Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 14 (8): 805-810, 2014
ISSN 1812-5654 / DOI: 10.3923/jas.2014.805.810

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 net work. 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 heuristice 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. N_{WSCT_n} is the quantity of the web services in $WSCT_n$. The quantity of

Corresponding Author: Shujun Pei, College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, 150080, China

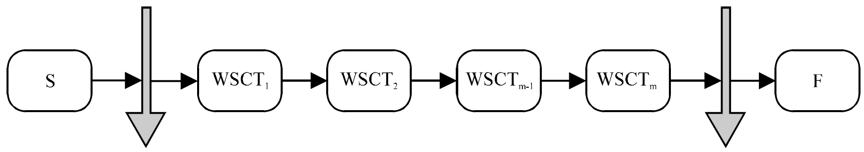

Fig. 1: Web services composition model

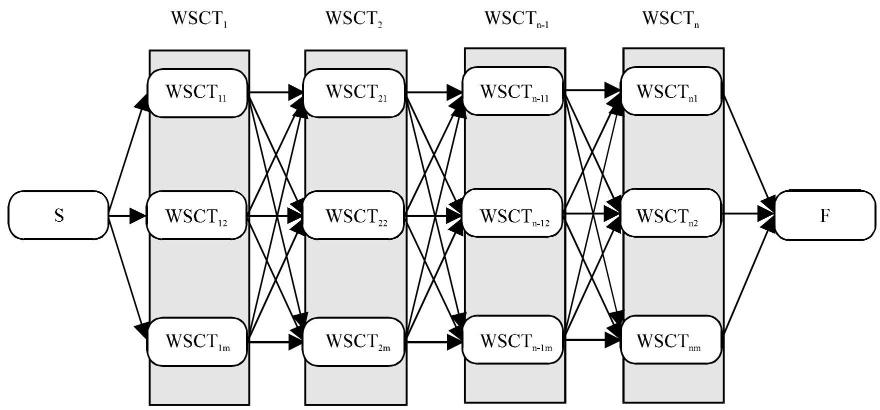

Fig. 2: Converted Web services composition model

Table 1: Calculating function of QoS attributes in combined web service

QoS attributes	Time(T(t))	Price(P(p))	Rob(R(t))
Expression	$\sum_{i=1}^{m} T(t_i)$	$\sum_{i=1}^{m} P(t_i)$	$\prod_{i=1}^{m} R(t_i)$

all the feasible solution is $N_{WSCT_1} \times N_{WSCT_2} \times N_{WSCT_3} \times ... N_{WSCT_N}$, it will be very time-consuming to find a optimal solution by general methods from such massive solutions, so it's essential to convert the original problem model.

Conversion of web services model: In current study of web services composition, scholars have emphasized the necessity and importance of QoS (Jiang *et al*., 2011). The combination methods of web services are different, the QoS description is not the same. In this article, we utilize the vector (Price, Time, Rob) to describe the QoS parameters, Price said that the price of web services, Time expressed web service response time, Rob represents web robustness. Because of the additivity of attribute Price and Time and the multiplicity of attribute Rob, the attribute Price and Time of the combined web service can be got by adding the Price and Time of single eb service and the attribute Rob of the combined web service can be got by multiplying the attribute Rob of single web service. So, the attribute value of the combined web service can be calculated by the function shown in Table 1.

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:

$$G = (P, WS_i, E, QoS_{ij})$$

$$P = \{S, WSCT_1, WSCT_2, WSCF_3 \ldots WSCT_n, F\}$$

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

$$WS_i = \{WSCT_{i1}, WSCT_{i2}, WSCT_{i3}, \ldots, WSCT_{ij}\}$$

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

$$E = \{(WSCT_{ik}, WSCT_{jy}) | WS_{in} \in WSCT_i, WS_{jn} \in WSCT_j, i \neq j\}$$

E represents the path between two web services in different WSCT:

$$QoS_{ij} = \{Price_{ij}, Time_{ij}, Resb_{ij}\}$$

QoS_{ij} represents QoS attributes of a specific web service ($WSCT_{ij}$).

