Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3451-3462
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6108

An optimized swarm intelligence algorithm based on the mass defence of bees

Robabeh Chanpa[a], Mohammad Ali Jabraeil Jamali[a,*], Abdolreza Hatamlou[b], Babak Anari[a]

[a]*Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran*
[b]*Department of Computer Engineering, Khoy Branch, Islamic Azad University, Khoy, Iran*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Swarm intelligence is a modern optimization technique and one of the most efficient techniques for solving optimization problems. Their main inspiration is the cooperative behavior of animals within specific communities. In swarm intelligence algorithms, agents work together and the collective behavior of all agent causes converge at a point close to the global optimal solution. In this paper, we model the behavior of bees in defending the hive against invading bees to provide a new optimization algorithm. In the proposed algorithm, the coordinated performance of bees in identifying the invader creating a circle around the invading bee and generating heat during the siege of the invading bee and also the heat emitted from each bee are modeled. The simulation results of the proposed algorithm show a successful competitive behavior in achieving the global optimum in comparison with the firefly, ant colony, artificial bee colony, whale and grey wolf algorithms.

Keywords: Swarm intelligence, Optimization, Algorithms, Mass defence of bees

1. Introduction

Artificial intelligence algorithms are now widely used to solve complex optimization problems. Swarm intelligence is a branch of artificial intelligence used to model social nature, such as ant colonies and bees. Social insect colonies include properties such as collective behavior and decentralized control [30]. Although these factors are not simple with their limited capabilities, they cooperate with the application of specific patterns of behavior and the tasks they use for survival. In swarm intelligence, populations are number of factors that interact locally with each other and in their own

*Corresponding author

Email addresses: (Robabeh Chanpa), m_jamali@itrc.ac.ir (Mohammad Ali Jabraeil Jamali), (Abdolreza Hatamlou), (Babak Anari)

Received: October 2021 *Accepted:* December 2021

environment and have these simple creatures behave cooperatively when working. Swarm intelligence techniques mathematically emulate the collective behavior of these organisms [4]. Stochastic and evolutionary optimization and search algorithms are efficient methods that are used especially to find optimal global solutions to problems. The random nature of these algorithms prevents them from getting stuck in local optimal points [22]. Many of these algorithms are inspired by biological factors, one of which is the mass defense algorithm of bees.

Meta-exploration, optimization techniques have become very popular over the past two decades. Surprisingly, some of them, such as genetic algorithms, ant optimization, and particle swarm optimization, are well known not only among computer scientists but also among other scientists. The reason can be summarized in four items: simplicity, flexibility, non-derivation mechanism and local optimal prevention [23]. Ant colonies or bees are a simple example of a population system.

For example, PSO algorithm simulates animal's social behavior, like insects, birds and fishes. There conform a cooperative way to find food, and each member in the swarms keeps changing the search pattern according to the learning experiences of its own and other members [16, 27]. ACO algorithm is introduced based on the foraging behavior of an ant for seeking a path between their colony and source food. The FA is derived from the behavior of fireflies in nature and includes advantages such as easy coding. However, this algorithm does not use the global optimal point, and this can prevent premature convergence similar to what happens in the PSO algorithm. in the GWO algorithm, the population moves towards optimal responses, or alpha, beta, and delta wolves. In each iteration, the optimal particles are determined based on the fitness function, and the best particle is called alpha. In this algorithm, solutions are found at each stage, regardless of the degree of fitness, and in other words, they are non-greedy [5, 24].

Over the past two decades, nature-inspired optimization algorithms have shown considerable success as intelligent optimization methods alongside classical methods. Optimization has been one of the most important areas of research in recent decades, resulting in the design of various types of algorithms. In many engineering cases, the objective function is usually dealt with, which must be optimized. In group methods, agents work together and the collective behavior of all agents leads to a convergence. There are various methods to reach this optimal point, and of course, each of these methods has disadvantages.

In all population-based algorithms such as PSO, GA, WOA , etc., which has social behavior among members of the population, several basic features must be considered: the ability of the algorithm to explore and search in all parts of the solution space and the ability to extract the best possible solution [8, 23].

