

## An improved Particle Swarm Optimization Algorithm for QoS-aware Web Service Selection in Service Oriented Communication

Wenbin Wang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications,  
187#, 10 Xi Tu Cheng Rd., Beijing, 100876, P.R.China  
E-mail: buptwwb@gmail.com

Qibo Sun<sup>1</sup>, Xinchao Zhao<sup>2</sup>, Fangchun Yang<sup>1</sup>

<sup>1</sup>State Key Laboratory of Networking and Switching Technology, <sup>2</sup> School of Science,  
Beijing University of Posts and Telecommunications, 187#, 10 Xi Tu Cheng Rd., Beijing, 100876, P.R.China

### Abstract

QoS-aware Web Service Selection is a crucially important issue in Service Oriented Communication which enables communication by integrating communication web services over Internet. Because of the growing number of candidate web services that provide the same functionality but differ in Quality of Service (QoS), it brings more challenges to select a combination of composite services with optimal QoS performance, while satisfying users' QoS constraints. Here, an improved Particle Swarm Optimization Algorithm (iPSOA) is proposed to solve this problem. In order to make the algorithm more suitable for QoS-aware Web Service Selection, firstly we redefine the parameters, such as position, velocity and updating operations. In addition, a Non-Uniform Mutation (NUM) strategy is applied to the global best particle to enhance the population diversity on the purpose of overcoming the prematurity of PSO and a non-uniform mutation function is also designed. Furthermore, the other two contributions are the Adaptive Weight Adjustment (AWA) and Local Best First (LBF) strategies to improve the convergence speed in global and local level respectively. The efficacy of the proposed algorithm for QoS-aware Web Service Selection is illustrated and compared with a modified Genetic Algorithm (GA), QCDSS and PSO, and the results of experimental evaluation indicate that our approach significantly outperforms the existing methods in execution time with better QoS performance while selecting combinations.

**Keywords:** QoS-aware, Web Service Selection, GA, iPSOA, NUM, AWA, LBF

### 1. Introduction

The service-oriented architecture <sup>1</sup> and its realization through web service technologies provide a promising solution for the seamless integration of telecom applications to create new value-added services. Industrial practices witness a growing interest in Service Oriented Communication <sup>2</sup>, with which communication capabilities and network applications are encapsulated as web services (WSs) to enable communication and multimedia delivery by compositing these WSs according to certain logic. With the growing number of

WSs that provide the same functionality but differ in QoS, it becomes more and more important in the selection of component services with regards to non-functional requirements <sup>3-4</sup>.

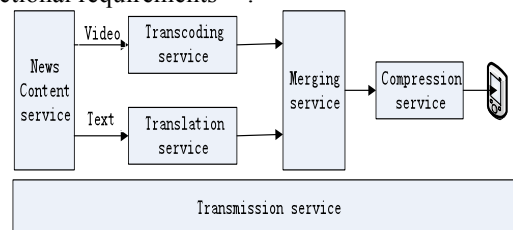


Fig.1. A composite service example: personal multimedia delivery service.

Take the personalized multimedia delivery scenario<sup>5</sup> for example as shown in Fig.1. A cellphone user requests the latest news from a service provider. Available multimedia content includes a news ticker and topical videos available in MPEG 2 only. The news provider has no adaptation capabilities, so additional services are required to serve the user's request: a transmission service for multimedia transmission, a transcoding service for the multimedia content to fit the target format, a compression service to adapt the content to the wireless link, a text translation service for the ticker, and also a merging service to integrate the ticker with the video stream for the limited cellphone display. In this scenario, these services are described in function, which can be implemented by WSs provided by different service providers. We call them tasks, which are composited to provide the whole multimedia delivery service (a composite service).

Towards this given composition request, which can be described in a workflow language (e.g. WS-BPEL<sup>6</sup>), services discovery engine uses existing infrastructure (e.g. UDDI<sup>7</sup>) to locate available web services for each task in the workflow by semantic functional matching between the tasks and service descriptions. As a result, a list of candidate web services is obtained for each task with different QoS parameters. Therefore, they can form different combinations (concrete composite services) for the request, which deliver the same functionality, but differ from each other in QoS performance. The user request can be associated with some end-to-end QoS requirements (like service price, response time and availability). The service composer has to ensure that the aggregated QoS values of the selected services match the user requirements, and obviously, users hope to subscribe the combination of services with optimal QoS performance, while satisfying their QoS constraints. With the growing number of candidate services for each task, the number of combinations increased exponentially, which brings more challenges of the QoS-aware web service selection problem.

This paper proposes an efficient QoS-aware web service selection approach based on an improved Particle Swarm Optimization Algorithm (iPSOA). First, this paper presents a QoS-aware service selection model, which uses multiple criteria decision making with weighted sum model to evaluate QoS performance of services, and we also transforms this model into a general optimization problem with constraints. Then iPSOA is proposed to find the optimal solution in a way that avoids an exhaustive search of the whole solution space for the problem. In addition, to make the algorithm more suitable for this problem, we redefine

the parameters of PSO. Furthermore, in order to improve the performance, mainly three optimal strategies are introduced to construct iPSOA: (1) NUM for the global best particle to overcome the prematurity; (2) AWA strategy in global level; and (3) LBF strategy in local level to improve the convergence speed. Experimental results indicate that the proposed service selection approach based on iPSOA significantly outperforms the recent related schemes in terms of efficiency and effectiveness.

The rest of this paper is organized as follows. In the next section, we firstly introduce QoS-aware service selection model and then discuss the related work. In Section 3, the details of iPSOA are given. And a service selection approach based on it is proposed in Section 4. Then the experimental evaluations and performance comparison with the recent solutions are presented in Section 5. Finally, Section 6 concludes this paper.

## 2. QOS-aware Service Selection Model and Related Work

### 2.1. Definitions of QoS

According to standards of ISO8402<sup>8</sup> and ITU E.800<sup>9</sup>, QoS may include a number of nonfunctional properties which are used to state the quality of WSs and marked by service providers, such as price, response time, availability, reputation, security and so on. However, for the sake of simplicity, four attributes: service price, response time, availability, and reputation are considered on conditions that do not affect the main results of this paper. The definitions and the computation method of them are given as follows.

**Service Price  $P_{ws}$ .** The fee that a service requester has to pay to the Service Provider for the service invocation. Price is measured in pounds (£).  $P_{ws}$  is divided into two parts:  $P_t$ , cost for transmission of request which is omitted in practice, and  $P_s$ , cost of services.

$$P_{ws} = P_t + P_s \quad (1)$$

**Response time  $T_{ws}$ .** The expected delay between the time instant when a request is sent and the time when the result is obtained. Response time is measured in seconds.

