http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (6): 1207-1211, 2010
ISSN 1812-5638

On Optimal Decision for QoS-Aware Composite Service Selection

[1]Xiaoqin Fan and [2]Xianwen Fang
[1]School of Computer and Information Technology, Shanxi University, Taiyuan Shanxi, 030006, China
[2]Department of Information and Computer Science,
Anhui University of Science and Technology, Huainan Anhui, 232001, China

Abstract: With the development of web service theories and technologies, it has been an effective approach to satisfy users' requirements through service composition. Service selection is an important part of service composition and its efficiency has direct impact on the quality of composite service. Therefore, in this study, a novel service selection algorithm, named Niche Particle Swarm Optimization (NPSO) algorithm is presented, which integrates Simulated Annealing (SA) and niche technique into Particle Swarm Optimization (PSO) and inherits the rapid local search ability of PSO and global convergence of SA. Experimental results show that NPSO algorithm is not only feasible but also efficient in the process of solving the web service selection problem.

Key words: Web service, service selection, particle swarm optimization, simulated annealing, niche technique

INTRODUCTION

Service-Oriented Computing (SOC) has been a new, promising computing paradigm that centers on the notion of service as the fundamental element for developing software applications and service is a self-describing component that should support a rapid and low-cost composition of distributed application (Brogi *et al*., 2008). Services are offered by service providers and published in the UDDI. User can send a request to UDDI and UDDI retrieves the candidate services which satisfy user's requirements and returns to user. For a specific request, which usually includes some functional descriptions, many candidate services may be returned, which own the same functional attributes but different Quality of Service (QoS), such as cost, availability, reliability and so on. Therefore, it becomes a critical problem how to efficiently select the appropriate service in the process of service composition. Aimed at the problem, a lot of research has been done by the domestic and foreign scholars from different angles. For example, Zeng *et al.* (2004) investigates the service selection using the approaches of global optimization and local optimization respectively after giving the QoS models of component service and composite service and the service selection problem with global optimization is formalized as a integer linear programming problem about single objective. The service selection problem with maximizing an application-specific utility function under the end-to-end QoS constraints is transformed to a multi-index multi-choice 0-1knapsack problem based on combinatorial model and a multi-constraint optimal path problem based on graph model separately (Yu *et al*., 2007). Ardagna and Pernici (2007) formalized the service selection with QoS constraints as a mixed integer linear programming. Aimed at the randomness of web service (Fan *et al*. 2009a, b) presented a Web service selection method based on Markov Decision Process (MDP), which improves the success rate of service composition greatly, but owns high computation cost. Yin *et al*. (2010) formalized the service composition in multi-network environment into the integer programming problem. On the basis of defining QoS model of Web service, the QoS computing methods of composite service are presented oriented to different process model and the optimal service selection is conducted based on integer programming (Huang *et al*., 2009). Aiming at the veracity of QoS information of component services, both the objective factors described by service providers and the subjective information with trustability evaluations from users are considered in the process of service selection (Wang *et al*., 2007). In order to detect and deal with false ratings by dishonest providers and users, a trust and reputation management method is introduced by Le-Hung *et al*. (2005). Considering the problem of uninformed behavior evolution of component service, Mei *et al*. (2008) proposed an adaptive service selection approach based on PageRank (PR) analysis and so on. For service selection method based on QoS can meet the user' global restriction more effectively, most of research are

Corresponding Author: Xiaoqin Fan, School of Computer and Information Technology, Shanxi University, Taiyuan Shanxi, 030006, China

QoS-aware. In fact, searching the optimal composition plan under the users' QoS global restrictions is the combination optimization problem, so it is inevitable for the exhaustive computing method to be poor in scalability and high in computational complexity. Evolutionary algorithm, as a kind of random search algorithm, is more suitable for this kind of combinatorial optimization problem. Fan *et al*. (2010) proposed a kind of web service selection method based on Particle Swarm Optimization (PSO) idea, in which Discrete PSO algorithm oriented to service selection is presented on the basis of defining velocity equation and position evolution equation, but its global convergent ability can be improved. Aimed at the web service composition with independent global constraints, Fang *et al*. (2009) designed a service selection method based on genetic algorithm. In this study, we focus on exploiting Particle Swarm Optimization (PSO) to achieve the better solution for web service selection. PSO is developed by Eberhart and Kennedy (1995). The position of one particle is corresponding to a solution of the solving problem. Liking a bird that flies to the food, one particle moves its position to a better solution according to the best particle's experience and its own experience. Every particle moves iteratively until the end of iterations. We call this process as evolution process. At the end of iterations, the position of best particle is the best solution of the solving problem. Because PSO owns the common limitation of all evolution algorithms, which is prone to fall into the local optimal position, in order to get better solution, Simulated Annealing (SA) and niche technique are integrated with the PSO. The proposed algorithm enhances the particle's searching ability and is suitable to solve the Web service selection problem. The experimental results show that the proposed algorithm is not only feasible but also efficient.

