ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2020.0309

COMPARISON OF HYBRID ELEPHANT HERDING OPTIMIZATION WITH DIFFERENT EVOLUTIONARY OPTIMIZATION ALGORITHMS

T. Mathi Murugan[1] and E. Baburaj[2]
[1]Faculty of Computer Science and Engineering, Sathyabama Institute of Science and Technology, India
[2]Department of Computer Science Engineering, Marian Engineering College, India

Abstract
Many optimization algorithms that imitate the social behaviour of animals and natural biological evolution have been proposed in the recently preceding years. These nature inspired algorithms known as evolutionary algorithms have considerably enhanced the development of the optimization process. In this paper, a hybrid elephant herding opposition algorithm is proposed and a comparative study is conducted to analyse the effectiveness of the proposed algorithm. For the purpose of the comparison, the optimization algorithms that have been taken up for the study are Refined Selfish Herd Optimization (RSHO), Spotted Hyena Optimization (SHO), Chicken Swarm Optimization (CSO) and Particle Swarm Optimization (PSO). Tests on 21 common benchmark functions have been conducted to evaluate the performance of the proposed algorithm. The results from the experiment concluded that the proposed algorithm performs better than the other algorithms.

Keywords:
Evolutionary Algorithm, Elephant Herding Optimization, Benchmark Functions

1. INTRODUCTION

The optimization algorithms which are inspired from nature are known as evolutionary algorithms. In the recent decades, many evolutionary algorithms that mimic the behaviour of animals have emerged [1]. In nature, the group of animals such as school of fish, swarm of bees and flocks of birds have certain biological behaviour. By observing their behaviour many optimization algorithms have been proposed to solve the real world problems [2]. The development of nature inspired algorithm lies in the point that it takes its sole inspiration from nature. These inspirations experienced from nature possess the capability to define and resolve complex tasks with intrinsically effective initial aspects and procedures with small or no awareness about the search space.

Numerous optimization algorithms are developed from the behaviour of some animals or insects in nature namely ant colonies, bees swarm, and so on. This is because the biological activities of birds and animals are responsible for specific roles both individually and as a group, to achieve a specific task in their daily routine or lifetime. As a result, they have attracted the attention of data analysts to resolve numerous difficulties in the science and engineering sector.

Metaheuristics are emerging approaches that are discovered through inspirations acquired from the biological behaviour of birds or insects or animals and are usually referred to as bio-inspired algorithms. For example, Ant Colony Optimization (ACO) simulates the food searching behaviour of ant colonies [3], Artificial Bee Colony algorithm mimics the cooperative behaviour of bee colonies [4], Grey Wolf Optimizer (GWO) emulates the hunting skill and social leadership of grey wolves [5]-[7], the Krill Herd technique simulates the mating behaviour of firefly insects [8] [9], Particle Swarm Optimization (PSO) mimics the biological behaviour of bird flocking and fish schooling [10], Whale Optimization Algorithm (WOA) imitates the actions of humpback whales [11], Social Spider Optimization Algorithm inspired from the nature of spiders [12], Lion Optimization Algorithm simulates the activities of lions and their co-operation characteristics [13] and so on. In mathematical designing, such a bio-inspired approach is defined as a system that obtains solution to any kind of optimization challenge. Furthermore, these are applied in different fields like data mining, machine learning, engineering design, etc.

One of the evolutionary algorithms proposed recently is the elephant herding algorithm (EHO) which mimics the herding behaviour of the elephants [14]. The elephants live in groups that are called clans consisting of males, females and calves. Here, the eldest female acts as the leader of these clans. The leader is called as the matriarch. However, after certain time, the male elephants leave the clan and they live independently. Even though they live on their own, they still stay in contact with the other elephants in the clan. The female elephants are more attached to the clan and they stay in the clan permanently. This herding behaviour of elephants is taken as the inspiration for elephant herding optimization algorithm.

In this paper, a hybrid EHO algorithm is proposed, wherein the quasi-opposition based learning is combined with the EHO algorithm for effective optimization. Quasi-opposition based learning has been proposed by Rahnamayan et al. [15], towards improving the candidate solution by contemplating present population in addition to its quasi-opposite population simultaneously.

The proposed algorithm then undergoes a comparison process with other evolutionary algorithms for evaluating the performance efficiency of the proposed algorithm. The evolutionary algorithms used in this comparative study include particle swarm optimization, refined selfish herd optimization, elephant herding optimization, spotted hyena optimization and chicken swarm optimization. Here, particle swarm optimization (PSO) simulates the social behaviour of flocks of birds or schools of fish [16]. The chicken swarm optimization (CSO) is the meta-heuristic algorithm depending on population, which mimics the behaviours of the chicken swarm [17]. The refined selfish herd algorithm (RSHO) imitates the behaviours observed from the natural interactions between a herd of prey and a pack of predators [18]. Spotted Hyena Optimizer (SHO) is inspired by the behaviour of spotted hyenas [19]. Generally, for all evolutionary algorithms, the method for the application is the same. At first the problem needs description to fit each process. After that the evolutionary algorithm is utilised to get the required optimal solution. The present study gives the comparison between these evolution based optimisation algorithms.

This comparative process reveals the strength of each of the existing algorithms and helps to identify the right algorithm for the optimization process of any given database. Each algorithm has specific advantages and weaknesses. If one algorithm uses minimum memory, less running time, and has improved final function value then that algorithm is considered as the best algorithm for the given dataset. By the comparison process the best optimization algorithm required for the database can be obtained.

But, the exact result does not occur often. Also, the results from the comparison algorithms are complicated. Yet, if the comparison process is done correctly, it delivers the best value practically. It exposes both advantages and disadvantages of the comparison algorithms. Thus, it helps us in choosing the best algorithm for the specific real world problem [20]. Many benchmarking functions are tested for the given algorithms for effective comparison between the algorithms.

2. COMPARISON ALGORITHMS

2.1 PARTICLE SWARM OPTIMIZATION

Particle swarm optimization algorithm was proposed by Kennedy and Eberhart. The algorithm mimics the social behaviour of a flock of migrating birds that were travelling to certain unknown place [21-24]. Here, the particle refers to each bird in the flock. The particle was similar to the chromosomes in genetic algorithms. But the difference in PSO algorithm was it never creates the new birds from the parent birds. Instead the birds in the flock develop their social behaviour and they move towards their destination. Substantially, it mimics the group of birds which interacts among themselves while flying. Each bird in the flock view in the certain direction and afterwards while communicating among themselves they recognise the bird which was in best location. Therefore, each bird moves towards the bird in best location as quick as possible depending on its present location. Then the bird examines the search space from its new location and this activity continues till the birds from the flock reach the desired destination. Here, the process includes both social communication among the birds and also their brilliance. Thus, the birds observe from its personal experience as well as from the experience of other birds among them. It is referred as the local search and global search in the algorithm.

In the PSO algorithm, the primary step is initialization, wherein the initial swarm of particles is created. Similar to the genetic algorithm, the solution representation is also employed in the PSO algorithm. The location and velocity of each particle are initialized randomly. Then the fitness value is calculated for each particle and it is then compared with prior best fitness value of both the particle and the entire swarm. Then the local best and global best position are updated on the suitable position. A new swarm is created by updating the velocity and position, if the stopping criterion is not encountered.

