

# Web services composition based on global semantics and QoS-aware aspect

Zhang Peiyun<sup>1,2</sup> Huang Bo<sup>1</sup> Sun Yamin<sup>1</sup>

(<sup>1</sup> School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)

(<sup>2</sup> School of Mathematics and Computer Science, Anhui Normal University, Wuhu 241000, China)

**Abstract:** A global semantics matching and QoS-awareness service selection are proposed when aimed at a web services composition process. Both QoS-aware matching and global semantic matching are considered during the global matching. When there are demands for global semantic matching and QoS of service composition, a concrete service set which meets the demands is selected for the whole service composition process and an optimal solution is also achieved. A QoS model is built and the corresponding evaluation method is given for the matching of the service composition process. Based on them, a genetic algorithm is proposed to achieve the maximal global semantic matching degree and fulfill the QoS requirements for the whole service composition process. Experimental results and analysis show that the algorithm is feasible and effective for semantics and QoS-aware service matching.

**Key words:** semantic matching; global matching; QoS model; genetic algorithm

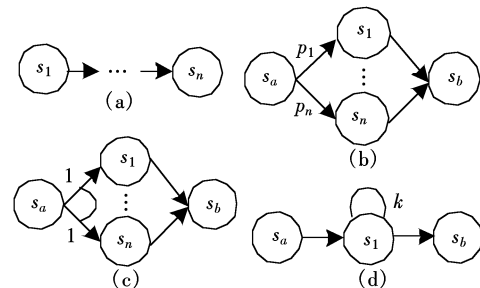
Nowadays, there are correlative studies on services composition based on QoS, such as integer planning<sup>[1-4]</sup>, heuristic approach<sup>[5]</sup> and genetic algorithm<sup>[6-9]</sup>, etc. It is pointed out that integer linear planning consumes too much time and faces the NP-hard problem during the service optimal composition process. The present research shows that, static web services composition cannot satisfy the demands of actual application, while fully intelligent services composition is a very complex process. Therefore, many applications and studies have focused on semi-automation services composition in which services composition process models need to be built. Appropriate services need to be found among numerous services to satisfy the requirements of users when aimed at the concrete model. As an intelligent optimal method, a genetic algorithm has the advantage of parallel computing and the ability of finding the optimized solution from a colony, and it does not need the heuristic formula which is related to an application background. The genetic algorithm only needs objective functions and corresponding fitness values; therefore, it is widely applied to find solutions for NP-complete problems. We adopt the genetic algorithm based on multi-QoS constraints to reach the global optimal targets and improve on the optimized composition approaches, because we pay attention to the issues of QoS con-

straints and semantic matching.

## 1 Global Semantics and QoS-Aware

### 1.1 QoS model, semantic matching and evaluation

International Organization for Standardization ISO8402 and ITUE. 800 hold that QoS consists of some non-functional properties, which include the cost, time, availability and reliability of a web service. Different control structures have different evaluation methods for those different QoS properties. Four control structures are illustrated in Fig. 1.



**Fig. 1** Four control structures. (a) Sequence; (b) Choice; (c) Parallel; (d) Loop

According to the four control structures, the corresponding aggregation functions and semantic matching are illustrated in Tab. 1. The first column in Tab. 1 shows the non-functional properties of web services, in which similarity is the semantic matching of control structure and the value of  $\text{sim}(rs_i, ws_i)$  comes from results of local semantic matching. In Tab. 1,  $p_i$  represents the chosen probability of the branch of choice structure and  $\sum_{i=1}^n p_i = 1$ , and  $k$  represents the number of loops.

### 1.2 Genetic algorithm based on global semantics and QoS-aware

Multi-objective optimization needs to be made as far as possible in a given field during multi-objective decision making process and the genetic algorithm can solve the problem. The general solution to multi-objective and multi-constraint problem is unifying multi-objective to one objective and optimizing the single objective. Based on the genetic algorithm, objective function  $F(P)$  is designed which utilizes four QoS properties (such as cost, time, availability and reliability) and semantic matching as objective guideline to minimize the values of cost and time while maximizing the values of availability, reliability and semantic matching. We convert the multi-objective optimization problem to a single objective problem by using weighted law. The objective function and its constraints are shown as follows:

Received 2008-04-15.

