Web Services Composition based on Domain Ontology and Discrete Particle Swarm Optimization

Zhenwu Wang [1] and Ming Chen [2]
Department of Computer Science and Technology, China
University of Petroleum, Beijing, China

Abstract.This paper proposes an approach for web services composition based on domain ontology and discrete particle swarm optimization (DPSO) algorithm. This method builds an optimized graph for service composition based on domain ontology and its reasoning capability, and then a discrete particle swarm optimization algorithm based on the graph is proposed to accomplish service composition. The simulation results show that it can produce good results, especially when the amount of web services is large.

The rapid development of web services shows that Internet software will be integrated following the service-oriented model (big granularity, loose coupling and dynamic banding) in the future. Single web service only provides the limited function, so web services composition is very important. Now most web services are composed by hand, so it is necessary to study web services composition based on semantic.

Literature [1] adopted the DOSCM method to study web services dynamic composition based on domain ontology, but it needs sort all the web services following the connection degrees among web services before composing them. Obviously it is very difficult to do so in the case of too many web services. In this paper, we adopt the discrete particle swarm optimization algorithm and domain ontology to compose web services dynamically based on literature [1],we can see that it has obvious effect when the service amount is large.

The remainder of this paper is organized as follows. Section 1 gives the description of web services composition based on web domain ontology; Section 2 is the overview of the discrete particle swarm optimization (DPSO) algorithm; Section 3 details the proposed web services composition method which bases on DPSO and domain ontology (we called this method as DOPSO); Section 4 is the experimental result, and finally section 5 concludes this paper.

1 The problem of web services composition based on domain ontology

The basic conceptions of domain ontology can refer to literature [1], on account of the paper's length,this paper will not discuss them. The problem of web services composition based on domain ontology can be described as follows.
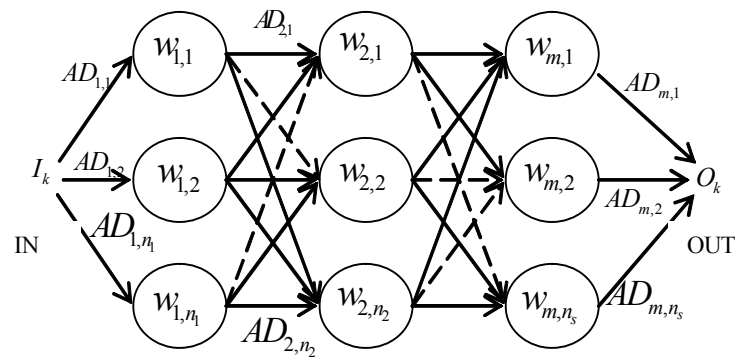

Fig. 1. WSC-Graph

The basic idea of DOPSO is that we can compute the connection degrees among web services based on conception similarity degrees and domain ontology to form an optimized web services composition graph (WSC-Graph), and then we can switch the web services composition problem to the problem that we find a path on the WSC-Graph which satisfies the users' requirement best. As described in figure 1,supposing composed service W has m basic web services, that is $W = (w_1, w_2, ..., w_m)$ （1）

For the web service requirement $WSR_k(I_k, O_k)$, there are n_1 web services which have the semantic connection relationship(that is to say, w_1 has n_1 candidate web services), and these n_1 web service also have n_2 subsequent web services, the rest may be deduced by analogy, and W_m has n_s candidate web services, that is

$I_k \equiv I_{1,1} \equiv I_{1,2} \equiv ... \equiv I_{1,n_1}$, $O_{1,1} \equiv I_{2,1} \equiv I_{2,n_2}$, ..., $O_{m,1} = O_{m,2} = ... O_{m,n_s} = O_k$. I_k in figure 1 presents the input of $WSR_k(I_k, O_k)$ and O_k is the output. The circles in figure 1 denote web services, the two web services which are connected by the directed edge have the signification of semantic connection, the broken lines in figure 1 show that the two web services connected by them have no signification of semantic connection, but for the simple aim, we still connect them and their semantic connection degrees are zero. Each path from I_k to O_k in WSC-Graph is a service composition result and the best one is the one which has the biggest web services composed satisfaction degree. The details of conceptions, such as the similarity degree between two domain ontology concepts, the semantic similarity degree between two conception sets and the formula about connection degree of two services, can refer to literature [1].

