Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 201 (2022) 362–395
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original Articles

Aphid–Ant Mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems

N. Eslami[a], S. Yazdani[a,*], M. Mirzaei[b], E. Hadavandi[a,c]

[a] Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran
[b] Department of Electrical and Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran
[c] Department of Industrial Engineering, Birjand University of Technology, Birjand, Iran

Received 6 December 2021; received in revised form 15 April 2022; accepted 14 May 2022
Available online 24 May 2022

Abstract

Swarm intelligence algorithms, which are developed for solving complex optimization problems designed by focusing on simulating the social behavior of one species of simple animals. However, simple animals utilize cooperation to work together that result in more complex and smarter behaviors. This paper proposes a novel population-based optimization paradigm for solving NP-hard problems called "Aphid–Ant Mutualism (AAM)" which is inspired by a unique relationship between aphids and ants' species. This relationship is called 'mutualism'. Despite the previous studies that the social behaviors of aphids and ants were simulated, AAM models mutual interaction among aphids and ants in nature. Thus, AAM has new features by incorporating heterogeneous individuals consisting of aphids and ants that live in various colonies together and have different decentralized learning behaviors and objectives. Inspired by nature, colony-based information exchange and using different search strategies including focusing on the individual's personal knowledge, learning from other colony's members and information sharing with adjacent colonies are used. This mutualism leads to converging to the global optimum and avoids premature convergence. Performance of AAM is assessed using statistical evaluation, convergence analysis, and a non-parametric Wilcoxon rank-sum test with a 5% significance degree on forty-one benchmarks selected from well-known functions of recent studies and more challenging benchmark functions called CEC 2014, CEC 2017 and also CEC-C06 2019 test suite. Statistical results and comparisons with other meta-heuristic algorithms demonstrate that the AAM algorithm provides promising and competitive outcomes. Furthermore, it can produce more accurate solutions with a faster convergence rate to the global optima.

Keywords: Aphid–Ant Mutualism; Swarm intelligence; Optimization; Nature-inspired metaheuristic; Adaptive search strategy; Population-based algorithm

1. Introduction

Various kinds of metaheuristic algorithms have been developed in recent years. They have successfully solved many real-world optimization problem [20]. Compared with traditional mathematical programming methods, metaheuristics are simple, flexible, derivation-free, and can escape from local optima. Typical metaheuristic

* Corresponding author.
E-mail addresses: nava1eslami@yahoo.com (N. Eslami), s_yazdani@iau-tnb.ac.ir (S. Yazdani), m.mirzaie@iau-tnb.ac.ir (M. Mirzaei), es.hadavandi@birjandut.ac.ir (E. Hadavandi).

https://doi.org/10.1016/j.matcom.2022.05.015
0378-4754/

algorithms are divided into two main categories that are evolutionary algorithms and swarm intelligence. These algorithms are developed by inspiring simple real-world phenomena such as biological evolution or animal social behaviors [13,20].

Many algorithms have been introduced to improve the performance of standard metaheuristic algorithms. However, referring to the No-Free-Lunch (NFL) theorem, all optimization algorithms introduced so far have an equal performance on average tackling with all possible optimization problems. Moreover, metaheuristics that utilize decentralized learning behaviors, heterogeneous individuals, and multi-search strategies achieve more success compared to other optimization algorithms on different kinds of optimization problems [19,20,31].

These ideas match with nature in which individuals with a variety of behaviors have more chance to survive. In nature, simple individuals can obtain complex behaviors to overcome inappropriate environmental situations by cooperation [25,45]. For instance, aphids and ants have lived together for several years and, their interaction called 'mutualism' leads to survival and individual growth rate in their populations. These numerous species have unique features that result in effective interaction with the surrounding environment and mutualism's development. Both aphids and ants have colony lives. In aphids' colonies, the features such as group life, the winged offspring named 'Alettes', sexual and asexual recombination are common. Aphids have three life spans and, in each of them, they utilize one of their sexual or asexual reproductive abilities in response to environmental situations or frequency of food resources [25,44].

Like aphids, ants are one of the most successful zoological feminine species. In ants' colony, three types of ants, namely queen, female, and male workers, live together [44]. Worker ants must traverse long distances to prepare the colony's required food, and as a result, need lots of energy that glucose in aphids' excrement is a good source. Aphids prepare food for ants, and, in return, ants protect them and their eggs against their enemies like ladybirds. Moreover, ants monopolize greater aphids that can produce more honeydew by pheromone or special odor and help aphids' displacement toward more appropriate food sources. Research shows that this mutual interaction has a proper influence on aphids' and ants' survival and growth rate and achieving their different goals that are not correlated [40,44].

It is evident that no one metaheuristic algorithm can theoretically consider as a general-purpose universally — best optimizer to solve all kinds of problem domains. Therefore, novel algorithms that can address many types of problems are required to provide more choices for researchers, and experts in different fields [19]. To introduce a more adaptable and efficient optimization algorithm, we focus on the mutualism of aphids and ants in nature. Although, before this article, some research has proposed algorithms based on the behavior of aphids and ants separately [12,13,20], to our best knowledge, no research has considered the mutualistic relationship of these two different species so far. In the current research, we propose a novel nature-inspired optimization algorithm referred to as "Aphid – Ant Mutualism (AAM)" to compete with other optimizers.

AMM starts by generating a random population of individuals. Then, the population is divided randomly into two types, which are aphids and ants. As happens in nature, aphids and ants generate colonies to benefit the mutualism. For this purpose, the aphids' population is evaluated using the fitness function and the predefined number of the best aphids are selected to form the colonies which are called ringleaders. Remaining aphids of the sorted population are called weak aphids. The power of the ringleaders determines the number of weak aphids and ants attracted to each colony. In each colony, aphids and ants utilize different searching strategies by concentrating on their own knowledge, observing, and learning from others and information sharing with adjacent colonies to update their positions. This decentralized interaction, colony-based information sharing, and diverse search strategies may result in the overall improvement of each agent's behavior and ultimately the entire population to make the AAM efficient for solving different optimization problems. The main contributions of this paper are summarized as follows:

- Introducing a novel population-based algorithm which is called AMM by inspiring the mutualism of aphids and ants in nature.
- AMM contemplates two distinct categories of individuals that are aphids and ants. Best aphids which are called ringleaders forms colonies consist of remaining aphids and ants.
- AMM incorporates different search strategies, which simulate the different collaborative behaviors that are presented in the natural life of aphids and ants. Thus, AMM utilizes heterogeneous individuals that enable support each other and can follow different search trajectories. These properties make AMM suitable for solving different and complex optimization problems.

- Inspired by nature, aphids with different fitness values can play different roles and their positions are updated based on three various strategies that are correlated to their qualities. Best aphids utilize asexual updating operator to evolve using their own experience. Furthermore, some weak aphids of the population are selected randomly by ants to change their position in the colony and explore new areas. Based on the life spans of the aphids in nature, they can also change their positions using sexual updating operator to use another aphid's experience. Based on ants' strategy in interaction with aphids, ants endeavor to move toward best aphids of the colony to gain more food. Moreover, to enhance the population diversity, some winged aphids which are called 'Alettes', try to fly using the wind blow to other colonies and spiral movement operator is designed to model this strategy.

We have compared the proposed AAM with ten other well-established optimization techniques that have shown promising results on many optimization problems in recent years on a set of 41 benchmark functions from commonly used functions, CEC 2014, CEC 2017 and CEC-C06 2019 test suite [5,12,20] using mean, standard deviation and a non-parametric Wilcoxon rank-sum test with a 5% significance degree and convergence curve analysis [12,40].

The rest of this paper is organized as follows. In Section 2, background works that are the source of inspiration for this idea are reviewed. Section 3 represents the fundamentals of the mutualistic behavior of aphids and ants. In Section 4, the novel optimization algorithm and its characteristics are described. Section 5 analyzes the experimental results extensively. Eventually, the conclusions and future research directions are given in Section 6.

2. Literature review

Population-based algorithms are successful approaches to solve complex optimization problems. However, selecting an appropriate algorithm from a large number of ones is challenging and time-consuming. Designing a versatile algorithm that can solve particular problems with different characteristics is a vital concern of the researchers. Algorithms that utilize diverse search approaches, can change their behavior during the search process, and follow different trajectories by incorporating multiple search operators and heterogeneous individuals have more chances to solve different optimization problems. However, most population-based algorithms utilize homogeneous individuals that behave similarly to each other during the optimization process. We investigate some of them as follows [12,40,46].

The most common and well-known algorithm in the population-based algorithms field is particle swarm optimization (PSO) which mimics flying birds [36]. In this method, there exist some particles, which have positions and velocities. The target is finding the best particle that presents the best solution in the search space [36]. However, the implementation of PSO is simple due to its few tuning parameters. Well-tuning the design parameters is a challenging task and may result in trapping in the local optima and premature convergence. Another example is differential evolution (DE) [10]. In this algorithm, the position of each individual is updated according to three different other individuals in the population. Although the DE is easy to implement and has few user-defined parameters, it is likely skipping promising areas [10].

As another instance, Dorigo et al. suggested ant colony optimization (ACO) [12], inspired by the social behavior of ants, to find the shortest path between the food sources and their nest by releasing the pheromone. The ACO demonstrated promising results in network routing tasks in networks with uncertain structures [12]. Moreover, Mirjalili proposed the ant lion optimizer (ALO) [27] that mimics the hunting strategy of antlions. However, compared to ACO, ALO is more efficient in many benchmarks and engineering design problems due to developing a perfect balance between exploration, and exploitation strategies. The slow speed of convergence to the global optimum is noticeable [27].

In another study, Saremi et al. proposed the grasshopper optimization algorithm (GOA) [37] inspired by the grasshopper swarms in nature. In this algorithm, the position of each grasshopper is adjusted according to the social interaction, gravity force, and wind advection. The original version of GOA could only solve single-objective problems with continuous nature [37]. Despite not utilizing different individuals with various searching strategies, GOA makes an acceptable balance between exploration and exploitation. Furthermore, Mirjalili et al. introduced the salp swarm algorithm (SSA) [28] that mimics navigating and foraging behaviors of salps in the oceans. This algorithm is easy to implement and flexible. Nevertheless, the lack of utilizing various searching behaviors is visible [28].

Heidari et al. proposed Harris hawks optimization (HHO) [19] based on the hunting strategy of Harris hawks called surprise pounce. This algorithm is easy to implement due to few exploration and exploitation mechanisms [19]. At the same time, there is a possibility of trapping in the local optima or premature convergence due to poor information sharing between different individuals with various searching mechanisms or evolutionary updating strategies.

Li et al. introduced slime mould algorithm (SMA) [22] inspired by oscillation mode of slime mould in nature. The proposed algorithm uses adaptive weights to form the optimal path for joining food. The proposed algorithm was compared with up-to-date metaheuristics on thirty-three benchmarks. It showed an acceptable efficiency on benchmark functions [22]. Moreover, Faramarzi et al. proposed equilibrium optimizer (EO) [14] based on control volume mass balance models used to estimate both dynamic and equilibrium states. Authors claimed that EO achieved acceptable results compared to well-known optimization algorithms in many benchmarks [14]. Although, it does not use information sharing between individuals with various searching strategies.

Braik et al. introduced a novel metaheuristic search algorithm called capuchin search algorithm (CapSA) [6] inspired by the dynamic behavior of capuchin monkeys during wandering and foraging over trees and riverbanks in forests while searching for food sources. Although, the results showed that CapSA achieves solutions that are more precise compared to competitive optimization methods [6]. CapSA has many tuning parameters that should render appropriately to gain high quality results. Furthermore, Nadimi-Shahraki et al. proposed an improved grey wolf optimizer (IGWO) [30] for solving engineering problems. This improvement is proposed to alleviate the lack of the diversity, the imbalance between the exploitation and exploration, and premature convergence of the grey wolf optimizer (GWO) algorithm. However, compared to GWO, IGWO is more efficient in many benchmarks and engineering design problems due to developing a perfect balance between exploration, and exploitation strategies and increment of diversity [30]. The lack of heterogeneous individuals with various searching strategies or evolutionary mechanisms is noticeable.

Hashim et al. introduced a novel nature-inspired optimization algorithm called honey badger algorithm (HBA) [18] based on the intelligent foraging behavior of honey badger. The dynamic behavior of honey badger with digging and honey finding approaches is utilized to develop exploration and exploitation phases of HBA. The efficiency of HBA was compared with ten well-known optimization algorithms to show its superiority in terms of convergence speed and exploration–exploitation balance [18].

Trojovsky et al. proposed pelican optimization algorithm (POA) [43] for engineering applications. POA is inspired by natural behavior of pelicans during hunting. Although, POA shows an acceptable balance between exploration and exploitation abilities [43]. The slow speed of convergence is visible. Moreover, Abdollahzadeh et al. proposed artificial gorilla troops optimizer (GTO) [1] algorithm based on gorillas' cooperative life. The results demonstrated GTO superiority particularly on high-dimensional problems [1]. However, lack of changing search strategies during optimization process is obvious.

Kou et al. introduced a novel metaheuristic called fast jaguar algorithm (FJA) [21]. FJA inherits jaguar algorithm (JA) [7] advantages and significantly improves its search ability. The fast-hunting strategy is incorporated in FJA to identify the local area effectively. However, FJA outperforms JA in many benchmarks, it needs historical information to effectively find better tendency [21].

Almost all the reviewed algorithms have centralized individuals and are inspired by the behavior of a single species in nature. However, in the real world; the decentralization of different species in various habitats is familiar, which leads to introducing behaviors that are more complex by communicating with each other to survive for several years. Focusing on these species can lead to introducing more general optimization algorithms. Because instead of using the population with the same searching behaviors, different individuals with various simple behaviors are utilized to observe different aspects of the search space and find the global optimum [16].

For example, Niu et al. [32] proposed a new version of PSO based on the master and slave relationship between species in nature. In this algorithm, one master swarm and a fixed number of slave swarms exist [32]. However, using different swarms decreases the chance of trapping in local optima by increasing the amount of decentralization. The utilization of a fixed number of subpopulations may result in premature convergence. Furthermore, information sharing exists inside the same subpopulations with similar behaviors.

Liang and Suganthan [24] introduced a multi-swarm version of PSO concerning the concept of local searching. In this method, the initial population is divided into many sub-swarms. The number of subswarms can differ according to the specific regrouping schedule. The outcomes showed that this algorithm has superior performance in many cases compared to the original PSO [24].

Table 1
Summarized comparison of the reviewed metaheuristics' some features.

Algorithm Name	Fast convergence	Appropriate balance of exploration and exploitation	Cooperation between various individuals	Low computational time	Trapping in the local optima
PSO	×	×	×	×	+
DE	+	×	×	+	×
ACO	×	×	×	×	+
ALO	×	+	×	×	+
GOA	×	+	×	×	+
SSA	+	+	×	+	×
HHO	+	+	×	+	+
SMA	+	+	×	+	×
EO	×	+	×	+	×
CapSA	+	×	×	+	+
IGWO	+	+	×	+	×
HBA	+	+	×	+	×
POA	×	+	×	+	×
GTO	+	+	×	+	×
FJA	+	+	×	×	×
Master-Slave PSO	×	+	×	×	×
Multi-Swarm PSO	+	+	×	+	+
Donkey and smuggler optimization algorithm	×	+	+	×	×
social spider optimization algorithm	+	+	×	+	×
Proposed AAM	+	+	+	+	×

However, by looking carefully at nature, we can find different types of species that interact to increase their survival chances. This phenomenon is the source of inspiration to develop bio-inspired algorithms with heterogeneous individuals. As an example of utilizing an omniscient group to provide the initial information, Shamsaldin et al. proposed the donkey and smuggler optimization algorithm for pathfinding [38]. The searching behavior of donkeys is the inspiration source of this algorithm. There are two different methods to implement search processes and path selection. In smuggler mode (omniscient mode), which is a passive stage, all possible routes are analyzed, and finally, the shortest path is picked. This model is utilized to prepare the required information for the next state. In the donkey state, which is done after smuggler mode, the optimal solution is selected based on different behaviors of donkeys like running, face, and suicide and face and support. It is worth mentioning that after finding the shortest path, the interaction of smugglers and donkeys as different species is disconnected, and this information sharing does not last until the end of the optimization process [38].

