2009 International Forum on Information Technology and Applications

A Global QoS Optimizing Web Services Selection Algorithm based on MOACO for Dynamic Web Service Composition

Fang Qiqing [1,3], Peng Xiaoming [2,3], Liu Qinghua [3], Hu Yahui [3]
[1] Commanding Communications Academy, Wuhan 430010, China
[2] Computer School of Wuhan University, Wuhan 430072, China
[3] Air Force Radar Academy, Wuhan 430019, China
fangqiqing@yeah.net; ryen79@163.com,lqh@163.com,hyh5800@163.com

ABSTRACT: Dynamic Web Service composition is a key technology for building the service-oriented, loose coupling and integrated applications. In a web service composition system Web Services can be selected and integrated to generate a workflow for the satisfaction of user functional requirements. Usually, for each task in the workflow a set of alternative Web Services with similar functionality is available, and that these Web Services have different QoS parameters. This leads to the general optimization problem of how to select Web Services for each task so that the overall QoS requirements of the composition are satisfied and optimal. However, the existing approaches are almost QoS local optimization or mono-objective based, and can not achieve global constraints set by user. In this paper, we present a novel global QoS optimizing and multi-objective Web Services selection algorithm based on Multi-objective Ant Colony Optimization (MOACO) for the Dynamic Web Service composition. Firstly, we propose a model of Web Services selection with QoS global optimization and convert this problem into a multi-objective optimization problem with user constraints. Furthermore, based on the MOACO, the proposed algorithm simultaneously optimizes the different QoS parameters of the workflow. In this way, a set of optimal solutions, known as Pareto set, is calculated. Experimental results prove the proposed algorithm outperforms the recently published QoS Global Optimization Based on Multi-objective Genetic Algorithm (MOGA), specially designed for solving the Web Services selection problem.

KEYWORDS: Web Services selection; Ant Colony Optimization; Global Optimization; Multi-Objective

I. INTRODUCTION

Service Oriented Architecture (SOA) is becoming the main technology for enterprise information integration. It may be the answer for the forever-change requirements and difficult integration problem of software development. Dynamic composition of Web Services[1] is the key technology of SOA implementation, and Web Services selection is an important issue in dynamic Web Service composition.

The complexity of Web Services selection for Web Service composition includes three main factors: 1) the large number of dynamic changed Web Services instances with similar functionality that may be available to a complex service; 2) the different possibilities of integrating service instance components into a complex service process; 3) various performance requirements (e.g. end-to-end delay, service cost and reliability) of a complex service. Web Service composition thus creates a QoS engineering problem since the service selection must select specific services to compose an efficient complex service with QoS assurance. However, the existing approaches[2-6] on service selection of dynamic Web Service composition are almost QoS local optimization or mono-objective based, and can not resolve the problem of Web Services selection with QoS global optimization and multi-objective.

Ant Colony Optimization (ACO) is a meta-heuristic proposed by Dorigo et al. [7]. The basic idea of it is to model the problem to solve the search for a minimum cost path in a graph, and to use artificial ants to search for good paths. The behavior of artificial ants is inspired from real ants: they lay pheromone on components (edges and/or vertices) of the graph and they choose their path with respect to probabilities that depend on pheromone trails that have been previously laid by the colony; these pheromone trails progressively decrease by evaporation. Intuitively, this indirect stigmergetic communication means aims at giving information about the quality of path components in order to attract ants, in the following iterations, towards the corresponding areas of the search space. In the last few years, ACO has empirically shown its effectiveness in the resolution of several different NP-hard combinatorial optimization problems. But there has been little or no reported application of ACO to Global Optimization of dynamic Web Services selection.

This work proposes for the first time to solve the Web Services selection problem using the Multi-Objective Ant Colony Optimization(MOACO), introduced in [8]. This algorithm optimizes several objectives simultaneously. Experimental results have recently demonstrated that MOACO is the best multi-objective ACO algorithm for the bi-objective Traveling Salesman Problem (TSP) [9].

