IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 14 November 2013; revised 14 March 2014; accepted 16 March 2014. Date of publication 13 April 2014;
date of current version 4 February 2015.

Digital Object Identifier 10.1109/TETC.2014.2316524

A Dynamic QoS-Aware Logistics Service Composition Algorithm Based on Social Network

YANG YU[1], JIAN CHEN[2], SHANGQUAN LIN[2], AND YING WANG[2]

[1]School of Software, Sun Yat-sen University, Guangzhou 510275, China
[2]School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China
CORRESPONDING AUTHOR: Y. YU (yuy@mail.sysu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 60873162, in part by the Natural Science Foundation of Guangdong Province under Grant S2012010009634, in part by the Research Foundation of Science and Technology Major Project in Guangdong Province under Grant 2012A010800012, in part by the Cooperation Project in Industry, Education and Research of Guangdong Province and Ministry of Education of China under Grant 2011B090400275, and in part by the Research Foundation of Science and Technology Plan Project in Guangzhou City under Grant 12A12051586.

ABSTRACT The public logistics platform aims to provide customers with end-to-end logistics services by finding and composing a huge quantity of web services from logistics service providers. But, traditional service composition required predefined business process so that its flexibility is far from satisfactory in the problem. Path planning can be a solution of finding a suitable business path during service composition, but the search space will increase dramatically with the growth of service quantity and is hard to get a result within a tolerable interaction time. In the context of big data, to quickly build a service path with the optimal global QoS has become a problem demanding prompt solution. Sociologists point out that companies prefer familiar partners in the commercial environment. Using this principle, a concept of partner circle is defined, which can significantly reduce the search space in path planning. Combining path planning with service composition, a PartnerFirst algorithm is presented based on the social network, which is the cooperation network of service providers here. Simulation experiment shows that the PartnerFirst algorithm outperforms current approaches over 10 times in efficiency, with just about 10% loss in QoS. The relationship between efficiency and service quantity of the PartnerFirst algorithm is nearly linear. It proves that using social network in dynamic service composition is efficient and effective.

INDEX TERMS Logistics path planning, service composition, social network, QoS, big data.

I. INTRODUCTION

The Public Logistics Platform can comprehensively integrate the logistics resources, improve the efficiency of supply chain and save cost, stronger the competitiveness of supply chain as well. Given a departure and a destination city with a QoS constraint, to build a logistics path with composed services which can satisfies the QoS constraint and with the optimal QoS has become a problem demanding prompt solution. QoS-aware Service Composition is a research hotspot in recent year, it enable users to quickly composite web service to achieve complex functionality with low cost.

Service-Oriented Computing (SOC) enables loosely composition of services to provide a complex function. There exist many achievements in the research area of service composition [1]. For service compositions, functional and non-functional requirements [2] need to be considered when choosing such services. The latter is specified by Quality of Service (QoS), including attributes such as latency, price, reliability etc. QoS is especially important when many functionally equivalent services are available.

However, traditional QoS-aware Service Composition technology requires predefined business process. This restricts the alternative scheme and limits the service composition optimization to the business process that we can only get the locally optimal solution or even cannot have an acceptable solution. Path Planning can somehow cover this shortage by creating the service path dynamically. As both QoS-aware Service Composition and Path Planning problem has been shown to be a NP-hard optimization problem [3], [4], though by using a combination of service composition and path planning can cover the shortage of flexibility, the price is the dramatically increase of search space. Especially in the big data environment, the search space will become dauntingly huge. The efficiency of such method has received serious challenge. In order to improve the efficiency, we aim to reduce the search space in service composition by using achievements in sociology with the premise of ensuring quality.

For example, considering the logistics service net in Fig.1, Vertex v_i refer to cities and s_i refer to logistics services (e.g. s_2 is a logistics service from v_0 to v_2). Each service has its own QoS. There might be more than one edge between two vertexes, for there might be multiple services with different QoS between two cities, e.g. there exist two services s_3 and s_4 between v_1 and v_3, these two services have the same functionality but with different QoS. Assume that a user Jack wants to deliver a consignment from v_0 to v_7, the given QoS constraint is that the price cheaper than 10 dollars, creditworthiness better than 90% and should be finished in 3 days. So Jack hopes to quickly discover a logistics path from v_0 to v_7, which not only can satisfy the QoS constraint but with the optimal total QoS.

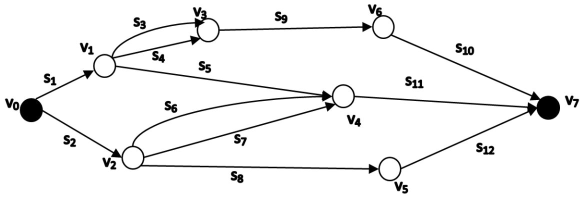


FIGURE 1. A logistics net.

Traditional Service Composition with predefined business process is a solution to this problem. Assume that the predefined path is (v_0, v_1, v_3, v_6, v_7), and then select logistics services between cities to create a composite service that satisfies the QoS constraints proposed by user.

As mentioned before, there might be many other available business processes. The pre-definition of such process will limit our options, e.g. use a process (v_0, v_1, v_3, v_6, v_7), build composite service on it and get the optimal solution, but this solution probably is a locally optimal solution, for the reason that it has not considered many possible processes like (v_0, v_2, v_4, v_7) and (v_0, v_2, v_5, v_7). Consider using Path Planning technology, it can find an optimal path (business process), but the search space will be too huge to get a result within a tolerable interaction time.

While the traditional method has some limitations because of the pre-defined business process, dynamic QoS-aware Service Composition can break the constraints of flexibility and locality by combination use of Service Composition and Path Planning. But then the increasing of search space during business process planning has become a challenge.

Thus, in order to lower the search space and raise efficiency, we make use of the rule '*in the commercial environment, Company prefer familiar partners*' from the sociology to correlate the Dynamic QoS-aware Service Composition with Social Network Analysis. According to the characteristics of the logistics service, the cooperation network is built based on the cooperate history of service providers and define the concept of *Partner Circle*. The choice of partners insists on the principle '*Survival of the fittest*'. Once a partner fails to provide high quality services, the service provider will reduce the probability to cooperate with him, on the other hand, if a service provider has not cooperate with a partner for a long time, the partner relationship will be removed. Besides, to avoid locality and maintain freshness, service providers should appropriately choose high quality partner outside his Partner Circle. This enables them to expand their Partner Circle with high quality service providers. Through the Partner Circle, we propose a PartnerFirst Algorithm. Simulation experiment shows that the algorithm outperforms other current algorithms over 10 times in execution efficiency, with just about 10% losses of QoS.

The structure of this paper is as follows. Section 2 discusses some representative related works. Section 3 defines the cooperation network and gives a QoS evaluation method to model our problem. Section 4 introduces the PartnerFirst algorithm. Section 5 examines the effectiveness of the PartnerFirst algorithm by simulation experiment. Section 6 concludes this paper.

II. RELATED WORK

This section gives a brief survey of the related researches on QoS-aware Service Composition, Social Network Analysis and Path Planning.

A. QoS-AWARE SERVICE COMPOSITION

In the web service composition area, QoS-aware service composition is to select suitable services based on the predefined business process and then bind them to the corresponding task in order to get the optimal QoS. The QoS-aware Service composition problem is a combinatorial optimization problem [5].

