Research on the Particle-Ant Colony Algorithm in Web Services Composition Problem

Shujun Pei, Xi Shi and Daming Hu
College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, 150080, China

Abstract: In order to improve the efficiency of finding the optimal solution for web services composition problem to meet increasingly sophisticated demand of users, an algorithm called Particle-Ant Colony Algorithm (PACA) based on Quality of Service (QoS) is proposed in this article. This algorithm converts the web services composition problem into shortest path problem of QoS-based directed acyclic graph. First, find several suboptimal paths by Particle Swarm Optimization Algorithm (PSOA) and initialize the pheromones of these paths, then find optimal solution by Ant Colony Optimization Algorithm (ACOA). The experiment result indicates that the PACA can effectively improve the capability of optimizing web services composition problem. The results of the research has theoretical and practical reference values to relevant issues.

Key words: Web service composition, particle-ant colony algorithm, quality of service, combinatorial optimization

INTRODUCTION

As the internet becoming more and more popular and the IT technology updating continuously, more and more heterogeneous service applications exist in network. Service-Oriented Computing (SOC) solves the problem that heterogeneous software cannot interoperate, so SOC become an important model that can utilize complex distributed applications in open and heterogeneous environment. Web service is the core technology of the new architecture. In order to improve the reusability of web services, single web service should be designed as simple as possible, but it is difficult to meet the needs of complex business. Therefore, it's need to combine multiple atomic web services to meet the ever-changing demand. Most web services have the same function but they have different QoS parameter values. How to select a number of different services with specific QoS values from massive web services to meet users requirements and integrate these selected services into an effective service chain is the web services composition problem. Web services composition is a NP-hard problem (Fang et al., 2009), the traditional algorithms like Greedy Algorithm and Exhaust Algorithm are simple and practicable when service domain scale is small. But in reality, the service domain is often and massive, in which case the traditional algorithm can not find the optimal solution. As more and more scholars turn their attention to web services composition problem, they all believe that the most appropriate method to solve this problem is heuristic algorithms, such as Genetic Algorithm (Ge et al., 2008), PSOA, ACOA. But each algorithm has its own shortcomings such as poor convergence performance, bad ability to find global optimal solution. In this article author proposed Particle-Ant Colony Algorithm (PACA) based on previous results which integrates the advantages of PSOA and ACOA and apply it to the web services composition problem.

WEB SERVICE COMPOSITION MODEL AND MODEL CONVERSION

WEB service composition model: This article considers only the sequence services composition model and other model such conditions, parallel, loop service composition model may be recurred into sequence service composition model by the method given by the literature (Fang et al., 2009).

These web services (the quantity is M) that have same connection interface and function, provided by different service providers, called web service Candidate Set (WSCT). The web services in the same WSCT have different QoS parameters value and they can be atomic web services.

Web services composition model is shown in Fig. 1. S represents the starting point of the model, F represents the end point of the model. Web service composition workflow starts from the S, then it select one web service according to QoS parameter value from different WSCT one by one until the workflow reaches F. Nws is the quantity of the web services in WSCT. The quantity of
From the above description, we could convert web service model into a directed acyclic graph model shown in Fig. 2.

Each web service in all WSCT can be considered as a node in the directed graph, the path between two web services in different WSCT could be considered as edge in the directed graph (Pfeffer et al., 2008). The QoS metric of a selected web service could be considered as the weight of the path. Now the web services composition model has been simplified into directed graph (G) with weights constraint of QoS metric:

\[ G = (P, WS, E, QoS) \]

\[ P = \{S, WSCT_1, WSCT_2, WSCT_i, WSCT_j, F\} \]

P represents a collection of several WSCT, S is the start point, F is the end point, WSCT_i expressed one WSCT:

\[ WS_i = \{WSCT_{i1}, WSCT_{i2}, WSCT_{i3}, \ldots, WSCT_{in}\} \]

WS_i represents all web services that can be used in one WSCT:

\[ E = \{(WSCT_{i1}, WSCT_{i2})|WS_{ij} \in WSCT_j, WS_{ij} \in WSCT_i, i \neq j\} \]
$E$ represents the path between two web services in different WSCT:

$$QoS_I = \{Price_i, Time_i, Resh_i\}$$

$QoS_I$ represents QoS attributes of a specific web service (WSCT$_I$).

Now, the web services composition problem has been converted into the shortest path Problem of Weighted directed acyclic graph based on QoS (Gao et al., 2009).