Besides, to verify the results obtained with the proposed algorithm, it is compared to the recently published QoS Global Optimization Based on Multi-objective Genetic Algorithm[10] (MOGA), specially designed for solving the Web Services selection problem.

The rest of the paper is organized as follows.

Section 2 discusses related works. Section 3 presents the Web Service composition model and defines some key concepts used throughout the paper. Section 4 formulates the global service selection problem and describes a Multi-Objective Ant Colony Optimization method to efficiently solve it. Section 5 presents a set of experiments comparing the global optimized approach with the recently published

QoS Global Optimization Based on Multi-objective Genetic Algorithm (MOGA).

II. Related Works

The majority of existing approaches[2-6] on Web Services selection almost QoS locally optimal or mono-objective. In [4,5], multi-restriction parameters of service QoS are transformed into single-goal function by weighted linear method. [6] has studied a similar approach. All of them use the integer linear programming method to solve the service selection problem, which is highly sensitive to the weights vector, and linear programming requires that objective functions and constraints are linear, to some extent practicability of this method is restricted.

Recently, [10] proposes a global optimizing and multi-objective algorithm based on Multi-Objective Genetic Algorithm and it simultaneously minimizes two objectives functions and two constraints for the case in [10]. But there has been little or no reported application of MOACO to Global Optimization of Dynamic Web Services selection.

III. Web Service Composition Model

Definition 1 (User requirements Model):

$R_{CW} =< input, output, precondition, result, Qos >$

Where, *input* and *output* are the set of input and output parameters of the composite Web Service; *precondition* specifies things that must be true of the world before executing the composite Web Service. *effect* characterizes the physical side-effects that execution of the composition Web Service. These four service description elements are often referred to as IOPE. It specifies the user's functional requirements and is possible to use algorithmic approaches, including AI Planning techniques, to construct complex services from simpler components without depending on a human programmer[11].

$Qos =< q_1, q_2, ..., q_m >$,it specifies the user's non-functional requirement. q_i is a QoS property, such as reliability, cost or response time.

Definition 2 (Web Service Type Model):

$WS_T =< input, output, precondition, result >$, A Web Service Type WS_T is a set of Web Service instances with similar functionality and consists of elements to specify the functionality of this Web Service Type.

Definition 3 (Web Service Instance Model):

A Web Service Instance WS_I consists of its functionality, service type and it's QoS properties.

$WS_I =< IOPE, WS_T, Qos >$

Where, $IOPE =< input, output, precondition, result >$,

WS_T is the service type of WS_I.

$Qos =< q_1, q_2, ..., q_m >$.

Further more, we can explicitly present a service instance WS_I of service type WS_T as WS_I^T.

Definition 4 (Abstract Service Plan Model):

An Abstract Service Plan is composed of individual Web Service Type. It means that a workflow template is defined but any concrete services are not allocated to the workflow. We denote:

$AP =< WS_{T1}, WS_{T2}, ..., WS_{Tn} >$,where WS_T is a web service type.

Definition 5 (Concrete Workflow Model):

$CW = \{WS_I^{T1}, WS_I^{T2}, ..., WS_I^{Tn}\}$,where WS_{Ik} is a web service instance.

Definition 6 (Web Service Composition Model):

Given a set of Web Service Types and the set of instances for each type, along with the specification of a new service which includes user functional requirements and non-functional requirements(QoS), create an concrete workflow that stitches together the desired functionality from the existing services, while satisfying the non-functional requirements.

IV. Global QoS Optimization Web Services Selection Algorithm based on MOACO

A. Problem Formulation

Assuming that there are total n steps in the Abstract Service Plan $AP =< WS_{T1}, WS_{T2}, ..., WS_{Tn} >$ which was generated by the first stage, each service type has M instances and each Web Service Instance has three QoS parameters, $WS_I.QoS =< Cost, Time, Reliability >$,Where, Cost is the price of the service instance. Time is the time taken to deliver services between service requesters and providers. Reliability represents the ability of a Web Service to perform its required functions under stated conditions for a specified time interval. User non-functional requirements for the concrete workflow is $R_{cw}.QoS = [C, T, 1/R]$,where C, T, R denote the global cost, the global time ,and the global reliability constraints of the Concrete Workflow given by the user.