QoS-aware service composition problem can be divided into two categories, the global optimization and local optimization problem according to the method used [6]. The local optimization method is to choose services for every task then use the evaluation function to sort the composite service by their QoS and finally get the optimal result by greedy selection. The global optimization method not only considers the QoS of a single service but the total QoS of the composite service in order to get the global optimal solution. The selection of each service is relevant.

1) LOCAL OPTIMIZATION APPROACH

In the aspect of local optimization, [7] achieve service composition based on a QoS feedback mechanism which contains five QoS attributes. It uses the Multiple Criteria Decision Making (MCDM) and Simple Additive Weighting (SAW) to calculate QoS then sort it to select the best one. The weight for each attribute is given by the service users. [8] establishes a QoS attribute matrix. After the normalization, the comprehensive QoS can be calculated through the matrix, so that the optimal solution can be generated. Though the local optimization approaches are efficient, they have the limitation of locality. On one hand, the local optimization approaches separately consider the QoS of each service so that the QoS of the composite service might not satisfy the constraints. On the other hand, because of some nonlinear QoS attributes, the comprehensive QoS do not have the nature of optimal substructure. So we cannot simply add up all the QoS attributes, which means the QoS of the composite service might not be the optimal one though the QoS for every single service is optimal.

2) GLOBAL OPTIMIZATION APPROACH

The global optimization approach can be divided into two categories: the accurate algorithm and the heuristic algorithm. The heuristic algorithm can again be divided into the heuristic algorithm and the meta-heuristic algorithm.

The simplest method of accurate algorithm is the exhaustion method, but it can just solve the problem with minor scale. Following the local optimization algorithm, [7] proposes a method based on integer programming achieve global optimization and implement the AgFlow framework. It uses the state diagram to model the business process as a Directed Acyclic Graph (DAG) to solve the service composition problem and define the concept of *service composition execution path*. [9] divides the QoS-aware service composition problem into two categories: a simple problem with just sequential structure and a complicate problem with complex structure. Then the problem becomes a Multiple Choice Knapsack Problem (MCKP) with single constraint. Finally get the solution by the Pisinger, Bellman-Ford and the shortest path algorithm. [10] proposes a path pruning algorithm based on the service relevancy to improve the method in [7] and [9].

In the heuristic aspect, Branch and Bounded Method [11] such as the A* pruning algorithm [12] uses the heuristic information to estimate the result. It limits the search space by dynamically evaluating the ′prospect′ of the current node to decide whether to choose it or not. [13] proposes the concept of skyline services to quickly reduce the search space. [14] and [15] further improve the efficiency by using the skyline services. The heuristic method relied heavily on the design of the evaluation function which is hard to access and it doesn't perform well when dealing with the massive services.

Meta-heuristic algorithm is a recent hotspot research area. According to the different strategy of '*survival of the fittest*', the Meta-heuristic includes algorithms like Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) etc. [16] proposes a lightweight composition method based on GA to solve the web service composition problem shows that the result is better than Integer programming with better expansibility and is able to deal with nonlinear function, it also supports re-binding through workflow technology. [17] uses GA to optimize the QoS and latency considering the services' location. [18]–[20] discompose the composite service to parallel execution path then model the service composition problem as a multi-objective optimization problem and use improved PSO to solve it. In order to cover the shortage of single meta-heuristic algorithm, [21] combines the ACO and GA, showing that the setting of the key parameter in ACO can be done by GA. Meta-heuristic algorithm has difficulty in balancing the efficiency and QoS when facing service composition problem. It also has the problem in convergence speed when dealing with huge quantity of services.

In conclusion, the challenge in QoS-aware service composition includes:

(1) Most approaches only support compositing services based on a predefined business process.

(2) In the massive services environment, current approaches lack the mechanism to balance the efficiency and QoS leads to the limitation in practical.

(3) Most approaches only concern about the QoS but do nothing in the cooperation relationship.

B. SOCIAL NETWORK ANALYSIS

In the sociology area, there are many classical theories about relationship among people which gives the foundation of our research. In 1960, Milgram found the *Six Degree of Separation* theory [22]. The small world experiment proves that everyone and everything is six or fewer steps away, by introduction from any other person in the world, so that a chain of ''a friend of a friend'' can be made to connect any two people in a maximum of six steps.'' This theory is verified in practical and applied to the areas such as information science, biology and communication technology etc. This theory shows that the distance among actors in a relation network is always short.

About the impact of relationship strength, [23] proposes the weak ties theory shows that the relationship among actors has two states of strong and weak and analyses the strength with the emotion, interaction frequency, reciprocal exchange and the density aspects. The experiment shows that strong relationship often brings many repeated information while weak relationship can become a bridge to convey messages. In the B2B area, [24] points out that good cooperation relationship can improve the degree of trust and reputation. [25] further analyzes the impact of strong and weak relationship. It shows that the strong relationship founded during long-term cooperation can reduce the cost and increase revenue, while weak relationship can more likely to discover new markets for a company and choose appropriate one to increase benefit.

In application, [26] uses a graph to build a business cooperation network and change the partner choose problem to a MOO problem. It optimizes the multiple QoS in the same time. [27] and [28] also use the cooperation relationship to choose partner. It evaluated the result in node attribute and network attribute. [29] applies the SNA to cooperation network. It uses a paper author/co-author relation data and analyzes the common interest and skill to form a team.

C. PATH PLANNING

AI Planning Tools is one way to compose service path. [30] gives a survey about the current AI Planning approaches. [31] proposes the Xplan to automatically build the service workflow and offers a re-planning component which can somehow deal with the service failure situation. [32] proposes a service description language PDDL to translate the goal service into the planning domain and then translated the results back into the service domain. [33] uses a dependency graph to reduce search space by assuming all services are stateless, and then finding path in the graph. [34] proposes HTN-planners SHOP2 to automatically compose DAML-S web services.

III. PROBLEM MODELING

In this section, the problem is modeled as a *logistics service composition model*. It is composed of two parts: A *Logistics Service QoS Evaluation Model* that evaluates the QoS of the logistics service. The other is A *Logistics Service Provider Cooperation Network* that enables the creation of the *Partner Circle*, which is the basis of the PartnerFirst algorithm.

A. LOGISTICS SERVICE QoS EVALUATION MODEL

In order to evaluate the total QoS of the composed logistics services, a QoS criterion is needed. The *Logistics Service QoS Evaluation Model* enables the QoS evaluation of the logistics service. It includes *Service Provider*, *Logistics Service*, *Logistics Service Set* and the *QoS attribute set*. The definition is given below.

Definition 3.1 (Service Provider): A Service Provider u is the one who provide services. Each of the service providers has his own *serviceList*, which records the services he provided.

Definition 3.2 (Logistics Service, Logistics Service Set): $Service = \{s_1, s_2, \cdots, s_m\}$ is a logistics service set where a service $s_i \in Service$ is a quintuple $s_i = (Id, Pro, Dep, Des, QoS)$ refers to *{Identity, Service Provider, Logistics Departure, Logistics Destination, QoS}*.

The QoS of the Logistics service partly reflects the service level. So the QoS is used to guide the service selection. According to the definition above, a QoS attribute set is defined as below.