2.2 CHICKEN SWARM OPTIMIZATION

The chicken swarm optimization was proposed by observing the scavenging behaviour of chicken swarm based on the ranking order of chicken [25]. The chicken swarm contains one rooster, numerous hens and lots of chicks. While scavenging, each time the roosters get the food favourably. Every time the hens go along with the roosters for food and the chicks follows the hens during scavenging. Each chicken in the chicken swarm pursue separate laws of motions. There will be competitions among each chicken in the chicken swarm based on specific hierarchical order. The location of chicken in the chicken swarm signifies sufficient solution of the optimization problem. Here, the roosters are referred as best chicken, while the chicks were referred as the worst chicken. The remaining chicken in the chicken swarm are considered as hens. Since the roosters have the priority for food, they are among the best fitting values while the chicks have less priority for food and thus they were among the least fitting values. The state spaces of chicken and chicken swarm are explained depending on the chicken swarm optimization.

2.3 REFINED SELFISH HERD OPTIMIZATION

Refined selfish herd optimization algorithm mimics the hunting behaviour of animals. The optimization observes the communication among the group of prey and bunch of hunters. This optimization process is based on the Hamilton's Selfish Herd theory. This theory depends on the principle that when the group of prey was hunted by the bunch of predators, the prey in the edges of the group has high risk to be hunted. At the same time, the prey at the centre part of the group has high chance of survival. This implies that the chance of survival depends on the position of the individual. Thus, each prey in the group moves inwards to avoid hunting and to increase the chance of survival. The RSHO algorithm is proposed by imitating this hunting behaviour of animals which avoids the risk of being hunted.

Here, the location where the hunting takes place is taken as search space and the position of the prey and predators are taken as solution for optimization problems. The distinctive movement process relates to the behaviour and functions of individuals presents in the herd. Also, the position of each member in the herd gets updated. Besides, based on the source of the individual, the operators used for updating the position also differ. If the individual is the prey, then the movement operator also vary.

2.4 SPOTTED HYENA OPTIMIZATION

Spotted hyena optimization mimics the behaviour of spotted hyena while hunting its prey [26]. The spotted hyenas live in a clan and among them the females are dominant. Here, the males in the clan leave the group when they become adults and then move to another clan. In the new clan, they are the least ranking member when it comes for the meal. Generally, the spotted hyenas live and hunt in groups. The spotted hyenas depend on the huge bunch of trusted friends for hunting. And for increasing their network they link with the other spotted hyenas and their clan. The other spotted hyenas may be the friend of a friend or they also have other kind of relationship with the members of the clan. They are helpful in the hunting process of the spotted hyenas. The spotted hyenas are very social creatures and they interact with the other members by specified signals. For recognizing their group and other entities, the spotted hyenas utilise multiple sensory actions. While making social decisions, it could differentiate between the members of other groups and its own group, and make decisions based on the identities and rank of each member.

The spotted hyenas chase their prey by their sensory organs. For effective cooperation among the spotted hyenas and for increasing the fitness the cohesive clusters are useful. The hunting method and social relationship of spotted hyenas are mathematically developed for designing SHO algorithm and for performing optimization.

3. PROPOSED ALGORITHMS

3.1 ELEPHANT HERDING OPTIMIZATION

Elephant herding optimization process mimics the herding behaviour of elephants [27] - [31]. Elephants are one of the largest animals in the land. Elephants usually use their trunk to grab objects and to draw water. Generally, elephants are social animals and they live in clans. The members of the clan are males, females, and calves. In the clan, the eldest female member acts as the leader of the clan and it is called the matriarch. There are many clans in the elephant group under the leadership of a matriarch. In the clan, the females usually like to live in groups while the males prefer to live alone. Although the males live alone, they still remain in contact with the other members of the clan by low frequency vibrations. Due to the changes that happen in the clan, the elephants undergo two different phases. Initially, the male elephants live under the leadership of the matriarch and this is called the updating phase. Then, the male elephants separate themselves from the clan to live alone, and this is called the separating phase. This herding pattern is captured and applied in the implementation of EHO algorithm, which, thus, involves an updating phase and a separating phase in its optimization process.

3.2 QUASI-OPPOSITION BASED LEARNING

Opposition based learning was proposed by Tizhooosh [32] for enhancing the candidate solution which considers both the current and opposite population simultaneously. The enhanced form of opposition based learning is the quasi- oppositional based learning [33] - [40] in which the candidate solution is improved by contemplating current and quasi-opposite population simultaneously. The quasi-opposite solution is examined continuously for enhancing the process with a suitable solution. Then the suitable solution is taken as the initial solution. The method initiates with two predictions which are closer. The same process is carried out for every solution in the present population. When compared to the random number and opposite number to the solution, the quasi-opposite direction is mostly closer.

In this paper, the quasi-opposition based learning process is utilised for population initialization.

3.3 HYBRID ELEPHANT HERDING OPTIMIZATION ALGORITHM

Elephant herding optimization, which mimics the herding behaviour of elephants was proposed by Wang et al. [41]. The EHO algorithm does not fit in the local minimum easily and has lesser number of control parameters. Although it has many advantages, it has a disadvantage in stochastic initialisation. To overcome this disadvantage, the quasi opposition based learning (QBL) is combined with elephant herding algorithm for effective optimization. By utilising quasi opposition based learning, the initialisation process is effectively carried out in the optimization.

3.3.1 Initialisation Phase:

The suitable initial candidate solutions can be obtained by using opposite points. Also, these opposite populations (OP) can be achieved without any previous understanding about the solutions. The following equations describe the quasi-opposite population initialisation. Here, y is taken as any real number between (a,b) and the quasi-opposite point y_i^{qo} is described as:

$$y_i^{qo} = rand\left(T_i, y_i^o\right) \tag{1}$$

where

$$T_i = \frac{a_j + b_j}{2},\ y_i^o \in \{1,2,3....,D\}.$$

For i=1:P (size of population)

For j=1:N (number of variables)

$OP_{i,j} = a_j + b_j - S_{i,j}$;

$$T_j = \frac{a_j + b_j}{2}$$

$$y_{i,j}^{qo} = rand\left(c_j, OP_{i,j}\right)$$

End

End

3.3.2 Updating Phase:

On the updating phase all the elephants live in the same clan and the matriarch will be the leader of the clan. The position updation of an elephant e in clan cl_i is mentioned in Eq.(1).

$$X_{new,cidx,j} = X_{cidx,j} + \delta \times (X_{best,cidx} - X_{cidx,j}) \times q \tag{2}$$

where $X_{new,cidx,j}$ and $X_{cidx,j}$ are respectively the new and old position of elephant j in clan $cidx$. $\delta \in [0,1]$ is a scale factor. $X_{best,cidx}$ denotes the best position in clan $cidx$. $q \in [0,1]$ is normally dispersed random value. However, the position of a matriarch is not updated with above Equation but for the fittest matriarch, it can be updated by following Eq.(3) and (4).

$$X_{new,cidx,j} = \varnothing \times X_{center,cidx} \tag{3}$$

$$X_{center,cidx} = \frac{1}{n_{cidx}} \times \sum_{j=1}^{n_{cidx}} X_{cidx,j} \tag{4}$$

where $\varnothing \in [0,1]$ is a scale factor. $X_{center,cidx}$ depicts the centre position of a clan $cidx$. n_{cidx} represents the total number of elephants in clan $cidx$. It is clear that the position of matriarch is updated with respect to the position of all other members in the same clan.

