International Journal of Engineering &Technology, 7 (4) (2018) 6109-6114

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.23127
Research paper

Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems

Saman M. Almufti [1] *, Renas R. Asaad [1], Baraa W. Salim [2]

[1] *College of Computer Science & Information Technology, Nawroz University Duhok-kurdistan Region, Iraq*
[2] *College of Engineering, Nawroz University Duhok-kurdistan Region, Iraq*
**Corresponding author E-mail: Saman.Almofty@gmail.com*

Abstract

Elephant Herding optimization algorithm (EHO) is a metaheuristic swarm based search algorithm, which is used to solve various optimization problems. EHO can be used to solve as benchmark problems, Services Selection in QoS-Aware Web Service Compositions, Energy-Based Localization, PID controller tuning, Appliance Scheduling in Smart Grid identification and other problems. The algorithm is deducted from the behavior of elephant groups in the wild. Were elephants live in a clan with a leader matriarch (Female elephant), while the male elephants separate from the group when they reach adulthood. This is used in the algorithm in two parts. First, the clan updating mechanism. Second, the separation mechanism.
In this paper, a review of the Elephant Herding optimization algorithm is presented. Moreover, a comparison of results of EHO compared to other optimization algorithm is presented based on previous work results. In the experimental results section, the result of EHO will be compared with the U-Turning Ant Colony Optimization Algorithm (U-TACO) in solving Traveling Salesman Problem (TSP), which is based on Ant Colony Optimization (ACO).

Keywords: *Metaheuristic; Elephant Herding Optimization (EHO); Optimization; Evolutionary Algorithms; Swarm Intelligences.*

1. Introduction

Generally the complexity of the optimizations problems have been increased in a manner that it becomes difficult for the traditional mathematical programming methods to solve and optimize them. Most of the real-life optimizations problems are nonlinear, complex, multimodal, and they have a incompatible objectives functions in which the process of obtaining an optimal or even near-optimal solutions becomes a difficult task, even for a single easy and linear objective functions, sometimes, an optimal solutions may not exists at all, generally, there is no guarantee of getting an optimal solution for optimization problems [1].
In the past few years, Nature-inspired Metaheuristics Optimization Algorithms become a very active area of researches and one of the most well known high-level procedure designed for generating, selecting or finding a heuristic that optimize solutions and provide a sufficiently better, improved and fittest solutions to a given objective functions for a real-life optimization problem [2], [3].
Swarm Intelligence consists of a collection of nature-inspired metaheuristics algorithms used the behaviors of real living animal groups such as a group of bats, school of fish, swarm of bees and a bunch of worms for solving and improving Optimization problem solutions [2 - 5].
Elephant Herding Optimizations (EHO) algorithm is a metaheuristic method inspired from the natural herding behaviors of real elephant in their clans [6]. EHO is designed for solving and optimizing the optimizations problems by considering two herding behaviors (Clan updating and separating) [6], [7]. This paper present a review on EHO based on previous work results. In the experimental results of EHO has been compared with U-TACO algorithm in solving STSP.

2. Swarm intelligence (SI)

Swarm intelligence is the field of developing and designing intelligent interactive multi-agent systems that cooperate to achieve a specific goal [2], [3], [6]. Swarm intelligence has been defined by Dorigo as “The emergent collective intelligence of groups of simple agents” [8]. Generally all Swarm-based algorithms are inspired from behaviors of social living beings that live to gather in a group or colonies [2], [6].
Several optimization techniques based on SI principles have been inspired from real collective behavior systems in the nature [2], [4]including Ant Colony Optimizations (ACO) by M. Dorgo in 1992 [2 – 6], [9], Particles Swarm Optimizations (PSO) by Eberhart and Kenedy in 1995 [2], Artificial Bee Colony (ABC) by Karabaga in 2005 [6], Glowworm Swarm Optimization [4,10], Firefly Algorithm by Xin-She Yang in 2009, and other optimization algorithms [5].

3. Elephant herding optimization

Elephant herding optimizations (EHO), is an intelligent swarm based metaheuristics search method proposed by Wang at the end of 2015 [11] for solving optimization problems. The algorithm arises from modeling of herding behavior of real elephants in nature.
The herding behavior can be summarized as follows:

- Elephants swarms consists of a number of sub-groups, called clans, which are comprised of a number of Female elephants and Calves [12], as shown in Fig. 1.
- Every clan moves under the supervision (leadership) of a matriarch (female adult elephant) [12] Fig. 1.

