Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An optimization Algorithm for QoS- Aware Web Service Selection based on Animal Scavenging Behaviour

Beena @ Hassina C. *
Department of Computer Science,
Pondicherry University
India

Sathya M.
Department of Computer Science
Pondicherry University
India

Abstract— Evolutionary algorithms (EA's) are quite popular as they are used for solving real world complex Np-hard problems. In this paper, a new stochastic Animal Scavenging Behaviour (ASB) algorithm based on the foraging behaviour of animals is presented. In ASB the initial population is divided into four categories of individuals namely producer, cluster heads, scroungers and rangers. The proposed scheme provides different forms of searching which are employed by the individuals to modify their search paths. Each Scrounger selects a cluster head as its spearhead and move towards it. Cluster Heads select the global best cluster head as the Producer and adjust their positions based on their information. Thus the proposed ASB algorithm follows these different tactics to mitigate the problem of getting struck at local optima and premature convergence. Web service selection a NP- hard, combinatorial optimization problem is a scheme for choosing appropriate concrete services that fulfils the user's Quality of Service (QoS) parameters from the registry to form the composite service. QoS parameters such as cost, response time, reliability, availability and accuracy of services is vital for optimal web service selection, a significant component of web service composition. Some EA's such as Genetic algorithm (GA), Particle swarm optimization (PSO) were used to solve the QoS driven web service selection problem. In this work we have applied ASB to solve the intricate service selection problem. To prove the robustness and efficiency of the performance of ASB, it is tested with benchmark functions both in unimodal and multimodal functions in high and low dimensions. The performance of ASB is also statistically compared with the other competing algorithms namely GA, PSO and GSO. Simulation results illustrates that ASB remarkably outperforms GA, PSO and GSO algorithms. The promising results obtained by ASB shows its capability of solving real world combinatorial problems.

Keywords*— Web service selection, NP-hard, Combinatorial optimization problem, concrete service, web service composition, Group Search optimizer (GSO), Quality of Service (QoS).*

I. INTRODUCTION

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 difficult problems than the standard methods [1]. 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. Evolutionary algorithms are classified into three types namely i) Evolutionary algorithms ii) Swarm Intelligence iii) Bacterial foraging algorithms. Many different techniques are employed for solving optimization problems and the most common method is solving using nature inspired algorithms. Evolutionary algorithms are primarily used as they can adapt to the dynamic changes in the environment, it can be crossed with other conventional methods and are helpful in solving issues that does not possess any solution. Evolutionary algorithms are nowadays employed in solving various real world applications such as combinatorial problems, network routing problems, image processing problems and multi-objective optimization problems. The evolutionary algorithms have extraordinarily enhanced to a prodigious level in recent years. The most familiar evolutionary computation approach is the Genetic Algorithm (GA) which is a population based algorithm used to solve combinatorial optimization problems. By employing the Darwin's theory of evolution the framed optimization problem can be resolved. In this method the population size is maintained to be constant and it involves operators such as natural selection, crossover and mutation for evolution process [2]. With the aid of these operators global search is done efficiently. The primary classic illustration in the field of swarm intelligence is particle swarm optimization (PSO) algorithm. By perceiving the cooperative behavior of animals these algorithms were devised. PSO is a classic swarm intelligence method that was developed by Eberhart and Kennedy. The PSO algorithm was inspired by the co-operative behavior of flock of birds. In PSO by choosing the particle with best fitness as gbest and moving near that particle, exploitation is achieved. Pbest denotes the previous best position of the particle and each particle remembers it pbest to discover the gap between its previous best position and global best position. Despite PSO being one of the dominant schemes for optimization problems it grieves from some problems such as stagnation and premature convergence. In order to overcome and improve these problems many methods such as inertia weight and time varying co-efficient were suggested [3]. The ACO algorithm is a swarm intelligent technique that is inspired by the cooperative rummaging behavior of ants. This algorithm was motivated by the laying of pheromones by ants. Ant Colony Optimization has been widely used to solve complex combinatorial optimization problems [4]. Simulated Annealing is an evolutionary algorithm that models the hardening process. In this process a material is heated above its melting temperature and then it is let to cool slowly to yield the crystalline lattice [5]. This reduces its energy probability distribution. The foraging behavior of bacteria has led to a source of developing a novel evolutionary algorithm coined as bacterial foraging algorithm. Some of the standard bacterial foraging algorithms are COSMIC and RUBAM. Another bio-inspired algorithm inspired from the mosquitoes was termed as MOX. The inspiration was drawn from discriminating performance of female mosquitoes in selecting an environment to lay their eggs and the reserve of those eggs to hatch into the next stage. [6] Presents a new optimization algorithm named as the cuckoo optimization algorithm (COA). This algorithm was inspired by the egg laying and breeding of cuckoos. The origin for the cuckoo optimization algorithm was driven by the effort taken by the cuckoos to survive. During this strive some of the cuckoo eggs are killed. The cuckoos that have survived colonize to a better environment and continue their reproduction cycle of laying eggs. Group search optimizer algorithm is a stochastic Evolutionary optimization algorithm that mimics the foraging behavior of animals. Group search optimizer algorithm (GSO) is a random search approach to solve large-scale optimization problems and combinatorial problems. When tested against benchmark functions, in low and high dimensions, the GSO algorithm has competitive performance to other EAs in terms of accuracy and convergence speed, especially on high-dimensional multimodal problems. In order to accomplish the foraging task in GSO algorithm the producer-scrounger strategy is employed [7]-[9]. Producing refers to the activity of searching for food and scrounging means joining the group for foraging [10].

