A New Multi-agent Approach for Solving Optimization Problems with High-dimensional: Case Study in Email Spam Detection

Hekmat Mohmmadzadeh, Farhad Soleimanian Gharehchopogh*

Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
Corresponding Author * {bonab.farhad@gmail.com}

Abstract

There exist numerous high-dimensional problems in the real world which cannot be solved through the common traditional methods. The metaheuristic algorithms have been developed as successful techniques for solving a variety of complex and difficult optimization problems. Notwithstanding their advantages, these algorithms may turn out to have weak points such as lower population diversity and lower convergence rate when facing complex high-dimensional problems. An appropriate approach to solve such problems is to apply multi-agent systems along with the metaheuristic algorithms. The present paper proposes a new approach based on the multi-agent systems and the concept of agent, which is named Multi-Agent Metaheuristic (MAMH) method. In the proposed approach, several basic and powerful metaheuristic algorithms, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Firefly Algorithm (FA), Bat Algorithm (BA), Flower Pollination Algorithm (FPA), Gray Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Crow Search Algorithm (CSA), Farmland Fertility Algorithm (FFA), are considered as separate agents each of which sought to achieve its own goals while competing and cooperating with others to achieve the common goals. In overall, the proposed method was tested on 32 complex benchmark functions, the results of which indicated effectiveness and powerfulness of the proposed method for solving the high-dimensional optimization problems. In addition, in this paper, the binary version of the proposed approach, called Binary MAMH (BMAMH), was executed on the spam email dataset. According to the results, the proposed method exhibited a higher precision in detection of the spam emails compared to other metaheuristic algorithms and methods.

Keywords: *Multi Agent Systems, High-dimensional, Optimization, Email Spam, Metaheuristic Algorithms*

1. Introduction

In recent years, the optimization problems are widely used in various fields. In these problems, the search space is exponentially increased with the problem dimensions [1] leading to higher complexity of the problem. Meanwhile, most of the real optimization problem is inherently big and complex resulting in the growing use of the complex high-dimensional optimization problems in a wide range of scientific applications [2, 3]. Accordingly, solving a high-dimensional problem is supposed as a considerable challenge, which has attracted the researchers' attention toward solving such problems in recent years. A high-dimensional optimization problem can be defined as a D-dimensional problem as ***Equation (1)*** [2, 3]:

$$\begin{cases} \min & f(x) \\ S.T. & lower \leq x \leq upper \end{cases} \qquad (1)$$

Where $f(x)$ is the objective function or optimization problem, $x= (x_1, x_2, x_3, \ldots, xD)$ indicates the variables of the problem with D dimensions, and $l= (l_1, l_2, l_3, \ldots, lD)$ and $u= (u_1, u_2, u_3, \ldots, uD)$ indicate the lower and upper bounds of each variable, respectively. Since such problems cannot be simply solved using the common traditional methods [4], the metaheuristic methods have been introduced as an alternative to the traditional optimization techniques. The metaheuristic methods are characterized with the advantage of finding the viable solutions over a reasonable time as well as high tolerance to difficult and complex problems [5].

The metaheuristic algorithms are considered among the most promising methods for solving the optimization problems. The investigations and studies by several researchers on metaheuristic algorithms have yielded various nature-inspired and non-`e-inspired algorithms, some of which include: GA [6], PSO [7], ABC [8], FA [9], BA [10], GWO [11], WOA [12], and FFA [4], etc. The main advantage of these algorithms is the use of the "*trial and error*" principle in searching for the solutions, as a result of which they have been used successfully for solving the global optimization problems. The metaheuristic algorithms mostly start with a random initial population and then, through a certain number of iterations, try to improve or change that population in accordance with the given objective function. As stated by Mirjalili et al., the metaheuristic algorithms are brought up to solve the optimization problems more than other methods due to four reasons [11]. First, these algorithms involve simple mathematical operators or processes that can be easily used in an optimization problem. Second, they have such flexibility that can be used for solving the optimization problems while entailing minimal changes in the problem. Third, since these algorithms start with finding random solutions for the optimization problems, there will be no need to find the optimal solution by calculating the derivative of the search space. And finally, the metaheuristic algorithms escape from the local optimum due to having more powerful processes in comparison with the traditional methods.

Notwithstanding the above-mentioned advantages and strengths, the metaheuristic algorithms have also some disadvantages. For instance, the increased problem dimensions may result in the lower population diversity or lower convergence rate of the algorithm. Besides, selection of the appropriate metaheuristic algorithm for solving an optimization problem is another challenge in this regard [13, 14]. However, these can be overcome and resolved by using a multi-agent system, which is a solution that divides a complex problem into several simpler ones. In a multi-agent system, the agents, as a community, attempt to achieve their own specific goals along with their competition and cooperation for achieving the common goals. In fact, an agent is a system that, by recognizing its surrounding environment, increases its position chance after analysis and examination of the environment. On this basis, in the present paper, a multi-agent system is used to solve the high-dimensional problems, so that each metaheuristic algorithm is considered as an agent. The objective of this paper is focused on the collaboration of several metaheuristic algorithms as well as prevention of the early or late convergence and escaping from local optimum.

This paper is aimed to present a new multi-agent system-based approach, which relates the concept of agents to the metaheuristic algorithms, and define an MAMH approach as the proposed method associated with the flowcharts and pseudocodes. In the multi-agent based approach, multiple basic and powerful algorithms have been used as separate individual agents. However, in the proposed method, it was attempted to use the operators of the algorithms independently; then, the algorithms were weighted in accordance with the power of the operators of each algorithm. Therefore, each individual agent seeks for its own goals while competing and cooperating to achieve the common goals. Since the MAMH approach has been designed for continuous problems, we have introduced another version of the approach, called BMAMH, based on the sigmoid transform function. In this approach, in order to modify the continuous position of the solutions in the shared memory, the sigmoid function is transformed form the zero-one mode to binary mode.

The rest of the paper is organized as follows. Section (2) deals with a review of the literature on the high-dimensional optimization problems. In Section (3), the multiple agents are described. The proposed method, including its theory, flowchart, and formulation, will be introduced in Section (4). Then, Section (5) addresses the examination and testing of the proposed method's efficiency and performance using the standard benchmark functions, the results of which are recorded and shown in graphical diagrams. And finally, Section (6) includes the general conclusion as well as some suggestions for future works.

2. Related Works

This section provides a review of the studies conducted on solution of the high-dimensional optimization problems. In the past decades, the researchers have growingly tended toward the use of metaheuristic methods for solving the high-dimensional optimization problems. For instance, ***Chu et al. (2008)*** proposed a *bacterial foraging optimization* ***(BFO)*** algorithm for solving the high-dimensional functions [15]. The proposed algorithm incorporates a combination of the advantages of the BFO and PSO algorithms yielding an improved convergence in solving the high-dimensional optimization problems. Of course, a new parameter has been introduced for adjusting the bacteria's path with regard to the best solution. Finally, the proposed method was tested on the standard benchmark functions in order to be compared with the PSO and BFO algorithms, the results of which indicated the proposed method's capability to improve the convergence rate and optimization precision. In [7], ***Achting*** used PSO in order to solve the high-dimensional optimization problems. The obtained results showed that a combination of the PSO algorithms with some concepts of the evolutionary algorithm, such as mutation operator, would significantly improve the PSO algorithm's performance in solving the high-dimensional optimization problems. In another study, ***Zhao et al. (2010)*** introduced a self-adaptive differential evolution algorithm with incorporated *multi-trajectory search (MTS)* and *JADE* mutation strategy for solving the high-dimensional optimization problems [16].

In [17], a new evolutionary search strategy was introduced for solving the global high-dimensional optimization problems. It showed that the evolutionary algorithms suffer from poor performance in searching for a high-dimensional space. Thus, a new search strategy, named SP-UCI, was introduced. This strategy featured a slope-based searching kernel and a scheme of maintaining the particle population's capability of searching over the full search space. The obtained results showed that the proposed method performed better than the differential evolution and PSO algorithms. In another study, ***Wang et al. (2013)*** presented a GPU-based parallel differential evolution algorithm with self-adaptive control parameters and disagreement-based learning for solving the high-dimensional optimization problems [18]. In this method, the self-adaptive parameters were used to prevent manual adjustment of the parameters, and the disagreement-based learning was used to improve the quality of the solutions. The obtained results indicated the superiority of the proposed method over the other six methods. In [19], in order to compensate the weak points of the PSO algorithm, the authors proposed an elite-grouped adaptive PSO algorithm for solving the high-dimensional optimization problems. In this method, the bad particles are changed by the elite particles leading to the increased population diversity and avoidance from falling into the local optima. The simulation results showed the proposed method's superiority over PSO.

Imanian et al. (2014) introduced a velocity-based ABC algorithm, named VABC, for solving the continuous high-dimensional optimization problems [20]. In this paper, it was claimed that the ABC suffered from low convergence rate and productivity, and thus the authors proposed a new equation in the *onlooker* phase based on the PSO strategy. As indicated by the empirical results of the tests on numerical benchmark functions, the proposed algorithm had good performance in comparison with the PSO and ABC algorithms. In another study, the researchers proposed a quantum particle swarm optimization algorithm based on a combination of the *simulated annealing* algorithm, *co-evolution* theory, and *diversity-guided* mutation for solving thigh-dimensional optimization problems [21]. The results indicated high levels of search precision and convergence rate for the proposed algorithm. ***Li et al.*** [3] introduced a hybrid algorithm called PS-ABC based on the ABC and PSO algorithms for solving the high-dimensional optimization problems. It used one local search phase of PSO and the two phases of *onlooker* and *improved scout bee* for global search. The obtained results indicated the proposed algorithm's effectiveness in solving the high-dimensional optimization problems.

In [22], ***Tuo et al.*** proposed a harmony search algorithm, called DIHS, for solving the multi-modal high-dimensional optimization problems, in which the two parameters *dynamic dimensionality reduction adjustment* and *dynamic FRET width* were used to avoid generation of invalid solutions and to balance between exploration and productivity, respectively. As indicated by the obtained results, the proposed method could provide more precise solutions in a shorter period of time. In another work, ***Mohapatra et al. (2017)*** introduced a modified competitive swarm optimizer (MCSO) for solving the large-scale optimization problems [23]. In this method, two thirds of the populations are updated by the *tri-competitive* criterion unlike the CSO. The main idea of this method is to maintain higher speed of discovery in the search space with higher convergence rate. In [24], ***Long et al.*** proposed a GWO algorithm with enhanced exploration, named EEGWO, for solving the high-dimensional numerical problems. In this work, in order to enhance the exploration, a new equation was used for position updating in; further, the balance between exploration and productivity was achieved by the nonlinear control parameter strategy. As shown by the comparison results, the proposed EEGWO algorithm could considerably improve the performance of the GWO algorithm. In [2], an improved sinusoidal algorithm, named ISCA, was proposed for solving the high-dimensional global optimization problems, in which a modified updating position equation was used for accelerating the convergence and a nonlinear parameter for maintaining the balance and exploration. The comparison results showed the ISCA method to be an effective method for solving the high-dimensional optimization methods.

3. Multi-agent Systems

Apropos of artificial intelligence, the scholars have not reached a consensus on a generally-accepted definition for "*agent*". An example of such definitions can be as "*agents can appear in a variety of physical forms ranging from robots to computer networks, which are dissimilar*" [25]. Therefore, *agent* is a concept in the subject areas of artificial intelligence and computer sciences, so that any agent is either a computer system or a human in a particular environment. Typically, the agents receive the information through their inputs and then, based on this information, begin to act upon their environment, which may be carried out independent from the other agents' actions. The *agent-based computation* is a broad scientific field of study, in which an agent can be a "*software agent*", a "*service*" [25], or any other thing that can be interrelated in some way with other agents in the environment [26].

In accordance with [25], which is a credible and valid reference, some of the major distinguishing features of an agent against the simple controllers include:

- ***Situatedness:*** It refers to an agent's interaction with its environment through sensors and operator actions. For any environment containing an agent, this feature is considered as an inevitable component of its design through which the inputs can be received directly.
- ***Autonomy:*** It is considered as the agent's independence or self-administration and can be defined as the agent's capability to select its own actions needless to any external intervention either from the other agents within the network or from humans.
- ***Inferential capability:*** It refers to the capability of an agent to act upon the abstract target features. For example, inferring an observation can be done by using a relevant content of the available information.
- ***Responsiveness:*** It means the capability of perceiving the conditions of an environment and giving a timely response in order to take into account any kind of change within the environment.

- ***Pro-activeness:*** An agent should react appropriately in order to increase the goal-based tasks. The agent should have the capability of not only giving a specific reaction to a particular change within the environment but also adapting itself with the emerging changes in a dynamic environment.
- ***Social behavior:*** A decision made by an agent should be independent and free from any external intervention. Nevertheless, when it comes to be in need for achieving a specific goal, the agent must have relationships and interactions with external sources; besides, it should be able to share such knowledge and help others agents solve a particular problem. In addition, each agent interacts with other agents as well as its external environment through social behavior (***Fig.1***) [26, 27].

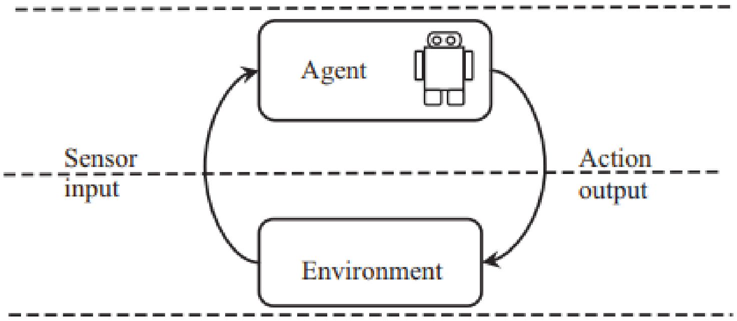


***Fig. 1**. A general view of an agent*

In a multi-agent system (MAS), a groups of interconnected independent agents act in an environment in order to achieve a common goal. The agents accomplish such a task through cooperation or competition with or without sharing their knowledge. In fact, a MAS is a system composed of multiple intelligent (smart) agents that are interconnected. These systems can be used for solving those problems that are impossible to be solved simply by an agent or an integrated system [28]. The agents may have a common goal or may seek for their own goals; also, the agents can change their situation, which is accomplished by incorporating the information and sources in the whole system [13, 28]. A multi-agent system is indeed a group of autonomous agents that are interconnected and act in a certain environment. An MAS is typically consisted of three elements [29]: a set of agents $A=\{a_1, a_2, \ldots, a_n\}$, an environment in which the agents live and act, and a set of reaction rules that control the interactions between the environment and the agents. Below, the reasons for using a multi-agent system, or in other words the advantages of using a multi-agent system, are described according to [25]:

- Increased operation efficiency and speed due to the parallel computations and simultaneous operation;
- Increased reliability and sustainability of the system;
- Scalability and flexibility for adding the agents in case of need;
- Reduced costs due to the fact that the individual agents incur lower costs than a centralized architecture;
- Substitutability: the agents have modular structure and, thereby, can be easily substituted in other systems or easily upgraded from an integrated system.

4. Proposed Method

This section is aimed to propose a multi-agent approach based on the metaheuristic algorithms for solving the high-dimensional optimization problems besides a binary model of the proposed approach for solving the binary problems. Seeking to facilitate a better conception of the proposed approach, the continuous and discrete versions of the model are described separately. Therefore, this section is divided into two subsections (4-1) and (4-2), which describe a multi-agent system based on the metaheuristic algorithms, named MAMH, and the binary version of the proposed MAMH method, named BMAMH, respectively.

4-1 Continuous version

Constructing a multi-agent system requires several attributes or concepts, as the main attributes or concepts, relying on which the system will be able to work properly. The major attributes of a multi-agent system include the agents, the environment containing the agents, the independent, social, and self-learning behaviors of each agent with regard to the individual or common goals, as well as the relationships between the agents. On the basis of the main attributes of a multi-agent system, we are going to present and describe a multi-agent approach based on the metaheuristic algorithms, named MAMH, which is designed for solving the high-dimensional optimization problems. To provide a general view of the proposed method, it is schematically illustrated in ***Figure (2)***.

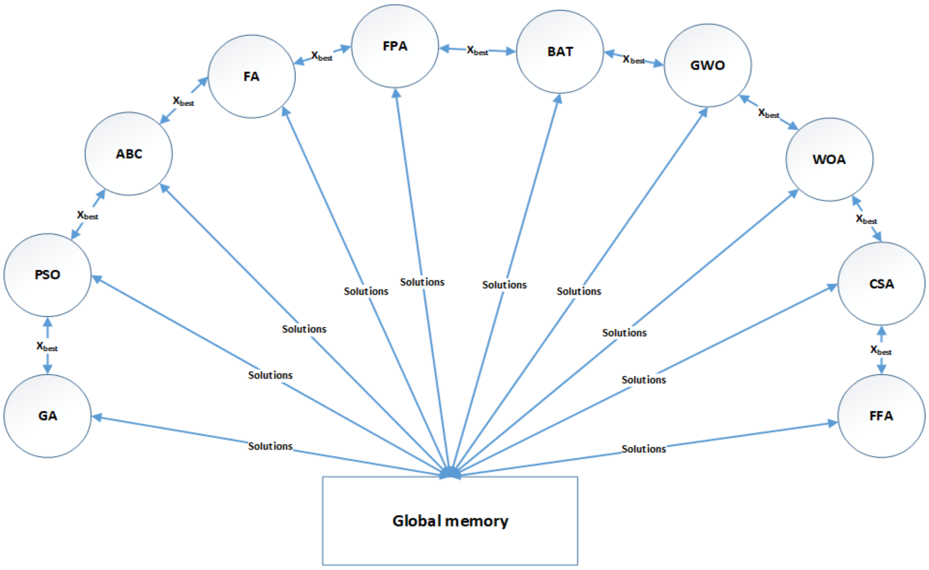


Fig. 2. *Overview of the proposed MAMH approach based on multi-agent system*

In the proposed method, each agent represents a metaheuristic algorithm. Each of these agents will be described hereinafter. ***Figure (2)*** demonstrates the relationships between the agents that are organized via architecture so that the agents and their relationships constitute an environment. In the present paper, a structure with *K* metaheuristic algorithms has been used, the population of each of which is considered as *N*. For all of these algorithms, the best solution is interchanged between the algorithms. In this paper, a total of 10 different metaheuristic agents have been used; besides, it has been attempted to adopt various metaheuristic algorithms that involve different processes and belong to different years. These algorithms associated with a general definition for each agent are provided below:

- ***GA***: It is one of the most basic evolutionary and metaheuristic algorithms, which is inspired by and based on the *"theory of natural selection"* and includes three main operators, namely *mutation*, *crossover*, and *selection*, as well as process that drives the biological evolution [30].
- ***PSO***: It is a swarm intelligence-based algorithm inspired by social behavior of the bird flocks. In this algorithm, each solution is defined by the vectors of *velocity* and *position* in the search space [31].

- ***ABC***: This algorithm, inspired by the honey bees' behavior in nature, was introduced by ***Karaboga et al.*** in 2005. The *artificial bee colony algorithm* divides the artificial bees into three groups including *employed bees*, *onlookers*, and *scouts* [32].
- ***FA***: This algorithm, the main idea of which has been inspired by the flashing behavior of the fireflies, was introduced by ***Yang et al.*** in late 2007 [9].
- ***BA***: The *BA* is a metaheuristic optimization algorithm presented by ***Yang*** in 2010. In fact, the bats behave based on three criteria, including *echolocation*, *pulse rate of emission*, and *loudness* [10].
- ***FPA***: Introduced by ***Yang*** in 2012, this algorithm has been inspired by the flowers' pollination process that is carried out by honey bees in nature and includes two operators for the local and global optimization processes [33].
- ***GWO***: This algorithm was introduced by ***Mirjalili et al.*** in 2014. It is based on the leadership hierarchy and hunting mechanism of the gray wolves in nature. In general, there are four groups of wolves among the gray wolves, namely *alpha*, *beta*, *delta*, and *omega* [11].
- ***WOA***: The WOA algorithm, which was introduced by ***Mirjalili et al.*** in 2016, has been inspired by the hunting behavior of humpback whale. WOA includes three operators for simulating the search for prey, encircling the prey, and bubble-net foraging behavior of the humpback whales [12].
- ***CSA***: The CSA algorithm, introduced by ***Askarzadeh et al.*** in 2016, has been designed and implemented based on the idea that the crows store their food excess in some nearby hidden places to retrieve it when the food is needed [34].
- ***FFA***: The FFA algorithm is a novel metaheuristic algorithm inspired by the farmland fertility in nature, which was introduced by ***Shayanfar & Gharehchopogh*** in 2018 [4]. This algorithm, by dividing the farmland into several sections, attempts to optimize the solutions of each section with optimal productivity (efficiency) from the two types of internal and externa memory.