$T_{ws}$  of a service typically comprises execution time ( $t_{transaction}$ ) and data transmission time ( $t_{transmission}$ ). It can be calculated by :

$$t_{ws} = t_{transaction} + t_{transmission} \quad (2)$$

$$T_{ws} = \sum t_{ws} / n \quad (3)$$

Generally,  $T_{ws}$  is defined by average  $t_{ws}$  of  $n$  times.

**Availability**  $A_{ws}$ . The probability that the service operation is accessible. Availability is a number in the range  $[0, 1]$ .  $A_{ws}$  is defined by the proportion of service's ready and successful request times to the request times ( $T_r$ ) in certain interval.

$$A_{ws} = T_r / n \quad (4)$$

Where  $n$  denotes the total requested times.

**Reputation**  $Rep_{ws}$ . A measure of the service invocation trustworthiness. It is defined as the ratio between the numbers of service invocations which comply the negotiated QoS over the total number of service invocations. It mainly depends on user's experiences of using services. Different users may have different opinions on the same service. The value of reputation is defined as the average ranking given to the service by  $n$  users.

$$Rep_{ws} = \sum Rep_{ws}^i / n \quad (5)$$

Where  $Rep_{ws}^i$  denotes the value given by the  $i$ th user, which is between 0 and 1.  $Rep_{ws}$  is a number in the range  $[0, 1]$ .

## 2.2. QoS computation of composite services

**Definition 1. Composite Service (CS):** Composed service is defined as a combination of several candidate services that has more powerful function. QoS attribute values of a composite service can be computed by the corresponding ones of candidate services.

**Definition 2. Task:** As basic function unit of composition service, a task contains four attributes: name, function, input, output, which presents an abstract class of some web services with the same or similar function, not a concrete one.

**Definition 3. Candidate Service:** Candidate service is the web service provided by different ISPs (Internet Service Providers), which can be used to fulfill specific tasks. Candidate services may have different QoS attribute values.

Composite services (CSs) are actually a service-based workflow with some atomic functions, and the QoS values of composite services are decided by the QoS values of their component services and the composite models. There are generally four basic composite models for Composite services: Sequence (Fig. 2a) which indicates that tasks are executed in order, Loop (Fig. 2b) which shows that execution of the same task for several times in order, Parallel (Fig. 2c) which indicates that all branches of tasks should run simultaneously, and all branches must be finished before the next node, and Branch (Fig. 2d) which

indicates that next task could begin after any of branches is completed, which have the same function.

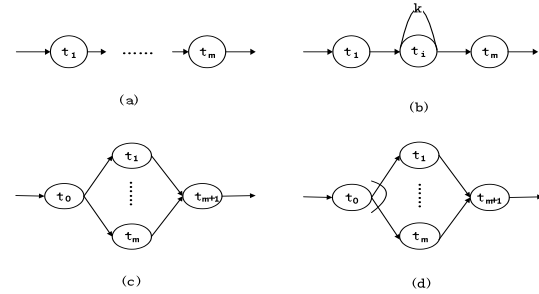


Fig. 2. Basic composite models of composite service

Towards a Composite Service, the QoS values will be perceived in an aggregate manner because it provides service as a whole. Based on the definition of QoS attributes, this study derives aggregative effects of a composite service from its constituent workflow patterns. The aggregate rules of each attribute are given in Table 1. In this paper, we focus on the sequential composition model because other models may be reduced or transformed to the sequential model<sup>15</sup>.

Table 1. QoS aggregate rules for composite services

| Model    | Price                  | Availability            | Response time          | Reputation                |
|----------|------------------------|-------------------------|------------------------|---------------------------|
| Sequence | $\sum_{i=1}^m P_{wsi}$ | $\prod_{i=1}^m A_{wsi}$ | $\sum_{i=1}^m T_{wsi}$ | $\prod_{i=1}^m Rep_{wsi}$ |
| Loop     | $kP_{ws}$              | $A_{ws}^k$              | $kT_{ws}$              | $Rep_{ws}^k$              |
| Parallel | $\sum_{i=1}^m P_{wsi}$ | $\prod_{i=1}^m A_{wsi}$ | $\max\{T_{wsi}\}$      | $\prod_{i=1}^m Rep_{wsi}$ |
| Branch   | $P_{wsi}$              | $A_{wsi}$               | $T_{wsi}$              | $Rep_{wsi}$               |

## 2.3. QoS-aware web service selection model

Various workflow patterns shape the candidate web services into a virtual composite service whose QoS performance will be perceived by a service requester in an aggregated manner. Furthermore, there are many candidate web services for each task, and a Composite Service may be composed of many tasks, so there are enormous combinations for a Composite Service.

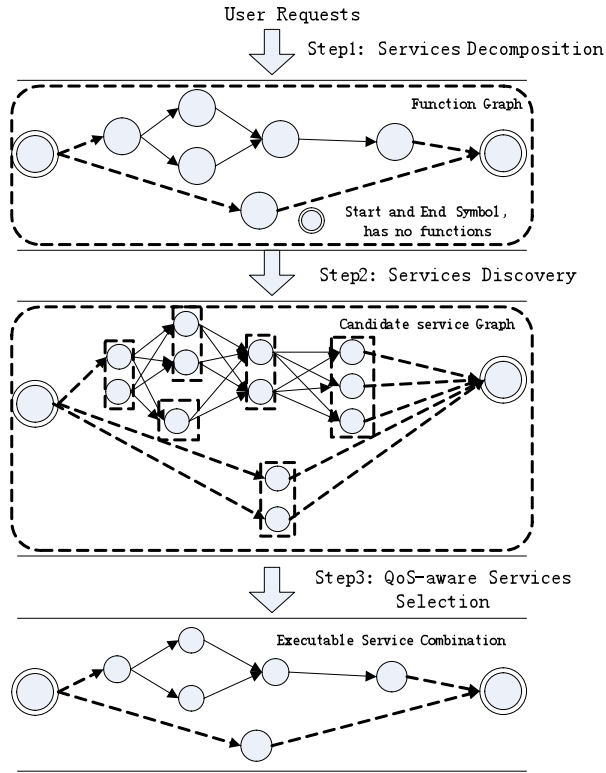


Fig. 3 Procedure on QoS-aware Web service composition

Let us continue the personal multimedia delivery service example in the *Introduction*. The service composition process is illustrated as Fig.3. If each task can be complemented by 20 candidate web services, there will be  $20^6$  composition schemes. Among the enormous composition schemes, QoS-aware web service selection must aim at two objects: (1) schemes should meet users' global QoS constraints; and (2) global QoS performance gets better. Therefore, QoS-aware Web Service Selection problem is transformed into a multi-objects optimization problem with constrained: four attributes of QoS are treated as objects, that is, lower price, shorter response time, higher availability and reputation, and service price and service response time are treated as two constraints.  $P_0$  and  $T_0$  are supposed to maximum price and maximum response time respectively that the user can bear, then global QoS optimization web service selection can be formalized as follows:

$$\begin{cases} f(x) = F(\min P, \min T, \max A, \max Rep) \\ \text{subject to } P \leq P_0, T \leq T_0 \end{cases} \quad (6)$$

Actually, there are many QoS attributes, and the most important is to map them to the optimization model,

therefore this model can be extended to more objects optimizations and more constraints.

