International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:4, 2015

Optimized Weight Vector for QoS Aware Web Service Selection Algorithm Using Particle Swarm Optimization

N. Arulanand, P. M. Ananth

***Abstract*—**Quality of Service (QoS) attributes as part of the service description is an important factor for service attribute. It is not easy to exactly quantify the weight of each QoS conditions since human judgments based on their preference causes vagueness. As web services selection requires optimization, evolutionary computing based on heuristics to select an optimal solution is adopted. In this work, the evolutionary computing technique Particle Swarm Optimization (PSO) is used for selecting a suitable web services based on the user's weightage of each QoS values by optimizing the QoS weight vector and thereby finding the best weight vectors for best services that is being selected. Finally the results are compared and analyzed using static inertia weight and deterministic inertia weight of PSO.

***Keywords*—**QoS, Optimization, Particle Swarm Optimization (PSO), weight vector, web services, web service selection

I. Introduction

WEB service framework brings a new revolution in traditional computing. Web services are considered as self-contained, self-describing, modular applications that can be published. It has an interface described in a machine readable format (WSDL). Other systems interact with the Web service in a way prescribed by its description using SOAP messages, usually conveyed using HTTP with an XML serialization in conjunction with other web related standards. Web Services are self-independent application that shows modular and also distributed concepts. Web service description is provided in the WSDL document and it can be accessed from the internet using SOAP protocol.

Service Oriented Architecture (SOA) is essentially a collection of web services that interact with each other on the network. By means of service oriented architecture (SOA) [1] based on web service technologies, enterprises can now address platform interoperability problems and therefore grasp ever changing business opportunities and challenges.

QoS (Quality of Service) is a key indicator for web service non-functional quality criteria, which can be used to distinguish web services with the same function. As the number of web services that offer similar functionality increases, QoS properties become a crucial issue during the selection and ranking of accurate web services. It is not hard to visualize that service requestors will face large number of choice of services that offer similar functionality.

Arulanand N. is with the PSG College of Technology, Coimbatore, Tamil Nadu, India (e-mail: arulnat@yahoo.com).

In general, the selection of a web service from a collection of service alternatives on the basis of two or more QoS factors is a multiple conditions decision making (MCDM) problem [2]. However, when building the system evaluation model, the traditional multiple conditions programming is not flexible enough because of the following two reasons.

a) Under many circumstances, qualitative QoS criteria values are often imprecisely defined or acquire as the quality of web services may have deviations due to run-time functional behaviors, hardware resource configuration of web services and the network connection status.
b) It is also not easy to exactly compute the weight of each QoS criterion since human conclusions are often vague.

The efficient selection method introduced in [3] provides the combined evaluation of QoS to select the best service from the list of candidate web services for selection. In the recent years, optimization has become a vital dynamic area of exploration as it is used to solve real world complex NP-hard problems. Since optimization algorithms possess diversification characteristic they are more powerful in solving complex problems than the standard methods [4]. The goal of optimization can either be to minimize the given objective function or to maximize the objective function and it is a method of experimenting on any theory that continuously tries to tune the input parameters in order to find the maximum or minimum output.

Particle Swarm Optimization (PSO) [5] is a computational technique that optimizes a problem by iteratively trying to progress a particle with regard to a given quality. It can be used on optimization tribulations where calculations are complex. PSO is a method used to explore the search space of a given problem to find the parameters necessary to capitalize on a particular objective.

In this paper, the issue of quantifying the weight of each QoS values is addressed, by upgrading the weight vector used in [6] and proposes an approach to select web services for different weight vectors with optimization technique using Particle Swarm Optimization algorithm to get the best weight vector and the best service for the given request. The results are compared among various PSO techniques namely static inertia weight, deterministic inertia weight and constriction factor.

II. Related Works

Service discovery is an important task before selection

process. Significant component involved in discovery is the matchmaking algorithm. To overcome the limitations of a syntax-based search author Paolucci proposed, matchmaking algorithms based on semantic techniques in [7].