In addition to these renowned methods of optimization, much exploration is done on evolutionary algorithms to solve complex real world problems. In this paper, a new stochastic Animal Scavenging Behaviour (ASB) based on the scavenging behaviour of animals is presented. In ASB the initial population is divided into four categories of individuals namely producer, cluster heads, scroungers and rangers. The proposed scheme provides different forms of searching which are employed by the individuals to modify their search paths. Each Scrounger selects a cluster head as its spearhead and move towards it. Cluster Heads select the global best cluster head as the Producer and adjust their positions based on their information. Rangers perform random walks while looking for its resources. Thus the proposed ASB algorithm follows these different tactics to mitigate the problem of getting struck at local optima and premature convergence.

The rest of the paper is organized as follows: In Section 2, the problem is formulated for QoS-Aware Web Service selection. Section 3 elucidates the proposed ASB algorithm and Section 4 illustrates the mapping of ASB to web service selection problem. Section 5 reports the numerical test functions, experimental settings of the algorithms and experimental result analysis done with other competing algorithms namely GA, PSO and GSO and finally section 6 concludes this work.

II. QOS-AWARE WEB SERVICE SELECTION PROBLEM FORMULATION

.

Web service selection is process to dynamically bind the services for every abstract task to form a composite service. The services are selected based on functional attributes and non-functional attributes. Functional attributes denotes the duty that the service has to perform and non-functional attributes denotes selection of services based on QoS metrics. The service provider defines the QoS metrics before supplying the services to the service requesters [11]. The user specifies the service request to the provider specifying the preferences of the service. If the user requirement is satisfied by a single service then the problem of service selection becomes very simple by selecting the service with best fitness. This service can be provided to the service requestor. However it is not possible to satisfy the user’s constraints with a single service. Since the problem domain is very large and to satisfy the customer requirement with their preferred non- functional quality, service selection is done. Web service has four structures: sequential (a), cycle (b), parallel (c) and branch structure (d) [12]. In the sequential structure, tasks are executed in a sequential order; In the cycle structure, a task will be executed for multiple times; In the parallel structure, all the parallel tasks can be executed simultaneously, but it cannot go to the next task until all the parallel tasks are finished; In the branch structure, each task in the branch can finish the same component function and system can go to the next task once one task in the branch has been finished. In this paper, the user request is focused and limit to cost, response time availability and reliability. The individual representation of composite services is represented in Fig.1

