Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021, 1397-1410
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/IJNAA.2021.5741

Wildebeest optimization algorithm based on swarm intelligence method in solving optimization problems

Somayeh Askarpour[a,*], Maryam Saberi Anari[a]

[a]*Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran.*

(Communicated by Ehsan Kozegar)

Abstract

Metaheuristic algorithms are an effective way to solve optimization problems and use existing phenomena in nature to solve these problems. Due to the independence of metaheuristic algorithms from the gradient information, the objective function can be used to solve large-scale problems by optimization solutions. The organisms' behavior in nature in their interaction with each other is one of the optimization methods that are modeled as swarm-based algorithms. Swarm-based algorithms are a set of metaheuristic algorithms which are modeled based on group behavior of their organisms and social interactions. The behavior of wildebeests in nature is considered as a swarm-based algorithm for survival because it can be seen that these organisms migrate in groups and try to survive for themselves and their own herd. In this paper, a new metaheuristic algorithm (WOA) based on migratory and displacement behavior of wildebeests is presented of solving optimization problems. In this algorithm, problem solutions are defined as wildebeest herds that search the problem space for appropriate habitat. The results of the implementation of a set of benchmark functions for solving optimization problems such as the Wildebeest Optimization Algorithm, Whale Optimization Algorithm, BAT, Firefly and Particle Swarm Optimization (PSO) algorithms show that the proposed algorithm is less error rate to find global optimum and also caught up rate in the local optimum is less than the methods.

Keywords: Wildebeest optimization algorithm, Swarm-Based algorithms, Optimization problems, Metaheuristic algorithm

---

*Corresponding author

Email addresses: somayeh.askarpour@gmail.com (Somayeh Askarpour), saberi-m@tvu.ac.ir (Maryam Saberi Anari)

Received: July 2021 *Accepted:* October 2021

1. Introduction

Finding an optimal solution among different solutions of an optimization problem is NP-hard and a challenging topic. Typically, the problems surrounding us have an optimal nature because they require that a solution be selected in such a way that the objective function of the problem is minimized or maximized. Optimization issues have an objective function as their descriptor, however, in some optimization problems, the multi-objective function defines the problem, which is why these problems are more challenging. Indeed, if optimization problems have one or more objective functions, then they are single-objective and multi-objective types respectively [3]. There are various types of optimization problems around us, which are Travelling Salesman Problem (TSP) [14], Graph Coloring [20] and 0-1 Knapsack Problem [2], among which are the classical example. An important challenge in optimizing problems is to find their optimal solution through a lot of solutions. The optimal solution can be calculated by some methods such as mathematics, but these methods have many limitations. For example, in these methods, gradient information of the objective function is required, which in most cases is not possible because the objective functions are applied complex and in some cases it does not have any gradient information. Unlike mathematics-based methods, benchmarked methods of existing rules in nature and metaheuristic algorithms have a lower limit to solve these problems. Nature and its rules have been able to survive for millions of years based on pseudo-random rules and evolution. Metaheuristic methods try to use the problem-solving method in nature to solve real-world optimization problems. The metaheuristic algorithms for solving optimization problems have great points that do not require gradient information of the objective function.

Metaheuristic algorithms can be modeled in a variety of ways in nature. For example, they can use organisms' behaviors and interactions with each other to solve optimization problems. Swarm-based algorithms are a special example of metaheuristic methods is modeling organisms' group behaviors for survival. An example of these behaviors is presented in finding the food by ants and modeling it in Ant Colony Optimization (ACO) Algorithm [6]. Swarm-based algorithms can be considered as the intelligence of population members in solving an optimization problem. In these algorithms, a population of solutions is created randomly, and each member, with the help of other members, tries to improve the position and chance of survival. Metaheuristic algorithms can be modeled on Darwinian biological rules and evolution. Evolutionary algorithms are metaheuristic methods that utilize merit of a population to survive in solving optimization problems. In metaheuristic algorithms with evolutionary approach assumed that there is a population of problem solutions influenced by evolution operators such as mutation and crossover. Ultimately, the merit members of each generation are considered for the next generation, and with the passage of time, demerit members will be removed from the population. Genetic Algorithm (GA) [17] and Differential evolution (DE) algorithm [18] are two examples of evolutionary methods for solving optimization problems. Physical rules in nature have also inspired metaheuristic methods, and these rules can be used to find the optimal solution to problems. For example, the rules governing the emission of water waves in the ocean, or the rules governing in sea water evaporation respectively inspire Water wave optimization (WWO) [19] and Water Evaporation Optimization (WEO) [7].