QoS-aware web services selection plays an important role in Service Oriented Architecture. In [8], all of the possible quality requirements were enumerated and systematized into several groups including runtime-related, transaction support related, configuration management and cost-related QoS, and security-related QoS. Also they shortly present their definitions or possible determinants. Unfortunately, they failed to present quantifiable measurements. Especially, the work was presented in [9], which is also similar to [10]. There are, however, some differences to our approach carried out in:

1) The algorithm uses average ranking, neglecting nuances in different quality properties.
2) A possible maximum value is used to normalize the QoS matrix, although such kind of value is worth deliberating
3) Upon analyzing the experimental data, after normalization, the final result looks as G' = {0.769, 1.429, 1.334, and 1.111}, {0.946, 0.571, 0.666, and 0.889}.

Therefore, the approach in [12] is to normalize each quality metric into values between 0 and 1 by specifically defined measurements, which are fair to each quality metric. But weightage of each QoS was taken as vague and selection results have more deviation. Since human judgments including preference are often vague.

The survey from [11] showed that the selection of an optimal service is often turns out to be an NP-hard problem. In order to tackle the NP-hard problem, the application of Efficient Evolutionary Algorithms (EEAs) as Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Bee Algorithm and Firefly Algorithm, Shuffled Frog Leaping Algorithm, Memetic algorithm etc., works efficiently on optimization problems.

In paper, the author introduces a method for optimization of continuous nonlinear functions called Particle Swarm Optimization (PSO). It is a population-based search algorithm and is initialized with random population solutions, called particles [5]. Unlike the other evolutionary computation techniques, each particle in PSO is associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to fly towards better search area during the course of the search process.

In, they propose an approach to select web services for composition with particle swarm optimization and they evaluate proposed approach experimentally on real QoS data [12]. Experimental results showed significantly improvement in time performance of web service selection process in service composition system. But still they have not addressed the QoS weightage issues.

In, the author compared three different versions of PSO (static inertia weight, deterministic inertia weight and constriction factor) by applying three different types of velocity equations in PSO algorithm and their experimental results are analyzed to find best suited approach to solve the given problem [13].

The work proposed by [14] applied the essential principles in the fuzzy set theory and model the decision making problem as Fuzzy Multiple Criteria Decision Making. The main contribution of is to balance the specific weight which imitates human rating and objective weight which denotes reliability of evaluation conditions to form a synthetic weight. The detailed study of the synthetic weight for QoS-aware web service selection application is also presented.

Moreover modeling the web service selection problem as FMCDM, in this article, they introduced a synthetic weight which combines both the subjective and objective weights. For subjective weights defined by human preference, they applied linguistic variables and fuzzy numbers. For objective weights, investigate entropy concepts to improve the judgment consistency. A synthetic parameter is introduced to balance the two weights. But still optimized solution for given weightage was not proposed.

The author in [15] has included the brief discussions of constriction factors, inertia weights, and tracking dynamic system. In [16], the author compares two evolutionary computation paradigms, genetic algorithm and particle swarm optimization. The operators of each paradigm are revised, concentrating on how each affects search behavior in the problem space. Datasets [17] (http://www.wsdream.net) are part of author Zibin's PhD research work. The main objective of these Web service research datasets is to provide real-world data for future research. Even if the dataset is real-time, It is inadequate for our combined evolution of QoS attributes.

Another dataset presented in [2], [17] provides a base for Web Service researchers. It is a subset of 2500 real web service implementations that exist on the Web today. Using Web Service Crawler Engine (WSCE) these services were collected. The public dataset contains a set of 9 Quality of Web Service (QWS) attributes that have measured using commercial standard tools. It is found to be best data sets for all kind of selection process carried out.

From all the related works we have found that the selection process can still be improved by adopting the optimization techniques to the existing methodology [6] using PSO [5], the proposed selection method of optimizing the weight vector will select the service most efficiently based on weightage of QoS properties.

III. System Design

QoS-aware web service selection process is the layered architecture where the selection of web services were done through layers namely

- Web service Discovery
- Web service Selection

A. Web Service Discovery

Web services provide access to software systems over the Internet using standard protocols. In most scenarios there will be a Web Service provider that publishes a service and a Web Service Consumer that uses this service. Web Service

Discovery is the process of finding an appropriate Web Service for a given task. Publishing a Web service includes creating a software artifact and making it available to potential consumers. Web Service Providers extend an endpoint with an interface description using the Web Services Description Language (WSDL) so the consumer can use the service. Optionally, a provider can explicitly register a service with a Web Services Registry such as Universal Description Discovery and Integration (UDDI) or publish additional documents intended to ease discovery such as Web Services Inspection Language (WSIL) documents. Service consumers or users can search Web Services manually or automatically. The implementation of UDDI servers and WSIL engines should provide simple search APIs or web-based Graphical User Interface to help in finding the Web services.