2 The discrete particle swarm optimization

Particle swarm optimization (PSO) is one of the evolutionary computational techniques. Suppose that the search space is D-dimensional, and the position of the ith particle of the swarm can be represented by a D-dimensional vector, $\bar{x_i} = \{x_{i,1}, x_{i,2}, ..., x_{i,D}\}$. The velocity (position change per generation) of the particle x_i can be represented by another D-dimensional vector, $\bar{v_i} = \{v_{i,1}, v_{i,2}, ..., v_{i,D}\}$. The best position previously visited by the ith particle is denoted as $PC_{i,best} = \{p_{i,1}, p_{i,2}, ..., p_{i,D}\}$ and its corresponding fitness value is denoted as $PL_{i,best}$. If the topology is defined such that all particles are assumed to be neighbors and g as

the index of the particle visited the best position in the swarm, then $GL_{best}=\{p_{g,1},p_{g,2},...,p_{g,D}\}$ becomes the best solution found so far and the responding fitness value is denoted as GC$_{best}$. The velocity of the particle and its new position will be determined according to the following two equations:

$$v_{i+1,d}=v_{i,d}+c_1r_1(p_{i,d}-x_{i,d})+c_2r_2(p_{g,d}-x_{i,d}) \quad (2)$$
$$x_{i+1,d}=x_{i,d}+v_{i+1,d} \quad (3)$$

Where c_1 and c_2 are acceleration coefficients regulating the relative velocity toward global and local best, r_1 and r_2 are two random numbers in [0,1]. Popularly, set a maximal iterative number (such as n$_{max}$) as the concluding condition.

Usually, PSO has been applied to solve the continuous space optimization problems, and now some scholars used it to solve the discrete space optimization problems. In literature[5], Clerc proposed a discrete particle swarm optimization algorithm(DPSO) to solve the TSP problem, and some scholars used DPSO to handle QoS multicast routing problems in literature[6][7]. In literature [8], a hybrid particle swarm optimization algorithm (HPSO) has been described to solve the TSP problem, which combined genetic algorithm. Based on literature [5], literature [4] introduced repulsion operator and learning operator into DPSO to restrain the stagnant phenomena, and acquired good effects. In this paper, we propose an improved discrete particle swarm optimization algorithm to solve the problem of web services composition, which based on literature [4].In the next section; we will detail the improved algorithm.

2.1   Position and velocity of particles

As described in figure 1, the position X can be described by the basic services which are invoked by the composite service.

$$X=(x_1,x_2,...x_i,...,x_n),1\leq i\leq n,1\leq x_1\leq n_1,...,1\leq x_n\leq n_s \quad (4)$$

Where x_i is the *ith* basic service, n is the amount of basic services which are invoked by the composite service. The dimension in X shows the composite sequence number of basic services. According to figure 1, x_i has n_i candidate web services.

The function of velocity is to change the positions of particles. Similar to the definition of position X, the velocity is defined as follows:

$$V=(v_1,v_2,...v_i,...,v_n),1\leq i\leq n,1\leq v_i\leq n_1,...,1\leq v_n\leq n_s \quad (5)$$

Where the dimension in V shows the composite sequence number of basic services, and n is the amount of basic services which are invoked by the composite service. Each item in V has two meanings: if v_i equals 0, it means no operation, else means to modify the data in this item to v_i.

2.2   Operations about position and velocity

1．Addition operation

The addition operation between the position and the velocity accomplishes the movement of the particle's position, and moves a particle to a new position. The formula is defined as follows:

X = X + V (6)

Each data in the new position is defined by the formula (7):

$$\begin{cases} x_i=v_i & if \ v_i\neq 0 \ or \ v_i\neq x_i \\ \Phi & else \end{cases} \quad (7)$$

When creating a new position X,   if v_i does not equal 0 or x_i it means x_i equals v_i, else means no operation.