Multi-objects optimization problem with constraints is characterized by conflicts between various objectives, which results in that the solution is not a single one but a set. Therefore, QoS performance score would be an effective way to make a decision in the solution space.

For the given Composite Services, the combination with higher score has better QoS performance than others. Therefore, the QoS-aware service selection problem is transformed into an optimization problem with constraints illustrated as follows.

$$\begin{cases} \text{Max}\{Score(CS)\} \\ \text{subject to: } Q \leq Q^0 \end{cases} \quad (7)$$

Where  $Score(CS)$  denote the performance score of a composite service, and  $Q_i^0$  refers the users' constraints of QoS values. For example, user would not pay more than 0.5 pounds for this service request and the services' response time cannot be more than 2 seconds. The goal of QoS-aware Web Service Selection is to select one component service from candidate services of each task so that the aggregated QoS scores of Composite Services can be maximized as well as the constraints is satisfied.

#### 2.4. Related work

QoS-aware web services selection plays an important role in web services composition<sup>1</sup>. To figure out QoS-aware web services selection, some approaches were proposed with the help of semantic web<sup>11</sup> and the QoS attributes computation<sup>10, 12-19, 24</sup>, and the latter approaches are more suitable to satisfy users' QoS requirements<sup>19</sup>.

In [12], local optimization approaches were presented, which selected web services according to a desired characteristic for each task, e.g., the price of a single WS was lower than a given threshold. However, local approaches only guaranteed the local QoS constraints and could not satisfy user's requirements at a global level. Ref.s [10, 13] discussed a global planning approach for selecting composite web services. They used (mixed) linear programming techniques to find the optimal selection of component services. Similar to this approach, Ref. [14] extended the linear programming model to include local constraints. Linear programming methods are very effective when the size of the problem

is small. However, these methods suffered from poor scalability<sup>15</sup>.

Since this problem is NP-hard<sup>16</sup>, many heuristic approaches have been carried out to do this. In [16-17] the authors proposed GA to find a near-to-optimal solution more efficiently than exact solutions, where the formulated optimization problem is an iterative procedure based on a constant-size population, which make GA be apt to very effectively perform global search. However, prematurity phenomenon of GA is the main drawback of being trapped into a local optimal solution<sup>18</sup>, which would lead worse QoS performance of the Composite Service. Therefore, some works<sup>18-19</sup> were done to deal with the prematurity of GA by population diversity keeping and initial population enhancing policies. However, the fact that the random evolutionary information of GA population is not so effectively utilized that it makes the algorithm inefficient in execution time. The PSOA<sup>20</sup> is another population-based heuristic global optimization technology, which is more efficient because the global and local best particles all guide the solutions evolution, i.e., the evolutionary information of PSOA is fully exploited<sup>21</sup>. However, fast convergence speed also means to fall into local optimum easily<sup>23</sup>. That is why the application of PSOA to QoS-aware Web Service Selection has not been fully explored in literatures<sup>24</sup> and this also constitutes the main focus of this research effort.

### 3. An Improved Particle Swarm Optimization Algorithm (iPSOA)

The PSOA was proposed firstly by Eberhart and Kennedy in 1995<sup>20</sup>, which is inspired by the social behavior of bird flocking or fish schooling, which is used for finding optimal regions of complex search spaces through the interaction of individuals in a population of particles. The iterative procedure of PSOA make it fit well to solve NP-hard optimization problems<sup>21-25</sup>. A PSOA maintains a swarm of particles and each individual includes three parameters, which are the current position  $X_i$ , the previous best position  $X_{pbest}^i$ , and the velocity  $V_i$ . The best solution found by each particle so far is stored in  $X_{pbest}^i$ , while the best of the swarm in  $X_{gbest}$ . New positions of particles are decided by the current positions and velocity. In essence, the trajectory of each particle is updated according to its own flying experience as well

as to that of the global best particle in the swarm. These parameters of particles updates as follows:

$$V_{i+1} = wV_i + c_1r_1(X_{pbest}^i - X_i) + c_2r_2(X_{gbest} - X_i) \quad (8)$$

$$X_{i+1} = X_i + V_{i+1} \quad (9)$$

Where  $w$  is the inertia weight,  $c_1$  is the cognition weight and  $c_2$  is the social weight; and  $r_1$  and  $r_2$  are two random values uniformly distributed in the range of  $[0, 1]$ .  $X_{gbest}$  and  $X_{pbest}^i$  can be obtained by calculating  $F(X_i)$ , which is called fitness function with the function of evaluating how better the positions of particles are for the target.

The PSOA is usually used to solve multi-objective optimization problems in continuous space because of its advantages such as intuition, easy to operate, easy to distributed execution, and efficiency. Good effects are achieved in the multi-objective optimization with multi-constrained problems by PSOA, but in the discrete domain, the application of the algorithm has not achieved desired results<sup>23</sup>. In addition, the swarm converges rapidly within the intermediate vicinity of the global best. However, such a high convergence speed often results in: (1) the lost of diversity and (2) premature convergence if the global best corresponds to local optima, which will lead to bad quality of solution. Therefore, there are three important issues must to deal with: (1) how to definite the parameters and operations in PSOA to make it suitable for QoS-aware Web Service Selection; (2) the prematurity phenomenon easily occurred for PSOA which may cause local optimum to the optimization problem, namely that the QoS performance of selected combination is not so ideal; (3) how to improve convergence speed affected by strategies for overcoming prematurity.

Thus, after definition of the parameters and operations to make PSOA suitable for QoS-aware Web Service Selection, a NUM strategy for the global best particle is introduced in order to overcome the prematurity phenomenon, yet mutation for the global best particle may slow the convergence speed, and then AWA strategy is applied to improve it in global level, which can improve the algorithm's execution efficiency. The following subsections describe the details of the application of iPSOA for QoS-aware Web Service Selection problem.

### 3.1. Definition of parameters and operations of PSOA

The PSOA is a general algorithm, when used to solving problems, the parameters and operations must be redefined according to the feature of problems. To make PSOA suitable to solve QoS-aware Web Service Selection, suppose that  $X_i = (x_1, x_2, \dots, x_m)$  denote combinations of a composite service, where  $x_i$  is the number of services selected, which are expressed as positions of particles in PSOA; and velocity  $V_i = (v_1, v_2, \dots, v_m)$  is used to guide the updating direction of particles, where  $v_i = 0$  or 1. The addition, multiplication and subtraction operations are redefined as follows.