FORMAL PROBLEM DEFINITION

Web service selection is to select one instance service for every abstract task in the composite process model and make the quality of composite service integrated by those instance services is optimal. In detail, for the process model containing n abstract tasks $t_1,t_2,...t_n$ the service selection is to locate n instance services satisfying the constraint conditions and the n instance services constitute the solution of the problem, which is denoted as n-tuple $(ws_1,ws_2,...ws_n)$ where, $ws_i(1\leq i\leq n)$ is the selected instance service of task t_i. Every n-tuple can be corresponding to one position of a particle in the n dimension space. Then service selection is to locate instance services that make the following formula fulfilled.

$$\max \sum_{i=1}^{m} c_i \overline{Q_i(x_1,x_2,\cdots x_n)} \tag{1}$$

$$\text{Constraints} \begin{cases} Q_1(x_1,x_2,\cdots x_n)\leq b_1 \\ \vdots \\ Q_m(x_1,x_2,\cdots x_n)\leq b_m \end{cases} \tag{2}$$

Here, the objective fitness function in Eq. 1 is to max the QoS value of composite service and Eq. 2 gives the constraints which is used to describe the users' personalized requirements. The detail implication can refer to the study (Fan *et al*., 2010).

BACKGROUND: AN INTRODUCTION TO PARTICLE SWARM OPTIMIZATION

The PSO is a novel evolutionary algorithm that was inspired by the motion of a flock of birds searching foods. At the beginning of evolutionary process, a set of particles we called as the swarm must be initiated randomly. Each particle can change its position in the search space just like a flying bird searching the food in the sky. During the evolutionary process, particle's velocity can be dynamically adjusted according to the comprehensive analysis of the flying experiences of itself and the swarm. The i-th particle is denoted as $X_i(x_{i1},x_{i2},...x_{id})$ and $V_i(v_{i1},v_{i2},...v_{id})$ is the present flying velocity of particle i and $P_i(p_{i1},p_{i2},...p_{id})$ is the best place that particle P_i has experienced, that is to say, P_g is the place with optimal fitness value so, far and called as local best position. P_g, called as global best position, is the best position that all particles have experienced. The basic evolution equations are as follows:

$$\begin{aligned} v_{id}(t+1) = \omega v_{id}(t) + c_1 r_{1d}(t)(P_{id}(t) - x_{id}(t)) \\ + c_2 r_{2d}(t)(P_{gd}(t) - x_{id}(t)) \end{aligned} \tag{3}$$

$$x_{id}(t+1) = x_{id}(t) + v_{1d}(t+1) \tag{4}$$

where, ω is the inertia weight which can make particle keep movement inertia and can expand the searching space so as to have ability to explore new position. c_1 and c_2 are accelerating constants and c_1, also named cognitive coefficient, reflects the effect of local best position on particle flying velocity and c_2 also named social learning coefficient reflects the effect of global best position on particle flying velocity, r_1~U(0,1), r_2~U(0,1), are two random variables independent each other. In Eq. 1, the first item shows that particle will accelerates with $\omega v_{id}(t)$, which means it has trust in present movement state and will fly with its inertia. The second item indicates particle think about itself and the last item means that particles will share their information and incorporate with each other. The particle position can be updated by Eq. 3.