- Male calves that reachs to adulthood, leave the clan whom they belong as shown in Fig. 2.

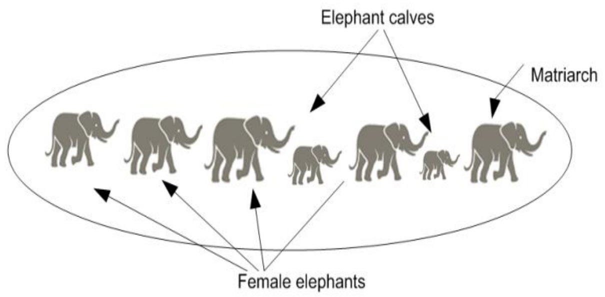

Fig. 1: Elephant Clan.

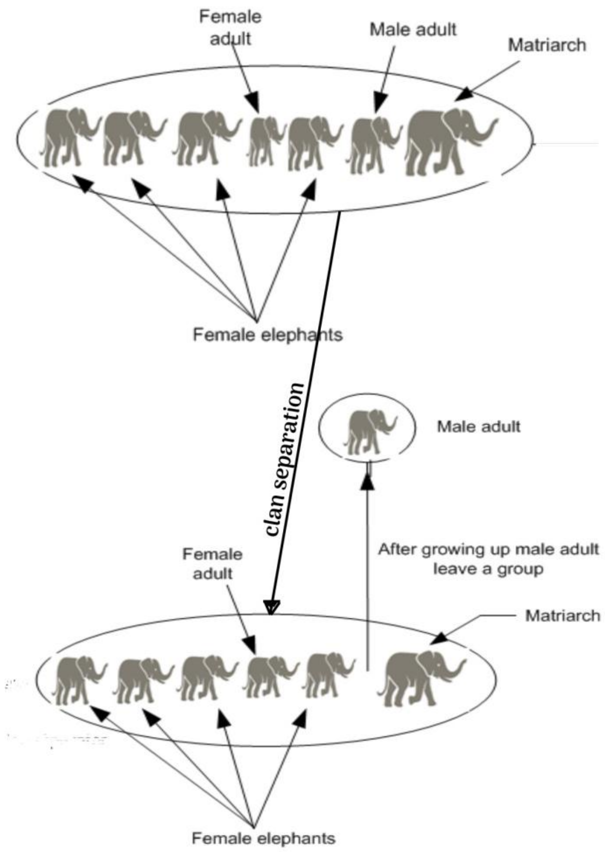

Fig. 2: Adult Male Elephant Separation.

EHO models the herding behaviors of elephants in two operations:

a) Clan update (which updates the elephants and matriarch current positions in each clan)

b) Separation (which enhances the population diversity in the next search phase).

a) Clan update:

In each clan, the female elephants lives under the leadership of a matriarch (Adult Female), and the position of other clan elephants are influenced by the matriarch positions, in a manner that in ci clan the position of j elephant is updating using Eq. (1) [11].

$$X_{new,ci,j} = X_{ci,j} + \alpha\left(X_{best\ ci} - X_{ci,j}\right) r \quad (1)$$

Where $X_{ci,j}$ denotes an old position whereas the $X_{new,ci,j}$ denotes a new updated position for j elephant in ci clan, α denotes a scale operators ∈ [0, 1] for determining the effects of matriarch ci on $X_{ci,j}$[11]. $X_{best,ci}$ denotes the matriarch of ci clan [12], and finally r is a type of stochastic distribution ∈ [0, 1] that can improves the diversity of elephant populations in the next search phase [13].

In Eq. (1) the matriarch elephant $X_{best,ci}$ in clan ci is not effected. $X_{best,ci}$ can be updated by Eq. (2).

$$X_{new,ci,j} = \beta X_{center,ci} \quad (2)$$

Where $X_{center,ci}$ denotes the center of the ci clan, generates from the information's obtained by the ci clan elephants, and β is the operator that determine the influences of the $X_{center,ci}$ on $X_{new,ci,j}$ and it ∈ [0, 1]. $X_{center,ci}$ for the d_{th} dimension can be found by Eq. (3).

$$X_{center,ci,d} = \frac{1}{n_{ci}} \sum_{j=1}^{n_{ci}} X_{ci,j,d} \quad (3)$$

Where d_{th} is a dimension between 1 and total number of dimensions $(1 \le d \le D)$. n_{ci} represent the population number in ci clan. $X_{ci,j,d}$ represents the d_{th} of elephant $X_{ci,j}$, $X_{center,ci}$ representing the center of ci clan [12].The clan update operation is shown in Algorithm 1 [11], [14].