**Subtraction ( $\ominus$ ):**  $X_i \ominus X_j$  indicates the distance of the position  $X_i$  and  $X_j$ . The value of the corresponding dimensions of result is 1 when the values of the corresponding elements of  $X_i$  and  $X_j$  are the same. Otherwise it is 0. For example,  $(1, 2, 3, 4, 4, 5) \ominus (1, 3, 2, 4, 5, 5) = (1, 0, 0, 1, 0, 1)$ .

**Addition ( $\oplus$ ):** is an operator that denotes the selection direction, and the result is a new velocity.  $P_1 V_1 \oplus P_2 V_2 \oplus \dots \oplus P_n V_n$  expresses that the result of the formula keeps  $V_1$  with probability  $P_1, \dots$ , and keep  $V_n$  with the probability  $P_n$  in corresponding dimension, where  $P_1 + P_2 + \dots + P_n = 1$ . For example,  $0.1(1, 0, 0, 1, 1, 1) \oplus 0.9(1, 0, 1, 0, 1, 0) = (1, 0, *, *, 1, *)$ . \* denotes uncertain to be 0 or 1. In this example, the first \* is equal to 0 with probability of 0.1, and 1 with probability of 0.9.

**Multiplication ( $\otimes$ ):** denotes updating process of particle's position, and the result is a new position. Operating rules: if the value of  $V_i$  equals to 1, the value of  $X_i$  will be kept; and otherwise, the value of  $X_i$  will change. Let's take  $(1, 2, 4, 3, 2, 5) \otimes (1, 0, 1, 0, 0, 1) = (1, *, 4, *, *, 5)$  as an example, and the value of \* is decided by the strategy adopted by algorithm, whose value will be selected randomly or in heuristic rules.

Based on the redefinition, the new velocity and position for each particle are determined according to the velocity and position update equations given by (10) and (11).

$$V_{i+1} = P_1 V_i \oplus P_2 (X_{pbest}^i \ominus X_i) \oplus P_3 (X_{gbest} \ominus X_i) \quad (10)$$

$$X_{i+1} = X_i \otimes V_{i+1} \quad (11)$$

### 3.2. Optimal strategies in iPSOA

#### 3.2.1 Non-Uniform Mutation (NUM) for the Global Best Particle

In PSOA, all other particles search along the direction of the global best one except for itself, whose evolution direction is blind, so it's easy to miss the optimal solutions and to result in prematurity. In [23], the mutation idea in PSOA is introduced to the movement of the global best particle in order to explore more new solutions in solution space. However, the probability of the mutation is very sensitive. Higher mutation probability expands the search space and makes it possible to find better solutions, but it reduces the convergence speed at the same time. Lower mutation probability makes the algorithm converge faster, but the opportunity of falling into local optima will increase. In this section, a non-uniform mutation function is designed based on our previous work<sup>25</sup>. The function has the characteristics of overall decreasing, but with random fluctuation in the execution process. The function is given as follows:

$$f(k, r) = \sigma * (1 - r^{(1 - \frac{k}{K})^b}) \quad (12)$$

Here  $K$ ,  $\sigma$  and  $b$  are constants.  $k$  and  $r$  are variables of the current generation and random number in  $(0, 1)$  respectively. How the mutation function  $f(k, r)$  depends on generation number  $k$  is analyzed as follows. Of course, generation number  $k$  is a discrete variable. Without loss of generality, we suppose  $k$  being a continuous variable in  $(0, K)$ , and obviously,  $f(k, r)$  is differential on  $k$ . At the same time, the random variable  $r$  is actually independent with generation number  $k$ . Therefore the mutation function  $f(k, r)$  is in essence a dualistic function on  $\{(k, r) | k \in (0, K), r \in (0, 1)\}$ .

Its partial derivative on  $k$  is shown as follows:

$$\frac{\partial f(k, r)}{\partial k} = \sigma r^{(1 - \frac{k}{K})^b} (1 - \frac{k}{K})^{b-1} \frac{b \ln r}{K} < 0 \quad (13)$$

$$\lim_{k \rightarrow K^-} f(k, r) = 0 \quad (14)$$

It can be seen that mutation probability "monotonously" decreases and approaches to zero supposing  $r$  being constant. However, as we know that  $r$  is an independent and uniform random variable in  $(0, 1)$  and the derivative on  $r$  of  $f(k, r)$  is:

$$\frac{\partial f(k, r)}{\partial r} = -\sigma * (1 - \frac{k}{K})^b * r^{(1 - \frac{k}{K})^b - 1} < 0 \quad (15)$$

Here  $r$  is elite to make the mutation decrease, but not monotonously. Namely, it reduces and wanes to zero in sense of probability. Thus this approach helps to reduce the likelihood of premature convergence and guides the search toward the promising domain area. Fig.4 illuminates the varying scheme of non-uniform mutation versus the generation number with the parameters ( $\sigma = 0.5, b = 5, K = 2000$ ).

From Fig.4, we can know that the mutation probability decreases in the overall with random fluctuation through the generations, and the mutation probability is near to 0 which is good for the convergence after the 1400<sup>th</sup> generation.

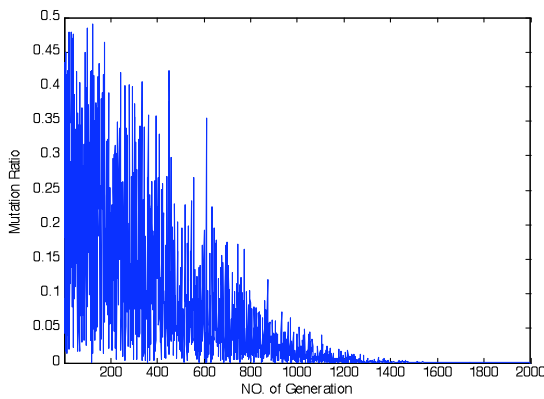


Fig. 4. Mutation probability v.s. generations.

### 3.2.2 Adaptive weight adjustment (AWA) strategy

The velocity  $V_i$  determines the moving direction and speed of particles. It may be trapped by the local optimal solution if the  $V_i$  is too fast, or it may affect the convergence speed for the slow velocity. From (8) and (10), we know that velocity  $V_{i+1}$  is determined by three parts, and the values of  $P_1, P_2, P_3$  would impose impacts on them directly. In PSOA, the  $P_1, P_2, P_3$  are constants or random, which are not well designed for global control of the velocity. Therefore, a dynamic weight adjustment mechanism is designed in order to make particles adjust their velocity adaptively according to ratio of the fitness values as following:

$$P_1 = \frac{F(X_i)}{F(X_i) + F(X_{pbest}^i) + F(X_{gbest})} \quad (16)$$

$$P_2 = \frac{F(X_{pbest}^i)}{F(X_i) + F(X_{pbest}^i) + F(X_{gbest})} \quad (17)$$

$$P_3 = \frac{F(X_{gbest})}{F(X_i) + F(X_{pbest}^i) + F(X_{gbest})} \quad (18)$$

Generally speaking, the inequation  $F(X_i) \leq F(X_{pbest}) \leq F(X_{gbest})$  is established and all other particles will close to the global best one with a certain speed. In this process,  $X_{gbest}$  obviously plays the most important role,  $X_{pbest}^i$  for the next place, and  $X_i$  the least.