As an instance for concentrating on different behaviors inside a single species, Cuevas et al. introduced a swarm optimization algorithm inspired by the behavior of the social spider [9]. There are two different kinds of spiders, namely male and female, demonstrating distinct search methods and abilities. Nevertheless, there still exist the same species with various exploratory or exploiting behaviors. These individuals could simulate different behaviors in the colony. The authors claimed that the performance of their method is promising in many benchmark functions [9].

From All the above, it is evident that no one metaheuristic algorithm can theoretically consider as a general-purpose optimizer to solve all kinds of problem domains. Therefore, novel algorithms that can address many types of problems are required to provide more choices for researchers, and experts in different fields [19,39]. In this paper, to introduce a more adaptable and efficient optimization algorithm, we focus on the mutualism of aphids and ants in nature. Subsequently, we introduced a novel nature-inspired optimization algorithm named "Aphid–Ant Mutualism (AAM)" to compete with other optimizers. A brief comparison of the proposed AAM and, other reviewed metaheuristics is summarized in Table 1. As it can be seen in Table 1, + denotes the algorithm has the described feature, while × demonstrates the feature in not included in the metaheuristic [8,26,35].

3. Inspiration source

In this section, we introduce the fundamental concepts, and inspiration source of the AAM algorithm. Then in the next part, a mathematical model based on this unique social behavior will be established.

Aphids and ants have lived together for several years in the same habitats and have the unique interactions with the surrounding environment and each other [39]. Both species live in the colonies and benefit from the group -life. Aphids are one of the most successful social insects of the 'Aphididae' family. They have three life spans, and in each of them, they utilize sexual or asexual reproductive abilities in response to environmental situations or the frequency of food resources. Furthermore, they can produce winged offspring called 'Alettes' to increase the chance of the colony's survival [33].

Like aphids, ants are one of the most successful zoological feminine species, and three types of ants, namely queen, female, and male workers, habit in a colony. Worker ants traverse long distances to prepare the colony's required food and, as a result, need lots of energy that glucose in aphids' excrement is a good source [17]. Aphids prepare food for ants. As a reward, ants protect them and their eggs against their enemies like ladybirds. Furthermore, ants monopolize greater aphids that can produce more excrement by pheromone or special odor and help aphids' displacement toward more appropriate food sources. Aphids interact with ants to be protected from their enemies and survive in poor environmental conditions and, ants need this mutualism to achieve sufficient food. Studies demonstrate that this mutual relationship has a perfect influence on aphids' and ants' survival, growth rate and achieving their uncorrelated goals [15,33,39]. Based on this unique mutualism, we try to design a new mathematical model. Afterward, a stochastic metaheuristic is introduced based on the proposed mathematical model to solve different optimization problems.

4. Method

In this section, the mathematical model of the proposed algorithm by inspiring the aforementioned bilateral relationship between aphids and ants are described. The main steps of AMM are generating an initial population, the colonies' creation, mutualism, ants' evolution, and aphids' flight. Like other evolutionary ones, AMM starts its search with randomly generated solutions which are divided into aphids and ants (candidate solutions and their counterparts). The population of ants is less than the population of aphids in the AAM because ants need the excrement of many aphids to gain the required energy for long displacements in nature. Section 4.1 describes the initialization of AMM's population.

Then, four various steps perform on the population to advance the search process and find the global optimum. In the first step, which is colonies' generation, the best aphids in the aphids' population compete to choose the plants with more vegetable saps (Section 4.2). Stronger aphids have more chances to obtain bigger plants. Therefore, more weak aphids attract to the colony and a bigger colony is created. In addition, bigger aphids' colonies can attract more ants and, therefore, more robust mutualism is formed. In the second step, referred to as the mutualism step (Section 4.3), aphids of each colony interact with aphids and ants of the colony to find and move toward better food sources. In the ant's evolution step (Section 4.4), ants evolve toward one of the best aphids in the colony to gain more excrement. Finally, in the fourth part of the algorithm (flight step (Section 4.5)), the best aphids in each colony that have appropriate weights to fly with the wind blow can move to adjacent colonies [20]. It increases their chance to find better food sources or to be monopolized by stronger ants. AMM is described in detail in the following subsections. Finally, the time complexity of the proposed algorithm is analyzed in Section 4.6.

4.1. Generating the initial population

AMM starts by generating a random population of N_{pop} individuals. Then, the population is divided randomly into two types which are aphids and ants. So, the population includes n aphids and m ants where $n > m$ because ants need the excrement of many aphids to gain the required energy for long displacements in nature. The population of aphids and ants is demonstrated in Eq. (1).

$$\begin{aligned} Aphid_{k,j} &= LB_j + rand \times (UB_j - LB_j), k = 1, 2, \ldots, n \\ Ant_{p,j} &= LB_j + rand \times (UB_j - LB_j), p = 1, 2, \ldots, m \end{aligned} \tag{1}$$

In Eq. (1), LB_j and UB_j show the lower and upper bounds of the jth variable, respectively. Furthermore, $rand$ is a uniform random number in the range of [0,1]. $Aphid_{k,j}$ demonstrates the kth aphid's position in the jth dimension and $Ant_{p,j}$ is the pth ant's position in the jth dimension in the population, respectively.

4.2. The colonies' creation

As happens in nature, aphids and ants generate colonies to benefit the mutualism. For this purpose, the aphids' population is evaluated using the fitness function and sorted based on the ranking value of the individuals. Afterwards, the predefined number of the fittest aphids which are called ringleaders are selected from the population to form the colonies. Remaining aphids of the population are called weak aphids. These weak aphids of the population will scatter among the ringleaders in the colonies' generation process. After colonies' generation, ants will join to the optimization process. It is worth mentioning that to improve the probability of finding the optimal solution for hard optimization problems with complex landscapes, each colony includes aphids and ants that use diverse search approaches. The same search approach may make AAM follow similar trajectories and be trapped in one of the local optima. In the following subsections, attraction of weak aphids' and, ants to the colonies are described in detail.

4.2.1. Weak aphids' attraction to the colonies

The weak individuals of the aphids' population are divided randomly among ringleaders based on ringleaders' attraction powers to form the aphids' colonies [4]. The attraction power of the nth ringleader is calculated using Eq. (2) [4].

$$NA_n = 2 - SP + 2 \times (SP - 1) \times \frac{\frac{\max_i\{Pr_i\} - Pr_n}{\sum Pr_i} - 1}{N_{col} - 1} \tag{2}$$

In Eq. (2), $\max_i\{Pr_i\}$ is the fitness of the best ringleader in the population, and Pr_n demonstrates the fitness of the nth ringleader. Moreover, $\sum Pr_i$ is the sum of all ringleaders' fitness, and we use the selection pressure to neutralize the effect of assigning relative fitness [4]. SP shows the selection pressure which is set to 1.1 that is the best value based on the literature [4,8,17,26], and N_{col} is the total number of weak aphids in the initial population. This ranking leads to uniformed scaling in the population and, efficient control of the selective pressure [29]. The number of weak aphids in the colony of a ringleader is directly proportionate to its relative power and is calculated using Eq. (3).

$$NC_n = round\{NA_n \times N_{col}\} \tag{3}$$

In Eq. (3), NC_n is the number of aphids in the nth colony. Moreover, N_{col} and NA_n are the total number of weak aphids in the initial population and the power of the nth ringleader, respectively. NC_n of the weak individuals of the aphid's population is chosen randomly to form the nth colony.

4.2.2. Ants' attraction to the colonies

In this step, the individuals of ants' population choose the colonies based on the attraction power of each ringleader that is directly correlated to the size of each colony. The number of ants attracted randomly to the nth colony is calculated by Eq. (4) [4].

$$Nant_n = round\{NA_n \times N_{Ants}\} \tag{4}$$

In Eq. (4), NA_n is the attraction power of the nth ringleader and, N_{Ants} is the total number of ants in the initial population. Then, in each colony, the fitness of aphids and ants are calculated using the fitness function. Fig. 1 demonstrates plant selection and creation of the initial colonies, schematically.

4.3. Mutualism

In AMM, different search strategies are designed to update the position of individuals in the population by inspiring the mutualism among aphids and ants in nature. During the evolutionary process of AAM, aphids with different fitness values can play different roles and their positions are updated based on three various strategies which are correlated to their qualities. Superior aphids are evolved using their own experience and perform local search to find better solutions in their neighborhoods. Furthermore, based on what happen in nature, some weak aphids are selected randomly to move using ants to explore new areas in the search space and maintain the population diversity. Finally, the remaining weak aphids that are not evolved so far, are updated either using their own experience or

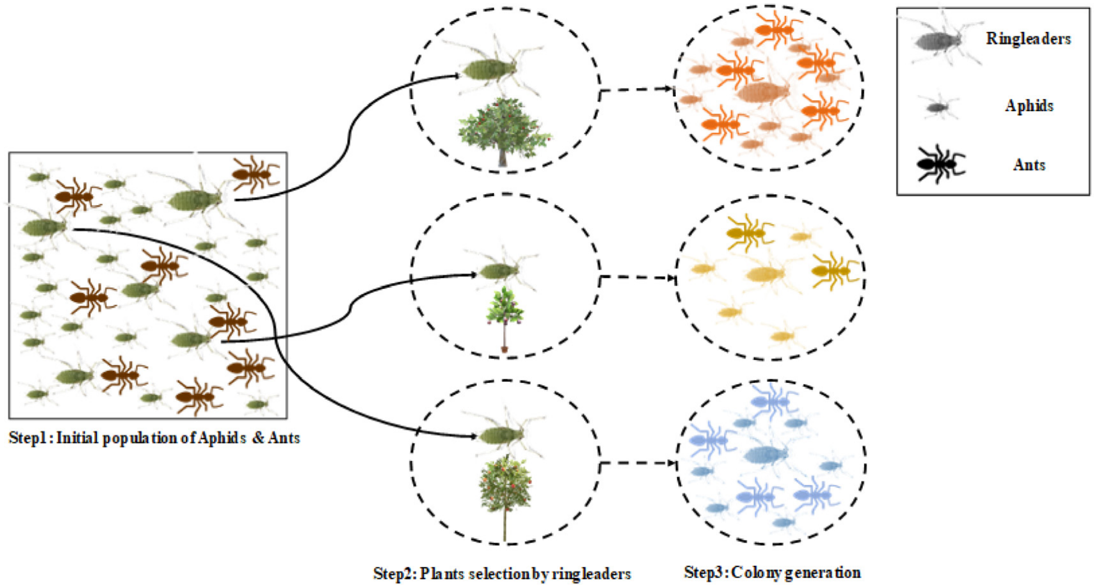


Fig. 1. Schematic presentation of initial colonies' generation.

other aphids to explore new areas around the other colony's members to find more appropriate food sources. These searching strategies will be explained in detail in the following subsections. Afterward, the ants' movement toward the best aphids using the marked paths by pheromone is described in Section 4.4 [12]. Finally, the best aphids' flight to adjacent colonies is explained in Section 4.5 [4,23].

4.3.1. Aphids' evolution based on their own experience (exploitation phase)

The strongest aphids in each colony are evolved using their own experience to exploit promising areas around their positions. A parameter that is called μ is used to control the number of selected aphids and its value is set 0.3 to update 30% of the best aphids in the sorted population using this searching strategy to propagate promising features in the population. Subsequently, it increases the chance of finding the global optima by exploiting promising areas. Eq. (5) demonstrates this searching strategy [20].

$$Aphid_{i,j}(t+1)^n = Aphid_{i,j}(t)^n + \alpha_1 \times \frac{randn}{t} \times (UB_j - LB_j) \tag{5}$$

In Eq. (5), $Aphid_{i,j}(t+1)^n$ shows the position of ith aphid in the jth dimension in the (t + 1)th iteration in the nth colony. Furthermore, $Aphid_{i,j}(t)^n$ demonstrates the position of this aphid in current iteration (t). $randn$ is a random number with normal distribution, and α_1 is a scaling parameter in the range of (0,1]. These parameters develop some random movements around the current position of the aphid. Moreover, UB_j and LB_j are the upper and lower bounds of the jth variable, respectively and, t is the current iteration number. As the optimization process progresses, the search domain around the selected aphid decreases to increase the chance of finding the global optimum.

4.3.2. Aphids' movement using ants (exploration phase)

Some weak aphids in the colonies are selected randomly and moved using randomly selected ants. In nature, ants of each colony change the position of aphids randomly to increase their change to find more vegetable sap and therefore, produce more excrement. The parameter that is called λ is utilized to select aphids for this updating strategy. The ith aphid's position in the jth dimension at (t + *1*)th iteration in the nth colony is calculated using Eq. (6).

$$Aphid_{i,j}(t+1)^n = Ant_{rand}(t)^n + \beta \times rand \times |Aphid_{i,j}(t)^n - Ant_{rand}(t)^n| \tag{6}$$

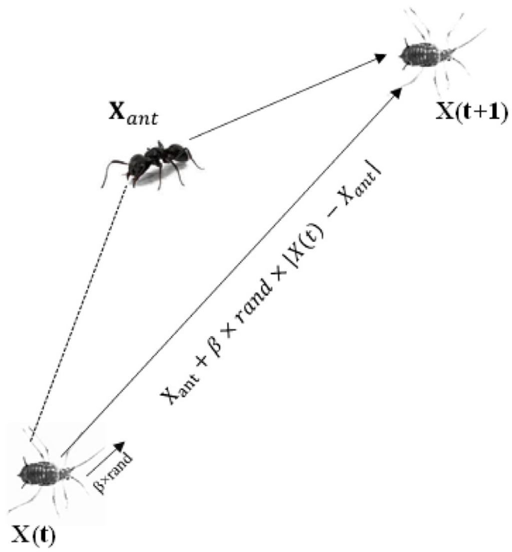


Fig. 2. Updating aphids' position using ants.

In Eq. (6), β is a random number with normal distribution in the range of [0,1], and $rand$ is also a random number with uniform distribution in this interval. Both parameters are used to control the influence of the distance between the aphid and the randomly selected ant on searching domain around the ant. Furthermore, $Ant_{rand}(t)^n$ demonstrates the randomly selected ant's position in the tth iteration in the nth colony. Fig. 2 shows the role of the selected ant in this updating strategy of aphids.

4.3.3. Aphids' movement using other aphids in the colony

The remaining weak aphids that are not evolved so far, are updated either using their own experience or other aphids [20]. Movement using other aphids leads to searching new areas in the colony to obtain better food resources and diversity increment in the population. Eq. (7) demonstrates this updating strategy [20]. The probability of remaining weak aphids' update using their own experience or other aphids is directly related to the iteration number and their fitness. Gradually, by increasing the iteration number and aphids' fitness, the proportion of weak aphids updated using their own experience increases compared to the ones evolved by other aphids. It enhances the exploitation rate in the final stages of the optimization process and, therefore, increases the chance of finding the global optimum.

$$Aphid_{i,j}(t+1)^n = Aphid_{j,rand}(t)^n + \alpha_2 \times rand \times |Aphid_{i,j}(t)^n - Aphid_{j,rand}(t)^n| \tag{7}$$

In Eq. (7), $Aphid_{i,j}(t+1)^n$ is the ith aphid's position in the jth dimension at $(t + 1)$th iteration in the nth colony. $Aphid_{j,rand}(t)^n$ indicates the weak aphid's position that is chosen randomly in the jth dimension at tth iteration in the nth colony. α_2 is a scaling parameter in the range of (0,2] to adjust the movement step, and $rand$ is a random number with uniform distribution in the range of (0,1]. In this strategy, two different solutions exchange information to find a better place in the colony and avoid premature convergence. In contrast, in the updating strategy based on aphid's own experience, the promising area around the current solution is investigated. In Fig. 3, the evolution using another aphid in the colony is demonstrated, schematically. Furthermore, in Algorithm 1, the pseudo-code of aphids' positions update in the colonies is explained.