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

Introducion 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 theirs 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 $P_i = (P_{i1}, P_{i2}, P_{i3},...P_{ij}|P_{i1}\in WS_1, P_{i2}\in WS_2... P_{ij}\in WS_j)^T$ 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, $V_i = (V_{i1}, V_{i2}, V_{i3}, V_{i4},..., V_{ij})^T$. 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:

$$QoS(p_{ij}) = \frac{W_r R(p_{ij})}{W_p P(p_{ij}) + W_t T(p_{ij})} \quad (1)$$

$$f(P_i) = \sum_{j=1}^{n} QoS(p_{ij}) \quad (2)$$

Wr, Wp, Wt is the weight of the QOS attribute, $\{W_r+W_p+W_t = 1\}$, R (p), P (p), T (p) denote the atomic web service QoS values of price, response time and robustness $pbest_i = f(P_i)$ $gbest = MAXf(P_i)$.

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

$$v_{ij}^{k+1} = v_{ij}^{k} + Select1_k(pbest_{ij}^{k} - x_{ij}^{k}) + Select2_k(gbest_j^{k} - x_{ij}^{k}) \quad (3)$$

$$Select1_k = \begin{cases} 1, cp\times rand_p > cg\times rand_g \\ 0, \quad\quad or \end{cases}$$
$$Select2_k = \begin{cases} 1, cp\times rand_p \le cg\times rand_g \\ 0, \quad\quad or \end{cases} \quad (4)$$

$$\text{randp} = \frac{\text{pbest}_i j^k}{\text{pbest}_{ij}{}^k + \text{gbest}_j^k} \qquad \text{randg} = \frac{\text{gbest}_j^k}{\text{pbest}_{ij}{}^k + \text{gbest}_j^k} \tag{5}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \tag{6}$$

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; randp expressed the probability that the particle coordinate would be updated by individual-extremum-value, randg expressed the probability that the particle coordinate would be updated by global-extremum-value, randp+randg = 1.

x^k_{ij} is the position of the jth dimension of the particle i at the Kth iteration, pbest_{ij}^k is the position of the jth dimension of individual-extremum-point of the particle i at the Kth iteration, gbest^k_j 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, $N_{min} \le N \le N_{max}$, 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 τ) that Improved ACO need based on paths collection O:

$$\tau_{ij} = C_1 \times \text{QoS}(O_{ij}) \tag{7}$$

τ_{ij} is the initial pheromone of the edge (i, j), C_1 is a constant, $\text{QoS}(O_{ij})$ 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 incloud the same edge (i, j), the τ_{ij} should be accumulated. To avoid premature convergence, we choice to use the max-min ant algorithm (MMAS) (Alaya *et al*., 2007), then $\tau_{ij} = \tau_{min}+\tau_{ij}$, τ_{min} is the minimum pheromone at every edge.

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

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\,\eta_{ij}^{\beta}(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}^{\alpha}(t)\,\eta_{is}^{\beta}(t)} & j \in \text{allowed}_k \\ 0 & \text{or} \end{cases} \tag{8}$$

$p^k_{ij}(t)$ means that the probability that ant k at node i select the next node j at the t-th iteration; allowed_k represents that nodes that ant k at node i can select; α represents Information concentration factor which means the importance of the pheromone in selecting node; β represents inspired factor which means the significance of the inspired information when ant select node:

$$q_{ij} = \sqrt{\frac{w_r R^2}{w_t T^2 + w_p P^2}} \tag{9}$$

$$w_t+w_p+w_r = 1 \tag{10}$$

q_{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 $\eta_{i,j} = C_2 \times q_{i,j}$. C_2 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:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t)+\tau_{ij}(t) \tag{11}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{i,j}^k(t) \tag{12}$$

ρ represents residual factor, It's used to avoid unlimited accumulation of pheromones, $\rho \subset [0,1)$, 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 increasement of the pheromone is too high or too low at one iteration to affect the next path-selection, can be setted as:

$$\tau_{min} \le \Delta t_{ij}(t) \le \tau_{max} \cdot \begin{cases} \Delta\tau_{ij}^k(t) = \dfrac{q_{ij}}{L_k} & \text{ant K pass the edge(i, j)} \\ 0 & \text{or} \end{cases} \tag{13}$$

$$L_k = C_3 \bullet \sqrt{\sum_{i=1}^{n-1}\left(\frac{1}{q_{i,i+1}}\right)^2} \quad (14)$$