One of the key features of the multi-agent systems is that the agent works cooperatively and coordinately with each other so that it can be considered as a kind of social behavior. The MAMH method assumes two connections for coordination, in the first of which all agents are joined to a shared memory and in the second one; the algorithms interchange the best solution found in each iteration. As such, each agent can be executed individually and independently within itself and meanwhile interconnect with others through the two above-mentioned ways. Moreover, each agent, in accordance with the process defined inside it, would yield local learning in other agents through its own knowledge of the given problem. ***Figure (3)*** shows the pseudocode for the proposed method based on the multi-agent condition. To provide a better understanding of the proposed method, it will be explained step by step below:

```
Begin
01: Global Memory (Generate initial population of solution $X_i$ (i=1... N)) by Equation (2);
02: Initialize parameters D, maxit, varmin, varmax, …
03: Calculate the fitness of each solution by f ($X_i$);
04: X*=The best Solution
05: Assign solutions to agents by Equation (3);
06: Calculate the mean fitness of each agent by Equation (4);
07: While (t < maximum number of iterations)
****Run each agent with new population****
        07-1: run agent one (GA   operators)
        07-2: run agent two (PSO   operators)
        07-3: run agent three (ABC   operators)
        07-4: run agent four (FA   operators)
        07-5: run agent five (FPA   operators)
        07-6: run agent six (BA   operators)
        07-7: run agent seven (GWO   operators)
        07-8: run agent eight (WOA   operators)
        07-9: run agent nine (CSA   operators)
        07-10: run agent ten (FFA   operators)
****   End all agent****
08: Calculate the mean fitness of each agent by Equation (4);
09: Update global memory by Equations (5:7);
10: Update solutions to agents by Equation (8:12);
11: Update X* if there is a better solution
12: t=t+1
13: End While
14: Print results.
15: End
```

***Fig. 3**. Pseudocode the proposed MAMH approach based on multi-agent system*

Now, let's explain the algorithm presented here.

Step 01 generating the initial population or shared memory: In this step, regarding the problem and objective function, the random initial population is generated and situated in the shared memory. The highest value of each dimension, lowest value of each dimension, objective function of the optimization problem, dimensions of the optimization problem, and highest number of iterations of each agent (here, each metaheuristic algorithm represents an agent) are determined in this step. The random solutions of the search space, or in other words the shared memory, can be generated using ***Equation (2)***:

$$M_{ij} = L_j + rand(0.1) \times (U_j - L_j) \qquad (2)$$

Where, U_j and L_j represent the upper and lower bounds of each dimension of the variable M that has been primarily specified through the benchmark function or the optimization problem, *rand* indicates the function of random numbers in the (0,1) interval, $j=[1, \ldots, D]$ indicates the dimension of the variable M, and $i=[1, \ldots, N]$ refers to each member of the available solutions existing in the shared memory.

Step 05 assigning the solutions to the agents: In this step, some solutions are assigned to each agent. These steps along with the “updating the population of each agent” step are the most important parts of the proposed approach; therefore, for the first time, all of the agents are assigned with an identical number of the available solutions in the shared memory. This can be defined by ***Equation (3)***:

$$PAgent_i = M_{Lj}; j = [1 \ldots D]; L = N/K \qquad (3)$$

Where, N is the total population or total number of available solutions in the shared memory, K is the number of agents existing in the environment, L is the number of solutions for each agent, M is the shared memory, and $PAgent_i$ is the population assigned to the i^{th} agent. This equation is used merely for the initial assignment of the available solutions in the shared memory to the agents and subsequently, in the "*updating the population of each agent*" step, a different population will be assigned to each agent in accordance with its performance. On this basis, in the proposed approach, once each agent is executed separately, some of the agent might be recognized as having better performance and some others as having worse performance; accordingly, the agent with better performance will be assigned with more solutions and the one with poor performance will be assigned with less solutions. Hence, prior to changing the solutions by each agent, the mean population of all agents should be calculated using ***Equation (4)***:

$$Fit\_Agent_i = Mean(\text{ all Fit}(x)\ in\ PAgent_i\ )\ .\ i = \{1.2. \dots . k\} \quad (4)$$

Where, *Fit_Agent$_i$* represents the value that specifies the quality of the solutions of each agent, so that each section is assigned with a specific value for its own. However, in the search space, this value is indeed the same as the mean fitness value of all solutions assigned to each agent. Thus, for each agent, the mean value of all of the solutions assigned to an agent is calculated and then stored in *Fit_Agent$_i$*. As a result, a certain number of solutions have been assigned to each agent.

Step 07 executing the agents: This step involves execution of each of the predefined metaheuristic agents on its own assigned solutions. As such, in this step, each agent would have a separate population as well as a series of separate processes for improving its population. In the proposed method, a total of 10 metaheuristic agents have been used, which are shown in ***Figure (2)*** and defined separately in ***Subsection (4-1)***. These agents are an indeed separate metaheuristic algorithm, each of which belongs to a different year and has its own specific features, solutions, and operators. Regarding the fact that the agent's population might be changed and improved after execution, the new mean population of all agents after execution of all of them can be calculated using ***Equation (4)***. Importantly, if an agent fails to improve its solutions, it will be assigned with the minimum number of solutions in the subsequent steps.

Step 09 updating the shared memory: In this step, the shared memory should be updated in accordance with the new population of the ten metaheuristic algorithms. For this purpose, the new population of each agent is stored in combinatorial form in a new memory named *Mem$_{Agent}$*, and then this memory is combined with the shared memory. Subsequently, the newly created shared memory is sorted and then selected with the same size as the original shared memory. ***Equations (5) & (6)*** are used to specify the way of updating the shared memory and combining it with the new memory:

$$\text{Mem}_{Agent} = [PopAgent_1 . \dots . PopAgent_i]; i = [1 \dots K]; \quad (5)$$

$$GlobalMemory_{new} = Sort[GlobalMemory_{old} . \text{Mem}_{Agent}]; \quad (6)$$

In ***Equation (5)***, *PopAgent* indicates the population of each metaheuristic agent, *K* indicates the number of agents existing in the environment, and *Mem$_{Agent}$* indicates the new memory in which the new solutions obtained from all agents are stored. In the next step, this new memory is combined with the old shared memory (*GlobalMemory$_{old}$*) and then sorted based on the objective function. Finally, both new and old solutions are stored in a new memory, named *GlobalMemory$_{new}$*, and then the updating of this shared memory is finalized using ***Equation (7)***:

$$GlobalMemory = GlobalMemory_{new}(1{:}N) \quad (7)$$

In ***Equation (7)***, the shared memory is updated based on all of the solutions existing in the *GlobalMemory$_{new}$*. Notice that since the *GlobalMemory$_{new}$* has been already sorted, the solutions 1 to *N* are selected as the final solutions. Therefore, the solutions with better values in terms of objective function are selected as the final solutions of the shared memory.

Step 10 updating the population of each agent: The proposed method has been designed in such a way that the agent with powerful processes will be assigned with a greater population of solutions while a smaller population of the solutions will be assigned to an agent that has weak processes. This step requires formulation in order for distinguishing the weak and powerful agents based on the *Fit_Agent* criterion. The first thing to do is to detect those agents that have yielded no improvement. In such case, the agents that have achieved a very slight improvement are assigned with a new population according to ***Equation (8)***:

$$PAgent_b = N_B \times \frac{1}{b}; b = [1: BAC] \tag{8}$$

In ***Equation (8)***, a certain percentage of the population existing in the shared memory is assigned to the weak agents, so that all of the bad agents with no improvement in the population are assigned with a certain portion of the population. In this equation, *BAC* represents the bad agents in the environment, and N_B is a parameter that can be adjusted based on the population of good agents. The size of population for good and bad agents can be determined using ***Equation (9)***:

$$N_G = N * \alpha; N_B = N - N_G \tag{9}$$

Where, N_G is the number of solutions assigned to all of the good or improved agents, N_B is the number of solutions assigned to all of the bad or unimproved agents, and α is a control parameter that is initiated at the beginning of the algorithm and has a value ranging between 0-1. Therefore, each of the bad agents takes a constant size of the population while the good agents yielding improvement in *Fit_Agent* are assigned with a greater population, proportionate to the degree of their improvement, according to the ***Equations (10)*** and ***(11)***:

$$PopgoodAgent_i = N_G * P_i; i = [1 \dots GAC] \tag{10}$$

$$P_i = \frac{w_i}{sum(W)}; W = [w_1 . w_1 . \dots . . w_k] \tag{11}$$

In ***Equation (10)***, *PopgoodAgent*$_i$ indicates the population of a good agent with the i^{th} counter, P_i is the percentage of improvement for each good agent that can be obtained by ***Equation (11)***, *GAC* is the number of good agents, w_i indicates the exact value of improvement for each agent, and *W* represents the total amount of improvement of all agents. As for the number of good and bad agents, it must be noted that sometimes it might be the case that none of the agents show improvement, which means that all of them are bad agents, or vice versa. If so, there will be no need for dividing the agents and, thereby, assignment of the solutions to the agents will be performed automatically and wholly.

4-2 Binary Version

In ***Subsection (4-1)***, we proposed a new method for solving the continuous high-dimensional optimization problems based on the multi-agent systems. In addition to the continuous optimization problems, there may be some binary high-dimensional problems, such as feature selection, that require a powerful algorithm. Since any continuous method can move in the binary space through applying some changes and conversions, it is supposed in this subsection to describe the binary version of the proposed approach named BMAMH, the continuous version of which was described in the ***Subsection (4-1)***. There are several different methods for converting the continuous methods to binary state. A common method for this purpose is the use of *sigmoid function*, which is called *S-shaped* [35, 36] in most of the references in literature. Most of the researchers tend to use this function for converting the continuous solutions to zero-one space. The *sigmoid* function is illustrated schematically in ***Figure (4)***.

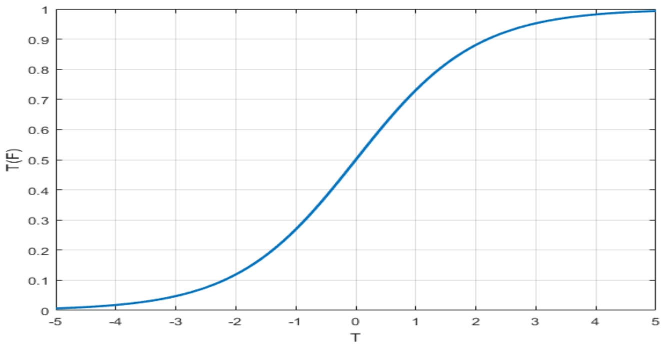


Fig. 4. Overview of the sigmoid transfer function

As can be seen in ***Figure (3)***, the sigmoid function causes the solutions to move between zero and one. Thus, in the proposed BMAMH model, initially the sigmoid function is used to change the continuous position of the solutions in the shared memory and convert it to a state between zero and one by means of ***Equation (12)***:

$$sg\left(GlobalMemory_i^d(t)\right) = \frac{1}{1+\mathrm{e}^{-GlobalMemory_i^k(t)}} \qquad (12)$$

Where, *GlobalMemory*$^d{}_i$ is the continuous value of the *i*th solution in the population of the shared memory in the *d*th dimension at iteration *t*. According to ***Figure (4)***, the sigmoid function's output is certainly a number between 0 and 1, the conversion of which to binary state requires thresholding. Such conversion can be performed using ***Equation (13)***, which is applied for the ultimate conversion of the solutions to binary state:

$$BMAMH\,{}_i^d(t+1) = \begin{cases} 0 & if\ rand < sg\left(GlobalMemory_i^d(t)\right) \\ 1 & if\ rand \geq sg\left(GlobalMemory_i^d(t)\right) \end{cases} \qquad (13)$$

Where, *GlobalMemory*$^d{}_i$ indicates the position of the *i*th solution in the population of shared memory at iteration *t* in the *d*th dimension. Also, *rand* represents a number between 0 and 1 with uniform distribution. On this basis, the existing solutions in the shared memory should move in a binary search space using the ***equations (12) & (13)***. Ultimately, this function has been incorporated in the proposed MAMH approach for the purpose of conversion to binary state. To provide a better understanding of the proposed approach, its pseudocode is presented in ***Figure (5)***.

```
Begin
01: Global Memory (Generate initial population of Solution $X_i$ (i=1... N)) $X_i \in$ random 0,1
02: Initialize parameters D, maxit, varmin, varmax, ….
03: Calculate the fitness of each solution by f ($X_i$);
04: X*=The best solution
05: Assign solutions to agents by Equation (3);
06: Calculate the mean fitness of each agent by Equation (4);
07: While (t < maximum number of iterations)
**Run each agent with new population**
07-1:    run agent one (GA   operators)
```

```
07-2:   run agent two (PSO  operators)
07-3:   run agent three (ABC  operators)
07-4:   run agent four (FA  operators)
07-5:   run agent five (FPA  operators)
07-6:   run agent six (BA  operators)
07-7:   run agent seven (GWO  operators)
07-8:   run agent eight (WOA  operators)
07-9:   run agent nine (CSA  operators)
07-10:   run agent ten (FFA  operators)
**  End all agent**
** Binary with Sigmoid function**
08:     For i=1 : N
09:         Convert Continuous solution to Binary solution use transfer function according to Equations (12) and (13)
10:     End for
**End binary stage**
11: Calculate the mean fitness of each agent by Equation (4);
12: Update Global memory by Equations (5:7);
13: Update solutions to agents by Equation (8:11);
14: Update X* if there is a better solution
15: t=t+1
16: End While
17: Print results.
End
```

Fig. 5*. Pseudocode the proposed BMAMH approach based on multi-agent system*

Figure (5) shows the proposed multi-agent system-based BMAMH approach for solving the binary problems with the help of *sigmoid* function. In this pseudocode, the lines 8 to 10 have been used as the conversion of the continuous shared memory to the binary shared memory.

5. Tests and Results

In order to evaluate the performance of the proposed algorithm, the tests were performed in two phases. In the first phase, the tests were performed on the benchmark functions using the proposed MAMH approach, for the purpose of which a total of 20 high-dimensional benchmark functions were used. These tests will be examined in details in ***Subsection (5-1)***. The second phase focused on testing the performance of the binary version of the proposed approach, BMAMH, in detection of spam emails. The ***Subsection (5-2)*** includes detailed description of the tests of the second phase along with the name and features of the spam email dataset as well as the primary settings and results of the proposed binary BMAMH approach on the spam dataset.

5-1 Optimization Functions

In this paper, a total of 32 high-dimensional benchmark functions were used to evaluate the proposed MAMH approach's efficiency and compare it with other metaheuristic algorithms. These standard benchmark functions are listed in ***Table (1)***, which includes the name, numerical range, and mathematical formula of these functions. In the present paper, the *CEC'2010* and *CEC'2017* benchmark functions were used to evaluate the proposed method. The *CEC'2010* functions were used in order that we can compare the proposed method with other works in this regard, while the *CEC'2017* functions were adopted for testing the proposed method due to their novelty and combinatorial nature. The functions (1 to 23) and (24 to 32) belong to the years 2010 and 2017, respectively.

Table 1: List of standard benchmark functions used.

No	Function Name	Function Formula	Search Range
F1	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$
F2	Sum-Square	$f_2(x) = \sum_{i=1}^{n} ix_i^2$	$[-10, 10]$
F3	Rotated hyper-ellipsoid	$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)$	$[-100, 100]$
F4	Rosenbrock	$f_4(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$
F5	Step	$f_5(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100, 100]$
F6	Quartic	$f_6(x) = \sum_{i=1}^{n} x_i^4$	$[-1.28, 1.28]$
F7	Noise	$f_7(x) = \sum_{i=1}^{n} x_i^4 + random[0,1)$	$[-1.28, 1.28]$
F8	Sum-Power	$f_8(x) = \sum_{i=1}^{n} \lvert x_i \rvert^2$	$[-1, 1]$
F9	Rastrigin	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$
F10	Ackley	$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]$
F11	Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 + \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]$
F12	Levy	$f_{12}(x) = \sum_{i=1}^{n} (x_i - 1)^2[sin^2(3\pi x_{i+1})] + sin^2(3\pi x_1)$ $+ \lvert x_n - 1 \rvert [1 + sin^2(3\pi x_n)]$	$[-10, 10]$
F13	Alpine	$f_{13}(x) = \sum_{i=1}^{n} \lvert x_i.\sin(x_i) + 0.1.x_i \rvert$	$[-10, 10]$
F14	Inverted Cosine Mixture	$f_{14}(x) = 0.1n - \left(0.1\sum_{i=1}^{n} cos(5\pi x_i) - \sum_{i=1}^{n} x_i^2 \right)$	[-1,1]

F15	Zakharov	$f_{15}(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 5ix_i\right)^2 + \left(\sum_{i=1}^{n} 5ix_i\right)^4$	$[-5, 10]$
F16	Pathological	$f_{16}(x) = \sum_{i=2}^{n} 0.5 + \frac{sin^2\left(\sqrt{x_{i-1}^2 + x_i^2}\right) - 0.5}{1 + 0.001(x_{i-1}^2 - 2x_{i-1}^2 + x_i^2)^{\,2}}$	[-100,100]
F17	Levy and Montalo	$f_{17}(x) = 0.1\left(sin^2(3\pi x_1)\right)$ $+ \sum_{i=1}^{n}(x_i - 1)^2\left(1 + sin2(3\pi x_{i+1})\right) + (x_n - 1)^2(1$ $+ sin^2(2\pi x_n))$	$[-5, 5]$
F18	Discus Function	$f_{18}(x) = 10^6 x_1^2 \sum_{i=2}^{D} x_i^2$	$[-100, 100]$
F19	Salomon	$f_{19}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	$[-100, 100]$
F20	Bent Cigar	$f_{20}(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	[-100,100]
F21	Sum of Different Power	$f_{21}(x) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	[-100,100]
F22	HGBat	$f_{22}(x) = \left\lvert(\sum_{i=1}^{D} x_i^2)^2 - (\sum_{i=1}^{D} x_i)^2\right\rvert^{1/2} + (0.5\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/D + 0.5$	[-100,100]
F23	High Conditioned Elliptic	$f_{23}(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}}\, x_i^2$	[-100,100]
F24	SR-F20	Shifted and Rotated Bent Cigar Function	[-100,100]
F25	SR-F21	Shifted and Rotated Sum of Different Power Function*	[-100,100]
F26	SR-F15	Shifted and Rotated Zakharov Function	[-100,100]
F27	SR- F4	Shifted and Rotated Rosenbrock's Function	[-100,100]
F28	SR- F9	Shifted and Rotated Rastrigin's Function	[-100,100]
F29	Hybrid1	g1 : Zakharov Function f3 g2 : Rosenbrock Function F04 g3: Rastrigin's Function F09	[-100,100]

F30	Hybrid2	g1 : High Conditioned Elliptic Function F23 g2 : Ackley's Function F10 g3: Rastrigin's Function F09 g4: HGBat Function F22 g4: Discus Function F18	[-100,100]

F31	Composition1	g1=Rosenbrock's Function F04;
		g2=High Conditioned Elliptic Function F23;
		g3=Rastrigin's Function F09;
			[-100,100]

F32	Composition2	g1=Ackley's Function F10
		g2=High Conditioned Elliptic Function F23
		g3=Girewank Function F11
		g4=Rastrigin's Function F09
			[-100,100]

In this section, the proposed MAMH approach is compared with the ***GA, PSO, ABC, FA, BA, FPA, GWO, WOA, CSA,*** and ***FFA*** algorithms. For this purpose, all of these algorithms were assumed with initial population of 100 and objective function calling order of 20000. Further, other parameters of each of these comparative algorithms are presented in ***Table (2)***.

Table 2: Parameters settings for each of the comparative algorithms.

Algorithm Name	Parameter
GA	Pc=0.5; pm=0.2;population size is 100
PSO	C_1=1.5;C_2=2;population size is 100
ABC	nOnlooker=2*npop; L=round(0.8*nVar*nPop);population size is 100
FA	$\alpha = 0.1; \beta = 2 * rand$;population size is 100
BAT	A =0.5; R = 0.9;population size is 100
FPA	p=0.8; population size is 100
GWO	population size is 100
WOA	population size is 100
CSA	AP=0.8;fl=1; population size is 100
FFA	w=1; Q = 0.6 ; α = 0.8; β= 0.9 ;population size is 100
MAMH	a=0.6;population size is 100

Given the settings of each of the comparative algorithms provided in ***Table (2)***, the proposed algorithm and other algorithms were executed on the 32 given benchmark functions at dimensions of 100, 200, 500, and 1000. However, since the CEC'2017 functions could be adjusted only at dimensions of 100, 50, and 30, the functions 24 to 32 were executed at their highest possible dimension, i.e. 100, but other functions could be tested at various dimensions. The value of 100 was assumed for the number of dimensions in the first test, the results of which are provided in ***Figures (6)-(9)*** and ***Table (3)***. As seen in ***Table (3)***, the statistical criteria, such as *the best, the worst, average,* and *standard deviation*, were used for exact analysis of the proposed algorithm and other comparative algorithms.