Algorithm 1: Clan Updating Operation [11], [14]

```
Begin
  For ( Ci = 1 : nClan )
   For (j = 1 : n_ci )
   Updates (X_ci,j) & find (X_new,ci,j) Eq. (1).
   If (X_ci,j == X_best,ci)
   Updates (X_ci.,j) & find (X_new.,ci,j)Eq. (2)
   End
   End
End
```

b) Separation:

In every elephants clans, male elephant leaves the group to live alone after it reaches Adult age. In optimization problems this separating process is called separating operator .In EHO method, the adult male elephant with the worst efficiency separates at each generation using Eq. (4) [11,12].

$$X_{worest,ci} = X_{min} + (X_{max} - X_{min} + 1) * r \quad (4)$$

Where $X_{worest,ci}$ denotes the worst male elephant in the ci clan [12]. X_{min} & X_{max} denotes the Lower & Upper bounds of elephants positions. r is a type of stochastic and uniform distribution ∈ [0, 1] [11 - 13].The separating operation is shown in Algorithm 2 [12].

Algorithm 2: Separating Operator [15]

```
Begin
For (Ci = 1 to nClan)
 Exchange worest elephants individual in Ci Eq. (4).
End
End
```

Based on the Elephants herding optimizations operations that are modeled on algorithms I and II, The EHO can be developed as shown in Algorithm 3 [11].

Algorithm 3: Elephant Herding Optimization [11]

Begin

Step I: Set counter of generations $C = 1$. in the search space, randomly initialize the set of individual elephant with uniform a distribution; set the kept elephants number ($nKEL$), and the $MaxGen$, scale operators(α & β), clan number (nClan) and number of elephants for C_{ith} clan $n_{.ci}$.

Step II: Evaluate each elephant individual according to their fitness

Step III: while $C < MaxGen$

Arrange all the elephants according to its fitness.

Save $nKEL$ the elephant individuals.

Implements the clan updating by (Algorithm 1).

Implements separating by (Algorithm 2).

Evaluates the elephant populations according to the new updated positions. Replace the worst elephant individuals with the $nKEL$ saved ones.
Step IV: Update the generation counter, $C = C + 1$.
Step V: Output. Optimal solution.
End

4. EHO in solving optimization problems

Elephant herding optimization algorithm that was successfully applied and developed to solve optimization problems [12]. In this section we will discuses some optimizations problems and show the performance of EHO based on previous researches works.

4.1. Benchmark problem

Benchmark problems are a set of standard optimization problems consists of various types of functions, used in testing and evaluate the evaluation, characterization and performance measurement of optimization algorithms under different environmental conditions [15].
In 2015 EHO was used for solving Benchmark problems and the results of the simulations were compared with differential evolution Algorithm (DE), Biogeography-based optimizations Algorithm (BBO) and genetic algorithm (GA) [11]. In the simulation the author applied algorithms on fifteen basic Benchmark problems F01-F15 shown in Table 1, the composition functions were chooses from IEEE CEC 2005 [12], [15].

Table 1: Benchmark Functions [15], [19]

fn	Names	fn	Names	fn	Names
F01	Ackley	F06	Griewank	F11	Penalty
F02	Alpine	F07	Holzman	F12	Perm
F03	Brown	F08	Levy	F13	Powell
F04	Dixon & Price	F09	Pathological	F14	Quartic
F05	Fletcher-Powell	F10	Penalty	F15	Rastrigin

For four methods (EHO, BBO, DE, GA), maximum generations and population size are set to 50. EHO parameters (α, β, nClan and nci) are respectively set to (0.5, 0.1, 5 and 10) [12].
Generally, all metaheuristic algorithms depend on certain stochastic distributions, that's why various number of iterations will generate dissimilar results for any given problem [11]. In this simulation, 100 iterations were executed to obtain fittest results for given problems shown in Table 2 [12].