B. Web Service Selection

The Web service selection mechanism is the process of selecting single or composite services. The selection mechanism uses the service framework and service classes from the other associated service. Under the user requests, the services with same functional characteristics and different non-functional characteristics are combined to make the optimal performance. These non-functional characteristics are related with weight vectors which contributes more for the selection process as because if the weight vector varies the result may also varies accordingly, so the varying weightage vectors are optimized to get possible best services in several iterations using particle swarm optimization.

Fig. 1 is the conceptual architecture of web service selection process using PSO algorithm. The following user's input are taken in consideration namely,

1. Input of the web service(has input tag value in Owl file)
2. Output of the web service(has output tag value in Owl file)
3. QoS Parameters

- Response Time in ms
- Availability in percentage
- Throughput in invokes/second
- Success ability in percentage
- Reliability in percentage

And along with weightage vector for each quality attributes, these weight vector is fixed and sometimes it will be taken as the input from the user. In the case of getting as input from the user, the results may deviate more often for the same request. As the weight vector varies according to user's needs, the results deviate and this creates vagueness. In order to clear this vagueness and to produce the optimal solution, the weight vector for service selection is optimized with randomly initialized weights using Particle Swarm Optimization (PSO) algorithm. The detailed system design given in Fig. 2 shows how the QoS aware service selection algorithms is integrated with PSO algorithm to give best web service with respect to the user's requirements using optimized weight vector.

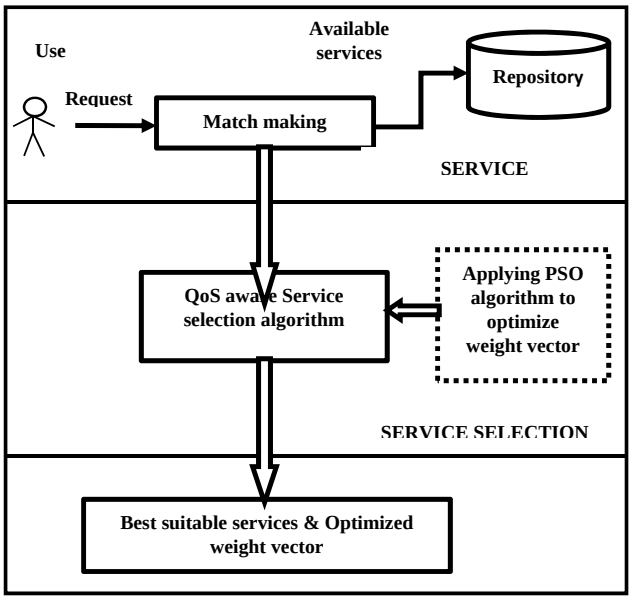

Fig. 1 Conceptual Architecture

IV. Proposed System

Web service selection process starts with service discovery phase followed by selection phase in which selection of web service is done with normalized QoS matrix proposed in [6] and PSO optimized weight vector. The steps are as follows as mentioned in Fig. 2.

Input: Input and Output of the web services and seven QoS attributes (Response Time, Availability, Throughput, Success ability, Reliability, Best Practices, and Latency)

Output: Best web service with respect to user's requirements and along with optimized weight vectors for the best service.

Step1. Data set upload

Step2. Getting the input from the user

Step3. Service discovery: Based on the web service Input and web service Output, the services were filtered, i.e., matching of has Input and hasOutput tag values in each Owl files of web service is found and only the exact match was taken as filtered result.

Step4. QoS matrix is generated with the filtered web service QoS attributes. QoS matrix is the *n*x*m* matrix where *n* is the no of web services (i.e., rows) and *m* is the types of quality attributes for each services (i.e., columns). Normalization of QoS Matrix: QoS matrix was normalized with *qmax* and *qmin* values and finally it is formulated within the range [0-1] where *qmax*=max value of the quality attribute, *qmin*= min value of the quality attribute in the QoS matrix.