Metaheuristic methods are not limited to organismic or immoral phenomena and evolutionary behaviors, but these algorithms have also modeled on single-cell biological mechanisms even for solving optimization problems, including Bacterial Foraging Optimization Algorithm (BFOA) [16] and Virus Optimization Algorithm (VOA) [8]. Plants are among the creatures whose evolution has been carried out for millions of years and are the first species of life, and the modeling of their behavior can lead to the creation of precise metaheuristic methods. For example, pollination behavior of flowers in the plants reproduction is an evolutionary method to be continued from generation to generation in plants, and the modeling of this behavior can be used in methods such as Flower Pollination Algorithm (FPA) [13]. One of the other metaheuristic methods, which is based on the behavior of the plants in nature and which is modeled on the root growth mechanism of the plants in optimal regions is called Runner-root algorithm (RRA) [12]. Metaheuristic methods can also be modeled on human behavior and a social, political or cultural relationship and interaction among humans. Teaching and learning behaviors among humans to gain knowledge are a clear example of group behavior between humans because people who become more knowledgeable are sharing their knowledge with others, one example of which is in the classroom. Teaching–Learning-Based Optimization (TLBO) algorithms [15] is a human-based metaheuristic algorithm. Metaheuristic algorithms do not necessarily imply the physical or biological phenomena in nature. Even in some cases, mathematical rules are used to solve optimization problems. The Sine Cosine Algorithm (SCA) [9] is an example of a metaheuristic algorithm that uses sinus and cosine function behavior to find the optimal solution. Metaheuristic algorithms can be categorized from different perspectives that one of these classifications are single-solution and multi-solution based. In the case of single-solution, the algorithm for each repetition only updates one solution in the problem space, but in multi-solution algorithms, there is a set of solutions that scan the problem space in each repetition. Multi-solution based algorithms due to the fact that there is a population of answers in each repetition, population-based algorithms are also called. The Simulated Annealing (SA) [5] and Particle Swarm Optimization Algorithm (PSO) are two examples of algorithms based on single-solution and multi-solution respectively. Organisms' group behaviors for survival are interested in nature, and also there are different organisms in nature that work together to survive. Among the organisms that have swarm-based behavior are African Wildebeests, living in large herds. Wildebeests' behavior is an appropriate sample of group behavior in nature for migrating from arid zones and dry grasslands to optimal areas, grasslands with water sources that can be used to solve optimization problems. In this paper, a new metaheuristic method based on the group behavior of African Wildebeests is presented.

2. Population-based algorithms

A Swarm-based algorithm is an example of metaheuristic algorithms that is modeled on group behavior of organisms with social interactions to solve optimization problems. Metaheuristic algorithms with swarm-based are population-based algorithms because problem solutions are considered as members of a population.

In population-based algorithms, in each repetition of the algorithm, there is a population of solutions that are scattered in the search space and seek global optimum solution.

In swarm-based algorithms, each solution, with the help of other problem information, or the status of other solutions or population members, is trying to move towards the optimum. So far, there are various behaviors in nature that have a swarm-based behavior to solve optimization problems. Some examples are referred to below:

Butterflies' behavior and the way their interaction is interesting because it is seen in these insects that they use chemical pheromones to communicate with each other. Butterfly optimization algorithm (BOA) is represented by the modeling of the mechanism for interacting between butterflies with the help of chemical pheromones. In this algorithm, the solution to the problem is considered as a butterfly, and each butterfly releases pheromone in space can lead other butterflies or solutions to the optimal points. Studies show that this algorithm is more accurate than the Genetic Algorithm (GA), Differential Evolution (DE) Algorithm, Cuckoo Search Algorithm and Particle Swarm Optimization (PSO) [1].

The study of the grasshoppers' behavior show that they live in groups in the nature and they fly as large groups and to the food sources. Grasshoppers are considered as agricultural pests and bring significant losses to agriculture and unfortunately a giant cloud of grasshoppers can devastate the nature. Flight and displacement of grasshoppers in a region are influenced by three major factors of wind, gravity and a tendency to move towards other grasshoppers. Grasshopper Optimization Algorithm (GOA) is a swarm-based algorithm that considers that three behaviors to solve an optimization problem. In the grasshopper optimization algorithm, each solution is assumed as a grasshopper and is affected by the earth gravity, wind and the effect of other grasshoppers to fly in the problem space. Implementation of the grasshopper optimization algorithm on benchmark functions shows that this algorithm is more accurate than PSO, BAT and Firefly Algorithms [10].