The algorithm flow for basic PSO can be summarized as follows:

Step 1: Initialize the population, in which every particle is generated with random position and velocity

Step 2: Update the local best position for every particle in population. That is to say, compare the present position with its best position experienced, if the former is with better fitness, local best position is updated with present position

Step 3: Update the global best position for population. In detail, for every particle in the population, we compare its local best position with global best position of population and if local best position possesses better fitness, the global best position is updated with the particle's local best position

Step 4: Compute the velocity and position for every particle according to Eq. 3 and 4

Step 5: Go back to step 2 until the termination criteria are met

WEB SERVICE SELECTION BASED ON NPSO

Some issues are in applying PSO algorithm to solve Web service selection. The original PSO is developed to solve continuous problem, but service selection is a combinatorial problem and the solution space is discrete. The first issue is find a suitable representation which is not only effective for particle corresponding to the candidate solution of Web service selection problem, can but also guarantee the result is reasonable after fly operator is taken on particles. The detailed description will be discussed firstly. The second issue is how to enhance the global search ability in solving Web service selection problem. Aimed at the problem, Simulated Annealing (SA) is applied and the detailed description is stated. Then the complete algorithm, named niche PSO (NPSO), is shown, which is consisted of SA and PSO.

Particle representation and operation: The searching space is created in a n dimensions space for a process model involving abstract tasks $t_1,t_2,...t_n$ and assumes for each abstract task t_i there are n_i candidate services which can be selected. The position of a particle consists of n dimensions and is represented with $(ws_1, ws_2,...ws_n)$, where, $ws_i(1 \le i \le n)$ is the selected instance service of task t_i. To ensure particles has the ability of flying to all possible position, v_i is defined as a n-tuple and lies between $-n_i$ and n_i. Based on the Eq. 3 and 4, the particle position may be beyond the scope where solutions exist. Therefore, for the d-th dimension $x_d(1 \le d \le n)$ of particle x, the following correcting mechanism is adopted.

$$x_d = \begin{cases} \lceil \mathrm{mod}(x_d / n_d) \rceil + n_d, & x_d < 1 \\ \lceil \mathrm{mod}(x_d / n_d) \rceil, & x_d > n_d \end{cases} \quad (5)$$

Based on above transition, the validity of particle position can be guaranteed always.

Simulated annealing: An algorithm simulating ideas and mechanism in the annealing of solids is named Simulated Annealing (SA). Since, its introduction by (Kirkpatrick *et al*., 1983), the SA algorithm has been successfully applied to many combination optimization problems. The key function of SA is to allow occasional alternations to accept worse solutions in order to increase the probability of jumping away from a local optimum and getting a better solution.

The SA algorithm starts from an initial individual P and initial temperature T and then the individual is perturbed randomly and becomes a new individual P_{new} by applying a suitable operation. Two objective functions E(P) and $E(P_{new})$ are evaluated respectively. For a maximum problem, P_{new} is accepted as a new state if the difference of two objective functions, $\Delta E = E(P_{new})-E(P)$, is bigger than 0. If $E(P_{new})-E(P) \le 0$, the new individual is accepted with probability given by:

$$\min\{1, \exp^{\frac{-\Delta E}{T}}\}$$

where, T is the current temperature and it is decreased iteration by iteration according to a referred cooling rate until to final temperature T_f. Under the temperature T, multi-perturbation can be done. In Web service selection problem, the n-tuple $(ws_1,ws_2,...,ws_n)$ can be looked on as the individual of SA algorithm, in which ws_i is the candidate service of abstract task t_i and the perturbation of individual is corresponding to the probability substitution of candidate service in the niche. For a particle, at the beginning of evolution, only the particle itself is in its niche. With the iteration more and more particles are appended until all particles, the whole swarm, are included in one niche. For one particle, perturbation means that some dimensions of the particle are substituted by those dimensions of another particle which is in the same niche. The computation of niche of a particle can be summarized as follows.

Algorithm 1: The algorithm for computing niche for particle

Input: the population$(x_1,x_2,...,x_m)$, fitness function f, QoS of all candidate services.
Output: the niche N_i for every particle $x_i(1\le i\le m)$ in the population
//x_i and (ws^i_1,ws^i_2) and $1\le i\le m$
//n is the number of abstract task in the process model.
For i=1, m
  For j=1, m AND $j\neq i$
  //Compute the distance between x_i and x_j and denoted as d_{ij}

$$d_{ij}=\sum_{k=1}^{n}|f(ws^i_k)-f(ws^j_k)|/n$$

  End for
$d_i=\min\{d_{ij}/1\le j\le m\}$.
$N_i=\{x_{ij}/d_{ij}=d_i\wedge j\neq i\wedge 1\le j\le m\}$
End for

After computing the niche for every particle in the population, the perturbation can be executed for every individual. In order to make the population own stronger ability of jumping away the local best position, SA is adopted. A complete individual enhancement scheme based on SA is listed in the following.