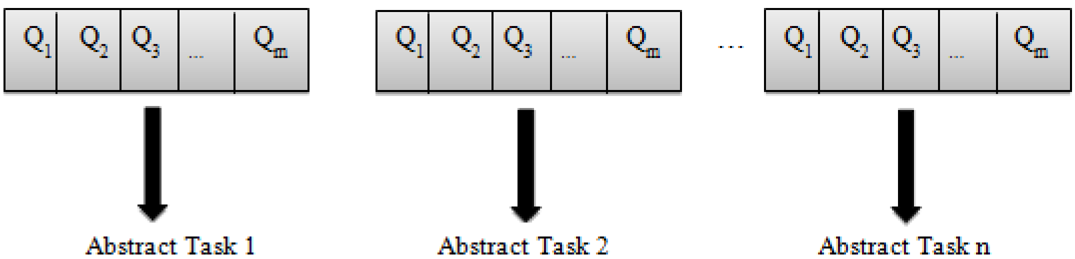

Fig.1 Individual representation of a composite service

Where Q_1 … Q_n denote the quality attributes of each abstract services. Table 1 shows the QoS model of Web service used in this project, and specifies the corresponding discussions of each QoS attribute. The combined measure is the linear combination of all the four objective functions such as availability, reliability, response time and price. The objective function for combined measure is given by

$$\begin{cases} \max F = fitness(CS) = \sum_{i=1}^{n} w_i Q_i \\ \text{s.t. } Q_i \le Q_i^{0}, \; i=1,\dots,n. \end{cases} \tag{1}$$

where n is the number of the non-functionality attributes (QoS), w_i denotes the corresponding weight of the ith QoS (Q_i) and $\sum_{i=1}^{n} w_i Q_i = 1$. Q_i^{0} denotes the global constraints given by users and Q_i is computed with the equations provided in Table 1 respectively and normalized.

Table 1 QoS Metrics for Web Service Selection

QoS Metrices for Web Service Selection		Aggregate Equation
Reliability	This metric denotes how exactly the requested web service is delivered to the service requester successfully.	$\prod_{i=1}^{n} R_i$ (a) $\prod_{i=1}^{n} R_i$ (b) $\prod_{i=1}^{n} R_i$ (c) max (R_i) (d)
Availability	This metric defines exactly how frequently the web service is obtainable for providing it to the service requester.	$\prod_{i=1}^{n} A_i$ (a) $\prod_{i=1}^{n} A_i$ (b) min (A_i) (c) max (A_i) (d)
Response Time	This metric denotes how much time a service provider takes to deliver an appropriate service.	$\sum_{i=1}^{n} T_i$ (a) $\sum_{i=1}^{n} T_i$ (b) max(T_i) (c) min (T_i) (d)
Cost	This metric denotes the cost required for the requested service and this is dependent on the quality of the service.	$\sum_{t=1}^{n} c_i$ (a) $\sum_{t=1}^{n} c_i$ (b) $\sum_{t=1}^{n} c_i$ (c) min (C_i) (d)

III. Animal Scavenging Behaviour Algorithm

This section describes in detail the Animal Scavenging Behaviour (ASB) which is based on the scavenging behaviour of group of animals. The ASB algorithm mainly focuses the cluster to cluster communication within the multi-cluster community. ASB is a robust, stochastic optimization algorithm based on the intelligent searching behaviour of group of animals. There are four types of members namely: Cluster Head, producers, scroungers and rangers. The proposed algorithm also follows the similar concept which is clearly explained below. This algorithm begins by generating random initial population consisting of individuals and each individual in the group is termed to be a member. For S-dimensional problems (S variables), an individual i is represented as $X_i = (x_1, x_2, x_n)$. PopSize N denotes the total number of individuals and *m* denotes the number of clusters. Cluster size is formulated using N / *m*. Cluster population have been chosen based on nearest neighbour method.