2. Subtraction operation

Two positions executing subtraction operation can produce a new velocity:

$$V=X_2-X_1 \quad (8)$$

Form the formula (8), we can see that the particle can move from position X_1 to X_2, if it has a velocity V. According to the formula (9), compared the data in each dimension between X_1 and X_2, if $x_{1,i}$ equals $x_{2,i}$, then v_i equals 0,else v_i equals $x_{2,i}$.

$$v_i = \begin{cases} 0 & if \quad x_{1,i} = x_{2,i} \\ x_{2,i} & else \end{cases} \quad (9)$$

3．Multiplication operation on velocity

Multiplication operation has the probability meanings, the formula is defined as follows:

$$V_2 = c \cdot V_1, c \in [0,1] \quad （10）$$

Where c is a constant, and it has the probability meanings. When computing V_2, we should produce a random number $randv$ ($randv \in [0,1]$) for each data $v_{1,i}$ in V_1. If $randv$ is less than c, then $v_{2,i}$ equals $v_{1,i}$,else $v_{2,i}$ equals 0.The formula is defined as follows:

$$v_{2,i} = \begin{cases} v_{1,i} & if \quad randv < c \\ 0 & else \end{cases} \quad (11)$$

4．The addition operation between velocities

Executing addition operation between two velocities can produce a new velocity:

$$V = V_1 + V_2 \quad （12）$$

Each data in V can be defined as follows:

$$v_i = \begin{cases} v_{2,i} & if \quad v_{2,i} \neq 0 \\ v_{1,i} & else \end{cases} \quad (13)$$

Generally, $V_1 + V_2 \neq V_2 + V_1$, and only if $V_1 = V_2$ we can conclude:

$$V_1 + V_2 = V_2 + V_1 \quad (14)$$

2.3   The movement equation of particles

For the particularities of the discrete space optimization problems, we modified the movement equation, and canceled the first item in the formula (2) because it is insignificance for discrete space optimization problems. The equation is defined as follows:

$$V = c_1(X_{pbest} - X) + c_2(X_{gbest} - X),$$
$$X = X + V. \quad （15）$$

2.4   variation operator introduce

DPSO also has some disadvantages, for example, easy to stagnant. In this paper, we introduce variation operator to add the diversity of particles

Def 1：the position comparability $s_{i,j}$: $s_{i,j} = \dfrac{1}{m}\sum_{k=1}^{m} iif(s_{i,k} = s_{j,k}, 1, 0)$ （20）

Where m is the position dimension，and if $s_{i,j} = 1$,it implies that the two particles have the same position，else implies that the two particles have the different position absolutely, and $s_{i,j} \in [0,1]$.

Def 2：the particle diversity d_i :the similar degree of the current position i, the native best position *pbest* and the whole best position *gbest* for a certain particle：

$$d_i = 1 - \frac{1}{3}(s_{i,pbest} + s_{i,gbest} + s_{pbest,gbest}) , \quad d_i \in [0,1] \quad （21）$$

Def 3:the particle swarm diversity $\overline{d_i}$: $\overline{d_i} = \frac{1}{n}\sum_{i=1}^{n} d_i$ （22）

Where n is the swarm size. If $\overline{d_i}$ is less than a certain constant $c_{\overline{d}}$ （for example 0.2），then start the variation operator to produce the diversity.

3 The DOPSO algorithm

The detailed algorithm can be described as follows:

1. Initialization：suppose the amount of particles is n, and the threshold of service satisfaction degree is ξ ,the probability constant in variation operation is $c_{\bar{d}}$, suppose the coefficients in the formula （15）are c_1, c_2 ;To produce the initial positions of particle swarm followed by the formula(4),that is $X_i, 1 \leq i \leq n$; To produce the initial velocity of each particle followed by the formula(6),that is $V_i, 1 \leq i \leq n$;To compute the fitness value for each particle, and according to the current position we can compute the fitness value $f_i, 1 \leq i \leq n$ followed by composition satisfaction degree which detailed in literature[1]; PC$_{i,best}$=X$_i$ and PL$_{i,best}$=f$_i$;GL$_{best}$=PL$_{i,best.}$ if and only if f$_i$=max{f$_k$},$1 \leqslant k \leqslant$ n;GC$_{best}$=PC$_{i,best}$,if and only if GL$_{best}$=PL$_{i,best}$;

2. To compare the current fitness value f_i with the fitness value $PL_{i,best}, 1 \leq i \leq n$,which is its own best position $PC_{i,best}, 1 \leq i \leq n$, if f_i is better than $PL_{i,best}$, $PL_{i,best}$ equals f_i ;

3. To compare the current fitness value f_i with the fitness value GC_{best} ,which is the best position in the swarm GL_{best} . For each particle, if f_i is better than GC_{best} , then GC_{best} equals f_i ;