**APPLICATION OF PACA IN WEB SERVICES COMPOSITION PROBLEM**

**Introduction of PSOA:** PSOA is a global optimization algorithm, proposed by Eberhart and Kennedy in 1995. The algorithm is based on the simulation of birds, fish and other groups’ foraging behavior. The advantage of this algorithm are (Yu and Zhang, 2009):

- This algorithm is simple and relatively easy to implement
- It initializes random Particle Swarm, has a strong capability of global search
- The use of the evaluation function to measure the degree of individual merits makes the search speed fast
- Its scalability is strong. The disadvantage is: It cannot take full advantage of the feedback from system to solve combinatorial optimization problems

**Introduction of ACOA:** ACOA is a combinatorial optimization algorithm proposed by the Italian scholar. It is based on the real ant foraging behavior and applied to TSP, job scheduling and many other classical combinatorial optimization problems. Some good results have been achieved. Its advantages are (Liu et al., 2008):

- It adopts a positive feedback mechanism, finally converge to the optimal solution by continuously updating pheromone
- It is a distributed optimization method and easy to achieve parallel implementation
- It is suitable for solving discrete optimization problems
- Good robustness

**Idea of PACA:** This algorithm is the fusion of the advantages of PSOA and ACOA, avoids their shortcomings. It utilizes PSOA’s rapid global convergence to generate relevant initial pheromone and ACOA’s parallelism, positive feedback mechanisms and high efficiency features to solve the combinatorial optimization problems. In this article, PACA will be applied to the web services composition problem to find the optimal solution.

**Improved PSOA for the initial pheromone:** Each path can be abstracted to $P_i = (P_{i_0}; P_{i_1}; P_{i_2}; P_{i_3}; ...; P_{i_{j_1}}; P_{i_{j_2}}; P_{i_{j_3}}; ...; P_{i_{j_z}}; P_{i_{j_z}})$ To utilize improved PSOA, vector $P_i$ can be considered as a particle, each coordinate represents a number of web services, the speed $V_i$ of $P_i$ is defined as the distance coordinate to move at each iteration, $V_i = (V_{i_0}; V_{i_1}; V_{i_2}; V_{i_3}; ...; V_{i_z})^T$. The number of the particle is $N$.

When algorithm is initialized, each particle will be assigned random coordinate (random path), every particle needs to track two extremum to update their coordinates at each iteration. The first is the particle’s the best solution, called individual extremum points (expressed by pbest), the second extremum point is the best solution of the all particles, called the global extremum point (expressed by gbest), pbest is the initial position of the particle at initialization, after each iteration, each particle calculate the fitness by fitness function, if fitness was better than pbest’s, then previous pbest would be replaced by new one, gbest is the best solution among all pbest. The fitness function is:

$$QoS(p_i) = \frac{W_R(p_i)}{W_R(p_i) + W_T(p_i)}$$

$$f(P_i) = \sum_{i=1}^{N} QoS(p_i)$$

$W_R$, $W_P$, $W_T$ is the weight of the QOS attribute, $\{W_P; W_T; W_r = 1\}$. $R(p)$, $P(p)$, $T(p)$ denote the atomic web service QoS values of price, response time and robustness pbest, $f(P_i)$ gbest = MAX$f(P_i)$.

When these two extremum points were found out, The particles is according to Eq. 3-6 to update their speed and coordinates:

$$V_{v^*} = V_{v^*} + Select\_1(pbest^* - X_{v^*}) + Select\_2(gbest - X_{v^*})$$

$$Select\_1 = \begin{cases} \frac{1 - \epsilon \times rand \_x}{0} & \text{if } cg \times rand \_x < 0 \\ \epsilon \times rand \_x & \text{else} \end{cases}$$  

$$Select\_2 = \begin{cases} \frac{1 - \epsilon \times rand \_y}{0} & \text{if } cg \times rand \_y < 0 \\ \epsilon \times rand \_y & \text{else} \end{cases}$$
\[
\text{rand}_p = \frac{\text{pbest}_{x_i}^k}{\text{pbest}_{x_i}^k + \text{gbest}^v} \\
\text{rand}_g = \frac{\text{gbest}^v}{\text{pbest}_{x_i}^k + \text{gbest}^v} \\
x_i^d = x_i^c + \text{rand}_d 
\]

(5)

(6)

cp is individual-extremum-point-replaced-factor, represents that the particle coordinates tend to be updated by individual-extremum-value, cg is global-extremum-point-replaced-factor, represents that the particle coordinates tend to be updated by global-extremum-value. \( cp + cg = 1 \). \( \text{rand}_p \) expressed the probability that the particle coordinate would be updated by individual-extremum-value, \( \text{rand}_g \) expressed the probability that the particle coordinate would be updated by global-extremum-value, \( \text{rand}_p + \text{rand}_g = 1 \).

