VOL. 9, NO. 12, DECEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A SURVEY ON APPLICATIONS OF EVOLUTIONARY TECHNIQUES IN WEB SERVICE SELECTION

K. Mohan[1] and C. Kalaiarasan[2]
[1]Department of AM and CS, PSG College of Technology, Tamil Nadu, India
[2]TamilNadu College of Engineering, Tamil Nadu, India
E-Mail: mrmohan78@gmail.com

ABSTRACT

Selecting best services over internet has remained an issue for the community of researchers who concern with the problem of selecting best services. The advances in the area of service selection in terms of quality have empowered the ability to host the artificial intelligence with the use of soft computing techniques for selecting web services dynamically in a timely manner. In this paper we study and analyze the performance of soft computing techniques including evolutionary algorithms such as Genetic Algorithm, Particle Swarm optimization, Ant Colony optimization and Memetic algorithm for selecting the suitable service on a travel domain space by considering the quality parameters. The results obtained in this extensive study are compared and analyzed based on their performance metrics.

Keywords: web services, genetic algorithm, particle swarm optimization, ant colony optimization, memetic algorithm, cuckoo search.

INTRODUCTION

The definition of web services by W3C consortium is "software designed to support interoperable machine to machine interaction over a network" [1]. Web service is a collection of set of protocols that communicate with each other. The platform of web service is a combination of HTTP protocol and XML messages. The open protocols make the web services to behave as platform independent. XML provides a way for communicating through different platforms and programming languages. Web services use XML to code and decode the data and SOAP to establish sessions for information exchange over network such as internet [2]. Web Service Description Language (WSDL) serves as a model and XML to describe the syntax of web services. This WSDL is associated with UDDI (Universal Description, Discovery and Integration).

Composition of the existing web services in to new services to achieve the desired task is termed as the Web Service Composition. This works well if a single service does not satisfy the user's functionalities and on the other hand if the combination of one or more services fulfill the needs. The service provider usually describes the service composition in UDDI [3]. The service requestors search these registries and find the services. The purpose of web service selection is to select an optimal web service that suits the client request. In case of static web service discovery a single provider is matched with a client. In case of dynamic discovery the services may grow often resulting in more than one service provider for a client. Hence a set of instructions or rules are followed to handle the client requests. Instructions refer to various selection techniques [4]. The architecture of web service is given in the Figure-1.

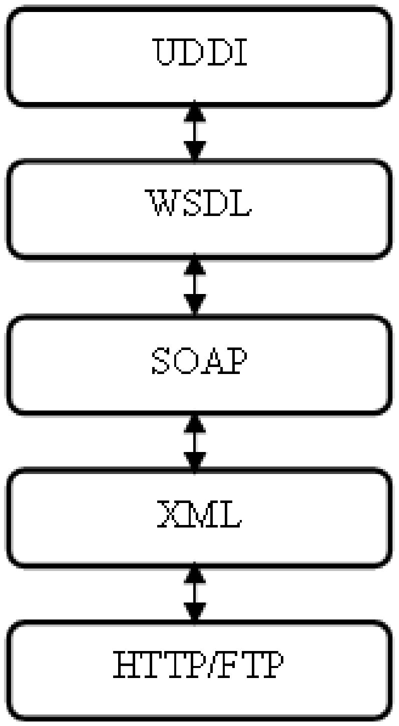


Figure-1. Architecture of web service.

This paper is organized as follows: Section II describes the related work done in this area. Section III gives the problem definition, Section IV discusses about the QoS model and evaluation. Section V presents a detailed view of algorithms followed by Section VI with experimental results and Section VII with conclusion drawn from experiments.

RELATED WORK

Problem of the dynamic web service selection considering the QoS factors to find an optimal solution can be addressed using the multi-objective genetic algorithm to form the service composition and to optimize with various fitness functions [5]. The prematurity is a major problem in genetic algorithm hence two policies namely initial population and evolution policy have been designed [6]. GA eliminates the individual with less feasibility during the selection process. These individuals might be an essential part to provide the optimal solution. Hence hybrid algorithms have been proposed which gives penalty for the infeasible individuals to obtain a optimum results [7]. A heuristic search that provides optimal solution with the assumption that similar web services contains different QoS and costs has been proposed and it is shown that it produces optimal solution nearer to 99% using simulation results [8]. An experiment conducted with the population based algorithms such as Particle Swarm Optimization (PSO), Bat Algorithm (BA) and Genetic algorithm (GA) for training feed forward networks had shown that the Bat algorithm outperforms the other two algorithms [9]. A survey on evolutionary techniques has been conducted and result states that large optimization problems cannot be solved by all evolutionary techniques [10]. Here performance of several evolutionary algorithms such as Genetic, PSO, Memetic, ACO and Cuckoo were compared and presented.

