 and N is also 3. The shortest path is $s_0 -> s_{11} -> s_{21} -> s_{33} -> s_0'$. The shortest distance from vertex S_0 to S_0' is 120012. So the optimal choice plan is selecting the first service in group1, the first service in group2 and the third service in group3.

6. PERFORMANCE STUDY

The time complexity of LA selection algorithm is $\mathcal{O}(MN^2)$, where M is the number of the groups of candidate services, namely the number of rows in Formula (1), and N is the number of candidate services each group, namely the number of columns in Formula (1). In most conditions the number of candidate services is confirmed, which is decided by the service discovery. Therefore, the selection time of LA is only determined by the number of the groups of candidate services (namely M). the time complexity of LA selection is $\mathcal{O}(M)$, which means it is linear time complexity.

We compared the time complexity of our algorithm with the original iteration algorithm mentioned in section 4.3. In order to consider the time complexity of the different sequential process, we evaluated the calculating with M changes from 5 to 5000. At the same time, The number of candidate services each group (namely N), the number of areas and the distance between each two areas are fixed. The comparison result is presented in Fig. 10. The x-axis is M, which is from 5 to 5000 and the y-axis is the logarithm of time. The dash line represents the iteration algorithm. It is exponential time complexity. The solid line represents our algorithm and it is linear time complexity.

7. CONCLUSION AND FUTURE WORK

In this paper, the service location is introduced as a new

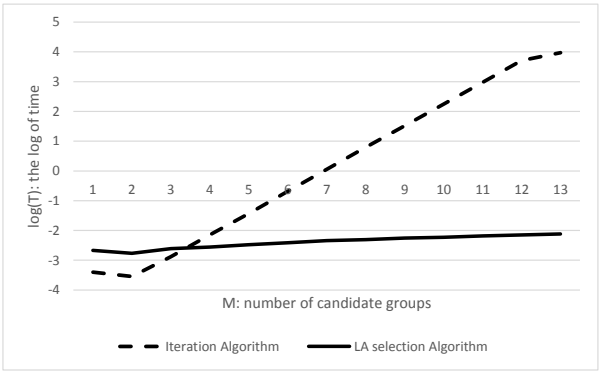

Figure 10 Comparison of algorithms

feature helping to select the optimal services with lowest transmission cost. The classical service selection methods mentioned in [1-3, 5] concerned with the general QoS evaluation in service selection. Data transmission cost is ignored by the classical service selection methods. However, the era of big data brings a new challenge for service selection. The huge size of data makes the transmission time been a majority cost. In this context, a service selection model with transmission time concerned is needed, which is also the motivation of our research. Location is introduced to describe the network context of a service with two distribution assumptions mentioned in section 3.2. The distance of any two services is introduced to represent the transmission speed between these two services. The evaluation function of service selection is the Simple Additive Weighting (SAW) of each distance between two nearby services in the sequential process. In order to solve this optimization problem, the original data are converted into a graph and the vertices invoked in the shortest path from the data reading service (namely s_0) to data writing service (namely s_0'), are the optimal selection of the original optimization problem, which is also the optimal selection. A fully case about the railway status detection is studied and the time complexity of our algorithm is analyzed. It is $\mathcal{O}(M)$ with the number of candidate services each group confirmed. Our future work will mainly focus on unifying the location aware selection model with the classical QoS aware selection model to propose a more general one and extending our model to other control process, such as loop and parallel process.

8. ACKNOWLEDGMENT

This work was supported by National Science and Technology Supporting Program of China (No. 2013BAH10F02), National Natural Science Foundation of China under Grant (No.61272129) and Zhejiang Provincial Natural Science Foundation of China under grant No.LY12F02029

9. REFERENCES

Al-Masri, Eyhab and Mahmoud, Qusay H. (2007). Discovering the best web service, 1257-1258.

Alrifai, Mohammad and Risse, Thomas. (2010a). Efficient qos-aware service composition, Springer.

Alrifai, Mohammad and Risse, Thomas. (2009). Combining global optimization with local selection for efficient QoS-aware service composition, 881-890.

Alrifai, Mohammad, Skoutas, Dimitrios and Risse, Thomas. (2010b). Selecting skyline services for QoS-based web service composition, 11-20.

Ardagna, Danilo and Pernici, Barbara. (2007). Adaptive service composition in flexible processes, *Software Engineering, IEEE Transactions on*, v(33), 369-384.