In the rest of the paper, in section 2, a brief review of swarm intelligence algorithms is given. In Section 3, the proposed swarm intelligence algorithm based on mass defense of bees is introduced and its mathematical modeling is stated. Section 4 shows the simulation results of the proposed algorithm and its comparison with other algorithms. At the end, the conclusion section and future work are presented.

2. Swarm intelligence algorithms

Researchers have studied some of the phenomena in nature to provide new methods for solving complex optimization problems. Holland proposed a genetic algorithm based on simulations of natural selection and the Darwin's theory of biological evolution. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection [14].

In recent decades, biologists and natural scientists have also been closely following the behavior of social insects because of the amazing ability of their natural intelligence system. In the late 1980s, however, computer scientists proposed a scientific study of these natural intelligence systems in the field of artificial intelligence. In 1989, the term "Intelligence Swarm" was first introduced by X. B. Wang et al [28] in the context of global optimization as a set of algorithms for intelligent robotic control. In 1995, particle swarm optimization was introduced by J. Kennedy et al [11, 17].

An example for population systems is PSO. In the PSO algorithm, members of the population solve the problem through the exchange of information [3, 15]. PSO is an optimization technique in which each particle tries to move in the direction in which the best individual and group experiments have taken place. The problem with this algorithm is that it gets caught up in local optima. This algorithm has a high convergence speed. The PSO algorithm performs better in terms of convergence than other traditional swarm intelligence algorithms [7**?** , 13, 31].

A few years later, in 2005, the ABC algorithm was introduced by D. Karabago as a new member of the family of swarm intelligence algorithms [2]. In ABC system, artificial bees fly around search space and some choose food sources depending on the experience of themselves and adjust their locations. Others choose the food sources randomly without using experience. If the nectar amount of a new source is higher than previous one, they memorize the new position and forget the previous one. The policy of this algorithm is to move towards better solutions by using the neighbor search mechanism and abandoning weak solutions.

The ACO algorithm has many advantages for creating efficiency and optimal use of the limited resources of wireless networks [12]. In This approach, the convergence speed is low and the need for large memory is one of its disadvantages. Instead of storing the information on the previous generation, this algorithm should store the total clone information [6].

FA introduced in 2010 modeled on the behavior of firefly is based on the flashing patterns and behavior of tropical fireflies and has some disadvantages such as access to several local optimizers. Likewise, similar to the other swarm intelligence algorithms, it performs local search from which sometimes it can't get rid of. [6]. One of the disadvantages of this algorithm is its low ability to find optimal solution, because the firefly is always moving in one direction [29].

In 2016, the GWO algorithm was introduced, which mimics the leadership hierarchy mechanism of gray wolves in hunting. Four types of gray wolves such as alpha, beta, delta and omega have been used to simulate leadership hierarchies. This algorithm can be used for challenging problems with unknown search spaces [5].

The WOA is a nature-inspired meta-exploration; this algorithm mimics the social behaviors of humpback whales. Humpback whales can recognize the location of hunt and encircle them. Then they assume that the current best candidate solution is the target prey or is close to the optimum. After the best search agent is defined, the other search agents will update their positions towards the best search agent. [23, 25].

Less exploration in space search is the major drawback of traditional approaches and this makes to get caught in local optimization. Modern approaches use random methods to solve the current problem. Exploration can be boosted applying a number of factors. These factors search different points in parallel in the problem space [4]. However, each of swarm intelligence algorithms has been successfully operated in solving a series of different problems in various fields [18, 20, 26].

All these artificial intelligence algorithms are inspired and shaped by simulation behavior and learning evolutionary mechanisms [21]. Almost all cases with aggressive behaviors are microorganisms or animals. Few researchers use human intelligence as a basis of artificial swarm intelligence algorithms. Each of the swarm intelligence algorithms has effectively improved efficiency in different sciences. Considering multiple efficiency parameters for convergence, it can be used effectively in configuring network parameters and multi-criteria decisions [8, 9, 10, 19].

3. Proposed mass defense algorithm of bees

The proposed algorithm simulates the defensive behavior of bees against invading red bees in the nature. Imagine several giant invading bees after noticing a specific odor on their watch; they land on a beehive and start beheading the bees inside. They trap skinny, very small bees inside the hive and separate their heads from their bodies using their powerful jaws.