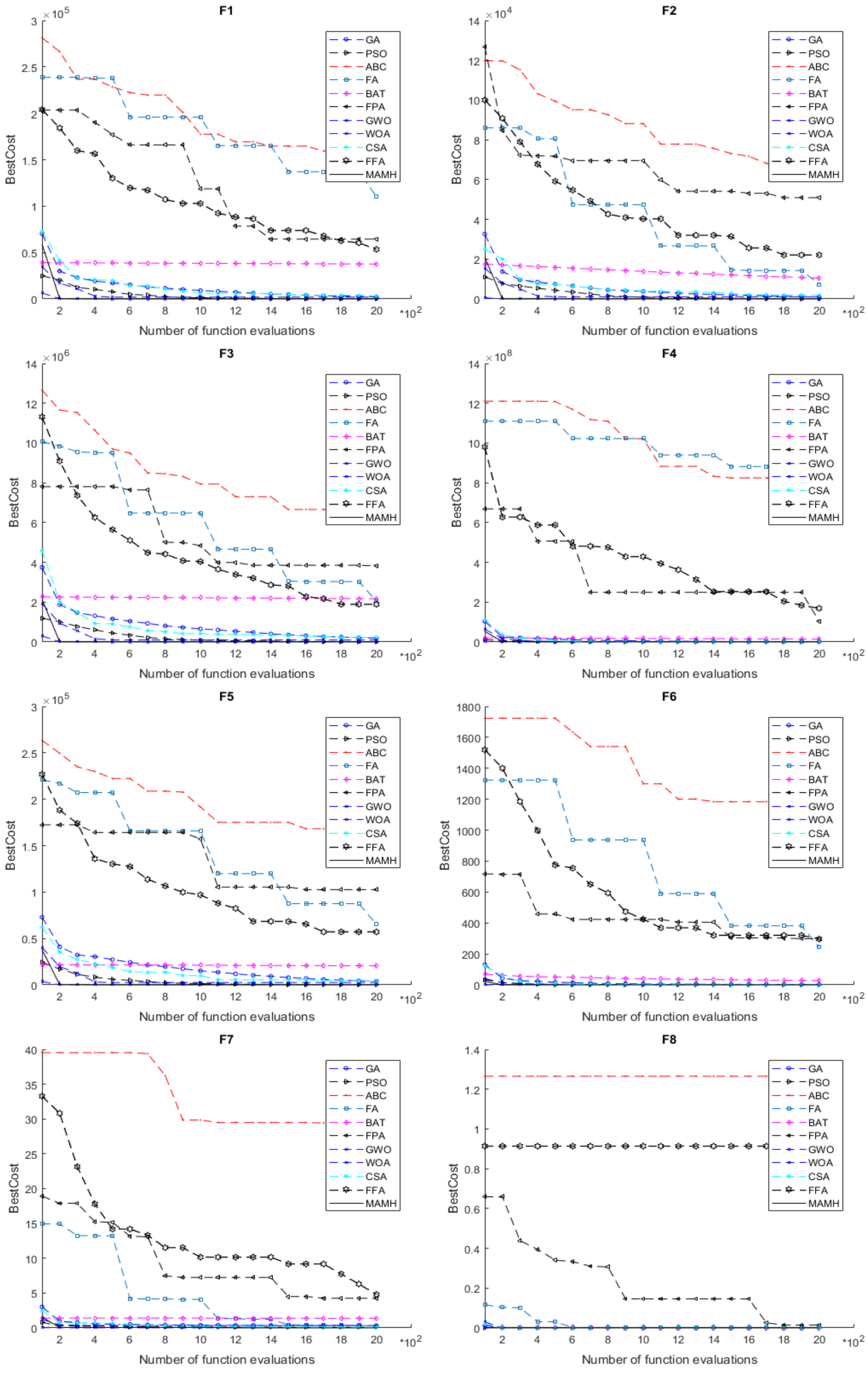
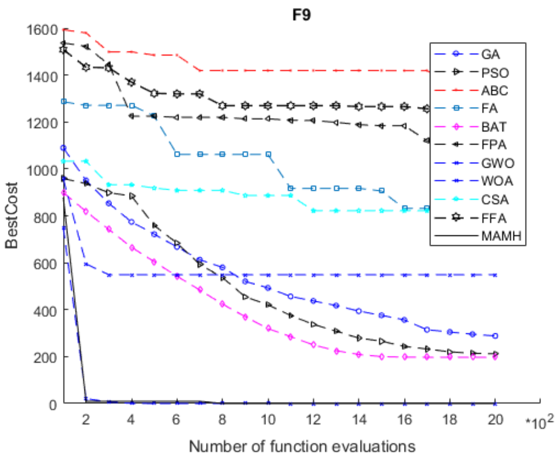
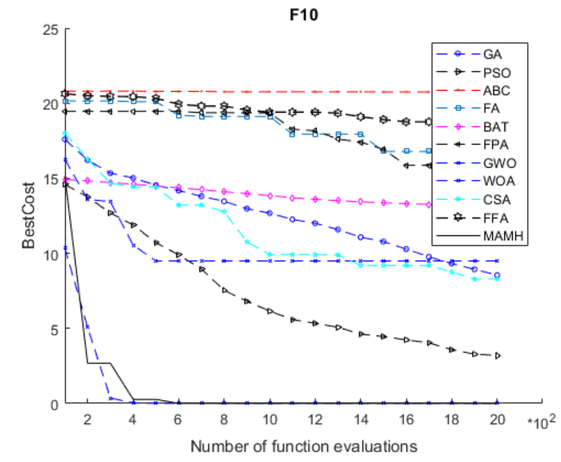
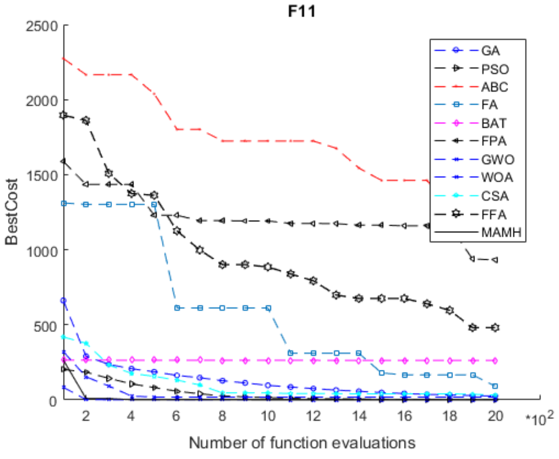
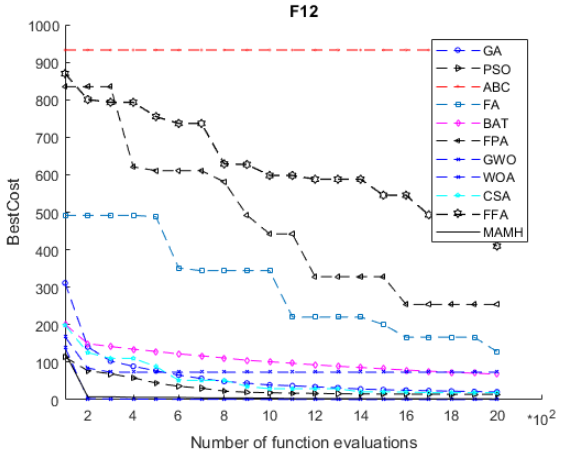

F1
× 10^5
BestCost
Number of function evaluations
*10^2
F2
× 10^4
F3
× 10^6
F4
× 10^8
F5
× 10^5
F6
F7
F8
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH
0
2
4
6
8
10
12
14
16
18
20

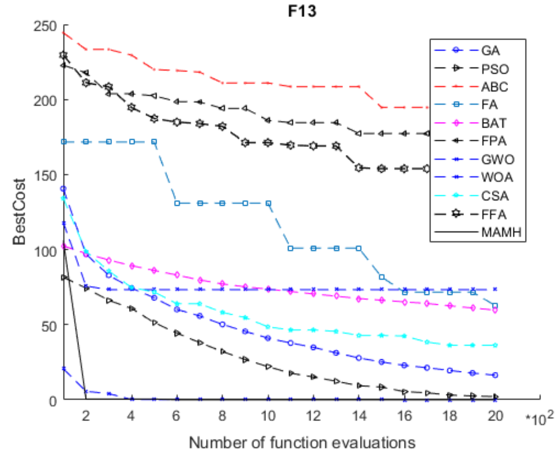
Fig. 6. Comparison of proposed algorithm with other algorithms in dimensions of 100 on functions 1-8

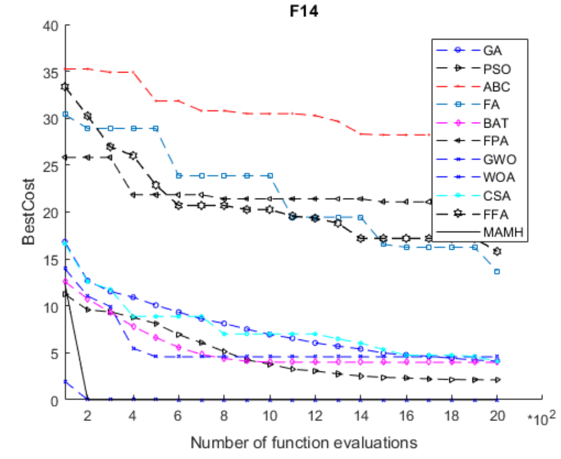

F9
BestCost
1600
1400
1200
1000
800
600
400
200
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

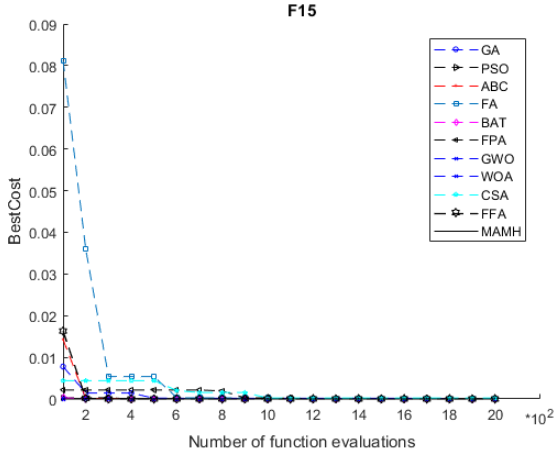

F10
BestCost
25
20
15
10
5
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F11
BestCost
2500
2000
1500
1000
500
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F12
BestCost
1000
900
800
700
600
500
400
300
200
100
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F13
BestCost
250
200
150
100
50
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F14
BestCost
40
35
30
25
20
15
10
5
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F15
BestCost
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

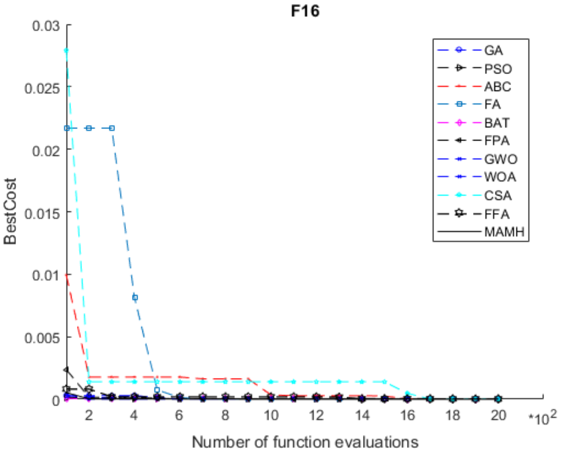
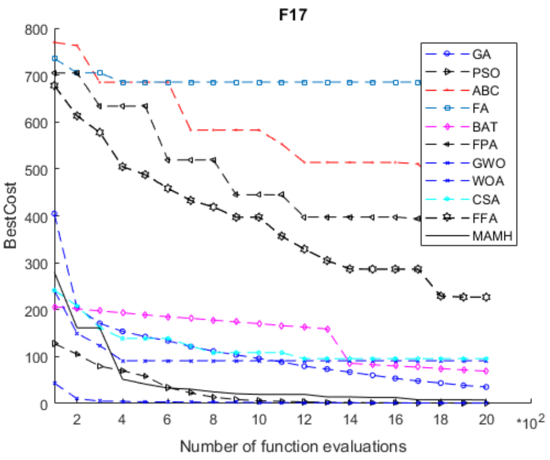
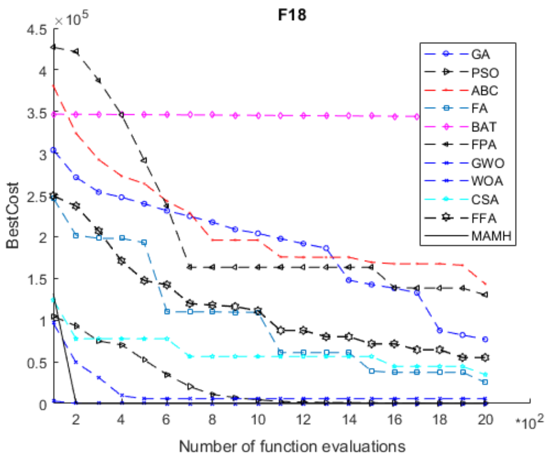
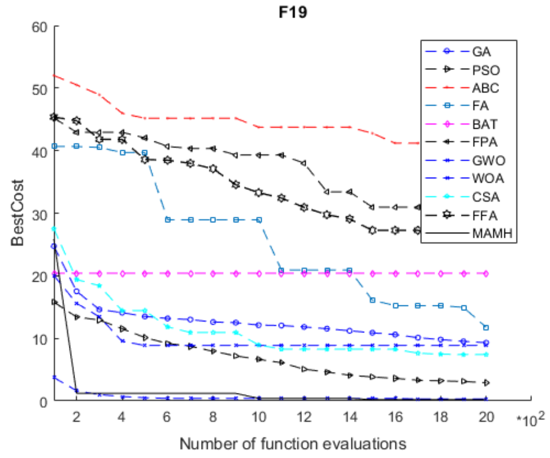
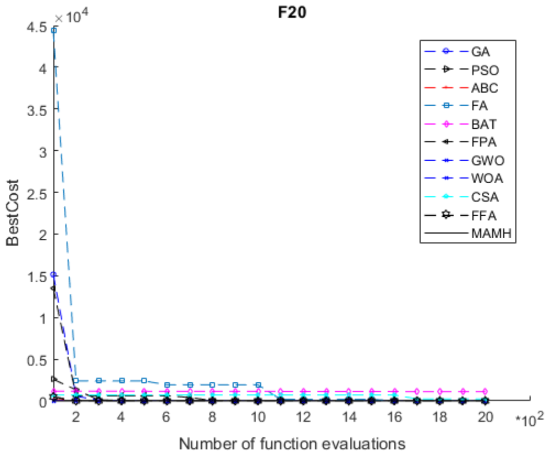

F16
BestCost
0.03
0.025
0.02
0.015
0.01
0.005
0
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

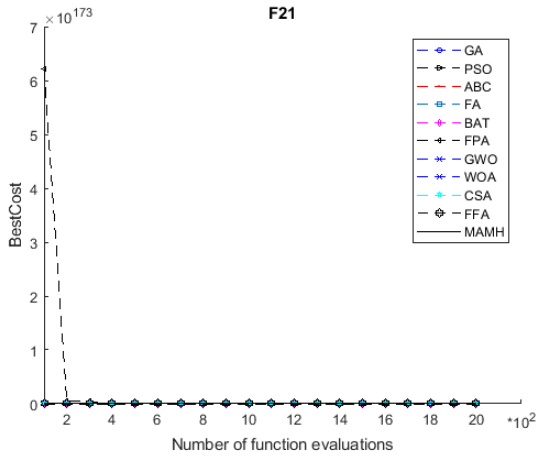
Fig. 7. Comparison of proposed algorithm with other algorithms in dimensions of 100 on functions 9-16

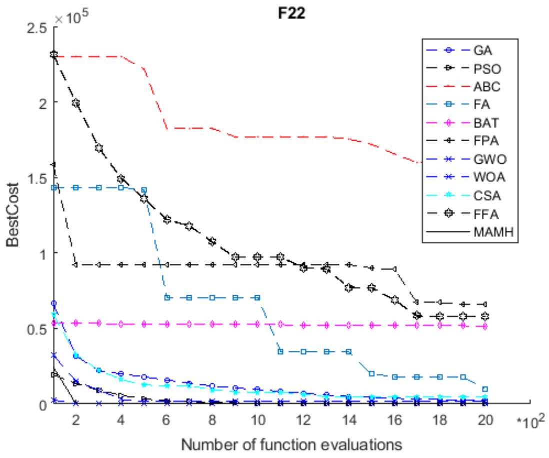

F17
BestCost
0
100
200
300
400
500
600
700
800
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

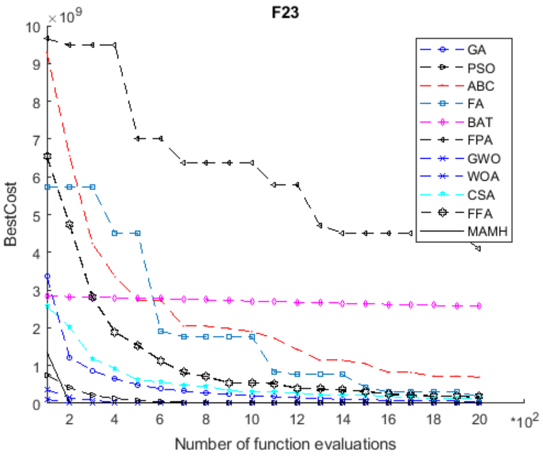

F18
× 10^5
BestCost
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F19
BestCost
0
10
20
30
40
50
60
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F20
× 10^4
BestCost
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F21
× 10^173
BestCost
0
1
2
3
4
5
6
7
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F22
× 10^5
BestCost
0
0.5
1
1.5
2
2.5
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH


F23
× 10^9
BestCost
0
1
2
3
4
5
6
7
8
9
10
2 4 6 8 10 12 14 16 18 20
*10^2
Number of function evaluations
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

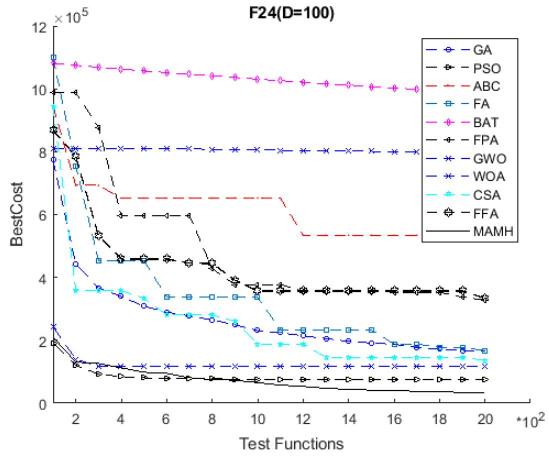

F24(D=100)
× 10^5
BestCost
0
2
4
6
8
10
12
2 4 6 8 10 12 14 16 18 20
*10^2
Test Functions
GA
PSO
ABC
FA
BAT
FPA
GWO
WOA
CSA
FFA
MAMH

Fig. 8. Comparison of proposed algorithm with other algorithms in dimensions of 100 on functions 17-24

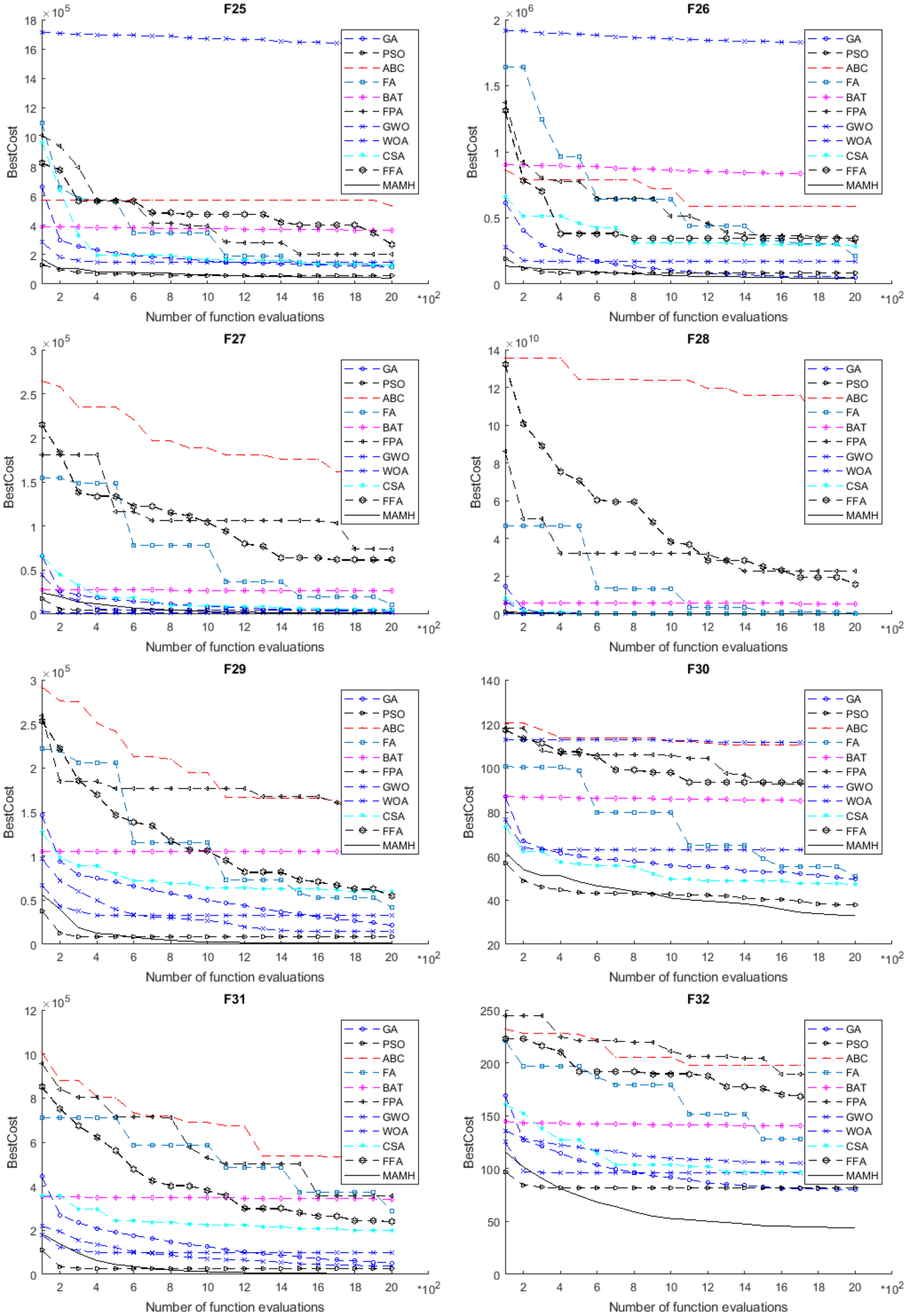


Fig. 9. Comparison of proposed algorithm with other algorithms in dimensions of 100 on functions 25-32

In the first test in ***Figures (6)-(9)***, our proposed algorithm and other comparative algorithms were executed on 100 dimensions, the results of which are provided based on the best solution. These results are typically used for demonstrating the convergence of the algorithms in accordance with the calling order of the objective function. As indicated by the results, the proposed method yielded effective and powerful results on the CEC'2010 and CEC'2017 functions; besides, it exhibited a good convergence on the functions 30 and 32 and some of the CEC'2017 functions so that it could maintain its convergence over the whole process. On this basis, it can be said that the proposed algorithm has accomplished improving the best solution in its population more than other algorithms; however, due to the use of multiple agents and other algorithms, the proposed algorithm is rather inclined toward the best algorithm in each execution.