Hunting in nature can be individual or swarm-based, but in many species, the evolution process has been that they have over time learned that group and swarm-based behaviors have a greater chance of hunting organisms. Spotted hyenas are one of the species that hunt in nature in groups. Spotted hyenas use an interesting and multi-step mechanism for prey hunting. First of all, they identify the prey for hunting, then surround and attack the prey at the right time. Spotted Hyena Optimizer (SHO) is one of the herd behavior and swarm-based behavior algorithms that utilizes the hunting behavior of spotted hyenas. In this algorithm, the optimal point is approximately equal to the best population member, and any problem solution or hyenas attempts to attack toward to this point. The simulation and implementation of the spotted hyena optimization algorithm show that in general is more accurate than Gravitational Search Algorithm, Grey Wolf Optimizer (GWO), Genetic Algorithm, and Particle Optimization Algorithm [4].

Whales and Humpback whales are among the aquatic mammals that have high intelligence and most of their behavior are social rather than individual, and in this regard, these creatures are in the category of social creatures with swarm-based behavior. Humpback whales are commonly team up to hunt shoal of fish, and it has been observed to use bubble hunts to hunt school of fish. Each of them, lead shoal of fish to an optimal point by creating bubble walls and attack school of fish at the right time. Whale Optimization Algorithm (WOA) works according to bubble-net feeding mechanism of whales. In this algorithm, each problem solution is a whale and uses upward-spirals and double-loops motion to navigate the problem space. Implementing WOA on benchmark functions and several examples of practical optimization problems show that this algorithm is more accurate than PSO, Gravitational Search Algorithm and Differential Evolution (DE) Algorithm [11].

3. Proposed algorithm

One of the evolutionary behaviors in organisms is the formation of a group to increase the chance of survival. Organisms have a greater chance of hunting, feeding, defending, and nesting by living in a group. With the passing of time, the organisms found out that they could increase their survival with the help of their own kind. There are various group behaviors in nature, one of these behaviors is the change of habitat and migration to find pasture and optimal food position. Immigration is abundant in nature in various organisms such as birds, fish, wildebeest, and other herbivores. Migration and habitat change is an evolutionary behavior to increase the chances of survival in many organism species. In fact, organisms with intelligent mechanisms try to choose routes for migration that increase their chances of survival. African wildebeests are the organisms that have swarm-based behavior among them. In Fig.1, an example of an African wildebeest is shown in grazing and feeding in a thick forest:

The behavioral study of African wildebeest suggests that their own kind friendship and defensive behavior is abundant. African wildebeests are among the organisms show social and swarm-based

Figure 1: An African Wildebeest is eating plants

behaviors. These organisms are wild species in nature that migrate in a group to reach a new habitat. African wildebeests use a cooperative and collaborative pattern and to search for an appropriate and optimal habitat in their migration. The behavioral mechanism of these organisms is such that they can help each other to migrate to the optimal areas of food and water. In these organisms, each population member emigrates according to the pattern of neighbors, and each population member attempts individually or by experience of their own herd to immigrate. In Fig. **??**, the behavior of migrating African wildebeests to find new pasture is shown with the help of other herds of their own kind:

Figure 2: African wildebeest herds are changing the habitat

The study of the swarm-based behavior of African wildebeest in migration indicates that they consider the following two factors for migration:

- African wildebeests move towards large herds and try to join themselves to a large hard because they are less likely to be hunted.
- African wildebeests select areas with more vegetation and forage to move and migrate.

In addition to the two important rules mentioned above, the migration of African wildebeests has taught them that only increase the risk of getting lost and being hunted, moreover, those who are in the herd have higher chance to survive and can also give birth. In this paper, a new optimization method based on the swarm-based behavior of African wildebeest is presented for solving optimization problems. In this algorithm, the following assumptions are considered:

- Each problem solution is considered as an African wildebeest, then evaluated by the objective function of the problem its merit is defined.
- Any population member with more merit will be more effective to lead other members towards the optimal answer so that each member is assigned a weight that is commensurate with their merits.
- Members who have more weight can attract more members and less weight it is the less able to lead.

In each repetition of the algorithm, population members are trying to move or migrate by impression of weight and merit of other herd members in a way that they search the search space problem.

Weak population members are gradually being omitted by factors such as starvation or hunting, and new members add to the herd by giving birth.