**Biography:** Zhang Peiyun (1974—), female, graduate, lecturer, njstzpy@yahoo.com.cn.

**Foundation items:** Specialized Research Fund for the Doctoral Program of Higher Education (No. 20050288015), Innovation Funds of Nanjing University of Science and Technology.

**Citation:** Zhang Peiyun, Huang Bo, Sun Yamin. Web services composition based on global semantics and QoS-aware aspect[J]. Journal of Southeast University (English Edition), 2008, 24(3): 296 – 299.

**Tab. 1** Aggregation functions and semantic matching of different QoS of different control structures

Properties	Sequence	Choice	Parallel	Loop
Cost	$\sum_{i=1}^n \text{Cost}(c_i)$	$\sum_{i=1}^n p_i \text{Cost}(c_i)$	$\sum_{i=1}^n \text{Cost}(c_i)$	$k\text{Cost}(c)$
Time	$\sum_{i=1}^n \text{Time}(t_i)$	$\sum_{i=1}^n p_i \text{Time}(t_i)$	$\max_{i=1}^n (\text{Time}(t_i))$	$k\text{Time}(t)$
Availability	$\prod_{i=1}^n \text{Availability}(a_i)$	$\prod_{i=1}^n p_i \text{Availability}(a_i)$	$\min_{i=1}^n (\text{Availability}(a_i))$	$\prod_{i=1}^k \text{Availability}(a)$
Reliability	$\prod_{i=1}^n \text{Reliability}(r_i)$	$\prod_{i=1}^n p_i \text{Reliability}(r_i)$	$\min_{i=1}^n (\text{Reliability}(r_i))$	$\prod_{i=1}^k \text{Reliability}(r)$
Similarity	$\sum_{i=1}^n \text{sim}(rs_i, ws_i)$	$\sum_{i=1}^n p_i \text{sim}(rs_i, ws_i)$	$\sum_{i=1}^n \text{sim}(rs_i, ws_i)$	$\text{sim}(rs_i, ws_i)$

$$\max F(P) = \sum_{i=1}^m (W_i Q_i') + W_s S' \quad (1)$$

s. t.

$$\begin{cases} Q_i(P) < \text{CSQ}_i(P) & \text{if } Q_i \text{ is cost constraint} \\ Q_i(P) > \text{CSQ}_i(P) & \text{if } Q_i \text{ is efficiency constraint} \\ S(P) > \text{CSQ}_s(P) \\ W_i > 0, W_s > 0, \sum_{i=1}^m W_i + W_s = 1 \end{cases} \quad (2)$$

where

$$Q_i' = \begin{cases} \frac{Q_{i,\max}(P) - Q_i(P)}{Q_{i,\max}(P) - Q_{i,\min}(P)} & \text{if } Q_i \text{ is cost constraint and } Q_{i,\max}(P) - Q_{i,\min}(P) \neq 0 \\ \frac{Q_i(P) - Q_{i,\min}(P)}{Q_{i,\max}(P) - Q_{i,\min}(P)} & \text{if } Q_i \text{ is efficiency constraint and } Q_{i,\max}(P) - Q_{i,\min}(P) \neq 0 \\ 1 & \text{if } Q_{i,\max}(P) - Q_{i,\min}(P) = 0 \end{cases} \quad (3)$$

$$S' = \begin{cases} \frac{S(P) - S_{\min}(P)}{S_{\max}(P) - S_{\min}(P)} & \text{if } S_{\max}(P) - S_{\min}(P) \neq 0 \\ 1 & \text{if } S_{\max}(P) - S_{\min}(P) = 0 \end{cases} \quad (4)$$