Table 3: Comparison of proposed algorithm with other meta-heuirustic on functions 1-32 with dimensions 100

		GA	PSO	ABC	FA	BAT	FPA	GWO	WOA	CSA	FFA	MAMH
F1	Best	2.34E+03	1.91E+02	1.59E+05	1.10E+05	3.76E+04	6.44E+04	1.96E+03	5.71E-06	3.05E+03	5.34E+04	3.80E-07
	worst	2.65E+03	3.71E+02	2.37E+05	1.37E+05	3.78E+04	3.22E+05	4.27E+07	1.48E+06	3.91E+05	3.97E+05	4.97E-05
	mean	2.59E+03	2.11E+02	1.98E+05	1.22E+05	3.76E+04	2.33E+05	3.16E+07	1.74E+05	2.93E+05	3.29E+05	1.12E-05
	Std	6.02E+01	2.85E+01	1.57E+04	5.78E+03	3.13E+01	5.51E+04	4.20E+06	3.41E+05	1.01E+05	3.14E+04	1.92E-05
F2	Best	1.04E+03	6.67E+01	6.64E+04	7.38E+03	1.05E+04	5.09E+04	1.05E+03	9.10E-08	1.86E+03	2.21E+04	5.68E-07
	worst	1.13E+03	1.55E+02	1.00E+05	1.43E+04	1.12E+04	1.61E+05	2.26E+07	3.19E+06	2.24E+05	2.06E+05	3.22E-04
	mean	1.10E+03	7.27E+01	8.41E+04	9.06E+03	1.06E+04	1.15E+05	1.62E+07	1.58E+05	1.45E+05	1.68E+05	7.25E-05
	Std	1.81E+01	1.08E+01	8.27E+03	9.93E+02	1.16E+02	2.87E+04	2.61E+06	4.16E+05	5.55E+04	1.58E+04	7.34E-05
F3	Best	1.88E+05	8.23E+03	6.36E+06	2.06E+06	2.17E+06	3.82E+06	9.93E+04	2.58E-05	2.30E+05	1.89E+06	5.36E-06
	worst	2.05E+05	1.68E+04	1.16E+07	3.06E+06	2.18E+06	1.65E+07	2.44E+09	6.47E+07	2.08E+07	2.13E+07	1.10E-04
	mean	2.00E+05	9.20E+03	8.45E+06	2.22E+06	2.17E+06	1.12E+07	1.64E+09	5.43E+06	1.38E+07	1.66E+07	5.90E-05
	Std	3.85E+03	1.52E+03	9.77E+05	1.62E+05	1.35E+03	2.66E+06	2.71E+08	1.00E+07	6.23E+06	1.73E+06	4.30E-05
F4	Best	8.37E+05	4.07E+03	7.86E+08	8.15E+08	1.42E+07	1.04E+08	1.64E+06	9.80E+01	3.40E+05	1.69E+08	9.82E+01
	worst	9.35E+05	2.74E+04	1.79E+09	9.11E+08	1.46E+07	1.58E+09	4.51E+13	1.61E+12	2.26E+09	2.23E+09	9.83E+01
	mean	9.07E+05	5.65E+03	1.22E+09	8.71E+08	1.43E+07	9.89E+08	2.45E+13	3.46E+10	1.36E+09	1.62E+09	9.83E+01
	Std	2.36E+04	3.42E+03	1.91E+08	2.67E+07	7.99E+04	3.84E+08	7.23E+12	1.97E+11	5.87E+08	2.16E+08	5.70E-02
F5	Best	3.86E+03	1.26E+02	1.53E+05	6.53E+04	2.06E+04	1.03E+05	2.55E+03	4.90E-01	3.70E+03	5.71E+04	2.65E+00
	worst	4.23E+03	4.03E+02	2.40E+05	8.75E+04	2.06E+04	3.23E+05	4.26E+07	1.71E+06	3.99E+05	3.93E+05	3.43E+00
	mean	4.11E+03	1.50E+02	1.98E+05	6.96E+04	2.06E+04	2.46E+05	3.18E+07	2.47E+05	2.95E+05	3.33E+05	2.98E+00
	Std	8.66E+01	3.61E+01	1.67E+04	3.91E+03	1.77E+01	4.67E+04	4.74E+06	3.84E+05	9.94E+04	2.78E+04	4.78E-01
F6	Best	1.87E+00	3.41E-01	1.18E+03	2.43E+02	2.74E+01	2.98E+02	1.51E+00	8.32E-03	6.46E-01	2.96E+02	9.18E-03
	worst	2.25E+00	1.52E+00	2.89E+03	3.82E+02	2.92E+01	2.55E+03	1.10E+08	5.50E+06	3.78E+03	4.01E+03	7.84E-02
	mean	2.11E+00	9.62E-01	1.70E+03	2.76E+02	2.80E+01	1.46E+03	3.97E+07	9.13E+04	2.33E+03	2.72E+03	5.33E-02
	Std	9.24E-02	3.00E-01	3.24E+02	2.34E+01	4.31E-01	6.45E+02	1.32E+07	6.17E+05	9.55E+02	4.74E+02	1.56E-02
F7	Best	3.17E-01	9.38E-02	2.94E+01	2.46E-01	1.36E+00	4.25E+00	3.31E-01	5.33E-03	9.27E-02	4.80E+00	1.99E-03
	worst	3.77E-01	1.07E+00	5.62E+01	4.30E-01	1.48E+00	5.41E+01	1.95E+06	7.30E+04	7.37E+01	7.68E+01	8.30E-02
	mean	3.57E-01	5.76E-01	4.19E+01	2.56E-01	1.40E+00	3.17E+01	9.36E+05	1.59E+03	4.56E+01	5.38E+01	5.55E-02
	Std	1.60E-02	2.88E-01	5.96E+00	5.78E-02	2.43E-02	1.33E+01	3.23E+05	8.48E+03	1.93E+01	7.56E+00	1.92E-02

F8	Best	6.39E-07	8.97E-23	1.27E+00	1.30E-06	5.17E-08	1.37E-02	6.40E-09	8.07E-17	1.83E-05	9.14E-01	3.89E-31
	worst	1.25E-06	2.70E-17	7.96E+00	5.83E-06	1.93E-07	4.56E+00	3.78E+119	1.23E+77	7.42E+00	7.06E+00	2.44E-21
	mean	9.46E-07	9.22E-19	3.81E+00	2.11E-06	7.47E-08	1.38E+00	9.29E+117	1.53E+75	3.24E+00	3.56E+00	4.00E-22
	Std	2.73E-07	4.23E-18	1.31E+00	1.29E-06	2.51E-08	9.31E-01	5.45E+118	1.37E+76	1.84E+00	1.20E+00	8.31E-22
F9	Best	2.88E+02	2.12E+02	1.36E+03	8.19E+02	1.97E+02	1.12E+03	5.49E+02	5.07E-07	8.22E+02	1.20E+03	1.25E-12
	worst	2.99E+02	2.24E+02	1.66E+03	9.07E+02	1.97E+02	1.73E+03	1.23E+05	6.15E+03	2.04E+03	2.08E+03	8.66E-10
	mean	2.97E+02	2.14E+02	1.52E+03	8.70E+02	1.97E+02	1.49E+03	9.20E+04	8.59E+02	1.73E+03	1.85E+03	1.71E-10
	Std	2.14E+00	2.75E+00	5.49E+01	2.70E+01	6.96E-03	1.33E+02	1.42E+04	9.96E+02	2.70E+02	1.09E+02	2.47E-10
F10	Best	8.55E+00	3.20E+00	2.07E+01	1.61E+01	1.32E+01	1.59E+01	9.50E+00	3.83E-04	8.29E+00	1.86E+01	1.10E-03
	worst	8.81E+00	3.55E+00	2.10E+01	1.69E+01	1.32E+01	2.05E+01	2.19E+01	2.18E+01	2.14E+01	2.15E+01	7.51E-03
	mean	8.74E+00	3.31E+00	2.09E+01	1.62E+01	1.32E+01	1.91E+01	2.17E+01	1.01E+01	2.03E+01	2.12E+01	6.13E-03
	Std	7.52E-02	6.97E-02	5.30E-02	2.97E-01	5.20E-04	9.43E-01	7.58E-02	8.70E+00	2.53E+00	1.09E-01	1.76E-03
F11	Best	2.55E+01	1.47E+00	1.28E+03	9.18E+01	2.63E+02	9.32E+02	1.84E+01	2.40E-06	3.19E+01	4.81E+02	1.50E-07
	worst	2.78E+01	2.57E+00	2.16E+03	1.67E+02	2.63E+02	3.04E+03	4.08E+05	1.08E+04	3.64E+03	3.71E+03	1.36E-05
	mean	2.73E+01	1.68E+00	1.76E+03	1.10E+02	2.63E+02	2.21E+03	2.92E+05	1.31E+03	2.53E+03	3.01E+03	3.04E-06
	Std	4.44E-01	2.35E-01	1.69E+02	1.03E+01	5.01E-02	4.85E+02	4.19E+04	2.29E+03	1.05E+03	2.62E+02	3.85E-06
F12	Best	2.08E+01	1.42E+01	9.32E+02	1.28E+02	6.81E+01	2.54E+02	7.35E+01	3.63E-01	1.87E+01	4.10E+02	2.00E+00
	worst	2.17E+01	1.52E+01	1.57E+03	1.70E+02	7.19E+01	1.39E+03	2.04E+05	1.44E+04	1.58E+03	1.66E+03	2.48E+00
	mean	2.15E+01	1.45E+01	1.22E+03	1.43E+02	6.90E+01	7.94E+02	1.27E+05	8.90E+02	1.05E+03	1.30E+03	2.13E+00
	Std	1.47E-01	2.26E-01	1.35E+02	1.43E+01	8.78E-01	2.22E+02	2.61E+04	2.02E+03	4.25E+02	1.56E+02	2.06E-01
F13	Best	1.64E+01	2.10E+00	1.95E+02	6.27E+01	5.97E+01	1.74E+02	7.33E+01	2.33E-05	3.63E+01	1.54E+02	8.92E-09
	worst	1.78E+01	5.26E+00	2.64E+02	7.67E+01	6.18E+01	2.63E+02	3.75E+03	8.21E+02	3.61E+02	3.82E+02	1.99E-08
	mean	1.74E+01	2.52E+00	2.29E+02	6.83E+01	6.02E+01	2.19E+02	2.90E+03	1.12E+02	2.71E+02	3.02E+02	8.48E-09
	Std	2.96E-01	4.47E-01	1.26E+01	3.15E+00	4.53E-01	2.09E+01	3.28E+02	1.39E+02	6.72E+01	2.38E+01	5.91E-09
F14	Best	4.11E+00	2.12E+00	2.80E+01	1.37E+01	3.99E+00	2.10E+01	4.58E+00	1.92E-09	4.10E+00	1.58E+01	6.75E-14
	worst	4.23E+00	2.32E+00	3.77E+01	1.63E+01	4.00E+00	4.51E+01	4.65E+03	2.48E+02	4.96E+01	4.96E+01	2.71E-11
	mean	4.19E+00	2.13E+00	3.22E+01	1.50E+01	3.99E+00	3.45E+01	3.29E+03	2.78E+01	3.81E+01	4.33E+01	7.78E-12
	Std	2.73E-02	2.73E-02	2.13E+00	5.11E-01	3.43E-04	5.74E+00	4.26E+02	5.37E+01	1.19E+01	3.05E+00	1.02E-11
F15	Best	4.61E-05	4.19E-55	1.32E-28	9.03E-07	6.26E-10	4.44E-07	7.68E-39	1.44E-16	2.33E-04	1.47E-32	2.81E-21
	worst	4.61E-05	7.45E-38	1.34E-13	9.84E-05	3.67E-07	3.37E+00	2.11E+08	7.30E+04	3.29E+04	3.78E+04	1.22E-19
	mean	4.61E-05	8.76E-40	1.34E-15	4.31E-05	7.23E-08	6.70E-02	1.52E+07	1.75E+03	2.41E+03	3.05E+03	2.00E-20
	Std	6.81E-21	7.55E-39	1.34E-14	2.64E-05	7.55E-08	3.84E-01	3.88E+07	8.90E+03	5.68E+03	6.09E+03	1.86E-20
F16	Best	4.99E-06	5.07E-13	3.21E-05	3.69E-06	7.06E-09	1.21E-06	2.22E-06	1.79E-06	1.93E-05	2.14E-05	7.51E-10
	worst	4.99E-06	2.79E+01	3.38E-01	4.73E-03	1.89E-06	1.11E-02	2.54E+10	8.67E+05	1.00E+06	1.42E+06	1.63E-07
	mean	4.99E-06	1.04E+00	3.96E-02	1.88E-03	1.68E-07	9.54E-04	1.11E+09	1.26E+04	8.82E+04	1.23E+05	8.89E-08
	Std	1.19E-20	3.29E+00	5.09E-02	1.41E-03	2.81E-07	1.68E-03	3.28E+09	9.71E+04	1.90E+05	2.43E+05	7.26E-08
F17	Best	3.53E+01	8.65E-01	4.32E+02	6.85E+02	6.90E+01	3.92E+02	9.13E+01	8.46E-01	9.56E+01	2.27E+02	7.43E+00
	worst	3.69E+01	1.16E+00	7.97E+02	7.76E+02	7.95E+01	1.10E+03	2.12E+05	1.23E+04	2.15E+03	2.09E+03	1.28E+01
	mean	3.64E+01	8.99E-01	5.96E+02	7.54E+02	6.98E+01	7.81E+02	1.28E+05	1.40E+03	1.23E+03	1.37E+03	1.07E+01
	Std	4.32E-01	4.61E-02	6.64E+01	2.02E+01	1.31E+00	1.46E+02	3.56E+04	2.87E+03	4.97E+02	3.41E+02	1.91E+00

F18	Best	7.69E+04	1.94E+02	1.43E+05	2.55E+04	3.44E+05	1.30E+05	5.72E+03	7.07E-06	3.47E+04	5.51E+04	7.93E-08
	worst	7.98E+04	2.98E+02	2.52E+05	4.21E+04	3.44E+05	2.14E+09	1.80E+12	4.53E+10	9.83E+09	9.81E+09	4.28E-03
	mean	7.92E+04	2.06E+02	2.03E+05	3.03E+04	3.44E+05	2.27E+07	2.82E+11	1.25E+09	2.87E+09	3.03E+09	6.30E-04
	Std	5.18E+02	1.21E+01	1.96E+04	4.20E+03	4.73E+01	2.14E+08	3.82E+11	5.70E+09	2.92E+09	2.77E+09	1.31E-03
F19	Best	9.30E+00	2.90E+00	4.12E+01	1.18E+01	2.04E+01	3.10E+01	8.86E+00	3.00E-01	7.41E+00	2.60E+01	2.00E-01
	worst	9.72E+00	5.55E+00	5.09E+01	1.50E+01	2.04E+01	5.83E+01	6.52E+02	1.50E+02	6.47E+01	6.49E+01	2.00E-01
	mean	9.62E+00	4.17E+00	4.64E+01	1.24E+01	2.04E+01	4.96E+01	5.70E+02	2.77E+01	5.16E+01	5.92E+01	2.00E-01
	Std	9.28E-02	7.47E-01	1.78E+00	5.93E-01	2.71E-08	6.56E+00	3.94E+01	3.29E+01	1.53E+01	2.71E+00	5.88E-08
F20	Best	1.06E+01	1.03E-50	1.46E-25	1.90E+01	1.10E+03	1.15E-01	9.64E-36	1.44E-09	2.34E+02	8.85E-32	2.13E-18
	worst	1.06E+01	1.19E-32	1.14E-07	2.73E+03	1.11E+03	3.11E+06	2.44E+12	8.63E+10	9.89E+09	9.91E+09	1.18E-13
	mean	1.06E+01	1.41E-34	1.14E-09	8.86E+02	1.10E+03	6.83E+04	3.99E+11	1.67E+09	3.01E+09	2.87E+09	3.80E-14
	Std	1.43E-14	1.20E-33	1.14E-08	7.48E+02	1.44E+00	3.88E+05	5.20E+11	9.89E+09	2.93E+09	2.92E+09	5.18E-14
F21	Best	4.59E+118	1.75E+62	1.00E+148	1.97E+115	6.44E+141	1.43E+155	8.39E+93	2.94E-10	3.92E+89	8.21E+130	1.19E+30
	worst	1.13E+119	3.79E+100	4.21E+175	1.48E+125	1.58E+142	1.26E+178	1.86e+308	7.18E+229	3.97E+201	8.47E+201	7.64E+70
	mean	8.73E+118	6.42E+98	7.50E+173	5.31E+123	7.34E+141	4.19E+176	1.86e+308	9.76E+227	8.04E+199	2.30E+200	1.48E+70
	Std	1.52E+118	4.90E+99	1.86e+308	2.12E+124	1.26E+141	1.86e+308	1.86e+308	1.86e+308	1.86e+308	1.86e+308	2.98E+70
F22	Best	2.16E+03	1.69E+02	1.57E+05	1.00E+04	5.17E+04	6.61E+04	1.81E+03	5.46E-01	4.61E+03	5.77E+04	5.00E-01
	worst	2.44E+03	1.78E+02	2.42E+05	1.81E+04	5.18E+04	3.33E+05	4.47E+07	6.10E+06	4.10E+05	4.30E+05	6.25E-01
	mean	2.37E+03	1.69E+02	1.97E+05	1.18E+04	5.17E+04	2.16E+05	3.23E+07	3.69E+05	2.90E+05	3.38E+05	5.26E-01
	Std	6.45E+01	1.16E+00	1.87E+04	1.04E+03	2.97E+01	6.61E+04	4.36E+06	8.94E+05	1.02E+05	3.17E+04	3.81E-02
F23	Best	4.86E+07	7.93E+04	6.95E+08	1.76E+08	2.57E+09	4.11E+09	1.22E+07	1.96E-02	1.38E+08	1.76E+08	1.41E-04
	worst	5.36E+07	8.77E+04	3.30E+09	3.06E+08	2.59E+09	1.48E+10	5.72E+12	1.01E+11	4.41E+10	3.79E+10	1.81E+00
	mean	5.26E+07	8.07E+04	1.34E+09	2.26E+08	2.57E+09	8.74E+09	2.33E+12	6.45E+09	2.43E+10	2.56E+10	6.51E-01
	Std	1.07E+06	1.67E+03	4.31E+08	2.65E+07	3.96E+06	2.47E+09	8.47E+11	1.65E+10	9.04E+09	6.21E+09	6.67E-01
F24	Best	1.42E+05	7.77E+04	6.03E+05	1.70E+05	6.08E+05	3.19E+05	1.56E+05	6.38E+05	1.65E+05	3.85E+05	3.63E+04
	worst	1.47E+05	7.77E+04	2.24E+06	2.45E+05	6.15E+05	1.38E+06	8.88E+09	2.87E+09	9.07E+07	9.65E+07	1.03E+05
	mean	1.45E+05	7.77E+04	1.03E+06	2.09E+05	6.09E+05	6.49E+05	1.57E+09	3.34E+08	1.99E+07	1.99E+07	4.67E+04
	Std	1.17E+03	2.65E-11	3.01E+05	2.58E+04	1.01E+03	1.77E+05	1.49E+09	7.16E+08	2.21E+07	2.14E+07	1.37E+04
F25	Best	1.19E+05	5.95E+04	5.32E+05	1.16E+05	3.66E+05	2.04E+05	1.49E+05	1.63E+06	1.30E+05	2.67E+05	4.11E+04
	worst	1.22E+05	5.95E+04	1.49E+06	1.58E+05	3.67E+05	1.18E+06	5.98E+09	9.36E+09	1.01E+08	1.36E+08	1.39E+05
	mean	1.21E+05	5.95E+04	9.45E+05	1.33E+05	3.66E+05	6.68E+05	1.53E+09	6.02E+08	1.48E+07	1.88E+07	5.29E+04
	Std	6.48E+02	1.95E-11	2.11E+05	1.46E+04	3.11E+02	1.91E+05	1.24E+09	1.34E+09	1.89E+07	1.98E+07	1.71E+04
F26	Best	5.23E+04	8.10E+04	5.89E+05	2.10E+05	8.25E+05	3.19E+05	1.69E+05	1.81E+06	2.86E+05	3.48E+05	4.18E+04
	worst	5.36E+04	8.10E+04	2.75E+06	3.55E+05	8.29E+05	1.59E+06	1.04E+10	7.78E+09	1.10E+08	2.20E+08	2.08E+05
	mean	5.33E+04	8.10E+04	1.03E+06	2.94E+05	8.26E+05	8.31E+05	1.88E+09	8.32E+08	2.18E+07	2.54E+07	6.09E+04
	Std	2.50E+02	3.11E-11	3.12E+05	3.58E+04	7.12E+02	2.66E+05	1.96E+09	1.62E+09	2.19E+07	3.40E+07	2.61E+04
F27	Best	2.65E+03	4.37E+03	1.60E+05	1.09E+04	2.66E+04	7.40E+04	4.81E+03	1.24E+03	5.00E+03	6.15E+04	9.62E+02
	worst	2.82E+03	4.37E+03	2.42E+05	1.99E+04	2.66E+04	3.25E+05	4.33E+07	2.08E+06	4.00E+05	4.01E+05	1.12E+03
	mean	2.76E+03	4.37E+03	2.01E+05	1.35E+04	2.66E+04	2.32E+05	3.17E+07	2.30E+05	2.62E+05	3.32E+05	9.85E+02
	Std	4.46E+01	9.09E-13	1.85E+04	1.23E+03	6.72E+00	5.95E+04	4.20E+06	3.97E+05	1.31E+05	2.99E+04	2.59E+01