4.4. Ants' evolution

The next position of ants directly depends on the aphid's position which is chosen as one of the top $10p\%$ individuals in the colony with $p\epsilon(0, 1]$. In this part, a parameter called 'pheromone' which is initialized randomly in the range of [−1,1] in the beginning of the optimization process is used to control the ants' movement toward

Algorithm1: Aphid position update in the colonies

Begin:

for *i = 1 to* $N_{Ringleader}$ *(for each colony)* ***do***

  for *j = 1 to* NC_{aphid} *(for all aphids in each colony)* ***do***

    if *rand <* μ ***then***

      Update $aphid_{i,j}$ *by equation (5);*

    else if *rand <* λ ***then***

      Update $aphid_{i,j}$ *using equation. (6);*

    else if *rand <= (Fitness(*$aphidi_{,j}$*) / (t*100))* ***then***

      Update $aphid_{i,j}$ *by equation. (7);*

    else

      Update $aphid_{i,j}$ *by equation (5);*

    end if

  end for

end for

End.

the selected best aphid. In each iteration, the pheromone amount decreases based on a parameter called evaporation rate. In initial iterations, the evaporation rate is a small number close to zero and it increases gradually to 1. As a result, in the early steps, the ants explore the space between the selected best aphids and their positions. While in the last iterations, they focus on exploitation of the promising areas around the best aphids. The evaporation rate is calculated using Eq. (8). Furthermore, the pheromone amount is updated using Eq. (9) [12]. Afterward, the position of ants in each colony is evolved using Eq. (10).

$$\rho = \frac{t - 0.1}{MaxIter} \tag{8}$$

$$Ant.Tau_i(t+1)^n = (1-\rho) \times Ant.Tau_i(t)^n \tag{9}$$

$$Ant_{i,j}(t+1)^n = Ant.Tau_i(t+1)^n \times |Aphid_{best,j}(t)^n - Ant_{i,j}(t)^n| \tag{10}$$

In Eq. (8), ρ is the pheromone's evaporation rate [12]. Moreover, t and $MaxIter$ are the current iteration and maximum number of iterations, respectively. In Eq. (9), $Ant.Tau_i(t)^n$ is the pheromone amount for the ith ant at tth iteration in the nth colony. Moreover, $Ant.Tau_i(t+1)^n$ is the pheromone amount for this ant in the next iteration. In Eq. (10), $Aphid_{best,j}(t)^n$ is the position of the best aphid in the jth dimension at tth iteration which is

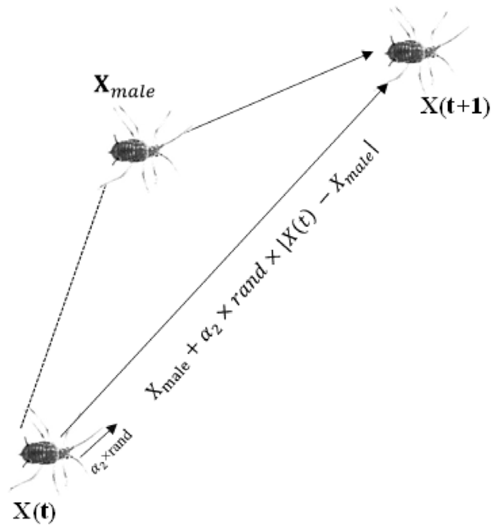


Fig. 3. Aphid update using other randomly selected aphid (X_{male}) in the colony.

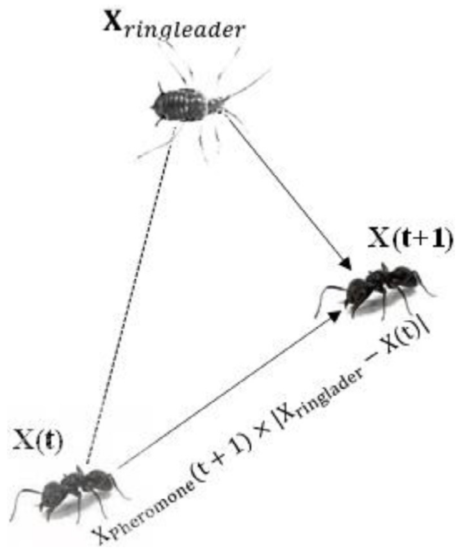


Fig. 4. Ants' update in each colony.

chosen as one of the top $10p\%$ individuals in the nth colony with $p\epsilon$ (0, 1]. Moreover, $Ant_{i,j}(t)^n$ is the position of the current ant in the jth dimension at tth iteration in the nth colony and $Ant_{i,j}(t+1)^n$ indicates the position of this ant in the next iteration. Fig. 4 demonstrates ants' position update in each colony using the randomly selected best aphid, which is demonstrated as $X_{ringleader}$.

4.5. Aphids' flight

Due to their small size, aphids can only migrate to adjacent colonies. $10p\%$ of the best-winged aphids (Alettes) in each colony with $p\epsilon(0, 1]$ whose fitness has not improved for consecutive iterations can utilize wind's blow and fly spirally to the other host plants with more sap to explore new areas in the search space. Destination colony is selected randomly among adjacent ones. Eq. (11) shows the flight formulation of aphids [23,29].

$$Aphid_{i,j}(t+1)^n = \left|Aphid_{best,j}(t) - Aphid_{i,j}(t)^n\right| \times e^{b*l} \times \cos(2\Pi l) + Aphid_{best,j}(t) \quad (11)$$

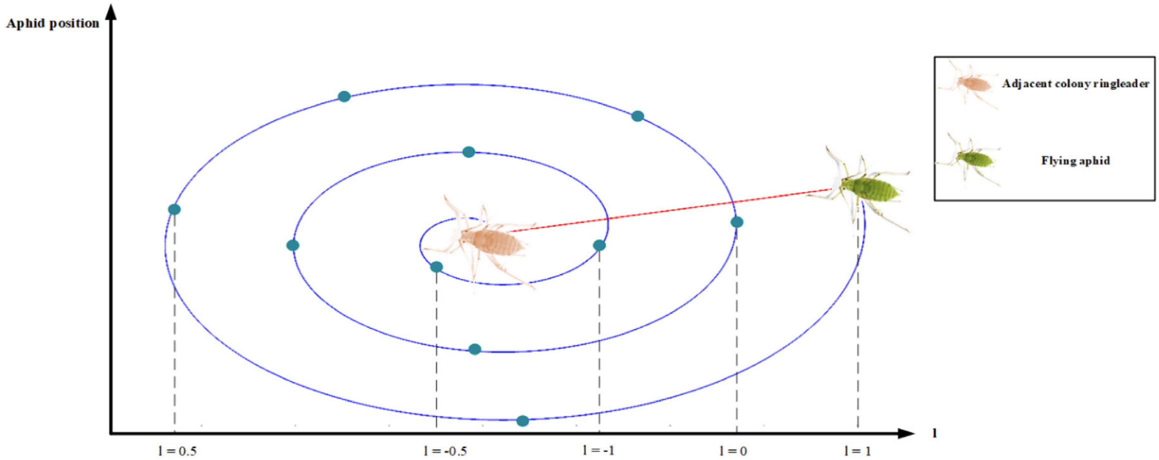


Fig. 5. Some of the positions reached by Alette in spiral flight to adjacent colonies.

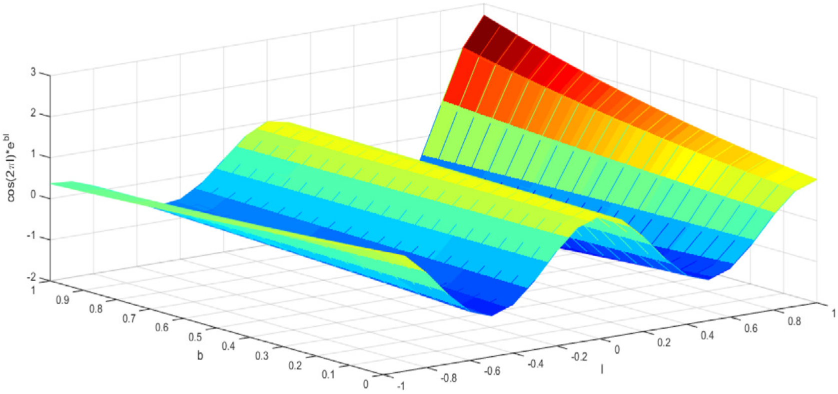


Fig. 6. Shape of spiral flight.

In Eq. (11), $Aphid_{best,j}(t)$ indicates one of the best aphids' positions in the jth dimension in adjacent colonies that is selected randomly in current iteration. The ith aphid is directed using wind's flow to the colony of this aphid. b is a parameter to specify the shape and the angle of the flight and is selected randomly in the range of (0,1]. Furthermore, l which is selected randomly in the range of [−1,1], controls the closeness of the aphid in the spiral movement to the adjacent colony's ringleader [23,29]. Fig. 5 demonstrates some of the positions reached by flying aphids (Alettes) in the spiral movement. Moreover, Fig. 6 shows the shape of spiral movement in this strategy. Furthermore, the pseudo-code of the proposed algorithm is presented in Algorithm 2 and, a brief schematic view of the algorithm is demonstrated in Fig. 7. Finally, in Table 2, parameters of AAM are summarized.

4.6. Computational complexity analysis

Computational complexity of an optimization algorithm can be described as a function that maps the runtime of the algorithm to the size of the specified problem. For this reason, Big-O notation is utilized here as a prevalent terminology [5]. The time complexity of AAM can be calculated using Eq. (12) as follows:

$$O\left(AAM\right) = O\left(initialization\right) + O\left(fitness\ evaluation\ \&\ sorting\right) + O\left(position\ update\right) \tag{12}$$

As Eq. (12) suggests, the time complexity of AAM depends on the maximum number of iterations (T), the population size (N_{pop}), and the problem's dimension (D). The overall time complexity of AAM under the termination

Algorithm2: Aphid – Ant Mutualism Algorithm

Begin:

Step1: Initialization of the initial population: Initialize the population of N_{pop} with n aphids and m ants in which $Aphid_{k,j}$, $Ant_{p,j}$, k = 1,….,n and, p =1,….,m are calculated by Eq. (1).
Set the iteration counter $t = 1$; Set the Number of Functions call iterator NFC = 0; Set the maximum iteration number MaxIter; Set the maximum number of function call M_NFC; Set μ, λ , α_1 and α_2.

Step2: Fitness evaluation: calculate the fitness of all aphids. Sort the fitness to select $N_{Ringleader}$ of the fittest aphids as ringleaders; Update NFC;

Step3: Colonies' generation: calculate the attraction power of each ringleader using Eq. (2). attract weak aphids (N_{pop} - $N_{Ringleader}$) in the initial population to each ringleader's colony based on Eq. (3). Furthermore, attract ants to each colony based on the attraction power of each ringleader by Eq. (4).

Step4: While the best solution is not found **or** NFC<M_NFC **do**

for i = 1 to $N_{Ringleader}$ (for each colony) **do**

  for j = 1 to NC_{aphid} (for all aphids in each colony) **do**

    if rand < μ (choose $aphid_{i,j}$ as one of the 30% of best individuals) **then**

      Update $aphid_{i,j}$ by Eq. (5);

    else if rand < λ **then**

      Update $aphid_{i,j}$ using Eq. (6);

    else if rand <= (Fitness ($aphid_{i,j}$) / (t*100)) **then**

      Update $aphid_{i,j}$ by Eq. (7);

    else

      Update $aphid_{i,j}$ by Eq. (5);

    end if

  end for

for k = 1 to to NC_{ant} (for all ants in each colony) **do**

Update evaporation rate using Eq. (8);

Update pheromone for each ant using Eq. (9);

  Update $ant_{i,j}$ by Eq. (10);

end for

  Evaluate each individual in the colony using fitness function;

  Update NFC;

  choose 10p% of the best aphids in each colony ($p\epsilon(0,1]$) ($Ringleader_j$);

  if new position of the $Ringleader_i$ is not better than the old one **then**

    update $Ringleader_i$ using Eq. (11);

  end if

  for j = 1 to NC (for all individuals in each colony) **do**

    if new position is better than old one for each individual **then**

      Accept it;

    else

      Withdraw it and continue with old position;

    end if

  end for

end for

$t = t+1;$

Step5: end while

Step6: Output the best solution

End.

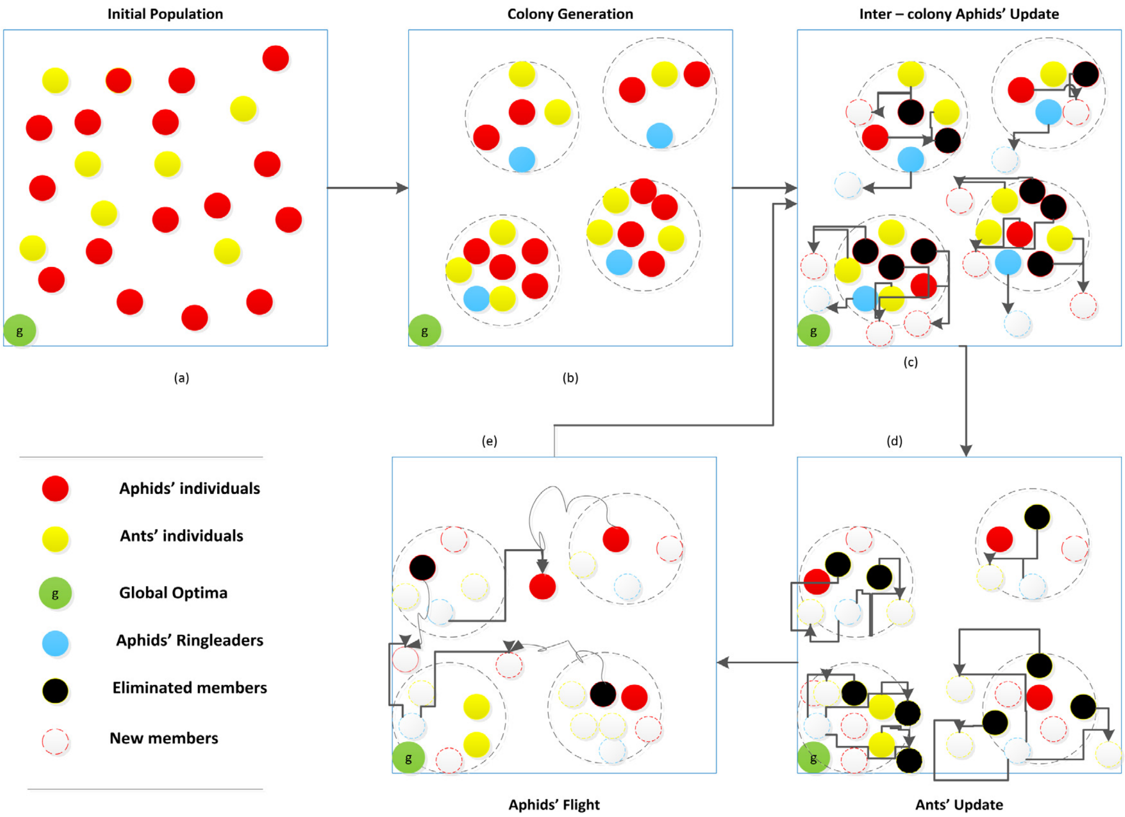


Fig. 7. Schematic brief view of the proposed algorithm.

Table 2
The parameters of AAM.

Parameter	Value
SP	1.1
α_1	$\epsilon(0, 1]$
μ	0.3
λ	0.5
β	randn[0,1]
α_2	$\epsilon(0, 2]$
b	rand(0,1]
l	rand[−1,1]

method can be computed as Eq. (13):

$$O\left(AAM\right) = O\left(N_{pop}\right) + O(N_{pop} + N_{pop}\log(N_{pop})) + O\left(\frac{2}{3}N_{pop} \times D\right) + O\left(\frac{N_{pop}}{3} \times D\right) \tag{13}$$

Therefore, the total complexity of AAM is summarized in Eq. (14):

$$O\left(AAM\right) = O(N_{pop} \times \left(1 + T \times N_{pop} \times \left(1 + \log\left(N_{pop}\right) + D\right)\right)) \tag{14}$$

As it is described, the complexity issue of the AAM is one of the polynomial orders, which can be considered as an efficient metaheuristic optimization algorithm [5,22].

Table 3
Description of unimodal benchmark functions.