\( x_i^d \) is the position of the jth dimension of the particle i at the Kth iteration, \( \text{pbest}_{x_i}^k \) is the position of the jth dimension of individual-extremum-point of the particle i at the Kth iteration, \( \text{gbest}^v \) is the position of the jth dimension of global-extremum-point at the Kth iteration.

In order to increase the convergence speed and the efficiency of the algorithm, the elimination mechanism has been used. The times of iterations is N, the threshold is Q, \( N_{\text{max}} \leq N \leq N_{\text{max}} \), the maximum iteration times is N_{max}, the minimum number of iterations times is N_{min}, when algorithm iterates half the N times, If one of the particles individual-extremum-point has not reached the threshold Q, this particles has been plunged into local optimal, its iteration should be terminated. After the times of iteration reached N, the paths, corresponding to all normal particles pbest, would be put into suboptimal solution collection. The improved ACO will start to run.

**Improved ACOA for optimal solution:** After the improved PSOA finished, we can get a group of suboptimal paths from start point S to end point F and pick out the top 20% paths which fitness are highest (expressed as O). Then, Set the Initial pheromone (expressed as \( \tau \)) that Improved ACO need based on paths collection O:

\[
\tau_i = C_{\tau} \times \text{QoS}(O_i) 
\]

(7)

\( \tau_i \) is the initial pheromone of the edge (i, j), \( C_{\tau} \) is a constant, \( \text{QoS}(O_i) \) is the fitness function of edge (i, j) and \( i \neq j \), the parameters the function need is from the web service that the edge (i, j) points to. If there are several paths inclucd the same edge (i, j), the \( \tau_i \) should be accumulated. To avoid premature convergence, we choose to use the max-min ant algorithm (MMAS) (Alaya et al., 2007), then \( \tau_{ij} = \tau_{ij}^\text{min} + \tau_{ij} \), \( \tau_{ij} \) is the minimum pheromone on every edge.

The State transition rules that Improved ACO utilizes is called Random proportion rule, it explains that how to calculate the probability that ant at node i select the next node j:

\[
p_{ij}(t) = \frac{\tau_{ij}^\alpha(t) \eta_{ij}(t)}{\sum_{0 \leq k \leq m} \tau_{ij}^\alpha(t) \eta_{ij}^k(t)} \\
\text{or} \\
0
\]

(8)

\( p_{ij}^\alpha(t) \) means that the probability that ant k at node i select the next node j at the t-th iteration; \( \eta_{ij} \) represents that nodes that ant k at node i can select; \( \alpha \) represents Information concentration factor which means the importance of the pheromone in selecting node; \( \beta \) represents inspired factor which means the significance of the inspired information when ant select node:

\[
\eta_{ij} = \sqrt{\frac{w_i R_i^2}{w_i R_i^2 + w_j R_j^2}} \cdot w_i + w_j - 1
\]

(9)

(10)

\( \eta_{ij} \) is the evaluating function of the selected web service node j, inspired function is closely related to QoS of the web service node that will be selected, so \( \eta_{ij} = C_{\eta} \cdot \eta_{ij}, C_{\eta} \) is a constant.

To avoid excessive residual pheromone submerged inspired information, pheromones must be updated by Eq. 11-12 after a ant Ants go through the path:

\[
\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \tau_{ij}(t)
\]

(11)

\[
\Delta \tau_{ij}(t) = \sum_{t=1}^{n} \alpha \tau_{ij}^k(t)
\]

(12)

\( \rho \) represents residual factor, It's used to avoid unlimited accumulation of pheromones, \( \rho \in [0, 1] \), represents Pheromone increment at edge (i, j) at t-th iteration, expressed that the Pheromone increment that the k-th ant left on edge (i, j) at t-th iteration. To avoid an increase of the pheromone is too high or too low at one iteration to affect the next path-selection, can be setted as:

\[
\tau_{ij} \leq \Delta \tau_{ij}(t) \leq \tau_{ij}^\text{max} \left\{ \begin{array}{ll}
\frac{\Delta \tau_{ij}^k(t)}{\Delta \tau_{ij}^k(t) + \tau_{ij}} & \text{ant K pass the edge(i,j)} \\
0 & \text{or}
\end{array} \right.
\]

(13)
\[ L_n = C_1 \cdot \left( \sqrt[3]{{\sum \frac{1}{n_{ij}}} } \right) \]  
(14)

\( L_n \) is the length of the path that ant \( k \) has passed, \( C_1 \) is a constant. If the quantity of the ant at a path is maximum, this path should be recognized as the optimal path.

**PARTICLE-ANT COLONY ALGORITHM WORKFLOW**

**Step 1:** The quantity of the particle group should be \( N_p \). Random path would be assigned to each particle. Set pbest, gbest based on Eq. 1-2, set the times of iterations M, the threshold \( Q \).