F28	Best	1.79E+07	6.85E+07	1.02E+11	3.34E+08	5.56E+09	2.26E+10	2.08E+08	8.40E+04	3.35E+07	1.59E+10	6.56E+05
	worst	2.02E+07	6.85E+07	2.15E+11	1.15E+09	5.61E+09	2.19E+11	5.73E+15	7.68E+14	2.82E+11	2.85E+11	5.08E+06
	mean	1.94E+07	6.85E+07	1.53E+11	4.78E+08	5.57E+09	1.16E+11	3.14E+15	1.04E+13	1.69E+11	2.02E+11	1.60E+06
	Std	5.87E+05	2.60E-08	2.28E+10	9.49E+07	7.40E+06	4.78E+10	9.18E+14	8.58E+13	6.83E+10	2.84E+10	9.51E+05
F29	Best	2.14E+04	8.28E+03	1.43E+05	4.17E+04	1.05E+05	1.51E+05	3.30E+04	1.43E+04	5.96E+04	5.46E+04	1.42E+03
	worst	2.27E+04	8.28E+03	2.53E+05	5.27E+04	1.05E+05	4.08E+05	4.71E+07	5.24E+06	4.97E+05	5.36E+05	1.54E+03
	mean	2.24E+04	8.28E+03	1.87E+05	4.45E+04	1.05E+05	2.80E+05	3.47E+07	4.67E+05	3.67E+05	4.18E+05	1.46E+03
	Std	2.82E+02	2.50E-12	2.16E+04	1.91E+03	5.87E-07	6.28E+04	5.07E+06	9.09E+05	1.12E+05	4.51E+04	3.08E+01
F30	Best	4.95E+01	3.80E+01	1.10E+02	5.08E+01	8.49E+01	9.28E+01	6.28E+01	1.12E+02	4.72E+01	9.36E+01	3.31E+01
	worst	5.00E+01	3.85E+01	1.38E+02	5.85E+01	8.51E+01	1.30E+02	2.66E+03	5.24E+02	1.49E+02	1.50E+02	4.54E+01
	mean	4.99E+01	3.80E+01	1.23E+02	5.47E+01	8.49E+01	1.16E+02	1.82E+03	1.82E+02	1.25E+02	1.38E+02	3.57E+01
	Std	7.20E-02	1.17E-01	5.42E+00	1.87E+00	3.59E-02	8.60E+00	3.38E+02	9.83E+01	2.80E+01	6.48E+00	2.52E+00
F31	Best	5.15E+04	2.76E+04	5.32E+05	2.89E+05	3.42E+05	3.56E+05	9.96E+04	3.76E+04	1.98E+05	2.38E+05	3.20E+03
	worst	5.42E+04	2.76E+04	9.06E+05	3.73E+05	3.43E+05	1.38E+06	2.11E+08	1.50E+07	2.22E+06	2.15E+06	4.09E+03
	mean	5.35E+04	2.76E+04	7.25E+05	3.25E+05	3.42E+05	9.30E+05	1.30E+08	1.75E+06	1.40E+06	1.58E+06	3.44E+03
	Std	5.98E+02	1.05E-11	7.75E+04	1.99E+04	1.75E+02	1.95E+05	2.21E+07	3.27E+06	4.77E+05	2.40E+05	2.39E+02
F32	Best	8.02E+01	8.17E+01	1.96E+02	1.20E+02	1.39E+02	1.86E+02	9.64E+01	1.04E+02	9.54E+01	1.60E+02	4.38E+01
	worst	8.08E+01	8.17E+01	2.74E+02	1.40E+02	1.40E+02	2.85E+02	4.49E+03	1.21E+03	5.31E+02	5.01E+02	4.78E+01
	mean	8.06E+01	8.17E+01	2.38E+02	1.31E+02	1.39E+02	2.34E+02	3.06E+03	3.20E+02	3.16E+02	3.40E+02	4.44E+01
	Std	1.37E-01	3.69E-14	1.53E+01	5.06E+00	1.07E-01	1.89E+01	5.07E+02	2.51E+02	9.24E+01	4.95E+01	8.43E-01

With regard to the first test performed on all functions with 100 dimensions, the results of which are provided in ***Table (3)*** and ***Figures (6)-(9)***, the proposed algorithm was recognized on the whole as a powerful algorithm. However, based on the results presented in these figures, it can be said that the proposed algorithm has exhibited more convergence in the 8 functions adopted from the CEC'2017 functions and, meanwhile, has shown a very powerful performance in the CEC'2010 functions. Moreover, the statistical criteria given in this table represent how the proposed algorithm changed the solutions existing in the population and converged them toward the intended goal. In this test, in case of 100 dimensions, the proposed approach had not only the best solution but also better standard deviation and mean values in most of the executions; hence, it can be claimed that the proposed algorithm has yielded higher degrees of improvement in all of the existing solutions in the population in comparison with other algorithms. However, due to the use of multi-agent systems and application of other algorithms, the proposed algorithm demonstrated greater tendency toward the best algorithm in each execution; as a result, the proposed algorithm would absolutely have a performance similar to that of one of the powerful algorithms (similar to the best agent). As can be inferred from the results, the proposed algorithm has acted sometimes similar to WOA and sometimes similar to PSO. The investigations indicated that the proposed algorithm has accomplished maintaining the relationships between the agents and displacing the solutions among them. In addition, regarding the fact that the functions 1 to 23 could have more than 100 dimensions, the proposed algorithm as well as the other ones was tested on these functions with 200, 500, and 1000 dimension. The obtained results are provided in ***Tables (4)*** to ***(6)***.

Table 4: Comparison of proposed algorithm with other meta-heuirustic on functions 1-23 with dimensions 200

		GA	PSO	ABC	FA	BAT	FPA	GWO	WOA	CSA	FFA	MAMH
F1	Best	3.06E+04	2.43E+03	4.70E+05	5.30E+04	7.79E+04	1.95E+05	3.20E+04	2.03E-05	1.04E+04	2.23E+05	7.25E+00
	worst	3.15E+04	3.14E+03	6.72E+05	7.92E+04	7.80E+04	6.54E+05	7.94E+07	2.83E+06	7.55E+05	7.57E+05	3.11E+04
	mean	3.12E+04	2.62E+03	5.70E+05	6.27E+04	7.79E+04	4.98E+05	6.57E+07	2.44E+05	5.78E+05	6.60E+05	2.22E+03
	Std	2.14E+02	1.43E+02	3.24E+04	3.64E+03	3.17E+01	1.23E+05	6.52E+06	4.55E+05	1.99E+05	4.60E+04	7.96E+03
F2	Best	1.55E+04	1.89E+03	4.44E+05	1.05E+05	6.02E+04	1.58E+05	2.15E+04	4.81E-07	8.28E+03	1.79E+05	4.85E+01
	worst	1.65E+04	2.42E+03	6.41E+05	1.34E+05	6.18E+04	6.69E+05	9.58E+07	3.80E+06	7.99E+05	8.46E+05	1.75E+05
	mean	1.63E+04	1.96E+03	5.08E+05	1.19E+05	6.06E+04	4.83E+05	6.56E+07	2.08E+05	5.54E+05	6.68E+05	3.37E+04
	Std	1.75E+02	6.90E+01	3.27E+04	6.80E+03	3.33E+02	1.29E+05	7.92E+06	4.84E+05	2.18E+05	4.99E+04	5.40E+04
F3	Best	1.58E+06	2.71E+05	4.29E+07	2.97E+06	7.36E+06	2.06E+07	1.50E+06	1.13E-04	9.16E+05	1.75E+07	8.74E+02
	worst	1.66E+06	3.19E+05	5.80E+07	4.95E+06	7.38E+06	6.71E+07	8.48E+09	2.72E+08	7.77E+07	7.66E+07	8.20E+06
	mean	1.64E+06	2.84E+05	4.99E+07	3.23E+06	7.37E+06	4.68E+07	6.33E+09	2.06E+07	5.58E+07	6.69E+07	2.09E+06
	Std	1.55E+04	1.06E+04	2.76E+06	2.38E+05	4.03E+03	1.26E+07	7.22E+08	5.41E+07	2.18E+07	4.81E+06	2.94E+06
F4	Best	6.81E+06	4.27E+05	2.50E+09	3.93E+07	2.97E+07	3.81E+08	3.34E+07	1.97E+02	7.71E+05	8.85E+08	2.15E+02
	worst	7.19E+06	7.40E+05	3.89E+09	1.04E+08	3.04E+07	3.04E+09	8.43E+13	8.84E+11	3.99E+09	3.72E+09	1.45E+07
	mean	7.09E+06	4.79E+05	3.20E+09	4.98E+07	2.99E+07	1.96E+09	5.59E+13	3.12E+10	2.59E+09	3.20E+09	1.03E+06
	Std	7.52E+04	4.83E+04	2.85E+08	9.23E+06	1.26E+05	7.73E+08	1.29E+13	1.39E+11	1.17E+09	2.59E+08	3.70E+06
F5	Best	2.60E+04	2.38E+03	4.98E+05	4.05E+04	9.61E+04	1.93E+05	2.65E+04	8.74E-01	7.02E+03	2.00E+05	2.62E+01
	worst	2.73E+04	2.88E+03	6.53E+05	6.68E+04	9.63E+04	7.12E+05	7.87E+07	9.66E+05	7.77E+05	7.42E+05	2.90E+04
	mean	2.71E+04	2.53E+03	5.69E+05	4.46E+04	9.62E+04	4.91E+05	6.50E+07	1.33E+05	5.90E+05	6.66E+05	5.15E+03
	Std	2.26E+02	1.04E+02	3.04E+04	3.58E+03	3.72E+01	1.32E+05	5.63E+06	2.28E+05	1.84E+05	3.62E+04	1.03E+04
F6	Best	2.98E+01	3.63E+00	8.48E+03	4.99E+01	2.69E+01	5.81E+02	5.71E+01	4.22E-03	2.91E+00	2.65E+03	1.86E-02
	worst	3.14E+01	5.79E+00	1.19E+04	1.52E+02	3.03E+01	1.03E+04	2.83E+08	1.33E+06	1.37E+04	1.39E+04	5.74E+00
	mean	3.10E+01	4.34E+00	1.04E+04	6.60E+01	2.75E+01	6.88E+03	1.78E+08	5.03E+04	9.00E+03	1.08E+04	1.25E+00
	Std	3.10E-01	3.85E-01	7.90E+02	1.24E+01	5.14E-01	2.52E+03	3.90E+07	1.89E+05	3.79E+03	1.13E+03	1.59E+00
F7	Best	1.01E+00	3.93E-01	8.30E+01	5.74E-01	1.79E+00	1.44E+01	2.13E+00	1.87E-02	1.33E-01	3.29E+01	7.24E-03
	worst	1.16E+00	1.40E+00	1.28E+02	1.35E+00	1.88E+00	1.10E+02	4.22E+06	4.39E+04	1.42E+02	1.32E+02	8.36E-01
	mean	1.13E+00	8.99E-01	1.06E+02	7.73E-01	1.83E+00	6.65E+01	1.96E+06	2.06E+03	8.75E+01	1.08E+02	4.27E-01
	Std	2.45E-02	2.81E-01	9.47E+00	1.53E-01	1.78E-02	2.65E+01	4.93E+05	7.47E+03	4.21E+01	1.08E+01	1.96E-01
F8	Best	6.80E-08	5.44E-21	1.90E+00	1.51E-01	8.29E-08	3.23E-02	1.74E-05	3.73E-16	1.43E-05	1.95E+00	3.83E-13
	worst	6.80E-08	3.19E-15	1.04E+01	4.58E-01	2.09E-07	4.41E+00	3.31E+278	5.17E+249	7.41E+00	9.23E+00	2.19E-03
	mean	6.80E-08	6.08E-17	4.29E+00	3.32E-01	1.16E-07	1.70E+00	3.31E+276	6.46E+247	3.58E+00	4.60E+00	3.69E-04
	Std	2.43E-12	3.39E-16	1.40E+00	8.23E-02	2.84E-08	1.24E+00	6.46E+280	6.46E+280	2.02E+00	1.28E+00	6.91E-04
F9	Best	9.73E+02	7.26E+02	3.12E+03	1.92E+03	4.37E+02	2.42E+03	1.76E+03	1.38E-04	1.80E+03	2.69E+03	8.29E+02
	worst	9.96E+02	7.87E+02	3.58E+03	2.08E+03	5.05E+02	3.45E+03	2.60E+05	1.70E+04	4.08E+03	3.96E+03	1.51E+03
	mean	9.91E+02	7.43E+02	3.34E+03	2.02E+03	4.51E+02	3.10E+03	1.91E+05	1.88E+03	3.42E+03	3.69E+03	1.09E+03
	Std	4.94E+00	1.40E+01	7.82E+01	4.46E+01	1.37E+01	2.45E+02	2.51E+04	2.70E+03	5.75E+02	1.28E+02	3.76E+02
F10	Best	1.16E+01	6.54E+00	2.09E+01	1.27E+01	1.44E+01	1.47E+01	1.23E+01	2.26E-04	8.38E+00	1.98E+01	2.59E-02
	worst	1.17E+01	6.87E+00	2.10E+01	1.46E+01	1.45E+01	2.06E+01	2.18E+01	2.17E+01	2.15E+01	2.14E+01	1.90E+01

	mean	1.17E+01	6.63E+00	2.10E+01	1.32E+01	1.44E+01	1.85E+01	2.17E+01	1.25E+01	1.99E+01	2.12E+01	1.02E+01
	Std	3.13E-02	6.28E-02	2.62E-02	2.61E-01	2.02E-02	1.21E+00	5.01E-02	8.43E+00	3.10E+00	7.20E-02	7.51E+00
F11	Best	2.12E+02	2.77E+01	4.36E+03	5.43E+02	4.65E+02	1.11E+03	2.37E+02	6.52E-07	6.08E+01	2.08E+03	1.08E+00
	worst	2.19E+02	3.29E+01	6.11E+03	8.63E+02	4.65E+02	5.85E+03	7.51E+05	5.63E+04	7.27E+03	6.90E+03	2.31E+02
	mean	2.17E+02	2.95E+01	5.06E+03	5.85E+02	4.65E+02	4.38E+03	5.83E+05	3.46E+03	4.71E+03	6.00E+03	5.15E+01
	Std	1.55E+00	1.27E+00	3.35E+02	4.81E+01	4.32E-02	1.15E+03	5.88E+04	1.02E+04	2.23E+03	3.59E+02	8.96E+01
F12	Best	6.02E+01	3.06E+01	2.04E+03	1.63E+02	2.42E+02	6.81E+02	3.58E+02	1.84E+00	4.15E+01	1.52E+03	1.27E+01
	worst	6.25E+01	3.45E+01	3.34E+03	2.44E+02	2.53E+02	2.54E+03	3.72E+05	2.92E+04	2.94E+03	3.51E+03	1.63E+01
	mean	6.20E+01	3.13E+01	2.59E+03	1.85E+02	2.44E+02	1.76E+03	2.66E+05	1.86E+03	2.15E+03	2.62E+03	1.30E+01
	Std	5.56E-01	5.89E-01	2.34E+02	1.52E+01	1.94E+00	4.47E+02	3.93E+04	4.10E+03	8.88E+02	2.44E+02	3.67E+00
F13	Best	7.47E+01	3.64E+01	4.76E+02	3.30E+02	1.37E+02	2.75E+02	2.01E+02	3.31E-05	6.77E+01	3.59E+02	1.77E-01
	worst	7.75E+01	4.27E+01	5.73E+02	3.70E+02	1.43E+02	5.70E+02	6.84E+03	2.12E+03	7.07E+02	6.91E+02	2.44E+02
	mean	7.69E+01	3.73E+01	5.21E+02	3.52E+02	1.38E+02	4.71E+02	5.80E+03	2.56E+02	5.44E+02	6.05E+02	7.20E+01
	Std	5.52E-01	8.15E-01	1.72E+01	1.18E+01	1.12E+00	5.48E+01	4.26E+02	4.03E+02	1.44E+02	3.60E+01	8.99E+01
F14	Best	1.27E+01	7.61E+00	7.48E+01	1.86E+01	1.13E+01	3.88E+01	2.57E+01	1.65E-10	9.49E+00	4.38E+01	5.75E-03
	worst	1.31E+01	8.04E+00	9.04E+01	2.29E+01	1.13E+01	8.88E+01	8.49E+03	8.54E+02	9.91E+01	9.61E+01	2.76E+01
	mean	1.30E+01	7.74E+00	8.25E+01	2.03E+01	1.13E+01	6.80E+01	6.82E+03	5.49E+01	7.90E+01	8.68E+01	5.40E+00
	Std	7.93E-02	9.58E-02	3.44E+00	7.56E-01	1.11E-03	1.35E+01	6.75E+02	1.14E+02	1.96E+01	4.46E+00	8.60E+00
F15	Best	1.76E-04	4.66E-56	6.65E-28	2.86E-02	7.79E-10	4.03E-06	3.92E-46	5.11E-17	6.60E-05	1.64E-33	1.33E-179
	worst	1.76E-04	1.17E-39	2.51E-07	1.20E-01	3.44E-07	6.19E-01	6.91E+08	1.51E+04	4.05E+04	3.57E+04	1.21E-04
	mean	1.76E-04	1.22E-41	2.51E-09	7.29E-02	5.87E-08	1.85E-02	2.14E+07	4.66E+02	3.76E+03	3.56E+03	2.85E-05
	Std	2.72E-20	1.17E-40	2.51E-08	2.36E-02	6.29E-08	8.71E-02	7.75E+07	1.91E+03	7.31E+03	7.20E+03	4.64E-05
F16	Best	1.89E-04	2.23E-15	6.82E-06	2.81E-06	4.46E-08	1.45E-09	1.66E-06	8.14E-07	1.58E-04	8.09E-07	3.95E-10
	worst	1.89E-04	2.21E+02	3.25E-01	2.69E-03	1.14E-06	1.18E-02	5.18E+10	2.02E+06	1.12E+06	1.57E+06	1.14E-02
	mean	1.89E-04	4.53E+00	4.05E-02	1.05E-03	1.73E-07	8.33E-04	1.77E+09	8.00E+04	8.95E+04	1.49E+05	1.75E-03
	Std	1.91E-19	2.52E+01	5.37E-02	7.46E-04	2.04E-07	1.74E-03	5.73E+09	3.46E+05	2.15E+05	2.81E+05	3.04E-03
F17	Best	2.04E+02	1.23E+01	1.27E+03	2.19E+02	3.40E+02	6.99E+02	3.00E+02	3.53E+00	2.08E+02	7.07E+02	7.59E+01
	worst	2.08E+02	1.51E+01	1.86E+03	2.86E+02	3.49E+02	2.21E+03	3.97E+05	2.52E+04	4.13E+03	4.17E+03	8.31E+02
	mean	2.07E+02	1.32E+01	1.53E+03	2.37E+02	3.41E+02	1.64E+03	2.50E+05	2.66E+03	2.46E+03	2.74E+03	3.01E+02
	Std	8.31E-01	6.03E-01	1.11E+02	1.40E+01	1.81E+00	3.40E+02	6.16E+04	5.61E+03	9.79E+02	6.67E+02	2.88E+02
F18	Best	2.41E+05	3.13E+03	5.11E+05	5.46E+05	4.34E+05	3.01E+05	3.97E+04	2.07E-03	1.22E+05	2.50E+05	6.97E-01
	worst	2.44E+05	3.92E+03	6.28E+05	6.54E+05	4.34E+05	4.89E+08	2.58E+12	9.00E+10	9.76E+09	9.91E+09	2.99E+03
	mean	2.43E+05	3.33E+03	5.59E+05	6.15E+05	4.34E+05	6.10E+06	3.03E+11	1.73E+09	2.57E+09	3.87E+09	5.44E+02
	Std	8.85E+02	1.49E+02	2.42E+04	2.92E+04	6.76E+01	4.90E+07	4.49E+11	1.04E+10	2.65E+09	3.48E+09	7.93E+02
F19	Best	2.64E+01	8.70E+00	7.28E+01	3.00E+01	3.00E+01	4.75E+01	2.02E+01	3.00E-01	9.75E+00	4.92E+01	2.10E+00
	worst	2.68E+01	1.12E+01	8.00E+01	3.40E+01	3.00E+01	8.52E+01	9.22E+02	2.19E+02	9.00E+01	8.91E+01	2.48E+01
	mean	2.67E+01	9.98E+00	7.65E+01	3.11E+01	3.00E+01	7.24E+01	8.10E+02	3.83E+01	7.38E+01	8.23E+01	5.59E+00
	Std	8.94E-02	6.88E-01	1.75E+00	9.11E-01	5.71E-08	9.73E+00	4.51E+01	4.83E+01	2.09E+01	2.86E+00	6.52E+00
F20	Best	3.85E+01	8.94E-50	6.89E-25	1.99E+00	6.80E+03	5.19E-01	9.49E-45	2.22E-14	5.44E+01	2.22E-29	4.43E-186
	worst	3.85E+01	3.40E-33	1.60E-08	3.20E+02	6.82E+03	1.09E+08	1.63E+12	1.87E+09	9.95E+09	9.91E+09	8.81E-08

	mean	3.85E+01	4.23E-35	1.60E-10	1.30E+02	6.80E+03	1.12E+06	3.13E+11	3.80E+07	3.02E+09	3.00E+09	3.25E-08
	Std	7.86E-14	3.42E-34	1.60E-09	8.86E+01	2.94E+00	1.09E+07	3.84E+11	2.17E+08	2.85E+09	3.07E+09	3.92E-08
F21	Best	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100
	worst	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100
	mean	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100
	Std	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.91E+84	3.90E+84	3.90E+84	3.90E+84
F22	Best	3.03E+04	3.01E+03	4.80E+05	1.09E+05	7.41E+04	1.54E+05	2.10E+04	5.60E-01	9.18E+03	2.34E+05	7.02E-01
	worst	3.15E+04	3.78E+03	6.58E+05	1.50E+05	7.42E+04	6.67E+05	8.00E+07	3.98E+06	7.52E+05	7.73E+05	2.34E+04
	mean	3.12E+04	3.26E+03	5.71E+05	1.31E+05	7.41E+04	4.76E+05	6.40E+07	4.11E+05	5.72E+05	6.75E+05	9.01E+03
	Std	2.43E+02	1.62E+02	3.17E+04	6.68E+03	2.95E+01	1.37E+05	6.65E+06	8.37E+05	2.21E+05	4.09E+04	1.08E+04
F23	Best	4.18E+08	2.17E+07	7.02E+09	6.39E+08	6.45E+09	6.90E+09	1.21E+08	6.82E-01	4.36E+08	1.98E+09	1.04E+06
	worst	4.38E+08	6.07E+07	1.53E+10	1.25E+09	6.48E+09	3.79E+10	8.69E+12	2.17E+11	7.32E+10	7.30E+10	4.70E+09
	mean	4.34E+08	2.44E+07	9.23E+09	8.30E+08	6.45E+09	2.44E+10	4.77E+12	1.90E+10	4.20E+10	5.08E+10	3.41E+08
	Std	3.71E+06	4.30E+06	1.35E+09	8.24E+07	6.46E+06	6.56E+09	1.15E+12	4.02E+10	1.81E+10	7.76E+09	1.00E+09