Definition 3.3 (QoS Attribute Set): A QoS attribute set is a quintuple $\{Pr,Du,Co,Ac,Re\}$, where Pr is the price of the service. Du is the duration of the service. Co is the completeness of the goods when the service has been finished. Ac is the accuracy of the transform order when finished. Re refers to the reputation of the service provider. Note that Pr, Du are summation attributes and Co, Ac, Re are multiplication attributes.

With the QoS attribute set, the QoS for a service can be represented by a vector:

$$(q_{Pr}(s), q_{Du}(s), q_{Co}(s), q_{Ac}(s), q_{Re}(s))$$

As stated before Dynamic QoS-aware Service Composition considers the logistics service to be a service path from departure city to destination city, which forms a logistics service path defined as

Definition 3.4 (Logistics Service Path): $path = \{serviceList, Dep, Des\}$ is a Logistics Service Path. It is an order list with three attributes: *service list*, *departure city* and *destination city*. $pathserviceList = (s_0, s_1, \cdots, s_k)$ is a service list, here k represents the number of service in the path and satisfy:

$$\text{for } 0 < i < k,\ s_i \cdot Dep = s_{i-1} \cdot Des \text{ and } s_{i+1} \cdot Dep = s_i Des.$$

The departure city of a path is represented as $path.dep = s_0.Dep$ and the destination city is represented as $path.des = s_k.Des$.

In order to give a global assessment to QoS, it is necessary to aggregate all the single QoS attribute. Assume that $cs = (ls_1, \ldots, ls_n)$ is a composite service on the logistics service path, the service path QoS is defined as follows:

Definition 3.5 (Service Path QoS): The aggregate QoS for *a path* $serviceList = (s_0, s_1, \ldots, s_k)$ *is*

$$\left(\sum_{i=0}^{k} q_{Pr}(s_i), \sum_{i=0}^{k} q_{Du}(s_i), \prod_{i=0}^{k} q_{Co}(s_i), \prod_{i=0}^{k} q_{Ac}(s_i), \prod_{i=0}^{k} q_{Re}(s_i)\right)$$

With the help of the above definitions, the QoS evaluation model are built as shown in Fig.2

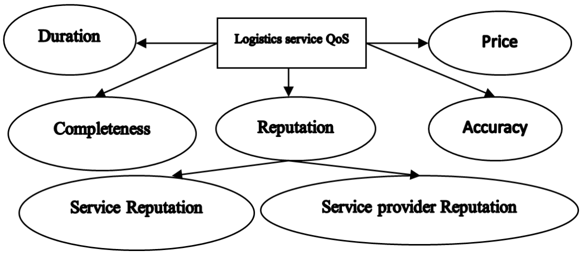


FIGURE 2. The QoS Evaluation Model of logistics service.

For the sake of separately units and value ranges of every QoS attribute, normalization is needed. On the other hand, the nonlinear attributes Co, Ac, Re lead to the inconsistency of global QoS and local QoS. The linear normalization method proposed in [35] is used here to solve the problem of inconsistency. After normalization, the QoS for a single service changes to: $\left(q^*_{Pr}(s), q^*_{Du}(s), q^*_{Co}(s), q^*_{Ac}(s), q^*_{Re}(s)\right)$. For a user given weight $(w_{Pr}, w_{Du}, w_{Co}, w_{Ac}, w_{Re})$, the comprehensive QoS cost is:

$$CQ(s) = w_{Pr} \cdot q^*_{Pr}(s) + w_{Du} \cdot q^*_{Du}(s) + w_{Co} \cdot q^*_{Co}(s) + w_{Ac} \cdot q^*_{Ac}(s) + w_{Re} \cdot q^*_{Re}(s) \quad (1)$$

By the utilization of linear normalization, the comprehensive QoS cost for a logistics service path can represented as:

$$CQ_{path}\left(path\right)=\sum\nolimits_{s_i\in path.serviceList} CQ\left(s_i\right) \tag{2}$$

B. SERVICE PROVIDER COOPERATION NETWORK

A *Logistics Service Provider Cooperation Network* is created based on the cooperation history of service providers.

A *Social Network* is composed of *Social Actors* and their *Relationships*. A *Social Actor* can be a person, a group or an organization [36]. The cooperation Network of service providers can be created according to their cooperation history. [37], [38] indicate that the cooperation experience between team members would have a positive impact. Notice that social relationship will be time decay. That means if the time of the latest cooperation between two service providers from now is too long, it can be considered as they have no social relationship.

Definition 3.6 (Social Network): Social Network can be regarded as an undirected graph $G_{SN} = (V_{SN}, E_{SN})$, where V_{SN} is a set of users and E_{SN} represents the relationship between them. For users u and v, if there exists $(u, v) \in E_{SN}$, indicates that u and v have social relationship. Social relationship is decided by the cooperation history of service providers. Considering the time decay, $G_{SN} = (V_{SN}, E_{SN})$ can be defined as follows:

$$\begin{aligned}&(u, v) \in E_{SN}\ only\ if\\&\qquad u \in V_{SN}, v \in V_{SN}\ and\ t_{max} > t_{cooperation}\left(u, v\right)\end{aligned}$$

Here $t_{cooperation}\left(u, v\right)$ is the time interval from the last cooperation, t_{max} is the threshold value of time. If $t_{cooperation}\left(u, v\right)$ exceeds t_{max}, the history of their last cooperation will not be considered.

The *Cooperation Distance* between vertexes is the edge number on the shortest path between them. The Distance Matrix which denotes the distances between each other can be obtained by Floyd algorithm [39].

To find the neighborhood of a user and get the coefficient degree, a *centric network* can be defined as:

Definition 3.7 (Centric Network): A centric network is a subgraph of a social network, denoted as $CG_{SN} = (CV(u), CE(u))$. $CV(u)$ is the related user of u(including u), $CE(u)$ is the related edge of u. For, $\forall\ v \in CV(u)(v \neq u)$, $h(u, v) = 1$, so the neighborhood of user u is denoted as $neighbor(u) = |SV(u)| - 1$

Base on the centric network, the coefficient of polymerization can be calculated by:

$$coefficient(u) = \frac{2(|CE(u)| - neighbor(u))}{neighbor(u) * (neighbor(u) - 1)} \tag{3}$$

Then the global coefficient of the two user u and v can be obtained by their average value:

$$gCoefficient(u, v) = \frac{coefficient(u) + coefficient(v)}{2} \tag{4}$$

Social Network founds the basis to create a *Partner Circle* for each service provider.

Definition 3.8 (Partner Circle): The Partner Circle for a service provider center is $P_{center} = \{u|h\left(center, u\right) \leq hop_{max}, u \in V_{SN}\}$, where hop_{max} is the max distance to center.

Lastly, the requirement for the logistics transport can be described as:

Definition 3.9 (Logistics Transport Requirement): Logistics Transport Requirement proposed by clients can be represented as a triple: *request* = (*Dep, Des, QoSCon*) refer to *(Departure requirement, Destination requirement, QoS constraint)*.

