

# Elephant Herding Optimization for Service Selection in QoS-Aware Web Service Composition

Samia Sadouki Chibani, Abdelkamel Tari

**Abstract**—Web service composition combines available services to provide new functionality. Given the number of available services with similar functionalities and different non functional aspects (QoS), the problem of finding a QoS-optimal web service composition is considered as an optimization problem belonging to NP-hard class. Thus, an optimal solution cannot be found by exact algorithms within a reasonable time. In this paper, a meta-heuristic bio-inspired is presented to address the QoS aware web service composition; it is based on Elephant Herding Optimization (EHO) algorithm, which is inspired by the herding behavior of elephant group. EHO is characterized by a process of dividing and combining the population to sub populations (clan); this process allows the exchange of information between local searches to move toward a global optimum. However, with Applying others evolutionary algorithms the problem of early stagnancy in a local optimum cannot be avoided. Compared with PSO, the results of experimental evaluation show that our proposition significantly outperforms the existing algorithm with better performance of the fitness value and a fast convergence.

**Keywords**—Elephant herding optimization, web service composition, bio-inspired algorithms, QoS optimization.

## I. INTRODUCTION

**N**OWADAYS many companies and organizations implement their provided functionalities in internet as web services. Each service is characterized by certain functionalities and a set of non functional aspects called quality of service (QoS) attributes. A large number of web services with the same functionalities and different Quality of Service (QoS such as price, response time, availability, reliability, reputation, security, throughput etc.) can be found. New functionalities can be provided by combining available services. Hence the appearance of a new emerging challenge: QoS aware web service composition QoS-WSC [1], [2]. The goal of QoS aware service composition is to find the best combination of services such that their aggregated QoS values should be optimized. QoS-WSC is a global optimization problem belonging to NP-hard class given the number of available services with same functionalities. Thus, an optimal solution cannot be found by exact algorithm within a reasonable time. Various approaches to address this problem have been proposed in literature, Different optimization algorithms have been used: integer programming, linear programming, dynamic programming, tabu search, local search, evolutionary algorithms, and hybrid algorithms. An overview may be found in [3]. This paper describes the application of Elephant Herding Optimization (EHO) to QoS aware web service composition problem. EHO is a swarm-based metaheuristic search method, inspired

by the herding behavior of elephant group [4]. In EHO, each elephant implements clan updating operator to change its position based on its current position and matriarch position in the corresponding clan. Subsequently, the worst elephant is replaced by separating operator. For the rest of the paper, a definition of the problem and an overview of the recent research efforts are described in Section II. Section III discusses the QoS-WSC problem. In Section IV the proposed approach is presented in which an application of Elephant Herding Optimization (EHO) to QoS aware web service composition is done. Finally, experimental studies are described in Section V, and conclusions and future work are outlined in Section VI.

## II. RELATED WORK

In literature review, many approaches are proposed to address the problem of QoS-aware web services composition which is solved by using exact or approximate algorithms. Traditional optimization techniques such as integer linear programming, multidimensional multiple-choice knapsack and graph theory, have been reported in [5] to address the QoS-WSC problem. These approaches are very effective when the size of the problem is small. However, these methods suffer from poor scalability due to the exponential time complexity. To overcome these problems, approximate algorithms based on evolutionary search are proposed to find a near to optimal solution. Evolutionary Computation methods such: Genetic Algorithm (GA) [6]-[8], Ant Colony Optimization (ACO) [9]-[11] and Particle Swarm Optimization (PSO) algorithm [12], [2] algorithms are used to find optimal service composition in a reasonably short time frame when a large number of web services and multiple QoS attributes were concerned. These methods simply reduce this problem to a single-objective problem by aggregating all objectives using a fitness function (eg: the weighted sum approach and fraction-based approaches). However, with applying these algorithms, we can not escape the early stagnation in a local optimum. For the purpose of solving the QoS-WSC problem, we present a swarm based Meta-heuristic search method, called EHO [4]. EHO is inspired by the herding behavior of elephant group. The strength of EHO algorithm is its strategy of dividing and combining the population of clans which allows the exchange of information between clans (local optimum) to achieve a global optimum.

## III. QoS AWARE WEB SERVICE COMPOSITION

### A. Quality of Service

QoS is an indicator to measure and describe some performance characteristics of a service such as response time,

S. Sadouki and A. Tari are with LIMED Laboratory, University of bejaia, Algeria (e-mail: samia\_chibani@yahoo.fr, abdelkamel.tari@univ-bejaia.dz).