Ardagna, Danilo and Pernici, Barbara. (2005). Global and local QoS constraints guarantee in web service selection,

Babamir, Seyed Morteza, Babamir, Faeze Sadat and Karimi, Somaye. (2011). Design and evaluation of a broker for secure web service composition, 222-226.

Bakhshi, Mahdi, Mardukhi, Farhad and Nematbakhsh, Naser. (2010). A fuzzy-based approach for selecting the optimal composition of services according to user preferences, 129-135.

Bellman, Richard. (1956). On a routing problem,

Benouaret, Karim, Benslimane, Djamal and Hadjali, Allel. (2012a). WS-Sky: An Efficient and Flexible Framework for QoS-Aware Web Service Selection, 146-153.

Benouaret, Karim, Benslimane, Djamal and Hadjali, Allel. (2012b). Selecting Skyline Web Services from Uncertain QoS, 523-530.

Bo, Yuan, BinQiang, Wang, Bo, Zhao, and Shasha, Song. (2011). Dynamic Methods of Component Composite Service Selection Based on Trust-Aware, Springer.

Cardellini, Valeria, Casalicchio, Emiliano, Grassi, Vincenzo, and Lo Presti, F. (2007). Flow-based service selection forweb service composition supporting multiple qos classes, 743-750.

Dijkstra, Edsger W. (1959). A note on two problems in connexion with graphs, *Numerische mathematik*, v(1), 269-271.

Funk, Caroline, Schultheis, Amelia, Linnhoff-Popien, Claudia, Mitic, Jelena, and Kuhmunch, Christoph. (2007). Adaptation of composite services in pervasive computing environments, 242-249.

Hart, Peter E., Nilsson, Nils J. and Raphael, Bertram. (1968). A formal basis for the heuristic determination of minimum cost paths, *Systems Science and Cybernetics, IEEE Transactions on*, v(4), 100-107.

Jin, Jun, Cao, Yuanda, Zhang, Changyou, Zhou, Ruitao, and Hu, Jingjing. (2011). Quality constraint driven local optimization for efficient service composition, 1-6.

Kang, Guosheng, Liu, Jianxun, Tang, Mingdong, Liu, Xiaoqing, and Fletcher, Kenneth K. (2011). Web service selection for resolving conflicting service requests, 387-394.

Klein, Adrian, Ishikawa, Fuyuki and Honiden, Shinichi. (2010). Efficient qos-aware service composition with a probabilistic service selection policy, Springer.

Kritikos, Kyriakos and Plexousakis, Dimitris. (2009). Mixed-integer programming for QoS-based web service matchmaking, *Services Computing, IEEE Transactions on*, v(2), 122-139.

Kun, Zhang, Hong, Zhang, Liming, Jiang, and Jian, Xu. (2010). Composite Agent Service Selection Algorithm for Non-functional Attributes Based on Simulated Annealing, 101-106.

Lecue, Freddy and Mehandjiev, Nikolay. (2011). Seeking quality of web service composition in a semantic dimension, *Knowledge and Data Engineering, IEEE Transactions on*, v(23), 942-959.

Li, Haifeng, Zhu, Qing and Ouyang, Yiqiang. (2011). Non-cooperative game based QoS-aware Web services composition approach for concurrent tasks, 444-451.

Li, Jing, Zhao, Yongwang, Liu, Min, Sun, Hailong, and Ma, Dianfu. (2010). An adaptive heuristic approach for distributed QoS-based service composition, 687-694.

Lim, Erbin, Thiran, Philippe, Maamar, Zakaria, and Bentahar, Jamal. (2011). Using 3-Way Satisfaction for Web Service Selection: Preliminary Investigation, 731-732.

Lin, Donghui, Shi, Chunqi and Ishida, Toru. (2012). Dynamic service selection based on context-aware QoS, 641-648.

Liu, Yutu, Ngu, Anne H. and Zeng, Liang Z. (2004). QoS computation and policing in dynamic web service selection, 66-73.

Ma, Shang-Pin, Kuo, Jong-Yih, Fanjiang, Yong-Yi, Tung, Chin-Pin, and Huang, Chun-Ying. (2010). Optimal service selection for composition based on weighted service flow and Genetic Algorithm, 3252-3256.

Maximilien, E. Michael and Singh, Munindar P. (2003). Agent-based architecture for autonomic web service selection,

Mohana, Rajni and Dahiya, Deepak. (2011a). Optimized Service Discovery Using QoS Based Ranking: A Fuzzy Clustering and Particle Swarm Optimization Approach, 452-457.