Name	Dimension	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} {x_i}^2$	30, 50,100	[100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30, 50, 100	[10,10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30, 50, 100	[100,100]	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30, 50, 100	[100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100\left(x_{i+1} - {x_i}^2\right)^2 + (x_i - 1)^2]$	30, 50, 100	[30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30, 50, 100	[100,100]	0
$f_7(x) = \sum_{i=1}^{n} i{x_i}^4 + random[0.1)$	30, 50, 100	[128,128]	0

5. Numerical experiments

This section demonstrates the effectiveness of the introduced algorithm on a set of 41 benchmark test functions, including 23 commonly used unimodal and, multimodal functions, as well as 18 functions from CEC 2014, CEC 2017 and, CEC-C06 2019 test suite. This study uses both quantitative and qualitative validation metrics on an all-inclusive set of ten competitive metaheuristics. Quantitative metrics include average, standard deviation, and non-parametric Wilcoxon rank-sum test with a 5% significance degree for various benchmarks and qualitative metrics include problems' morphologies and convergence curves.

In Section 5.1, the experimental settings and description of benchmark functions are explained in detail. Furthermore, the compared optimizers are discussed in Section 5.2. Finally, the detailed experimental results and extensive analysis of AAM parameters are presented in Sections 5.3, 5.4, 5.5, and 5.6, respectively.

5.1. Experimental settings

To verify the performance of the proposed algorithm, we employ 41 benchmark functions, as listed in Tables 3–7 [2,3,11]. These benchmarks generally are categorized into four different groups. The first set is unimodal benchmarks (F1–F7) that have specific global optimums and highlight the exploitative characteristics of the proposed algorithm (Table 3) [28,37]. The second group consists of multi-modal functions (F8–F23) that have several local optimums which is used to evaluate the exploratory behavior of AAM and its ability to prevent premature convergence is demonstrated in Tables 4–5 [19,41]. The third set is selected from shifted, rotated, and expanded functions in CEC 2014 and CEC 2017 that are harder to solve than the classical benchmarks. They are utilized in many studies to evaluate the global convergence speed of the algorithms and can assess the potential of AAM to make a perfect balance between exploration and exploitation behaviors and escaping from the local optima. Moreover, they evaluate the ability of the proposed algorithm in dealing with non-separable, noisy, and multidimensional problems. Details of the discussed functions are listed in Table 6 [2,3,11,19,37].

Furthermore, as the fourth group, a set of ten benchmark functions, called CEC-C06 2019 that have been proposed by professor Suganthan et al. [5] is used to evaluate the performance of AAM. Table 7 describes this group of benchmark functions. F32 and F36–F41 are shifted and rotated functions while F33–F35 are neither shifted nor rotated. They are used in many recent studies to evaluate the ability of the algorithm to make a perfect balance between exploration and exploitation and scaping from local optima in dealing with noisy and multidimensional problems.

All algorithms were performed under the same situations to achieve fairness in comparative experiments. Among them, to reduce the impacts of random parameters on the results and explore the stability of the proposed algorithm compared to others, all algorithms were run individually 30 times in each benchmark function and averaged as the

Table 4
Description of multimodal benchmark functions.

Name	Dimension	Range	$f_{\min}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\|x_i\|})$	30, 50, 100	[500,500]	$-418.9829 \times n$
$f_9(x) = \sum_{i=1}^{n} [x_i{}^2 - 10\cos(2\pi x_i) + 10]$	30, 50, 100	[5.12,5.12]	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i{}^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30, 50, 100	[32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i{}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30, 50, 100	[600,600]	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$; $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a0 < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30, 50, 100	[50,50]	0
$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30, 50, 100	[50,50]	0

final running result. For all of them, the maximum number of function evaluations (MNFEs) is set to $D \times 1000$ as the termination condition, where D represents the dimension of the test functions and, a total population number of 50 is utilized [2,11]. All the algorithms are implemented in MATLAB software (R2016b) in a 32-bit computer with Windows 10 operating system and 4.00 GB RAM.

5.2. Comparison of AAM with other optimization algorithms

In order to show the superiority of our proposed AAM algorithm, we compare it with ten other well-known optimizers, namely PSO [36], DE [10], ALO [27], HHO [19], GOA [37], SSA [28], EO [14], SMA [22], IGWO [30], Ali Baba and the forty thieves (AFT) [5] that is a novel metaheuristic that inspired from human interactions. We have used AFT for comparison because research shows the metaheuristics that inspired by human interactions demonstrate more promising results [5].

It is worth mentioning that the compared algorithms include both novel algorithms (e.g., AFT) and state-of-the-art methods (e.g., PSO and DE) that have promising results in other studies. In algorithm selection, we try

Table 5
Description of fixed-dimension multimodal benchmark functions.

Name	Dimensions	Range	$f_{\min}$
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6})^{-1}$	2	[−65,65]	1
$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1 \left(b_i{}^2 + b_i x_2\right)}{b_i{}^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0.00030
$f_{16}(x) = 4x_1{}^2 - 2.1x_1{}^4 + \frac{1}{3}x_1{}^6 + x_1x_2 - 4x_2{}^2 + 4x_2{}^4$	2	[−5,5]	−1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1{}^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[−5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1{}^2 - 14x_2 + 6x_1x_2 + 3x_2{}^2\right)]$	2	[−2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i\, exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i\, exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.5363

Table 6
Description of shifted rotated functions from CEC 2014 and CEC 2017.

Name	Dimensions	Range	Function bias
Rotated high conditioned elliptic function	30	[100,100]	−450
Shifted Schwefel's function	30	[100,100]	−450
Shifted and rotated Rosenbrock's function	30	[100,100]	390
Shifted and rotated Griewank's function	30	[0,600]	−180
Shifted and rotated Ackley's function	30	[32,32]	−140
Shifted Rastrigin's function	30	[5,5]	−330
Shifted and rotated Weierstrass function	30	[0.5,0.5]	90
Shifted and rotated expanded Griewanks's plus Rosenbrock's function	30	[−3,1]	−130

Table 7
Description of benchmark functions of CEC-C06 2019 test suite.

Name	Dimensions	Range
Shifted and rotated expanded Scaffer's F6 function	10	[100,100]
Storn's Chebyshev polynomial fitting problem	9	[8192,8192]
Inverse Hilbert matrix problem	16	[16 384,16 384]
Lennard-Jones minimum energy cluster	18	[4,4]
Rastrigin's function	10	[100,100]
Griewangk's function	10	[100,100]
Weierstrass function	10	[100,100]
Modified Schwefel's function	10	[100,100]
Happy cat function	10	[100,100]
Ackley function	10	[100,100]

to consider and compare the performance of our algorithm with methods that have proper exploration abilities like DE, those that can exploit the search space appropriately like PSO, and others try to provide a perfect balance between explorative and exploitative behaviors like another selected algorithm especially, SMA and

Table 8
The parameter settings of compared algorithms.

Algorithm	Parameter settings
PSO	$c_1 = 1.5; c_2 = 2; vMax = 6$
DE	$scaling\ factor = rand\,(0.2, 0.8)\,; crossover\ probability = 0.5$
HHO	$scaling\ factor\ in\ levy\ flight = 1.$
GOA	$intensity\ of\ attraction = 0.5; Attractive\ length\ scale = 1.5;$ $C_{min} = 0.00004; C_{max} = 1$
SSA	$c_1 \epsilon\,[0\ 1]\,; c_2 \epsilon [0\ 1]$
EO	$a_1 = 2; a_2 = 1; G_0 = 0.5$
SMA	$z = 0.03$
IGWO	$a = [2, 0]$
AFT	$\alpha_0 = 1; \alpha_1 = 2; \beta_0 = 0.1; \beta_1 = 2$

IGWO. This comparison can highlight promising features and the appropriate ability of our algorithm to tackle different classes of the optimization problems. The parameters of these algorithms are the same as described in Table 8 [5,10,14,19,22,27,28,30,36,37].

5.3. Experimental results

Table 9 shows the obtained results of the proposed algorithm and the compared ones in scalable functions (F1–F13) with 30 dimensions and population size of 50. The results of AAM are promising in most of the evaluated functions (except F10–F11), which shows the proper explorative and exploitative abilities of the proposed algorithm. Moreover, as shown in Table 10, the results of our algorithm in multi-modal functions (F14–F23) with fixed dimensions and population size of 50 are significantly better in almost all cases (except F21). It demonstrates the ability of our method to prevent premature convergence and evade trapping of local optima. In shifted, rotated, and extended functions (F24–F31) from CEC 2014 and CEC 2017, AAM achieves high-quality solutions along with a fast convergence rate. It shows the perfect balance between the exploration and exploitation abilities of the proposed algorithm (Table 11). Table 12 shows the performance of AAM compared to ten other comparative methods on CEC – C06 2019 test suite. AAM shows acceptable performance specially in shifted, rotated and, multidimensional functions of this group (except F38). Moreover, to demonstrate variations between our method and other optimizers, we perform a non-parametric Wilcoxon rank-sum test with a 5% significance degree for scalable functions (F1–F13) with 30 dimensions. As shown in Table 13, in most cases, the results of AAM are significantly more remarkable than other optimization algorithms.

Furthermore, convergence graphs of the AAM and the other evaluated algorithms and parameter spaces of some benchmark functions are shown in Fig. 8 for unimodal scalable functions (F1–F13) with 30 dimensions, Fig. 9 for multi-modal benchmarks, and Fig. 10 for shifted and rotated ones, respectively. These results consist of problems' morphology and convergence curve for each function. Convergence metric demonstrates alternation of the fitness value for the best ringleader (best solution) during the optimization process. In most convergence curves, the reduction pattern in the early steps of optimization is apparent. It shows perfect soft alternation from exploration to exploitation during different stages of the algorithm. Based on this observation, it is conducted that AAM has a rapid convergence rate, and it can achieve promising results in low iteration numbers. Compared to the other ten comparative algorithms, AAM continues the search from the more promising areas due to using the information exchange between different search agents with various behaviors. Furthermore, in shifted, rotated, and extended functions, the proposed algorithm obtains a perfect performance and the higher convergence velocity. For example, in F27, other algorithms cannot find the global optimum properly, while AAM can find it rapidly and precisely.

5.4. Impact of dimension increment

The results of scalable functions (F1–F13) with 50 and 100 dimensions are reported in Tables 14–15. The results of AAM are more promising than ten other comparative algorithms in most of the benchmark functions. Furthermore, a non-parametric Wilcoxon rank-sum test with a 5% significance degree is performed to demonstrate

Table 9
Results of benchmark functions (F1–F13) with 30 dimensions and population size of 50 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	AVG	**9.93e−3**	1.88e−1	9.49e+0	4.89e−1	8.16e−1	2.57e−1	7.88e−2	6.75e+2	1.10e−2	2.54e−2	8.35e+0
	STD	**1.57e−2**	2.24e0	2.40e+1	6.26e−1	2.91e+0	2.58e+0	3.46e−1	5.06e+2	3.02e−2	2.11e−2	6.80e+0
F2	AVG	**1.00e−2**	2.31e−2	5.27e−2	3.07e−2	1.51e−2	2.71e−2	1.09e−2	2.20e+1	1.08e−1	1.23e−2	3.51e+0
	STD	**2.41e−3**	2.42e−1	1.12e−1	5.20e−2	9.11e−2	1.48e−1	3.42e−2	1.21e+1	3.19e−3	5.26e−3	2.95e+0
F3	AVG	**4.51e−3**	1.13e−1	2.63e+0	3.10e−2	4.51e−1	8.86e−1	2.62e−1	2.94e+4	6.83e−2	1.64e+2	8.88e+2
	STD	**8.73e−3**	1.36e+0	8.59e+0	7.56e−2	1.84e+0	3.42e+0	2.48e+0	6.80e+3	1.90e−1	7.45e+1	3.15e+2
F4	AVG	**4.42e−2**	8.41e−2	2.05e+0	4.85e−1	6.24e−2	2.68e−1	1.18e−1	4.08e+1	2.61e−1	3.34e−1	1.49e+1
	STD	**8.81e−3**	6.91e−1	1.90e+0	9.57e−1	3.48e−1	1.15e+0	5.20e−1	7.10e+0	3.54e−2	1.68e−1	2.74e+0
F5	AVG	**2.18e−2**	1.74e−1	3.06e+0	4.75e−1	3.31e−1	6.32e−1	1.03e+0	5.94e+5	6.29e−1	2.74e+1	6.80e+2
	STD	**2.70e−2**	1.64e+0	3.15e+0	2.21e+0	1.04e+0	2.32e+0	1.18e+1	4.77e+5	9.27e−1	1.04e+0	5.65e+2
F6	AVG	**1.47e−2**	1.15e+0	1.15e+1	6.67e−1	4.09e−1	8.03e−2	2.03e−1	6.37e+2	1.65e−1	4.68e−1	6.15e+0
	STD	**6.94e−2**	1.16e+1	3.48e+1	1.22e+0	2.27e+0	5.88e−1	1.82e+0	2.51e+2	2.26e−1	2.94e−1	4.97e+0
F7	AVG	**3.15e−4**	1.21e−3	8.40e−3	1.21e−3	3.20e−3	1.60e−3	4.54e−4	9.55e+1	2.35e−1	2.04e−2	5.86e+3
	STD	**6.85e−5**	3.74e−3	9.80e−3	2.60e−3	1.06e−2	4.60e−3	9.22e−4	1.71e+1	2.07e−1	7.40e−3	7.14e+3
F8	AVG	**−7.23e+5**	6.41e+0	9.06e+1	3.15e+1	7.19e+0	2.10e+0	1.31e+0	−6.49e+3	−1.26e+4	−5.55e+3	−6.65e+3
	STD	4.22e+4	3.97e+1	1.04e+2	7.84e+1	6.85e+1	9.20e+0	6.57e+0	4.94e+2	**4.02e−1**	2.05e+2	3.65e+2
F9	AVG	**7.01e−3**	8.14e−2	1.16e+0	1.93e−1	1.24e−1	4.24e−2	5.06e−2	1.27e+2	7.69e−2	7.02e+1	7.42e+1
	STD	**1.92e−2**	5.97e−1	1.13e+0	3.07e−1	4.76e−1	2.03e−1	2.03e−1	2.69e+1	2.35e−1	5.36e+1	2.13e+1
F10	AVG	1.20e−1	7.22e−2	2.37e+0	9.93e−1	2.38e−1	1.75e−1	1.38e−1	1.98e+1	7.40e−3	**2.97e−4**	4.24e+0
	STD	3.18e−2	5.59e−1	1.70e+0	1.27e+0	8.75e−1	7.54e−1	4.86e−1	8.55e−1	1.49e−2	**1.21e−4**	1.03e+0
F11	AVG	2.01e−2	4.00e−2	2.35e−1	7.77e−2	2.14e−2	2.00e−2	4.11e−2	6.54e+0	**1.28e−2**	4.97e−2	1.05e+0
	STD	**4.21e−3**	2.81e−1	3.68e−1	1.32e−1	6.31e−2	5.85e−2	1.34e−1	2.77e+0	2.79e−2	8.17e−3	5.66e−2
F12	AVG	**6.13e−3**	6.94e−3	1.13e+0	1.20e−1	1.55e−2	1.38e−1	4.01e−2	2.23e+1	5.23e−2	3.37e−2	8.24e+0
	STD	1.60e−2	7.23e−2	1.39e+0	1.05e+0	4.55e−2	9.14e−1	2.75e−1	4.16e+1	**1.18e−3**	3.60e−2	3.45e+0
F13	AVG	**6.94e−3**	5.24e−2	1.30e+2	2.00e−2	2.85e−2	1.19e−2	1.10e−2	4.68e+5	8.17e−3	4.09e−1	3.22e+1
	STD	**2.53e−3**	5.80e−1	6.69e+2	4.57e−2	1.87e−1	8.32e−2	5.91e−2	6.36e+5	1.23e−2	1.77e−1	1.47e+1