Table 2. Best Functions Results [11], [12]

	EHO	GA	DE	BBO
F01	1.3E-3	13.76	15.82	4.08
F02	1.8E-4	12.69	9.97	0.37
F03	2.3E-6	2.2E-16	1.63	2.2E-16
F04	0.74	9.3E5	8.7E5	4.8E4
F05	2.5E5	8.5E4	1.3E5	2.6E4
F06	1.00	10.27	13.09	3.52
F07	7.6E-14	1.6E3	475.88	60.00
F08	1.27	12.41	7.17	0.40
F09	3.20	4.07	1.46	4.34
F10	0.23	13.90	32.71	3.26
F11	1.33	2.2E5	2.6E5	118.50
F12	4.0E45	6.0E51	5.8E37	6.0E51
F13	8.8E-7	191.00	334.18	23.00
F14	8.4E-16	2.2E-16	0.18	2.2E-16
F15	1.8E-5	9.00	79.00	2.2E-16

Table 2 shows that EHO is the best-used algorithm as it finds the best solution for F01, F02, F04, F06, F07, F10, F11 and F13 [12]. The best solutions for F09 and F12 where founded by DE. BBO have finds the best solutions for F03, F05, F08, F014 and F15. GA obtained the best result for F03 and F14 [11].

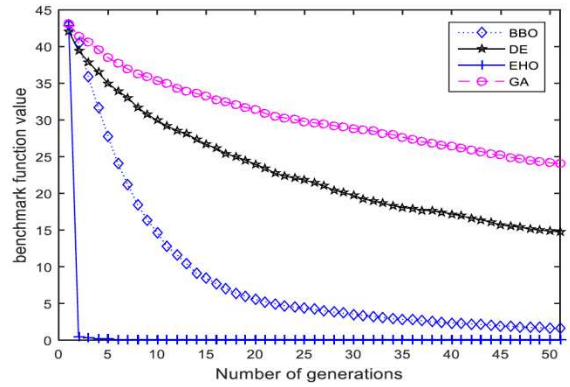

Fig. 3: F02 Alpine Function's Convergent Curves.

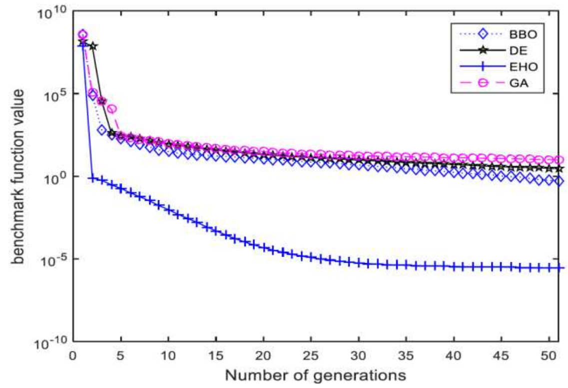

Fig. 4: F03 Brown Function's Convergent Curves.

The convergent of Benchmark F02 (Alpine) and F03 (Brown) function are shown in Fig. 3 and 4, and it's clearly shown that EHO has the best performance between the four proposed algorithms (EHO, BBO, DE and GA) in the optimization process.

4.2. Service selection in QoS-WSC

With the extremely increasing in Web services, the quality of service-aware service selection (QoS-WSC) is an active researches area on the field of Web services compositions [16]. QoS-WSC is one of the complex combinatorial optimizations problem, which requires finding an optimal composition plan that maximizes user's Quality of Service (QoS) requirement [16], it belongs to NP-hard Problems in which its almost not easy to find an optimal solution by exact algorithms in a reasonable time for a given number of available services with same functionalities [4], [17].
QoS-WSC word derived from two syllables QoS and WSC, where QoS is quality of service and WSC aware web services compositions.
1) Quality of Service (QoS):
QoS is a term used to describe and measure some characteristics related to the performance of services [18]. The aim of the selection process is to choose among services discovered according to their functionalities and performances such as Cost, Reliability, time, confidentiality, Availability, Reputation, etc. [19].
2) Web Service Composition (WSC):
Web services compositions (WSC) is the mechanism for combining and reusing web services by considering a certain relations to construct a new Web services that satisfies the complexity of users requirements [20].
In the simulation, the author applied EHO algorithm and the results were compared with the Particle Swarm Optimization (PSO) algorithm. Both EHO & PSO were experimentally tested on similar conditions [7], and performed on several test cases. Test cases were defined by an abstract workflow along with a certain number of concrete service for every abstract. Each workflow in the test cases contains $n \in [10 : 100]$ abstract services and $n \in [100 : 1000]$ concrete services. A public database contains 2507 records characterizing real web services is used as data source [21].