The scenario we came up with seems very scary, but it does happen in reality. Of course, bees usually try to repel the invaders by biting them, but unfortunately they are not able to penetrate the powerful shell of their bodies. Eventually the bees die one after another, and it is interesting to know that many of them are killed by one of these red bees in just one minute. At this rate, 30 of these red invaders will be able to kill 30,000 bees in just a few hours.

In addition to being a danger to humans, red bees can also destroy the life of bees, but to prevent this from happening, bees take an interesting way called mass defense. Invading red bees are strong, but bees are also equipped with a strong defense mechanism against them, they can create a ball around the watch bee and shake it so much that it dies due to rising body temperature. In this way, they save their hive from the red bees that will come there when they detect the smell.

If a red bee finds a beehive, the hive's inhabitants do not rush into it and instead allow it to enter, creating a specific odor that has already been mentioned and dragging its entire species to the nest. But this is a trap, and as soon as the beekeeper enters, the inhabitants of the hive form a ball around it. The bees then begin to shake their bodies to raise their temperature, thereby cooking the watch bee inside. As a result, carbon dioxide accumulates inside the ball, killing the invading bee. But another interesting thing about these bees is their body anatomy, which lacks a heart and pumps blood to different organs by contracting the body. Therefore, bees always try to squeeze the ball around it to prevent blood from circulating inside its organs. So rising temperatures, carbon dioxide buildup and blood flow are all causes of red bee death. This action will also cause the death of the defending bees, and if the attacking bee is not destroyed, the new defenders will replace the destroyed defenders [1].

3.1. Mathematical model of the proposed method

Stochastic, evolutionary and optimization algorithms are efficient methods that are used especially to find optimal global solutions for complicated problems. The random nature of these algorithms prevents them from getting stuck in local optimal solutions. In practical optimization problems such as engineering designs, management of organizations and economic systems, the main focus is usually on obtaining global optimal solutions. Many of these algorithms are inspired by biological systems. The proposed algorithm is based on the mass defense of bees against red bees. The algorithm consists of 2 phases: defense phase and update phase.

Defense phase The defending bees circle around the attacker and begin to vibrate and generate heat. With increasing distance to radius r, the heat intensity $\theta(i,j)$ decreases. As a result, we have Eq. (1):

$$\theta(i,j) \sim \frac{1}{r^2} \tag{1}$$

The heat emitted $\theta(i,j)$ is directly related to the number of vibrating bees (N_i). The more bees, the more heat is generated. Eq. (2):

$$\theta(i,j) \sim N_i \tag{2}$$

There are two main variables in the bee mass defense algorithm: Changes in heat intensity and its formulation.

As the heat produced by the bee's increases, their lifespan actually decreases, because they get heatstroke and die. If their lifespan is indicated by L; we have Eq. (3):

$$\theta(i,j) \sim \frac{1}{L_i} \tag{3}$$

The invading bee is gradually destroyed by heat radiation. Heat is transmitted only through radiation. The $population_{size}$ parameter indicates the initial number of besieging bees. Eq. (4):

$$\theta_0 = \frac{Fitness(s_{best})}{population_{size}} \tag{4}$$

θ_0: This parameter indicates the initial heat amount for each bee.

$Fitness(s_{best})$: This fitting an answer that is based on problem-based revelation.

$\theta(i,j)$: Heat radiated from the defending bees i to the attacker j. Eq. (5):

$$\theta(i,j) = \frac{\theta_0}{L_i} e^{-\gamma\pi r^2_{i,j}} \tag{5}$$

γ: Heat absorption coefficient, by the attacker. This variable can be equal to 1.

$\pi r^2_{i,j}$: Fitting the place in radius r.

The distance between the defending bees i and the attacking bee j can be the Euclidean distance or any other distance in the radius r.

The sum of the heat radiated from all the defending bees is given by the following equation Eq. (6):

$$\theta_{total} = N \sum_{i,j=1}^{n} \theta(i,j) \tag{6}$$

θ_{total}: The sum of the heat radiated from all the defending bees.