In Eq. (1),  $F(P)$  is the objective function and  $P$  represents the execution path.  $Q_i(P)$  and  $S(P)$  represent the values of the  $i$ -th QoS and the semantic matching of execution path  $P$  respectively;  $m$  is the number of constraints;  $W_i$  and  $W_s$  represent the weights of  $Q_i(P)$  and  $S(P)$ , respectively. The weight factor is designated according to the degree of importance of each property. Formula (2) shows that the constraint set of objective functions consists of multi-QoS and semantic matching which realize a certain constraint value  $\text{CSQ}_i(P)$ . In this paper, Time and Cost are cost constraints while Availability and Reliability are efficiency constraints. Similarity is also an efficiency constraint. According to cost and efficiency constraints,  $Q_i(P)$  and  $S(P)$  are standardized in formulae (3) and (4).  $Q_{i,\max}(P)$  and  $Q_{i,\min}(P)$  are the maximum value and the minimum value of  $Q_i$  of the composite service, respectively.

Considering the solving framework of the genetic algorithm and the characteristics of the problem, we conceive the idea of the genetic algorithm in this paper in the hope that we code certain web service composition processes into a chromosome and produce a more suitable chromosome for users by crossover and mutating and selection operations. The process continues and realizes global search in solution space to maximize users' objective functions; at the same time the QoS constraints and semantic matching are satisfied.

The algorithm stops when obtaining a suitable chromosome for users, the chromosome is a concrete service composition sequence corresponding to an abstract service composition. The corresponding genetic algorithm (GE\_AL) is shown as follows:

### Algorithm GE\_AL

**Step 1** An initial population is produced and  $N$  chromosomes are produced randomly. The number of genes in each chromosome is  $L$  and each gene is coded in natural numbers which comes from the range of  $[1, M]$ .  $N$  is the population number;  $L$  is the abstract services number of the composite services process;  $M$  is the candidate number of each abstract service.

**Step 2** The fitness function is designed and fitness values of individual chromosomes are computed by using a fitness( $x$ ) function, and then an individual chromosome is selected by the fitness value. The bigger the value, the higher the chance of joining to construct a new population.

**Step 3** The next generation is selected by roulette gambling with elitist selection strategy, in order to strengthen the ability of keeping an excellent chromosome and overcome the random error of sampling. If the excellent chromosome is not copied in the next generation, it will be copied to the following generation and the worst one will be deleted.

**Step 4** Two selected individual chromosomes intercross according to crossover probability  $P_c$ ,  $P_c \in [0, 1]$ . Single point crossover is adopted and breakpoint is obtained randomly.

**Step 5** Random mutation is adopted with probability  $P_m$  and  $P_m \in (0, 1)$ . A chromosome is selected randomly (an abstract web service of the service composition process) and one candidate service is utilized to replace the current concrete service of the chromosome.

**Step 6** The next generation is accepted and is put into new population.

**Step 7** The stop guideline is set. The maximum evolution generation  $\text{Gen}_{\max}$  is set beforehand which is as the stop condition. If the condition is satisfied, then the GE\_AL algorithm stops and the best solution of the current population is put out. If the condition is not met then go to step 2.

The coding pattern of GE\_AL is an integer code which codes a service composition process to a chromosome. The gene in the chromosome represents the number of an abstract service whose value comes from the range of candidate numbers according to the abstract service. In step 2 when constructing the fitness function, a limited optimized problem

adopts the penalty function method to punish the chromosomes which do not meet the constraint conditions. We can include penalty function and objective function to fitness function(fitness) which is shown as follows:

$$\text{fitness} = F(P) - \lambda \text{Punish}(P) \quad (5)$$

where  $F(P)$  is from formula (1),  $\text{Punish}(P)$  is the penalty function and  $\lambda$  is the scale factor of penalty function and  $\lambda > 0$ .