Assume that each optimization problem solution is considered as a member of the population of African wildebeest. In accordance with equation 1, this solution is defined in D-dimensional space:

$$S_i = \{S_i^1, S_i^2, \ldots, S_i^D\} \tag{1}$$

In this equation, $S_i$is a member of the i-th population and S_i^j is considered the j-th dimension of i-th solution. A set of initial solutions is considered as a population in equation 2:

$$S = \{S_1, S_2, \ldots, S_N\} \tag{2}$$

In this equation, N is assumed to be equal to the size and population number, S is population or a set of problem solutions. In the wildebeest optimization algorithm is assumed that each wildebeest has a chance of survival in terms of its merit, and, on the other hand, one weight can be assigned to each wildebeest that indicates the importance of its rule in the herd. The most and least important can be considered as one and zero respectively. To determine the weight and importance of each solution, you can use the objective function or f. Before evaluating the population members, it is required that each population member to be distributed in the problem space randomly according to equation 3, to be evaluated by the objective function:

$$S_i = L + (U - L), rand(0, 1) \tag{3}$$

In the above equation, L and U represent respectively the lower and upper bound of each dimension of the problem. In equation 4, we can determine the weight and value of any solution to lead a herd or its importance in the next generation of a herd:

$$W_i = \frac{f(S_i) - f^{min}}{f^{max} - f^{min}} \tag{4}$$

In this equation, f^{max} and f^{min} are considered to be the maximum and minimum merit of wildebeest respectively. $f(S_i)$ is the merit of a population member, and ultimately, W_i is the weight of the wildebeest S_i. Each population member, in terms of its weight, plays a role in guiding the herd to the optimal points. It is assumed that the highest merit population member leads the most number of wildebeests in its environment to the optimal point.

Suppose that the most merit population member with the f^{max} merit has been able to choose α member of its own environment for migration and displacement. According to equation 5, a member of the merit weight W_i can lead a number of members of its own environment:

$$n_i = \alpha W_i \tag{5}$$

In this equation, S_i member chooses a number of members which is called n_i to lead the herd. According to equation 6, the size of group members can be created around population members is one number greater than the size of the initial population or N:

$$n = \sum_{i=1}^{N} n_i > N \tag{6}$$

When the initial population is steady in the proposed method, the best way is eliminating some population members in each repetition:

- Each population member has a chance of survival and a chance of being lost or hunted.
- Each wildebeest in a larger herd has less chance of being lost
- On the contrary, each wildebeest in the small herd has more chance of being lost and hunted.

In fact, with this mechanism, the non-optimal answers will be eliminated from the population in each repetition with the passage of time. The chance of hunting or losing any population member is described in equation 7:

$$p(S_i) = 1 - W_i \tag{7}$$

In each repetition of the algorithm, population members can be as a leader or head of the herd to change the habitat of their own environment and equation 8 is presented for modeling this behavior:

$$S_i^{new} = S_i + rand(-1, +1), r(t) \tag{8}$$

In this equation, S_i is the status of the leader of the herd, $r(t)$ is the radius of spreading out the members around the leader of the herd in the t repetition of the algorithm and S_i^{new} is the position of a population member that is around a wildebeest as head of the herd to change the habitat. To increase the convergence of the algorithm, it's better to choose a dynamic $r(t)$ and decrease it repeatedly based on its repetition that finally, the nature of the algorithm in the first repetition is global and in the last repetition is local. Therefore, the equation 9 is suggested to change and reduce nonlinear of $r(t)$:

$$r(t) = L.\frac{r(t)}{t} \tag{9}$$

In this equation, L is a parameter in the $range[1, 4]$, $r(1)$ is the radius of spreading out the members around the leader of the herd in the t repetition of the algorithm.

4. Implementation

An optimization problem can be described with mathematical function. Objective functions are actually represent an optimization problem and finding their minimum or maximum is important. Evaluation functions or benchmark functions are an example of multivariate mathematical optimization problems to evaluate the performance of metaheuristic methods. The ultimate purpose in a benchmark function with optimizations is to be found by the metaheuristic method. Various samples of benchmark functions are used to measure the convergence of metaheuristic methods, including the Sphere, Rosenbrock, Rastrigin, Schwefel, Griewank and Ackley benchmark functions. The three-dimensional graph of the Sphere, Rastrigin, Schwefel, Griewank and Ackley benchmark functions is respectively shown in Fig. 3, 4, 5, 6 and 7.