### 3.3. Procedure Description for iPSOA

The procedures of iPSOA are given as following.

**Step1** Initialization: Particle swarm size  $M$ , maximal generation number  $K$ . Generating initial position  $X_0$  and velocity  $V_0$  for each particle randomly, and checking the feasibility of particle's position (position must meet QoS constraints).

**Step2** Iteration control: if ( $k == K$ ), goto Step4; else continue;

**Step3** Iteration:

**Step3(1):** Calculating fitness values  $F(X_i)$  for each particle;

if ( $F(X_i) > F(X_{pbest}^i)$ )  $X_{pbest}^i = X_i$ ;

if ( $F(X_{pbest}^i) > F(X_{gbest})$ )  $X_{gbest} = X_{pbest}^i$ ;

**Step3(2):** Updating position  $X_i$  and velocity  $V_i$  according to equation (10) and (11);

**Step3(3):** NUM for the global best particle, generating a new position for it;

**Step4** Output: outputting composition scheme and the corresponding fitness values.

**End.**

## 4. QOS-aware Service Selection Approach Based On iPSOA

Since QoS-aware Web Service Selection is a proved NP-hard optimization problem<sup>16</sup>, if the positions  $X_i$  are considered as the solutions of the objective optimization problem, which are mapped to combinations of a Composite Service, and  $F(X_i)$ , the fitness values, as performance scores of combinations in QoS-aware Web Service Selection problem, the combinations of high QoS performance would be selected through iPSOA after iterative procedure. However, there are following issues to be solved.

### 4.1. Coding Strategy

First, a task number is given to each task according to the model of a composition service in a certain order, as well as a candidate number to every candidate web service in each task. For a given composite service, the

task number in certain model is definite, so we use a fixed-length integer coding strategy to identify a composite scheme, which can be map to positions of particle. Let us take the example in Fig.1 and Fig.2 as an example. There are six tasks in this composite service, so the length of the position is 6. Supposing there are ten candidate services for each task, then a composite scheme (1,3,4,9,5,8) shows that task 1 selects the No.1 candidate service, and task 2 selects the No.3 candidate service, and so on.

#### 4.2. Fitness Function Design

Fitness values are used to evaluate the performance of current positions of particles to guide other particles to follow the global best particle. Toward different problem, the fitness function is designed separately. In the QoS-aware web service selection problem, the goal of it is to select one component service from candidate services of each task so that the aggregated QoS performance of Composite Service can be maximized as well as the constraints are satisfied. Since conflicts may happen between various objectives, which results in that the solution is not a single one but a set. Thus a performance score which can deal with the trade-off among objectives need to be design, which is also needed for the fitness evaluation in iPSOA.

In service composition, the fitness function must reflect the global QoS of a composite web service. In this paper, a weighted sum of each attributes for the QoS of a composite web service is adopted.

$$f(x) = \sum_{i=1}^4 w_i Q_i \quad (19)$$

Here  $w_i$  denotes the corresponding weight for a certain attribute of QoS,

and  $\sum_{i=1}^4 w_i = 1$ .  $Q_1, Q_2, Q_3, Q_4$  denote

$P, A, T, Rep$  respectively mentioned in section However the attribute values is not on the same level and have different units, such as service execution time with the unit of *ms* (millisecond) and service price with the unit of \$ (U.S dollars). If the values of  $Q_i$  is calculated directly by formula (19), it will inevitably lead to unfair evaluation. From another point of views, different attributes will lead different effects for the integer evaluation. For example, service price and service execution time will have negative impacts on evaluation, whose values are the larger, the worse. However service reliability and service availability have

positive impacts, whose values are the larger the better. Therefore,  $Q_i$  need to be normalized using the method proposed by [10]:

$$Q_i' = \begin{cases} \frac{Q_i - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}}, & Q_i^{\max} - Q_i^{\min} \neq 0 \\ 1, & Q_i^{\max} - Q_i^{\min} = 0 \end{cases} \quad (20)$$

$$Q_i' = \begin{cases} \frac{Q_i^{\max} - Q_i}{Q_i^{\max} - Q_i^{\min}}, & Q_i^{\max} - Q_i^{\min} \neq 0 \\ 1, & Q_i^{\max} - Q_i^{\min} = 0 \end{cases} \quad (21)$$

Here  $Q_i^{\max}, Q_i^{\min}$  denote the maximum and minimum of  $i$ th attribute of QoS for all the candidate services. So, the fitness function can be expressed as:

$$f'(x) = \sum_{i=1}^4 w_i Q_i' \quad (22)$$

Here  $\sum_{i=1}^4 w_i = 1$ ,  $0 < w_i < 1$ ,  $0 \leq Q_i' \leq 1$ . Generally,

end users usually assert their function requirements as well as global constraints, e.g.  $P \leq 10, T \leq 100$ . So, the composite scheme will be rejected firstly if the QoS violates the constraints.

#### 4.3. Local Best First (LBF) strategy

In PSOA, the candidate service is selected randomly for each task, which would cost lots of time until the right one is selected, and this reduce the convergence speed of PSOA. In order to improve the convergence speed, we presented a LBF strategy here enlightened by the combining global optimization with local selection approach in [15].

Intuitively, the combinations will have high QoS performance scores if every task has high QoS value. So, the particles will have high fitness. The result is that every particle has so high fitness value that entire population has very high fitness value more possibly. The LBF strategy means the larger QoS score of a candidate service is, the higher the probability to be selected is for a task. Higher probability to be selected of candidate services with high QoS scores may lead to quick convergence for iPSOA in local optimization level.

The evaluation model of QoS presented in this paper can also be used to calculate the QoS score of every candidate web service, and the probability of a candidate service to be selected is the result of its QoS scores divided by the sum of QoS scores of all candidates of the same task.



The fitness value of candidate web service can be calculated by:

$$f_i' = w_1 A_i + w_2 Rep_i + w_3 P_i + w_4 T_i \quad (23)$$

Here the value of  $w_1$ 、 $w_2$ 、 $w_3$ 、 $w_4$  are as same as those in formula(22), and  $A_i$ 、 $Rep_i$ 、 $P_i$ 、 $T_i$  denote the QoS value of the  $i$ th candidate web service. Suppose a task with  $m$  candidate web services, and then each candidate service is selected with the probability  $\frac{f_i'}{\sum_{j=1}^m f_j'}$ .

#### 4.4. Selection Approach based on iPSOA

The iPSOA can perform optimal search in the case of the QoS-aware web service selection problem considered in this paper. Supposing the positions of particles in iPSOA as the combinations of a composite service and the fitness function to be QoS score of each combination, the right combination with best QoS score can be found through iterative update procedure of iPSOA. The flowchart of the selection approach using iPSOA is illustrated as Fig.5.