Table 10
Results of benchmark functions (F14–F23) with population size of 50 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F14	AVG	**1.30e−3**	6.19e−1	7.20e−1	7.18e−1	5.05e+0	3.79e−1	1.03e−1	9.98e−1	9.98e−1	9.98e−1	9.98e−1
	STD	1.84e−2	3.26e+0	1.78e+0	1.95e+0	7.90e−1	1.18e−1	**1.98e−16**	2.83e−16	8.31e−11	2.81e−11	4.20e−16
F15	AVG	**6.79e−4**	7.69e−4	2.16e−2	1.70e−3	1.80e−3	7.71e−4	4.51e−3	2.00e−3	7.61e−4	7.41e−4	1.23e−3
	STD	**2.98e−5**	5.20e−3	7.80e−3	2.60e−3	2.50e−3	6.19e−4	1.50e−3	3.73e−3	2.17e−4	3.88e−5	3.64e−3
F16	AVG	**−1.03e+0**	−1.01e+0	−1.02e+0	−1.01e+0	−1.02e+0	−1.02e+0	−1.02e+0	**−1.03e+0**	**−1.03e+0**	**−1.03e+0**	**−1.03e+0**
	STD	**5.76e−16**	4.22e−2	8.53e−2	6.05e−14	1.04e−13	2.56e−3	6.12e−16	5.90e−16	4.88e−7	1.04e−10	6.05e−16
F17	AVG	**3.98e−1**	5.95e−1	4.11e−1	4.04e−1	3.99e−1	4.05e−1	4.03e−1	**3.98e−1**	**3.98e−1**	**3.98e−1**	**3.98e−1**
	STD	**0.00e+0**	1.13e−13	5.22e−2	2.19e−14	7.69e−14	1.86e−3	**0.00e+0**	**0.00e+0**	4.77e−7	1.28e−12	**0.00e+0**
F18	AVG	**3.00e+0**	3.42e+0	3.43e+0	3.34e+0	3.01e+0	3.03e+0	3.06e+0	**3.00e+0**	**3.00e+0**	**3.00e+0**	**3.00e+0**
	STD	**1.13e−16**	3.47e−1	6.05e−1	4.79e−13	9.67e−13	1.54e−1	1.78e−15	1.94e−15	1.50e−6	2.29e−2	3.03e−15
F19	AVG	**−3.34e+0**	−2.99e−1	−3.32e−2	−2.99e−1	−2.99e−1	−2.99e−1	−2.99e−1	−3.004e−1	−3.004e−1	−3.004e−1	−3.004e−1
	STD	2.48e−16	**0.00e+0**	3.15e−10	2.26e−16	2.26e−16	2.80e−16	2.26e−16	2.26e−16	2.26e−16	3.15e−12	7.23e−7
F20	AVG	**−3.32e+0**	−1.85e+0	−3.29e+0	−3.30e+0	−3.21e+0	−3.12e+0	−3.31e+0	−3.20e+0	−3.21e+0	−3.32e+0	−3.20e+0
	STD	3.02e−11	1.76e−1	2.77e−1	2.25e−13	3.05e−11	1.03e−1	1.66e−15	2.05e−2	2.18e−2	1.48e−2	**1.47e−15**
F21	AVG	−5.24e+0	−4.91e+0	−2.01e+0	−6.91e+0	−2.52e+0	−6.91e+0	−5.01e+0	−5.03e+0	**−1.02e+1**	−8.28e+0	−5.27e+0
	STD	1.10e−6	6.58e−1	5.96e−1	**5.25e−11**	2.81e+0	1.61e+0	6.13e−1	1.66e+0	1.74e−3	2.11e+0	9.22e−1
F22	AVG	**−1.04e+1**	−4.44e+0	−5.72e+0	−1.03e+1	−2.55e+0	−5.90e+0	−4.99e+0	−7.20e+0	−1.03e+1	−1.03e+1	−9.19e+0
	STD	**3.02e−4**	1.50e+0	3.78e+0	1.69e+0	2.83e+0	2.35e+0	1.35e+0	3.54e+0	1.61e−3	8.84e−3	2.53e+0
F23	AVG	**−1.05e+1**	5.50e+0	5.22e+0	3.18e+0	8.22e+0	2.73e+0	5.43e+0	−9.87e+0	−1.04e+1	−1.04e+1	−1.03e+1
	STD	**1.25e−3**	5.34e−1	3.18e+0	3.56e+0	3.44e−1	2.77e+0	1.22e−1	2.05e+0	1.93e−3	6.13e−3	1.41e+0

variations between our algorithm and other optimizers. The *p*-values of the Wilcoxon rank-sum test for scalable functions (F1–F13) are demonstrated in Tables 16–17 with 50 and 100 dimensions, respectively. Based on the values in Tables 16 and 17, it is clear that the differences between our algorithm and other methods are statistically significant in most cases. In Table 16, it can be seen that in F4 and F6, the performance of AAM is close to SMA and in F9 its performance is as well as GOA, DE and SMA. Furthermore, in F11 and F12, AAM performs close to GOA and SMA, respectively. Moreover, in 100 dimensions, GOA, DE, ALO, PSO, SMA and IGWO have close performance in some benchmark functions (Table 17). Consequently, it can be considered that increasing the problem dimensions does not degrade the performance of the proposed algorithm and its stability is retained.

5.5. Influence of the population size

To evaluate the effect of mutualistic relationship of different individuals with various searching behaviors on the performance of the AAM, we utilize different numbers of search agents consisting of 50, 100, 150 and to achieve

Table 11
Results of benchmark functions (F24–F31) with population size of 50 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F24	AVG	**1.72e+3**	4.53e+8	1.58e+8	3.29e+8	1.90e+8	4.24e+8	1.21e+8	1.05e+8	3.69e+7	1.65e+7	2.30e+7
	STD	**1.14e+4**	2.99e+8	1.43e+8	4.94e+8	1.95e+8	3.41e+8	2.62e+8	4.68e+7	1.81e+7	6.09e+6	1.19e+7
F25	AVG	**5.72e−1**	7.59e+4	7.55e+4	4.44e+4	4.49e+4	7.77e+4	2.30e+4	5.35e+4	2.29e+4	1.08e+4	3.02e+4
	STD	**2.48e+0**	2.03e+4	3.67e+4	2.69e+4	1.75e+4	2.07e+4	1.49e+4	1.64e+4	7.77e+3	7.28e+3	7.28e+3
F26	AVG	**2.24e+1**	1.62e+2	1.17e+2	1.05e+3	2.27e+2	1.76e+2	7.84e+1	3.89e+8	2.03e+5	1.65e+6	1.67e+5
	STD	**2.33e+2**	1.31e+3	5.98e+2	3.56e+3	7.12e+2	1.15e+3	4.59e+2	4.02e+8	1.03e+5	2.17e+6	1.74e+5
F27	AVG	**2.10e+0**	3.92e+1	3.92e+1	3.95e+1	3.90e+1	4.01e+1	3.95e+1	4.70e+3	4.76e+3	4.70e+3	6.35e+3
	STD	7.27e+0	3.98e−1	3.24e+0	3.30e+0	3.23e+0	2.25e+0	3.23e+0	1.46e+1	1.79e+1	**1.53e−1**	2.68e+2
F28	AVG	2.10e+1	2.00e+1	1.99e+1	2.11e+1	1.99e+1	6.59e+0	2.00e+1	2.10e+1	2.11e+1	**6.08e+0**	2.11e+1
	STD	3.89e−2	3.06e−2	1.63e+0	8.03e+0	1.63e+0	3.68e+0	5.41e−2	7.23e−2	6.59e−2	**6.40e−3**	1.17e−1
F29	AVG	**4.28e+0**	4.20e+2	2.93e+2	2.16e+2	1.61e+2	1.72e+1	1.72e+1	1.10e+2	1.26e+2	1.35e+2	1.66e+2
	STD	**1.63e−2**	1.82e+0	9.12e+1	1.47e+0	2.79e+1	6.59e−2	3.39e−1	2.22e+1	2.59e+1	5.95e+1	2.73e+1
F30	AVG	**3.99e+0**	4.69e+1	4.13e+1	2.69e+1	3.48e+1	1.47e+1	1.44e+1	3.52e+1	2.96e+1	4.06e+1	3.32e+1
	STD	**1.11e−2**	1.69e+0	3.34e−2	2.21e+0	3.39e+0	2.85e−1	3.85e−1	2.57e+0	3.46e+0	4.70e+0	3.34e+0
F31	AVG	**1.73e+0**	4.63e+1	2.73e+1	5.75e+0	2.56e+1	5.91e+2	4.22e+1	1.79e+1	1.58e+1	1.48e+1	2.57e+0
	STD	**1.99e−1**	7.70e+1	8.16e+0	4.73e−1	2.36e+1	3.42e+2	1.24e+2	8.43e+0	3.08e+0	2.67e+0	1.58e+1

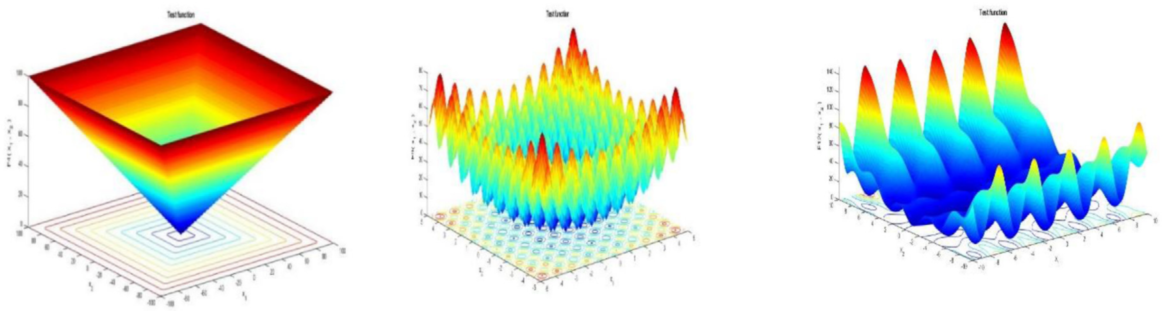
Table 12
Results of benchmark functions (F32–F41) with population size of 50 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F32	AVG	**1.30e+1**	1.42e+1	1.38e+1	1.31e+1	1.39e+1	1.40e+1	1.38e+1	1.32e+1	1.35e+1	1.33e+1	1.35e+1
	STD	1.22e−1	**7.05e−2**	1.16e+0	1.10e+0	3.16e−1	2.16e−1	1.70e−1	3.88e−1	3.27e−1	4.37e−1	4.60e−1
F33	AVG	**1.68e+4**	8.76e+10	1.00e+11	2.90e+8	4.54e+10	9.60e+10	4.20e+9	2.19e+8	1.33e+5	3.61e+8	5.23e+8
	STD	**6.81e+5**	5.54e+11	1.71e+11	4.38e+8	3.92e+10	1.68e+11	3.50e+9	4.88e+8	7.18e+5	3.81e+8	6.17e+8
F34	AVG	**1.72e+1**	3.24e+2	4.48e+3	1.73e+1	1.74e+1	6.83e+2	1.73e+1	1.73e+1	1.73e+1	1.73e+1	1.73e+1
	STD	1.18e−6	1.52e+3	3.84e+3	5.01e−4	1.90e−2	2.21e+3	9.05e−8	**1.01e−8**	6.70e−4	3.51e−4	1.98e−6
F35	AVG	**1.27e+1**	**1.27e+1**	1.28e+1	**1.27e+1**	**1.27e+1**	1.28e+1	**1.27e+1**	**1.27e+1**	**1.27e+1**	**1.27e+1**	**1.27e+1**
	STD	**3.13e−15**	9.46e−4	1.04e+0	3.61e−15	1.24e−12	5.18e−4	3.61e−15	6.13e−15	7.53e−4	9.58e−8	4.04e−15
F36	AVG	**1.43e+1**	1.65e+4	4.89e+3	1.59e+1	6.45e+1	1.26e+3	2.20e+1	1.46e+1	3.81e+1	4.77e+1	3.89e+1
	STD	1.48e+1	1.21e+3	5.87e+3	**1.93e−4**	5.26e+1	2.80e+3	8.57e+0	6.47e+0	1.44e+1	9.76e+0	1.94e+1
F37	AVG	**1.04e+0**	5.07e+0	2.87e+0	1.27e+0	1.25e+0	2.12e+0	1.15e+0	1.07e+0	1.38e+0	1.57e+0	1.16e+0
	STD	2.52e−1	2.47e−1	1.10e+0	**1.52e−7**	1.04e−1	7.01e−1	1.08e−1	4.37e−2	2.10e−1	1.54e−1	9.32e−2
F38	AVG	1.03e+1	1.20e+1	1.04e+1	**5.23e+0**	1.59e+1	1.05e+1	1.05e+1	1.10e+1	1.16e+1	1.11e+1	1.16e+1
	STD	5.15e−1	1.05e+0	1.38e+0	**6.80e−3**	1.91e+0	5.94e−1	2.67e+0	8.46e−1	1.44e+0	5.55e−1	1.12e+0
F39	AVG	**1.62e+2**	1.51e+3	9.34e+2	3.01e+2	6.09e+2	5.68e+2	2.75e+2	1.89e+2	3.14e+2	6.99e+2	3.23e+2
	STD	**1.51e+0**	2.21e+2	2.48e+2	3.95e+0	2.75e+2	3.67e+2	1.61e+2	1.73e+2	2.16e+2	1.81e+2	2.11e+2
F40	AVG	**1.98e+0**	1.41e+3	1.53e+3	2.40e+0	3.06e+0	4.00e+2	2.37e+0	2.48e+0	3.30e+0	2.58e+0	2.58e+0
	STD	4.80e−1	3.52e+2	1.80e+3	1.75e−1	4.46e−1	7.39e+2	**5.42e−2**	7.90e−2	5.54e−1	1.05e−1	1.50e−1
F41	AVG	**1.79e+1**	2.06e+1	2.05e+1	2.00e+1	2.00e+1	2.04e+1	2.02e+1	1.93e+1	2.03e+1	1.80e+1	1.99e+1
	STD	8.62e−2	1.50e−1	1.69e+0	**7.25e−9**	8.99e−2	9.70e−2	1.89e−1	4.69e+0	1.79e−1	6.55e+0	3.28e+0

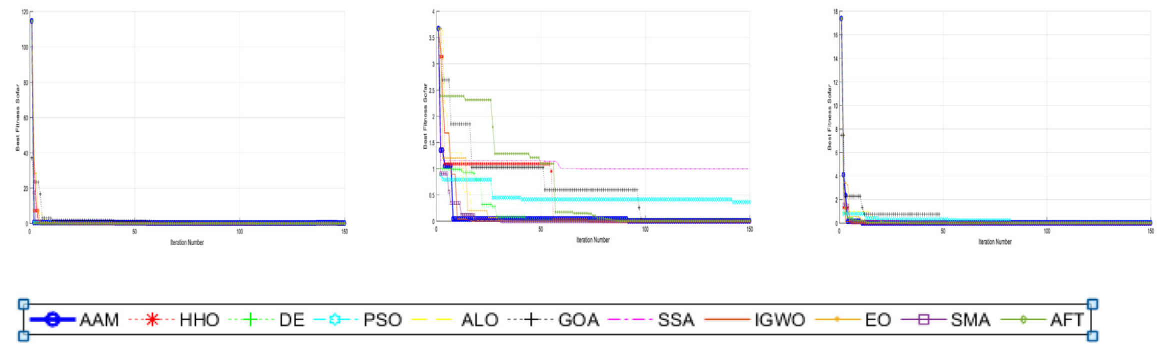
fairness in comparative experiments, the dimensions set to 30 for scalable functions (F1–F13) for all population sizes. The results compared to other studied algorithms are shown in Tables 9–12 for population size of 50 and Tables 18–25 for 100 and 150 search agents, respectively. In scalable function (F1–F13), our proposed algorithm with 100 search agents outperforms other algorithms in almost all cases (except F11). Moreover, in multi-modal functions (F14–F23), using 100 search agents leads to promising results in most functions except F15 that AFT outperforms AAM. Furthermore, in shifted, rotated and extended functions (F24–F31), AAM achieves promising results compared to other comparative optimizers with 100 search agents. Moreover, in CEC-C06 2019 benchmark functions (Table 21), AAM outperforms other ten comparative functions in almost all cases. By increasing search individuals, the results become more promising, and, with 150 search agents, the proposed algorithm outperforms other optimizers in almost all evaluated functions. It is due to increasing the amount of information sharing between individuals (aphids and ants) with different searching behaviors. This mutualism leads to finding promising areas, which contain the global optimum faster and prevent premature convergence to the local optima.

Table 13
p-values of the Wilcoxon rank-sum test with 5% significance for F1–F13 with 30 dimensions (p-values $\geq$ 0.05 are shown in bold face).