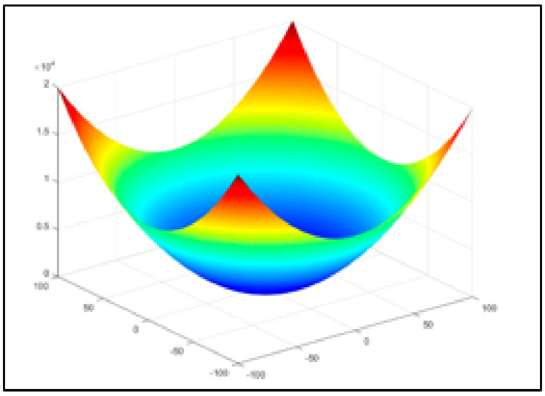

Figure 3: 3D graph of the Sphere benchmark function

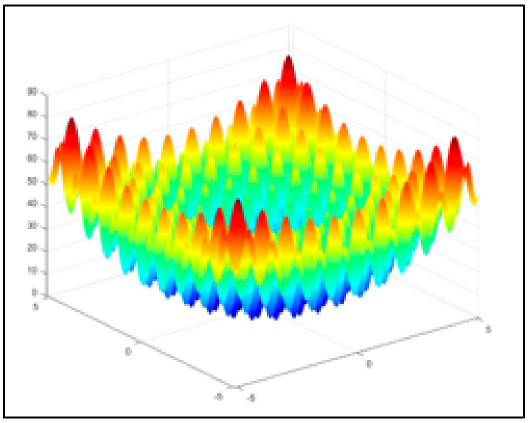

Figure 4: 3D graph of the Rastrigin benchmark function

Each of the Sphere, Rosenbrock, Rastrigin, Schwefel, Griewank, and Ackley benchmark functions has a mathematical standard that global minimum is considered as a global optimum in this function. Some benchmark functions such as Sphere and Rosenbrock have small space complexity, and these types of functions have no local optimization, whereas benchmark functions such as Rastrigin, Schwefel, and Griewank have also global optimum and local optimization and because of this, their search space is more complex than the Sphere and Rosenbrock benchmark functions. In Table 1, a list of some of the most widely used benchmark functions is shown with their standards in n space:

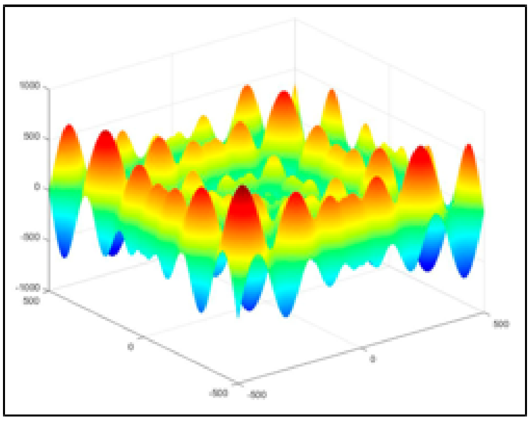

Figure 5: 3D graph of the Schwefel benchmark function

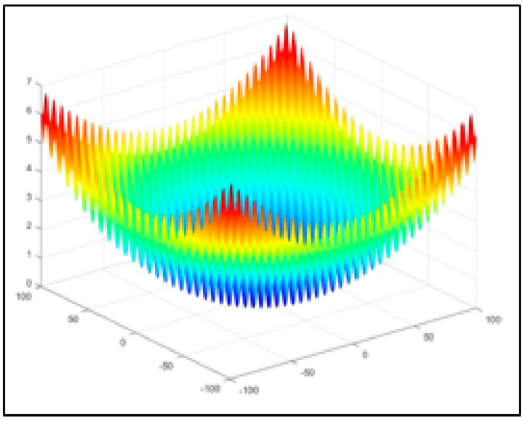

Figure 6: 3D graph of the Griewank benchmark function

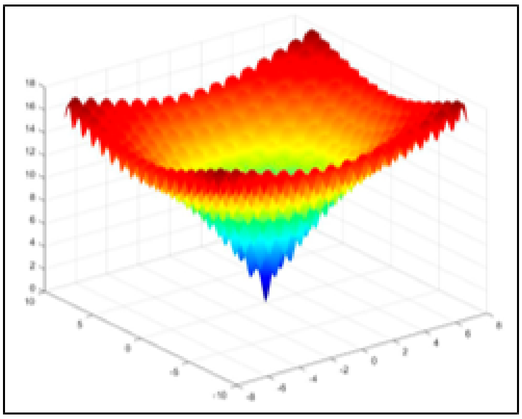

Figure 7: 3D graph of the Ackley benchmark function

Table 1: The standard of benchmark functions used for evaluation

Function	Expression formula
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigin	$f_3(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$
Schwefel	$f_4(x) = \sum_{i=1}^{n} \left[-x_i sin(\sqrt{\|x_i\|})\right]$
Griewank	$f_5(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$
Ackley	$f_6(x) = -20exp(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} - exp(\frac{1}{n} \sum_{j=1}^{n} cos(2\pi x_i)) + 20 + e$