$$\text{Punish}(P) = \sum_{i=1}^m \left( \frac{\Delta Q_i}{Q_{i,\max}(P) - Q_{i,\min}(P)} \right) - \frac{\Delta S}{S_{\max}(P) - S_{\min}(P)} \quad (6)$$

where  $m$  is the number of QoS,  $\Delta Q_i$  shows the value of  $Q_i$  is beyond or under the value of  $\text{CSQ}_i$ , and  $\Delta S$  represents the value of  $S$  under  $\text{CSQ}_i$ . The expressions of  $\Delta Q_i$  and  $\Delta S$  are as follows:

$$\Delta Q_i = \begin{cases} Q_i(P) - \text{CSQ}_i & \text{if } Q_i(P) > \text{CSQ}_i \text{ and } Q_i \text{ is cost constraint} \\ \text{CSQ}_i - Q_i(P) & \text{if } Q_i(P) < \text{CSQ}_i \text{ and } Q_i \text{ is efficiency constraint} \\ 0 & \text{others} \end{cases}$$

$$\Delta S = \begin{cases} \text{CSQ}_s - S(P) & \text{if } S(P) < \text{CSQ}_s \\ 0 & \text{others} \end{cases}$$

It is a critical problem whether the genetic algorithm is convergent to a global optimal solution. Rudolph has proved that the standard genetic algorithm has no convergence solution<sup>[10]</sup>; the reason is that the optimal solution has disappeared.

**Theorem 1** If there is a service composition path which meets the constraint conditions, then the GE\_AL algorithm can search for the feasible path when given big enough population and maximum evolution generation  $\text{Gen}_{\max}$ .

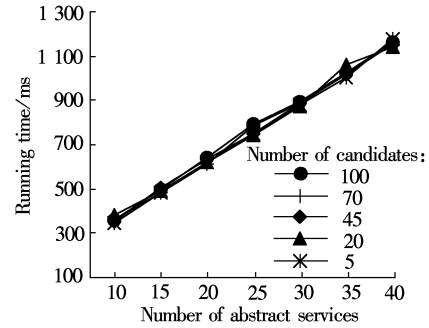
**Proof** In the GE\_AL algorithm, four strategies are adopted when searching for the feasible path: 1) Mutation probability  $P_m \in (0, 1)$ ; 2) Crossover probability  $P_c \in [0, 1]$ ; 3) Proportion selection; 4) Optimal individual chromosome is reserved before selection operation for the next optimal generation. Theorem 2.7 in Ref. [10] proved that the genetic algorithm has a convergent optimal solution when the four conditions are met. Therefore, the GE\_AL algorithm can search for feasible composition path when given big enough population and maximum evolution generation.

## 2 Experiments

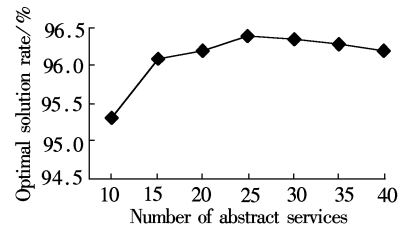
Given the number of abstract service of composition processes and the candidate number of each abstract service, the QoS values of each candidate service and local semantic matching are generated randomly. The number of the population is 50. The maximum evolution generation  $\text{Gen}_{\max}$  is 1 500.  $P_c = 0.7$  and  $P_m = 0.1$ .

To analyze the time of the GE\_AL algorithm when changing the number of abstract services and the number of candidate services, several kinds of test cases are put up; the abstract services numbers are 10, 15, 20, 25, 30, 35 and 40, respectively. The candidate services numbers are 5, 20, 45, 70 and 100, respectively and each candidate service has four QoS properties and semantic matching. Cost and Time are in

the range of  $[1, 100]$  and  $[1, 200]$ , respectively. Availability, Reliability and sim are in the range of  $[0, 1.0]$ . Aimed at different services composition processes, the objective function  $F(P)$  is applied and the optimal services composition is adopted with maximum fitness value. The GE\_AL algorithm is run for 10 times and the mean values are final results. The experimental results are illustrated in Fig. 2 and Fig. 3.



**Fig. 2** Running of GE\_AL and the objective function is  $F(P)$



**Fig. 3** Optimal solution rate of GE\_AL and the objective function is  $F(P)$

In Fig. 2, the running time varies little when the number of candidate services to a certain services composition process is changed. Moreover, the running time is basically linear when the number of abstract services (from 10 to 40) is changed.

In Fig. 3, when given the number of candidate services and given the values of multi-QoS and semantic matching, aimed at objective function  $F(P)$ , the optimal solution rate exceeds 95%.

