1-1-2012

A survey on bio-inspired algorithms for web service composition

Lijuan Wang
University of Wollongong, lw840@uow.edu.au

Jun Shen
University of Wollongong, jshen@uow.edu.au

Jianming Yong
University of Southern Queensland

A survey on bio-inspired algorithms for web service composition

Abstract

Web service composition has become a promis- ing technology in a variety of e-science or e-business areas. There are a variety of models and methods to deal with this issue from dierent aspects. Bio-inspired algorithms are becoming main approaches and solutions. This paper reviews the current researches on web service composition based on bio-inspired algorithms, such as Ant Colony Optimization(ACO), Genetic Algorithm(GA), Evolutionary Al- gorithm(EA) and Particle Swarm Optimization(PSO). By analyzing and investigating dierent approaches, this paper gives an overview about the researches on bio-inspired al- gorithm in web service composition and point out future directions.

Keywords

composition, service, web, survey, algorithms, bio, inspired

Disciplines

Physical Sciences and Mathematics

Publication Details

Wang, L., Shen, J. & Yong, J. (2012). A survey on bio-inspired algorithms for web service composition. Proceeding of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (pp. 569-574). USA: IEEE.

A Survey on Bio-inspired Algorithms for Web Service Composition

Lijuan Wang[1], Jun Shen[1] and Jianming Yong[2]
[1]Faculty of Informatics, University of Wollongong
Wollongong, NSW, Australia
[2]Faculty of Business and Law, University of Southern Queensland
Toowoomba, QLD, Australia
E-mail: lw840@uowmail.edu.au, jshen@uow.edu.au, yongj@usq.edu.au

Abstract—**Web service composition has become a promising technology in a variety of e-science or e-business areas. There are a variety of models and methods to deal with this issue from different aspects. Bio-inspired algorithms are becoming main approaches and solutions. This paper reviews the current researches on web service composition based on bio-inspired algorithms, such as Ant Colony Optimization(ACO), Genetic Algorithm(GA), Evolutionary Algorithm(EA) and Particle Swarm Optimization(PSO). By analyzing and investigating different approaches, this paper gives an overview about the researches on bio-inspired algorithm in web service composition and point out future directions.**

Index Terms—**Bio-inspired algorithms, web service composition, quality of service.**

I. Introduction

Web service composition is to compose the existing web services into a new web service to accomplish tasks. If no single service can satisfy users' functionalities, it is necessary to combine existing services into a new one in order to fulfill the request. During the process of composite service, there are many candidate services with the same functionality but different quality of service (QoS) attributes. How to choose the right services to compose in order to get a optimal solution for the task is a challenging problem. So far there are many methods and models for web service composition. Meanwhile, we need autonomous, scalable, adaptable, robust, and reliable service oriented systems. It is recognized that biological systems require the similar attributes. Hence, some researches are emerging to use biologically inspired algorithms to solve web service composition problem. We will review a variety of bio-inspired algorithms in web service composition from different aspects in the following sections first. Then we conclude with future work.

II. Models of web service composition

As web services are in a dynamically changing computing environment, service oriented systems use late-binding mechanism to address the high dynamicity challenge. The key principle of late-binding make services easily be selected and bound on demand at run-time. An abstract service is used to refer a specific feature in a service orchestration, several concrete services are available to realize such a feature. All concrete services mapping to an abstract service may have the same functional but different non-functional properties. So most methods regard the composition problem as to find a combination of concrete services based on QoS criteria. There are two kinds of QoS driven service selection approaches for composite service execution: one is based on local constraints and the other on global constraints. The local optimization assures the performance of each individual task, e.g., the limit of the cost of one operation. The global one considers QoS constraints and preferences as a whole, e.g., when the whole response time is constrained. Also there are different workflow structures for web service composition, such as sequence, flow(concurrence), switch(choice) and while(loop). We use aggregation functions for each QoS parameter to compute QoS attributes for the overall composition. And the aggregation functions are different for different workflow structures and different quality metrics. In addition, there are two kinds of calculation methods based on QoS attributes[3], namely exhaustive method and approximate algorithm. The exhaustive method calculates all of the candidate paths to choose the best plan, but this lead to poor scalability and heavy calculation cost. The approximate algorithm chooses an ideal composition plan which is infinitely close to the best one. The heuristics methods are approximate solutions to calculate QoS attributes. Anyhow, finding a service plan with global constraints is known as a NP-hard problem[10]. So there are a number of models for web service composition. In [35], the web service composition was regarded as a knapsack problem. In [16], it was considered as a multiple-choice knapsack problem. It could also be modeled as a graph based on graph theory. Then the problem is transformed into finding the order of nodes in the graph to generate an executable workflow to satisfy user's requests. In the service composition graph, services represent the nodes, and edges represent two services are binding together. Because of the dependency constraint and the conflict constraint among services[27], the optimal solution of service composition is also regarded as a constrained combinatorial optimization issue. Meanwhile, since a number of non-functional properties are inherent to web services and there are different executable workflows in general, some methods for QoS-aware web service com-