Finally, with the above definitions, the problem can be formally described as below:

Given a Logistics Transport Requirement *request*, to find a *service path* within a tolerable interaction time where $path.serviceList = (s_0, s_1, \cdots, s_k)$ satisfies the following requirements:

$$path.dep = request.Dep,\ \ path.des = request.Des \tag{5}$$

$$Min \quad CQ_{path}(path) \tag{6}$$

$$s.t. \quad Q_i(path) \leq QoSCon_i\ or\ Q_i(path) \geq QoSCon_i \tag{7}$$

$$\begin{aligned}s.t. \quad &\forall u\,(u \in P \rightarrow u \in P_{center})\\&P = \{s_0.Pro, s_1.Pro, \cdots, s_k.Pro\}\end{aligned} \tag{8}$$

Here, (5) denotes the result path should start with the departure city and eventually reach the destination city. (6) ensures the QoS cost is minimal. In (7), $Q_i\left(path\right)$is an attribute i of the QoS attribute for the composite service; $QoSCon_i$ is the boundary constraint of attribute i. For cost attributes, the target value should be less than the constraint value and for efficiency attributes the target value should be greater than the constraint value. This makes sure that every single QoS constraint is satisfied. (8) denotes that the service provider should search their Partner Circle first in order to reduce the search space to improve efficiency. Here center is the service provider who launches the logistics service, P_{center} is the Partner Circle of center.

IV. SOLUTION

The PartnerFirst algorithm can solve the problem based on the model in section 3. It uses the Partner Circle to improve the A* heuristic search algorithm. This section will explain the main idea and principles of the algorithm by discussing its target, designing tactics and implementation.

[3] has proved that the problem is a NP Complete problem. Using accurate algorithm will take exponential time to get an answer. Though heuristic algorithm can largely improve the efficiency, it still has many limitations especially in the massive services environment; the huge size of search space keeps the time complexity high. Through the use of service provider Cooperation Network, the PartnerFirst algorithm can significantly reduce the search space by preferentially searching services in the Partner Circle. Finally, the algorithm can get a service path with optimal QoS within a tolerable interaction time by reasonable pruning strategy.

A. DESIGNING TACTICS

As mentioned before. PartnerFirst algorithm is an improvement of A* heuristic search algorithm. The main designing tactics include:

1) PREFERENTIALLY SEARCH THE PARTNER CIRCLE

Since the company's cooperation relationship has repetitive features. It means that the service provider who launches the logistics will preferentially search the service in his Partner Circle while searching for a service path. This can not only significantly reduce the search space to increase efficiency, but also lower the cost by cooperating with the familiar partners in the Partner Circle.

2) APPROPRIATE ADD SERVICE OUTSIDE PARTNER CIRCLE

The choice of partners insists on the principle '*Survival of the fittest*'. Once a partner fails to provide high quality services, the probability to cooperate with him will be decreased. If a service provider has not cooperated with a partner for a long period of time, the partner relationship will be removed. Meanwhile, to ensure that the high quality service provider outside Partner Circle can be chosen, service providers should appropriately choose other good services to extend his Partner Circle with service providers who provide high quality services.

This two designing tactics are based on the rule *in the commercial environment, Company prefers familiar partners* and it also reflects the traditional *strong and weak relationship theory* in sociology.

B. IMPLEMENTATION

Base on the model in section 3, PartnerFirst algorithm has three stages. Fig.3 gives an outline of the algorithm.

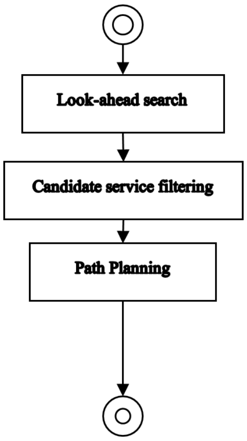


FIGURE 3. **Outline of the PartnerFirst algorithm.**

1) LOOK-AHEAD SEARCH

In the look-ahead search stage, the vertexes and edges that obviously cannot reach the destination or does not satisfy constraints will be quickly discovered and the algorithm will get rid of them. On one hand, it can reduce the search space and raise efficiency; on the other hand it can provide more accuracy heuristic information to the path planning stage. So the convergence can speed up by determining the lower bound of each attribute.

2) CANDIDATE SERVICE FILTERING

This is the main stage to achieve the implementation of the design tactics by using the Cooperation Network to quickly filter out the irrelevant or poorly performed service. Meanwhile, appropriately add services provided by provider outside the Partner Circle as well. This stage build the candidate *composite service graph* of acceptable size to improve efficiency.

3) PATH PLANNING

With the result of previous steps, the Path Planning stage uses the heuristic information to search optimal services. The PartnerFirst algorithm will search the Partner Circle first.

The rest of this section will discuss each of the above stage in detail.

a: LOOK-AHEAD SEARCH

Use the method in [12], the QoS between each city can be calculated by the Dijkstra algorithm [40]. It is called the *ideal QoS* of the city, represented as $bq(curCity, des) = \{bq_{Pr}, bq_{Du}, bq_{Co}, bq_{Ac}, bq_{Re}\}$ and refer to *{best(cheapest) Price, best(lowest) Duration, best Completeness, best Accuracy, best Reputation}*. The *ideal QoS* of the city indicates each QoS attribute's upper bound (for efficiency attributes) or lower bound (for cost attributes) so we can use it to aid the pruning in A* search. When there exists a QoS attribute of the current path cannot satisfy the constraints even by choosing the city with *ideal QoS*, the path will be pruned and then the heuristic information is updated as well.

b: CANDIDATE SERVICE FILTERING

In the context of massive data, directly use the A* algorithm is inefficiency because of the huge search space. The reduction of the search space can be achieved by the utilization of the Partner Circle. Besides, to avoid locality, there is a probability *pextend* of extending services outside Partner Circle. This enables the high quality services outside Partner Circle could be chosen. The pseud code of this stage is shown in Alg1.

Lines 1-8 add the services provided by providers in the Partner Circle into the candidate service list *candidServices*. Lines 8-14 add high quality services outside Partner Circle to the *candidServices* with probability *pextend*.

c: PATH PLANNING

The object of this stage is to find service paths from the candidate service graph created in the previous stage that satisfy the constraints. Considering the characteristic of logistics service, the PartnerFirst algorithm improves the A* heuristic search algorithm based on the model given in

Algorithm 1 GetPartnerServices(H, $pCenter$, hop_{max})

```
Input: H, service provider cooperation distance matrix;
       pCenter, sponsor of logistics service;
       hop_max, max distance from the sponsor.
Output: candidServices. candidate services after filtering
1.  for every service provider v
2.      if $H_{pCenter,v} \le hop_{max}$ then
3.          for all $service \in v.serviceList$
4.              candidServices.add(service);
5.          end for
6.      end if
7.  end for
8.  Group all the services by (departure, destination) and
    add the services with the best QoS of each group to the
    service queue LS
9.  for all $s \in LS$
10.     if rand()<pextend then
11.         candidServices.add(s);
12.     end if
13. end for
14. return candidServices;
```

section 3 and can get a result within a tolerable interaction time.