In each cluster compute the fitness of the individuals and rank the individuals in descending order of their fitness. Select the best individuals in each cluster to be the cluster heads. Select the global authoritative cluster head as the Producer. The producer constantly looks and finds the resources and the scroungers just join the producer. During iterations, the member that is found to have the best fitness value is chosen as the producer. The producer scans the environment to look for its resources. Scanning is a vital factor of search orientation. In the algorithm, at the *k*th iteration the producer X*p* behaves as follows. The producer will first scan at zero degree and then choose three random points i).A point at zero degree ii) A point crosswise at the right hand side of the producer iii)A point crosswise at the left hand side of the producer using

$$X_z = X_p{}^k + r_1\, l_{max} D_p{}^k(\phi^k) \quad (3)$$

$$X_r = X_p{}^k + r_1\, l_{max} D_p{}^k(\phi^k + r_2\, \theta_{max}/2) \quad (4)$$

$$X_l = X_p{}^k + r_1\, l_{max} D_p{}^k(\phi^k - r_2\, \theta_{max}/2) \quad (5)$$

where r1 is where $r1 \in R1$ is a normally distributed random number with mean 0 and standard deviation 1 and $\mathbf{r}2 \in Rn-1$ is a uniformly distributed random sequence in the range *(*0, 1*)* and is the maximum pursuit angle and ∈R1 and maximum pursuit distance ∈R1. If the producer finds a better position than its current position then it will move to that point otherwise it will stay in its current position and turn its head angle using the formula

$$\phi^{k+1} = \phi^k + r_2\, \alpha_{max} \quad (6)$$

In case the producer cannot find a better position after *a* iterations, then it will turn its head back to zero degree (9) where $a \in R1$is a constant.

$$\phi^{k+a} = \phi^k \quad (7)$$

Rest 80% of individuals forms the scroungers and remaining members form rangers. Scroungers try to find the resource by moving towards its pertinent cluster head using

$$x_i{}^{k+1} = x_i{}^k + r_3 \;.\; (x_p{}^k - x_i{}^k) \quad (8)$$

Rangers wander in the search space to search for the resource. This clustered approach diversifies the searching process and improves the convergence speed**.**