Definition 7 (Service Instance Selection Graph):

Web Service Instance Selection graph is modeled as a weighted directed graph $G =< V, E >$ as Fig. 1. Where $V = WS_I^{T1} \cup WS_I^{T2} \cup ... \cup WS_I^{Tn} \cup \{S, T\}$, $WS_{Ij}^{Ti} \in WS^{Ti} \in AP$ ($i \le n, j \le M$) represents the Web Service Instances of the Web Service Type belong to the Abstract Service Plan. S and T are virtual beginning vertex and target vertex added to the graphic for analyze the problem of service selection. E is the set of potential links between the Web Service Instances of the neighbored service types in the Abstract Service Plan. The weight vector of each node is the QoS parameter vector of Web Service Instance. For $\forall n \in V$,We define each parameters as $Cost(n)$, $Time(n)$ and $Reliability(n)$ .Specially, the Qos of the virtual nodes(S and D) and the links in this graph is 0.

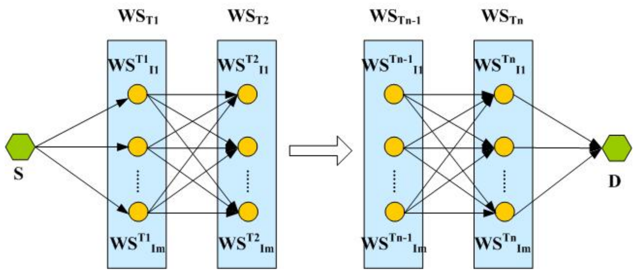

Figure 1. Web Service Instances Selection Graph

Based on the directed graph $G=<V,E>$,let $P=P(S,D)$ be a path from the source node S to destination node D .We define the following parameters for the path P :

$$Cost(P)=\sum_{n\in P} Cost(n)\,. \tag{1}$$

$$Time(P)=\sum_{n\in P} Time(n)\,. \tag{2}$$

$$\mathrm{Re}\,liability(P)=\prod_{n\in P} \mathrm{Re}\,liability(n)\,. \tag{3}$$

Definition 8 (Web Service Instance Selection Problem) :

Using the above definitions, a Web Service Instances selection problem may be stated as a MOP [8] that tries to find a path $P=P(S,D)$ that simultaneously satisfies the following objectives(searching for an optimal/near-optimal Concrete Workflow that minimizes the cost and the time, maximizing its reliability, while also satisfies the user QoS restrictions):

Considering there two solutions P and P' are the path from the source node S to destination node D :

$x=[Cost(P),Time(P),1/\mathrm{Re}\,liability(P)]$,

$z=[Cost(P'),Time(P'),1/\mathrm{Re}\,liability(P')]$.

Only one of the following three conditions can be given:

1) *$x\succ z$ (x dominates z),* if $x_i \le z_i \wedge x_i \ne z_i, \forall i\in\{1,2,3\}$;
2) *$z\succ x$ (z dominates x),* if $z_i \le x_i \wedge z_i \ne x_i, \forall i\in\{1,2,3\}$;
3) *$x \sim z$ (x and z are non-comparable),* if $x_i \succ z_i \wedge z_i \succ x_i, \forall i\in\{1,2,3\}$

A decision vector P is non-dominated with respect to a set Q if: $P\succ P' \wedge P\succ R_{CW}.QoS, P'\in Q$,When P is non-dominated with respect to the user QoS requirements and the whole domain of feasible solutions, it is called an optimal Pareto solution; therefore, the Pareto optimal set P_{true} may be formally defined as:

$P_{true}=\{\ P\in X_f \mid P$ is non-dominated with respect to $X_f\ \}$.

The corresponding set of objectives $P_{true}=f(X_{true})$ constitutes the Optimal Pareto Front.