Step5. Applying weight vector to normalized matrix using PSO algorithm

1) Initialize each particle as a 1x7 matrix randomly in a search space. Also initialize position and velocity=0 for the particles.
2) For each particle's position Evaluates fitness value

$$\text{Max (M)}= \sum_{i=1}^{m}(q_{ij} \ * \ W_i) \quad (1)$$

3) If the current fitness value is better than tje personal best, then set the current value as the new personal best for that particle.
4) Choose best fitness value from all the particles as gbest.
5) Update particles velocity using the following equation

Equation for particles velocity

$$V[]_{(id+1)} = w*v[]_{id} + c1*r1*(p[]_{pb} - x[]_{id}) + c2*r2*(p[]_{gb} - x[]_{id}) \quad (2)$$

Equation for Position update

$$X[]_{(id+1)} = x[]_{id} + V[]_{(id+1)} \quad (3)$$

6) If the fitness value converges to a point where no further improvements are not possible then stop the iteration and display the optimal solution, otherwise repeat from step 2.

V. Experimental Results

In order to evaluate the proposed approach, experiments are conducted using QWS real datasets. It comprises measurements of 7 QoS attributes for 2500 real-world web services. These services were collected from search engines, service portals, public sources on the Web, including UDDI registries, and their QoS values were using commercial benchmark tools. More details about this dataset can be found in [2], [17]. For weight vector optimization using PSO, the parameters are set as follows,

Population size = 50 (randomly initiated weight vector with the following condition)

$$\sum_{i=0}^{7} W_i \leq 10 \quad (4)$$

where W is the 1x7 matrix, each element in the matrix corresponds to weight of particular QoS attribute, c1=2.1, c2=2.1, number of iterations=50, r1 and r2 are random variables between 0 and 1 range and the inertia weight is updated in the following ways

A. Static Inertia Weight

The concept of an inertia weight was used in order to get better control on exploration and exploitation. The inclusion of inertia weight in the PSO algorithm was first reported in the literature by [14]. Equation (2) describes inertia weight approach (IWA), the velocity equation with an inertia weight included.

In this paper, inertia weight w is set 0.4 in the velocity vector update equation. It is a scaling variable that controls the impact of the previous velocity while computing the new velocity. Inertia weight values larger than one will characteristically cause the particle to speed up and explore larger regions of the search space; while smaller values will cause the particle to gradually slow down and do a finer search of the region.