$$l_i{}^{k+1} = a\,.\,r_1\, l_{max} \quad (9)$$

$$x_i{}^{k+1} = x_i{}^k + l_i\, D_i{}^k(\phi^{k+1}) \quad (10)$$

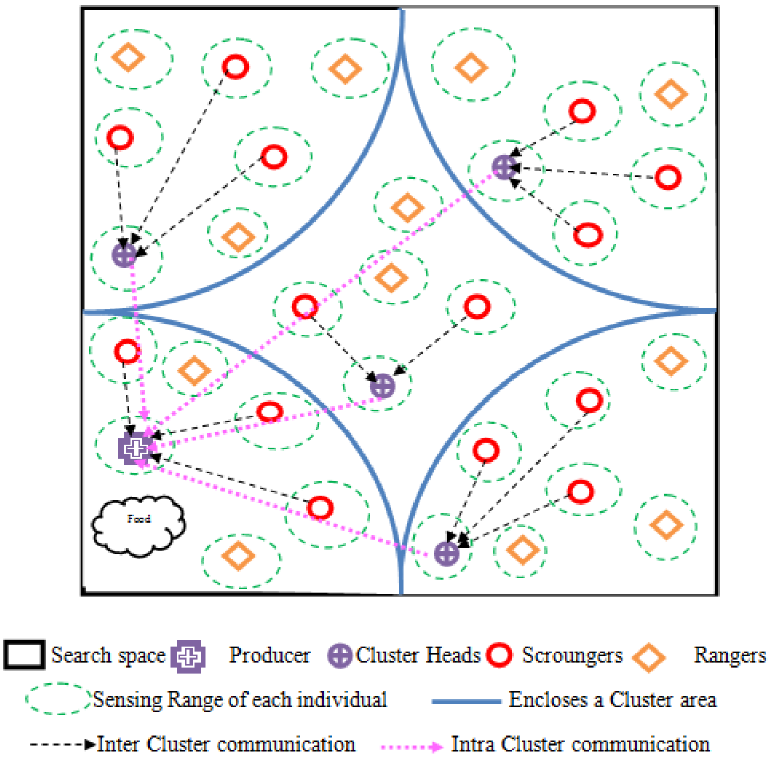

Fig.2 Animal Scavenging Behavior

At the *k*th iteration, it generates a random head angle ϕi ; and then it chooses a random distance (9) and move to the new point (10). Even in case a cluster encounters the problem of getting struck at local optima the other clusters actively participate in the foraging activity thus providing a solution to the problem and thereby enhancing diversification in the search space. Random walks, which are thought to be the most efficient searching method for randomly distributed resources are employed by the rangers. The proposed ASB algorithm will employ the multi-cluster co-operative scheme that provides inter-group communication that significantly improves the convergence performance leading to the global optimum. There exists inter and intra zone communication between the clusters to prove the efficiency of the algorithm. The pseudocode of ASB is shown in Fig.3.

Procedure ASB

Step 1: Randomly initialize population P with positions X_i and viewing angle φ_i of each member;

Step 2: $\forall$ X_i in P compute sensing potential using (2);

Step 3: Split the population into 'm' clusters (C_i) ;

Step 4: $\forall$ (X_i) in C_i, Calculate the fitness;

i. Elect some authoritative members based on fitness as Cluster Heads;
ii. Scroungers are allocated to their respective Cluster Heads;
iii. Remaining members that are dispersed in the search space become Rangers;

Select the global authoritative Cluster Head as Producer X_l ;

Step 5: For i=1 to P do:

i. Producer tries to move to better position using (3)- (5);
ii. Move the Scroungers of each cluster towards their pertinent Cluster Head using (6) – (7);
iii. Move the Cluster Heads of each cluster towards the Producer and move the Scroungers of each cluster as the same as their Cluster Head using (8);
iv. Rangers perform random walks in search space to look for the resources using (9)-(10);
v. Compute the fitness of the current individuals;

End

Step 6: Repeat until Termination Condition is satisfied;

Fig. 3 Pseudo Code of ASB Algorithm

IV. MAPPING OF ANIMAL SCAVENGING BEHAVIOUR FOR OPTIMAL WEB SERVICE SELECTION

Animal Scavenging Behaviour (ASB) a nature inspired stochastic optimization algorithm can efficiently solve the optimization and NP hard complex problems. Hence it is a convincing way to apply ASB algorithm to solve the NP hard web service selection (WSS) problem. ASB is capable of endlessly adjusting to fluctuations in the environment when searching for global optimal solutions. Therefore, we have applied ASB algorithm to web service selection because the parameters are real numbers. To get a wide range of possible solutions, this algorithm first chooses the initial population randomly and then divides the service set into service groups (cluster formation). ASB algorithm performs the local and global search. In local search, it satisfies the QoS factors such as availability, cost, response time and reliability by applying the fitness function. The randomly selected services are evaluated and sorted in the form of descending order. After a number of iterations of local search, the evolution is carried out in order to determine the global optimal service. Improve the non-qualified services based on the fitness function. The local search continues until convergence to an optimum service is reached that satisfies the user's constraints. The mapping of ASB to service selection problem is clearly elucidated below:

A. *Selection of Random population of Services*

From the set of web services initial population of services are generated by randomly selecting some services and then splitting the service set into various service groups (clusters). Each service group may contain the related and different service functionalities.

B. *Fitness Evaluation using Formulated Fitness Function*

The fitness function is defined in such a way that, it needs to maximize some QoS attribute such as reliability and availability, while minimizing other QoS attributes such as response time and cost. Hence the fitness is evaluated using equation (1).

C. ***Inter-Cluster Communication***

The limited number of iteration is carried out in the local search in order to find the optimal solution the global search is done by considering the QoS factors such as reliability, availability, response time and cost. The service which does not meet the user's request is improved.

D. ***Intra-Cluster Communication***

The evolved group of services shares the information that is obtained. The global search persists for a definite number of iterations in order to attain the global optimal service.