Algorithm 2: SA algorithm

Input: Initial individual Y_{best}, perturbation times l
Output: Enhanced individual Y_{best}
If $(T>T_f)$
  Compute $f(Y_{best})$.
  For i = 1, l
  Compute the niche for all particles in the population based on Algorithm 1
  Y_{new} is perturbed and generate a new individual Y_{new}.
  Compute $f(Y_{new})$ and let Y_{new}.
  If $\Delta E<0$
    $Y_{best}=Y_{new}$
    $f(Ybest)=f(Y_{new})$

  Else If (R = rand()<min{i1,exp $^{-\Delta E/T}$})
    $Y_{best}=Y_{new}$
    $f(Y_{best})=f(Y_{new})$
  End if
End for
  $T=T\alpha$
End if

Niche PSO algorithm oriented to service selection: In this study, we integrated SA and niche technique into PSO and NPSO oriented to Web service selection is designed. In NPSO, a particle is represented by a n-tuple and every particle moves its position according to Eq. 3 and 4. In the process of evolution, if the particle's position is out of the solution space, correcting mechanism can be adopted to make solution effective. Aimed at the common limitation that the evolutionary algorithms are prone to fall into the local optimum, SA is used to improve the global search ability and in order to enhance the local search ability the individual is disturbed for several times in every generation. The detailed NPSO algorithm is summarized as follows:

Algorithm 3: NPSO algorithm

Input: Process model with n tasks, Candidate services with QoS, Constraint conditions, Fitness functionf
Output: Composite services satisfying the constraints
Initialize the position and velocity for all particles of a population.
Do while termination criteria are not met
  For each particle do
    Compute the niche and execute the perturbation based on algorithm 2.
    Update the local best position.
  End for
  Update the global best position of the population.
  For each particle do
    Move particle to the next position according to Eq. 3 and 4.
    If the particle is out of the solution space
    Correct the particle's position according to Eq. 5.
  End if
 End for
End while

RESULTS

In order to evaluate the service selection algorithm, a great deal of experiments have been performed on a wide set of randomly generated process instances, because of the page constraint, here only part of experiment results are presented. The candidate services corresponding to every abstract task are generated according to uniform distribution with the given mathematical expectation and variance. In this study, MILP is used to compute the optimal solution and the corresponding computation cost and comparisons are made between them on the same experimental condition. The fitness values and computation cost based on NPSO and MILP are recorded in Fig. 1 and 2, respectively.

Figure 1 shows that based on the NPSO algorithm the fitness value increases rapidly and gets to the optimal value gotten based on MILP after the algorithm iterates about 400 times, which means NPSO has better global convergence property. Figure 2 shows, for the NPSO method, with the increase of iteration, its computation