Cooperation Function

A cooperation Function is needed in order to evaluate the cooperation cost. Here, the cooperation cost is defined based on the centric network.

$$
\begin{aligned}
commCost(u, v) = \big(& (h(u, v) - 1) \\
& * (1 - gcoefficient\ (u, v)) + 1\big) \\
& * cooperationDistance(u, v)
\end{aligned} \tag{9}
$$

Here $gcoefficient(u, v)$ is the global coefficient of user u and v, which can be calculated by (4). Cooperation distance is the shortest path between u and v, it can be obtained by the Floyd algorithm. Notice that $commCost(u, u) = 0$ and if there is no path between u and v, then $commCost(u, v) = maxValue$

Evaluation Function

An evaluation function is the key of a heuristic algorithm. As the goal is to optimize the QoS cost of a composite service, thus QoS cost is the main part of the evaluation function. For the purpose of quickening the convergence speed; the evaluation function also needs to have a degree of destination-oriented. So the evaluation function is defined as

$$
\begin{aligned}
cost\ (path\ (dep, cur)) = \alpha(& CQ_{path}\ (path\ (dep, cur)) \\
& + CQ\ (bq\ (path.des, request.des)) + \beta cos\theta
\end{aligned} \tag{10}
$$

Here α, β denote the weight of QoS cost and distance cost. In our experiment, we set $\alpha = 0.7$, $\beta = 0.3$. $CQ_{path}(path)$ is the QoS cost of current path. $CQ(bq(path.des, request.Des))$ is the QoS cost from the current city of the current path to destination and can be calculated through the *ideal QoS* obtained in the look-ahead search stage. θ is the angle between the vector of departure to current city and the vector of departure to the destination.

Algorithm 2 PartnerFirst(H,$pCenter$,hop_{max}, Service, request)

```
Input: H, service provider cooperation distance matrix;
       pCenter, sponsor of the logistics;
       hop_max, the max distance from the sponsor;
       Service, a set contains all the services;
       request, user requirements.
output: path, logistics service path
1.  Obtained the ideal QoS Bq of every by the method in [12]
2.  candidServices←GetPartnerServices(H,pCenter, hop_max);
3.  path.serviceList=null;
4.  path.dep =   request.Dep; path.des =   request.Des;
5.  priority_queue.add(path);
6.  while priority_queue is not null
7.      curPath = priority_queue.pop();
8.      curCity = curPath.des;
9.      if curPath.des == request.Des then
10.         return curPath;
11.     end if
12.  for all $s \in candidServices$
13.     if s.Dep== curCity then
14.         if curPath contains s.Des then
15.             jump to 12;
16.         end if
17.     for each QoS index$i \in$ I, I = {Pr,Du,Co,Ac,Re}
18.      if $Q_i$ (curPath) and Bq (curCity, request.Des) do
         not satisfy the QoS constraint then
19.                 jump to 12;
20.      end if
21.     end for
22.             newPath.serviceList = curPath;
23.             newPath.add(s);
24.             newPath.city = s.Des;
25.     end if
26.  end for
27.  end while
```

Pruning Strategy

During the path planning process, services with poor performance or unable to satisfy the constraints can be pruned according to the evaluation function. Though evaluation function can evaluate paths through comparison, it is done in runtime. To improve efficiency, certain pruning strategy is necessary.

Strategy 1 (Loop Pruning): If the destination of the current extended service is contained in the current path, then skip this service.

Strategy 2 (QoS Constraint Pruning): If the service chosen to extend the path makes the path cannot satisfy any QoS boundaries, and then prune this path. The upper bound and lower bound of each QoS attribute can be obtained in the look-ahead search stage.

Implementation

Alg2 is the pseudo code of the PartnerFirst algorithm.

Lines 1-2 complete the first two stages which calculate the *ideal QoS* from every city to the destination and the filtered *candidate service list*; Lines 6-27 is the main body of A* algorithm, here *priority queue* is the priority queue for QoS of the path calculated through formula (4); Lines 9-11 judge whether the current path has reached the destination, if yes, the algorithm ends and returns the path; lines 12-26 extend the path; lines 14-16 do the loop pruning and lines 17-21 do the QoS constraint pruning.

V. EXPERIMENT

This section first explains the data generation rule of the simulation experiment and then verifies the effect of the first two stages of the PartnerFirst algorithm. Finally compares the PartnerFirst algorithm with A* pruning algorithm, genetic algorithm (GA), and the three-stage algorithm proposed in our previous work [41]. The comparison is done through the execution time, QoS cost and the cooperation cost. At last, the analysis of the experiment result is given.

4) DATA GENERATION RULE

Considering the characteristic of logistics service and the way of cooperation among service providers, the experiment data is generated through simulation of the behavior among the service providers. The generation rule is shown in Table.1.

TABLE 1. Data generation rule.

services	Generation rule	Generate randomly, every service provider provides services within a certain scope (e.g. service provider in Hong Kong just provides the logistics services to the cities around 300 miles from Hong Kong.)
	QoS	Price and Duration is randomly chosen in [0.5, 1.5] and is related to the distance between departure and destination. Completeness, Accuracy and Reputation is randomly chosen in [0.5, 1.0]
Trade History	Generation rule	Randomly chosen the departure and destination and then randomly select the candidate service set, finally use the A* pruning algorithm to simulate trading.

A. EFFECT OF THE FIRST TWO STAGES

This chapter uses comparison experiments to verify the look-ahead search and the candidate services filtering have positive effects on the problem. The experiment settings are shown in Table.2.

TABLE 2. Experiment settings.

Service Quantity	400 thousnnd
Trading Record	≈40000 times
Execution times	50
pextend	20%
Experiment process	➢ Initialize data ➢ Run the original algorithm and record the result ➢ Run the algorithm with out certain stages and record the result
Comparison aspect	Computation time, QoS cost

1) EFFECT OF DESTINATION-ORIENTED

The destination-oriented is mentioned in formula (10). Comparison experiment is conducted in order to explain the effect of the destination-oriented.

As shown in Fig.4. The execution time reduces in all the cases. The best improvement case is about 1 second. On the other hand, the QoS cost just raise slightly as Fig.5 shows, the most dramatically raise of QoS cost is just about 0.02. These two experiments prove that the destination-oriented can quicken the convergence speed while keeps the QoS cost in a certain level.

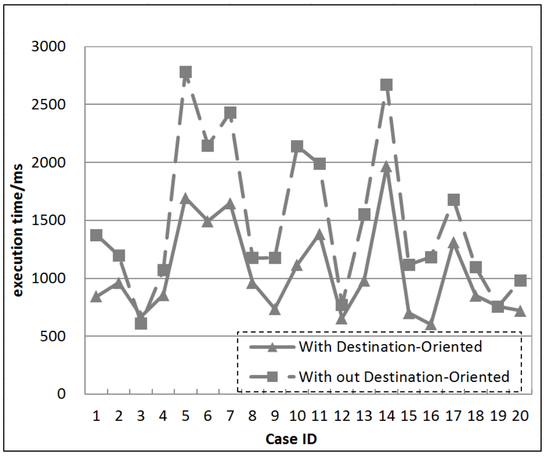


FIGURE 4. The effect of Destination-Oriented on execution time.