E. ***Termination***

If the stopping criteria is reached the algorithm can be terminated by considering the composite service with higher fitness as the near global optimal solution satisfying user constrains or otherwise, the process iterates.

```
Pseudo code : WSS using ASB
Input: n abstract tasks with its QoS values;
Generate random services of P solution (X_i individuals);
For n iterations do;
        Compute fitness of all services using formulated fitness;
        Sort the services P in descending order of fitness;
        Divide P into m service clusters;
        For each cluster do;
                Discover the best service;
                  Improve the worst service by comparing with best service;
                  Re-evaluate fitness of all services;
        End;
        Repeat for specific number of iterations;
        Combine all the evolved service zones;
        Check if termination = true;
End;
Output: Optimal composite service satisfying user constraints;
```

Fig. 4 WSS using ASB algorithm

V. RESULT ANALYSIS

A. *Experimental Setup*

The experimental settings were as follows. For the randomly generated initial population of ASB the initial head angle φ^0 of each individual is set as $\left(\frac{\pi}{4}, \ldots, \frac{\pi}{4}\right)$ in the search space. The parameters are computed using the formulae namely, maximum pursuit angle θ_{max} by $\frac{\pi}{a^2}$; constant *a* by $round\ (\sqrt{n}+1)$; maximum pursuit distance l_{max} by $\sqrt{\sum_{i=1}^{n}(U_i - L_i)^2}$ and maximum turning angle α_{max} by $\theta_{max}/2$, where *n* is the dimension of the search space; L_i and U_i are the lower and upper bounds for the *i*th dimension. Since the number of function evaluations to execute the algorithms under consideration for our experiment setting is similar to the methodology in the values are directly taken from the table V, VII, IX and XI [7].

Table 2 QoS Metrics for Web Service Selection

Function Name	Function	Category	Dim
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	Unimodal	30
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1))^2$	Unimodal	30
Step	$f_3(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	Unimodal	30
Rastrigin	$f_4(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)^2$	Multimodal High Dim	30
Ackley	$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	Multimodal High Dim	30
Griewank	$f_6(x) = \frac{1}{4000}\sum_{i=1}^{30}(x_i - 100)^2 - \prod_{i=1}^{n} cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	Multimodal High Dim	30

Branin	$f_7(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 0\left(-\frac{1}{8\pi}\right)\cos x_1 + 10$	Multimodal Low Dim	30
Six Hump Camel Back	$f_8(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Multimodal Low Dim	2
Goldstein Price	$f_9(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 +$ $- 32x_1 + 12x_1^2 +$ $48x_2 - 36x_1x_2 + 27x$	Multimodal Low Dim	2

Table 3 Comparison results of Mean and Standard Deviation of ASB, GA, PSO and GSO

Function	**Algorithms**	**Mean**	**S.D.**
Sphere	GA	3.1711	1.6621
	PSO	**3.69E-37**	2.46E-36
	GSO	1.95E-08	1.16E-08
	ASB	3.86E-14	1.26E-12
Rosenbrock	GA	338.5616	361.497
	PSO	37.3582	32.1436
	GSO	49.8359	30.1771
	ASB	**12.4789**	21.859
Step	GA	3.697	1.9517
	PSO	0.146	0.4182
	GSO	1.60E-02	0.1333
	ASB	**1.72E-18**	8.11E-17
Rastrigin	GA	**0.6509**	0.3594
	PSO	20.7863	5.94
	GSO	1.0179	0.9509
	ASB	5.9409	2.19E-08
Ackley	GA	0.8678	0.2805
	PSO	1.34E-03	4.24E-02
	GSO	2.65E-05	3.08E-05
	ASB	**3.58E-10**	5.67E-12
Griewank	GA	1.0038	6.75E-02
	PSO	0.2323	0.4434
	GSO	3.08E-02	3.09E-02
	ASB	**1.32E-12**	1.21E-11
Branin	GA	0.4040	1.0385E-02
	PSO	0.4040	6.8805E-02
	GSO	0.3979	0
	ASB	**0.2654**	7.32E-02
Six Hump Camel Back	GA	-1.0298	3.1314E-03
	PSO	-1.0160	1.2786E-02
	GSO	**-1.0316**	0
	ASB	-1.0856	3.8781E-02
Goldstein Price	GA	7.5027	10.3978
	PSO	3.0050	1.2117E-03
	GSO	3.2512	0
	ASB	**2.3114**	2.14E-01