## 3 Conclusion

The advantages of web services make people catch the vision of broad applications. This paper proposes a global service matching method in a macroscopical process in which multi-QoS is considered and the QoS model is built. The analysis of the algorithm shows that GE\_AL is convergent. The results of service matching are obtained and maximal global semantic matching is achieved by the genetic algorithm and the QoS properties can meet the users' demands. The experiment of GE\_AL illustrates that service matching results in efficiency and approaches the best resolution.

## References

- [1] Zhao Junfeng, Xie Bing, Zhang Lu, et al. A web services composition method supporting domain feature [J]. *Chinese Journal of Computers*, 2005, **28**(4): 731 – 738. (in Chinese)

- [2] Zeng Liangzhao, Boualem Benatallah, Ngu Anne, et al. QoS-aware middleware for web services composition [J]. *IEEE Transactions on Software Engineering*, 2004, **30**(5): 311 – 327.
- [3] Jang Jae-Ho, Shin Dong-Hoon, Lee Kyong-Ho. Fast quality driven selection of composite web services[C]//*Proceedings of the European Conference on Web Services( ECOWS'06)*. Zurich, Switzerland, 2006: 87 – 98.
- [4] Ardagna Danilo, Pernici Barbara. Global and local QoS constraints guarantee in web service selection[C]//*Proceedings of the IEEE International Conference on Web Services (ICWS'05)*. Orlando, FL, USA, 2005: 806 – 807.
- [5] Rainer Berbner, Michael Spahn, Nicolas Repp, et al. Heuristics for QoS-aware web service composition [C]//*4th IEEE International Conference on Web Services(ICWS'06)*. Chicago, Illinois, USA, 2006: 72 – 82.
- [6] Zhang Chengwen, Su Sen, Chen Junliang. Genetic algorithm on web services selection supporting QoS [J]. *Chinese Journal of Computers*, 2006, **29**(7): 1029 – 1037. (in Chinese)
- [7] Canfora Gerardo, Di Penta Massimiliano, Esposito Raffaele, et al. An approach for QoS-aware service composition based on genetic algorithms[C]//*Genetic and Evolutionary Computation Conference( GECCO'05)*. Washington, DC, USA, 2005: 1069 – 1075.
- [8] Liu Shulei, Liu Yunxiang, Zhang Fan, et al. A dynamic web services selection algorithm with QoS global optimal in web services composition [J]. *Journal of Software*, 2007, **18**(3): 646 – 656. (in Chinese)
- [9] Yang Lei, Dai Yu, Zhang Bin, et al. Dynamic selection of composite web services based on a genetic algorithm optimized new structured neural network[C]//*2005 International Conference on Cyberworlds*. Singapore, 2005: 515 – 522.
- [10] Chen Guoliang. *Genetic algorithm theory and application* [M]. Beijing: Posts and Telecom Press, 1999. (in Chinese)

## 基于语义与 QoS 全局感知的 web 服务组合

张佩云<sup>1,2</sup> 黄 波<sup>1</sup> 孙亚民<sup>1</sup>

(<sup>1</sup> 南京理工大学计算机科学与技术学院, 南京 210094)

(<sup>2</sup> 安徽师范大学数学计算机科学学院, 芜湖 241000)

**摘要:**针对服务组合流程,提出了语义与 QoS 全局感知的服务组合. 在全局语义匹配过程中,既考虑了全局匹配,又考虑了从 QoS 角度进行匹配. 当对服务组合有全局语义满足及 QoS 约束要求时,在全局范围里选择满足整个服务组合流程的 QoS 约束和语义匹配度要求的具体服务集,并实现服务组合的优化解. 建立了全局匹配的 QoS 模型及其评价方法,基于该模型及评价方法,采用遗传算法实现全局语义匹配度最大化及满足用户的 QoS 指标需求. 实验结果和分析表明,基于语义与 QoS 感知的服务匹配算法是可行和有效的.

**关键词:**语义匹配;全局匹配;QoS 模型;遗传算法

**中图分类号:**TP311