If the bee is initially in the radius r, the bee is likely to move at a distance x_i from the circumference of the circle. Eq. (7):

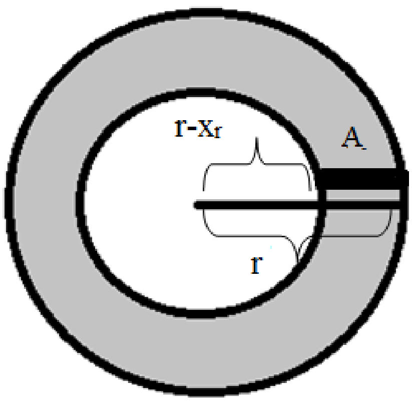


Figure 1: Move the bees to the center of the circle

$$A = \frac{\pi(r^2 - (r - x_r)^2)}{\pi r^2} = \frac{r^2 - (r^2 - 2rx_r + x_r^2)}{r^2} = \frac{2rx_r - x_r^2}{r^2} \tag{7}$$

The amount of displacement of each bee towards the center of the circle with radius r is obtained from the Eq. (8):

$$x_i^{t+1} = x_i^t - A \tag{8}$$

Update phase

To update the amount of heat radiated, we have Eq. (9):

$$\theta(i,j) = (1-\rho)\theta(i,j) + \rho\Delta\theta(i,j) \tag{9}$$

ρ: The update coefficient is equal to 0.1.

If a bee dies, it is replaced by another bee that performs better. As a result we have Eq. (10):

$$If L_i = 0 \ then \ \theta(i,j) = \theta(i,j) + \Delta\theta(i,j) \tag{10}$$

Bees that perform better replace dead bees. As a result we have Eq. (11):

$$\Delta\theta(i,j) = Fitness(s_{best}) \ if i \in N \tag{11}$$

Maximum efficiency is achieved when the number of besieging bees ($f_{population}$) and their lifespan ($f_{lifetime}$) and their heat ($f_{temperature}$) are high and they are less distance ($f_{distance}$) from the center of the circle. As a result, we have Eq. (12).

$$\begin{aligned} F_1 &= f_{lifetime} + f_{temperature} + f_{population} \\ F &= \frac{\alpha}{f_{distance}} + (1-\alpha).F_1, \quad \alpha \in [0,1] \end{aligned} \tag{12}$$

According to the explanations provided in Section 3.1, the pseudo-code of the algorithm is as shown in Fig. 2.

4. Experimental results

The swarm intelligence algorithms that have been compared with the proposed algorithm are given in Table 1.

Table 1: Compared swarm intelligence algorithms

Algorithm	Inspiration	Year
Particle swarm optimization (PSO)[20]	Bird flock	1995
Artificial bee colony optimization (ABC)[21]	Bee colony	2005
Ant colony optimization (ACO) [22]	Ant colony	2006
Firefly algorithm (FA) [22]	Firefly	2008
Grey wolf optimization (GWO)[6]	Grey wolf pack	2014
Whale optimization (WOA) [9]	Humpback whales	2016

Algorithm1 Pseudo-code of Mass Defense of Bees

```
Function MDB (problem) returns a state that is a local maximum
    Input: population size, problem size, γ, θ, ρ
    Output: Sbest
        S_best ← create best solution (problem_size);
        θ ← Initialize heat (problem_size, S_best);
    While ¬ stop condition do
        For k=1 to population_size do
            S_k ← solution (γ, θ, r, L);
            If fitness (S_k)>= fitness ( S_best) then
                S_best ← S_k;
        End;
Local update (θ,S_k, ρ);
    End;
Global update(θ, S_best);
 End;
Return S_best;
```