3.3.3 Separating Phase:

On separation process, the male elephant gets out of the herd and lives alone. This process is assumed to be the replacement of the worst elephant from each clan in EHO algorithm. This is described in Eq.(5).

$$X_{worst,cidx} = X_{low} + (X_{high} - X_{low} + 1) \times q \tag{5}$$

where $X_{worst,cidx}$ defines the worst elephant in a clan $cidx$. The higher and lower limits of the position of elephant are X_{high} and X_{low} respectively. $q \in [0,1]$ was a normally distributed random number. Algorithm 1 gives the pseudocode for the EHO algorithm.

Algorithm 1: EHO Algorithm Pseudo Code

Step 1: Initial Parameter Setting

Initialize (Generation limit count, total population size, function evaluations)

Initialize the population

Calculate elephant's fitness for each individual

Step 2: Sort the population from most fit to least fit

Set generation counter t=1, maximum generation *MaxFEs*

while t < *MaxFEs* do

Sort all elephants according to their fitness

Step 3: Clan Updating Operator

for *cidx*=1 to *nClan* (numClan for all clans in elephant population) do

for *j*=1 to *nci* (newclanindex for all elephants in clan cidx) do

if $X_{cidx,j} = X_{best,ci}$ then

Update $X_{cidx,j}$ (old elephant) and generate $X_{new,cidx,j}$ (new elephant) according to Eq.(2)

else

Update $X_{cidx,j}$ (old elephant) and generate $X_{new,cidx,j}$ (new elephant) according to Eq.(3)

end if

end for *j*

end for *cidx*

Step 4: Separating operator

for *cidx*=1 to *nClan* (numClan for all clans in elephant population) do

Replace the worst elephant in clan *cidx*

end for *cidx*

Step 5: Calculate fitness

Evaluate population by the newly updated positions

t=t+1 (until Maximum number of generation)

end while

Return best solutions for all clans.

4. RESULTS AND DISCUSSION

Different nature-inspired optimization techniques are presented in this research for the comparative analysis of the proposed algorithm. The main aim of this research is to analyse the efficiency of the proposed algorithm by comparing it with other optimization algorithms. The techniques are implemented in MATLAB2018a software running on a Windows8.1 operating system. It is indispensable to distinguish the performance of each algorithm to find its merits as well as demerits. For evaluating the algorithm performance, the proposed algorithm is compared with other evolutionary optimization algorithms like Refined Selfish Herding Optimization (RSHO), Spotted Hyena Optimization (SHO), Chicken Swarm Optimization (CSO), and Particle Swarm Optimization (PSO). For the sake of analysis, 21 benchmark functions are conducted on the proposed algorithm and the other comparison algorithms.

4.1 BENCHMARK FUNCTIONS AND EVALUATION MEASURES

In this study, 21 benchmark functions are utilised for analysing the efficiency of the proposed algorithm by comparing it with other evolutionary algorithms. The purpose of using these benchmark functions is to create the rational comparison among these evolutionary algorithms. Here, each benchmark function is tested for twenty trial runs. For comparing different algorithms, the best, average and worst values are utilised as the performance metrics in the experimental process. Also, the standard deviation is used for estimating the normal distribution of the results. The standard deviation analyses the variation among each data point corresponding to the mean. The success rate is determined by the number of successful runs against the total number of runs. Here the successful run is known as the best solution created by the algorithm in the encoded accuracy level of the problem. Towards determining the success rate twenty trial runs have been performed for each problem. Based on the outcome, the performance analysis of the comparative algorithms has been carried out using conditions like the success percentage, which represents the number of trials needed for attaining the required target value, the average value acquired in the trials of the solution, and the processing time required to attain the optimum target value. The processing has also been subjected to calculate the speed of evolutionary algorithms since each evolutionary cycle has different number of generations for different algorithms. The benchmark functions used in this comparison is given in Table.1.

Table.1. Benchmark functions

No	Name
F01	Ackley
F02	Alpine
F03	Dixon and Price
F04	Griewank
F05	Levy
F06	Periodic Function
F07	Perm 0, D, Beta
F08	Perm D, Beta
F09	Powell
F10	Qing
F11	Quartic
F12	Rastrigin
F13	Rosenbrock
F14	Salomon
F15	Scwphele
F16	Sphere
F17	Styblinski-tank
F18	Sum of different powers
F19	Sum Squares
F20	Trid
F21	Zakharov

For the experimental process, the efficiency of the proposed algorithm is determined on the conventional benchmark functions by comparing the proposed algorithm with other evolutionary optimization algorithms. The evolutionary algorithms used for comparison are PSO, CSO, RSHO and SHO. For testing, each function has been repeated 20 times independently and the error values among the best function values are acquired in each run and the true optimal values are recorded. The results obtained by the comparison algorithms for the benchmark functions are given in Table.2. From the results, it can be observed that the hybrid EHO outperforms other algorithms on many benchmark functions.

Table.2. Results of the benchmark functions for the algorithms

Test Function		EHO-QBL	RSHO	SHO	CSO	PSO
Ackley	Best	$6.1726e^{-166}$	$3.5136e^{-79}$	$4.0207e^{-12}$	$2.6954e^{-09}$	0.0100
	Worst	0.0733	$1.0988e^{+03}$	$2.1128e^{+03}$	$8.9464e^{+03}$	0.3964
	Average	$3.9030e^{-04}$	1.5722	5.4504	75.7858	0.0124
	Std.Dev	0.0043	34.8549	85.0834	499.6849	0.0217
Alpine	Best	$1.2437e^{-107}$	$4.5995e^{-31}$	$2.9075e^{-15}$	$2.7034e^{-06}$	1.0001
	Worst	0.1209	0.2168	3.0650	0.3607	5.5152
	Average	$7.7150e^{-04}$	0.0058	0.0117	0.1716	1.0326
	Std.Dev	0.0069	0.0183	0.1320	0.4618	0.2977
Dixon and Price	Best	$4.2633e^{-40}$	$5.5698e^{-10}$	$2.5053e^{-07}$	$1.8932e^{-04}$	0.0100
	Worst	$3.1141e^{+03}$	$7.3263e^{+03}$	$9.5618e^{+03}$	0.0020	0.3235
	Average	6.7522	20.3013	168.8719	$7.3408e^{-04}$	0.0124
	Std.Dev	139.8232	276.4036	610.2292	$4.2658e^{-04}$	0.0207
Griewank	Best	$1.6673e^{-42}$	$2.6983e^{-13}$	$2.2950e^{-05}$	0.0019	0.0999
	Worst	27.3304	54.3303	53.6576	0.0081	0.0999
	Average	0.1899	0.4011	2.8278	0.0022	0.0999
	Std.Dev	1.1012	2.3253	7.3123	$6.0202e^{-04}$	$6.1642e^{-07}$
Levy	Best	$2.4416e^{-10}$	0.0118	0.0990	8.1107	28.6567
	Worst	$2.9999e^{+06}$	$1.8069e^{+07}$	0.0990	$1.8045e^{+07}$	29.4907
	Average	$3.0200e^{+03}$	$9.7111e^{+04}$	0.0990	$2.1109e^{+04}$	28.6634
	Std.Dev	$9.4867e^{+04}$	$9.3615e^{+05}$	$1.2496e^{-15}$	$5.7504e^{+05}$	0.0402
Periodic Function	Best	$3.5858e^{-13}$	$2.3938e^{-09}$	$1.4008e^{-04}$	7.2030	26.0101
	Worst	$2.6234e^{+03}$	$8.7643e^{+03}$	0.1217	7.4154	31.0795
	Average	6.1854	104.8044	0.0047	7.2033	26.2464
	Std.Dev	107.7660	534.3915	0.0134	0.0068	1.0438
Perm 0, D, Beta	Best	$2.5703e^{-05}$	$7.9040e^{-05}$	$7.4180e^{-04}$	0.0074	0.0090
	Worst	1.3572	0.0315	1.3555	1.9805	0.1137
	Average	0.0030	2.8183e-04	0.0051	0.0359	0.0409
	Std.Dev	0.0470	0.0017	0.0508	0.1242	0.0428
Perm D, Beta	Best	$-4.0714e^{+03}$	$-2.0458e^{+03}$	-719.5274	-2.2178	-0.3326
	Worst	$-1.3228e^{+03}$	$-1.2850e^{+03}$	-370.7664	-1.5807	-0.0749
	Average	$-3.7468e^{+03}$	$-2.0377e^{+03}$	-706.0230	-1.8953	-0.3323
	Std.Dev	467.0256	46.0918	56.7463	0.2095	0.0083
Powell	Best	$8.8816e^{-16}$	8.8818e-16	1.9835	11.9424	33.8285
	Worst	20.6931	20.2121	78.8973	61.5945	97.5459
	Average	0.3413	0.4922	4.2358	16.3957	38.1373
	Std.Dev	2.0087	1.5521	11.4582	4.6543	11.4564
Qing	Best	$8.8814e^{-16}$	$4.4409e^{-15}$	$1.5099e^{-14}$	$2.0905e^{-05}$	0.0453
	Worst	20.3505	20.5324	17.0042	16.9728	0.4111
	Average	0.3365	0.3678	0.1508	1.6855	0.0471