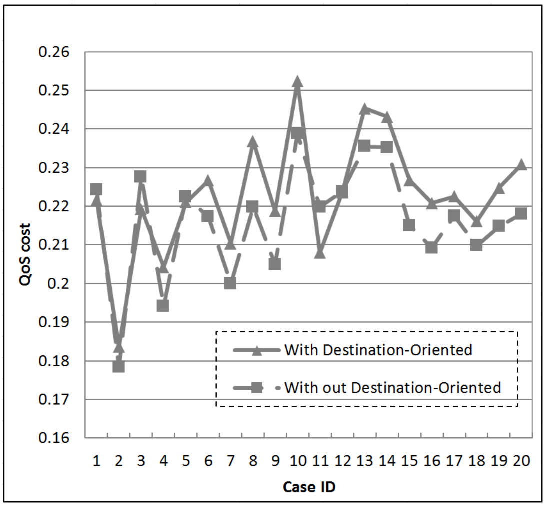


FIGURE 5. The effect of Destination-Oriented on QoS cost.

2) EFFECT OF LOOK-AHEAD SEARCH

The purpose of the look-ahead search is to get rid of services which clearly not satisfy the constraint, and update the heuristic information as well.

Fig.6 is the result on execution time. The execution time reduces significantly in all cases after processing the look-ahead search. The best improvement is about 1.5 seconds. Fig.7 shows that the QoS cost has slightly increase in most cases. But comparing to the improvement in execution time, the total efficiency has risen. With the growth of the service quantity, the effect becomes more clearly. So the look-ahead search has a contribution on the problem.

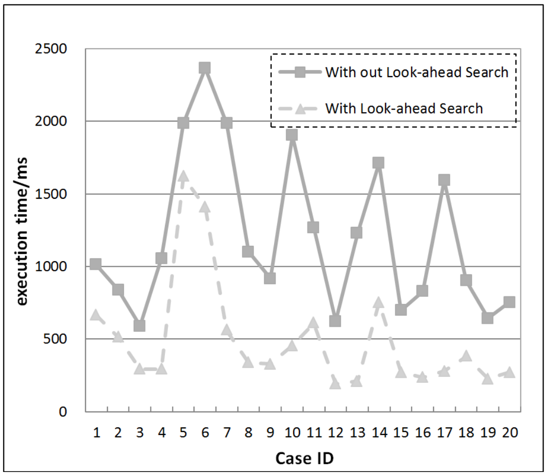


FIGURE 6. The effect of Look-ahead Search on execution time.

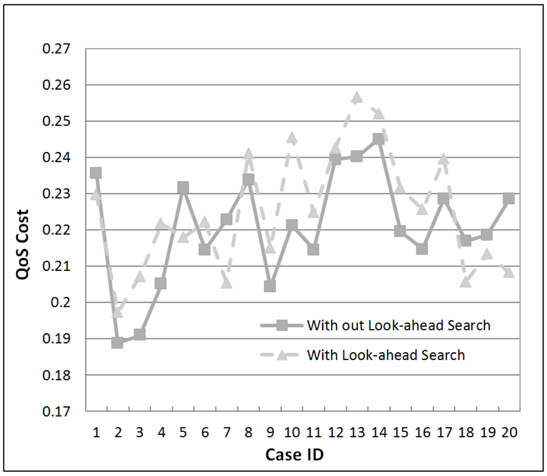


FIGURE 7. The effect of Look-ahead Search on QoS cost.

3) EFFECT OF CANDIDATE SERVICES FILTERING

The Candidate Service Filtering is to reduce the search space by using the cooperation network. It is also the main difference compare to A* pruning algorithm. So the experiment is made by comparing with A* pruning algorithm.

As shown in Fig.8, The execution time are lower than the A* pruning algorithm in all cases after processing the candidate service filtering. At the same time, the QoS cost has slight raised as shown in Fig.9. Again, comparing to the raise of the QoS cost, the efficiency improvement is more clearly and stably. As the main purpose is to raise efficiency with limited lost in cost, the candidate service filtering is relatively efficient.

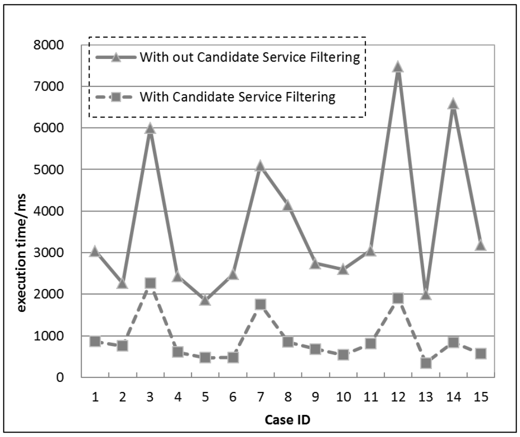


FIGURE 8. The effect of Candidate Service Filtering on execution time.

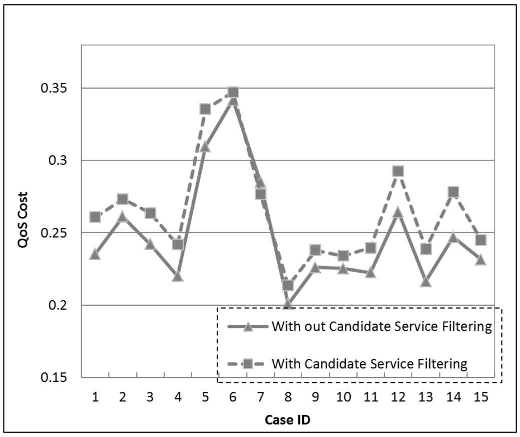


FIGURE 9. The effect of Candidate Service Filtering on QoS cost.

B. COMPARISON EXPERIMENT

This section evaluates the effectiveness of the whole Partner-First algorithm.

1) EVALUATION

In order to compare with the A* pruning algorithm, GA algorithm and the three-stage algorithm, all of these algorithms ought to be executed in the same condition. Firstly, the goal of every algorithm needs to be unified which is to optimize the QoS to keep the QoS cost low. Notice that the GA algorithm here is an extension of [16] by predefining the path using the Dijkstra algorithm. The setting of the experiment is shown in Table.3

2) DIFFERENT SERVICE QUANTITY

Firstly, the experiment is conducted in different service quantity. The result is shown in Fig.10 and Fig.11

TABLE 3. **Experiment settings.**

Service data(10 thousand)	2, 5, 10, 15, 25, 40, 60, 130
Trade history	≈40000 times
Execution time	50
pextend	20%
others	Distance cost = true Cooperation cost boundary = true prune strategy = true
GA parameters	Population numbers and iteration times and crossover rate: adjust according to the service quantity. Keeps the diversity between QoS cost in a low level
Three stage algorithm parameters	Ant number: adjust according to the service quantity. Keeps the diversity between QoS cost in a low level
Experiment process	➤ Initial system data ➤ Run the algorithms in the same environment and Record the results
Comparison aspect	Execution time and QoS cost