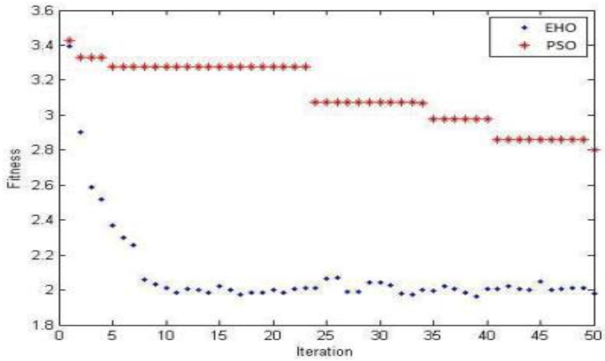

Fig. 5: Fitness Over 50 Iterations[6].

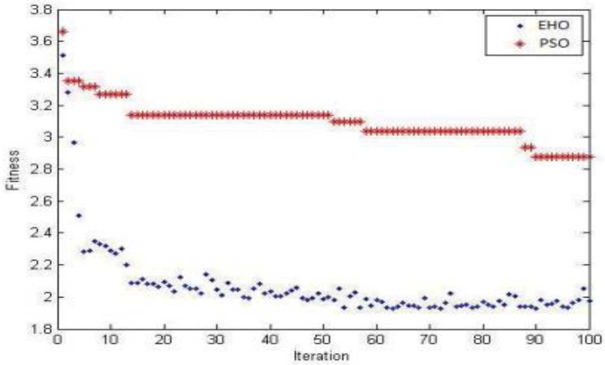

Fig. 6: Fitness Over 100 Iterations.

Fig. 5 and Fig. 6 Shows the Graphical Representations of Fitness value of the solution of algorithms over 50 and 100 iterations, 30 abstract services and 500 concrete services per abstract service [7].

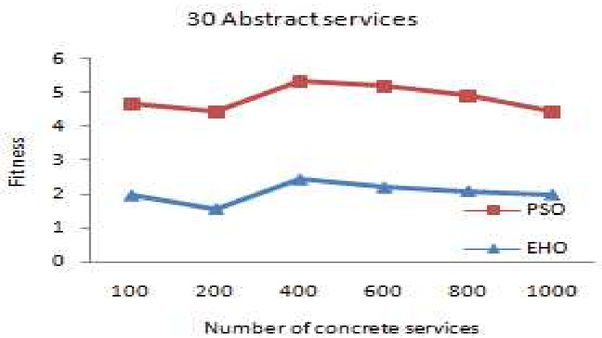

Fig. 7: Fitness Evaluation According to Number of Concrete Services[6].

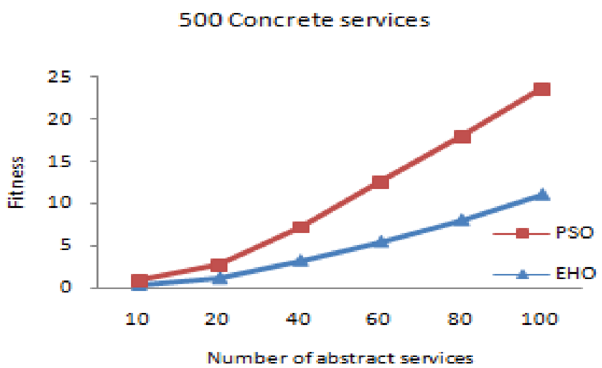

Fig. 8: Fitness Evaluation According to Number of Abstract Services[6].

Fig. 7 shows comparison of fitness between EHO and PSO for concretes services in range [100 : 1000]. Fig. 8 shows comparison of fitness between EHO and PSO for number abstract services variant from 10 to 100. It is clearly shown in Fig. 5, 6, 7 and 8 that EHO has the better performance than PSO in the optimization process.

4.3. Wireless sensor network localization problem

Wireless Sensor Network (WSN) is a distributed low-cost sensor nodes collects data within the sensing range and transmit them to the Base Station or sink. The ideal WSN should be smart and software programmable, scalable, reliable and accurate over the long term, energy efficient, requires low maintenance and low cost to purchase and install, [22].

The challenges in Wireless Sensor Network (WSN) such as localization, connectivity, node scheduling, network lifetime, data gathering node deployment, coverage, node clustering. WSN belongs to NP hard problems.

In 2018, EHO was used to solve Wireless Sensor Network Localization Problem and the results of the simulations were compared with particle swarm optimizations (PSO) algorithm, multi-stages particle swarm optimizations (MPSO) algorithm, artificial bee colony (ABC) algorithm, and multi-stages artificial bee colony (MABC) algorithm [23].