Table 5: Comparison of proposed algorithm with other meta-heiurustic on functions 1-32 with dimensions 50

		GA	PSO	ABC	FA	BAT	FPA	GWO	WOA	CSA	FFA	MAMH
F1	Best	1.26E+05	3.75E+04	1.52E+06	1.35E+05	1.79E+05	5.36E+05	2.60E+05	4.81E-08	2.03E+04	8.03E+05	5.12E-06
	worst	1.28E+05	4.38E+04	1.82E+06	2.23E+05	1.79E+05	1.72E+06	1.99E+08	7.01E+06	1.92E+06	1.83E+06	2.04E-04
	mean	1.27E+05	4.01E+04	1.66E+06	1.45E+05	1.79E+05	1.31E+06	1.70E+08	5.77E+05	1.42E+06	1.66E+06	1.10E-04
	Std	2.86E+02	1.25E+03	6.27E+04	1.22E+04	6.83E+01	3.22E+05	1.24E+07	1.20E+06	5.42E+05	6.75E+04	9.37E-05
F2	Best	3.16E+05	7.47E+04	3.70E+06	2.84E+06	3.47E+05	1.22E+06	5.04E+05	7.06E-06	5.88E+04	1.89E+06	9.25E-05
	worst	3.20E+05	8.16E+04	4.49E+06	3.03E+06	3.54E+05	4.52E+06	5.21E+08	5.51E+07	4.70E+06	4.59E+06	2.88E-02
	mean	3.19E+05	7.75E+04	4.10E+06	2.96E+06	3.48E+05	3.24E+06	4.17E+08	4.79E+06	3.74E+06	4.15E+06	1.74E-02
	Std	1.10E+03	1.32E+03	1.65E+05	4.75E+04	1.47E+03	9.07E+05	3.32E+07	1.02E+07	1.10E+06	2.04E+05	1.38E-02
F3	Best	2.60E+07	9.13E+06	3.76E+08	3.53E+08	3.53E+07	8.91E+07	4.64E+07	5.14E-03	4.16E+06	1.93E+08	2.75E-09
	worst	2.66E+07	9.64E+06	4.55E+08	3.63E+08	3.54E+07	4.19E+08	5.07E+10	3.29E+09	4.58E+08	4.71E+08	7.43E-08
	mean	2.65E+07	9.29E+06	4.13E+08	3.59E+08	3.54E+07	3.17E+08	4.15E+10	3.39E+08	3.72E+08	4.20E+08	3.66E-08
	Std	9.30E+04	9.63E+04	1.51E+07	2.55E+06	1.95E+04	8.85E+07	3.09E+09	6.06E+08	1.15E+08	2.04E+07	3.32E-08
F4	Best	4.27E+07	1.41E+07	6.99E+09	7.08E+08	1.31E+08	1.12E+09	4.97E+08	4.94E+02	1.84E+06	4.41E+09	4.94E+02
	worst	4.43E+07	1.55E+07	9.31E+09	9.00E+08	1.34E+08	8.62E+09	2.25E+14	3.39E+12	9.28E+09	9.14E+09	4.95E+02
	mean	4.37E+07	1.48E+07	8.11E+09	7.40E+08	1.32E+08	5.96E+09	1.69E+14	1.20E+11	6.58E+09	8.10E+09	4.94E+02
	Std	3.75E+05	2.98E+05	4.83E+08	6.89E+07	3.41E+05	2.15E+09	2.84E+13	4.67E+11	2.92E+09	4.49E+08	8.73E-02
F5	Best	1.18E+05	4.40E+04	1.50E+06	5.63E+05	1.92E+05	5.51E+05	2.40E+05	3.41E+00	2.11E+04	7.74E+05	6.91E+00
	worst	1.19E+05	4.90E+04	1.81E+06	6.27E+05	1.93E+05	1.73E+06	2.11E+08	5.35E+06	1.86E+06	1.81E+06	9.43E+00
	mean	1.18E+05	4.59E+04	1.67E+06	5.80E+05	1.92E+05	1.28E+06	1.67E+08	6.67E+05	1.46E+06	1.66E+06	8.99E+00
	Std	2.84E+02	9.28E+02	7.01E+04	1.76E+04	5.54E+01	3.41E+05	1.27E+07	1.29E+06	4.39E+05	6.29E+04	8.69E-01
F6	Best	3.84E+02	1.12E+02	5.87E+04	3.96E+02	1.22E+02	8.85E+03	3.77E+03	1.51E-02	2.39E+01	3.11E+04	3.55E-02

	worst	3.95E+02	1.27E+02	8.50E+04	1.11E+03	1.37E+02	7.15E+04	1.95E+09	1.60E+06	7.41E+04	7.62E+04	1.08E-01
	mean	3.91E+02	1.17E+02	6.86E+04	5.16E+02	1.25E+02	4.65E+04	1.34E+09	8.36E+04	5.66E+04	6.72E+04	8.98E-02
	Std	2.99E+00	2.69E+00	4.88E+03	9.31E+01	2.43E+00	1.77E+04	2.03E+08	2.44E+05	2.26E+04	4.78E+03	1.77E-02
F7	Best	3.90E+00	1.94E+00	2.29E+02	3.33E+00	7.69E+00	2.38E+01	2.55E+01	4.15E-03	1.90E-01	1.48E+02	2.16E-02
	worst	4.03E+00	3.07E+00	3.08E+02	8.75E+00	7.96E+00	2.76E+02	9.55E+06	1.68E+05	3.12E+02	3.06E+02	6.34E-02
	mean	3.98E+00	2.48E+00	2.69E+02	4.35E+00	7.80E+00	1.86E+02	6.09E+06	3.56E+03	2.13E+02	2.68E+02	4.34E-02
	Std	3.16E-02	3.30E-01	1.56E+01	7.71E-01	5.67E-02	7.08E+01	1.08E+06	1.91E+04	1.01E+02	1.57E+01	1.20E-02
F8	Best	2.02E-05	1.09E-19	2.30E+00	2.56E-06	5.90E-08	1.01E-01	4.04E-03	1.45E-14	2.44E-05	2.40E+00	2.02E-22
	worst	2.02E-05	1.84E-13	9.80E+00	1.63E-05	2.11E-07	7.46E+00	1.00E+100	7.85E+301	9.01E+00	9.01E+00	2.86E-17
	mean	2.02E-05	3.02E-15	5.36E+00	5.64E-06	9.45E-08	3.11E+00	1.00E+100	9.82E+299	4.19E+00	5.30E+00	1.30E-17
	Std	1.80E-13	1.93E-14	1.59E+00	3.53E-06	3.03E-08	1.99E+00	3.90E+84	2.43E-01	2.35E+00	1.48E+00	1.36E-17
F9	Best	3.85E+03	2.69E+03	8.38E+03	5.04E+03	1.98E+03	5.72E+03	5.47E+03	8.22E-07	4.68E+03	7.40E+03	3.32E-10
	worst	3.91E+03	2.82E+03	9.14E+03	5.19E+03	2.39E+03	8.98E+03	5.70E+05	6.24E+04	9.85E+03	1.00E+04	2.75E-09
	mean	3.89E+03	2.73E+03	8.83E+03	5.06E+03	2.03E+03	7.91E+03	4.79E+05	5.94E+03	8.68E+03	9.27E+03	1.35E-09
	Std	1.36E+01	2.29E+01	1.16E+02	7.27E+01	6.02E+01	7.97E+02	3.58E+04	8.87E+03	1.29E+03	2.23E+02	1.22E-09
F10	Best	1.46E+01	1.02E+01	2.10E+01	1.49E+01	1.46E+01	1.58E+01	1.70E+01	2.45E-04	8.60E+00	2.05E+01	9.59E-07
	worst	1.46E+01	1.04E+01	2.11E+01	1.64E+01	1.48E+01	2.05E+01	2.18E+01	2.17E+01	2.13E+01	2.13E+01	1.18E-04
	mean	1.46E+01	1.03E+01	2.11E+01	1.52E+01	1.47E+01	1.89E+01	2.17E+01	1.09E+01	2.00E+01	2.12E+01	5.83E-05
	Std	1.00E-02	4.96E-02	1.94E-02	1.76E-01	2.29E-02	9.65E-01	3.24E-02	8.39E+00	3.12E+00	4.76E-02	4.71E-05
F11	Best	9.35E+02	3.15E+02	1.34E+04	1.09E+04	1.94E+03	3.22E+03	1.98E+03	8.40E-08	1.57E+02	7.58E+03	3.50E-07
	worst	9.47E+02	3.38E+02	1.69E+04	1.15E+04	1.94E+03	1.50E+04	1.71E+06	8.25E+04	1.64E+04	1.67E+04	1.81E-05
	mean	9.44E+02	3.24E+02	1.50E+04	1.12E+04	1.94E+03	1.15E+04	1.52E+06	7.29E+03	1.29E+04	1.51E+04	7.65E-06
	Std	2.85E+00	4.31E+00	6.95E+02	1.51E+02	9.40E-02	3.17E+03	8.61E+04	1.45E+04	4.63E+03	6.22E+02	7.51E-06
F12	Best	5.21E+02	2.27E+02	5.60E+03	1.17E+03	8.17E+02	1.89E+03	1.67E+03	1.25E+00	1.26E+02	5.20E+03	1.10E+01
	worst	5.31E+02	2.50E+02	7.13E+03	1.66E+03	8.44E+02	6.51E+03	8.43E+05	1.01E+05	7.32E+03	7.34E+03	1.49E+01
	mean	5.29E+02	2.32E+02	6.49E+03	1.31E+03	8.22E+02	5.14E+03	6.74E+05	5.45E+03	5.65E+03	6.44E+03	1.25E+01
	Std	2.37E+00	3.59E+00	3.15E+02	8.51E+01	4.64E+00	1.08E+03	6.37E+04	1.46E+04	1.91E+03	3.59E+02	1.40E+00
F13	Best	3.29E+02	1.94E+02	1.35E+03	9.93E+02	4.65E+02	8.69E+02	7.16E+02	1.42E-05	1.67E+02	1.06E+03	2.22E-06
	worst	3.33E+02	2.07E+02	1.47E+03	1.10E+03	4.72E+02	1.45E+03	1.66E+04	3.00E+03	1.63E+03	1.62E+03	6.66E-04
	mean	3.32E+02	1.98E+02	1.42E+03	1.05E+03	4.66E+02	1.26E+03	1.46E+04	5.57E+02	1.34E+03	1.50E+03	1.63E-04
	Std	5.84E-01	2.19E+00	2.71E+01	2.52E+01	1.39E+00	1.35E+02	8.19E+02	7.89E+02	3.77E+02	5.23E+01	2.50E-04
F14	Best	4.86E+01	3.41E+01	1.95E+02	8.11E+01	2.27E+01	1.06E+02	6.66E+01	4.76E-09	2.52E+01	1.38E+02	1.29E-09
	worst	4.90E+01	3.56E+01	2.31E+02	8.87E+01	2.33E+01	2.23E+02	2.01E+04	1.15E+03	2.34E+02	2.34E+02	7.70E-07
	mean	4.88E+01	3.47E+01	2.17E+02	8.23E+01	2.28E+01	1.78E+02	1.70E+04	1.00E+02	1.98E+02	2.17E+02	7.00E-08
	Std	9.27E-02	2.50E-01	6.84E+00	1.60E+00	1.34E-01	3.45E+01	1.18E+03	1.79E+02	4.66E+01	7.04E+00	1.99E-07
F15	Best	4.35E-10	3.28E-56	2.75E-28	1.16E-02	2.77E-10	7.03E-07	2.24E-38	7.52E-16	4.37E-05	1.28E-33	5.84E-20
	worst	4.35E-10	2.48E-39	4.69E-12	6.54E-02	3.50E-07	3.24E+00	4.34E+08	1.18E+05	3.30E+04	3.35E+04	8.64E-17
	mean	4.35E-10	4.24E-41	4.69E-14	3.88E-02	7.67E-08	4.49E-02	2.65E+07	3.60E+03	2.48E+03	3.39E+03	2.32E-17
	Std	5.20E-25	3.01E-40	4.69E-13	1.48E-02	6.67E-08	3.27E-01	7.18E+07	1.70E+04	5.64E+03	6.17E+03	3.35E-17
F16	Best	8.35E-03	4.80E-10	6.83E-06	4.73E-07	1.71E-04	3.80E-07	4.35E-05	5.60E-07	2.85E-05	4.62E-06	2.53E-07

	worst	8.35E-03	1.59E+02	3.84E-01	3.86E-03	2.50E-04	8.25E-03	1.02E+11	3.31E+06	1.49E+06	1.12E+06	9.39E-07
	mean	8.35E-03	3.30E+00	3.92E-02	1.46E-03	2.04E-04	6.12E-04	3.80E+09	4.84E+04	1.14E+05	9.28E+04	7.98E-07
	Std	2.27E-17	1.65E+01	5.60E-02	1.12E-03	1.85E-05	1.03E-03	1.17E+10	3.72E+05	2.56E+05	1.94E+05	2.04E-07
F17	Best	8.81E+02	1.67E+02	4.16E+03	7.84E+02	1.26E+03	2.48E+03	9.64E+02	2.97E+00	5.95E+02	2.66E+03	5.80E+01
	worst	8.85E+02	1.88E+02	5.39E+03	1.01E+03	1.27E+03	5.95E+03	9.36E+05	5.82E+04	1.03E+04	9.80E+03	6.57E+01
	mean	8.84E+02	1.76E+02	4.63E+03	8.49E+02	1.26E+03	4.37E+03	6.01E+05	8.19E+03	6.55E+03	6.80E+03	6.19E+01
	Std	1.08E+00	4.46E+00	2.18E+02	3.95E+01	2.60E+00	7.02E+02	1.43E+05	1.35E+04	2.41E+03	1.57E+03	2.47E+00
F18	Best	3.96E+05	4.39E+04	1.59E+06	1.41E+05	1.77E+06	5.53E+05	3.06E+05	8.48E-05	1.94E+05	8.38E+05	1.44E-05
	worst	4.00E+05	4.64E+04	1.88E+06	2.31E+05	1.77E+06	4.79E+08	1.64E+12	1.34E+10	9.56E+09	9.98E+09	1.66E-03
	mean	3.99E+05	4.49E+04	1.74E+06	1.63E+05	1.77E+06	7.14E+06	2.86E+11	7.01E+08	2.67E+09	3.82E+09	4.79E-04
	Std	5.01E+02	5.64E+02	4.97E+04	1.92E+04	1.64E+02	4.81E+07	3.95E+11	2.30E+09	2.65E+09	3.22E+09	6.27E-04
F19	Best	3.27E+01	2.71E+01	1.22E+02	6.67E+01	4.46E+01	5.68E+01	5.40E+01	2.00E-01	1.49E+01	9.41E+01	1.90E-04
	worst	3.29E+01	3.00E+01	1.35E+02	7.21E+01	4.46E+01	1.34E+02	1.46E+03	4.00E+02	1.36E+02	1.36E+02	2.04E-01
	mean	3.29E+01	2.86E+01	1.30E+02	6.83E+01	4.46E+01	1.14E+02	1.31E+03	7.17E+01	1.18E+02	1.30E+02	1.12E-01
	Std	3.46E-02	8.19E-01	2.81E+00	1.09E+00	1.21E-07	1.78E+01	4.51E+01	8.91E+01	2.84E+01	2.44E+00	7.32E-02
F20	Best	1.63E+04	2.35E-50	3.27E-25	1.71E+01	3.93E+01	9.59E-03	3.94E-36	3.15E-13	4.35E+02	6.13E-29	1.11E-19
	worst	1.63E+04	2.98E-30	1.13E-10	2.99E+02	4.04E+01	1.35E+08	2.18E+12	2.84E+10	9.96E+09	9.85E+09	2.94E-17
	mean	1.63E+04	3.11E-32	1.13E-12	1.32E+02	3.96E+01	1.42E+06	2.97E+11	9.12E+08	3.20E+09	3.11E+09	1.43E-17
	Std	7.31E-12	2.98E-31	1.13E-11	8.96E+01	2.26E-01	1.35E+07	3.78E+11	4.17E+09	3.20E+09	2.94E+09	1.22E-17
F21	Best	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	0.00E+00
	worst	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100
	mean	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100
	Std	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.91E+84	3.90E+84	3.90E+84	3.90E+84
F22	Best	1.24E+05	3.97E+04	1.47E+06	5.34E+05	2.49E+05	4.07E+05	2.43E+05	8.53E-01	1.65E+04	8.03E+05	5.04E-01
	worst	1.25E+05	4.47E+04	1.81E+06	6.06E+05	2.49E+05	1.68E+06	1.97E+08	1.08E+07	1.82E+06	1.88E+06	5.21E-01
	mean	1.25E+05	4.20E+04	1.66E+06	5.69E+05	2.49E+05	1.26E+06	1.66E+08	9.57E+05	1.51E+06	1.67E+06	5.09E-01
	Std	2.38E+02	9.47E+02	6.04E+04	1.66E+04	6.73E+01	3.36E+05	1.02E+07	2.18E+06	4.44E+05	6.33E+04	5.34E-03
F23	Best	2.77E+09	4.07E+08	5.16E+10	1.53E+10	1.10E+10	2.29E+10	2.37E+09	9.08E-04	9.62E+08	2.05E+10	1.50E-09
	worst	2.86E+09	5.13E+08	8.75E+10	1.97E+10	1.11E+10	1.14E+11	1.74E+13	2.97E+11	1.52E+11	1.53E+11	1.14E+02
	mean	2.84E+09	4.22E+08	6.26E+10	1.75E+10	1.10E+10	7.80E+10	1.16E+13	4.68E+10	9.95E+10	1.22E+11	2.03E+01
	Std	1.78E+07	1.53E+07	5.64E+09	9.92E+08	1.21E+07	2.23E+10	1.87E+12	7.98E+10	4.36E+10	1.20E+10	4.00E+01

According to the results obtained from executing the proposed algorithm and other comparative algorithms on dimensions of 200 and 500 which are provided in ***Tables (4) & (5)***, respectively, the proposed algorithm has maintained its efficiency despite the increase in dimensions of the optimization problem, whereas some of the comparative algorithms lost their efficiency with increase in the dimensions. In the test with 200 dimensions, the proposed algorithm could appropriately improve its population, in addition to the best feasible solution, with regard to the worst, average, and standard deviation values. Notwithstanding, it exhibited poor performance in a few of the functions. In the test with 500 dimensions in ***Table (5)***, the proposed algorithm could prove more of its efficiency and exhibited better performance than most of the comparative algorithms in most of the functions. These results indicate that the proposed algorithm could make best use of the processes of different powerful agents during the execution time so that it could, on the one hand, establish desirable interactions among them and, on the other hand, excluded the weak agents in order to exhibit a much better performance at higher dimensions. To provide a better view of the results based on ***Table (5)***, these results are illustrated graphically in ***Figures (10)-(12)***. It should be noted that the calling order of functions was assumed equal to 20000 in all tests. Subsequently, the proposed algorithm and other ones were tested on functions (1) to (23) with 1000 dimensions, the results of which are presented in ***Table (6)***.