The Multi-objective Ant Colony Optimization algorithm is used to solve this problem in this paper.

B. Multi-objective Ant Colony Optimization algorithm for Web Service Instance Selection

Our Web Service Instance Selection Algorithm is base on the generalization of the Multi-objective Ant Colony Optimization algorithm (MOACO), proposed in [8]. This approach uses a colony of ants for the construction of m solutions P at every generation. Then, the known Pareto Front P_{know} [12] is updated, including all non-dominate solutions. Finally, the pheromone matrix τ_{ij} is updated. We modified it to solve the Web Service Instances selection problem, as shown in Fig. 2.

```
**Procedure** MOACO4WS
initialize $S, D, N_r, \varphi$
initialize $\tau_{ij}$
**while** stop criterion is not verified
    **repeat-for** $k=1$ to $m$
        Construct Solution $P$ (See Fig. 4)
        **if** ( $P$  $\{P_x \mid P_x\in P_{know}\}$ ) then
      $P_{know}=P_{know}\cup P-\{P_y \mid P\rightarrow P_y\}\forall P_y\in P_{know}$
        **end-if**
    **end-repeat-for**
    Update of $\tau_{ij}$
**end-while**
**End-Procedure**
```

Figure 2. General Procedure of MOACO

The update of pheromone matrix τ_{ij} depends on the state of P_{know} . If P_{know} was modified, then τ_{ij} is re-initialized $(\tau_{ij}=\tau_0)$ to improve exploration; otherwise, a global update of τ_{ij} is made using the solutions of P_{know} for a better exploitation, as shown in Fig. 3.

```
**repeat-for** every $P\in P_{know}$
    **repeat-for** every $P_{know}$
  $\tau_{ij}=(1-\rho)\cdot\tau_0+\rho\cdot\Delta\tau$
    **end-repeat-for**
**end-repeat-for**
```

Figure 3. Global Update of τ_{ij}

With $\Delta\tau = 1 \Big/ \sum_{\forall T \in P_{know}} (\sqrt{Cost^2(P) + Time^2(P) + (1/\mathrm{Re}liability(P))^2}$,

and $\rho \in (0,1]$,Trail persistence.

```
Precedure Construct Solution
Set tabuList = $\phi$ ; Set $P = \phi$ ;
Set i = 0;
while ( $|P| < n$ ) do
    Set $N_i$ as the nodes in the neighborhood of node i that the ant has not visited yet.
    If $N_i \neq \phi$ Then
        repeat-for each node $j \in N_i$
            Set the probability of being chosen
            $p_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \subset N_i} [\tau_{is}]^\alpha [\eta_{is}]^\beta}$
            select randomly next node $j$
            $P = P \cup \{j\}$
            $tabuList = tabuList \cup \{j\}$
            Update parameters
            $i = j$ ;
        end-repeat-for
    else
        reset parameters
end-while
end precedure
```

Figure 4. Procedure for constructing solution

In the above Procedure η_{ij} is the heuristic value of moving from node i to node j. We set it as:

$$\eta_{ij} = 1 \Big/ \sqrt{Cost^2(j) + Time^2(j) + (1/\mathrm{Re}liability\ (P))^2}$$

V. EXPERIMENTAL RESULTS

In the following section we describe the computational tests which we performed in order to compare the solution quality and performance of the proposed Algorithm (MOACO) with the recently published QoS Global Optimization Based on Multi-objective Genetic Algorithm (MOGA)[10].

Four tests were performed for the 4 groups presented in Table 1（Where n is the Web Service Type number of the Abstract Service Plan and M is the Web Service Instance number of each Web Service Type）. Each test consists of 2 runs for 100 and 200 iterations. Both algorithms, MOACO and MOGA, have been implemented on a computer with Intel Pentium Dual 2.00GHz processor,4GB of RAM, and the operating system Windows XP professional. The compiler used was Microsoft Visual C++ 2005.