PROBLEM DEFINITION

A motivating example for this web service selection problem is arranging a travel plan. A person who chooses to take travel will approach a service provider (agent) to book his tickets. In addition the providers may also offer certain functionalities such as arranging hotels, cabs etc. The customer may choose a provider whose cost is less and quality is good. Taking this situation as an example we have designed a lets travel website which can be used by customers to search the flights, hotels and cabs. Three providers each having link with one another: Flight Booking Service, Hotel Booking Service, Car Renting Services. The service composition consists of numerous services with same functionalities. When a consumer looks into service registry, a list of available services for their category about source, destination places, available flights, hotels, cabs are listed. When a request is made a suitable service is displayed as outcome. The architecture of travel plan is shown in the Figure-2.

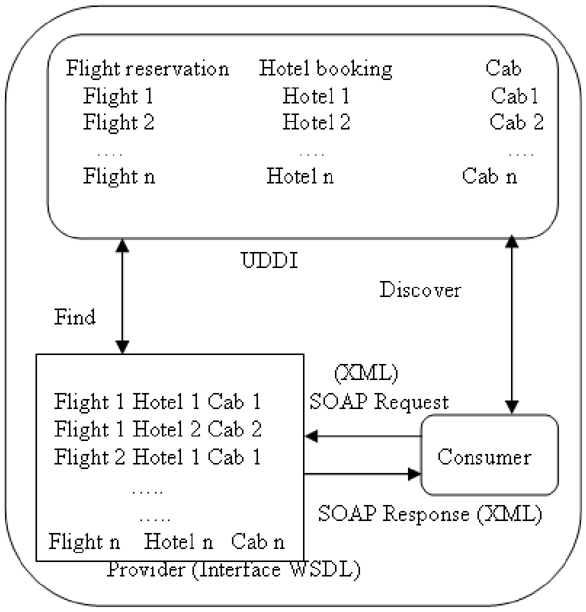


Figure-2. Architecture of service selection in travel plan domain.

QoS evaluation model

Multiple web services are provided by different service providers with both the functional (Purchase, Payment) and non-functional requirements (Response Time, Reliability, Availability) properties. Selecting services based on the QoS for simple and complex tasks involves updating of QoS attributes continuously. This can be updated by monitoring the QoS parameters by a third party broker. Here, we have identified a set of four attributes namely Reliability, Availability, Response Time and Cost. These can be divided into two groups, the first group of QoS attribute values should be minimized i.e. higher the value of response time and cost lower will be the quality. The second group has to be maximized such as Availability, Reliability. QoS is a combination of various qualities that includes the following:

- **Response time:** Response time is the time interval between the service request and service response between consumer and provider.
- **Cost:** Cost represents the money that the user pays to avail the services
- **Availability:** It is the percentage of time the service is available for the consumer.
- **Reliability:** It is the measure of degree of the services to be capable of maintaining service and service quality. Eg. Number of failures per month or year.

The overall score is calculated as follows:

$$\text{Score} = \begin{cases} \sum_{j=1}^{n} \frac{Q_j^{max} - q_{ij}}{Q_j^{max} - Q_j^{min}} \text{ if } Q_j^{max} - Q_j^{min} \neq 0 \\ \sum_{j=1}^{n} \frac{q_{ij} - Q_j^{min}}{Q_j^{max} - Q_j^{min}} \text{ if } Q_j^{max} - Q_j^{min} \neq 0 \end{cases} \quad (1)$$

Where Q_{ij} represents the quality matrix with i services and j attributes. Q_j^{max} represents the maximum value in the quality matrix. Q_j^{min} represents the minimum value present in the quality matrix.

EVOLUTIONARY ALGORITHM FOR WEB SERVICE SELECTION

Many researches have been made in the past decades to address the problem of web service selection using evolutionary algorithms. Here the travel plan dataset is tested against the Genetic Algorithm, Particle Swarm Optimization, Memetic algorithm, Ant Colony Optimization and Cuckoo Search.

GENETIC ALGORITHM

Genetic algorithm works on a random set of initial population. The populations are represented as chromosomes. These chromosomes are used to represent possible solution that matches the user's request. The population is then subjected to a fitness function in which fitness of each individual is calculated. The fitness function for the web service selection scenario is given by

$$\text{Fitness Function} = \sum_{i=1}^{n} \frac{q_{i,resp} + q_{i,cost}}{q_{i,rel} + q_{i,ava}} \quad (2)$$

Where represents the response time of i_{th} service.