The real coded – GA is run using the GAOT toolbox with heuristic cross over and uniform mutation. The initial population is randomly generated and its set to a size of GSO is set to 48 and 50 for the rest of the algorithms. The other parameters like mutation and crossover probability are set default. The PSO algorithm incorporated here is a standard one with acceleration constants c_1 and c_2 of 2.0 and inertia weight ω starts at 0.9 and ends at 0.4.. The designation 't_{obs}' refers to the observed value sampling distribution of t for degrees of freedom, df=1998. The cluster size '*m*' in ASB was set to

5. If the observed t equals or exceeds the critical value $\pm$ 1.96, we conclude that our result is significant beyond the .05 level of significance. The proposed algorithm discussed above has been implemented in MATLAB.

B. *Performance Comparison of ASB with Competing Algorithms on Benchmark Functions*

The algorithms are tested upon a set of bench-mark suite, which are given in Table 2. The set of benchmark functions are categorized into unimodal, multimodal, and low-dimensional multimodal functions. We compared the performance of ASB with different EAs such as i. Genetic algorithm (GA) ii. Particle swarm optimization (PSO) iii. Group search optimizer (GSO). The results obtained by ASB algorithm have been averaged for 25 runs and compared with competing algorithms GA, PSO and GSO and it is presented in Table 3. From the results obtained, it is evident that the proposed ASB algorithm is capable of finding the global optimal solutions in all benchmark problems. It is evident from Table 2, ASB performs better than all the competing algorithms for all the unimodal functions except for f1 and in the case of multimodal high dimensional problems ASB performs better than GA and PSO for all functions except f_4 .

C. *Performance Comparison of ASB with Competing Algorithms for Service Selection*

The mean and the Standard Deviation were computed for 30 independent runs and the results obtained with ASB were compared with that of GA, PSO and GSO. This test data is clearly presented in Table 4. From Table 4 we can observe MEAN best of ASB are much better than those of GA, PSO and GSO. Similarly with the increase in the number of tasks the 'Mean Best difference' between GA, PSO, GSO and ASB are becoming bigger. i.e., the differences of MEAN items between ASB and GSO are 0.88 and 2.37 for 20 and 40 abstract tasks. In the case of Standard Deviation also ASB outperforms GA, PSO and GSO proving its superiority. These results indicate that ASB algorithm proposed in this paper has powerful search ability, excellent convergence property and stability when compared with GA, PSO and GSO algorithm for web service selection.

Table 4 Comparison results of Mean and Standard Deviation of ASB, GA, PSO and GSO when 20 and 40 abstract tasks are considered

Items	**Tasks**	**20**	**40**
Mean	GA	10.4256	20.8941
	PSO	10.3378	20.5783
	GSO	10.5896	20.2176
	ASB	11.4621	22.5896
Standard Deviation	GA	0.1538	0.7524
	PSO	0.1540	0.5678
	GSO	0.1537	0.2476
	ASB	0	0.0189