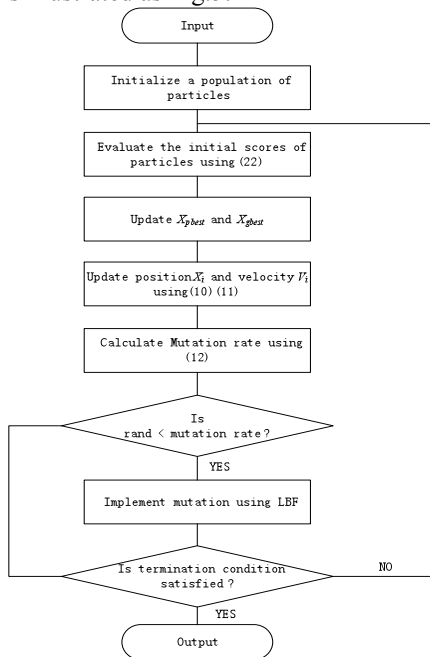


Fig. 5. Flowchart of iPSOA for QoS-aware Web Service Selection problem

As shown in Fig.5, the **Input** includes the numbered candidate web services and their QoS parameters, workflow language described Composite Services, the constraints of users (like response time <5s and price<2£). Then, the iPSOA procedure is called, and

needed to note is that the positions of particles must be checked when new positions are generated through constraints of users. As soon as the termination condition is satisfied (like generation number<2000), the combination of the best QoS performance (highest QoS score) will be outputted.

#### 5. Simulation Results and Analysis

Two parts of simulations are taken to validate the effectiveness of iPSOA with the AWA, NUM and LBF strategies for solving the QoS-aware web service selection problem. Numerous simulation comparisons have been performed to illustrate the effectiveness of each strategy respectively in the first part, and the second part gives the comparisons among four algorithms, namely QCDSS<sup>24</sup>, CoDiGA<sup>19</sup>, iPSOA, PSOA, to show the integrated performance of iPSOA. All simulations were conducted on several PCs with same configuration of Intel Core Duo (T7100) 2.1GHz processor, 2.50GB of RAM, Windows XP Pro. All hosts connected to a LAN through 100Mb/s Ethernet cards.

Here, the comparisons are taken in the same composite service request and the following data were adopted, including workflows of different sizes (the number of tasks were 10, 15, 20, 25 and 30 in our experiments), the number of candidate web services for each task (they were randomly obtained from 10 to 20) and four QoS attributes for each web service (two negative QoS attributes and two positive QoS attributes). A simplified representation of candidate web service was used, including an ID number and some QoS data that were retrieved at random in the range of defined values.

##### 5.1. Simulations on optimal strategies for PSOA

In order to improve the convergence and overcome the prematurity of PSOA, three strategies, namely NUM, AWA and LBF, were proposed. This section will validate their efficiency by comparisons with PSOA without these strategies. The comparisons of algorithms are set up with the same condition, including population of particles, the parameters of NUM, and the fitness function, and the population size was 30. The simulations in this part also have the same number of tasks (30). Furthermore, the  $P_1$ ,  $P_2$ ,  $P_3$ , equal to 0.1, 0.2, and 0.7 respectively which lead the best conditions we found of PSOA. The entire statistical data are the average values of algorithms running for independent 50 times.

### 5.1.1 Experiments on NUM

The abbreviation “PSOA-NUM” represents the PSOA with the NUM strategy and the abbreviation “PSOA-M (0.5)” represents PSOA with the mutation strategy of fixed mutation rate 0.5. Fig.6 plots the fitness function evolution of these three algorithms across the PSOA generations.

In Fig.6, these three curves all converges after the 1700th generation and PSOA-M (0.5) reaches the best fitness values, PSOA the worst, and SPOA-NUM is in the middle.

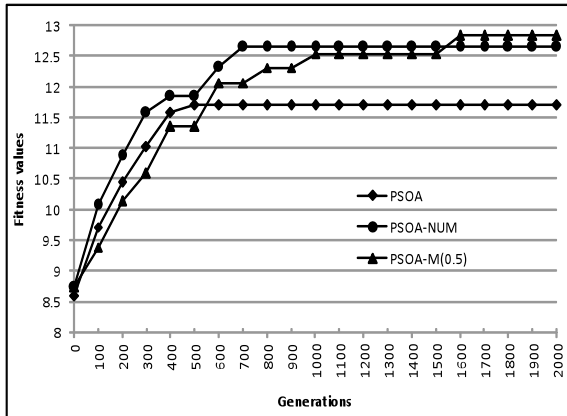


Fig.6 Fitness comparisons for NUM

As shown in Fig.6, PSOA converges quickly but reaches the worst fitness value. That is because particles congregated fleetly to the global best one whose nondirectional movement leads to prematurity. In PSOA-M (0.5), a fixed mutation rate is introduced to the global best particle. We find that PSOA-M (0.5) could find better values in the whole process. This means the mutation strategy do good in searching for better solutions. However, fixed high mutation probability of the global best particle leads to longer running time of the algorithm and slow convergence. By comparison, PSOA-NUM converges more quickly and reaches a relatively worse value. That is because the NUM strategy can enlarge the particles searching space in the earlier time and make them congregated fleetly step by step.

### 5.1.2 Experiments on AWA

The abbreviation “PSOA-AWA” represents the PSOA with the AWA strategy and the PSOA adopts a certain weight approach where the  $P_1$ ,  $P_2$ ,  $P_3$ , equal to 0.1, 0.2, and 0.7 respectively which lead the best conditions we found of PSOA in experiments.

Fig. 7 plots fitness function evolutions across generations. The upper curve stands for the PSOA-

AWA and the lower one stands for PSOA. These means PSOA-AWA could have larger fitness values than PSOA.

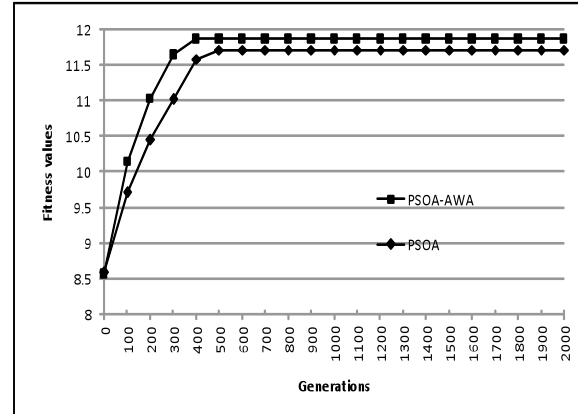


Fig.7 Fitness comparisons for AWA

As shown in Fig.7, PSOA-AWA converges quickly and reaches the stable fitness value in less time than PSOA, and even obtains a better converged value. That is because AWA strategy ensures the particles adjust their velocity adaptively so that they can follow the global best particle with high velocity and converge quickly. As we know that the velocity of PSOA-AWA is determined by the particles' current velocities, their local best positions and the global best position. Since  $X_{g_{best}}$  is the best position of current particles,  $X_{p_{best}}^i$  is the best position of each particle, and  $X_i$  is the current position, each particle will adjust its velocity adaptively according to the proportion of each part. PSOA applies fixed weight approach, with which it may be trapped by the local optimal solution if the  $V_i$  is too fast, or it may affect the convergence speed for the slow velocity. From Fig. 7, we can find that although the fitness value of PSOA-AWA is better than PSOA, but the superiority is limited. AWA strategy does well in both the convergence and fitness value.