EHO parameters (α, β, $nClan$ and nci) are respectively set as: (0.5, 0.1, 5, 10), and the maximum generation $MaxGen$ set to 800, that produce 40000 objectives evaluation functions. The algorithm was executed over 30 iterations and Number of Nodes =1000.

In the MSN the objective function is the minimizations of mean squares errors (MSE) between the real node coordinates and the real and estimated distance of computed nodes coordinate as shown in Eq. (5)[24].

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(d_i - \hat{d}_i)^2 \quad (5)$$

Where di is actual distance, $\hat{d}_i$ is estimated distance, $M \geq 3$ denotes the anchors number in transmissions range R.

Table 3 gives the simulation results; it shows the fittest values for MSE, obtained by applying deferent algorithms in comparative study [24]. The best outputs are presented in bold format [24].

Table 3: Simulation Results for Best Values Obtained for MSE Objective Function [24]

An-chor	No. (%) Range	PSO	MPSO	ABC	MABC	EHO
2.5	20	28.149	27.667	45.128	40.038	27.493
5	20	27.273	27.073	43.386	38.038	26.923
7.5	20	26.156	26.035	40.925	36.374	26.019
10	20	25.273	25.035	40.079	34.193	24.978
2.5	35	27.32	27.167	43.99	39.273	27.18
5	35	26.089	25.936	41.759	35.238	25.903
7.5	35	25.169	25.005	40.279	33.273	24.971
10	35	24.763	24.573	38.462	30.283	24.591
2.5	50	26.537	25.893	41.534	35.928	25.857
5	50	25.534	25.382	40.178	33.229	25.428
7.5	50	24.81	24.673	38.289	32.037	24.668
10	50	24.485	24.109	36.972	30.028	24.097

Table 4, shows the mean values that obtained by each algorithm. The best outputs are presented in bold format [24].

Table 4: Simulation Results for Mean Values Obtained for MSE Objective Function [24]

An-chor	No. (%) Range	PSO	MPSO	ABC	MABC	EHO
2.5	20	28.149	27.667	45.128	40.038	27.582
5	20	27.273	27.073	43.386	38.038	26.983
7.5	20	26.156	26.035	40.925	36.374	26.031
10	20	25.273	25.035	40.079	34.193	25.013
2.5	35	27.320	27.167	43.990	39.273	27.180
5	35	26.089	25.936	41.759	35.238	25.927
7.5	35	25.169	25.005	40.279	33.273	25.019
10	35	24.763	24.573	38.462	30.283	24.635
2.5	50	26.537	25.893	41.534	35.928	25.872
5	50	25.534	25.382	40.178	33.229	25.443

7.5	50	24.810	24.673	38.289	32.037	24.681
10	50	24.485	24.109	36.972	30.028	24.106

It is clearly shown in tables of experimental results (Table 3 and 4) for the best and mean values that EHO is the best algorithm between the five used algorithms for solving and optimizing the Localizations Problems in Wireless Sensors Network (WSN).

4.4. Traveling salesman problem (TSP)

Traveling Salesman Problem (TSP) is a NP-hard problem widely used in testing and evaluating optimizations algorithms [4,6]. In a given graph, a complete TSP tour requires visiting all cities once and return to the starting city [4,6,7]. Symmetric TSP (STSP) is a kind of TSP in which for graph G(C,A) the distance d(City i, City j) = d(City j, City i) and the number of possible tours is (n-1)!/2 for n cities [6,7]. In STSP the optimal tour can be obtained by calculating the summation of the length between cities as shown in Eq. (6) [4].

$$Btour = \left(\sum_{i=1}^{n-1} d_{\pi(i)\,\pi(i+1)}\right) + d_{\pi(n)\,\pi(1)} \qquad (6)$$

Where p is a probability list of cities with minimum distance between city (pi and pi+1) [6].
In 2019 EHO has been used to solve STSP problems [25], The EHO was designed for solving continued optimization problems [15], so it cannot be directly applied to TSP which is a combinatorial optimization problems, because in the continued optimization problems the solutions represented by a single value whereas in the case of combinatorial optimization problems the results represented in a victor [25].
The Discrete Cuttlefish Optimization algorithm (DCOA) was used to solve STSP problems [26]. Table 5 shows a comparison between EHO and DCOA best results in solving STSP. Table 5 shows a comparison between EHO and DCOA in solving STSP.