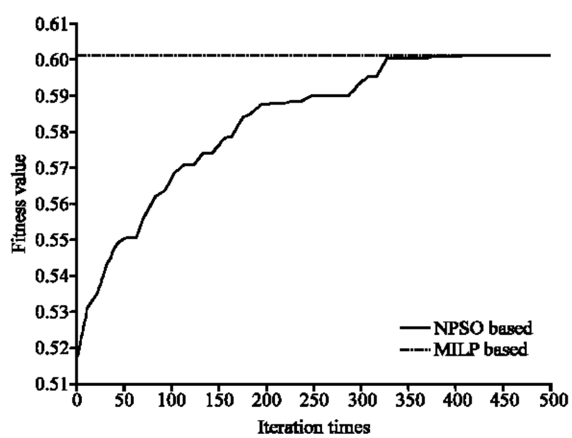

Fig. 1: Comparison about fitness value

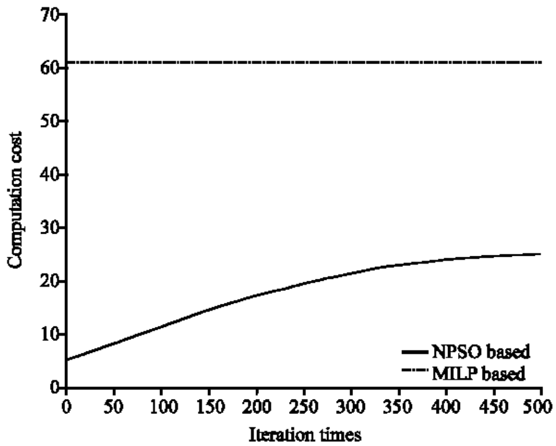

Fig. 2: Comparison about computation cost

cost increases with decreasing speed until getting the optimal value. Figure 1 and 2 show, when the iteration times are 400 the optimal solution based on NPSO has been equal to the global optimal solution, while the computation cost is only 41.6% of MILP. That is to say, NPSO method can solve the Web service selection problem effectively even if the number of candidate services is large.

CONCLUSION

In this study, aimed at the Web service selection problem, the NPSO algorithm integrating SA and niche technique into PSO is designed, which combines the rapid local search ability of PSO and global convergence of SA and experimental results show that NPSO algorithm is both feasible and efficient.

ACKNOWLEDGMENT

We would like to thank the support of the National Major Fundamental Research Program of China under Grant No. 2010CB328100, the National Natural Science Foundation of China under Grant No. 90718012, No. 90818023 and No. 60803034, the National High-Tech Research and Development Plan of China under Grant No. 2007AA01Z136.

REFERENCES

Ardagna, D. and B. Pernici, 2007. Adaptive service composition in flexible processes. IEEE Trans. Software Eng., 33: 369-384.

Brogi, A., C. Sara and P. Razvan, 2008. Semantics-based composition-oriented discovery of Web services. ACM Trans. Inter. Technol., 8: 1-19.

Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, Oct. 4-6, Nagoya, Japan, pp: 39-43.

Fan, X., C. Jiang and X. Fang, 2009a. An efficient approach to web service selection. http://books.google.com.pk/ books?id=9CHsRhx9vLwC&pg=PA271&dq=2009.+An+Efficient+Approach+to+Web+Service+Selection.+LNCS5854&hl=en&ei=bsXfS8GtCMO2rAfp7cnuBg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CDgQ6AEwAA#v=onepage&q&f=false.

Fan, X.Q., C.J. Jiang, J.L. Wang and S.C. Pang, 2009b. Random-QoS-aware reliable web service composition. J. Software, 20: 546-556.

Fan, X.Q., C.J. Jiang, X.W. Fang and Z.J. Ding, 2010. Dynamic web service selection based on discrete particle swarm optimization. J. Comput. Res. Dev., 47: 147-156.

Fang, X.W., C.J. Jiang and X.Q. Fan, 2009. Independent global constraints-aware web service composition optimization. Inform. Technol. J., 8: 181-187.

Huang, A.F., C.W. Lan and S.J. Yang, 2009. An optimal QoS-based web service selection scheme. Inform. Sci., 179: 3309-3322.

Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, 1983. Optimization by simulated annealing. Science, 220: 671-680.

Le-Hung, V., H. Manfred and A. Karl, 2005. QoS-based service selection and ranking with trust and reputation management Proceedings of the 13th International Conference on Cooperative Information Systems, Oct. 31-Nov. 4, Switzerland, pp: 466-483.

Mei, L.J., W.K. Chan and T.H. Tse, 2008. An adaptive service selection approach to service composition. Proceeding of the IEEE International Conference on Web Service, Sept. 23-26, IEEE Computer Society, Washington, DC., USA., pp: 70-77.

Wang, H.C., C.S. Lee and T.H. Ho, 2007. Combining subjective and objective QoS factors for personalized web service selection. Expert Syst. Appl., 32: 571-584.

Yin, K., B. Zhou, S. Zhang, H. Jiang and J. Cristoforo, 2010. Optimizing services composition in multi-network environment. Inform. Technol. J., 9: 399-411.

Yu, T., Y. Zhang and K.J. Lin, 2007. Efficient algorithms for web services selection with end-to end QoS constraints. ACM Trans. Web, Vol. 1. 10.1145/1232722.1232728

Zeng, L.Z., B. Boualem, A.H.H. Ngu, M. Dumas, J. Kalagnanam and H. Chang, 2004. QoS-aware middleware for web services composition. IEEE Trans. Software Eng., 30: 311-327.