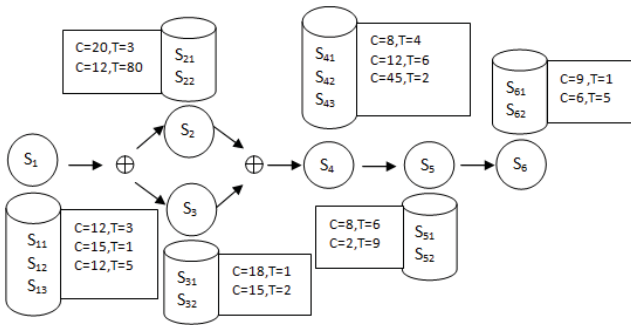


Fig. 1 Example of workflow with bindings between abstract and concrete services

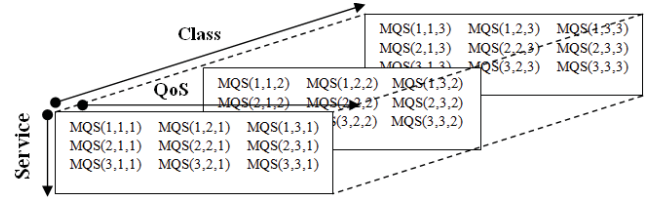


Fig. 2 Multidimensional matrix for the QoS

availability, price, reputation etc [13]. Lets  $VQS$  a set of quality attributes of the service  $S$ ,  $VQS = \{VQS(q), q = 1 : nbq\}$  where  $VQS(q)$  determines the value of the  $q$ th quality attribute of  $S$  and  $nbq$  is the number of quality attributes.

### B. Web Service Composition

Web service composition comes from software reuse. Its basic idea is combining the existing web services according to a certain relation to construct a new or better web service to satisfy a complex users requirements [14].

A composite service  $CS$  is a triple  $CS = \langle S, R, VQC \rangle$  where:

- $S = \{S_c, c = 1 : nbc\}$  Is a subset of registered abstract services, which can be composed together and satisfy the users needs.  $S_c$  may be an atomic service or a composite service. Each web service  $S_c$  has a set of concrete web services offering the same functionalities with different designed by a class  $Cc$ .
- $R = \{., !, \oplus, \otimes\}$  is a set of composite operators where  $(.)$  is a sequential operator,  $(!)$  is a loop operator,  $(\otimes)$  is a parallel operator and  $(\oplus)$  is a conditional operator. Web service composition model is specified as a workflow consisting of a set of abstract services. At run time, concrete Web service is selected, and invoked for each abstract service.
- $QoS$  of a composite service is defined as:  $VQC = \{VQC(q), q = 1 : nbq\}$  where  $VQC(q)$  determines the value of the  $q$ th quality attribute of  $CS$  and  $nbq$  is the number of quality attributes.  $VQC(q)$  for different composition operator is given in Table. I.

We implement QoS of all concrete services as a multidimensional matrix Fig. 2.

where:  $MQS(s,q,c)$  is the  $q$ th propriety of QoS value and the index of the concrete service in a service class  $c$ . A normalization of the values of the QoS attributes to the same scale is realised in order to avoid inaccurate evaluation due to different measurement metrics used for different QoS

attributes. The QoS attributes can be classified into two groups: Positive and Negative attributes.

The values of positive attributes need to be maximized (Availability, Reliability, throughput), while the negative values attributes must to be minimized (Price, Reponse time). In the normalization phase, positive and negative QoS attributes are scaled in different ways:

Positive attributes:

$$MQS'(s, q, c) = \frac{MQS(s, q, c) - Qmin(q)}{Qmax(q) - Qmin(q)} \quad (1)$$

Negative attributes:

$$MQS'(s, q, c) = \frac{Qmax(q) - MQS(s, q, c)}{Qmax(q) - Qmin(q)} \quad (2)$$

where  $Qmax(q)$ ,  $Qmin(q)$  are the maximum and minimum values of the  $q$ th attribute and  $MQS(s, q, c)$  is the value of the  $q$ th attribute for a selected candidate service  $s$  in the class  $c$ . Positive attributes can be transformed into negative attributes by multiplying their values by  $-1$ . So we will transform the negative attributes, and in this case all attributes will be minimized.

### C. Fitness Function

The fitness  $F$  is given by a traditional weighted Sum function of the different QoS values of a composition.

$$F(CS) = \sum_{q=1}^{nbq} w_k * VQC(q) \quad (3)$$

Where  $\sum_{q=1}^{nbq} w_q = 1$ ,  $w_q$  presents the users preference weight.