**Step 2:** Every particle begins to iterate according to Eq. 3-6. Halfway through the iteration, eliminate the particles which pbest is smaller than \( Q \).

**Step 3:** When iteration is complete, check out every particle’s pbest and select the top 20% highest fitness of pbest. These pbest should be transformed into the path, then the improved ACOA begins to operate.

**Step 4:** Set each path’s initial pheromone according to Eq. 7. Set the quantity of ants A, the times of iterations M, Information concentration factor \( \alpha \), inspiration factor \( \beta \) and residual factor \( \rho \).

**Step 5:** Iteration begins to execute based on the Eq. 8-13.

**Step 6:** When the iteration is finished, the path which has the maximum quantity of the ants is the optimal solution.

**EXPERIMENTS AND ANALYSIS**

To validate the effect and feasibility of PACA as described above. In this section, the classical ACOA and the PACA will be applied in web service composition problem and compare the effect. The all parameters that used in above formula are shown in Table 2. To verify the results objectively and effectively, this article assumes there were 20 WCST, 30 basic services in each WCST. \( P \), \( T \), \( R \) is the QoS attributes of basic web service. \( P \) represents the price (in the range 0-20), \( T \) represents the time (in the range 5-40), \( R \) represents robustness (range 0-1), these 3 attributes value are generated randomly. Limited by space, author will lists only a group of QoS attributes of one WCST (Table 3). Other WCST’s QoS parameters are also randomly generated in the same way. From the experiment data (Fig. 3) can be seen that the PACA can find the better solution within the same time.

**Table 2:** Parameters that used in above formula

<table>
<thead>
<tr>
<th>Name of parameters</th>
<th>Value of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times of experiment</td>
<td>50</td>
</tr>
<tr>
<td>Quantity of particles ( N_p )</td>
<td>20</td>
</tr>
<tr>
<td>Quantity of ants ( A )</td>
<td>20</td>
</tr>
<tr>
<td>Times of iteration ( M )</td>
<td>30</td>
</tr>
<tr>
<td>Information concentration factor ( \alpha )</td>
<td>1</td>
</tr>
<tr>
<td>Inspired expectation factor ( \beta )</td>
<td>2</td>
</tr>
<tr>
<td>Residual factor ( \rho )</td>
<td>0.95</td>
</tr>
<tr>
<td>Quantity of WSCT</td>
<td>20</td>
</tr>
<tr>
<td>Quantity of web service</td>
<td>30</td>
</tr>
<tr>
<td>pbest replacing factor ( cp )</td>
<td>0.4</td>
</tr>
<tr>
<td>gbest replacing factor ( cg )</td>
<td>0.6</td>
</tr>
<tr>
<td>Weight of QoS attribute ( W_p = 0.2, W_t = 0.5, W_r = 0.3 )</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3:** QoS attributes of WCST.

<table>
<thead>
<tr>
<th>Service</th>
<th>WCST</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td></td>
<td>9.0</td>
<td>23.0</td>
<td>16.0</td>
<td>24.0</td>
<td>19.0</td>
<td>20.0</td>
<td>11.0</td>
<td>14.0</td>
<td>6.0</td>
</tr>
<tr>
<td>P</td>
<td>1.0</td>
<td>15.0</td>
<td>13.0</td>
<td>14.0</td>
<td>13.0</td>
<td>8.0</td>
<td>8.0</td>
<td>16.0</td>
<td>6.0</td>
<td>16.0</td>
</tr>
<tr>
<td>R</td>
<td>0.5</td>
<td>0.1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>10.0</td>
<td>11.0</td>
<td>12.0</td>
<td>13.0</td>
<td>14.0</td>
<td>15.0</td>
<td>16.0</td>
<td>17.0</td>
<td>18.0</td>
</tr>
<tr>
<td>P</td>
<td>17.0</td>
<td>38.0</td>
<td>11.0</td>
<td>37.0</td>
<td>33.0</td>
<td>25.0</td>
<td>20.0</td>
<td>14.0</td>
<td>31.0</td>
<td>13.0</td>
</tr>
<tr>
<td>R</td>
<td>0.9</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.8</td>
<td>0.5</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
</tr>
</tbody>
</table>
CONCLUSION

In this article, author attempts to combine the advantages of these two algorithms to propose PACA and applies the algorithm into web services composition problem. This algorithm avoid the slow convergence of the PSOA and falling into local optimum, improves the efficiency of simple web service composition optimization. There are important theoretical referenced value to relevant issues. The further optimization of PACA is the next work of this subject and author will improve practicality to satisfy the actual demand of web services composition problem.

ACKNOWLEDGMENT

This study is supported by high and new technology industry foundation of Harbin Grant by No. 20120056.

REFERENCES