To evaluate the proposed algorithm, benchmark functions are used as the objective function and compared and evaluated simultaneously with several optimization algorithms. To evaluate the proposed method, the accuracy and convergence indicators are used toward to optimal solutions. In these implementations:

1. A standard cost function such as Sphere is selected
2. Then the proposed algorithm is implemented with that cost function
3. The calculation error diagram of the global optimum in terms of the algorithm repetition was drawn.
4. For better evaluation, the experiment runs 30 times with a population of 10 and a repetition of 100 and the average of results are shown in the output.

In order to determine the efficiency in Matlab programming environment, the proposed algorithm is compared with several optimization algorithms such as Whale optimization algorithm, Bat, Firefly and Particle swarm optimization (PSO) algorithms, and the results are presented. In implementing the proposed method on an objective function, each of the mentioned algorithms is also implemented on the objective function so that we can have a comparison between them in terms of accuracy.

In these evaluations, the value of the parameter L in the proposed method is equal to 2.5, $r(1)$, or the distribution radius of the members around the head of the herd is 0.5, individual and collective learning coefficients of the Particle algorithm are 2.05 and 2.05 respectively, and in the Bat algorithm, the local search probability is 0.5 and the coefficient ε is equal to 0.01 and the initial loudness value is 0.5. In the firefly algorithm, the initial attraction is assumed to be equal to 2 and the coefficient of the convergence parameter is also considered to be 0.2.

In whale optimization algorithm, the parameters are considered exactly dedicated from the base paper of whale optimization algorithm.

The convergence diagram of the proposed algorithm and other metaheuristic algorithms is presented on the mathematical benchmark functions used in Fig. 8, 9, 10, 11 and 12.