978-1-4673-1210-3/12/$31.00 ©2012 IEEE

position adopt multi-dimensional and/or multi-objective QoS model to represent constraints when composing web services.

III. QoS attributes and models in service composition

In the presence of multiple web services provided by different organizations with functional(e.g., ticket purchase, payment) and non-functional properties(e.g., response time, reliability)[20]. The first step of web service composition is to select the component web services. When more than one candidate service providing equivalent functionalities to satisfy the user's request, a choice needs to be made to select appropriate web services to participate in a given composite service. Quality of Service(QoS) for web services refers to various non-functional attributes used for discriminating web services with identical functionality. However, there is no standard to include all or any subset of the non-functional properties in developing web services. The international quality standard ISO 8402 describes quality as the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs[22]. A set of 13 quality attributes have been identified as important factors in the design of SOAs[28]: interoperability, reliability, availability, usability, security, performance, scalability, extensibility, adaptability, testability, auditability, operability, deployability, and modifiability[23]. Usually, four different QoS attributes are considered in most researches: response time, cost, reliability and availability. The reason for choosing these four is that they represent a selection of relevant characteristics in the field of web services[21]. Among different QoS properties of web services, some properties are user independent and have identical values for different users such as price and popularity, which are usually offered by service providers or third-party registries. Some QoS properties are user dependent and have different values for different users such as response time, invocation failure rate[47], which are usually specified by requester. Typically, one may choose the cheapest service while another may choose the fastest or maybe a compromise between the two. Hence, the end users always specify constraints on the values of some attributes, such as the upper values of the price. And some service like temperature service could have domain-specific QoS attributes, such as precision or refresh frequency. It should be noted that the service providers can estimate the values of QoS attributes which always specified in the Service Level Agreement (SLA). Due to the fact that services are usually distributed across the Internet, some QoS attributes such as availability and successful execution rate are affected by communication link. So QoS of web services is multi-dimensional and we should adopt different methods to measure these QoS properties. Meanwhile, QoS attributes can be classified into measurable and unmeasurable properties. A QoS metric is used to measure QoS attribute[22]. It has specific value types which are associated to a specific unit. For example, we can use average execution time metric to measure the execution time QoS attribute. Unmeasurable QoS attributes represent static information which is qualitative in nature[22]. We can use a partial ordering function to represent the values they take. Such as the robustness/flexibility attribute have the value type {inflexible, flexible, very flexible} which can be mapped to the set {0, 1, 2}. The authors in [26] gave the definition and measurement of five generic quality criteria for elementary services: execution price, execution duration, reputation, successful execution rate and availability. Anyhow, the overall QoS of a composite service is determined by the values of QoS attributes of concrete services and the composition workflow structure, which can be computed by applying the aggregation functions as described in Table 1. All the functions in the table are recursively. Note the table is not complete, it just contains rules mostly used in researches and could be extensible based on the application. For a Sequence workflow of tasks($t_1, \ldots, t_M$), the *Time* and *Cost* functions are additive while *Availability* and *Reliability* functions are multiplicative. For a Switch workflow of tasks($t_1, \ldots, t_N$) with probability($p_1, \ldots, p_N$,$\sum_{i=1}^{N} p_i = 1$) is always evaluated as a sum of the attribute value of each task. Loop construct with k iterations of task t is equivalent to a Sequence construct of k copies of t[10].