	Std.Dev	1.3233	1.8350	1.1352	3.0988	0.0212
Quartic	Best	$2.1404e^{-48}$	2.6186e-04	0.0345	0.4479	7.2030
	Worst	43.6674	0.0121	88.3442	82.5288	7.4370
	Average	0.2855	$4.9561e^{-04}$	0.2300	1.9249	7.2033
	Std.Dev	2.0525	0.0012	3.0198	5.7740	0.0077
Rastrigin	Best	$2.3558e^{-31}$	$6.5521e^{-11}$	$1.8530e^{-07}$	0.0010	1.3669
	Worst	$1.8593e^{+07}$	$6.9613e^{+05}$	$7.9193e^{+04}$	$5.5897e^{+08}$	1.5701
	Average	$1.8671e^{+04}$	$4.2362e^{+03}$	80.8666	$1.8035e^{+06}$	1.3676
	Std.Dev	$5.8796e^{+05}$	$5.3811e^{+04}$	$2.5047e^{+03}$	$2.6751e^{+07}$	0.0111
Rosenbrock	Best	$1.3498e^{-32}$	$9.9225e^{-11}$	$1.9815e^{-08}$	0.1037	9.8879
	Worst	8.2634	$7.3874e^{+06}$	$1.5251e^{+06}$	$1.2561e^{+09}$	11.4702
	Average	0.0953	$5.8344e^{+04}$	$3.1632e^{+03}$	$4.3360e^{+06}$	9.9063
	Std.Dev	0.7332	$5.3290e^{+05}$	$6.0439e^{+04}$	$5.9501e^{+07}$	0.1010
Salomon	Best	0.9980	0.9980	0.9980	1.9920	9.8879
	Worst	11.4269	141.1721	234.1605	39.8466	11.5214
	Average	1.0144	2.6512	1.7836	2.2702	9.8972
	Std.Dev	0.3546	8.4432	8.5103	1.8493	0.1022
Scwphele	Best	$2.6186e^{-04}$	$4.3936e^{-04}$	$5.9510e^{-04}$	$6.1400e^{-04}$	5.9698
	Worst	0.0090	0.1411	0.0534	0.0584	92.7087
	Average	$3.4940e^{-04}$	0.0024	$7.7947e^{-04}$	0.0010	9.6355
	Std.Dev	$6.3859e^{-04}$	0.0076	0.0022	0.0030	6.3001
Sphere	Best	$3.7003e^{-04}$	0.0017	1.9838	4.9748	18.9042
	Worst	0.4471	0.8936	77.7183	100.4747	87.1393
	Average	0.0011	0.0089	2.6305	9.8551	22.9808
	Std.Dev	0.0142	0.0408	4.2437	6.3532	11.3073
Styblinski-tank	Best	$1.0104e^{-14}$	$7.6755e^{-04}$	$9.2450e^{-04}$	0.3979	1.0001
	Worst	3.2700	0.0153	0.7124	2.0694	5.3786
	Average	0.0173	0.0013	0.0051	0.4094	1.2784
	Std.Dev	0.1618	0.0014	0.0237	0.0666	1.0277
Sum of different powers	Best	$1.8748e^{-06}$	$1.1415e^{-05}$	0.0021	0.0147	3.0000
	Worst	3.6748	0.0142	0.1557	25.6604	963.0417
	Average	0.1721	$1.5617e^{-04}$	0.0124	0.2236	6.0185
	Std.Dev	0.4579	$5.4573e^{-04}$	0.0243	0.1781	48.1635
Sum Squares	Best	$4.4409e^{-15}$	$2.5329e^{-09}$	$4.4873e^{-08}$	$2.6185e^{-04}$	3.0000
	Worst	20.7325	$7.3533e^{+03}$	$6.7.35e^{+06}$	0.0083	265.0918
	Average	0.3250	131.6229	$6.8.86e^{+03}$	$4.6344e^{-04}$	3.3830
	Std.Dev	1.9862	578.8782	$2.1200e^{+05}$	$8.6542e^{-04}$	8.3118
Trid	Best	$1.6391e^{-10}$	0.0025	0.0125	0.09980	9.8879
	Worst	$2.9059e^{+07}$	0.1859	$3.7235e^{+08}$	5.9314	11.1127
	Average	$6.5960e^{+04}$	0.0065	$0.6562e^{+06}$	1.0655	9.8925
	Std.Dev	$1.3003e^{+06}$	0.0216	$2.2690e^{+07}$	0.3186	0.0691
Zakharov	Best	$4.8648e^{-08}$	$1.3467e^{-04}$	1.0001	1.1509	28.6569
	Worst	3.8679	$3.1545e^{+05}$	5.7973	$4.6079e^{+06}$	29.3696
	Average	0.1890	678.4004	1.1989	$4.9264e^{+03}$	28.6657
	Std.Dev	0.5014	$1.1332e^{+04}$	0.9429	$1.4588e^{+05}$	0.0384

ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2020.0309

In this experiment, the benchmark functions utilised are uni-model function, multi-model function, hybrid function and composite function. From Table.2 we can know that the hybrid EHO algorithm performs better than the other evolutionary algorithms used for comparison. Here, the proposed algorithm got best values for all the benchmark functions. For F01, F02, F15 and F16 benchmark functions the proposed algorithm outperforms all the comparison algorithms for best, average and standard deviation values. Also, the benchmark functions F09 and F14 have the same values for hybrid EHO and RSHO coincidentally. For the benchmark functions F01, F02, F09, F13, F14, F15, F16 and F21 the average values of the hybrid EHO are better when compared to the other evolutionary algorithms.