## IV. APPROACH DESCRIPTION

Elephant Herding Optimization (EHO) [4] is a new kind of swarm-based meta- heuristic search method proposed for solving optimization problem. EHO is inspired by the herding behavior of elephant group, Elephants are social in nature and the elephant group is composed of several clans. Elephants belonging to different clans live together under the leadership of a matriarch, male elephants remain solitary and will leave their family group while growing up. The behavior of elephant herding is modeled by clan updating and separating operators. In EHO, each elephant implements clan updating operator to change(update) its position based on its current position and matriarch position in the responding clan. Subsequently, the worst elephant is replaced by separating operator. EHO is characterized by a strategy of decomposition of population to

TABLE I  
 THE AGGREGATION OF ATTRIBUTES FOR DIFFERENT COMPOSITION STRUCTURE

Attributes	Sequential	Parallel	Conditional	Loop
Additive	$\sum_{i=1}^n VQS_i(q)$	$\sum_{i=1}^n VQS_i(q)$	$\sum_{i=1}^n VQS_i(q) * p_i$	$n * VQS_i(q)$
Multiplicative	$\prod_{i=1}^n VQS_i(q)$	$\prod_{i=1}^n VQS_i(q)$	$\sum_{i=1}^n VQS_i(q) * p_i$	$(VQS_i(q))^n$
Max-operator	$\sum_{i=1}^n VQS_i(q)$	$Max(VQS_i(q))$	$\sum_{i=1}^n VQS_i(q) * p_i$	$n * VQS_i(q)$

sub-populations (clan); this process helps the full exchange of information and benefits the algorithms global search ability. In order to make the herding behavior of elephants solve QoS-WSC problem, some rules are defined:

- 1) An Elephant represents a composition solution in QoS-WSC problem.
- 2) Elephant population is composed of some clans, and each clan has fixed number of elephants.
- 3) A fixed number of male elephants will leave their family group and live solitarily far away from the main elephant group at each generation (separating operator).
- 4) Elephants in each clan live together under the leadership of a matriarch.

The notations used in this paper are:

- $C_i$  : the  $i_{th}$  clan.
- $X_{i,j}$  : the old position for elephant  $j$  in clan  $C_i$ .
- $X_{i,j,d}$  : the  $d - th$  element of the position  $X_{i,j}$ .
- $X_{new,i,j}$ : newly updated position for elephant  $j$  in clan  $C_i$ .
- $X_{best,i}$  : the best elephant individual according to the fitness value in clan  $C_i$ .
- $X_{center,i}$  : the center of clan  $C_i$ .
- $X_{center,i,d}$  : is the  $d$ -th element of the center of clan  $C_i$ .
- $X_{worst,i}$  : the worst elephant individual according to the fitness value in clan  $C_i$ .
- $X_{max}$  : the upper bound for the position.
- $X_{min}$  : the lower bound for the position.
- $rand$  ,  $\alpha$  ,  $\beta$  and  $r$  : random numbers between 0 and 1.
- $nClan$  : the number of clan in the population.
- $N_i$ : the number of elephant in clan  $C_i$ .

The EHO Strategy is summarized in the following steps:

- Step 1:* Initialization. Set generation counter  $t = 1$ ; initialize the population  $P$ ; the maximum generation  $MaxGen$ .
- Step 2:* Sort all the elephants according to their fitness.
- Step 3:* Divide the whole elephant population into some clan according to their fitness.
- Step 4:* Implement clan updating operator for all elephants in each clan using (4); (5) and (6) for the matriarch:

$$X_{new,i,j} = X_{i,j} + \alpha * (X_{best,i} - X_{i,j}) * r \quad (4)$$

$$X_{new,i,j} = \beta * X_{center,i} \quad (5)$$

$$X_{new,i,d} = \frac{1}{N_i} * \sum_{i=1}^{N_i} X_{i,j,d} \quad (6)$$

*Step 5:* Implement separating operator to the worst elephant in each clan by (7):

$$X_{worst,i} = X_{min} + (X_{max} - X_{min} + 1) * rand \quad (7)$$

*Step 6:* Evaluate population by the newly updated positions and Sort all the elephants according to their fitness  $t = t + 1$ .

*Step 7:* Combine all clans into one population.

*Step 8:* If  $t = MaxGen$  return  $X_{1,1}$  the best solution, else go to *Step 2*.