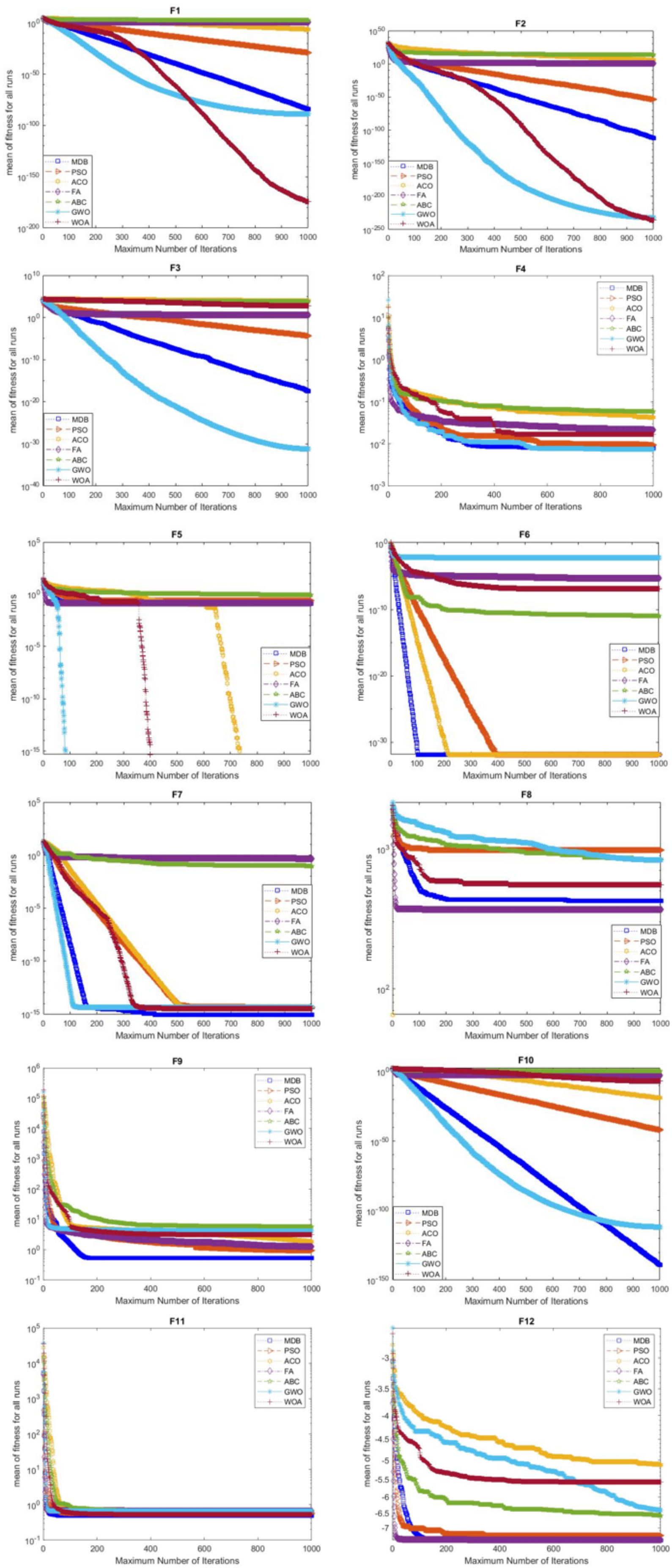
Figure 2: Pseudo code of Mass Defense of Bees algorithm

In this paper, we compare and evaluate the proposed algorithm with other well known nature inspired algorithms including ABC, ACO, PSO, FA, WOA and GWO. 20 benchmark functions have been used for comparison of algorithms. The features and characteristics of used functions are summarized in Table 2 [4, 23]. We have used Matlab for implementing the algorithms.

Table 2: The used benchmark functions

No	Formula	Bounds	D
F01	$\sum_{i=1}^{d} x_i^2$	$[-100.100]^d$	20
F02	$\sum_{i=1}^{d} \lvert x_i\rvert^{i+1}$	$[-100.100]^d$	20
F03	$\sum_{i=1}^{d}\sum_{i=1}^{i} x_i^2$	$[-65.65]^d$	20
F04	$\sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600.600]^d$	4
F05	$10d + \sum_{i=1}^{d} x_i^2 - 10cos(2\pi x_i)$	$[-5.12.5.12]^d$	4
F06	$sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + sin^2(2\pi w_d)]\ Where\ w_i = 1 + (x_i - 1)/4$	$[-5.12.5.12]^d$	4
F07	$20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)\right) + 20 + exp(1)$	$[-32.32]^d$	8
F08	$418.9829d - \sum_{i=1}^{d} x_i sin(\sqrt{\lvert x_i\rvert})$	$[-500.500]^d$	8
F09	$\sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)]$	$[-10.10]^d$	8
F10	$\sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + (\sum_{i=1}^{d} 0.5ix_i)^4$	$[-5.10]^d$	8
F11	$(x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	$[-10.10]^d$	8
F12	$-\sum_{i=1}^{d} sin(x_i)sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0.\pi]^d$	8
F13	$\sum_{i=2}^{d/4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-2})^4 + 10(x_{4i-3} + x_{4i})^4]$	$[-10.10]^d$	10
F14	$\sum_{i=1}^{d} \lvert x_i sin(x_i) + 0.1x_i\rvert$	$[-10.10]^d$	10
F15	$0.5\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i) + 39.16599d$	$[-100.100]^d$	10
F16	$x_1^2 + 10^6\sum_{i=1}^{d} x_i^2$	$[-10.10]^d$	10
F17	$0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$[-10.10]^d$	2
F18	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$[-4.5.4.5]^d$	2
F19	$f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5.5]^d$	2
F20	$f(x) = x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1 + 4\pi x_2) + 0.3$	$[-100.100]^d$	2