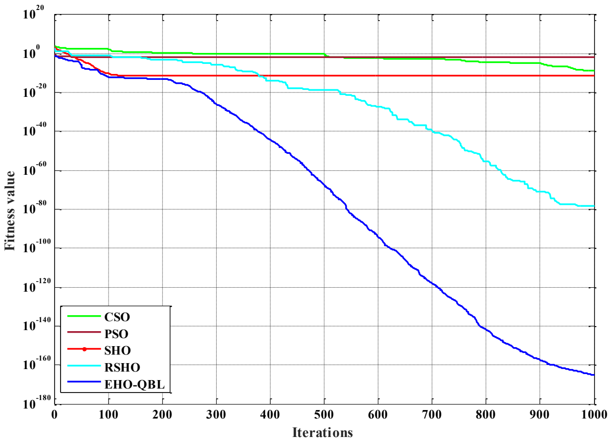

Fig.1. Convergence curve for Ackley function

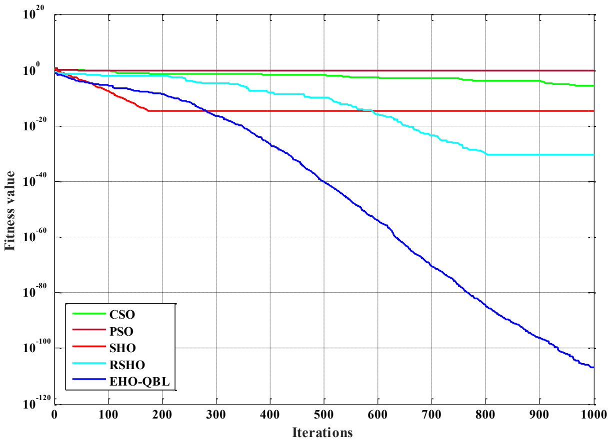

Fig.2. Convergence curve for Alphine function

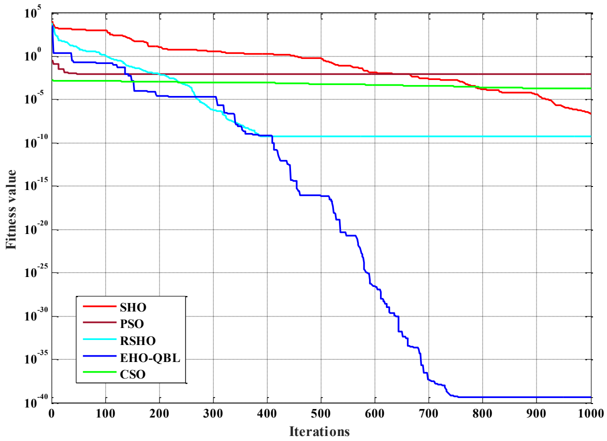

Fig.3. Convergence curve for Dixon and Price function

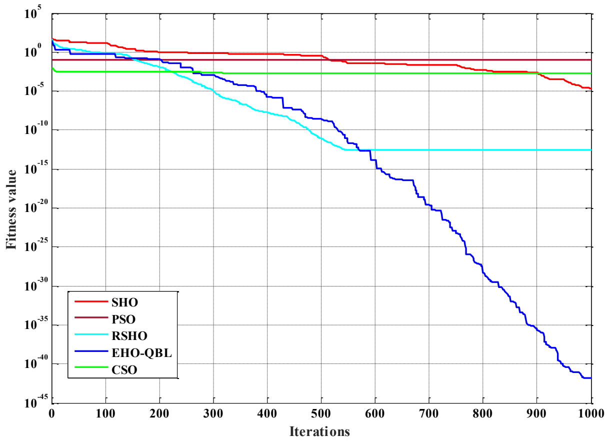

Fig.4. Convergence curve for Griewank function

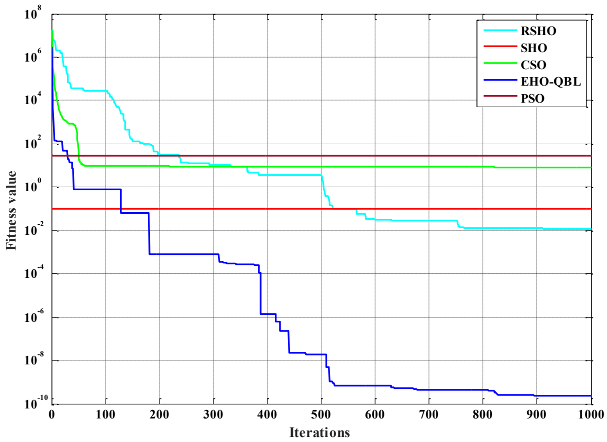

Fig.5. Convergence curve for Levy function

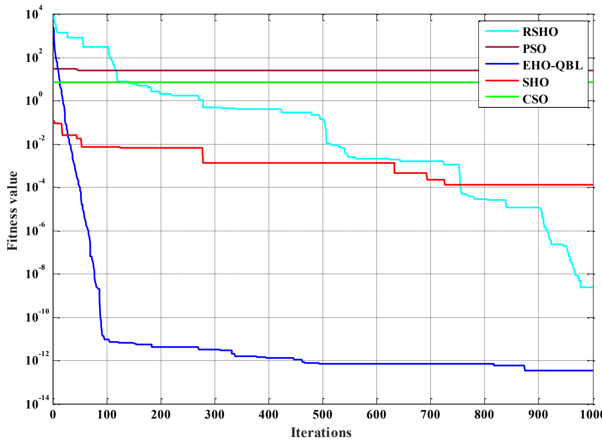

Fig.6. Convergence curve for Periodic function

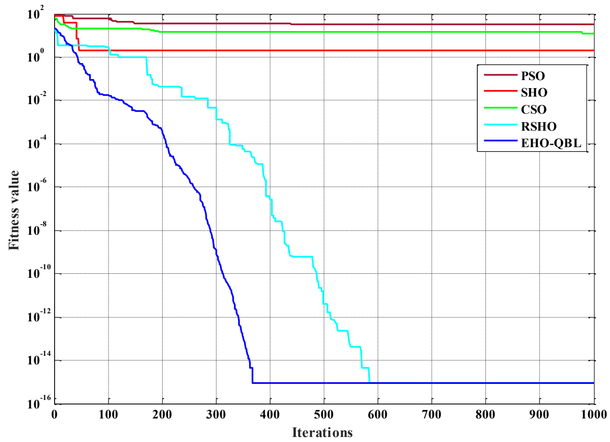

Fig.9. Convergence curve for Powell function

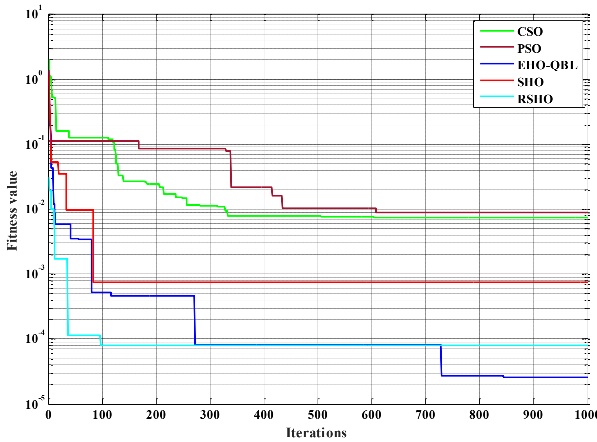

Fig.7. Convergence curve for Perm 0, D, Beta function

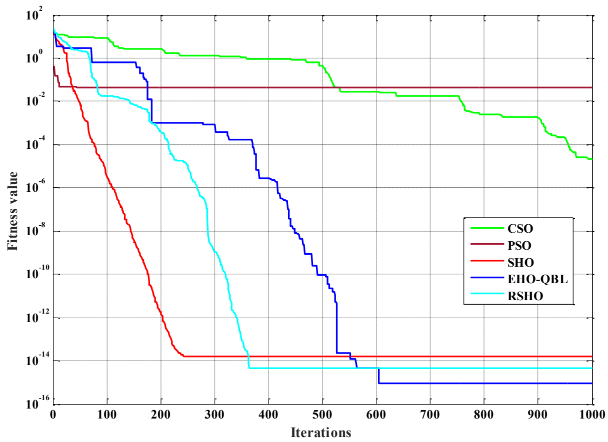

Fig.10. Convergence curve for Qing function

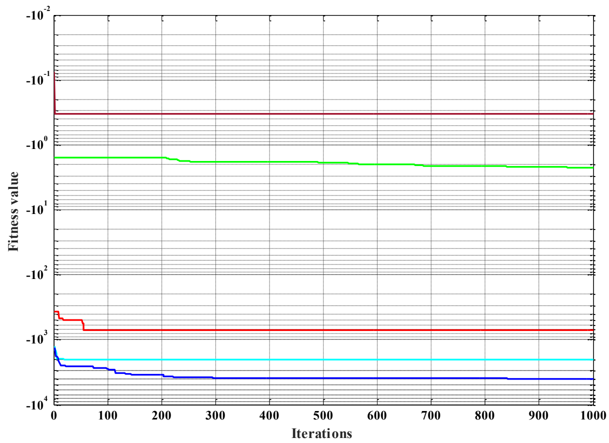

Fig.8. Convergence curve for Perm D, Beta function

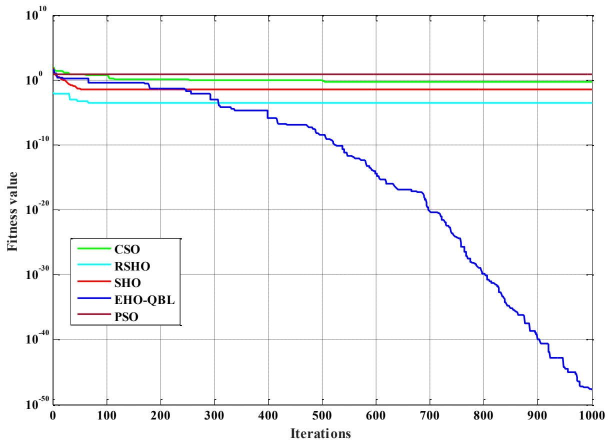

Fig.11. Convergence curve for Quartic function

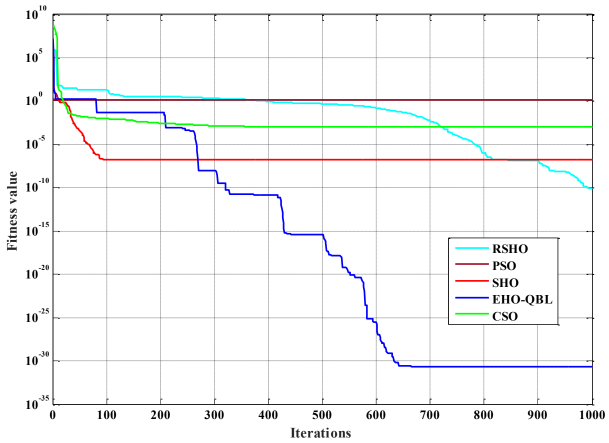

Fig.12. Convergence curve for Rastrigin Function

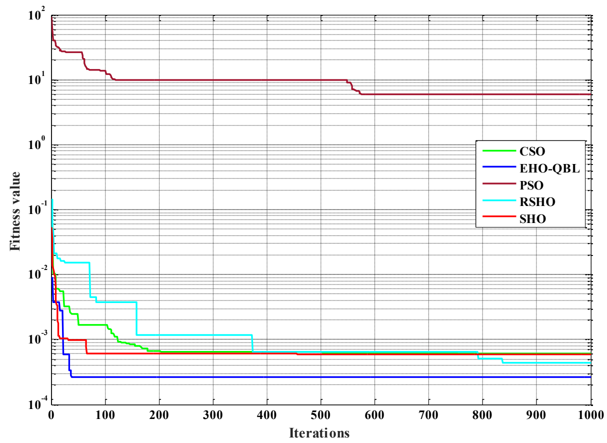

Fig.15. Convergence curve for Scwphele Function

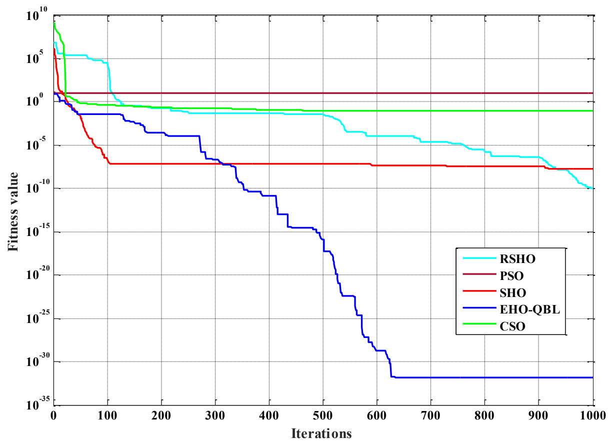

Fig.13. Convergence curve for Rosenbrock Function