	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	6.72e−10	3.02e−11	5.57e−10	2.15e−10	1.96e−10	1.70e−9	3.02e−11	3.02e−11	**4.38e−1**	3.02e−11
F2	2.15e−6	4.69e−8	4.12e−6	5.87e−4	6.77e−5	2.42e−2	9.92e−11	3.02e−11	3.02e−11	3.02e−11
F3	5.07e−10	3.02e−11	4.62e−10	4.20e−10	3.02e−11	1.33e−4	3.52e−7	3.02e−11	3.02e−11	3.02e−11
F4	1.64e−5	3.02e−11	4.18e−9	1.08e−2	9.26e−9	7.66e−5	1.46e−10	3.02e−11	3.02e−11	3.02e−11
F5	5.49e−11	3.02e−11	2.61e−10	1.17e−9	1.61e−10	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F6	3.02e−11	3.02e−11	3.02e−11	8.99e−11	3.08e−8	2.37e−10	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F7	2.49e−6	3.02e−11	8.49e−9	5.46e−9	6.53e−8	3.39e−2	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F8	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F9	8.35e−8	3.02e−11	4.98e−11	3.02e−11	4.42e−6	1.07e−9	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F10	1.40e−3	8.99e−11	2.57e−7	6.91e−4	2.24e−2	4.22e−4	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F11	5.87e−4	1.41e−9	1.17e−5	4.69e−8	**1.30e−1**	1.68e−4	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F12	**1.34e−1**	3.47e−10	5.07e−10	1.78e−4	1.33e−10	1.03e−6	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F13	6.72e−10	3.02e−11	5.09e−8	3.57e−6	1.30e−3	1.03e−2	3.02e−11	3.69e−11	3.34e−11	3.02e−11



(a)

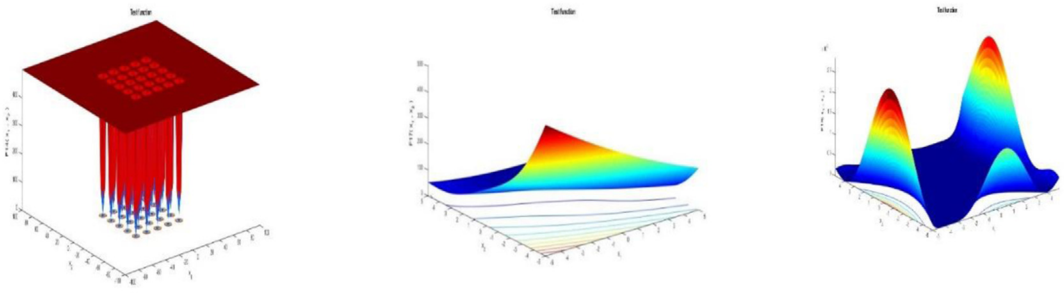


(b)

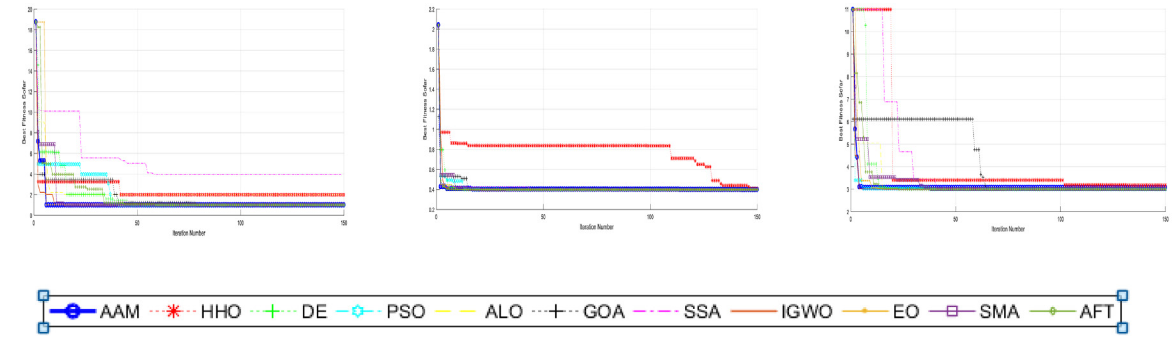
Fig. 8. (a) The problem space of unimodal scalable function (F4, F9 and F12) (b) The convergence curves of the proposed algorithm and other compared methods for these three functions.

5.6. Evaluating the effect of the free parameters

In the proposed algorithm, α_1 and α_2 are free parameters. They represent the step size in agents' movement using their own experience and individuals' update using other aphids in the colonies, respectively. Furthermore, there are some random numbers with different distributions (e.g., β) as mentioned in Table 2 to make an acceptable balance between exploration and exploitation abilities. We examine different values for α_1 and α_2 for 41 benchmark functions to analyze the effect of the free parameters on the performance of the proposed algorithm and find the best values for them. The sensitivity of the AAM to these two parameters is shown in Tables 26–31. The best

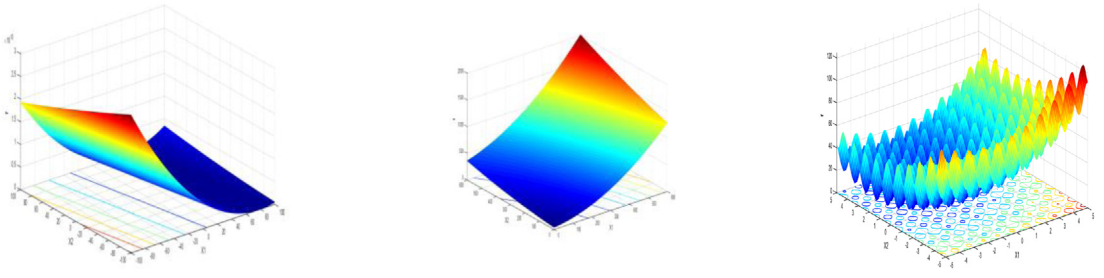


(a)

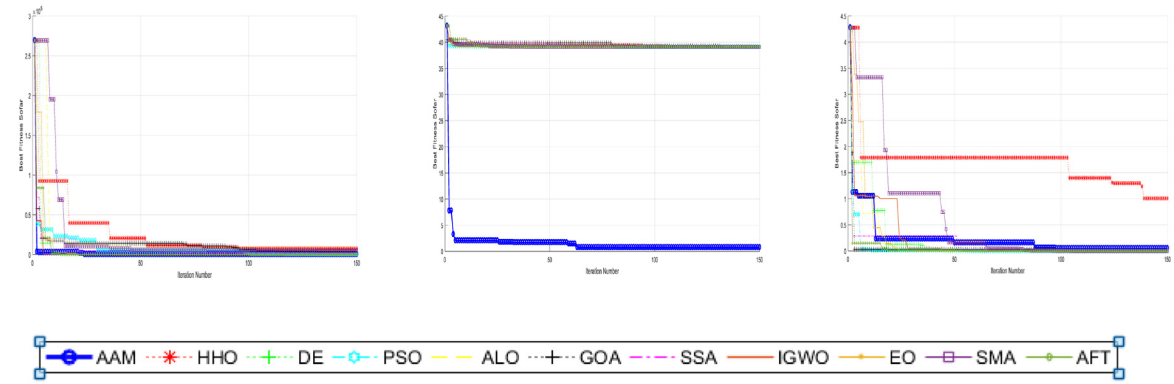


(b)

Fig. 9. (a) Problem space of multi-modal functions (F14, F17 and F18) (b) The convergence curves of the proposed algorithm and other compared methods for these three functions.



(a)



(b)

Fig. 10. (a) Problem space for shifted, rotated, and extended functions from CEC 2014 and CEC 2017 (F24, F27 and F29) (b) The convergence curves of the proposed algorithm and compared methods for these three functions.

Table 14
Results of benchmark functions (F1–F13) with 50 dimensions (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	AVG	2.26e−1	4.19e+0	2.00e+4	7.75e−1	2.04e+4	3.66e+4	1.81e+0	1.25e+4	5.97e−3	**3.68e−3**	1.96e+2
	STD	**8.02e−2**	4.21e+1	1.04e+4	9.00e+3	8.59e+3	3.31e+4	2.90e+0	5.71e+3	1.69e−1	2.11e−1	7.15e+1
F2	AVG	**7.52e−3**	5.86e−2	6.13e+6	4.34e+9	2.08e+20	3.22e+20	6.66e+0	9.55e+1	2.37e−2	8.43e−3	1.45e+1
	STD	**8.02e−2**	4.76e−1	5.87e+7	5.32e+10	7.75e+20	1.99e+21	6.07e+0	2.70e+1	2.89e−1	2.23e−1	4.93e+0
F3	AVG	**6.66e−1**	4.29e+0	6.13e+4	1.12e+4	6.35e+4	1.55e+5	9.23e+2	8.19e+4	8.35e−1	3.54e+3	3.92e+3
	STD	**3.31e+0**	3.48e+1	4.31e+4	1.23e+4	1.81e+4	4.31e+4	1.65e+3	1.65e+4	1.60e+0	1.68e+3	9.36e+2
F4	AVG	**1.72e−1**	9.94e−1	5.57e+1	1.42e+1	4.09e+2	8.31e+1	5.92e−1	5.52e+1	2.12e−1	3.64e+0	1.88e+1
	STD	5.01e−1	7.55e+0	6.57e+0	6.86e+0	7.14e+0	3.40e+0	4.51e−2	5.76e+0	**4.42e−2**	1.05e+0	2.78e+0
F5	AVG	**6.51e−1**	1.20e+1	1.20e+7	3.36e+5	2.13e+7	1.02e+8	2.71e+2	1.28e+7	1.44e+0	5.00e+1	1.19e+4
	STD	**2.48e+0**	1.47e+2	2.89e+7	1.84e+6	2.63e+7	1.32e+8	5.26e+1	6.39e+6	3.54e+0	1.61e+0	5.93e+3
F6	AVG	7.45e−1	2.41e+0	2.21e+4	1.95e+3	3.67e+4	3.45e+4	1.99e+1	1.19e+4	**2.84e−1**	3.72e+0	2.14e+2
	STD	3.27e+0	2.91e+1	1.06e+4	2.31e+3	9.60e+3	2.80e+4	8.59e+0	4.07e+3	**5.37e−1**	7.16e−1	7.50e+1
F7	AVG	**2.30e−3**	5.10e−3	6.62e+1	3.33e+0	3.01e+1	7.13e+1	8.49e+0	2.21e+9	2.77e−1	4.27e−1	6.12e+5
	STD	**1.19e−2**	2.84e−2	3.43e+1	1.23e+0	5.50e+1	9.05e+1	1.57e+1	1.68e+9	4.26e−1	2.47e−1	3.78e+5
F8	AVG	**−2.42e+4**	−8.15e+2	−7.45e+3	−9.11e+3	−8.78e+3	−7.10e+3	−3.51e+1	−1.16e+4	−2.10e+4	−8.50e+3	−1.35e+4
	STD	1.04e+3	7.05e+1	1.85e+3	2.78e+3	1.22e+3	1.43e+3	5.90e+0	8.23e+2	**1.06e+0**	2.72e+3	3.40e+2
F9	AVG	**2.45e−1**	8.99e+0	4.25e−1	1.10e+2	3.48e+2	4.85e−1	1.48e+2	3.30e+2	2.79e−1	1.13e+2	1.44e+2
	STD	**4.91e−1**	5.30e+1	1.21e+2	1.39e+2	1.33e+2	7.79e+0	1.38e+2	5.89e+1	6.17e−1	6.29e+1	3.92e+1
F10	AVG	**1.12e−1**	3.97e−1	1.59e+1	5.21e+0	1.64e+1	1.66e+1	2.14e+0	1.99e+1	9.69e−1	8.47e−1	7.33e+0
	STD	7.69e−1	8.30e−1	1.49e+0	4.65e+0	1.61e+0	2.64e+0	5.53e−1	3.07e−1	1.62e−2	**2.41e−3**	1.14e+0
F11	AVG	**9.87e−2**	4.74e+0	1.63e−1	1.63e+1	1.28e+2	3.05e+2	5.13e−1	1.03e+2	2.22e−1	1.50e−1	2.94e+0
	STD	2.41e−1	4.76e+1	1.00e+2	1.91e+1	7.56e+1	2.62e+2	8.79e−2	3.88e+1	5.04e−2	**1.85e−2**	9.51e−1
F12	AVG	**9.04e−2**	1.17e−1	1.55e+7	2.29e+4	9.46e+6	1.89e+8	2.17e+0	7.58e+6	3.83e−1	2.57e−1	1.70+1
	STD	**5.23e−1**	9.08e−1	8.32e+7	7.00e+5	2.77e+7	2.68e+8	5.85e−1	1.10e+7	7.23e−1	1.59e+0	6.85e+0
F13	AVG	**2.27e−2**	3.26e−2	3.57e+7	1.86e+5	4.48e+7	4.33e+3	3.77e−1	4.32e+7	3.53e−1	3.09e+0	1.00e+2
	STD	**1.05e−1**	2.65e−1	3.57e+7	3.70e+6	8.97e+7	6.30e+3	4.92e−1	2.35e+7	6.22e−1	6.54e−1	2.09e+1

performance of the algorithm corresponds $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$. On 14 benchmark functions, these values lead to the best results compared to other values. There is another case which results in relatively good performance of the algorithm i.e., $\alpha_1 = 1$ and $\alpha_2 = 2$.

6. Conclusion & future works

In this work, a new population-based algorithm called AAM, which is inspired by the mutualism of aphids and ants' species has been proposed. In nature, it can be seen that simple animals utilize cooperation to work together that result in more complex and smarter behaviors. For this reason, despite the previous studies that the social behaviors of aphids and ants were simulated, separately, AAM models mutual interaction among aphids and ants. Therefore, AAM has new features by incorporating heterogeneous individuals consisting of aphids and ants that live in various colonies together and have different decentralized learning behaviors and objectives. Inspired by nature, using different search strategies including focusing on the individual's personal experience, learning from other colony's members and information sharing with adjacent colonies using flight operator are utilized. Forty-one unconstrained benchmark functions were used to evaluate the performance of the proposed AAM. The abilities of exploration, exploitation, and preventing premature convergence of the proposed algorithm were evaluated by unimodal, multi-modal, and shifted, rotated, and extended benchmark functions from well-known functions of recent studies, CEC 2014, CEC 2017 and, CEC-C06 2019 test suite using mean, standard deviation, convergence curve analysis and, a non-parametric Wilcoxon rank-sum test with a 5% significance degree. The results demonstrated that our algorithm achieved high-quality solutions in all of the benchmark functions compared to the other ten comparative algorithms.

For future works, we will consider Chao-based phases and more dynamic updating strategies to improve AAM. Moreover, this nature-inspired method will be used to develop low-cost and autonomous robotic swarms to explore and retrieve valuable resources in space missions [42] and also, multi-task optimization [34]. In addition, a multi-objective, large-scale and, feature selection version of the proposed algorithm will be introduced to solve engineering design problems.