The simulation results and convergence behavior of the algorithms for the benchmark functions is shown in Fig. 3 (F1–F20).


F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
mean of fitness for all runs
Maximum Number of Iterations
MDB
PSO
ACO
FA
ABC
GWO
WOA

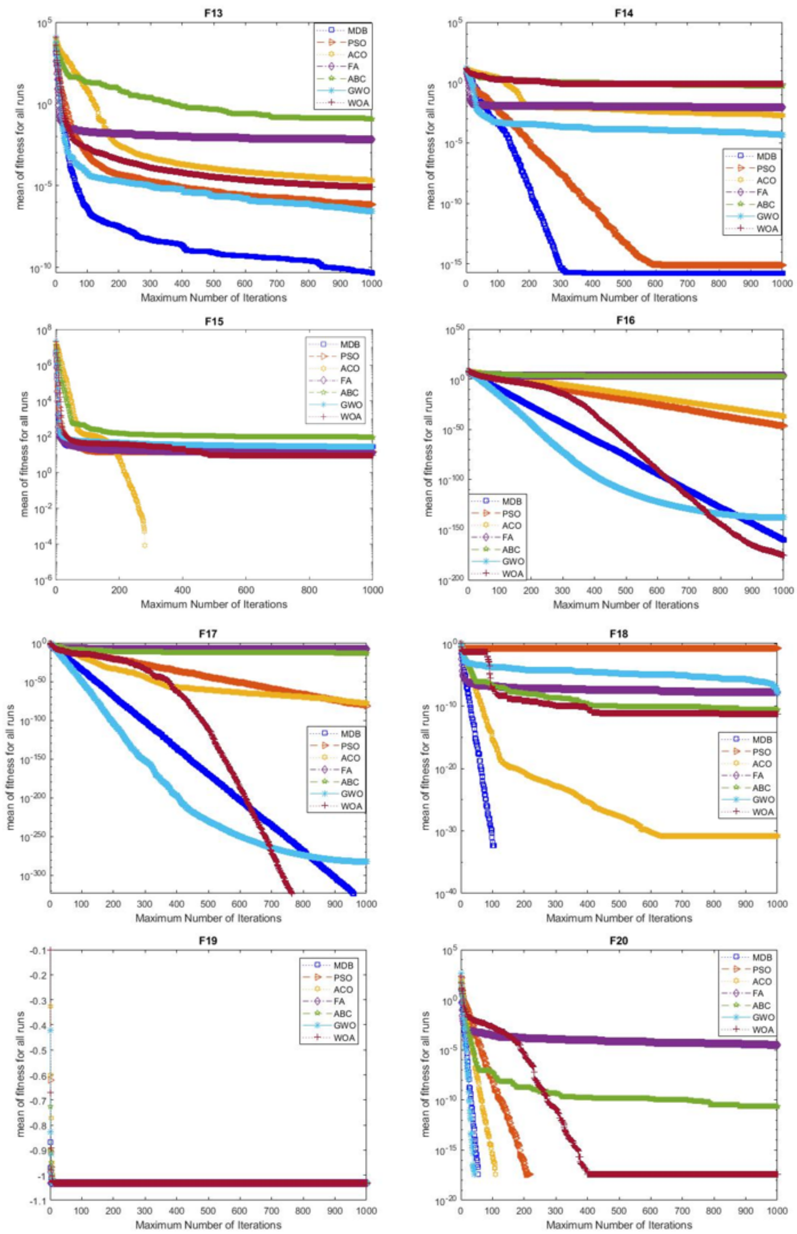


Figure 3: The simulation results and convergence behavior of the algorithms for the benchmark functions (F1 – F20).