### 5.1.3 Experiments on LBF

LBF strategy is based on the notion that combinations would have higher QoS value if every task had high QoS value. However, this is not reasonable completely because the QoS aggregate rules are not all linear, but we find that the services with higher scores are more probable to bring a higher aggregate QoS score.

Fig. 8 plots fitness function evolutions across generations for PSOA-LBF and PSOA. The upper curve stands for the PSOA- LBF and the lower one stands for PSOA. These means PSOA-LBF could have larger fitness values than PSOA.

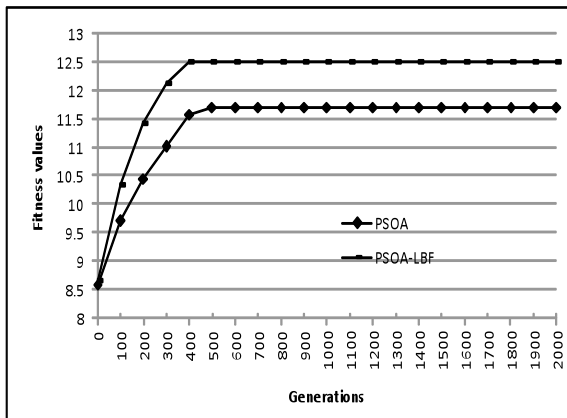


Fig.8 Fitness comparisons for LBF

As shown in Fig.8, PSOA-LBF converges to a much better value than PSOA in a little less time. On one hand, the LBF strategy can select the candidate services with high QoS performance score in high probability avoiding absolutely randomness in local level for the purpose on converging quickly and reaching better values; On the other hand, probability method can keep the diversity of the particles so that some better solutions would not be missed.

## 5.2. Experiments on iPSOA

The effectiveness of each strategy had been validated in section A, and all of them improved the convergence and fitness values to overcome prematurity to some extent. Now the integration of them, which is an improved PSOA, will be examined through comparing with QCDSS<sup>23</sup>, CoDiGA<sup>19</sup> and PSOA. And all these experimental data are the average values of algorithms running for 30 times. The same parameters are adopted for these four algorithms: the population size of CoDiGA is 200, crossover probability is 0.7, and mutation probability is 0.1, while QCDSS, PSOA, and iPSOA's population size is 30.

Firstly, let us study the convergence of them because the execution time of algorithms is concerned easily. Needed to state is that because these four algorithms can be divided into two classes: Genetic Algorithm and Particle Swarm Optimization Algorithm, while CoDiGA belongs to the former and other three to the latter. Two classes have different operation in each generation, so Fig. 9 describes the fitness function evolution across execution time. The composite service

is composited of 30 tasks, and the execution time is the "first hit" time of the best fitness values.

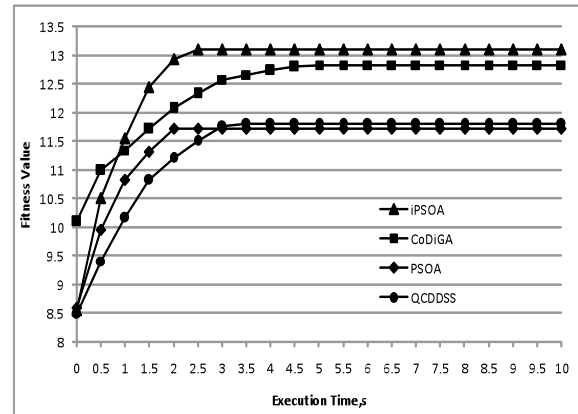


Fig.9 Convergence of fitness for 30 tasks

Although CoDiGA takes a best initial fitness due to its enhanced initial population policy, it converges slowest as Fig.9 shows because of the random evolution operations. As comparisons, QCDSS, PSOA and iPSOA converge faster, which are more efficient for QoS-aware Web Service Selection problem. However, these three algorithms are different: QCDSS converge slowest because it adopted a binary coding approach for the standard PSOA while a decimal one is applied in iPSOA. PSOA and iPSOA converge faster, but PSOA falls into local trap and converge to a smaller fitness value. Namely, iPSOA can select web services with best QoS performance. We can also find that the integration excels the three single policies in fitness value by comparison between iPSOA and PSOA.

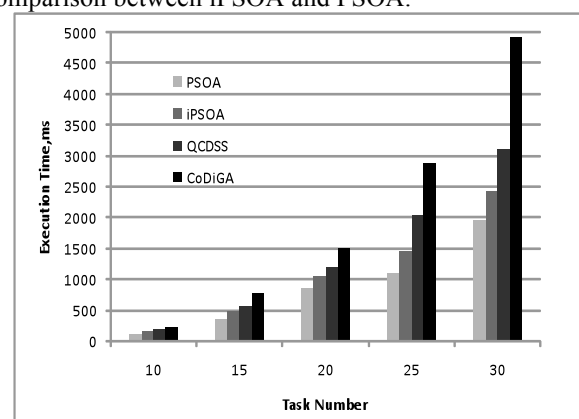


Fig. 10. Comparison of execution times

Fig.10 describes statistical experiment results of average time on the conditions that the maximal fitness values in Table 2 are reached. Table 2 describes the

maximal fitness values in the time shown in Fig.10. These data are also collected from 30 times independent runs of each algorithm. The generations of algorithms are 2000.

Table 2. Comparison of fitness values for different task number.

| Task number | Best Fitness Values |         |         |         |
|-------------|---------------------|---------|---------|---------|
|             | iPSOA               | CoDiGA  | PSOA    | QCDSS   |
| 10          | 4.3629              | 4.1059  | 3.5962  | 3.6741  |
| 15          | 6.5387              | 6.3963  | 5.8622  | 5.9417  |
| 20          | 8.8672              | 8.7219  | 7.949   | 8.1105  |
| 25          | 10.9538             | 10.7367 | 9.8715  | 10.1603 |
| 30          | 13.1014             | 12.826  | 11.7123 | 11.801  |

As described above, the characters of iPSOA obtain higher fitness value, faster convergence, which is the most efficient and best algorithm for QoS-aware Web Service Selection. The integration of AWA, NUM and LBF can bring better fitness and faster convergence. Especially, the execution time of iPSOA for 30 tasks is only a half of CoDiGA, but an even better value. Although the execution time of PSO is shortest, the fitness values are not satisfied. From above analysis, iPSOA has best performance among these four algorithms. At the same time, it outperforms CoDiGA in execution time, while slightly longer time when comparing with PSO. QCDSS has a slower convergence and small fitness value compared with others.