All the QoS attributes of concrete services can be classified into two groups: the first group of QoS attributes whose values should be minimized, i.e., the higher the value, the lower the quality, such as response time and cost; and the second group of QoS attributes whose values should be maximized, such as availability and reliability. For the convenience of description, some name the two groups as minimization attributes and maximization attributes respectively, and some name them as cost type and utility type respectively[34]. Some use utility function to evaluate the multi-dimensional quality of a web service. Due to the variety of QoS are measured in different units and non-commensurable, web services' QoS attributes need to be normalized before to be combined together. Most of the researches about web service composition adopted the Simple Additive Weighting (SAW) approach in Multiple Criteria Decision Making (MCDM)[15] technique to select an optimal web service. The SAW approach includes two phases: scaling phase and weighting phase. The following formulas are used to compute the overall score for each web service:

$$score(s_i) = \begin{cases} \sum_{j=1}^{r} (\frac{Q_j^{max} - Q_{ij}}{Q_j^{max} - Q_j^{min}}) * W_j & \text{if } Q_j^{max} - Q_j^{min} \neq 0 \\ \sum_{j=1}^{r} (\frac{Q_{ij} - Q_j^{min}}{Q_j^{max} - Q_j^{min}}) * W_j & \text{if } Q_j^{max} - Q_j^{min} \neq 0 \\ \sum_{j=1}^{r} 1 * W_j & \text{if } Q_j^{max} - Q_j^{min} = 0 \end{cases} \quad (1)$$

In the above equations, it is supposed each service has r quality criteria, a matrix Q is built, in which each row Q_j corresponds to a web service s_i while each column corresponds to a quality dimension. And Q_j^{max} and Q_j^{min} are the maximal value and minimal value of a quality criterion in matrix Q respectively. $W_j \in [0,1]$ represents the weight

TABLE I
QoS Aggregation functions for different workflow structures

QoS Attributes	Sequence	Switch	Flow	Loop
Time (T)	$\sum_{i=1}^{M} T(t_i)$	$\sum_{i=1}^{N} p_i * T(t_i)$	$Max\{T(t_i)_{i\in\{1,\dots,P\}}\}$	$k * T(t)$
Cost (C)	$\sum_{i=1}^{M} C(t_i)$	$\sum_{i=1}^{N} p_i * C(t_i)$	$\sum_{i=1}^{P} C(t_i)$	$k * C(t)$
Availability (A)	$\prod_{i=1}^{M} A(t_i)$	$\sum_{i=1}^{N} p_i * A(t_i)$	$\prod_{i=1}^{P} A(t_i)$	$A(t)^k$
Reliability (R)	$\prod_{i=1}^{M} R(t_i)$	$\sum_{i=1}^{N} p_i * R(t_i)$	$\prod_{i=1}^{P} R(t_i)$	$R(t)^k$
Custom Attribute (F)	$f_{se}(F(t_i)_{i\in\{1,\dots,M\}})$	$f_{sw}((p_i, F(t_i))_{i\in\{1,\dots,N\}})$	$f_{fl}(F(t_i)_{i\in\{1,\dots,P\}})$	$f_{lo}(k, F(t))$

of criterion j which values provided by the users based on their own preferences. The authors in [34] considered this approach just selected services based on QoS and it did not include the users' requirements, so the authors gave a tuple to represent the users' specification of their QoS requirement. The tuple included the lower bound and upper bound of users' acceptable QoS, and the weighting factor and also the user's confidence in the former three values. Then the authors normalized QoS based on the four parameters in the tuple.

How to select the appropriate services that meet some Quality of Service(QoS) requirements becomes a research issue. And different users may have different requirements and preferences regarding QoS. For example, a user may give more importance to the cost than to the execution duration, while another user may prefer lower execution duration while satisfying certain cost. Hence, we need QoS-aware approach to service composition by taking into account users' constraints and preferences.

IV. Different bio-inspired algorithms for web service composition

This section presents main bio-inspired algorithms for web service composition in recent years. The approaches used to solve web service composition include Ant Colony Algorithm, Genetic Algorithm, Evolutionary Algorithm, and Particle Swarm Optimization. We will discuss these methods based on some recent papers.