Table 6: Comparison of proposed algorithm with other meta-heuirustic on functions 1-32 with dimensions 1000

		GA	PSO	ABC	FA	BAT	FPA	GWO	WOA	CSA	FFA	MAMH
F1	Best	2.83E+05	1.81E+05	3.10E+06	5.71E+05	4.32E+05	9.45E+05	7.47E+05	1.56E-04	4.86E+04	1.90E+06	**8.42E-05**
	worst	2.85E+05	1.96E+05	3.66E+06	7.28E+05	4.33E+05	3.46E+06	4.00E+08	1.63E+07	3.53E+06	3.57E+06	**9.27E+00**
	mean	2.84E+05	1.88E+05	3.34E+06	5.97E+05	4.32E+05	2.63E+06	3.52E+08	1.04E+06	2.89E+06	3.34E+06	**1.94E+00**
	Std	3.57E+02	2.73E+03	9.64E+04	1.85E+04	1.10E+02	5.99E+05	1.82E+07	2.63E+06	1.02E+06	8.37E+04	**2.78E+00**
F2	Best	1.62E+06	8.00E+05	1.54E+07	1.80E+06	1.88E+06	4.40E+06	3.37E+06	5.66E-04	2.05E+05	9.03E+06	**5.31E-05**
	worst	1.63E+06	8.70E+05	1.80E+07	2.85E+06	1.90E+06	1.68E+07	1.98E+09	6.20E+07	1.79E+07	1.79E+07	**3.24E-02**
	mean	1.62E+06	8.32E+05	1.66E+07	1.92E+06	1.88E+06	1.24E+07	1.73E+09	4.71E+06	1.52E+07	1.66E+07	**1.28E-02**
	Std	2.32E+03	1.35E+04	5.51E+05	1.57E+05	4.20E+03	3.72E+06	9.49E+07	1.00E+07	4.14E+06	5.04E+05	**1.47E-02**
F3	Best	1.84E+08	7.28E+07	1.53E+09	1.21E+08	1.65E+08	5.29E+08	3.37E+08	1.31E-02	2.18E+07	8.65E+08	**6.13E-04**
	worst	1.85E+08	7.70E+07	1.83E+09	2.00E+08	1.66E+08	1.66E+09	2.06E+11	8.74E+09	1.83E+09	1.82E+09	**4.85E+00**
	mean	1.85E+08	7.46E+07	1.67E+09	1.29E+08	1.65E+08	1.24E+09	1.73E+11	1.07E+09	1.34E+09	1.67E+09	**2.61E+00**
	Std	2.53E+05	8.96E+05	6.25E+07	1.03E+07	5.38E+04	3.39E+08	1.06E+10	2.11E+09	5.76E+08	5.03E+07	**2.31E+00**
F4	Best	1.46E+08	1.14E+08	1.42E+10	4.06E+09	1.45E+08	1.23E+09	2.31E+09	9.89E+02	4.88E+06	1.15E+10	**9.90E+02**
	worst	1.48E+08	1.25E+08	1.79E+10	5.36E+09	1.46E+08	1.68E+10	5.41E+14	1.71E+13	1.82E+10	1.79E+10	**9.90E+02**
	mean	1.48E+08	1.19E+08	1.62E+10	4.23E+09	1.45E+08	1.08E+10	3.92E+14	6.10E+11	1.36E+10	1.63E+10	**9.90E+02**
	Std	3.32E+05	2.20E+06	6.63E+08	2.22E+08	2.82E+05	4.83E+09	4.50E+13	2.14E+12	5.78E+09	6.80E+08	**1.24E-01**
F5	Best	2.61E+05	1.75E+05	3.05E+06	2.81E+05	4.22E+05	1.23E+06	7.61E+05	6.46E+00	4.26E+04	1.94E+06	**3.64E+01**
	worst	2.62E+05	1.87E+05	3.62E+06	4.40E+05	4.23E+05	3.39E+06	3.90E+08	2.75E+07	3.51E+06	3.70E+06	**3.78E+01**
	mean	2.62E+05	1.81E+05	3.33E+06	3.06E+05	4.22E+05	2.79E+06	3.51E+08	3.44E+06	3.04E+06	3.32E+06	**3.67E+01**
	Std	2.56E+02	2.23E+03	9.60E+04	1.67E+04	8.03E+01	5.76E+05	1.82E+07	6.74E+06	8.30E+05	9.86E+04	**8.64E-01**
F6	Best	3.06E+03	1.16E+03	2.33E+05	4.32E+03	8.57E+02	3.87E+04	3.02E+04	1.79E-01	5.10E+01	1.86E+05	**1.34E-02**
	worst	3.08E+03	1.29E+03	3.01E+05	8.35E+03	1.01E+03	2.76E+05	8.59E+09	5.44E+08	3.03E+05	2.95E+05	**1.08E-01**
	mean	3.07E+03	1.21E+03	2.71E+05	4.58E+03	8.93E+02	2.01E+05	6.20E+09	1.55E+07	2.27E+05	2.68E+05	**5.80E-02**
	Std	6.60E+00	2.67E+01	1.41E+04	5.07E+02	3.07E+01	6.76E+04	8.31E+08	7.74E+07	9.20E+04	1.26E+04	**2.55E-02**
F7	Best	9.24E+00	4.53E+00	4.83E+02	5.23E+00	1.17E+01	5.16E+01	7.44E+01	**6.30E-03**	2.07E-01	3.99E+02	3.15E-02
	worst	9.35E+00	5.60E+00	5.88E+02	1.27E+01	1.19E+01	5.48E+02	1.85E+07	1.35E+05	5.84E+02	6.07E+02	**1.19E-01**
	mean	9.32E+00	5.04E+00	5.36E+02	6.27E+00	1.18E+01	3.89E+02	1.29E+07	5.45E+03	4.73E+02	5.36E+02	**8.40E-02**
	Std	2.52E-02	2.85E-01	2.18E+01	8.36E-01	4.70E-02	1.52E+02	1.62E+06	1.89E+04	1.69E+02	2.38E+01	**2.46E-02**
F8	Best	2.64E-05	1.54E-18	2.75E+00	2.10E+00	9.50E-08	5.29E-02	1.00E-02	1.70E-20	7.47E-06	2.13E+00	**6.36E-21**
	worst	2.66E-05	1.00E+00	1.15E+01	4.84E+00	2.95E-07	1.05E+01	1.00E+100	6.11E+224	9.97E+00	9.92E+00	**1.38E-17**
	mean	2.64E-05	4.11E-02	6.15E+00	3.70E+00	1.44E-07	4.44E+00	1.00E+100	7.64E+222	4.64E+00	5.63E+00	**6.84E-18**

	Std	7.61E-08	1.97E-01	1.52E+00	9.35E-01	3.29E-08	2.88E+00	3.90E+84	3.17E+01	2.88E+00	1.54E+00	**6.35E-18**
F9	Best	9.19E+03	7.00E+03	1.77E+04	1.25E+04	5.94E+03	1.22E+04	1.18E+04	1.68E-05	9.42E+03	1.53E+04	**1.07E-09**
	worst	9.23E+03	7.21E+03	1.86E+04	1.28E+04	6.34E+03	1.84E+04	1.11E+06	9.12E+04	1.95E+04	1.93E+04	**9.39E-09**
	mean	9.22E+03	7.09E+03	1.81E+04	1.26E+04	6.01E+03	1.64E+04	9.70E+05	9.71E+03	1.70E+04	1.86E+04	**6.68E-09**
	Std	7.32E+00	4.81E+01	1.78E+02	1.22E+02	7.88E+01	1.57E+03	5.23E+04	1.52E+04	2.76E+03	3.06E+02	**3.73E-09**
F10	Best	1.50E+01	1.36E+01	2.11E+01	1.60E+01	1.60E+01	1.49E+01	1.84E+01	**1.64E-04**	9.03E+00	2.06E+01	6.45E-04
	worst	1.50E+01	1.37E+01	2.11E+01	1.72E+01	1.60E+01	2.01E+01	2.18E+01	2.17E+01	2.13E+01	2.13E+01	**2.07E-03**
	mean	1.50E+01	1.37E+01	2.11E+01	1.62E+01	1.60E+01	1.87E+01	2.17E+01	1.03E+01	2.00E+01	2.12E+01	**7.37E-04**
	Std	3.92E-03	1.50E-02	1.20E-02	1.46E-01	1.04E-02	1.02E+00	2.29E-02	8.57E+00	2.83E+00	3.32E-02	**1.91E-04**
F11	Best	2.29E+03	1.43E+03	2.82E+04	2.64E+03	2.62E+03	7.38E+03	6.51E+03	2.93E-07	4.27E+02	1.71E+04	**2.48E-09**
	worst	2.30E+03	1.52E+03	3.21E+04	3.90E+03	2.62E+03	3.04E+04	3.46E+06	1.66E+05	3.13E+04	3.19E+04	**1.06E-04**
	mean	2.29E+03	1.47E+03	3.00E+04	2.72E+03	2.62E+03	2.33E+04	3.14E+06	1.41E+04	2.50E+04	2.99E+04	**3.38E-05**
	Std	2.49E+00	1.97E+01	7.08E+02	1.47E+02	1.14E-01	6.13E+03	1.54E+05	2.93E+04	9.99E+03	8.21E+02	**4.52E-05**
F12	Best	1.16E+03	6.42E+02	1.17E+04	1.37E+03	1.85E+03	4.40E+03	4.34E+03	4.46E+00	2.23E+02	1.17E+04	**1.66E+01**
	worst	1.17E+03	6.84E+02	1.40E+04	1.90E+03	1.90E+03	1.30E+04	1.63E+06	6.01E+04	1.41E+04	1.40E+04	**1.74E+01**
	mean	1.17E+03	6.56E+02	1.29E+04	1.48E+03	1.86E+03	1.02E+04	1.36E+06	7.50E+03	1.12E+04	1.30E+04	**1.72E+01**
	Std	2.45E+00	8.29E+00	4.76E+02	6.22E+01	8.34E+00	2.55E+03	9.64E+04	1.36E+04	3.70E+03	4.88E+02	**2.33E-01**
F13	Best	8.48E+02	6.50E+02	2.77E+03	1.60E+03	8.75E+02	1.74E+03	1.61E+03	1.67E-04	3.09E+02	2.33E+03	**1.03E-06**
	worst	8.52E+02	6.67E+02	3.03E+03	1.82E+03	8.93E+02	2.94E+03	3.14E+04	4.28E+03	3.18E+03	3.22E+03	**1.41E-04**
	mean	8.51E+02	6.56E+02	2.93E+03	1.68E+03	8.78E+02	2.63E+03	2.93E+04	5.70E+02	2.73E+03	2.99E+03	**6.67E-05**
	Std	8.40E-01	3.95E+00	4.86E+01	4.04E+01	3.41E+00	2.75E+02	8.73E+02	8.30E+02	7.10E+02	8.11E+01	**6.45E-05**
F14	Best	1.17E+02	9.34E+01	4.10E+02	1.83E+02	5.69E+01	2.15E+02	1.73E+02	2.17E-08	4.45E+01	3.04E+02	**5.26E-11**
	worst	1.17E+02	9.60E+01	4.64E+02	1.97E+02	5.91E+01	4.39E+02	3.96E+04	2.61E+03	4.56E+02	4.60E+02	**1.92E-08**
	mean	1.17E+02	9.42E+01	4.35E+02	1.87E+02	5.75E+01	3.86E+02	3.56E+04	3.64E+02	3.85E+02	4.35E+02	**1.38E-09**
	Std	1.12E-01	5.27E-01	9.84E+00	2.88E+00	5.33E-01	5.64E+01	1.70E+03	5.94E+02	1.07E+02	9.33E+00	**2.76E-09**
F15	Best	2.48E-04	**6.81E-55**	4.42E-28	9.72E-07	2.94E-10	5.95E-09	2.31E-39	7.75E-17	9.48E-05	1.15E-33	1.01E-13
	worst	2.48E-04	**7.57E-36**	5.69E-14	1.36E-04	2.92E-07	1.49E+00	3.06E+08	1.00E+05	3.93E+04	3.87E+04	2.28E-10
	mean	2.48E-04	**7.93E-38**	5.69E-16	4.78E-05	6.80E-08	2.08E-02	1.19E+07	1.40E+03	3.83E+03	4.45E+03	5.34E-11
	Std	2.72E-19	**7.58E-37**	5.69E-15	4.10E-05	6.17E-08	1.52E-01	3.91E+07	1.12E+04	7.78E+03	7.47E+03	4.92E-11
F16	Best	2.75E-04	6.74E-12	7.18E-07	8.01E-05	1.30E-05	2.08E-07	3.40E-06	7.42E-09	1.97E-05	6.27E-06	**5.08E-13**
	worst	2.75E-04	6.63E+00	1.00E-01	5.49E-03	2.90E-05	7.52E-03	5.20E+10	2.58E+07	9.29E+05	1.45E+06	**1.30E-05**
	mean	2.75E-04	4.87E-01	3.54E-02	2.02E-03	1.78E-05	8.45E-04	3.38E+09	6.22E+05	1.20E+05	1.08E+05	**9.21E-06**
	Std	4.36E-19	8.54E-01	4.01E-02	1.53E-03	3.61E-06	1.38E-03	9.40E+09	3.40E+06	2.35E+05	2.22E+05	**3.58E-06**
F17	Best	2.64E+03	6.92E+02	8.94E+03	2.24E+03	2.12E+03	5.76E+03	3.14E+03	3.01E+00	1.19E+03	5.87E+03	**2.80E+02**
	worst	2.65E+03	7.57E+02	1.07E+04	2.84E+03	2.15E+03	1.17E+04	1.88E+06	1.03E+05	1.92E+04	1.98E+04	**2.90E+02**
	mean	2.65E+03	7.13E+02	9.89E+03	2.40E+03	2.13E+03	9.02E+03	1.33E+06	1.26E+04	1.13E+04	1.37E+04	**2.75E+02**
	Std	1.62E+00	**9.78E+00**	3.35E+02	1.46E+02	3.30E+00	1.30E+03	3.20E+05	2.16E+04	5.39E+03	3.40E+03	3.56E+01
F18	Best	9.20E+05	2.21E+05	3.32E+06	5.41E+05	3.99E+06	1.01E+06	8.90E+05	3.35E-03	4.19E+05	1.96E+06	**8.77E-04**
	worst	9.23E+05	2.36E+05	3.96E+06	7.50E+05	3.99E+06	3.34E+08	1.68E+12	1.44E+10	9.82E+09	9.83E+09	**5.11E-01**
	mean	9.22E+05	2.29E+05	3.76E+06	5.93E+05	3.99E+06	6.53E+06	3.13E+11	9.10E+08	2.94E+09	3.53E+09	**2.08E-01**

	Std	6.31E+02	2.92E+03	1.07E+05	3.45E+04	2.38E+02	3.32E+07	3.69E+11	2.76E+09	2.97E+09	2.90E+09	**2.13E-01**
F19	Best	5.72E+01	4.69E+01	1.76E+02	5.45E+01	5.75E+01	1.06E+02	8.96E+01	2.00E-01	2.01E+01	1.42E+02	**9.99E-02**
	worst	5.74E+01	4.94E+01	1.90E+02	6.55E+01	5.75E+01	1.86E+02	2.00E+03	6.00E+02	1.90E+02	1.89E+02	**9.99E-02**
	mean	5.73E+01	4.81E+01	1.83E+02	5.55E+01	5.75E+01	1.61E+02	1.88E+03	9.88E+01	1.69E+02	1.84E+02	**9.99E-02**
	Std	6.85E-02	7.55E-01	2.58E+00	1.37E+00	4.38E-08	2.30E+01	4.85E+01	1.23E+02	4.13E+01	2.32E+00	**3.74E-06**
F20	Best	7.63E+03	**4.18E-50**	1.52E-25	2.61E+01	8.99E+03	4.77E-02	7.94E-34	3.47E-10	1.34E+01	4.79E-31	3.60E-19
	worst	7.63E+03	**1.25E-29**	3.30E-07	5.42E+02	9.01E+03	5.42E+08	1.29E+12	7.65E+09	9.92E+09	9.74E+09	3.17E-17
	mean	7.63E+03	**1.25E-31**	3.30E-09	2.54E+02	8.99E+03	5.66E+06	2.29E+11	1.24E+08	3.35E+09	3.50E+09	1.14E-17
	Std	1.01E-11	**1.25E-30**	3.30E-08	1.51E+02	6.07E+00	5.42E+07	3.21E+11	8.58E+08	3.04E+09	3.13E+09	1.27E-17
F21	Best	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	**0.00E+00**
	worst	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	**1.00E+100**
	mean	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	1.00E+100	**1.00E+100**
	Std	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.90E+84	3.91E+84	3.90E+84	3.90E+84	**3.90E+84**
F22	Best	3.13E+05	1.70E+05	3.16E+06	2.80E+05	3.13E+05	7.68E+05	7.11E+05	1.06E+00	4.88E+04	1.87E+06	**5.64E-01**
	worst	3.15E+05	1.80E+05	3.58E+06	4.10E+05	3.13E+05	3.51E+06	3.89E+08	9.63E+07	3.54E+06	3.57E+06	**1.28E+00**
	mean	3.15E+05	1.74E+05	3.35E+06	2.89E+05	3.13E+05	2.71E+06	3.52E+08	7.69E+06	2.88E+06	3.33E+06	**6.82E-01**
	Std	4.10E+02	2.43E+03	8.70E+04	1.59E+04	7.59E+01	6.37E+05	1.69E+07	1.60E+07	1.03E+06	9.08E+04	**2.38E-01**
F23	Best	1.06E+10	3.46E+09	1.52E+11	1.51E+10	3.18E+10	6.54E+10	1.34E+10	1.28E+00	3.24E+09	7.26E+10	**2.23E-05**
	worst	1.07E+10	3.88E+09	2.07E+11	2.27E+10	3.19E+10	2.41E+11	3.12E+13	2.93E+12	2.94E+11	2.75E+11	**5.64E-02**
	mean	1.07E+10	3.60E+09	1.80E+11	1.65E+10	3.18E+10	1.66E+11	2.41E+13	1.50E+11	2.06E+11	2.45E+11	**1.72E-02**
	Std	3.02E+07	7.95E+07	1.18E+10	9.80E+08	1.74E+07	4.96E+10	3.03E+12	4.10E+11	8.42E+10	1.52E+10	**2.44E-02**

Based on the results of the test on functions (1)-(23) with 1000 dimensions, which are presented in ***Table (3)***, and with regard to the *best*, *worst*, *average*, and *standard deviation* criteria, the proposed algorithm was recognized as a powerful algorithm in case of 1000 dimensions as well and could maintain its efficiency over the whole process. In this test, the proposed algorithm could prove its superiority over others in 19 functions, out of 23, in terms of all statistical criteria, which implies its capability of solving the high-dimensional optimization problems. However, it showed a slightly worse performance than the PSO and WOA algorithms in only 4 functions. On this basis, it can be said that the proposed algorithm could yield higher degrees of improvement in all the solutions existing in its population compared to other algorithms. Of course, it showed greater tendency toward the best existing algorithm in each execution due to the use of multi-agent systems and application of other algorithms. The investigations indicated that the proposed algorithm could desirably maintain the relationships between the agents and displace the solutions among them. The main objective of all of these investigations was to evaluate the proposed algorithm in terms of the relationships between the agents (inter-agent relationships) and to know how an algorithm with multiple metaheuristic agents can have a successful performance in a high-dimensional case. Based on the tests conducted in this section, the approach with multi-agent systems can be considered as an important method for taking the advantage of the strongpoints of multiple metaheuristic algorithms as a multi-agent algorithm. In the following sections of the present paper, the binary version of the propose algorithm will be applied for detection of spam emails and the obtained results will be presented.

5-2 Email Spam Detection

doi:10.20944/preprints202001.0317.v1

This subsection is aimed to evaluate the binary BMAMH approach in regard of spam email detection. In ***Subsection (5-1)***, the MAMH approach was executed on 32 benchmark optimizer functions, the results of which indicated considerable superiority of the proposed approach over other metaheuristic algorithms in terms of precision and convergence. For the purpose of this test, a valid spam email dataset was used, which will be described in ***Subsection (5-2-1)***. Using the stratified sampling method, all data were divided into two datasets, namely *test* and *training* datasets. It was attempted to include both spam and non-spam emails in both datasets in order to have a better evaluation of the performance and efficiency of the proposed algorithm and the other comparative algorithms. Moreover, to conduct the tests in this section, the spam email dataset was divided into two sections, including *training (70%)* and *test (30%)*.

5-2-1 Dataset and Objective Function

This subsection provides further explanation on the ***Spam base*** dataset [37]. *Spam base* is a dataset containing 4661 email messages, 1113 (39.4%) of which are spam and 2788 (60.6%) are non-spam emails. Each record in this dataset has 58 features, the last one of which indicates the spam email (1) and non-spam email (0). The first 48 features imply the frequency of some certain keywords, referring to the percentage of the word(s) in an email that corresponds to a particular word. The other 6 features represent the frequency of characters, which indicates the percentage of those characters in the email that correspond to a particular character. The last 3 features contain information on the dataset. This dataset has been introduced as one of the credible and appropriate datasets for spam emails in [37, 38]. Here we are supposed to define the objective function. Defining an objective function for a feature selection problem requires a classification algorithm. Regarding the fact that most of the researchers prefer to use the simplest classification method for feature selection, which is known as K-nearest neighbors ***(KNN) classifier*** [41-48], we use the same method to define the objective function of the feature selection problem in the present paper. Here, ***Equation (14)*** is used as a multi-objective function for feature selection:

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|} \tag{14}$$

Where, $\alpha\gamma_R(D)$ is the classification error rate of a classifier, $|R|$ indicates multi-linearity of the selected subset, $|N|$ is he total number of the features existing in the dataset, and the two parameters α and β indicate the importance of the classification quality and the subset's length, respectively. The values of these two parameters were considered as $\alpha\epsilon[0,1]$ and $\beta=(1-\alpha)$, which have been adopted from the paper in [45]. In fact, each solution is evaluated based on the proposed multi-objective function, which depends on the KNN classifier. In the proposed multi-objective function, in order to balance between the number of selected features in each solution (minimum) and precision of classification (maximum), the fitness function in ***Equation (14)*** is used to evaluate each of the metaheuristic algorithms.