First, an initial population of  $P$  elephants (composition solutions) is generated randomly. Elephants are sorted according to their fitness value, then, the entire population is divided into sub-populations ( $nClan$ ) such that each clan  $C_i$  containing  $N_i$  elephants. Second, within each clan  $C_i$ , an evolution process is applied to improve all elephant in  $C_i$  with clan updating operator, a separating operator is implemented to replace only the worst elephant in each clan  $C_i$ . After the local searching for all clan, all elephants in all clan are combined (mixed) and reordered. Once the termination condition is reached, the position of  $P(1)$  is returned as optimal a composition, otherwise the algorithm reiterates until Giter.

## V. EXPERIMENTATION

In this section, we report the results of simulation experiments used to study the performance of the proposed solution (EHO) in comparison with PSO algorithm applied for QoS-WSC. The algorithms are implemented with Matlab R2012 on a HP dc7900 machine with 2 Intel Duo 2.4 GHZ processors and 2 GB RAM.

### A. Dataset and Test Case Generation

The analysis is performed on several test case. A test case is defined by an abstract workflow along with a certain number of alternative concrete services defined for each abstract service. The test cases are defined by varying the number of abstract services in the workflow from 10 to 100. The number of concrete services varies from 100 to 1000 and three QoS constraints are considered. The QoS attributes of these concrete services are collected from real web services. A public database [15] containing 2507 records characterizing real web services is used as data source (random services are selected from this database). For each test case with a different number of abstract services or concrete services, each algorithm is executed 50 times to evaluate the algorithms performance. For the QoS, three objectives are considered: Minimize the Response time, Maximize the Availability and Reliability.

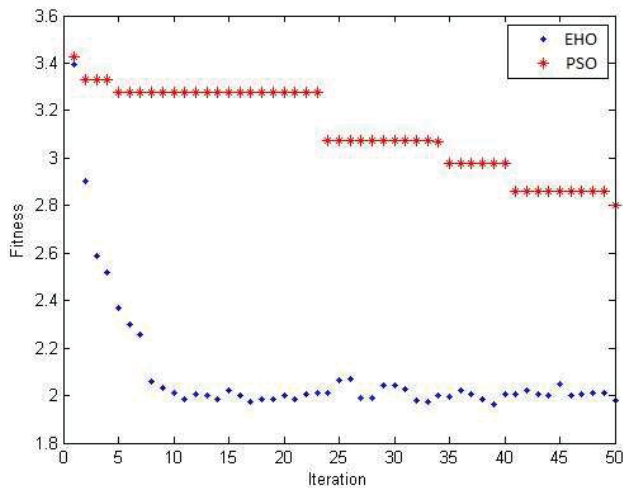


Fig. 3 Fitness over 50 iterations

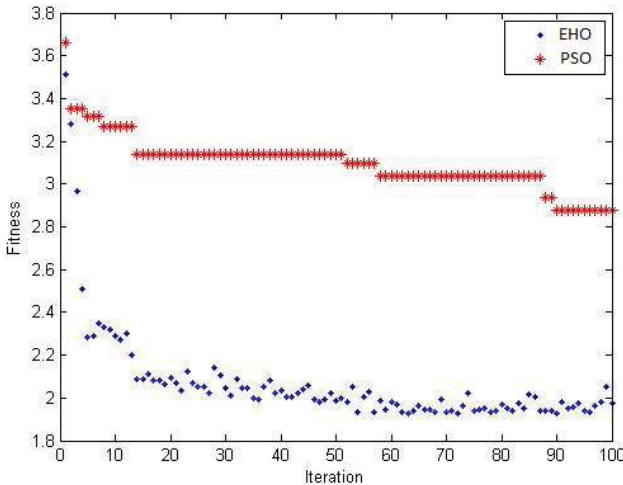


Fig. 4 Fitness over 100 iterations

### B. Metrics and Result Analysis

This section presents a set of comparative experiments between the proposed algorithm EHO and PSO. In this paper, we use first the Graphical representation of fitness of the solution over the iterations (identified solutions) than the fitness evaluation with increasing the number of abstract services and Concrete services.

1) *Graphical Representation of Fitness:* Figs. 3 and 4 show the Fitness value of the solution over 50 and 100 iterations, the problem scenario implements 30 abstract services and 500 concrete services per abstract service. These figures show that the EHO algorithm a fast convergence compared with PSO one, with best fitness value.