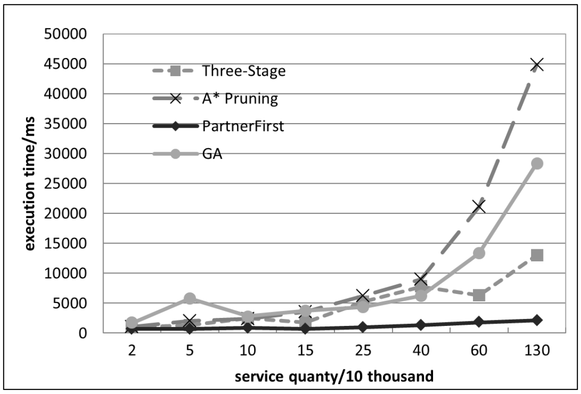


FIGURE 10. **Execution time comparison in different service quantity.**

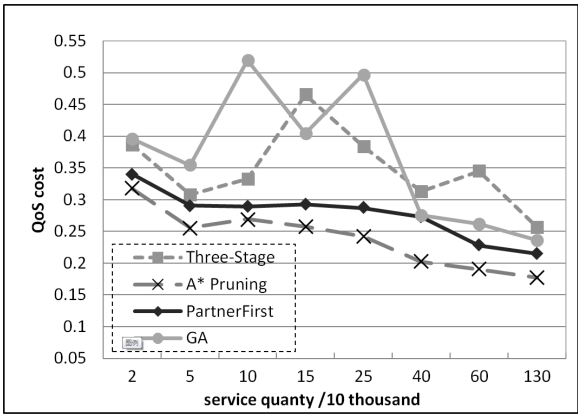


FIGURE 11. **QoS cost comparison in different service quantity.**

As shown in Fig.10 and Fig.11, when the service quantity is less than 50,000, there are slightly difference in execution time among all algorithms and the A* pruning algorithm has the best performance. But with the increase of service quantity, the efficiency of A* pruning algorithm dramatically decline, while the PartnerFirst algorithm is more stabilize and keeps the execution within 4s, that is about 10 times better than GA and three-stage algorithm. On the other hand, though PartnerFirst algorithm is better in execution time, its QoS cost is higher than the A* pruning algorithm but with the max gap just around 0.05. In total, the PartnerFirst algorithm has a much better performance and its convergence speed is faster than GA and three-stage algorithm. Thus the PartnerFirst algorithm can significantly improve the execution efficiency with the premise of low growth of QoS cost and generate the optimal solution within an acceptable period of time. This proves that it is more suitable in the application of recommending service path to users.

3) SAME SERVICE QUANTITY

The next experiment is conducted based on the same service quantity (1.3 million) in order to evaluate the performance of the PartnerFirst algorithm work in different cases. The dots in different shapes denote cases using different algorithms. The cases are randomly generated to run the experiment. Cooperation cost and execution time is used here to evaluate the method.

As shown in Fig.12 and Fig.13, The execution time, the QoS cost and the cooperation cost of the PartnerFirst algorithm is better than GA and three-stage algorithm in most cases. The PartnerFirst algorithm keeps its execution time within 3 seconds while the other two are mostly between 3 to 7 seconds. In the QoS cost aspect, the convergence speed of GA and three-stage algorithm is not ideal, most are distributed in the area above 0.3. Noticed that the cooperation cost equal 0 means a single service can finish the logistics task and does not need cooperation.

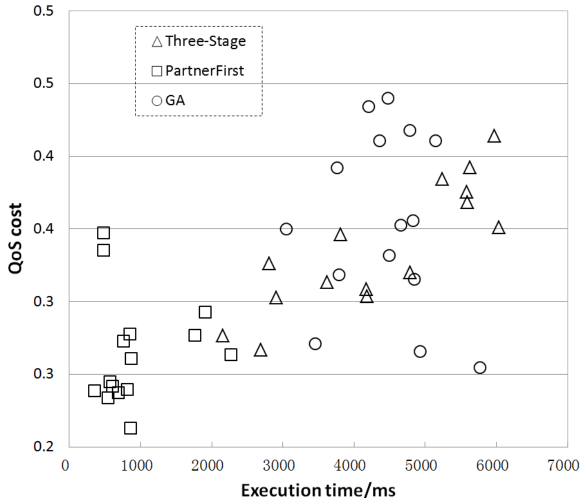


FIGURE 12. **The execution time and the QoS cost comparison in the same service quantity.**

In summary, The PartnerFirst algorithm is better in total. Though it might not get the optimal result, it has a better

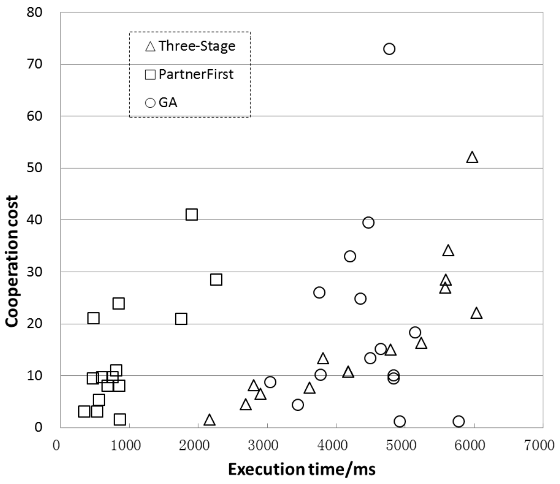


FIGURE 13. **The execution time and the Cooperation cost comparison in the same service quantity.**

efficiency and can get the optimal solution in most cases. These experiments show that the PartnerFirst algorithm is able to solve the problem even in the environment of massive services.

VI. CONCLUSION AND FUTURE WORK

In this paper, by the utilization of the research achievement in the sociology, we propose the PartnerFirst algorithm based on the concept of Partner Circle. The PartnerFirst algorithm achieves the purpose of path planning through the improved A* algorithm, which can significant reduce the search space by preferentially searching the services in the Partner Circle. The experiment shows that the PartnerFirst algorithm can generate an optimal service path within an acceptable period of time (over 10 times better than A*) and the price is just about 10% raise in the QoS cost. It also proves that the usefulness of the algorithm especially in the massive services context. But there are still problems required further research, include:

1) The logistics service contains only sequence pattern. In order to extend the concept of partner circle to normal service composition problem, more complicate patterns need to be considered.

2) The cooperation relationship between service providers just uses some simple result in Social Network Analysis. More intensive achievement requires further discussion.

3) The PartnerFirst algorithm still has room for improvement, like the self-adaptability which requires further consideration of the parameters etc.

REFERENCES

[1] A. Strunk, "QoS-aware service composition: A survey," in *Proc. IEEE 8th Eur. Conf. Web Services*, Dec. 2010, pp. 67–74.

[2] J. O'Sullivan, D. Edmond, and A. T. Hofstede, "What's in a service?" *Distrib. Parallel Databases*, vol. 12, nos. 2–3, pp. 117–133, 2002.

[3] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.

[4] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. Field Robot.*, vol. 26, no. 8, pp. 651–668, 2009.

[5] Y. Dong, H. Ni, H. Deng, and L. Liu, "Service selection strategy offering global optimal quality of service," *J. Chin. Comput. Syst.*, vol. 32, no. 3, pp. 455–459, 2011.

[6] J. Li, J. Xia, W. Tang, J. Zeng, X. Wang, and L. Wu, "Survey on web services selection algorithms based on QoS," *Appl. Res. Comput.*, vol. 27, no. 10, pp. 3622–3627, 2010.

[7] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.

[8] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic web service selection," in *Proc. 13th Int. World Wide Web Conf. Alternate Track Papers Posters*, 2004, pp. 66–73.

[9] T. Yu and K. J. Lin, "Service selection algorithms for web services with end-to-end QoS constraints," *Inf. Syst. E, Bus. Manage.*, vol. 3, no. 2, pp. 103–126, 2005.

[10] Y. Dai, L. Yang, B. Zhang, and Y. Gao, "QoS for composite web services and optimizing," *Chin. J. Comput.*, vol. 29, no. 7, pp. 1167–1178, 2006.