Table 15
Results of benchmark functions (F1–F13) with 100 dimensions (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	AVG	**1.43e+3**	2.46e+5	1.65e+4	1.77e+4	3.37e+4	1.43e+3	1.03e+4	9.81e+4	4.82e+3	2.56e+3	2.03e+3
	STD	6.09e+3	2.33e+4	7.19e+3	9.11e+4	1.47e+4	5.91e+4	1.82e+4	1.64e+4	1.44e+3	7.07e+3	**4.73e+2**
F2	AVG	**5.35e+10**	8.72e+14	9.58e+24	1.07e+12	7.22e+14	5.35e+15	2.96e+16	3.14e+12	1.46e+11	4.36e+11	4.32e+11
	STD	**5.19e+9**	1.07e+16	8,27e+10	1.31e+13	8.84e+16	5.19e+19	3.62e+24	3.68e+11	1.85e+10	1.03e+10	5.27e+10
F3	AVG	**1.03e+3**	1.64e+5	1.64e+4	5.80e+4	8.99e+4	6.01e+3	7.30e+4	3.00e+5	9.51e+3	3.95e+4	1.65e+4
	STD	6.46e+4	8.79e+4	9.50e+4	5.05e+4	8.61e+4	1.66e+5	1.30e+5	5.71e+4	6.99e+3	9.30e+3	3.19e+3
F4	AVG	**4.57e+0**	1.39e+1	2.86e+1	2. 07e+1	3.86e+1	9.41e+1	5.37e+0	6.35e+1	2.38e+1	1.75e+1	2.12e+1
	STD	1.07e+1	9.30e+0	6.13e+0	6.48e+0	7.74e+0	1.27e+0	7.24e+0	5.29e+0	**5.16e−1**	3.13e+0	2.45e+0
F5	AVG	1.03e+3	9.39e+6	8.24e+6	6.72e+6	2.30e+7	5.00e+5	7.91e+6	2.13e+8	7.94e+4	**5.78e+2**	1.46e+5
	STD	1.05e+4	1.01e+8	6.76e+6	1.64e+7	3.97e+7	5.83e+4	4.56e+7	7.16e+7	2.72e+5	**2.27e+2**	3.54e+4
F6	AVG	**2.55e+3**	1.74e+5	1.85e+4	1.32e+4	2.82e+4	1.39e+5	9.26e+3	9.32e+4	2.20e+4	6.27e+4	2.45e+3
	STD	**5.84e+2**	2.09e+4	8.53e+3	9.65e+3	1.54e+4	5.84e+4	1.89e+4	1.84e+4	4.65e+3	9.90e+3	4.16e+2
F7	AVG	**7.56e+1**	1.28e+2	1.15e+2	1.66e+2	3.66+2	7.56e+2	1.10e+2	4.68e+5	5.41e+2	1.67e+5	1.96e+7
	STD	5.05e+1	1.39e+2	**1.83e+1**	2.17e+1	5.31e+2	5.05e+2	5.73e+1	1.72e+5	2.15e+2	3.47e+4	6.39e+6
F8	AVG	**−1.43e+13**	−4.03e+3	−1.02e+3	−9.81e+3	−1.79e+3	−1.05e+3	−1.53e+3	−2.03e+4	−3.20e+4	−1.18e+4	−1.68e+4
	STD	3.69e+3	4.69e+3	4.28e+3	4.29e+3	1.71e+3	2..23e+3	4.96e+3	**1.13e+3**	8.56e+3	4.75e+3	2.14e+3
F9	AVG	**1.17e+1**	3.44e+3	8.82e+1	7.15e+2	5.99e+2	1.25e+1	6.80e+2	9.61e+2	1.50e+2	2.37e+2	4.07e+2
	STD	1.85e+2	1.95e+2	2.01e+2	2.18e+2	2.37e+2	1.34e+2	2.71e+2	7.06e+1	6.46e+1	5.43e+1	5.26e+1
F10	AVG	**1.55e−1**	4.24e+1	1.72e−1	1.25e+1	1.66e−1	1.14e+0	1.01e+1	2.00e+1	9.27e+0	4.93e−1	8.50e+0
	STD	1.33e+0	2.34e+0	1.63e+0	1.65e+0	1.53e+0	7.97e−1	2.93e+0	**4.39e−4**	1.94e−1	1.45e−2	9.85e−1
F11	AVG	**1.05e+1**	1.92e+3	1.49e+2	1.12e+2	2.67e+2	1.14e+1	9.03e+1	8.30e+2	2.08e+1	9.06e+2	1.83e+1
	STD	6.61e+2	1.97e+2	7.25e+1	8.37e+2	1.26e+2	5.74e+1	1.65e+2	1.97e+2	2.00e+2	**4.83e−2**	3.03e+0
F12	AVG	**1.08e+0**	2.19e+9	3.08e+6	1.98e+9	1.01e+7	1.17e+7	7.74e+9	3.32e+8	1.20e+0	1.38e+0	2.80e+1
	STD	**4.61e−1**	2.34e+8	4.87e+6	1.72e+7	5.50e+7	7.73e+8	7.16e+7	1.69e+8	2.86e+0	6.69e−1	1.01e+1
F13	AVG	**9.64e+0**	3.27e+9	4.48e+8	1.98e+8	6.20e+8	2.20e+7	2.10e+9	6.60e+8	4.91e+0	1.22e+1	4.08e+3
	STD	4.49e+0	3.79e+8	**1.53e+7**	2.91e+7	1.42e+7	1.47e+8	1.54e+8	3.03e+8	2.38e+0	**2.23e+0**	3.44e+3

Table 16
p-values of the Wilcoxon rank-sum test with 5% significance for F1–F13 with 50 dimensions (p-values $\geq$ 0.05 are shown in bold face).

	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	3.02e−11	4.90e−3	**3.04e−1**	2.07e−2	1.90e−3	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F2	3.02e−11	2.57e−7	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	**5.90e−1**	3.02e−11
F3	3.02e−11	6.36e−5	2.39e−8	2.00e−3	9.10e−3	4.50e−11	3.02e−11	1.17e−5	3.02e−11	3.02e−11
F4	3.00e−6	5.19e−7	2.32e−2	1.03e−6	3.01e−7	4.57e−9	3.02e−11	**1.05e−1**	3.08e−8	3.02e−11
F5	3.02e−11	6.07e−11	3.02e−11	5.46e−9	4.90e−3	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F6	3.02e−11	2.01e−4	3.02e−11	1.84e−2	2.90e−3	3.02e−11	3.02e−11	**1.05e−1**	2.03e−9	3.02e−11
F7	1.10e−8	1.78e−10	3.82e−10	9.83e−8	3.65e−8	1.22e−2	3.02e−11	1.70e−9	6.77e−5	3.02e−11
F8	6.55e−4	1.91e−2	1.12e−2	3.30e−3	8.84e−7	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11
F9	3.02e−11	**1.41e−1**	1.61e−6	2.40e−3	**3.71e−1**	1.78e−4	3.02e−11	**2.40e−1**	3.02e−11	3.02e−11
F10	3.02e−11	1.08e−2	8.70e−3	4.51e−2	2.71e−2	6.74e−6	3.02e−11	1.46e−10	7.00e−3	3.02e−11
F11	3.02e−11	**1.62e−1**	3.02e−11	2.30e−3	2.42e−2	3.02e−11	3.02e−11	1.64e−5	8.30e−3	3.02e−11
F12	4.60e−3	3.02e−11	3.02e−11	3.02e−11	9.53e−7	3.02e−11	4.08e−11	**3.80e−2**	7.35e−6	2.99e−17
F13	3.02e−11	4.57e−9	3.02e−11	6.52e−9	1.84e−2	3.02e−11	4.50e−11	3.02e−11	3.02e−11	3.02e−11

Table 17
p-values of the Wilcoxon rank-sum test with 5% significance for F1–F13 with 100 dimensions (p-values $\geq$ 0.05 are shown in bold face).

	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	3.02e−11	1.43e−8	4.50e−11	2.00e−6	2.51e−2	1.56e−8	9.05–17	2.88e−4	5.09e−8	1.09e−11
F2	3.02e−11	3.02e−11	3.02e−11	2.03e−7	3.02e−11	3.02e−11	3.02e−11	2.57e−7	8.26e−4	3.02e−11
F3	3.02e−11	2.83e−8	8.89e−10	3.02e−11	1.91e−2	1.07e−9	3.02e−11	2.13e−9	6.70e−11	1.02e−5
F4	3.02e−11	8.00e−3	4.51e−2	9.50e−3	2.13e−4	**7.24e−2**	3.02e−11	1.64e−5	5.07e−10	2.88e−6
F5	2.15e−10	3.02e−11	3.02e−11	3.02e−11	1.17e−5	3.02e−11	3.02e−11	**2.12e−1**	**2.40e−1**	1.47e−7
F6	3.02e−11	2.37e−10	4.20e−10	6.72e−10	3.18e−4	3.35e−8	3.02e−11	8.49e−9	3.02e−11	9.79e−5
F7	2.15e−10	2.15e−10	3.02e−11	1.70e−8	2.28e−5	3.47e−10	3.02e−11	**2.92e−2**	1.21e−10	6.19e−11
F8	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.02e−11	1.21e−12	1.70e−8	3.16e−10
F9	3.02e−11	**3.11e−1**	1.50e−2	1.75e−5	**3.04e−1**	2.43e−5	1.21e−12	3.02e−11	1.29e−6	2.15e−10
F10	9.26e−9	**2.12e−1**	3.20e−3	**1.45e−1**	1.12e−2	2.92e−2	3.02e−11	1.21e−12	4.37e−6	3.02e−11
F11	3.02e−11	4.20e−10	6.70e−11	9.83e−8	**1.86e−1**	1.96e−10	3.02e−11	1.21e−12	2.84e−4	2.19e−8
F12	3.02e−11	3.02e−11	3.02e−11	3.02e−11	3.16e−5	4.50e−11	3.02e−11	**2.40e−1**	**3.50e−1**	3.02e−11
F13	3.02e−11	3.02e−11	3.02e−11	6.72e−10	4.94e−5	4.20e−10	3.02e−11	1.29e−3	2.96e−5	3.02e−11

Table 18
Results of benchmark functions (F1–F13) with 100 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	AVE	**7.45e−2**	2.67e−1	6.63e+0	1.53e+0	3.02e−1	4.93e−1	5.48e−1	1.56e+1	5.94e−1	5.80e−1	3.34e−1
	STD	8.14e−1	2.75e+0	1.073e+1	6.99e+0	9.74e−1	3.03e+0	3.66e+0	1.18e+1	**0.00e+0**	3.83e−1	1.95e−1
F2	AVE	**5.40e−2**	5.50e−2	1.72e−1	7.23e−2	2.35e+0	8.50e−2	6.06e−2	1.59e+1	1.10e−1	1.25e−1	1.33e+0
	STD	1.51e−1	5.08e−1	2.75e−1	1.03e−1	6.48e+0	6.04e−2	1.17e−1	1.12e+1	5.28e−1	**6.29e−7**	2.10e+0
F3	AVE	**1.14e−1**	1.60e−1	7.00e+0	4.33e−1	2.39e+0	6.62e−1	2.63e−1	2.26e+2	3.92e−1	2.17e+0	3.83e+2
	STD	1.03e+0	1.89e+0	2.185e+1	3.39e+0	1.12e+1	3.79e+0	1.74e+0	6.63e+1	**0.00e+0**	2.49e+0	1.32e+2
F4	AVE	**1.56e−1**	9.67e−1	1.06e+0	4.39e−1	1.84e−1	9.54e−1	1.90e−1	3.15e+1	7.62e−1	2.16e−1	9.37e+0
	STD	**2.53e−1**	7.58e+0	9.38e−1	7.70e−1	5.69e−1	1.02e+0	4.72e−1	7.18e+0	3.09e−1	2.59e−1	2.52e+0
F5	AVE	**1.83e−1**	4.54e−1	2.30e+0	9.20e−1	2.60e+0	1.36e+0	1.69e+0	3.01e+2	2.86e+1	2.61e+1	2.57e+2
	STD	**3.00e−1**	5.31e+0	1.80e+1	5.50e+0	1.49e+1	4.16e+0	5.91e+0	2.66e+2	6.58e−1	1.26e+0	2.82e+2
F6	AVE	**4.47e−1**	1.00e+0	5.35e+0	1.16e+0	6.44e−1	3.07e+0	6.32e−1	6.06e−1	5.82e−1	5.44e−1	5.48e−1
	STD	2.31e+0	9.79e+0	1.14e+1	3.07e+0	5.50e+0	2.51e+1	8.61e−1	**4.46e−5**	2.83e−1	2.35e−1	1.93e−1
F7	AVE	**3.10e−3**	8.33e−1	1.30e−2	8.00e−3	3.40e−3	9.48e+0	4.80e−3	9.59e−3	4.85e−3	5.76e−3	3.71e+1
	STD	**2.40e−3**	9.58e+0	1.44e−2	6.10e−3	2.50e−3	1.57e+1	1.40e−2	4.00e−3	3.28e−3	2.41e−3	8.99e+1
F8	AVE	**−4.14e+4**	−8.37e+2	−8.00+2	−8.26e+2	−8.25e+2	−8.35e+2	−8.37e+2	−9.06e+3	−1.17e+4	−7.71e+3	−6.75e+3
	STD	1.34e+2	**4.70e+0**	7.48e+0	6.90e+1	7.83e+1	1.61e+1	6.37e+0	6.16e+2	1.55e+3	1.94e+3	4.30e+2
F9	AVE	**1.20e−1**	1.26e−1	1.83e−1	3.87e−1	1.35e−1	1.59e−1	1.43e−1	6.86e−1	1.30e−1	3.12e+1	6.98e+1
	STD	4.81e−1	9.52e−1	7.14e−1	7.21e−1	6.17e−1	5.37e−1	6.26e−1	3.76e−1	**0.00e+0**	2.59e+1	2.34e+1
F10	AVE	**3.18e−1**	3.81e−1	2.81e+0	1.40e+0	4.38e−1	3.93e−1	3.50e−1	9.34e−1	8.88e−1	6.05e−1	2.98e+0
	STD	7.10e−1	2.23e+0	2.82e+0	1.62e+0	1.21e+0	1.14e+0	1.14e+0	8.12e−1	**0.00e+0**	2.83e+0	7.56e−1
F11	AVE	6.31e−2	3.14e−2	3.48e−1	1.59e−1	5.01e−2	4.48e−1	6.08e−2	**1.48e−3**	5.73e−2	4.34e−3	3.92e−1
	STD	1.99e−1	3.09e−1	3.75e−1	2.30e−1	**1.78e−1**	1.84e−1	2.03e−1	**5.96e−3**	1.54e−1	8.69e−3	1.88e−1
F12	AVE	**3.36e−2**	4.86e−2	7.80e−1	9.39e−2	8.54e−2	1.69e−1	6.23e−2	9.67e−2	1.73e−1	7.59e−2	5.08e+0
	STD	**1.26e−1**	5.22e−1	1.57e+0	4.43e−1	3.541e−1	1.28e+0	3.36e−1	5.16e−1	8.09e−1	6.56e−1	2.22e+0
F13	AVE	**3.33e−2**	6.79e−2	1.32e−1	5.33e−2	9.53e−2	7.73e−2	3.38e+0	4.49e−2	1.81e+0	2.12e−1	1.26e+1
	STD	**4.33e−2**	6.91e−1	3.22e−1	3.03e−1	4.77e−1	4.69e−1	4.12e+1	5.00e−2	1.08e+0	1.31e−1	1.08e+1

Table 19
Results of benchmark functions (F14–F23) with 100 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F14	AVG	**9.98e−1**	3.03e+0	1.55e+0	2.51e+0	1.07e+0	3.35e+0	2.12e+0	**9.98e−1**	**9.98e−1**	**9.98e−1**	**9.98e−1**
	STD	**3.47e−3**	1.20e+0	1.48e+0	2.57e+0	4.7e−1	2.48e+0	6.21e−1	1.84e−1	5.55e−1	3.82e−1	2.58e−1
F15	AVG	1.31e−3	4.20e−3	1.37e+0	3.70e−3	2.60e−3	2.40e−3	1.40e−2	3.43e−3	6.70e−3	3.46e−3	**3.08e−4**
	STD	3.54e−4	4.23e−2	4.68e−1	6.40e−3	6.40e−3	2.50e−3	5.70e−3	8.79e−5	2.83e−4	1.71e−4	**2.01e−11**
F16	AVG	**−1.03e+0**	**−1.03e+0**	−9.98e−1	−1.02e+0	−1.01e+0	−1.02e+0	−1.02e+0	**−1.03e+0**	**−1.03e+0**	**−1.03e+0**	**−1.03e+0**
	STD	2.14e−2	7.96e−2	8.58e−2	8.42e−2	9.86e−2	2.54e−2	**6.90e−3**	5.90e−1	6.57e−1	1.29e−1	5.22e−1
F17	AVG	**3.98e−1**	4.14e−1	5.34e−1	4.55e−1	4.24e−1	4.14e−1	4.28e−1	**3.98e−1**	**3.98e−1**	**3.98e−1**	**3.98e−1**
	STD	9.88e−14	1.84e−1	1.64e−1	9.84e−2	1.05e−1	9.92e−2	2.03e−1	**0.00e+0**	8.97e−5	5.07e−13	**0.00e+0**
F18	AVG	**3.00e+0**	7.25e+0	9.54e+0	3.81e+0	3.16e+1	8.91e+0	3.90e+0	3.01e+0	3.01e+0	3.02e+0	3.01e+0
	STD	1.85e−12	4.55e+1	1.20e+1	4.68e−13	9.11e−10	4.55e+1	3.24e−11	**1.79e−15**	4.74e−9	9.66e−13	2.01e−15
F19	AVG	**−3.34e+0**	−7.23e−2	−2.38e−2	−3.01e−1	−3.01e−1	−7.34e−2	−5.40e−3	−3.01e−1	−3.01e−1	−3.01e−1	−3.01e−1
	STD	**2.50e−3**	1.07e−1	3.15e−1	2.26e−1	2.26e−1	3.70e−2	2.94e−2	2.26e−1	2.26e−1	2.26e−1	9.09e−1
F20	AVG	**−3.25e+0**	6.08e−1	2.15e−1	4.02e−1	2.14e−1	7.78e−1	2.81e−1	−3.20e+0	−3.18e+0	−3.21e+0	−3.21e+0
	STD	6.84e−2	8.76e−2	5.26e−1	4.64e−1	2.63e−1	2.13e−1	1.66e−1	**1.20e−2**	4.27e−2	2.25e−2	2.17e−2
F21	AVG	**−1.02e+1**	−4.91e+0	−2.95e+0	−2.24e+0	−4.66e+0	−3.82e+0	−2.57e+0	−8.89e+0	−7.27e+0	−1.01e+1	−8.03e+0
	STD	1.13e+0	5.15e−1	1.78e+0	5.14e−1	1.15e+0	2.95e+0	2.60e−1	2.37e+0	2.57e+0	**1.43e−3**	3.12e+0
F22	AVG	**−1.04e+1**	−4.98e+0	−5.77e+0	−2.40e+0	−4.76e+0	−3.63e+0	−4.89e+0	−9.52e+0	−9.12e+0	**−1.04e+1**	−9.96e+0
	STD	1.24e+0	5.58e−1	2.99e+0	4.21e−1	9.40e−1	2.72e+0	6.25e−1	2.02e+0	2.38e+0	**2.43e−3**	1.69e+0
F23	AVG	**−1.05e+1**	−5.01e+0	−3.25e+0	−2.51e+0	−4.78e+0	−4.39e+0	−4.82e+0	−1.01e+1	−7.51e+0	−1.04e+1	−9.61e+0
	STD	**5.26e−4**	5.85e−1	1.19e+0	6.85e−1	1.16e+0	2.71e+0	1.75e+0	1.54e+0	2.94e+0	1.33e+0	2.44e+0