5-2-1 Results of Evaluation of the Binary Approach (BMAMH)

This subsection provides the results obtained from executing the binary approach BMAMH and other comparative algorithms on the spam email dataset, followed by a discussion on these results. In this work, all algorithms were assumed with an equal number of initial population, i.e. 10, but different numbers of iterations. Other parameters of each of these algorithms are provided in ***Table (2)***. Subsequently, the binary approach BMAMH as well as the other comparative algorithms was executed on the *Spam base* dataset with different numbers of iteration. The results of this test are presented in ***Figure (10)*** and ***Table (7)*** with regard to the objective function's value.

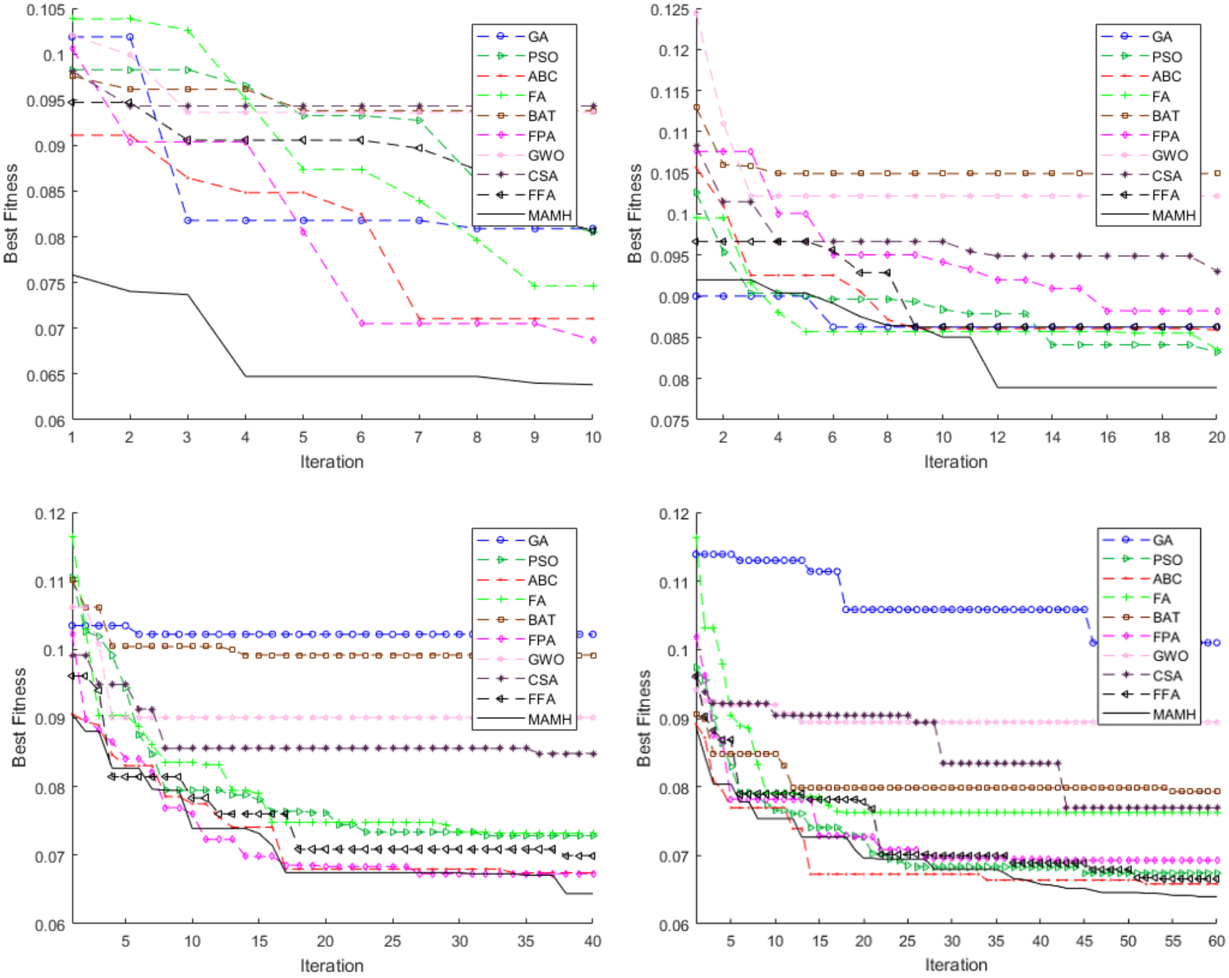


Fig. 10. Comparison proposed algorithm with other algorithms on spam email dataset (Best answer)

Figure (10) shows the results obtained from executing the binary approach BMAMH associated with other comparative algorithms on the spam email dataset at different numbers of iterations, including 20, 40, 60, and 10. The results indicated the better performance of the BMAMH approach compared to other algorithms. At all of the different iterations, the binary approach has accomplished exhibiting a better performance than the other algorithms, indicating its capability and strength for solving a feature selection problem of binary type. Besides, with increase in the number of iterations, the comparative algorithms lost their efficiency while the binary approach, due to the use of multiple powerful agents, could maintain its efficiency over the whole process despite the increased number of iterations. In order to show the binary approach's efficiency, the results of this test are represented in ***Table (7)*** as the statistical criteria of the best solution and the average of the existing solutions in the population of each algorithm.

Table 7: Comparison of BMAMH approach with other comparative algorithms on Email spam dataset with statistical criteria.

Algorithm	**Iteration 10**		**Iteration 20**		**Iteration 40**		**Iteration 60**	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean
GA	9.86E-02	1.02E-01	8.63E-02	8.63E-02	1.02E-01	1.02E-01	1.01E-01	1.01E-01
PSO	8.16E-02	6.65E-02	8.32E-02	6.51E-02	7.28E-02	7.84E-02	6.75E-02	6.83E-02

ABC	8.45E-02	9.88E-02	8.59E-02	9.61E-02	6.74E-02	7.60E-02	6.58E-02	7.28E-02
FA	8.75E-02	7.94E-02	8.36E-02	6.53E-02	7.32E-02	8.40E-02	7.62E-02	7.66E-02
BAT	1.02E-01	7.94E-02	1.05E-01	6.53E-02	9.92E-02	5.60E-02	7.93E-02	5.11E-02
FPA	9.20E-02	1.33E-01	1.01E-01	1.22E-01	8.39E-02	1.01E-01	8.41E-02	9.83E-02
GWO	8.38E-02	8.45E-02	8.82E-02	9.02E-02	6.73E-02	6.86E-02	6.93E-02	6.94E-02
WOA	8.38E-02	8.45E-02	8.82E-02	9.02E-02	6.73E-02	6.86E-02	6.93E-02	6.94E-02
CSA	9.27E-02	1.65E-01	9.31E-02	1.59E-01	8.48E-02	1.63E-01	7.69E-02	1.67E-01
FFA	9.27E-02	1.57E-01	9.31E-02	1.86E-01	8.48E-02	1.62E-01	7.69E-02	1.69E-01
MAMH	8.03E-02	1.82E-01	7.89E-02	8.85E-02	6.44E-02	7.29E-02	6.40E-02	7.20E-02

Results of the first test in this subsection, as shown in ***Figure (10)*** and ***Table (7)***, indicate that, with increase in the number of iterations, the binary approach has accomplished exhibiting a very much better spam email detection performance than the other comparative algorithms in terms of the statistical criteria such as the best and average values as well as the objective function's value. The binary approach BMAMH, at the first iterations, acted just like the other powerful algorithms but, as the number of iterations was increased, it could demonstrate its superiority over other comparative algorithms due to the use of different and powerful multi-agent systems. Subsequently, in order for further testing of the binary approach BMAMH, it was evaluated in terms of precision of classification, the results of which are shown in ***Figure (10)***.

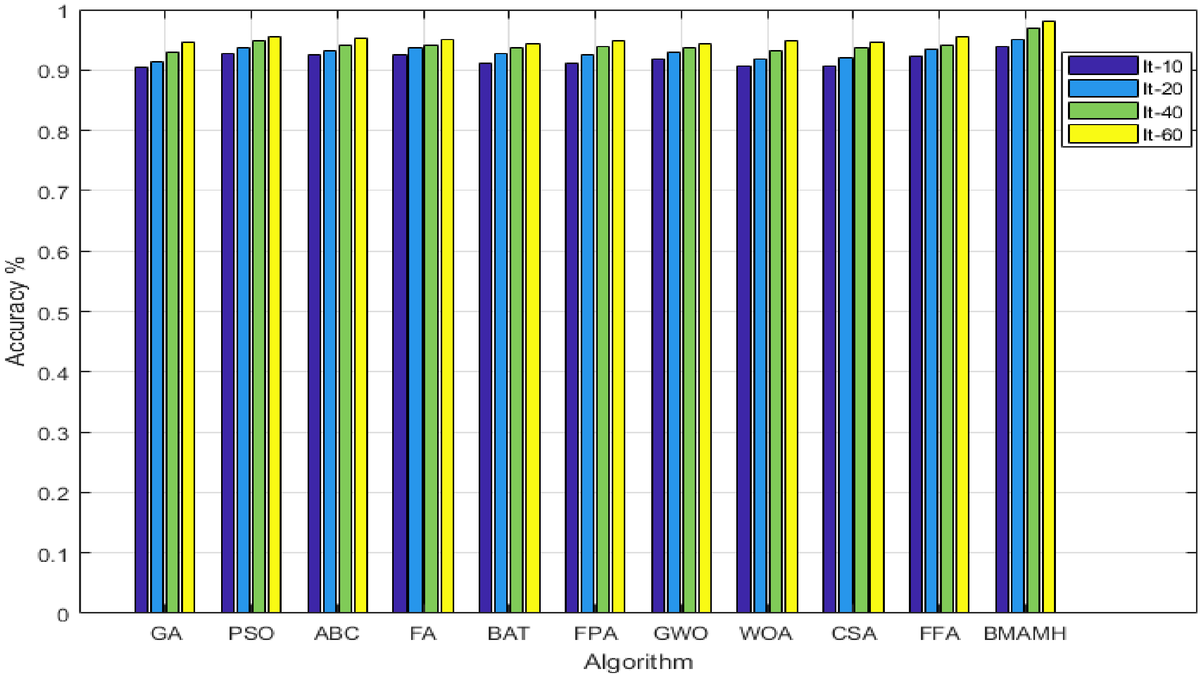


Fig. 11. Comparison of BMAMH approach with other comparative algorithms on Email Spam dataset (Classification accuracy)

Figure (11) shows the results of executing the binary approach BMAMH associated with other comparative algorithms on the spam email dataset at different numbers of iterations, including 10, 20, 40, and 60. These results indicate the higher precision of the binary approach's performance in comparison with other comparative algorithms. The binary approach BMAMH could show a better performance than the other algorithms at all of the above-mentioned iterations, indicating the powerfulness and capability of this algorithm in feature selection as well as the precision of the classifier algorithms. Moreover, with increase in the number of iterations of the algorithms, the precision of the binary approach was increased while maintaining its efficiency. In fact, the binary approach BMAMH succeeded maintaining its efficiency with increase in the number of iterations because of using multiple powerful agents.

6. Conclusion and Future Works

The metaheuristic algorithms have been growingly used for solving the optimization problems in recent decades. Besides, the complex high-dimensional optimization problems are continuously developed. However, when encountering complex and high-dimensional problems, these algorithms reveal their weaknesses such as lower population diversity and lower convergence rate. Therefore, most of the researchers and scholars are attempting to improve the metaheuristic methods in order to provide a better way for solving such problems. Each of these metaheuristic algorithms may have some strong points and some weak points. In the present paper, in order for the algorithms to be efficient, each of them was considered as an agent in a multi-agent system. As the first step, the new approach, named ***MAMH*** that was based on a multi-agent system and the concept of agent was introduced. In the proposed approach, several basic and powerful algorithms including ***GA, PSO, ABC, FA, BA, FPA, GWO, WOA, CSA,*** and ***FFA*** were considered as separate agents, each of which sought to achieve its own goals while competing and cooperating with others for common goals. Afterwards, the binary model of the proposed approach was presented for solving the binary-state problems.

Once the proposed MAMH approach and the binary BMAMH approach were described in details, the two approaches underwent various tests in order for evaluation. First, the proposed MAMH approach was executed on 32 benchmark functions adopted from the CEC'2010 and CEC'2017 benchmark functions with dimensions of 100, 200, 500, and 1000. According to the obtained results, the proposed approach was shown to be effective and powerful method in solving the high-dimensional optimization problems. Moreover, the proposed binary BMAMH approach was executed on the spam email dataset with 57 features. Also, a multi-objective function was used for feature selection. The obtained results indicated the higher precision of performance of the proposed binary BMAMH approach, compared to other methods, in terms of detection of spam emails and the objective function of feature selection. In future, we will attempt to use the proposed MAMH approach in various applications such as image processing, parameter optimization, datamining, and multi-objective optimization and use the proposed binary BMAMH approach for the purpose of feature selection on other datasets as well as binary optimization problems.

Acknowledgment

We appreciate the support of this research from the computer company of Apameh system and Vice-Chancellor for Research of Urmia Branch, Islamic Azad University, and we also thanks to Dr. Bagherzadeh, Dr Masdari, and Dr. Majidzadeh, whos have been working hard to accomplish this research.

7. References

1. Yazdani, M. and F. Jolai, *Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm.* Journal of computational design and engineering, 2016. **3**(1): p. 24-36.
2. Long, W., et al., *Solving high-dimensional global optimization problems using an improved sine cosine algorithm.* Expert Systems with Applications, 2019. **123**: p. 108-126.
3. Li, Z., et al., *PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems.* Expert Systems with Applications, 2015. **42**(22): p. 8881-8895.
4. Shayanfar, H. and F.S. Gharehchopogh, *Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems.* Applied Soft Computing, 2018. **71**: p. 728-746.
5. Meng, X.-B., et al., *A new bio-inspired optimisation algorithm: Bird Swarm Algorithm.* Journal of Experimental & Theoretical Artificial Intelligence, 2016. **28**(4): p. 673-687.
6. Sivanandam, S. and S. Deepa, *Genetic algorithm optimization problems*, in *Introduction to Genetic Algorithms*. 2008, Springer. p. 165-209.
7. Achtnig, J., *Particle swarm optimization with mutation for high dimensional problems*, in *Engineering Evolutionary Intelligent Systems*. 2008, Springer. p. 423-439.
8. Karaboga, D. and B. Akay, *A comparative study of artificial bee colony algorithm.* Applied mathematics and computation, 2009. **214**(1): p. 108-132.
9. Yang, X.-S., *Firefly algorithm.* Nature-inspired metaheuristic algorithms, 2008. **20**: p. 79-90.
10. Yang, X.-S., *A new metaheuristic bat-inspired algorithm*, in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. 2010, Springer. p. 65-74.
11. Mirjalili, S., S.M. Mirjalili, and A. Lewis, *Grey wolf optimizer.* Advances in engineering software, 2014. **69**: p. 46-61.
12. Mirjalili, S. and A. Lewis, *The whale optimization algorithm.* Advances in Engineering Software, 2016. **95**: p. 51-67.
13. Pisarski, S., et al. *Evolutionary multi-agent system in hard benchmark continuous optimisation*. in *European Conference on the Applications of Evolutionary Computation*. 2013. Springer.
14. Golmohamadi, H., et al., *A multi-agent based optimization of residential and industrial demand response aggregators.* International Journal of Electrical Power & Energy Systems, 2019. **107**: p. 472-485.
15. Chu, Y., et al. *A fast bacterial swarming algorithm for high-dimensional function optimization*. in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. 2008. IEEE.
16. Zhao, S.-Z., P.N. Suganthan, and S. Das, *Self-adaptive differential evolution with multi-trajectory search for large-scale optimization.* Soft Computing, 2011. **15**(11): p. 2175-2185.
17. Chu, W., X. Gao, and S. Sorooshian, *A new evolutionary search strategy for global optimization of high-dimensional problems.* Information Sciences, 2011. **181**(22): p. 4909-4927.
18. Wang, H., S. Rahnamayan, and Z. Wu, *Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems.* Journal of Parallel and Distributed Computing, 2013. **73**(1): p. 62-73.
19. Yu, H. and X. Li. *High dimensional problem based on elite-grouped adaptive particle swarm optimization*. in *International Conference on Intelligent Computing*. 2013. Springer.
20. Imanian, N., M.E. Shiri, and P. Moradi, *Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems.* Engineering Applications of Artificial Intelligence, 2014. **36**: p. 148-163.
21. Liu, F. and Z. Zhou, *An improved QPSO algorithm and its application in the high-dimensional complex problems.* Chemometrics and Intelligent Laboratory Systems, 2014. **132**: p. 82-90.

22. Tuo, S., et al., *A harmony search algorithm for high-dimensional multimodal optimization problems.* Digital Signal Processing, 2015. **46**: p. 151-163.
23. Mohapatra, P., K.N. Das, and S. Roy, *A modified competitive swarm optimizer for large scale optimization problems.* Applied Soft Computing, 2017. **59**: p. 340-362.
24. Long, W., et al., *An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization.* Engineering Applications of Artificial Intelligence, 2018. **68**: p. 63-80.
25. Balaji, P. and D. Srinivasan, *An introduction to multi-agent systems*, in *Innovations in multi-agent systems and applications-1*. 2010, Springer. p. 1-27.
26. Wooldridge, M., *An introduction to multiagent systems*. 2009: John Wiley & Sons.
27. Zheng, X.-l. and L. Wang, *A multi-agent optimization algorithm for resource constrained project scheduling problem.* Expert Systems with Applications, 2015. **42**(15-16): p. 6039-6049.
28. Passos, C.A., V.M. Iha, and R.B. Dominiquini, *A multi-agents approach to solve job shop scheduling problems using meta-heuristics.* IFAC Proceedings Volumes, 2010. **43**(17): p. 198-203.
29. Liu, J., H. Jing, and Y.Y. Tang, *Multi-agent oriented constraint satisfaction.* Artificial Intelligence, 2002. **136**(1): p. 101-144.
30. Schmitt, L.M., *Theory of genetic algorithms.* Theoretical Computer Science, 2001. **259**(1-2): p. 1-61.
31. Eberhart, R. and J. Kennedy. *A new optimizer using particle swarm theory*. in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. 1995. Ieee.
32. Karaboga, D. and B. Basturk. *Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems*. in *International fuzzy systems association world congress*. 2007. Springer.
33. Yang, X.-S. *Flower pollination algorithm for global optimization*. in *International conference on unconventional computing and natural computation*. 2012. Springer.
34. Askarzadeh, A., *A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm.* Computers & Structures, 2016. **169**: p. 1-12.
35. Hussien, A.G., et al., *S-shaped binary whale optimization algorithm for feature selection*, in *Recent trends in signal and image processing*. 2019, Springer. p. 79-87.
36. Mafarja, M., et al. *S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem*. in *Proceedings of the international conference on future networks and distributed systems*. 2017. ACM.
37. Mark Hopkins, E.R., George Forman, Jaap Suermondt *UCI Machine Learning Repository: Spambase Data Set*, in *Hewlett-Packard Labs*. 1999.
38. Koprinska, I., et al., *Learning to classify e-mail.* Information Sciences, 2007. **177**(10): p. 2167-2187.
39. Mafarja, M.M. and S. Mirjalili, *Hybrid Whale Optimization Algorithm with simulated annealing for feature selection.* Neurocomputing, 2017. **260**: p. 302-312.
40. Mafarja, M. and S. Mirjalili, *Whale optimization approaches for wrapper feature selection.* Applied Soft Computing, 2018. **62**: p. 441-453.
41. Mabodi, K., Yusefi, M., Zandiyan, S., Irankhah, L., & Fotohi, R. (2020). Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication. The Journal of Supercomputing, 1-25.
42. Seyedi, B., & Fotohi, R. (2020). NIASHPT: a novel intelligent agent-based strategy using hello packet table (HPT) function for trust Internet of Things. The Journal of Supercomputing, 1-24.
43. Fotohi, R. (2020). Securing of Unmanned Aerial Systems (UAS) against security threats using human immune system. Reliability Engineering & System Safety, 193, 106675.

44. Fotohi, R., Firoozi Bari, S., & Yusefi, M. (2020). Securing Wireless Sensor Networks Against Denial-of-Sleep Attacks Using RSA Cryptography Algorithm and Interlock Protocol. International Journal of Communication Systems, 33(4), e4234.

45. Liao, T. and R. Kuo, *Five Discrete Symbiotic Organisms Search Algorithms for Simultaneous Optimization of Feature Subset and Neighborhood Size of KNN Classification Models.* Applied Soft Computing, 2017.

46. Mafarja, M., et al., *Evolutionary Population Dynamics and Grasshopper Optimization Approaches for Feature Selection Problems.* Knowledge-Based Systems, 2017.

47. Rajamohana, S. and K. Umamaheswari, *Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection.* Computers & Electrical Engineering, 2018.

48. Altman, N.S., *An introduction to kernel and nearest-neighbor nonparametric regression.* The American Statistician, 1992. **46**(3): p. 175-185.

49. Emary, E., H.M. Zawbaa, and A.E. Hassanien, *Binary ant lion approaches for feature selection.* Neurocomputing, 2016. **213**: p. 54-65.