L_k is the length of the path that ant k has passed, C_3 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 begin 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 α, inspiration factor β and residual factor ρ

Step 5: Iteration begin 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

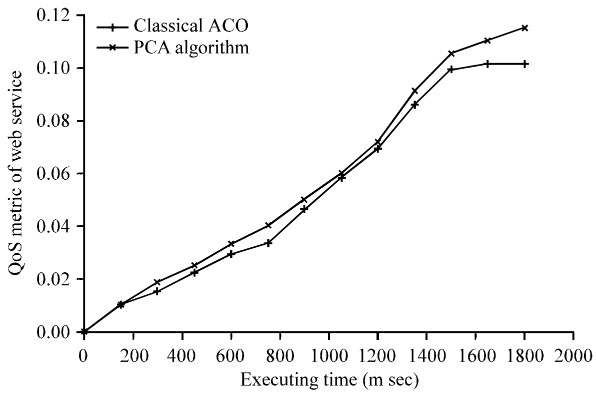

Fig. 3: Experiment data Figure

Table 2: Parameters that used in above formula

Name of parameters	Value of parameters
Times of experiment:	50
Quantity of particles N_p:	20
Quantity of ants A:	20
Times of iteration M:	30
Information concentration factor α	1
Inspired expectation factor β	2
Residual factor ρ	0.95
Quantity of WSCT	20
Quantity of web service	30
pbest replacing factor cp	0.4
gbest replacing factor cg	0.6
Weight of QoS attribute	$W_P = 0.2$, $W_T = 0.5$, $W_R = 0.3$

EXPERIMENTS AND ANALYSIS

To validate effect and feasibility of PACA 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. In order 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 WSCT'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 3: QoS attributes of $WCST_1$

	$WSCT_1$									
Service	0	1	2	3	4	5	6	7	8	9
T	9.0	23.0	16.0	24.0	19.0	20.0	11.0	14.0	6.0	37.0
P	1.0	15.0	13.0	14.0	13.0	8.0	8.0	16.0	6.0	16.0
R	0.5	0.1	0.6	0.4	0.6	0.8	0.3	0.5	0.8	0.5
	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0
T	28.0	38.0	11.0	37.0	33.0	25.0	20.0	14.0	31.0	13.0
P	17.0	10.0	13.0	19.0	9.0	1.0	17.0	13.0	7.0	20.0
R	0.9	0.2	0.1	0.2	0.8	0.5	0.3	0.1	0.2	0.4

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 theefficiency 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 practicability 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

Alaya, I., C. Solnon and K. Ghedira, 2007. Ant colony optimization for multi-objective optimization problems. Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, October 29-31, 2007, Patras, Greece, pp: 450-457.

Fang, Q., X. Peng, Q. Liu and Y. Hu, 2009. A global QoS optimizing web services selection algorithm based on MOACO for dynamic web service composition. Proceedings of the International Forum on Information Technology and Applications, Volume 1, May 15-17, 2009, Chengdu, China, pp: 37-42.

Gao, X.Y., L.Q. Sun and D.S. Sun, 2009. An enhanced particle swarm optimization algorithm. Inform. Technol. J., 8: 1263-1268.

Ge, J.K., Y.H. Qiu, C.M. Wu and G.L. Pu, 2008. Summary of genetic algorithms research. Appl. Res. Comput., 25: 2911-2916.

Jiang, H., X. Yang, K. Yin, S. Zhang and J.A. Cristoforo, 2011. Multi-path QoS-aware web service composition using variable length chromosome genetic algorithm. Inform. Technol. J., 10: 113-119.

Liu, L.P., Z.G. Chen and A.X. Liu, 2008. Research on web services composition based on particle swarm optimization. Comput. Eng., 34: 104-112.

Pfeffer, H., D. Linner and S. Steglich, 2008. Modeling and controlling dynamic service compositions. Proceedings of the 3rd International Multi-Conference on Computing in the Global Information Technology, July 27-August 1, 2006, Athens, pp: 210-216.

Yu, X. and T. Zhang, 2009. Convergence and runtime of an ant colony optimization model. Inform. Technol. J., 8: 354-359.