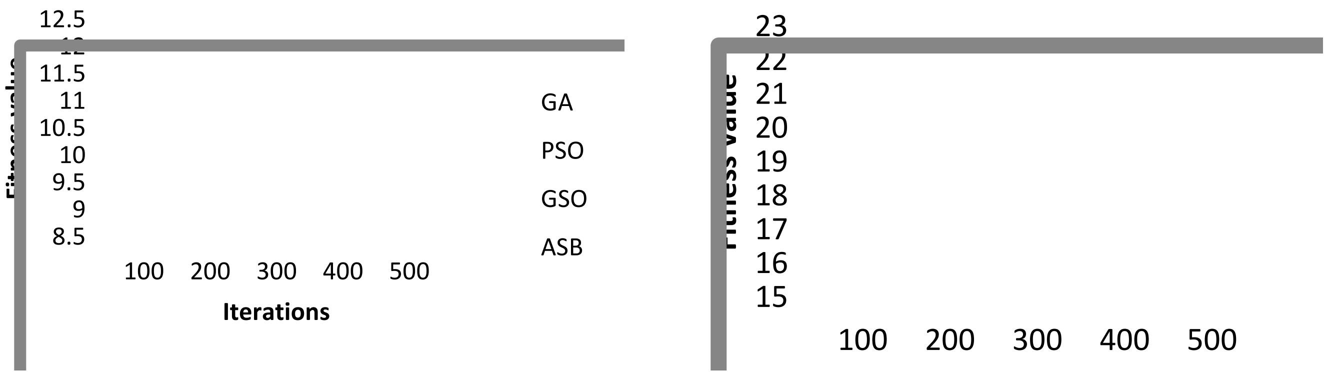

Fig.5 Comparison of ASB with competing algorithms with 20, 40 abstract tasks are considered

VI. Conclusion

In this paper, a new Animal Scavenging Behavior (ASB), inspired from the foraging behavior of animals, is introduced for solving complex optimization problems. The proposed scheme provides both local and global forms of searching which are employed by the individuals to modify their search paths. The entire search space in ASB is portioned into clusters based on the sensing potential of the individuals. The high fit members in each cluster forms the cluster head and rest of the members form scroungers and rangers. Each Scrounger selects a cluster head as its spearhead and move towards it. Cluster Heads select the global best cluster head as the Producer and adjust their positions based on their information. Thus by performing these activities both global and local strategies work in parallel. Rangers perform random walks while looking for its resources. Thus the proposed ASB algorithm has the following advantages

i. ASB mitigates the problem of getting struck at local optima.
ii. Diversified searching in the search space.
iii. Both local and global search is carried out parallel to avoid premature convergence.

To evaluate and validate the proposed method several benchmark problems both in unimodal and multimodal categories were employed. In addition, to show the applicability of ASB in solving problems it is applied to intricate service selection problem. The experimental results of the ASB algorithms indicate that this algorithm provides promising results than GA, PSO and GSO and is more convenient and suitable for solving real world complex problems.

REFERENCES

[1] Cohon, J. L., and Marks, D. H., "A Review and Evaluation of Multi-objective Programming Techniques", Journal of Water Resources, 1975.
[2] L.Davis, Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991, Chapter 6&7.
[3] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, 2008. Particle swarm optimization: Basic concepts, variants and applications in power systems, IEEE Transaction on Evolutionary Computation., 12:171–195
[4] Xiao Zheng, Jun-Zhou Luo, Ai-Bo Song, Ant Colony System Based Algorithm for QoS-Aware Web Service Selection. pp. 39-50.
[5] S. Bandyopadhyay, S. Saha, U.Maulik, K. Deb, "A simulated annealing based multi objective optimization algorithm", IEEE Trans. Evolutionary computation, vol.12, no.3, pp. 269–283, 2008.
[6] Ramin Rajabioun, "Cuckoo Optimization Algorithm ", Applied Soft Computing, , vol.11, pp. 5508–5518, 2011
[7] S. He, Q. H. Wu, and J. R. Saunders,"Group Search Optimizer: An Optimization Algorithm Inspired by Animal searching Behavior", IEEE transactions on evolutionary computation, vol. 13, no. 5, October 2009
[8] Debao Chena, Jiangtao Wanga, Feng Zoua, Weibo Houb, Chunxia Zhao, "An improved group search optimizer with operation of quantum-behaved swarm and its application", Journal of Applied Soft Computing, vol. 12, pp.712–725, 2012.
[9] T. Ray and K. M. Liew, 2003. Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Transaction on Evolutionary Computation, 7: 386–396.
[10] C.J.Barnard and R.M.Sibly, 1981. Producers and scroungers: A general model and its application to captive flocks of house sparrows, Journal of Animal Behavior, 29: 543–550.
[11] Xiao-Qin Fan,Xian-Wen Fang,Chang-Jun Jiang, Research on Web Service Selection based on cooperative evolution, An International journal on Expert Systems and Applications, vol.38, Issue.8, pp.9736-9743, 2011.
[12] AngusF.M.Huang, Ci-WeiLan, Stephen. J.H.Yang, "An optimal QoS based Web service selection scheme", Journal of information Sciences, Elsevier, vol:179, Issue19, pp. 3309-3322, September,2009.