2) *Fitness Evaluation:* In our case, we tend to minimize the fitness value. We can see the fitness comparison when the number of concrete services varies from 100 to 1000 in Fig. 5, number abstract services ranges from 10 to 100 in Fig. 6. These figures show that the fitness values achieved by EHO algorithm are best comparing with the results obtained by PSO. The previous experiments show that EHO algorithm can identify a composite service with an optimal fitness value with

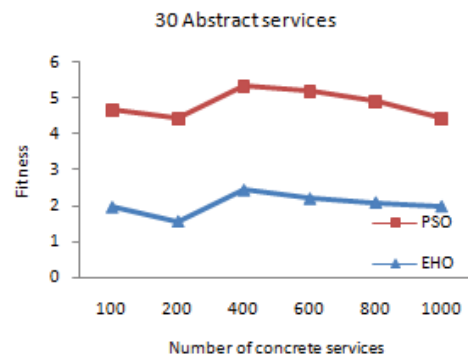


Fig. 5 Fitness evaluation according to number of concrete services

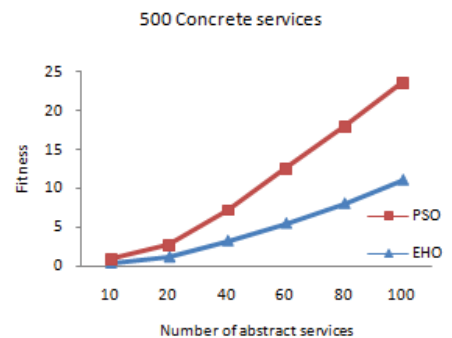


Fig. 6 Fitness evaluation according to number of abstract services

a fast convergence compared with PSO. Thus, our approach offers a more efficient and scalable solution for the service selection problem and is more suitable for a selection problem with a high complexity.

## VI. CONCLUSION

This paper presents an application of Elephant herding optimization algorithm for addressing the QoS aware web service composition, which is inspired by the herding behavior of elephant group. Compared with PSO, EHO, for a large dataset, it is shown to be an efficient algorithm for solving the QoS aware service web selection for composition problems. It offers excellent performances in terms convergence speed, scalability and fitness evaluations, which measures the complexity of the algorithm. Future work includes applying the proposed solution with Pareto approach (Multi-objective strategy).

## REFERENCES

- [1] L.Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. *Qos-aware middleware for web services composition*. IEEE Trans. Softw. Eng., 2004.
- [2] R. Marler and J. Arora. *Survey of multi-objective optimization methods for engineering*. International journal of Struct. Multidiscipl. Optim., 2004.
- [3] A. Strunk, QoS-aware service composition: a survey, in: Proceedings of the 2010 Eighth IEEE European Conference on Web Services, ECOWS 10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 6774.
- [4] G.-G. Wang, S. Deb, and L. Coelho. *Elephant herding optimization*. In 3rd International Symposium on Computational and Business Intelligence, 2015.
- [5] S. Yulu and C. Xi. *A survey on qos-aware web service composition*. In Third International Conference on Multimedia Information Networking and Security (MINES), 2011.

- [6] G. Canfora, M. Penta, R. Esposito, and M. L. Villani. *An approach for qos-aware service composition based on genetic algorithms*. In conference on Genetic and eEvolutionary computation GECCO 05, 2005.
- [7] M. C. Jaeger and G. Muehl. *Qos-based selection of services: The implementation of a genetic algorithm*. In Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference, 2007.
- [8] W.C. Chang, C.S. Wu, and C. Chang. *Optimizing dynamic web service component composition by using evolutionary algorithms*. In IEEE International Conference on Web Intelligence, 2005.
- [9] S.R.Dhore and M. Kharat. *Qos based web services composition using ant colony optimization: mobile agent approach*. International Journal of Advanced Research in Computer and Communication Engineering, 2012.
- [10] Q. Wu and Q. Zhu. *Transactional and qos-aware dynamic service composition based on ant colony optimization*. Future Generation Comp. Syst, 2013.
- [11] J. Gatha Jayjit and V. Gohel Piyush. *Optimization with agent based approach*. International Journal of Emerging Technologies and Innovative Research, 2015.
- [12] W. Wang, Q. Sun, X. Zhao, and F. Yang. *An improved particle swarm optimization algorithm for qos -aware web service selection in service oriented communication*. International Journal of Computational Intelligence Systems, 2010.
- [13] S. Kalepu, S. Krishnaswamy, and S. W. Loke. *Verity: A qos metric for selecting web services and providers*. In Proceedings of the Fourth International Conference on Web Information Systems Engineering Workshops (WISEW03), 2004.
- [14] A. L. Lemos, F. Daniel, and B. Benatallah. *Web service composition: A survey of techniques and tools*, acm computing surveys. ACM Computing Surveys, 2015.
- [15] E. Al-Masri and Q.H. Mahmoud, *Qos-based discovery and ranking of web services*. In Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN, 2007.