- represents the cost of i_{th} service.
- represents the Reliability of i_{th} service.
- represents the Availability of i_{th} service.

Individuals are ranked based on their fitness values. Lesser the fitness value more it is optimized. Candidate services with lower fitness are selected and subjected to crossover and mutation. Crossover is used to alter the chromosomes in the population. It can be one of the two categories: Real Valued and Binary valued. There are three types of crossover exists: One-point, two-point, Uniform crossover. Example for two point crossover is shown below:

10001010 + 11011111 = 10011110

Mutation is the process of flipping of bits. An example of mutation is as follows:
11001000 → 10001000

The result of chromosomes obtained after the above methods form the new off spring for the next population. This process repeats until the specified number of generations is reached or an optimum solution is found. In our approach the fitness value of each service is calculated using (2). Here individual with less fitness value is considered to be more fit. So the individuals are ranked from lower to higher fitness values. We have used tournament selection with k=2 for selection technique. The selected services are crossovered and mutated to produce new service with higher quality. As generations increases a best service will be produced as result. Algorithm for genetic algorithm is given as follows:

```
a)   // Set GA parameters
n->population size, G->Total Number of Generations
μ-> Mutation Rate, α->Crossover Rate
b)   // Initialize the population
P_k:Population created randomly with specified population size
c)   // Evaluate the Pk
Calculate fitness (i) for each I □ P_k
Do
{

Selection
        Tournament selection with specified k value is applied for selection.

Crossover
        Select pairs from sorted list, perform crossover such that α x n = No. of pairs to be coupled for mating
Place the newer off springs in P_{k+1}

Mutation
        Select μ x n individuals of P_{k+1}; invert the bits of selected individual.
d)   // Evaluate P_{k+1}
Compute fitness (i) of each individual, i ∈ P_{k+1}
e)   // Increase the generation
k=k+1;
}
Return fittest individual at each generation.
Continue Steps 3 to 5 until the specified G is reached.
```

PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization falls into the category of evolutionary algorithm. PSO uses an objective function which is optimized by searching through the population. Unlike genetic algorithms the PSO does not have crossover and mutation operators. Similar to genetic algorithm, the populations are initialized and randomly generated. The solutions called as particles are allowed to freely move in multidimensional search space.

During the flight movement each particle has its own velocity and position based on its own experience and best experience of entire population. In the each iteration the particles move towards a best solution [11]. In our approach, initial global fitness is set to zero. At first iteration a local best service is obtained. It is then assigned to global best. For the next iteration another local best service is obtained and is compared with the global best value. If it is local best is better than global best, the global best is updated. This process repeats until a global best fit is obtained. Algorithm for PSO is given as follows:

```
A.  // Set the parameters
I->Iterations, gbest-> Global best, lbest->Local best,
S-> Particle Size
Initially set gbest=0, lbest=0
B.  //  Randomly Initialize the particles
$P_s$ : population with specified population size.
C.  // For each particle
Initialize the particle $P_s$
D.  do
For each particle
Calculate fitness values
If the fitness value is better than lbest set it as lbest
end
E.  Choose the particle with highest fitness value and set
    it as best
F.  Calculate the particle velocity
G.  Update the particle position
H.  Repeat till maximum iterations I or minimum error is
    obtained
```

ANT- COLONY OPTIMIZATION

Initially the population is randomly generated and it is constructed as web service composition (WSC) graph. The n services are initially positioned on the source vertex. The aim of ant colony optimization algorithm is to find an optimal path from source to destination. An ant is a computational agent and the work of ant is to construct solutions for problem iteratively. While finding path the ant will make state transition rule to choose another ant. Using local updating rule it modifies the amount of pheromone on the visited edges. Once the ants completed their travel, the vertex which ant visits composes an execution path. The amounts of pheromone on edges on the best optimal service execution path are modified with global updating rule. Higher the amount of pheromone more chances to be chosen [12]. In our approach the populations are combination of services along with total fare. An acyclic directed graph is constructed with the services. Here each node corresponds to particular service and each edge corresponds to fitness value. Three levels of nodes are used here flights, hotels and cabs. At each level the algorithm selects a service or node with higher fitness value. Thus at all the three levels selection is made and finally optimized services are obtained. Algorithm for ACO is given as follows.

```
(i)   Initialize the parameters
n-> Total  number of services, m-> Total number of ants
(ii)  Generate the population randomly
(iii) Do
Randomly position m ants on n services
For services = 1 to n
For ant = 1 to m
Select the next service according to the exploration and
exploitation mechanisms
Apply local updating rule
End
End
(iv)  Apply global updating rule using the best
(v)   ant found
(vi)  Until End condition
```