A. Ant Colony Algorithm

Ant colony algorithm studies models derived from the behaviors of some real ants, it is widely used for combinatorial optimization problem. The two features of ant colony algorithm are global positive feedback and local heuristic. When solving web service composition based on ant colony algorithm, the problem is modeled as a weighted directed acyclic graph with a start point S and a target point T. Then we can solve this issue by setting the start point S as the ants nest and the target point T as the food source, and the QoS constraints as the weights of the edges. Thus, the problem is transformed from selecting optimal schema for composite web service into selection of the optimal path in the weighted directed acyclic graph. A variety of researches adopted Ant Colony Optimization (ACO) algorithm to solve web service composition problem. The differences among them include the initial population, the update and evaporation strategy of pheromone, the heuristic information, and how to set the pheromone to denote information. With regard to update of the pheromone, there were many methods such as combining local strategy and global strategy, combining local strategy and offline strategy, using chaos operator, and adjusting pheromone dynamically according to the changes of QoS attributes. All these researches tried to improve the speed and computational efficiency, and to avoid the algorithm to trap in local optimization. The authors in [42] pointed out that the selection of the parameters in ant colony algorithm had great effect on the efficiency, so the authors used genetic algorithm to set the key parameters, then presented a dynamic web service composition algorithm based on the combination of ant colony algorithm and genetic algorithm. The experimental results showed that the combined algorithm could accelerate the speed of service selection. The paper [4] presented a technique for semantic web service composition which combined a graph model with the ant colony optimization meta-heuristic. The authors in [36] used chaos operator to make up the shortage of ant colony algorithm. The objective of the new chaos ant colony algorithm was to improve the efficiency of web service composition. To apply random, ergodic chaos operator could avoid ant colony algorithm to trap into local optimum. Another paper [29] analyzed the problem of service selection based on the principle of ant colony algorithm and gave the performance of ant colony algorithm under different parameters. The differences among those researches include initial population, the update and evaporation strategy of pheromone, the heuristic information, and how to set the pheromone to denote information. Meanwhile, when the problem of web service composition is modeled as a graph, different methods use different representation. In paper [37], the node in the graph denoted web service and the edge denoted the interaction between services. The authors in [39] used a node represented a service orchestration, an edge represented a service, and each path in the graph represented a complete services composition. Also, the authors pointed out that one kind of pheromone in ant colony algorithm could not deal with the multiple QoS attributes, so the paper presented a dynamic ant colony algorithm optimization method based on multi pheromones.

B. Genetic Algorithm

Some researches adopt linear programming, dynamic programming, or integer programming method to solve web service composition problem. In [25], the authors adopted linear integer programming(IP) to solve QoS-aware composition problem based on the global planning approach. Since the total number of execution plans will be exponential when the number of tasks and candidate web services are very large, the integer programming method for selecting an optimal execution plan can avoid generating all possible execution plans. But the objective function and the constraints as inputs in IP method must be linear. One of the weakness of IP is that it uses linear aggregation functions for different QoS attributes. If we adopt non-linear integer programming then scalability becomes a problem. Genetic algorithm(GA) is unconstrained procedures to solve combinatorial optimizing problems[32]. In [11], the author proposed a GA-based approach for QoS-aware web service composition. By comparing the GA approach with linear integer programming method, the authors pointed out that integer programming is preferable when the workflow size is limited. The paper [21] implemented a genetic algorithm for QoS-based selection of services. The authors investigated the impact of different parameters, such as mutation rate and fitness function on the optimization capability of the genetic algorithm. By comparing the genetic algorithm with a branch-and-bound approach and the exhaustive search, the results showed that the setup values along with the given optimization goals and constraints prevented a genetic algorithm from efficiently identifying very near optimal solutions. Moreover, different parameters have different influences on the performance of genetic algorithm[30]. The paper [31] investigated how fitness function and a mutation policy based on relation matrix coding scheme effected the performance of genetic algorithm for QoS-aware web services selection. Their experimental results showed that the new fitness function and mutation policy improved the convergence speed of genetic algorithm. The paper [27] proposed a hybrid genetic algorithm utilizing a local optimizer to improve the fitness value of the individuals in the population which improved the overall QoS value. One of the differences among the existing genetic algorithm for web service composition is the encoding method. Some adopted one dimensional chromosome encoding method[11], [43]. But when the number of candidate services were very large, the readability of the chromosomes was very weak. On the other hand, this method could not represent the semantic information. So some presented genetic algorithm based on relation matrix coding scheme[44]. However, this method frequently generated illegal individuals and the efficiency was lower. The paper [5] proposed a tree-encoding model which could express various composition relationships and carry static-model of service workflow and supported re-planning at run-time.