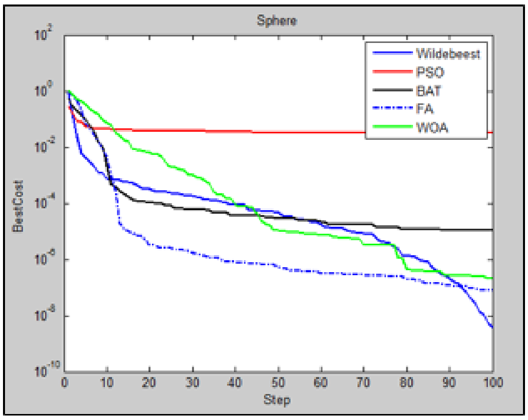

Figure 8: Comparison and evaluation of the Sphere benchmark function

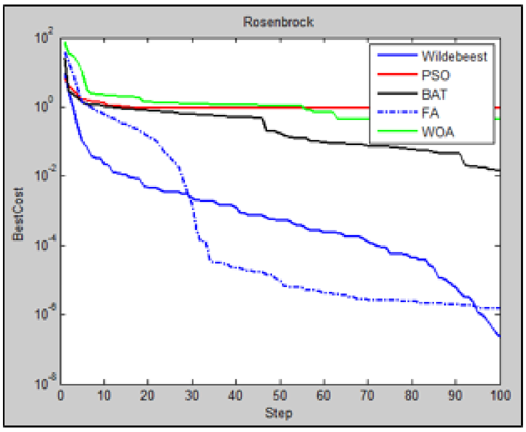

Figure 9: Comparison and evaluation of the Rosenbrock benchmark function

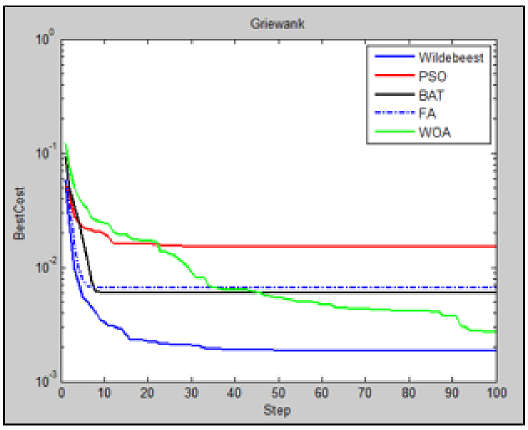

Figure 10: Comparison and evaluation of the Griewank benchmark function

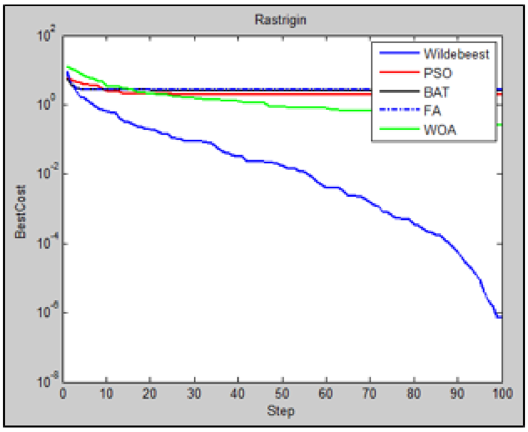

Figure 11: Comparison and Evaluation of the Rastrigin Benchmark function

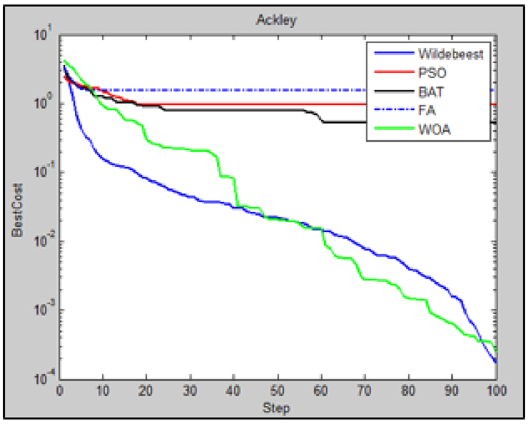

Figure 12: Comparison and evaluation of the Ackley benchmark function

The comparison of the calculation global optimum error rate in terms of repetition in the proposed algorithm, the Whale optimization algorithm, Bat, Firefly and Particle optimization algorithms show that in all algorithms the optimal global error rate is depend on the repetition of the downward algorithm, which means that each of these algorithms has been able to guide the population members and thus the merit population member in terms of their repetition to the optimal problem or objective function, but the error reduction rate is not the same based on repetition in these algorithms . The evaluations show that the proposed method has a higher reduction error rate in terms of repetition in benchmark functions, in other words, the proposed method has been able to lead the population members faster to the global optimum. Our evaluation results show that the proposed method in the complex benchmark functions such as Rastrigin, Griewank, and Ackley reduces the error rate with higher slope in comparison with other algorithms. Furthermore, other algorithms move in this benchmark function with lower slope towards to the optimal point, and this is emphasized than in some experiments, algorithms such as PSO, Fireflies, Bats, and Whale are caught up in the local optimum. In general, if the slope of the error reduction chart is very small in terms of repetition, it indicates that the method is susceptible to being caught up in local optimum. In the benchmark functions of Rastrigin, Griewank, and Ackley, the proposed method continuously reduces its slope, and this suggests that the algorithm is seldom caught up in the local optimum.

Our evaluation results in the Rastrigin, Griewank, and Ackley benchmark functions show that if 50 experiments are performed for each benchmark function, the probability of catching up in the local optimum of the proposed method, Whale, Firefly, PSO and Bat optimization algorithms respectively equal to 5.33%, 6.66%, 12%, 10.66% and 16%. Comparison of the proposed method and other algorithms with simpler benchmark functions such as Sphere and Rosenbrock suggests that the proposed algorithm could find global optimum with finite error rate, then Firefly algorithm iss the second rank. Our results indicate that the proposed algorithm and Whale algorithm are more suited in the complex benchmark functions. Furthermore, the proposed method and Firefly are more accurate in the simple benchmark functions. The results of the implementation of the proposed algorithm on the Sphere, Rosenbrock, Griewank, Rastrigin and Ackley benchmark functions show that the average error rate in finding the global optimum is order of $10^{-9}, 10^{-7}, 10^{-2.5}, 10^{-6}, 10^{-4}$ and it provides less error rate than Whale, PSO, Bat and Firefly optimization algorithms. In general, it can be concluded that the proposed method is less caught up in the local optimum than Whale, PSO, Bat and Firefly optimization algorithms, and it reduces the optimal error calculation rate with higher slope by repetition. Eventually, the proposed algorithm calculates the optimal of these benchmark functions with less error than other algorithms in the last repetition.