4. To compute the diversity of particles $\overline{d}_i$,and if $\overline{d}_i$ is less than $c_{\bar{d}}$,then it should start variation operator to produce a new velocity for all the particles;

5. To evolve the positions and velocities followed by the equation(15);

6. If the result does not reach to the threshold values, then return to the step (2).

4 Simulation experimentation

We compared the proposed method with the DOSCM method in literature [1], and for the justice aim, we adopt the same software and hardware environment. CPU is Intel Pentium IV 2.4GHz,memory is 1 GB ,operation system is XP professional ,the development language is java and the IDE is Eclipse3.1.The amount of particle swarm is 200, $c_1 = 0.2$, $c_2 = 0.3$, $c_{\bar{d}} = 0.2$,and the algorithm will run 50 times.

In this paper, we generate random data as connection degree data and used them in the experiment. We adopt five test data musters, the web services amount is 300,900,1500,2100,2700 respectively. The smallest services satisfaction degree threshold is 0.6 in the both methods

CASE 1 composition success probability

Table 1. DOPSO: DOSCM

Satisfaction degree / Services amount	DOPSO	DOSCM
300	0.80	0.81
900	0.82	0.83
1500	0.84	0.84
2100	0.85	0.84
2700	0.89	0.88

From the data in table 1, the DOPSO has unconspicuous effect when web services amount is little and with the increase of web services amount, we can see that the composition success probability of DOPSO is similar to that of DOSCM.
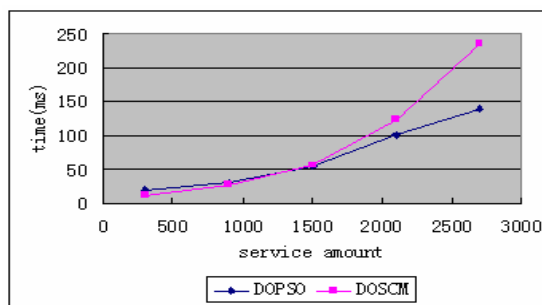
CASE 2 services composition efficiency

Fig .2. services composition efficiency

    Figure 2 describes the relation between the services total amount and the composed time. We can see that DOPSO has an unconspicuous effect in the case of less web services, and with the increase of web services amount, the composed time which DOPSO used is less than that of DOSCM, it implies that DOPSO has an obvious effect when web services amount is large.

5   Conclusion

From the results of experiment we can see that the discrete particle swarm optimization algorithm has good application property to solve the problem of web services composition based on domain ontology, and it provides a new way to study web service composition. But at the same time, the particle swarm optimization algorithm also has some defaults, for example easy to stagnate. So the next work is to design better particle swarm optimization algorithm.

References

    1. M. Li, D.Z. Wang, X.Y. Du, and S. Wang, "Dynamic Composition of Web Services based on Domain Ontology", *Chinese Journal of Computers,* 28(4),645-650(2005).

    2. J. Wu, Z. h. Wu, Y. Li and S.G. Deng, "Web Service Discovery and Similarity of Words", *Chinese Journal of Computers,* 28(4), 596-602(2005).

    3. J.F. Zhao, B. Xie, L. Zhang, F.q. Yang, "A Web Services Supporting DomainFeature", *Chinese Journal of Computers*, 28(4),732-738(2005).

    4. Y.W. Zhong, J.g. Yang and Z.Y. Ning, "Discrete particle swarm optimization algorithm for TSP problem", *System Engineering –Theory & Practice*,6,89-93(2006).

    5.M. Clerc, "Discrete particle swarm optimization",Onwubolu GC ,Babu BV, New Optimization Techniques in Engineering ,SpringerVerlag (2004).

    6. D.R. Pan, "QoS Multicast Routing Optimization Algorithm Based on SCE Algorithm and Particle Swarm Optimization Algorithms", *Computer and Information Technology,* 14-18,(2006).

    7. J. Qing, W.b. Xu and J. Sun, "QoS Multicast Routing Optimization Algorithm Based on Particle Swarm Optimization Algorithms", *Computer Engineering and Applications,* 27(1),106-108(2006).

    8. S. Gao, B. Han, X.J. Wu, and J.Y. Yang, "Solving traveling salesman problem by hybrid particle swarm optimization algorithm", *Control and Decision,,*19(11), 1287-1289(2004).