The genetic algorithm has the problem of prematurity which obstructs the algorithm from further improving. Two immune algorithm were proposed in the paper [12] and [17] to solve the premature convergence of genetic algorithms, and both of them were optimization methods for service composition with global QoS constraints. The difference between them are the encoding method and the mutation process. Tabu search as a heuristic procedure for solving optimization problem can escape the trap of local optimality. A hybrid GA-Tabu algorithm was proposed in the paper [33]. The Tabu search was used to select the chromosome with best fitness after the crossover and mutation operation by a genetic algorithm. The approach increased diversity of population and more search space, and also escaped the trap of local optimality. The paper [18] presented metaheuristic based algorithms using Tabu search and hybrid genetic algorithm. By comparing their performance with iterative steepest descent and a basic genetic algorithm, the experimental results showed the performance was related with the problem instances and run times. Another paper [13] chose simulated annealing algorithm instead of Tabu search to prevent the searching process from trapping into local optima. The paper [41] pointed out that workflow construction and QoS were two important criteria used to analyze the characteristic of each approach of web service composition. In a word, some methods are QoS-based service composition using genetic algorithm[25], [11], [21], [18], [13], [46], [40], [41] and some are workflow based[9].

Genetic Algorithm is a powerful tool to solve combinatorial optimizing problems[32]. No wonder a variety of researches adopted GA to solve QoS-based web service composition problem. These GA-based approaches for web service composition focus on different aspects, such as encoding scheme, fitness function with both static or dynamic penalty factor, operators(selection, mutation and crossover), population diversity handling, and QoS model. Meanwhile, some researches focus on other aspects, such as the speed of service selection, the convergence and speed of the algorithm, run-time fault-tolerance, flexibility, environmental dynamic adaptivity, and efficiency in the large solution space. The involved encoding schemes include integer array coding scheme, binary string coding scheme, one dimension coding scheme and relation matrix coding scheme. The involved selection operators include roulette wheel selection, rank-based and elitism selection, and binary tournament mechanism. And the involved crossover operators include two-points crossover and single point crossover. Usually the fitness function for a genome g looks as follows:

$$F(g) = \frac{W_1 * Cost(g) + W_2 * ResponseTime(g)}{W_3 * Availability(g) + W_4 * Reliability(g)} + W_5 * pf \tag{2}$$

QoS attribute factors(i.e., Cost(g), ResponseTime(g) etc.) are normalized in the interval $[0, 1]$. $W_1, \ldots, W_5$ are real, positive weighting factors. In particular, $W_1, \ldots, W_4$ indicate the importance of a QoS attribute that a service user gives, while W_5 weights the penalty factor. And pf indicates the penalty factor for individuals that violate constraints, maybe someone adopts static penalty while others adopt a dynamic penalty. All the QoS attributes that need to minimize should be as numerator and others need to maximize should be as denominator in the above formula and vice versa. It should be noted that the above formula could be extensible depend on particular application cases. Some of the approaches study the influence of the design of GA on its behavior and performance when it is used in QoS-based web service composition, while others focus on how to set parameters to improve the performance by adopting different policy. Usually, most of the GA-based approaches for web service composition consider the problem such as the convergence speed, the search capability, the computing time and scalability. However, there is no standard model or definition of the key parameters of GA for QoS-based web service composition.

C. Evolutionary Algorithm

Some researches regarded the web service composition as a multi-objective optimization problem, then adopted different methods to deal with this issue, such as evolutionary algorithms[1], [2], particle swarm optimization[19], [6], [24], [8], [14], [38]. The paper [45] presented an adaptive framework using cultural algorithm for QoS-aware web service composition. By comparing the performance of this method with genetic algorithm, the experimental results showed the former one was slightly surpassed than that of the latter one. Other evolutionary algorithms in web service composition[2], [1], [7] also described the problem as a multi-objectives multi-constraints requests, and investigated evolutionary algorithms to deal with the optimization problem.