Table 5: Simulation Results for Best Values Obtained for STSP Problems Using EHO and DCOA [26]

STSP problem	Optimal Tour	EHO	DCOA
***Eil*51**	426	426	426
***St*70**	675	675	675
***Eil*76**	538	538	538
***kroD*100**	21294	21294	21294
Eil101	629	630	629
Bier127	118282	118282	118313
Ch130	6110	6110	6110
***Ch*150**	**6528**	**6550**	6528

Table 5, Shows the result of STSP problems by using EHO and DCOA, it shows that both algorithms have an extremely close result, and both algorithms are successfully well adapted to solve the problems.

5. Experimental results

This section presents the performance of using U-Turning Ant Colony Optimization (U-TACO) to solve STSP [2], [4], U-TACO results are compared with EHO result [25]. Both algorithms tested in some instances from the TSPLIB library [27]. Table 6 shows the comparison of EHO and U-TACO algorithms.

Table 6: Simulation Results for Best Values Obtained for STSP Problems Using EHO and U-TACO [4]

STSP problem	Optimal Tour	EHO	U-TACO
Eil51	426	426	**427**
berlin52	7542	7542	**7542**
St70	675	675	**679**
Eil76	538	538	**538**
Pr76	108159	108159	**108487**
kroA100	21282	21282	**21282**
kroB100	22141	22141	**22200**
kroC100	20749	20749	**20798**
kroD100	21294	21294	**21455**
Eil101	629	630	**629**
Lin105	14379	14379	**14379**
Pr107	44303	44303	**44303**
Pr124	59030	59030	**59030**
Pr136	96772	96772	**100702**
Ch150	6528	6550	**6547**
kroA150	26524	26524	**26618**
kroB150	**26130**	**26130**	**26242**

The result table shows that the EHO algorithms is better than U-TACO in finding optimal solutions for STSP.

6. Conclusion

This paper presents a review on one of swarm inspired algorithms known as Elephant herding optimizations (EHO) to deal with global optimizations missions. by idealizing the elephant behavior in nature through “updating & separating” operation. EHO has no complicated operators, which makes its implementation easy and fast. At the early steps of EHO method, the clan updating and separation operations take place, Subsequently, the worst elephant individual is replaced by randomly generated elephant individual, and this can significantly optimize the solutions for different optimizations problem (Wireless Sensor Network Localization Problem, QoS aware web service composition, Benchmark problem, and Traveling Salesman Problem) and the results of simulations are compared with other optimizations algorithms.
Based on the Tables and Figures of results that were obtained by EHO in solving various optimizations problems, We conclude that EHO has a good characteristics as optimization algorithm and it can be used for solving complex optimizations problems.
This paper has experimentally been shown that the adapted EHO method can find much better solutions on most STSP problems compared with U-Turning Ant Colony Optimization (U-TACO).
Despite various advantages of the EHO method, EHO has some limitations:
Firstly, in our present work, little effort is spent on fine-tuning the parameters used in EHO method. Parametric study should be focused in our future work.
Secondly, the herding behavior of elephants in nature is more complex than clan updating and separating operators. More behavior of elephants should be modelled and incorporated into EHO method.
Thirdly, EHO was designed for solving continued optimization problems, so it cannot be directly applied to combinatorial optimization problems.

References

[1] X. Yang, “Metaheuristic Optimization” Scholarpedia, 6(8), p.11472, 2011. https://doi.org/10.4249/scholarpedia.11472.

[2] S. Almufti, "U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem", Hdl.handle.net, 2018. [Online].

[3] S. Almufti, “Using Swarm Intelligence for solving NPHard Problems,” Academic Journal of Nawroz University, vol. 6, no. 3, pp. 46–50, 2017. https://doi.org/10.25007/ajnu.v6n3a78.

[4] S. Almufti and A. Shaban, "U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem", Academic Journal of Nawroz University, vol. 7, no. 4, pp. 45-49, 2018. https://doi.org/10.25007/ajnu.v6n4a270.

[5] J. Rajpurohit, T. Sharma and A. Abraham, "Glossary of Metaheuristic Algorithms", International Journal of Computer Information Systems and Industrial Management Applications, vol. 9, pp. 181-205, 2017.

[6] R. Asaad and N. Abdulnabi, "Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems", Academic Journal of Nawroz University, vol. 7, no. 3, pp. 1-6, 2018. https://doi.org/10.25007/ajnu.v7n3a193.