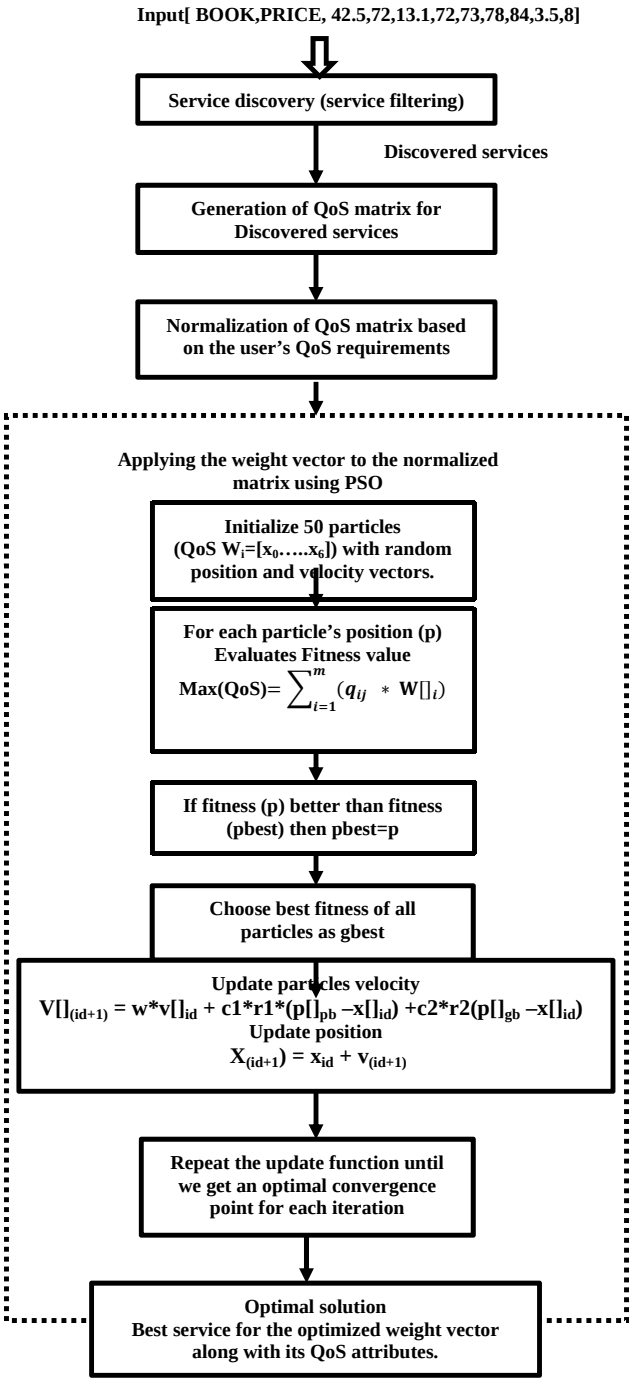

Fig. 2 Detailed System Design

B. Deterministic Inertia Weight

For the deterministic inertia weight calculation, the inertia weight w is gradually increased in the velocity update equation (2) to get more refined solution. The following weight function expressed in (5) is applied.

$$\mathbf{w = w_{min} - \frac{w_{min} - w_{max}}{n} \times i} \quad (5)$$

where $w_{max:}$ final inertia weight; $w_{min:}$ initial inertia weight; n: Maximum number of iterations; i:Current iteration.

A dynamically changing inertia weight provides PSO a better performance over a fixed value. It can be changed linearly over the course of PSO running or dynamically changed, based on the measurement of the PSO performance. Here for our experiments we set w_{max}= 0.9,w_{min} =0.4, n=50 and the weight is gradually increased from 0.4 to 0.9 over the iteration.

C. Constriction Factor k_f

Constriction factor $k_{f,}$ improves PSO's ability to constrain and control velocities and it is introduced by [5].

k_f, is computed as follows:

$$k_f = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|} \quad (6)$$

where

$$c=c1+c2 \text{ and } c > 4$$

A simplified method of incorporating it is represented in,

$$V[]_{(id+1)} = k_f*[v[]_{id} + c1*r1*(p[]_{pb} - x[]_{id}) + c2*r2*(p[]_{gb} - x[]_{id})] \quad (7)$$

A constriction coefficient is introduced so that it can guarantee a PSO to converge. Mathematically, the parameters c_1 and c_2 are equivalent. Here, according to the Clerc's constriction factor, if c is set to 4.1, then the constant multiplier k_f becomes 0.72.

Fig. 3 shows the results obtained from PSO algorithm with the above mentioned static inertia weight (w), deterministic inertia weight ($w_{max,}$ w_{min}) and Constriction parameters (k_f), and it shows the global best values during each iteration. The results show that the aggregated QoS values which accounts for the best services selection reaches its maximum fitness values at iteration 11 for static inertia weight and iteration 15 for deterministic inertia weight and constriction factor $k_{f..}$

The personal and global component variables, p_b and g_b, control the effect of the personal best and global best positions respectively. They are defined as p_b = rand ( ) × c_1 and g_b = rand ( ) × c_2 where rand ( ) generates random values between 0 to 1. The parameters c_1 and c_2 are not critical for PSO's convergence. However, proper fine-tuning will result in faster convergence and lessening of local maxima. PSO has a very little range to fine tune the parameter. Different inertia weights w, acceleration constants c_1 and c_2 have been chosen. Sensitivity analysis for parameters of PSO algorithm is carried out with different combinations of parameters.

To find the optimal values of the parameters for a population size and maximum number of iterations, a thorough sensitivity analysis is carried out for different combinations of parameter settings. It is observed that the maximum fitness value is at population size of 50 with the maximum number of iterations 50. For each selected w, c_1 and c_2, the fitness value obtained from simple PSO is recorded. It has been found that when w = 0.4, c_1 = 2.1 and c_2 =2.1 the run finds better optimum than all other values of w, c_1 and c_2.