D. Particle Swarm Optimization

Particle swarm optimization (PSO) is one of the evolutionary computational techniques. When to use the general PSO algorithm to solve the web service composition, it has slow convergence speed and strict speed demand[24]. Normally web service composition is environmental dynamical process. So the paper [24] proposed an environment-aware particle swarm algorithm. The paper [19] presented a hybrid algorithm for web service composition, which used genetic algorithm to search the problem space, then used particle swarm optimization to enhance local search ability. In [38], the authors used chaos algorithm to overcome the low efficiency and partial optimization of PSO. The results showed the new chaos particle swarm optimization algorithm could find more optimal solutions than PSO and GA algorithm.

V. Conclusion

Web service composition is a key issue in service oriented architecture. This paper focus on web service composition based on bio-inspired algorithms. By reviewing different models and approaches, we conclude that there are still many problems to deal with in the future. The ant colony algorithm focused on problems in the aspect of theory such as initialization parameter and information update. On the other hand, the efficiency of ant colony algorithm is very low. The genetic algorithm, particle swarm algorithm and Tabu search have the problem of premature convergence, so they will be easily trapped in the local optimality. In genetic algorithm, the mutation and crossover operator are relatively fixed and randomly search without guide will cause degeneration. So many efforts are made to improve the performance through combining them, such as hybrid genetic algorithm, chaos particle swarm optimization algorithm, chaos ant colony algorithm, a combination of genetic algorithm and ant colony algorithm, a combination of genetic algorithm and Tabu search, a combination of genetic algorithm and cultural algorithm, a combination of genetic algorithm and particle swarm optimization and so on. All the methods expect better speed, convergence, adaptivity and high efficiency in large search space. Although some researches compared ACO with GA, or compared ACO and GA with other heuristic algorithms, they just used their own simulated environment and test dataset. To the best of our knowledge, there is no benchmark tool or framework for web service composition based on bio-inspired algorithms. Different approaches have different merits, but there is no standard or qualitative comparison among these algorithms. How to make the bio-inspired algorithm solve web service composition problem with higher performance, how to set parameters of the bio-inspired algorithm, which algorithm is the fitted one to solve a specific problem, remain unknown and call for future research. As all of these are open issues until now, we need further researches on bio-inspired algorithms in web service composition.

References

[1] Arion de Campos Junior, Aurora T.R.Pozo, Silvia R.Vergilio, Taylor Savegnago, *Many-objective evolutionary algorithms in the composition of web services*, *2010 Eleventh Brazilian Symposium on Neural Networks*, pp. 152-157, 2010.

[2] Brahim Batouche, Yannick Naudet, Frederic Guinand, *Semantic web services composition optimized by multi-objective evolutionary*, *2010 Fifth International Conference on Internet and Web Applications and Services*, pp. 180-185, 2010.

[3] Chengwen Zhang, *Adaptive genetic algorithm for QoS-aware service selection*, *IEEE Workshops of International Conference on Adavanced Information Networking and Applicaitons*, pp. 273-278, 2011.

[4] Cristina Bianca Pop, et al.,*Ant-inspired Technique for Automatic Web Service Composition and Selection*, *12th International symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pp.449-455, 2010.

[5] Chunming Gao, Meiling Cai, and Huowang Chen, *QoS-aware service composition based on tree-coded genetic algorithm*, *31st Annual International Computer Software and Applications Conference*, pp. 361-367, 2007.

[6] Cao Jiuxin, Sun Xuesheng, Zheng Xiao, Liu Bo, Mao Bo, *Efficient multi-objective services selection algorithm based on particle swarm optimization*, *2010 IEEE Asia-Pacific Services Computing Conference*, pp. 603-608, 2010.

[7] Chun-Che Huang, Wen-Yau Liang, Wei-Che Lee and Yung-Huan Lai, *Constrained evolutionary computing approach to web service compositions*, *International Journal of Systems Science*, Vol. 42, No. 10, pp. 1625-1638, 2011.

[8] Fei Li, Yaning Huang, *An web service selection optimization method based on particle swarm optimization*, *2010 International Conference On Computer Design And Applications*, pp. 477-481, 2010.

[9] Florin-Claudiu Pop, Marcel Cremene, Mircea-Florin Vaida, *Workflow based service composition using genetic algorithm*,

ACTA technica napocensis, Electronics and Telecommunications, Vol. 51, No. 2, pp. 26-31, 2010.

[10] G. Canfora, M. Di Penta and R. Esposito, etc., *A lightweight approach for QoSaware service composition*, *Proc. 2nd International Conference on Service Oriented Computing (ICSOC'04)*, New York, USA, pp. 36-47, 2004.