## 6. Conclusions

Web service selection with global QoS constraints is an active research area. It is important to select which services to be used in a composite web service according to requester's QoS requirements. In this paper, we consider a novel heuristic approach to solve the QoS-aware Web Service Selection problem by using an improved PSO which is widely used to solve large-scale optimization problems, and mainly make three contributions which are summarized as follows:

(1) After presenting a QoS-aware service selection model, which can be transformed into a general constrained optimization problem, we redefine the parameters and operations of PSO to make it suitable to solve QoS-aware Web Service Selection problem.

(2) We present NUM strategy for the global best particle to overcome the prematurity of POSA, and a

NUM function is also designed. The incorporation of the non-uniform mutation operator in iPSOA significantly improves the diversity of PSO's population and overcome the prematurity situation effectively, so that iPSOA can find combinations with better QoS performance.

(3) In order to improve the convergence, AWA and LBF strategies are adopted, which ensure faster convergence towards satisfactory solutions. Results illustrate the proposed strategies greatly improve the convergence, which means iPSOA reduces execution time while achieving a stable QoS performance results.

Experimental results show that iPSOA, which is an integration of proposed optimal strategies, offers a feasible and practically optimal approach that can be implemented in QoS-aware Web Service Selection problem. It's believed that our work will inspire the exploration on solving this problem by using PSO algorithm.

However, the proposed iPSOA does not take adaptive termination of algorithms into account, which is acknowledged as a difficult problem in this field and that is our future work.

## Acknowledgements

This work was supported by project National Key Basic Research Program of China (973 Program) under the Grant No.2009CB320406 and No.2009CB320504, Foundation for Innovative Research Groups of the National Natural Science Foundation of China under the Grant No. 60821001, Natural Science Foundation of China under Grant No.61070205 and No.61070206, and the Fundamental Research Funds for the Central Universities (BUPT2009RC0701) of China.

## References

1. M.P.Papazoglou, D. Georgakopoulos, Service oriented computing: Introduction, *Communication of the ACM*, 46(10) (2003) 24-28.
2. W. Chou, L. Li, F. Liu, Web Services for Communication over IP, *IEEE Communications Magazine*, 46(3) (2008) 136-143.
3. A. F. Huang, C. W. Lan, J.H. Yang, An optimal QoS-based Web service selection scheme, *Information Sciences*, 179 (19) (2009) 3309-3322.
4. P. Wang, QoS-aware web services selection with intuitionistic fuzzy set under consumer's vague perception, *Expert Systems with Applications*, 36(3)(2009) 4460-4466.

5. M. Wagner, W. Kellerer, Web services selection for distributed composition of multimedia content, in *Proceedings of ACM International Conference on Multimedia*(New York, USA, 2004), pp. 104–107.
6. OASIS, Web Services Business Process Execution Language (WSBPEL) Version 2.0, Available at: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, 2007
7. OASIS, UDDI Version 3.0.1, UDDI Spec Technical Committee Specification, Available at: [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm), 2004
8. ISO8402: Quality management and quality assurance vocabulary, 1994.
9. ITU-T Recommendation E.800, Terms and Definitions Related to Quality of Service and Network Performance Including Dependability, 1994.
10. L.Z. Zeng, B. Benatallah, A.H. Ngu, D. Marlon, K. Jayant, C. Henry, QoS-aware middleware for web services composition, *IEEE Transactions on Software Engineering*, 30 (5)(2004) 311-327.
11. C. Zhou, L.T. Chia, B.S. Lee, DAML-QoS ontology for web services, in *Proceedings of IEEE International Conf. on Web Services*(San Diego , USA,2004),pp. 472-479.
12. Y. Liu, A.H. Ngu, L.Z. Zeng, QoS computation and policing in dynamic web service selection, in *Proceedings of 13rd International Conf. on World Wide Web* (New York, USA,2004), pp. 66 -73.
13. D. Ardagna, B. Pernici, Global and local QoS constraints guarantee in web service selection, in *Proceedings of IEEE Int. Conf. on Web Services* (Orlando, USA ,2005),pp. 805-806.
14. D. Ardagna, B. Pernici, Adaptive service composition in flexible processes, *IEEE Transactions on Software Engineering*, 33 (6) (2007) 369-384.
15. A. Mohammad, R. Thomass, Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition, in *Proceedings of IEEE Int. Conf. on Web Services* (Los Angeles, USA, 2009),pp. 881-890.
16. L.J. Zhang, B. Li, T. Chao, H. Chang, On demand web services-based business process composition, in *Proceedings of IEEE Int. Conf. on System, Man, and Cybernetics*(Washington DC, USA, 2003) pp. 4057-4064.
17. G. Canfora, M.D. Penta, R. Esposito, M.L. Villani, A lightweight approach for QOS-aware service composition, in *Proceedings of 2nd Int. Conf. on Service Oriented Computing* (New York, USA, 2004), pp. 36-47.
18. C.W. Zhang, S. Su, J.L. Chen, DiGA: Population diversity handling genetic algorithm for QoS-aware web services selection, *Computer Communications*, 30(5) (2007) 1082-1090.
19. Y. Ma, C.W. Zhang, Quick convergence of genetic algorithm for QoS-driven web service selection, *Computer Networks*, 52 (5)(2008) 1093-1104.
20. J. Kennedy, R.C. Eberhart, Particle swarm optimization, in *Proceedings of IEEE Int. conf. on Neural Networks*, 1995, 1942-1948.
21. T.O. Ting, K.P. Wong, C.Y. Chung, Hybrid constrained genetic algorithm/particle swarm optimization load flow algorithm, *IET Generation, Transmission & Distribution*, 2 (6)(2008) 800-812.
22. L. Hu, X.L. Che, X.C. Cheng, Bandwidth Prediction based on Nu-Support Vector Regression and Parallel Hybrid Particle Swarm Optimization, *International Journal of Computational Intelligence Systems*, 3 (1)(2010) 70-83.
23. X.C. Zhao, X.S. Gao, Z.C. Hu, Evolutionary programming based on non-uniform mutation, *Applied Mathematics and Computation*, 192 (1)(2007) 1-11.
24. C.H. Hu, X.H. Chen, X.M. Liang, Dynamic services selection algorithm in Web services composition supporting cross-enterprises collaboration, *Journal of Central South University of Technology*, 16(2) (2009), 269-274.
25. X.C. Zhao, A perturbed particle swarm algorithm for numerical optimization, *Applied Soft Computing*, 10 (1) (2010) 119–124.