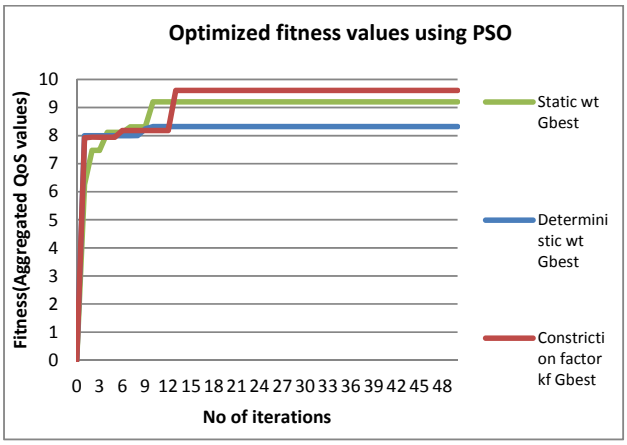

Fig. 3 Fitness analysis using optimized weight vector

VI. Conclusion

The proposed improvement preserves the architecture presented in Wang's work and proposes an extension that includes the optimization of weight vector for the given user's request. The selection process is improved effectively by optimizing the randomly initiated weight vectors with fine-tuned static inertia weight, deterministic inertia weights and constriction factor. From the results, the constriction factor yields higher fitness among all other versions of PSO algorithm and hence it is a suitable approach for the selection process.

References

[1] Z. Stojanovic, A. Dahanayake and H. Sol, "Modeling and design of service oriented architecture", *in Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, Hague, Netherlands, Vol. 5, pp. 4147- 4152, Oct.2004.

[2] C.L. Huang and K. Yoon, "Multiple Criteria Decision-making", *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, New York, 1981.

[3] E. Al-Masri, and Q.H. Mahmoud, "Qos-based discovery and ranking of web services", *in Proc. 16th IEEE Int. Conf. Comp. Commun. Networks*, pp. 529-534, 2007.

[4] J. L. Cohon and D.H. Marks, "A Review and Evaluation of Multi-objective Programming Techniques", *J. Water Res.*, Vol. 11, no. 2, pp. 208–220, April 1975.

[5] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", *in Encyclopedia of Machine Learning*, Springer US, 2010, pp. 760-766.

[6] X. Wang, T. Vitvar, M. Kerrigan and I. Toma, "A qos-aware selection model for semantic web services", *in Proc. 4th Int. Conf. Service-Oriented Comput. –ICSOC 2006*. Springer Berlin Heidelberg, 2006. pp. 390-401.

[7] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, "Semantic matching of web services capabilities", *Lecture Notes in Computer Science The Semantic Web—ISWC*, Vol. 2342, 2002, pp 333-334.

[8] Y. Liu, A.H.H. Ngu and L. Zeng, "QoS Computation and Policing in Dynamic Web Service Selection", *in Proc. 13th Int. Conf. World Wide Web*, New York, USA, pp. 66-73, 2004.

[9] X. Wang, T. Vitvar, M. Kerrigan and I. Toma, "A qos-aware selection model for semantic web services", *in Proc. 4th Int. Conf. Service-Oriented Comput. –ICSOC 2006*. Springer Berlin Heidelberg, 2006. pp. 390-401.

[10] C. Rajeswary, "A survey on efficient evolutionary algorithms for web service selection", *Int. J. Manag. IT and Engg.* Vol. 2, no. 9, pp. 177-191, 2012.

[11] T. Zhang, "QoS-aware Web Service Selection based on Particle Swarm Optimization", *J. Netw.*, Vol. 9, no. 3, Mar. 2014, pp. 565-570.
[12] N. Arulanand., S.Subramanian, and K.Premalatha, "Optimized Bin Bloom Filter for Spam Filtering using Particle Swarm Optimization", *European J. Scientific Research*, pp 199-213, 2012.
[13] Y. Liu, A.H.H. Ngu and L. Zeng, "QoS Computation and Policing in Dynamic Web Service Selection", *in Proc. 13th Int. Conf. World Wide Web*, New York, USA, pp. 66-73, 2004.
[14] RC. Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources", *in Proc. 2001 EEE Int. Conf. Evol. Comput.* Vol. 1, Seoul, South Korea, pp. 81-86, 2001.
[15] R.C. Eberhart and S. Yuhui, "Comparison between` genetic algorithms and particle swarm optimization", *Evolutionary Programming VII*. Springer Berlin Heidelberg, Vol. 1447, 1998, pp 611-616.
[16] Y. Zhang, Z. Zheng, and M.R. Lyu, "Wsexpress: A qos-aware search engine for web services", *in Proc. IEEE 20th Int. Conf. Web Services*, Miami, Floarida, pp. 91-98, 2010.
[17] E. Al-Masri, and Q.H. Mahmoud, "Discovering the best web service", *in Proc. 16thInt. Conf. World Wide Web*, New York, USA, pp. 1257-1258, 2007.