[11] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: Macmillan, 1979.

[12] G. Liu and K. G. Ramakrishnan, "A* prune: An algorithm for finding K shortest paths subject to multiple constraints," in *Proc. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2. Apr. 2001, pp. 743–749.

[13] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based web service composition," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 11–20.

[14] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient web service composition with end-to-end QoS constraints," *ACM Trans. Web*, vol. 6, no. 2, p. 7, 2012.

[15] N. Hiratsuka, F. Ishikawa, and S. Honiden, "Service selection with combinational use of functionally-equivalent services," in *Proc. IEEE ICWS*, Jul. 2011, pp. 97–104.

[16] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "A framework for QoS-aware binding and rebinding of composite web services," *J. Syst. Softw.*, vol. 81, no. 10, pp. 1754–1769, 2008.

[17] A. Klein, I. Fuyuki, and S. Honiden, "SanGA: A self-adaptive network-aware approach to service composition," *IEEE Trans. Services Comput.*, vol. 99, p. 1, Jan. 2013.

[18] S. Liu, Y. Liu, F. Zhang, G. Tang, and N. Jing, "A dynamic web services selection algorithm with QoS global optimal in web services composition," *J. Softw.*, vol. 18, no. 3, pp. 646–656, 2007.

[19] Y. Wang, J. Li, Z. Chen, Y. Meng, and H. Ding, "Web services selection algorithm with QoS global optimal," *Appl. Res. Comput.*, vol. 27, no. 5, pp. 1659–1661, 2010.

[20] G. Kousalya, D. Palanikkumar, and P. R. Piriyankaa, "Optimal web service selection and composition using multi-objective bees algorithm," in *Proc. 9th IEEE Int. Symp. Parallel Distrib. Process. Appl. Workshops*, May 2011, pp. 193–196.

[21] Z. Yang, C. Shang, Q. Liu, and C. Zhao, "A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm," *J. Comput. Inf. Syst.*, vol. 6, no. 8, pp. 2617–2622, 2010.

[22] J. Travers and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, no. 4, pp. 425–443, 1969.

[23] M. S. Granovetter, "The strength of weak ties," *Amer. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, 1973.

[24] E. Li, P. T. Tseng, and E. Lu, "Measuring the strength of partner relationship in B2B EC: An exploratory research," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2007, pp. 6738–6743.

[25] S. G. Lazzarini and T. R. Zenger, "The strength of churning ties: Tie portfolios, tie interdependence, and the dynamics of interorganizational relationships," Working paper, Insper Inst. Education and Research, Brazil, 2007.

[26] J. Crispim and J. Sousa, "A multi-criteria decision support system for the formation of collaborative networks of enterprises," in *Collaborative Networks and Their Breeding Environments* (International Federation for Information Processing). Valencia, Spain: Springer-Verlag, 2005, pp. 143–154.

[27] J. Crispim and J. Sousa, "Multiple criteria partner selection in virtual enterprises," in *Collaborative Networks and Their Breeding Environments* (International Federation for Information Processing). Guimarães, Portugal: Springer-Verlag, 2007, pp. 197–206.

[28] T. Jarimo *et al.*, "Hierarchical multi-attribute decision support approach to virtual organization creation," in *Collaborative Networks and Their Breeding Environments* (International Federation for Information Processing). Valencia, Spain: Springer-Verlag, 2005, pp. 135–142.

[29] M. Cheatham and K. Cleereman, "Application of social network analysis to collaborative team formation," in *Proc. Int. Symp. CTS*, Las Vegas, NV, USA, May 2006, pp. 306–311.

[30] J. Peer, "Web service composition as AI planning—A survey," Univ. St. Gallen, St. Gallen, Switzerland, Tech. Rep., 2005.

[31] M. Klusch, A. Gerber, and M. Schmidt, "Semantic web service composition planning with OWLS-Xplan," in *Proc. AAAI Fall Symp. Semantic Web Agents*, Arlington, VA, USA, 2005.

[32] M. Ghallab *et al.* (1998). PDDL—The planning domain definition language. New Haven, CT, USA, Tech. Rep. TR-98-003/DCS TR-1165 [Online]. Available FTP: ftp.cs.yale.edu/pub/mcdermott/siftware/pddl.tar.gz

[33] S. V. Hashemian and F. Mavaddat, "A graph-based framework for composition of stateless web services," in *Proc. 4th ECOWS*, Dec. 2006, pp. 75–86.

[34] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau, *Automating DAML-S Web Services Composition Using SHOP2*. Berlin, Germany: Springer-Verlag, 2003, pp. 195–210.

[35] Q. Hu, "Composite web service selection system based on QoS linearization and shortest path," *China Sciencepaper*, vol. 7, no. 4, pp. 267–276, 2012.

[36] J. Liu, *An Introduction to Social Network Analysis*. Beijing, China: Social Sciences Academic, 2003, pp. 1–24.

[37] A. Bajaj and R. Russell, "AWSM: Allocation of workflows utilizing social network metrics," *Decision Support Syst.*, vol. 50, no. 1, pp. 191–202, 2010.

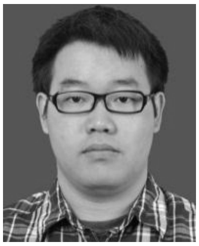
[38] P. Hadaya and L. Cassivi, "The role of joint collaboration planning actions in a demand-driven supply chain," *Ind. Manage. Data Syst.*, vol. 107, no. 7, pp. 954–978, 2007.

[39] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, p. 345, 1962.

[40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[41] Z. Huang, J. Xu, and Y. Yu, "Three-stage algorithm for dynamic service composition," *Comput. Integr. Manuf. Syst.*, vol. 18, no. 8, pp. 1711–1718, 2012.

YANG YU received the bachelor's and master's degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in computer science from Sun Yat-sen University (SYSU), Guangzhou, China, in 1988, 1991, and 2007, respectively. From 1991 to 2002, he was a Senior Engineer and CTO with a software company. Since 2002, he has been a Vice Professor with the School of Information Science and Technology at SYSU, where he has been a Full Professor with the School of Software since 2011. He is a reviewer of several international journals, including the IEEE Transactions on Automation Science and Engineering, *Journal of Supercomputing*, and *Concurrency and Computation: Practice and Experience*. He is a Senior Member of CCF and a member of ACM. His research interests include workflow, service computing, social computing, and software engineering.



JIAN CHEN received the bachelor's degree in computer science and technology from the Guangdong University of Technology in 2012. He is currently pursuing the master's degree with the Department of Computer Science, Sun Yet-sen University. His research interests in regard to service compositions, including genetic algorithm, QoS optimization, dynamic service selection, service reliability, and workflow planning.

SHANGQUAN LIN received the bachelor's degree in computer science and technology from Sun Yat-sen University in 2011, where he is currently pursuing the master's degree with the Department of Computer Science. His research interests include team formation, workflow, social network, and task scheduling.

YING WANG received the B.Sc. degree in computer science and technology from Jiangxi Normal University in 2012. She is currently pursuing the master's degree with the Department of Computer Science, Sun Yat-sen University. Her research interests in regard to service compositions, including efficient QOS optimization and intelligent algorithms, and trust and reputation model.