5. Conclusion

Organisms' behavior is considered as problem-solving methods in nature, and these methods are modeled in the form of metaheuristic algorithms. Swarm-based behavior of organisms is one of the evolutionary behaviors for survival and it is inspired by Swarm-based intelligence algorithms. In these algorithms, each problem solution, along with other solutions, looks for the optimal problem. In swarm-based behavior, the intelligence of the whole population is used to find optimal solutions. The swarm-based behavior of African wildebeest in migration is an example of the swarm-based behavior that the organisms do to stay alive. In this research, a new metaheuristic algorithm is presented based on the behavior of changing the African wildebeest habitat, the hunting of weak members and the replacement of merit members. African wildebeest optimization algorithm is a form of competition for survival which formulated in solving optimization problems.

The results of the implementation of the proposed method on benchmark functions showed that the proposed method is less error rate than whale optimization algorithm, Bat, Firefly and Particle swarm optimization (PSO) algorithms succeeded in finding global optimum and it has also less tendency to converge to the local optimum. In the future study, a new practical version of the African wildebeest optimization algorithm for developed medical images segmentation to detect diseases such as brain tumors with higher accuracy. In fact, our future research combines the fuzzy clustering algorithm with African wildebeest optimization algorithm to increase the accuracy of diagnosing brain tumors.

References

[1] S. Arora and S. Singh, *Butterfly optimization algorithm: a novel approach for global optimization*, Soft Computing, (2018) 1-20.

[2] Y. Atay, I. Koc, I. Babaoglu and H. Kodaz, *Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms*, Applied Soft Computing, 50 (2017) 194-211.

[3] N. Delgarm, B. Sajadi, F. Kowsary and S. Delgarm, *Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO)*, Applied Energy, 170 (2016) 293-303.

[4] G. Dhiman and V. Kumar, *Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications*, Advances in Engineering Software, 114 (2017) 48-70.

[5] A. E. S. Ezugwu, A. O. Adewumi and M. E. Frîncu, S*imulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem*, Expert Systems with Applications, 77 (2017) 189-210.

[6] H. Ismkhan, *Effective heuristics for ant colony optimization to handle large-scale problems*, Swarm and Evolutionary Computation, 32 (2017) 140-149.

[7] A. Kaveh and T. Bakhshpoori, *Water evaporation optimization: a novel physically inspired optimization algorithm*, Computers & Structures, 167 (2016) 69-85.

[8] M. D. Li, H. Zhao, H. Weng and T. Han, *A novel nature-inspired algorithm for optimization: Virus colony search*, Advances in Engineering Software, 92 (2016) 65-88.

[9] S. Mirjalili, *SCA: a sine cosine algorithm for solving optimization problems*, Knowledge-Based Systems, 96 (2016) 120-133.

[10] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, *Grasshopper optimization algorithm for multi-objective optimization problems*, Applied Intelligence, 48(4) (2018) 805-820.

[11] S. Mirjalili and A. Lewis, *The whale optimization algorithm*, Advances in Engineering Software, 95 (2016) 51-67.

[12] T. T. Nguyen, T. T. Nguyen, A. V. Truong, Q. T. Nguyen and T. A. Phung, *Multi-objective electric distribution network reconfiguration solution using runner-root algorithm*, Applied Soft Computing, 52 (2017) 93-108.

[13] S. M. Nigdeli, G. Bekdaş and X. S. Yang, *Application of the flower pollination algorithm in structural engineering*, In Metaheuristics and optimization in civil engineering, (2016) 25-42.

[14] E. Osaba, X. S. Yang, F. Diaz, P. Lopez-Garcia and R. Carballedo, *An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems*, Engineering Applications of Artificial Intelligence, 48 (2016)59-71.

[15] V. K. Patel and V. J. Savsani, *A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO)*, Information Sciences, 357 (2016) 182-200.

[16] K. Tang, X. Xiao, J. Wu, J. Yang and L. Luo, *An improved multilevel thresholding approach based modified bacterial foraging optimization*, Applied Intelligence, 46(1) (2017) 214-226.

[17] Y. Yuan, H. Xu, B. Wang and X. Yao, *A new dominance relation-based evolutionary algorithm for many-objective optimization*, IEEE Transactions on Evolutionary Computation, 20(1) (2016) 16-37.

[18] H. Zaheer, M. Pant, S. Kumar, O. Monakhov, E. Monakhova and K. Deep, *A new guiding force strategy for differential evolution*, International Journal of System Assurance Engineering and Management, 8(4) (2017) 2170-2183.

[19] Y. J. Zheng, *Water wave optimization: a new nature-inspired metaheuristic*, Computers & Operations Research, 55 (2015) 1-11.

[20] Y. Zhou, J. K. Hao and B. Duval, *Reinforcement learning based local search for grouping problems: A case study on graph coloring*, Expert Systems with Applications, 64 (2016) 412-422.