The proposed algorithm is very competitive in terms of convergence of different functions. This algorithm searches the entire space to find the target point and searches the solution space more carefully and after finding it, offers the best solution. This algorithm converges quickly and provides clear and coherent results for a certain number of iterations. The proposed algorithm has given good results in comparison with other algorithms in the number of given iterations.

As seen from the results in Fig. 3, the proposed algorithm is superior or competitive in most of the test functions. It converges to better solutions in most of the test functions or at least converges faster than other algorithms for global optimization solutions. However, the algorithms follow the No Lunch Free theorem [4]. According to the No Lunch Free theorem, there is no optimization algorithm that is suitable for all types of problems. Therefore, for some functions, the proposed algorithm provides better results, and for some functions, other algorithms provide better results. But overall, it seems that this algorithm is an extraordinary algorithm that can be a solution to the local optimal problem. The MDB algorithm is a desirable optimization tool. Part of this property is due to meet multivariate goals in achieving convergence that is unique to bee behavior.

As mentioned before, setting parameters are very important for collective intelligence algorithms.

5. Conclusions

In this paper, the MDB swarm intelligence algorithm is presented based on the mass defensive behavior of bees. This algorithm has been tested for 20 basic functions and has been compared with other modern optimization algorithms such as WOA, GWO, PSO, ACO, ABC and FA. The simulation results show that the MDB algorithm provides good exploration and productivity, which are the main success factors in swarm intelligence methods. In this algorithm, several different parameters such as temperature, distance, number and lifespan of bees are effective to achieve convergence, and this feature means that the selection of several criteria has led to the flexibility of the algorithm in setting parameters to achieve. The bee collective defense algorithm can be used to solve multi-objective optimization problems and the problems of WSN and Internet of Thing networks and their clustering and other sciences that require flexibility and coordination between multiple parameters. In addition, the combination of this algorithm with other algorithms will play an important role in future research.

References

[1] M. Arca, A. Papachristoforou, F. Mougel, A. Rortais, K. Monceau, O. Bonnard, P. Tardy, D. Thiéry, JF. Silvain and G. Arnold, *Defensive behaviour of Apis mellifera against Vespa velutina in France: testing whether European honeybees can develop an effective collective defence against a new predator*, Behavioural Process. 1(106) (2014) 9–122.

[2] M. Alam, A, khan and I. Khan, *Swarm intelligence in Menets: a survey*, Int. J. Emerg. Res. Manag. Technol. 5(5) (2016) 141–150.

[3] B. Bai, Z. Guo, C. Zhou, W. Zhang and J. Zhang, *Application of adaptive reliability importance sampling-based extended domain PSO on single mode failure in reliability engineering*, Inf. Sci. 546 (2021) 42–59.

[4] D. Bairathi and D. Gopalani, *A novel swarm intelligence based optimization method: harris hawk optimization*, Int. Conf. Intell. Syst. Design Appl. 2018, pp. 832–842.

[5] J. Bansal and S. Singh, *A better exploration strategy in grey wolf optimizer*, J. Ambient Intell. Humanized Comput. 12(1) (2021) 1099-1118.

[6] O. Bello and S. Zeadally, *Communication issues in the Internet of Things (IoT)*, In Next-Generation Wireless Technologies, Springer, London, (2013) 189–219.

[7] E. Bonabeau, M. DDRDF, M. Dorigo, G Théraulaz and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.

[8] B. Cao, X. Wang, W. Zhang, H. Song and Z. Lv, *A many-objective optimization model of industrial internet of things based on private blockchain*, IEEE Network. 34(5) (2020) 78–83.

[9] B. Cao, J. Zhao, Y. Gu, Y. Ling and X. Ma, *Applying graph-based differential grouping for multiobjective large-scale optimization*, Swarm Evol Comput. 53 (2020) 100626.