TABLE I. AMOUNT OF WEB SERVICE TYPE AND WEB SERVICE INSTANCE USED FOR THE TESTS

Test Group	n	M
Group1	5	10
Group2	10	20
Group3	20	20
Group4	20	40

The QoS values of functional equivalent services were varying according to some gaussian distribution function, and better response time or availability offers corresponded to higher costs. The user's constraint for the concrete workflow is $\varphi = [C, T, R] = [800, 10, 0.3]$.The number of ants in each colony for MOACO are set to 30. Other parameters of algorithm are set $\alpha = 1.0$, $\beta = 1.0$, $\rho = 0.3$.

The comparison procedure used for each group was the following:

1) Each algorithm was run five times to calculate an average.

2) For each algorithm, five sets of non-dominated solutions were obtained $(P_1, P_2, ..., P_5)$, and a overpopulation P_{over} was calculated as the union of the five sets.

3) Dominated solutions were deleted from P_{over} , forming the Pareto set of each algorithm:

P_{ACO} : Pareto Front obtained of the 5 runs using MOACO.

P_{GA} :Pareto Front obtained of the 5 runs using MOGA.

4) A set of solutions $\hat{P}$ was obtained as follows:

$\hat{P} = P_{ACO} \vee P_{GA}$.

5) Dominated solutions were eliminated from $\hat{P}$, to obtain an approximation of P_{true} , called P_{apr} .

Table 2 presents the number of solutions $P \in P_{apr}$ found for every test group.

TABLE II. AMOUNT OF OPTIMAL SOLUTIONS FOR EACH TEST GROUP

	Group1	Group2	Group3	Group4
$\|P_{apr}\|$	15	24	36	51

The experiments results is shown in Table 3 .Where, the average number of concrete workflow solutions of each algorithm that are in P_{apr} ,denote as $[\in P_{apr}]$ .The set of solutions that are dominated by P_{apr} is denoted as $[P_{apr} \succ]$ .The number of found solutions is $[|P_{a\lg}|]$ and the percentage of solutions present in P_{apr} is $[\%(\in P_{apr})]$.

TABLE III. AMOUNT OF OPTIMAL SOLUTIONS FOR EACH TEST GROUP

Test Group	Iterations	Algorithm	$\in P_{apr}$	$P_{apr} \succ$	$\lvert P_{alg} \rvert$	$\%(\in P_{apr})$
Group1	100	P_{ACO}	14.5	0	14.5	97%
		P_{GA}	13.8	0	13.8	92%
	200	P_{ACO}	15	0	15	100%
		P_{GA}	15	0	15	100%
Group2	100	P_{ACO}	22.1	4.2	26.3	91%
		P_{GA}	17.3	1	18.3	72%
	200	P_{ACO}	23.3	5.1	28.4	97%
		P_{GA}	21	7.3	28.3	88%
Group3	100	P_{ACO}	30.1	3.3	33.4	84%
		P_{GA}	26.8	6.4	33.2	74%
	200	P_{ACO}	33	2.1	35.1	92%
		P_{GA}	29.3	7.3	36.6	81%
Group4	100	P_{ACO}	41	6.3	47.3	80%
		P_{GA}	33	9.4	42.4	64%
	200	P_{ACO}	45.2	7.1	52.3	89%
		P_{GA}	38.7	12.3	51	76%

From the above table, we can see:

- In Group1, when the iteration number is 200,both the founded solutions of MOACO and MOGA are almost belong to P_{apr} .But when the iteration number is 100,MOACO overcoming MOGA.
- For a larger number of Web Service Instance and Web Service type (Group2-Group4), the MOACO also demonstrated to be better than the MOGA. In fact, MOACO obtained a larger number of solutions belonging to P_{apr} , for all run times.

Also, the running time between our algorithm and that the genetic algorithm is compared in this experiment. The time for finding out the optimal concrete workflows is showed in Fig. 5. It can be seen that the time astringency of algorithm proposed in this paper is better than the MOGA.