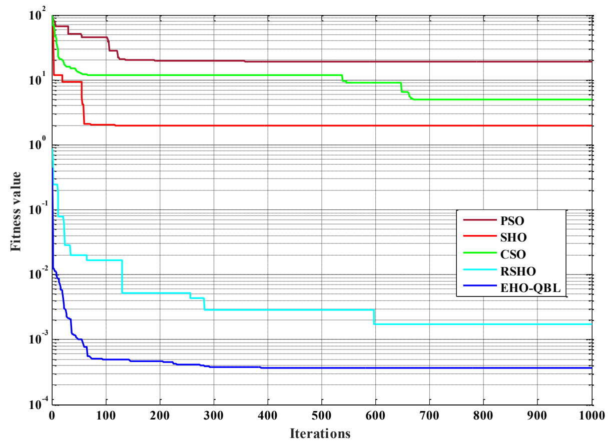

Fig.16. Convergence curve for Sphere Function

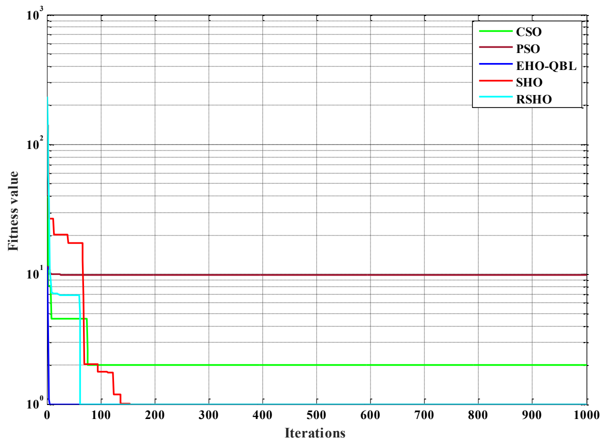

Fig.14. Convergence curve for Salomon Function

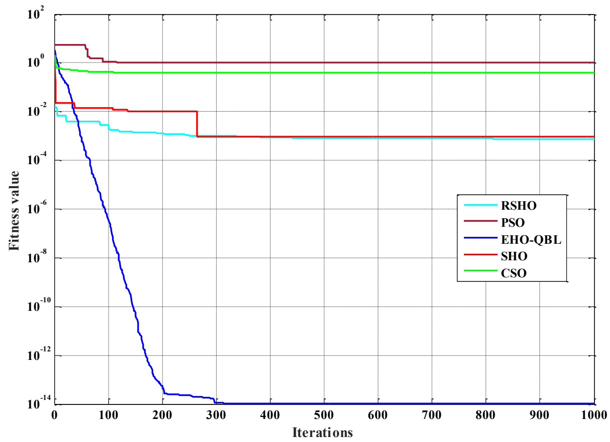

Fig.17. Convergence curve for Styblinski-tank Function

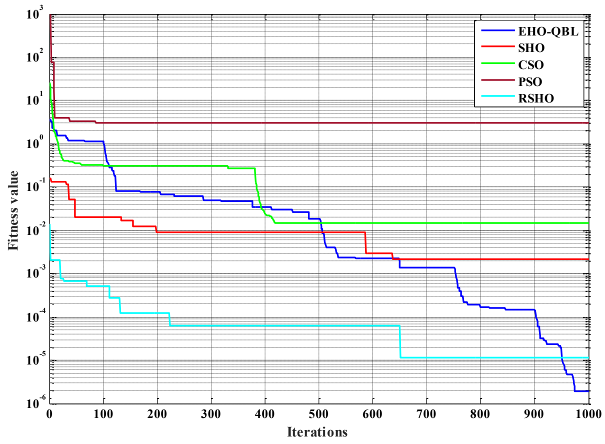

Fig.18. Convergence curve for Sum of different powers function

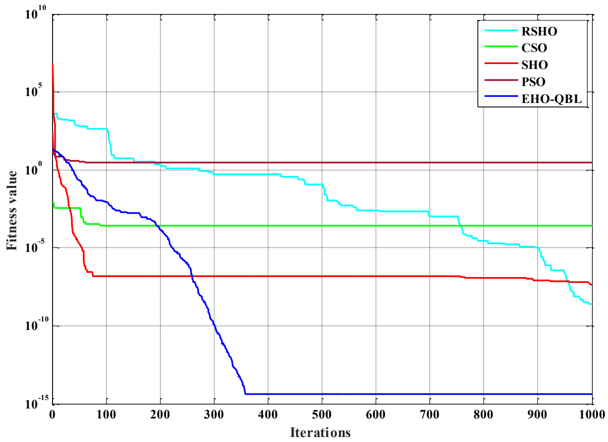

Fig.19. Convergence curve for Sum Squares function

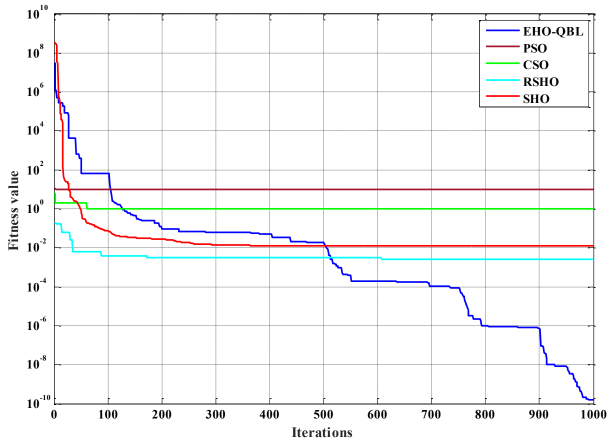

Fig.20. Convergence curve for Trid function

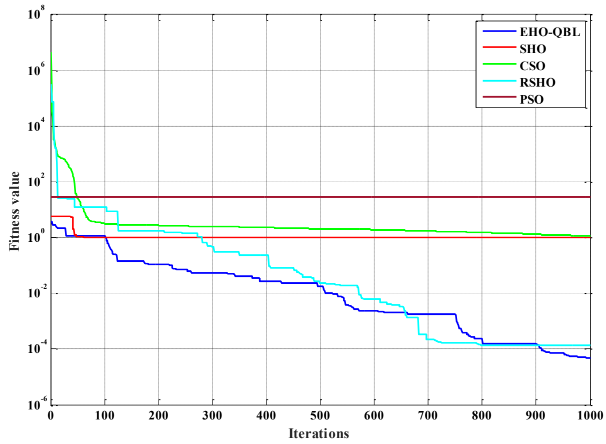

Fig.21. Convergence curve for Zakharov function

5. CONCLUSION

This paper presents the hybrid EHO algorithm and compares it with other evolutionary algorithms to substantiate how it supersedes the other algorithms on many benchmark functions. Towards achieving such effective optimization, the EHO algorithm is hybridized with the quasi opposition based learning. Subsequently, the proposed algorithm is tested on 21 most commonly used benchmark functions for analysing its efficiency. For the sake of comparison, this study has performed the same tests on other optimisation algorithms, namely Refined Selfish Herd Optimization (RSHO), Spotted Hyena Optimization (SHO), Chicken Swarm Optimization (CSO), and Particle Swarm Optimization (PSO). The test/experimental results reveal that the proposed hybrid EHO algorithm outperforms the other algorithms. Hence, it is evident that the proposed algorithm could be the most preferred choice for effectively and efficiently solving several optimization problems.

REFERENCES

[1] S. Binitha and S.S. Sathya, "A Survey of Bio Inspired Optimization Algorithms", *International journal of Soft Computing and Engineering*, Vol. 2, No. 2, pp. 137-151, 2012.

[2] A. Darwish, "Bio-Inspired Computing: Algorithms Review, Deep Analysis, and the Scope of applications", *Future Computing and Informatics*, Vol. 3, No. 2, pp. 231-246, 2018.

[3] R.S. Parpinelli, H.S. Lopes and A.A. Freitas, "Data Mining with An Ant Colony Optimization Algorithm", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, pp. 321-332, 2002.

[4] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization", Technical Report, Department of Computer Engineering, Erciyes University, Vol. 200, pp. 1-10, 2005.

[5] E. Emary, H.M. Zawbaa and A.E Hassanien, "Binary Grey Wolf Optimization Approaches for Feature Selection", *Neurocomputing*, Vol. 172, pp. 371-381, 2016.