MEMETIC ALGORITHM

Evolutionary algorithms are not well suited for search including combinatorial spaces. A popular local search technique such as Memetic algorithm is an extension of evolutionary algorithms which aim to refine solutions by improving fitness of individuals by applying techniques such as hill-climbing or stimulated annealing. Hill climbing is an optimization technique which belongs to a family of local search algorithm. This algorithm works well in the situations where the problem domains have many solutions, out of which a best solution is selected. This works by randomly choosing a solution and iteratively the solutions are changed such that each time it improves a little. If no improvements the algorithm terminates. In this paper the solutions are selected based on the cost. For further iterations the solution looks for other best services based on the cost. As a result a best service is selected and is given as input to algorithm. These subsets applied to algorithm follow the same process as GA to select best solution with the best global fitness value. Algorithm for MA is shown as follows.

```
a)  Begin
b)  Initialize the population
c)  Calculate fitness of each individual
d)  do
Select parents
Recombine them to produce offspring’s
Mutate the offspring’s
Perform local search
Begin
while (termination condition is not satisfied)
do
New solution ← Neighbor (best solution);
If (new solution > actual solution)
Best solution ← actual solution
End if
end
end while
End
e)  while (termination ends)
```

CUCKOO SEARCH

Cuckoo search is an optimization algorithm. Unlike other birds cuckoo will lay their eggs in other species nest. If those birds discover that it is not its own, it either throws away or abandons its nest. The host bird will discover the alien egg with probability p € [0, 1]. A fraction of nests n are replace with new nests (with new random solutions). The fitness of solution is proportional to the objective function. Here each egg in the nest represents a solution and each egg by cuckoo represents a new solution. The cuckoo can place its egg in any of the randomly chosen nest. The best nest with high quality eggs are promoted to next generation. The basic steps of the cuckoo search are defined as follows.

To generate a new solution, x(t+1) a cuckoo is selected by Levy flight search.

$$x_i^{t+1} \blacksquare x_i^{t}\ \alpha \oplus_{Levy}^{\delta}$$

Here step size α=1. A levy flight is performed when α > 0. The above equation is essential for a stochastic random walk. In general Markov chain is used to find the next stats/location depending on the current location. In our work the random walk is via Levy flights and it is more efficient in exploring search space. Levy walk generates new solutions over the best solution obtained. To avoid the local optimum problem, a set of solutions whose locations are far enough from the current best solutions must be generated [13, 14]. The algorithm for the cuckoo search via Levy flights is given as follows:

```
a) Begin
b) Generate initial population $X_i$ (i=1, 2, 3, ..n)
c) while (t > MaxGen) or (Stopping criterion)
Obtain a cuckoo randomly from Levy flights
Evaluate quality/fitness $F_i$
Choose a nest among n (say, j) randomly
If ($F_i > F_j$)
replace j by the new solution
End
Abandon the fraction (Pa) of worse nests
and build new nests at new locations via
Levy flights
Keep the best solutions
Find the current best by ranking the solutions
d) end while
e) Post process result and visualization
f) End
```

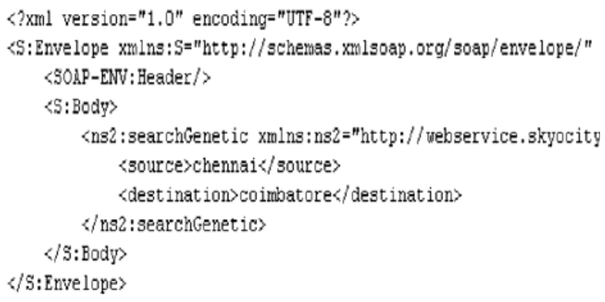
EXPERIMENTAL RESULTS

Experiments have conducted with the lets travel database using Genetic Algorithm, Particle Swarm Optimization, Memetic algorithm, Ant colony Optimization and Cuckoo search. The experiments were conducted on a core of Intel ® Core ™ 2 Duo CPU E7500 (2.93 GHz) on an Acer laptop with 2 GB RAM memory using Java version 1.6.0 JDK Runtime Environment on Windows 7. We have taken a total of 20 services each (Flight, Hotel, Cab) with four QoS parameters namely Reliability, Availability, Cost and Response Time. A sample of this service composition is show in the Table-1.