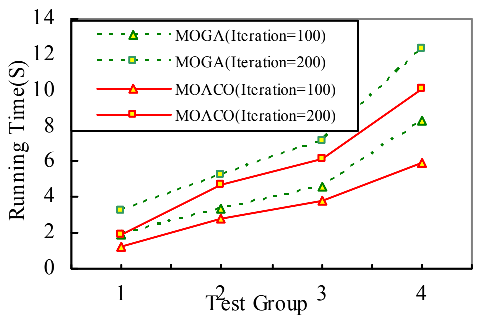

Figure 5. The comparison of execution time

VI. CONCLUSION

A global optimization and multi-objective Web Services selection algorithm based on MOACO is proposed to resolve the question of multi-objective services composition optimization with QoS constraints for dynamic Web Service Composition. Considering the presented experimental results, MOACO is able to find more best solutions than the recently published QoS Global Optimization Based on Multi-objective Genetic Algorithm (MOGA). Experimental results also indicate the feasibility and efficiency of the algorithm.

As future work, we will propose the QoS assessment model for the other types of composition processes, such as switch, while and so on. We will compare this algorithm with different parameter settings.

ACKNOWLEDGMENT

The work reported in this paper is supported by the National High Technology Research and Development Program of China (2007AA01Z138).

REFERENCES

[1] B. Benatallah, M. Dumas, Q.Z. Sheng, A. Ngu, "Declarative composition and peer-to-peer provisioning of dynamic Web services," Proc. 18th Int'l Conf. on Data Engineering, IEEE Computer Society, San Jose, 2002, pp. 297-308.

[2] F. Casati, S. Ilnicki, L.J. Jin, V. Krishnamoorthy and M.C. Shan, "eFlow: A platform for developing and managing composition e-services," Tech.Report, HPL-2000-36, HP laboratories Palo Alto, 2004.

[3] P. Grafen, K. Aberer, Y. Hoffner and Y.H. Ludwig," Cross-low:Cross-organizational workflow management in dynamic virtual enterprises," International Journal of Computer Systems Science & Engineering, 2000,15(5),pp. 277-290

[4] Y.T. Liu, H.H. Anne, L.Z. Zeng, "QoS Computation and Policing in Dynamic Web Services selection," Proc. WWW2004, ACM, New York, 2004, pp. 66-73.

[5] C. Jorge, S. Amit, M. John, "Quality of Service for workflows and Web Service Processes," Journal of Web Semantics, 2004,1(3),pp. 281-338

[6] R. Aggarwal, K. Verma, J. Miller and Milnor, "Constraint driven Web service composition in METEOR-S," Proc. IEEE International Conference on Service Computing (SCC'04), IEEE Computer Society, Shanghai, 2004, pp. 23−30.

[7] M. Dorigo, and G.D. Caro, "The Ant Colony Optimization meta-heuristic," New Ideas in Optimization, McGraw Hill, London,1999,pp.11-32.

[8] M. Schaerer, and B. Barán, "A Multi-objective Ant Colony System For Vehicle Routing Problem With Time Windows," Proc. IASTED International Conference on Applied Informatics, Innsbruck,2003.

[9] C. Garcí a-Martínez, O. Cordón, and F. Herrera, "An Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-criteria TSP," Proc ANTS 2004 -Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence,Springer-Verlag, Bruselas, 2004.

[10] Shulei Liu, Yunxiang Liu, Ning Jing, Guifen Tang, and Yu Tang, "A Dynamic Web Services selection Strategy with QoS Global Optimization Based on Multi-objective Genetic Algorithm," Proc. Grid and Cooperative Computing (GCC 2005), Springer Berlin, Heidelberg,2005, pp. 84-89.

[11] Sirin E, Parsia B, Wu D, Hendler J and Nau D, "HTN planning for Web service composition using SHOP2," Journal of Web Semantics, Elsevier, 2004, pp. 377-396.

[12] D. A. Van Veldhuizen, “Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations,” Ph. D. thesis Air Force Institute of Technology, 1999.