[11] Gerardo Canfora, MassimiliaNo Di Penta, Raffaele Espositio, Maria Luisa Villani, *An Approach for QoS-aware Service Composition based on Genetic Algorithms*, *GECCO '05 Proceedings of the 2005 conference on Genetic and eVolutionary computation*, pp. 1069-1075, 2005.

[12] Gao Yan, et al., *Immune Algorithm for Selecting Optimum Services in Web Service Composition*, in *Wuhan University Journal of Natural Sciences*, Vol. 11, No. 1, pp. 221-225, 2006.

[13] Gao Zhipeng, Chen Jian, Qiu Xuesong, Meng Luoming, *QoE/QoS driven simulated annealing-based genetic algorithm for web services selection*, *The Journal of China Universities of Posts and Telecommunications*, Vol. 16, Issue. 08, pp. 102-107, 2009.

[14] Hong Xia, Zengzhi Li, *Particle swarm algorithm for the quality of service-oriented web services selection*, *2009 Second International Symposium on Knowledge Acquisition and Modeling*, pp. 303-306, 2009.

[15] H.C-Land, K.Yoon, *Multiple Criteria Decision Making*, *lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, 1981.

[16] Juhnyoung Lee, *Matching Algorithms for Composing Business Process Solutions with Web Services*, *Proceedings of the 4th International Conference on ECommerce and Web Technologies (ECWEB03)*, pp. 393-402, Prague, Czechoslovakia, October 2003. Springer Press.

[17] Jiuyun Xu, Stephan Reiff Marganiec, *Towards Heuristic Web Services Composition Using Immune Algorithm*, *IEEE International Conference on Web Services*, Beijing, China, pp. 238-245, 2008.

[18] Jose Antonio Parejo, Pablo Fernandez, Antonio Ruiz Cortes, *QoS-aware services composition using Tabu search and hybrid genetic algorithms*, *Actas de Talleres de Ingeniera del Software y Bases de Datos* Vol. 2, No. 1, pp. 55-66, 2008.

[19] Jian Liu, Jun'e Li, Kaipei Liu, Wen Wei, *A hybrid genetic and particle swarm algorithm for service composition*, *Sixth International Conference on Advanced Language Processing and Web Information Technology*, pp. 564-567, 2007.

[20] Joyce EI Haddad, Maude ManNouvrier, and Marta Rukoz, *TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition*, *IEEE Transactions on Services Computing*, Vol.3, No.1, pp.73-85, 2010.

[21] Jaeger, Michael C., Muehl, Gero, *QoS-based Selection of Services: The Implementation of a Genetic Algorithm*, *Communication in Distributed Systems (KiVS)*, ITG-GI Conference , pp. 1-12, Feb. 26, 2007.

[22] Kyriakos Kritikos and Dimitris Plexousakis, *Requirements for QoS-Based Web Services Description and Discovery*, *IEEE Transactions on Services Computing*, Vol.2, No.4, pp.320-337, 2009.

[23] Lianzhang Zhu and Xiaoqing Liu, *Technical Target Setting in QFD for Web Service Systems Using an Artificial Neural Network*, *IEEE Transactions on Services Computing*, Vol.3, No.4, pp.338-352, 2010.

[24] Long Jun, Gui Weihua, *An environment-aware particle swarm optimization algorithm for services composition*, *International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2009.

[25] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang,*QoS-aware middleware for web services composition*, *IEEE Transactions on Software Engineering*, 30(5), May 2004.

[26] Liangzhao Zeng, Benatallah, B, Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H., *QoS-aware middleware for Web services composition*, *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311- 327, May 2004.

[27] Maolin Tang, Lifeng Ai, *A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition*, *Proceeding of the 2010 World Congress on Computational Intelligence*, pp. 268-275, 2010.

[28] M. Yang, Y. Li, S. Li, and P. Li., *ANN-Based Fuzzy Reasoning to Determine the Importance of Technical Requirements in QFD*, *Proc. Fourth Intl Conf. Wireless Comm.*, Networking and Mobile Computing (WiCOM 08), pp. 1-5, 2008.