Mohana, Rajni and Dahiya, Deepak. (2011b). Designing QoS Based Service Discovery as a Fuzzy Expert System, Springer.

Nemhauser, George L. and Wolsey, Laurence A. (1988). Integer and combinatorial optimization,

Papadimitriou, Christos H. and Steiglitz, Kenneth. (1998). Combinatorial optimization: algorithms and complexity,

Paulraj, D., Swamynathan, S. and Madhaiyan, M. (2012). Process model-based atomic service discovery and composition of composite semantic web services using web ontology language for services (OWL-S), *Enterprise Information Systems*, v(6), 445-471.

Ran, Shuping. (2003). A model for web services discovery with QoS, *ACM Sigecom exchanges*, v(4), 1-10.

Sipser, Michael. (2006). Introduction to the Theory of Computation,

Soydan Bilgin, A. and Singh, Munindar P. (2004). A DAML-based repository for QoS-aware semantic web service selection, 368-375.

Suleiman, Basem, Da Silva, Carlos Eduardo and Sakr, Sherif. (2011). One Size Does Not Fit All: A Group-Based Service Selection for Web-Based Business Processes, 253-260.

Sun, Qibo, Wang, Shangguang and Yang, Fangchun. (2010a). Quick service selection approach based on particle swarm optimization, 278-284.

Sun, Qibo, Wang, Shangguang, Zou, Hua, and Yang, Fangchun. (2010b). QoS-aware web service selection with the skyline, 928-932.

Tang, Mingdong, Jiang, Yechun, Liu, Jianxun, and Liu, Xiaoqing. (2012). Location-aware collaborative filtering for qos-based service recommendation, 202-209.

Vu, Le-Hung, Hauswirth, Manfred and Aberer, Karl. (2005). QoS-based service selection and ranking with trust and reputation management, Springer.

Wagner, Florian, Ishikawa, Fuyuki and Honiden, Shinichi. (2011). Qos-aware automatic service composition by applying functional clustering, 89-96.

Wang, Pengwei, Ding, Zhijun, Jiang, Changjun, and Zhou, Mengchu. (2011). Automated web service composition supporting conditional branch structures, *Enterprise Information Systems*, 1-26.

Wang, Ping, Chao, Kuo-Ming, Lo, Chi-Chun, and Farmer, Ray. (2011). An evidence-based scheme for web service selection, *Information Technology and Management*, v(12), 161-172.

Yu, Tao and Lin, Kwei-Jay. (2005a). Service selection algorithms for composing complex services with multiple QoS constraints, Springer.

Yu, Tao and Lin, Kwei-Jay. (2005b). Service selection algorithms for Web services with end-to-end QoS constraints, *Information Systems and E-Business Management*, v(3), 103-126.

Zeng, Liangzhao, Benatallah, Boualem, Dumas, Marlon, Kalagnanam, Jayant, and Sheng, Quan Z. (2003). Quality driven web services composition, 411-421.

Zeng, Liangzhao, Benatallah, Boualem, Ngu, Anne HH, Dumas, Marlon, Kalagnanam, Jayant, and Chang, Henry. (2004). QoS-aware middleware for web services composition, *Software Engineering, IEEE Transactions on*, v(30), 311-327.

Zhang, Ming-Wei, Zhang, Bin, Liu, Ying, Na, Jun, and Zhu, Zhi-Liang. (2010). Web service composition based on QoS rules, *Journal of Computer Science and Technology*, v(25), 1143-1156.

Authors

Ying Li received the B.S., M.S. and Ph.D. degrees in computer science from Zhejiang University, China,in 1994, 1997 and 2000, respectively. He is currently an associate professor with the College of Computer Science, Zhejiang University, and a Visiting Professor with the University of California at Santa Barbara. He is currently leading some research projects supported by National Natural Science Foundation of China and National High-tech R&D Program of China (863 Program). His research interests include service computing, business process management and compiler.

Zhiling Luo is the Ph.D. student in College of Computer Science, Zhejiang University, China. He received his B.S. in Computer Science from Zhejiang University in 2012. His research interests include service computing, social network and data mining.

Jianwei Yin is currently a professor in the College of Computer Science, Zhejiang University, China. He received his Ph.D. in Computer Science from Zhejiang University in 2001. He is the visiting scholar of Georgia Institute of Technology, US, in 2008. His research interests include distributed network middleware, software architecture and information integration.