Table-1. Service composition sample.

Flight name	Flight ID	Hotel name	Hotel ID	Cab name	Cab ID
Jet Airways	1001	Hilton	H1	SRS	C1
Spice Jet	1002	The Park	H2	Marutham	C2
Indigo Airlines	1003	Green Park	H3	Sathya	C3
JetKonnect	1004	Fortune	H4	KPN	C4
Air India	1005	Accord	H5	Air Zone	C5

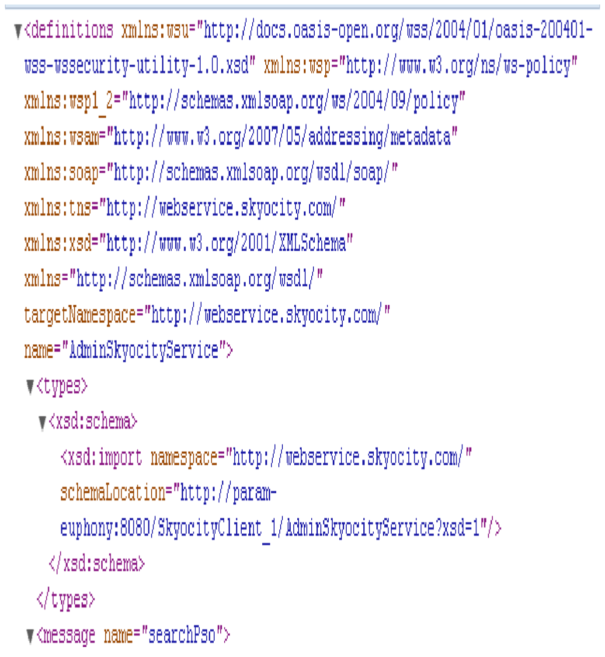
The service provider sends a WSDL file to the UDDI registry. This UDDI registry tells the customers that which services belongs to which provider. The service requestor contacts the UDDI registry to find the service provider for the service they needs. The requestor then contacts the service provider using SOAP protocol (SOAP request is made). The service provider validates the service request and sends SOAP response through XML files. The requestor validates the response and decodes the data. For example a consumer plans to travel from Coimbatore to Chennai. He then looks for the travel package (Flight, Hotel, Cab) in UDDI registry. If cosumer wants to select a provider who satisfies their travel plan requirements, the customer contacts the registry using SOAP request. The SOAP request for this scenario is given in Figure-3.



```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:searchGenetic xmlns:ns2="http://webservice.skyocity
            <source>chennai</source>
            <destination>coimbatore</destination>
        </ns2:searchGenetic>
    </S:Body>
</S:Envelope>
```

Figure-3. SOAP request.

Once the request for travel is made, the available services are listed. These resultant services may have same functional requirements but different QoS attributes. The validation of the SOAP request is made with xml files is shown in Figure-4.



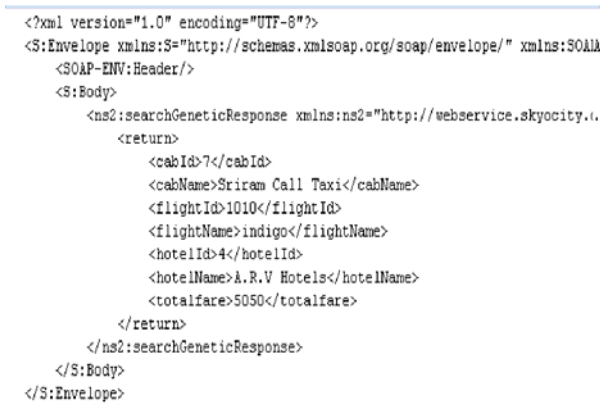
```
▼<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
 wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
 xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
 xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:tns="http://webservice.skyocity.com/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="http://schemas.xmlsoap.org/wsdl/"
 targetNamespace="http://webservice.skyocity.com/"
 name="AdminSkyocityService">
 ▼<types>
  ▼<xsd:schema>
     <xsd:import namespace="http://webservice.skyocity.com/"
     schemaLocation="http://param-
     euphony:8080/SkyocityClient_1/AdminSkyocityService?xsd=1"/>
   </xsd:schema>
  </types>
 ▼<message name="searchPso">
```

Figure-4. Web Service Description Language (WSDL).