[29] Rongxi Wang, Li Ma, and Yanping Chen, *The Application of Ant Colony Algorithm in Web Service Selection*, *International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2010

[30] Rojas I, Gonzalez J, Pomares H, Merelo J J, Castillo P A, Romero G, *Statistical analysis of the main parameters involved in the design of a genetic algorithm*, *IEEE Transactions on Systems, Man, and CyberneticsPart. C: Applications and Reviews*, Vol. 32(1), pp. 31-37, 2002.

[31] Sen Su, Chengwen Zhang, and Junliang Chen, *An Improved Genetic Algorithm for Web Services Selection*, *International Federation for Information Processing*, pp.284-295, 2007.

[32] Srinivas M, Patnaik L.M, *Genetic algorithms: a survey*, *IEEE computer*, Vol. 27, pp. 17-26, 1994.

[33] Sondos Bahadori, Somayeh Kafi, Kamran Zamani far, Mohammad Reza Khayyambashi, *Optimal web service composition using hybrid GA-TABU search*, *Journal of Theoretical and Applied Information Technology*, Vol. 9, No. 1, pp. 10-15, 2009.

[34] Stephen S. Yan and Yin Yin, *QoS-Based Service Ranking and Selection for Service-Based Systems*, *IEEE International Conference on Service Computing*, pp. 56-63, 2011.

[35] Tao Yu, Yue Zhang, and Kwei-Jay Lin, *Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints*, *ACM Transactions on the Web*, Vol.1, No.1, Article 6, May 2007.

[36] Wang Yunwu, *Application of Chaos Ant Colony Algorithm in Web Service composition based on QoS*,*International Forum on Information Technology and Applications*, pp. 225-227, 2009.

[37] Wang li, and He Yanxiang, *A web service composition algorithm based on global QoS optimizing with MOCACO*, *Algorithms and architectures for parallel processing*, Vol. 6082, pp. 218-224, 2010.

[38] Wang Li, He Yanxiang, *Web service composition based on QoS with chaos particle swarm optimizaiton*, *2010 6th International Conference on Wireless Communications Networking and Mobile Computing*, pp. 1-4, 2010.

[39] Yamei Xia, Junliang Chen, and Xiangwu Meng, *On the dynamic ant colony algorithm optimization based on multi-pheromones*, *IEEE/ACIS International Conference on Computer and Information Science*, pp. 630-635, 2008.

[40] Yue Ma, Chengwen Zhang, *Quick convergence of genetic algorithm for QoS-driven web service selection*, *The International Journal of Computer and Telecommunications*, Vol. 52, Issue. 5, pp. 1093-1104.

[41] Yong-Yi Fanjiang, Yang Syu, Chun-Hung Wu, Jong-Yih Kuo, and Shang-Pin Ma, *Genetic algorithm for QoS-aware dynamic web services composition*, *International Conference on Machine Learning and Cybernetics*, Vol. 6, pp. 3246-3251, 2010.

[42] Zongkai Yang, Chaowang Shang, Qingtang Liu, and Chengling Zhao, *A Dynamic Web Services Composition Algorithm Based on the Combination of Ant Colony Algorithm and Genetic Algorithm*, *Journal of Computational Information Systems*, Vol.6, No.8, pp.2617-2622, 2010.

[43] Zhang Liangjie, Li Bing, *Requirements driven dynamic services composition for web services and grid solutions*, *Journal of Grid Computing*, Vol 2, pp. 121-140, 2004.

[44] Zhang Chengwen, Su Sen, Chen Junliang, *Genetic algorithm on web services selection supporting QoS*, *Chinese Journal of Computers*, Vol. 29. No. 7, pp. 1029-1037, 2006.

[45] Ziad Kobti, and Wang Zhiyang, *An adaptive approach for QoS-aware web service composition using cultural algorithms*, *Advances in artificial intelligence*, Vol. 4830, pp. 140-149, 2007.

[46] Zhang C, Su S, Chen J, *A Novel Genetic Algorithm For Qos-Aware Web Services Selection*, *IEEE CEC06 and EEE06*, USA. LNCS, Vol. 4055, pp. 224-235, Springer-Verlag, Berlin, 2006.

[47] Zibin Zheng, Hao Ma, Michael R.Lyu, and Irwin King, *QoS-Aware Web Service Recommendation by Collaborative Filtering*, *IEEE Transactions on Services Computing*, Vol.4, No.2 pp.140-152, 2011.