[6] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang and W. Cai, "Grey Wolf Optimizer for Parameter Estimation in

Surface Waves", *Soil Dynamics and Earthquake Engineering*, Vol. 75, pp. 147-157, 2015.
[7] S. Mirjalili, "How Effective is the Grey Wolf Optimizer in Training Multi-Layer Perceptrons", *Applied Intelligence*, Vol. 43, No. 1, pp. 150-161, 2015.
[8] A.H. Gandomi and A.H. Alavi, "Krill Herd: A New Bio-Inspired Optimization Algorithm", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 17, No. 12, pp. 4831-4845, 2012.
[9] X.S. Yang, "Engineering Optimization: An Introduction with Metaheuristic Applications", Wiley Press, 2010.
[10] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.
[11] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm", *Advances in Engineering Software*, Vol. 95, pp. 51-67, 2016.
[12] E. Cuevas, M. Cienfuegos, R. Rojas and A. Padilla, "A Computational Intelligence Optimization Algorithm based on the Behavior of the Social-Spider", *Proceedings of IEEE International Conference on Computational Intelligence Applications in Modelling and Control*, pp. 123-146, 2015.
[13] M. Yazdani and F. Jolai, "Lion Optimization Algorithm (LOA): A Nature-Inspired Metaheuristic Algorithm", *Journal of Computational Design and Engineering*, Vol. 3, No. 1, pp. 24-36, 2016.
[14] Mostafa A. Elhosseini, Ragab A. El Sehiemy, Yasser I. Rashwan, and X.Z. Gao, "On the Performance Improvement of Elephant Herding Optimization Algorithm", *Knowledge Based Systems*, Vol. 166, pp. 58-70, 2019.
[15] Shahryar Rahnamayan, Hamid R. Tizhoosh and Magdy M.A. Salama, "Quasi-Oppositional Differential Evolution", *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 2229-2236, 2007.
[16] J. Vesterstrom and R. Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization and Evolutionary Algorithms on Numerical Benchmark Problems", *Proceedings of IEEE International Conference on Evolutionary Computation*, 1980-1987, 2014.
[17] Dinghui Wu, Shipeng Xu and Fei Kong, "Convergence Analysis and Improvement of the Chicken Swarm Optimization Algorithm", *IEEE Access*, Vol. 4, pp. 9400-9412, 2016.
[18] Adiljan Yimit, Koji Iigura and Yoshihiro Hagihara, "Refined Selfish Herd Optimizer for Global Optimization Problems", *Expert System with Applications*, Vol. 139, pp. 1-16, 2020.
[19] Gaurav Dhiman and Vijay Kumar, "Spotted Hyena Optimizer: A Novel Bio-Inspired Based Metaheuristic Technique for Engineering Applications", *Proceedings of IEEE International Conference on Advances in Engineering Software*, pp. 1-23, 2017.
[20] Vahid Beiranvand, Warren Hare and Yves Lucet, "Best Practices for Comparing Optimization Algorithms", *Proceedings of IEEE International Conference on Optimization and Engineering*, pp. 815-848, 2017.
[21] Voratas Kachitvichyanukul, "Comparison of Three Evolutionary Algorithms: GA, PSO, and DE", *Industrial Engineering and Management Systems*, Vol. 11, No. 3, pp. 215-223, 2012.
[22] Pradeep Jangir, Siddharth A. Parmar, Indrajit N. Trivedi and R.H. Bhesdadiya, "A Novel Hybrid Particle Swarm Optimizer with Multi Verse Optimizer for Global Numerical Optimization and Optimal Reactive Power Dispatch Problem", *Engineering Science and Technology, an International Journal*, Vol. 20, No. 2, pp. 570-586, 2017.
[23] H.L. Shieh, "Modified Particle Swarm Optimization Algorithm with Simulated Annealing Behavior and its Numerical Verification", *Applied Mathematics and Computation*, Vol. 218, No. 8, pp. 4365-4383, 2016.
[24] Harish Garg, "A Hybrid PSO-GA Algorithm for Constrained Optimization Problems", *Applied Mathematics and Computation*, Vol. 274, pp. 292-305, 2016.
[25] Jiquan Wang, Zhiwen Cheng, Okan K. Ersoy, Mingxin Zhang, Kexin Sun and Yusheng Bi, "Improvement and Application of Chicken Swarm Optimization for Constrained Optimization", *IEEE Access*, Vol. 7, pp. 58053-58072, 2019.
[26] Gaurav Dhiman and Vijay Kumar, "Multi-Objective Spotted Hyena Optimizer: A Multi-Objective Optimization Algorithm for Engineering Problems", *Knowledge-Based Systems*, Vol. 150, pp. 175-197, 2018.
[27] Ivana Strumberger, Nebojsa Bacanin and Milan Tuba, "Hybridized Elephant Herding Optimization Algorithm for Constrained Optimization", *Proceedings of IEEE International Conference on Hybrid Intelligent Systems*, 158-166, 2017.
[28] Jiang Li, Lihong Guo, Yan Li and Chang Liu, "Enhancing Elephant Herding Optimization with Novel Individual Updating Strategies for Large-Scale Optimization Problems", *Mathematics*, Vol. 7, No. 5, pp. 395-412, 2019.
[29] Alaa A.K. Ismaeel, Islam A. Elshaarawy, Essam H. Houssein, Fatma Helmy Ismail and Aboul Ella Hassanien, "Enhanced Elephant Herding Optimization for Global Optimization", *IEEE Access*, Vol. 7, pp. 34738-34752, 2019.
[30] Ivana Strumberger, Nebojsa Bacanin, Slavisa Tomic, Marko Beko and Milan Tuba, "Static Drone Placement by Elephant Herding Optimization Algorithm", *Proceedings of IEEE International Conference on Telecommunication*, pp. 1-8, 2017.
[31] Syed Muhammad Mohsin, Nadeem Javaid, Sajjad Ahmad Madani, Syed Muhammad Abrar Akber, Sohaib Manzoor and Javed Ahmad, "Implementing Elephant Herding Optimization Algorithm with different Operation Time Intervals for Appliance Scheduling in Smart Grid", *Proceedings of IEEE International Conference on Advanced Information Networking and Applications*, pp. 1-8, 2018.
[32] H.R. Tizhoosh, "Opposition-Based Reinforcement Learning", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 10, no. 3, pp. 578-585, 2006.
[33] M. Basu, "Quasi-Oppositional Differential Evolution for Optimal Reactive Power Dispatch", *Electrical Power and Energy Systems*, Vol. 78, pp. 29-40, 2016
[34] Provas Kumar Roy and Dharmadas Mandal, "Oppositional Biogeography-Based Optimisation for Optimal Power Flow", *International Journal on Power and Energy Conversion*, Vol. 5, No. 1, pp. 47-69, 2014.

[35] Qingzheng Xu, Na Wang, Feng Zou and Jungang Yang, "Exploring the Reasons Behind the Good Performance of Opposition-Based Learning", *IEEE Access*, Vol. 7, pp. 7259-7272, 2019.

[36] Provas Kumar Roy, Chandan Paul and Sneha Sultana, "Oppositional Teaching Learning Based Optimization Approach for Combined Heat and Power Dispatch", *International Journal of Electrical Power and Energy Systems*, Vol. 57, pp. 392-403, 2014.

[37] Provas Kumar and Sudipta Bhui, "Multiobjective Quasi-Oppositional Teaching Learning based Optimization for Economic Emission Load Dispatch Problem", *International Journal of Electrical Power and Energy Systems*, Vol. 53, pp. 937-948, 2013.

[38] Mehmet Ergezer, Dan Simon and Dawei Du, "Oppositional Biogeography-Based Optimization", *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 1-8, 2009.

[39] Mohamed Abd Elaziz, Diego Oliva and Shengwu Xiong, "An Improved Opposition-Based Sine Cosine Algorithm for global optimization", *Expert Systems with Applications*, Vol. 90, pp. 484-500, 2017.

[40] Qingzheng Xu, Lei Wang, Na Wang, Xinhong Hei and Li Zhao, "A Review of Opposition-Based Learning from 2005 to 2012", *Engineering Applications of Artificial Intelligence*, Vol. 29, pp. 1-12, 2014.

[41] G.G. Wang, S. Deb and L.D.S. Coelho, "Elephant Herding Optimization", *Proceedings of 3rd International Symposium on Computational and Business Intelligence*, pp. 1-5, 2015.