In this situation the evolutionary algorithms are applied to help the customers in choosing best services according to their needs. Here we have performed experiments to make a decision on the standardized algorithm that would help the customers to choose services based on their requirements. The initial parameters for running the experiments are shown as below:

Total Combinations: 120, Total Iterations: 20
Initial Global Fitness Value: 0

SOAP response to the above SOAP request is shown in Figure-5.



```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOA
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:searchGeneticResponse xmlns:ns2="http://webservice.skyocity.
            <return>
                <cabId>7</cabId>
                <cabName>Sriram Call Taxi</cabName>
                <flightId>1010</flightId>
                <flightName>indigo</flightName>
                <hotelId>4</hotelId>
                <hotelName>A.R.V Hotels</hotelName>
                <totalfare>5050</totalfare>
            </return>
        </ns2:searchGeneticResponse>
    </S:Body>
</S:Envelope>
```

Figure-5. SOAP Response.

Each algorithm is made to run for 20 iterations. According to our fitness function, it is to be noted that lower the fitness values higher will be the quality. From the above Table (See Table-2) it is clear that the ACO outperforms the other four techniques.

Table-2. Fitness values obtained from GA, PSO, ACO, MEMETIC and CUCKOO Search Algorithm.

Algorithms	Fitness values
GA	0.73120
PSO	0.6592
ACO	0.629090
Memetic	0.64142
Cuckoo	0.63599

CONCLUSIONS

In this paper, evolutionary techniques such as GA, PSO, ACO, MA and Cuckoo were demonstrated in a realistic scenario and successfully applied for the web service selection problem. Algorithms that solves large optimisation problems need more computational time such as GA and PSO. To select services based on the quality factors these algorithms were not satisfactory and hence needs to be integrated with other decision making algorithms such as fuzzy logic. A combination of these algorithms can also be made to overcome the time complexity and local optimisation problems which can provide better exploration of search process and applied for future work.

REFERENCES

[1] http://www.w3.org/.

[2] Indrani Balasundram and E. Ramaraj. 2001. Prevention of SQL Injection Attacks by Using Service Oriented Authentication Technique. International Journal of Modeling and Optimization. 3(3): 302-306.

[3] S.A. Ludwig. 2011. Single-objective versus multi-objective genetic algorithms for workflow composition based on service level agreements. In Proceedings of IEEE International Conference on Service Oriented Computing and Applications (SOCA 2011), Irvine, California, USA.

[4] Pandey. A. and S.K. Jena. 2009. Dynamic approach for web services selection. Proceedings of the International Multi Conference of Engineers and Computer Scientists.

[5] Y. Charif-Djebbar and N. Sabouret. 2006. Dynamic Web Service Selection and Composition: An Approach Based on Agent Dialogues. Proceedings of the 2006, IEEE/WIC/ACM, Springer, Hong Kong. 4294/2006: 515-521.

[6] Yue Ma and Chengwen Zhang. 2008. Quick convergence of genetic algorithm for QoS-driven web service selection. Computer Networks. 52(5): 1093-1104.

[7] Liu Shu-Lei, Liu Yun-Xiang, Zhang Fan, Tang Gui-Fen and Jing Ning. 2007. A Dynamic Web Services Selection Algorithm with QoS Global Optimal in Web Services Composition. Journal of Software. 18: 648-656.

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

[9] R. Berbner, M. Spahn, N. Repp, O. Heckmann and R. Steinmetz. 2006. Heuristics for QoS-aware web service composition. In Proceedings of the IEEE International Conference on Web Services (ICWS2006), IEEE Computer Society. pp. 72-82.

[10] Khan. K and Sahai. A. 2012. A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. International Journal of Intelligent Systems and Applications (IJISA). 4(7): 23-29.

[11] Xia, Y. Chen, Z. Li and H. Gao and Y. Chen. 2009. Web Service Selection Algorithm Based on Particle Swarm Optimization. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing. pp. 467-472.

[12] X. Zheng, J. Luo and A. Song. 2007. Ant Colony System Based Algorithm for QoS-Aware Web Service Selection. In Grid Service Engineering and Management. pp. 39-50.

[13] Yang X. S. and Deb S. 2010. Engineering Optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation. 1: 330-343.

[14] Sourav Samantaa, Nilanjan Deyb, Poulami Dasb, Suvojit Acharjeec and Sheli Sinha Chaudhuri. 2012. Multilevel Threshold Based Gray Scale Image Segmentation using Cuckoo Search. In proceedings of ICECIT, Elsevier. pp. 27-34.