[7] S. Chibani and A. Tari," Elephant Herding Optimization for Service Selection in QoS-Aware Web", International Journal of Computer and Information Engineering, 2017.

[8] Y. Li, "Solving TSP by an ACO-and-BOA-based Hybrid Algorithm". International Conference on Computer Application and System Modeling, pp. 189–192. IEEE Press,New York ,2010.

[9] M. Dorigo," Optimization, Learning and Natural Algorithms", PhD thesis, Politecnico di Milano, Italy, 1992.

[10] C. Kahraman and G. Kayakutlu, Energy management - collective and computational intelligence with theory and applications. Cham, Switzerland: Springer, 2018. https://doi.org/10.1007/978-3-319-75690-5.

[11] G. Wang, L. Dos Santos Coelho, X. Gao and S. Deb, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour", International Journal of Bio-Inspired Computation, vol. 8, no. 6, p. 394, 2016. https://doi.org/10.1504/IJBIC.2016.10002274.

[12] G. Wang, S. Deb and L. Coelho, "Elephant Herding Optimization", 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), 2015. https://doi.org/10.1109/ISCBI.2015.8.

[13] G. Wang, S. Deb, X. Zhao and Z. Cui, "A new monarch butterfly optimization with an improved crossover operator", Operational Research, vol. 18, no. 3, pp. 731-755, 2016. https://doi.org/10.1007/s12351-016-0251-z.

[14] L. Barolli, F. Xhafa and J. Conesa, Advances on Broad-Band Wireless Computing, Communication and Applications. New York: Springer, 2017. https://doi.org/10.1007/978-3-319-49106-6.

[15] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization", Nanyang Technological University, Singapore, 2005. https://doi.org/10.1109/TSE.2004.11.

[16] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam and H. Chang, "QoS-aware middleware for Web services composition", IEEE Transactions on Software Engineering, vol. 30, no. 5, pp. 311-327, 2004. https://doi.org/10.1007/s00158-003-0368-6.

[17] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering". International journal of Struct. Multidiscipl. Optim, 2004.

[18] S. Kalepu, S. Krishnaswamy and Seng Wai Loke, "Verity: a QoS metric for selecting web services and providers", Fourth International Conference on Web Information Systems Engineering Workshops, 2003. Proceedings. Available: 10.1109/wisew.2003.1286795.

[19] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam and Q. Sheng, "Quality Driven Web Services Composition", In Twelfth International Conference of WWW, May 20-24, Budapest, 2003. https://doi.org/10.1145/775209.775211.

[20] A. Lemos, F. Daniel and B. Benatallah, "Web Service Composition", ACM Computing Surveys, vol. 48, no. 3, pp. 1-41, 2015. https://doi.org/10.1145/2831270.

[21] E. Al-Masri and Q. Mahmoud, "QoS-based Discovery and Ranking of Web Services", 2007 16th International Conference on Computer Communications and Networks, 2007. https://doi.org/10.1109/ICCCN.2007.4317873.

[22] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", Computer Networks, vol. 38, no. 4, pp. 393-422, 2002. https://doi.org/10.1016/S1389-1286(01)00302-4.

[23] I. Strumberger, M. Beko, M. Tuba, M. Minovic and N. Bacanin, "Elephant Herding Optimization Algorithm for Wireless Sensor Network Localization Problem". IFIP Advances in Information and Communication Technology, pp.175-184, 2018. https://doi.org/10.1007/978-3-319-78574-5_17.

[24] I. Strumberger, E. Tuba, N. Bacanin, M. Beko and M. Tuba, "Monarch butterfly optimization algorithm for localization in wireless sensor networks", 2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA), 2018. https://doi.org/10.1109/RADIOELEK.2018.8376387.

[25] A. Hossam, A. Bouzidi and M. Riffi, "Elephants Herding Optimization for Solving the Travelling Salesman Problem", Advances in Intelligent Systems and Computing, pp. 122-130, 2019. https://doi.org/10.1007/978-3-030-12065-8_12.

[26] M. Riffi and M. Bouzidi, "Discrete cuttlefish optimization algorithm to solve the travelling salesman problem", 2015 Third World Conference on Complex Systems (WCCS), 2015. https://doi.org/10.1109/ICoCS.2015.7483231.

[27] G. Reinelt, "TSPLIB", Elib.zib.de, 1997. [Online]. Available: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html. [Accessed: 18- Feb- 2019].