[10] G. Ding, Y. Qiao, W. Yi, W. Fang and L. Du, *Fruit fly optimization algorithm based on a novel fluctuation model and its application in band selection for hyperspectral image*, J. Ambient Intell. Humanized Comput. 12(1) (2021) 1517–1539..

[11] L. Ding, S. Li, H. Gao, Y. Liu, L. Huang and Z. Deng, *Adaptive neural network-based finite-time online optimal tracking control of the nonlinear system with dead zone*, IEEE Trans. Cybernet. 2019.

[12] M. Dorigo, M. Birattari and T. Stutzle, *Ant colony optimization*, IEEE Comput. Intell. Mag. 1(4) (2006) 28–39.

[13] A. Gandomi and A. Alavi, *Krill herd: a new bio-inspired optimization algorithm*, Commun. Nonlinear Sci. Numerical Simul. 17(12) (2012) 4831–4845.

[14] J. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, Evol. Comput. 8(4) (1992) 373–391.

[15] B. Hu and B. Yang, *A particle swarm optimization algorithm for multi-row facility layout problem in semiconductor fabrication*, J. Ambient Intell. Humanized Comput. 10(8) (2019) 3201–3210.

[16] M. Janah and Y. Fujimoto, *Study comparison between firefly algorithm and particle swarm optimization for SLAM problems*, Int. Power Electron. Conf. (IPEC-Niigata 2018-ECCE Asia), 2018, pp. 681–687.

[17] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proc. ICNN'95-Int. Conf. Neural Networks, 1995, pp. 1942–1948.

[18] J. Liu, C. Wu, G. Wu and X. Wang, *A novel differential search algorithm and applications for structure design*, Appl. Math. Comput. 268 (2015) 246–269.

[19] S. Liu, F. Chan and W. Ran, *Decision making for the selection of cloud vendor: An improved approach under group decision-making with integrated weights and objective/subjective attributes*, Expert Syst. Appl. 55 (2016) 37–47.

[20] Z. Lv and L. Qiao, *Deep belief network and linear perceptron based cognitive computing for collaborative robots*, Appl. Soft Comput. 92 (2020) 106300.

[21] H. Ma, L. Xu and G. Yang, *Multiple environment integral reinforcement learning-based fault-tolerant control for affine nonlinear systems*, IEEE Trans Cybernet. 2019.

[22] M. Mavrovouniotis, C. Li and S. Yang, *A survey of swarm intelligence for dynamic optimization: Algorithms and applications*. Swarm Evol. Comput. 33 (2017) 1–17.

[23] S. Mirjalili, S. Mirjalili and A. Lewis, *Grey wolf optimizer*, Adv. Engin. Software 69 (2014) 46–61.

[24] S. Mirjalili and A. Lewis, *The whale optimization algorithm*, Adv. Engin. Software 95 (2016) 51–67.

[25] S. Thennarasu, M. Selvam and K. Srihari, *A new whale optimizer for workflow scheduling in cloud computing environment*, J. Ambient Intell. Humanized Comput. 12(3) (2021) 3807–3814.

[26] B. Wang, B. Zhang, X. Liu and F. Zou, *Novel infrared image enhancement optimization algorithm combined with DFOCS*, Optik 224 (2020).

[27] D. Wang, D. Tan and L. Liu, *Particle swarm optimization algorithm: an overview*, Soft Comput. 22(2) (2018) 387–408.

[28] X. Wang, Y. Zhan, L. Wang and L. Jiang, *Ant colony optimization and ad-hoc on-demand multipath distance vector (AOMDV) based routing protocol*,. Fourth Int. Conf. Natural Comput. 2008, pp. 589–593.

[29] X. Yang, *Firefly algorithms for multimodal optimization*, Int. Symp. Stoch. Algorithms 2009, pp. 169–178.

[30] A. Zengin and S. Tuncel, *A survey on swarm intelligence based routing protocols in wireless sensor networks*, Int. J. Phys. Sci. 5(14) (2010) 2118–2126.

[31] L. Zhang, L. Liu, X. Yang and Y. Dai, *A novel hybrid firefly algorithm for global optimization*, PloS One. 11(9) (2016).