Table 20
Results of benchmark functions (F24–F31) with 100 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F24	AVG	**9.38e+2**	5.14e+4	2.57e+3	9.41e+2	4.53e+3	2.04e+3	1.62e+3	1.88e+7	3.42e+7	1.16e+7	1.35e+7
	STD	3.13e+3	4.91e+5	5.42e+3	**1.86e+3**	1.78e+4	1.07e+4	5.54e+3	6.80e+6	1.60e+7	4.26e+6	5.58e+6
F25	AVG	**1.8e+0**	2.41e+0	1.82e+1	3.43e+0	4.87e+4	1.89e+0	9.89e+2	1.11e+4	4.09e+4	6.90e+3	1.84e+4
	STD	4.74e+0	2.71e+1	2.75e+1	2.26e+1	9.01e+3	7.37e+0	**6.17e−1**	2.71e+3	1.38e+4	2.92e+3	5.81e+3
F26	AVG	**7.64e+0**	1.61e+2	6.81e+2	6.96e+2	1.16e+2	2.12e+1	1.24e+1	2.83e+3	2.04e+3	1.37e+6	7.70e+3
	STD	1.93e+1	8.87e+2	8.03e+2	3.21e+3	2.07e+1	8.93e+1	**1.18e+1**	4.03+3	2.83e+3	2.41e+6	6.90e+3
F27	AVG	**6.23e+0**	3.94e+1	3.93e+1	3.92e+1	3.90e+1	3.92e+1	3.95e+1	4.70e+3	4.70+3	4.70e+3	6.05e+3
	STD	1.24e+1	3.139e+0	3.25e+0	3.26e+0	3.21e+0	3.46e+0	1.95e+0	5.20e+0	2.78e+0	**4.68e−2**	2.42e+2
F28	AVG	**2.01e+1**	2.12e+1	2.50e+1	2.02e+1	2.07e+1	2.40e+1	2.13e+1	2.11e+1	2.11e+1	2.10e+1	2.11e+1
	STD	4.89e−2	**4.02e−2**	3.97e+0	1.06e−1	9.58e−2	3.46e+0	6.70e−2	6.08e−2	6.87e−2	7.11e−2	9.46e−2
F29	AVG	**2.59e+0**	4.16e+2	3.04e+2	1.44e+2	1.48e+2	2.93e+2	1.05e+2	6.89e+1	6.64e+1	1.06e+2	2.06e+2
	STD	**2.45e−1**	1.93e+1	6.51e+1	3.13e−1	3.53e+1	4.90e+1	3.27e+1	1.75e+1	2.50e+1	5.22e+1	4.06e+1
F30	AVG	**4.04e+0**	4.37e+1	2.53e+1	2.65e+1	2.97e+1	4.30e+1	2.93e+1	2.57e+1	2.52e+1	4.00e+1	3.01e+1
	STD	**9.27e−1**	1.44e+0	4.22e+0	4.10e+0	3.55e+0	6.25e+0	4.23e+0	3.91e+0	3.50e+0	3.55e+0	3.98e+0
F31	AVG	**2.15e+0**	5.00e+1	4.95e+1	9.93e+0	1.19e+1	3.87e+2	1.13e+1	6.03e+0	8.05e+0	1.20e+1	9.86e+0
	STD	**1.91e−1**	8.14e+1	3.30e+1	7.47e−1	3.61e+0	4.03e+2	5.24e+0	1.45e+0	2.62e+0	4.06e+0	2.11e+0

Table 21
Results of benchmark functions (F32–F41) with population size of 100 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F32	AVG	**1.19e+1**	1.42e+1	1.36e+1	1.35e+1	1.35e+1	1.36e+1	1.34e+1	1.26e+1	1.30e+1	1.29e+1	1.38e+1
	STD	**1.05e−1**	8.27e−1	1.16e+0	7.30e−1	3.63e−1	2.48e−1	2.76e−1	3.58e−1	4.76e−1	4.73e−1	2.36e−1
F33	AVG	**1.36e+4**	2.13e+10	1.14e+11	7.01e+7	4.54e+9	1.11e+11	2.60e+9	1.31e+8	1.86e+5	2.79e+8	2.87e+8
	STD	**6.60e+4**	1.89e+11	3.77e+11	8.30e+7	4.31e+9	3.75e+11	2.13e+9	3.74e+8	3.01e+5	3.75e+8	3.39e+8
F34	AVG	**1.73e+1**	1.90e+2	3.17e+3	**1.73e+1**	**1.73e+1**	4.40e+2	**1.73e+1**	**1.73e+1**	**1.73e+1**	**1.73e+1**	**1.73e+1**
	STD	1.03e−5	1.34e+3	2.26e+3	1.15e−4	1.08e−2	1.56e+3	1.56e−7	**1.36e−11**	4.94e−4	3.08e−4	1.49e−8
F35	AVG	**1.27e+1**	1.28e+1	1.28e+1	**1.27e+1**	1.28e+1	1.28e+1	**1.27e+1**	**1.27e+1**	**1.27e+1**	**1.27e+1**	**1.27e+1**
	STD	**2.98e−15**	7.20e−4	1.04e+0	3.61e−15	1.16e−13	3.18e−4	1.90e−6	3.76e−15	2.84e−4	5.34e−8	3.61e−15
F36	AVG	**1.16e+1**	1.34e+4	2.58e+3	3.33e+1	7.90e+1	9.99e+2	4.98e+1	1.17e+1	3.05e+1	4.70e+1	2.60e+1
	STD	**1.84e+0**	1.89e+3	2.94e+3	1.06e+1	4.00e+1	2.64e+3	1.08e+2	3.84e+0	1.24e+1	8.75e+0	1.09e+1
F37	AVG	**1.02e+0**	4.73e+0	2.66e+0	1.22e+0	1.19e+0	2.16e+0	1.16e+0	1.04e+0	1.28e+0	1.56e+0	1.18e+0
	STD	3.13e−2	2.41e−1	9.66e−1	**2.99e−11**	9.38e−2	6.17e−1	1.01e−1	3.73e−2	1.23e−1	1.24e−1	1.21e−1
F38	AVG	**9.91e+0**	1.21e+1	1.01e+1	1.45e+1	1.38e+1	1.17e+1	9.96e+0	1.09e+1	9.93e+0	1.05e+1	1.14e+1
	STD	**6.05e−1**	1.10e+0	1.32e+0	8.46e−1	1.82e+0	6.26e−1	2.29e+0	7.51e−1	1.38e+0	6.56e−1	6.54e−1
F39	AVG	**5.32e+1**	1.37e+3	7.89e+2	5.65e+1	3.28e+2	3.93e+2	1.09e+2	1.68e+2	2.14e+2	5.82e+2	9.19e+1
	STD	1.23e+2	2.32e+2	3.56e+2	**2.61e+1**	2.45e+2	4.10e+2	1.57e+2	1.45e+2	2.26e+2	1.71e+2	1.88e+2

(*continued on next page*)

Table 21 (*continued*).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F40	AVG	**2.34e+0**	1.03e+3	8.78e+2	2.35e+0	2.56e+0	2.44e+2	2.39e+0	2.40e+0	2.68e+0	2.50e+0	2.40e+0
	STD	1.82e−1	4.20e+2	1.04e+3	**1.89e−2**	1.44e−1	6.06e+2	2.63e−2	2.83e−2	1.84e−1	1.01e−1	4.98e−2
F41	AVG	**2.03e+1**	2.07e+1	2.05e+1	2.04e+1	2.04e+1	2.06e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.05e+1
	STD	6.01e−2	3.69e−2	1.69e+0	**2.45e−8**	3.97e−2	1.03e−1	1.90e−1	1.16e−1	4.73e+0	9.13e−2	1.59e−1

Table 22
Results of benchmark functions (F1–F13) with 150 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F1	AVE	**3.14e−2**	1.30e+0	6.20e+0	6.10e−2	7.09e−1	1.09e+0	4.00e−1	3.51e−2	3.25e−2	3.45e−2	4.25e−2
	STD	**9.08e−2**	1.44e+1	1.23e+1	3.33e−1	2.58e+0	6.08e+0	4.41e+0	4.67e−1	1.00e+0	3.53e−1	9.91e−2
F2	AVE	**2.30e−2**	6.30e−2	2.72e−1	5.99e−2	5.42e+0	6.33e−2	5.60e−2	2.99e−2	6.68e−2	2.58e−2	4.17e−1
	STD	5.07e−2	6.13e−2	2.26e−1	9.57e−2	5.45e+0	1.50e−1	4.32e−2	**1.06e−10**	3.60e−10	1.30e−7	2.35e−1
F3	AVE	**8.63e−2**	4.70e−1	4.47e+0	2.38e+0	1.77e−1	1.34e+0	3.30e−1	8.94e−2	4.13e−1	3.77e−1	2.20e+2
	STD	**2.23e−1**	4.69e+0	6.01e+0	6.43e+0	7.84e−1	1.24e+1	3.88e+0	2.77e−1	2.26e−1	3.36e−1	9.26e+1
F4	AVE	**2.29e−1**	9.05e−1	6.05e−1	2.66e−1	4.03e+0	3.41e−1	1.03e+0	6.85e−1	3.68e−1	3.18e−1	7.08e+0
	STD	1.46e−1	7.06e−1	6.17e−1	3.80e−1	7.59e+0	1.21e+0	1.27e+0	7.43e−1	1.39e−1	**7.47e−3**	1.84e+0
F5	AVE	**2.38e−1**	8.26e−1	1.60e+0	3.26e−1	3.65e−1	1.45e+0	6.58e+0	2.55e+1	2.72e+1	2.53e+1	1.53e+2
	STD	**1.61e−1**	5.48e+0	1.33e+0	3.95e−1	1.33e+0	6.81e+0	1.04e+1	2.97e−1	4.92e+0	5.21e−1	1.22e+2
F6	AVE	**3.05e−1**	9.39e−1	5.32e+0	2.07e+0	5.33e−1	6.13e−1	4.91e+0	3.75e−1	4.87e−1	3.83e−1	4.72e−1
	STD	1.68e+0	9.69e+0	6.14e+0	4.30e+0	1.72e+0	2.85e+0	1.32e+1	**1.69e−5**	1.07e−1	1.96e−1	3.85e−2
F7	AVE	**6.68e−4**	1.30e−3	1.04e−2	2.00e−3	2.90e−3	1.80e−3	1.41e−2	1.64e−3	8.33e−4	3.36e−3	2.03e+0
	STD	**1.90e−3**	8.50e−3	9.20e−3	6.20e−3	2.20e−3	5.40e−3	1.03e−2	9.96e−3	7.60e−3	2.24e−3	1.48e+0
F8	AVE	**−1.25e+4**	−8.38e+2	−7.83e+2	−8.29e+2	−8.31e+2	−8.31e+2	−8.35e+2	−8.75e+3	−1.17e+4	−8.17e+3	−7.57e+3
	STD	5.43e+2	**1.99e+0**	7.81e+1	6.88e+1	6.87e+1	3.05e+1	1.37e+1	6.58e+2	1.68e+3	1.85e+3	3.95e+2
F9	AVE	8.06e−2	1.51e−1	8.65e−1	2.60e−1	1.53e−1	1.49e−1	7.47e−1	1.08e−1	**0.00e+0**	3.26e+1	8.20e+1
	STD	1.45e−1	8.93e−1	8.24e−1	5.03e−1	4.78e−1	4.06e−1	6.03e−1	9.36e−1	**0.00e+0**	2.84e+1	2.39e+1
F10	AVE	**2.83e−1**	3.71e−1	1.88e+0	6.02e−1	3.83e−1	3.82e−1	9.67e−1	3.40e−1	8.88e−1	3.37e−1	2.42e+0
	STD	**3.95e−1**	2.13e+0	1.88e+0	1.27e+0	1.23e+0	1.15e+0	1.26e+0	8.13e−1	3.54e−1	8.26e−1	7.79e−1
F11	AVE	**4.72e−2**	4.85e+0	3.72e−1	6.32e−2	4.93e−2	5.95e−2	2.35e−1	5.07e−1	2.10e−1	4.84e−2	1.14e−1
	STD	**7.56e−2**	4.67e+1	5.90e−1	1.02e−1	1.76e−1	1.97e−1	9.37e−2	6.16e−1	3.00-1	7.68e−2	6.64e−1
F12	AVE	**7.52e−2**	8.89e−2	6.10e−1	3.85e−1	8.17e−2	1.99e−1	1.02e+0	1.45e−1	1.13e−1	8.10e−2	4.13e+0
	STD	**2.91e−1**	9.29e−1	8.09e−1	7.51e−1	4.95e−1	1.40e+0	9.82e−1	7.97e−1	5.43e−1	5.01e−1	1.84e+0
F13	AVE	**2.23e−2**	3.80e−2	8.55e+0	2.59e−2	2.94e−2	4.89e−2	8.43e−2	6.58e−2	1.71e+0	8.43e−2	3.51e+0
	STD	5.18e−2	3.72e−1	5.06e+1	9.79e−2	1.02e−1	2.80e−1	**1.86e−2**	1.98e−2	1.07e+0	8.66e−2	3.92e+0

Table 23
Results of benchmark functions (F14–F23) with 150 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F14	AVG	**9.98e−1**	6.00e+0	2.89e+0	9.99e−1	1.63e+0	1.16e+0	1.06e+0	**9.98e−1**	**9.98e−1**	**9.98e−1**	**9.98e−1**
	STD	**1.56e−3**	5.18e−1	2.67e+0	3.68e−1	9.87e−1	6.96e−1	3.62e−1	1.84e−1	1.51e−1	4.37e−1	1.89e−1
F15	AVG	**2.26e−4**	2.80e−3	4.50e−3	2.10e−3	2.60e−3	2.30e−3	9.30e−3	2.44e−3	7.17e−4	4.37e−4	7.04e−4
	STD	**2.55e−4**	1.40e−3	6.90e−3	2.70e−3	2.80e−3	4.80e−3	8.80e−3	6.09e−3	3.42e−4	2.96e−4	4.62e−4
F16	AVG	**−1.03e+0**	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0	−1.02e+0
	STD	6.74e−4	1.51e−2	8.39e−2	8.59e−2	8.84e−2	6.39e−2	1.34e−2	6.25e−16	5.92e−10	1.58e−10	**5.46e−16**
F17	AVG	**3.98e−1**	4.00e−1	4.29e−1	4.34e−1	4.00e−1	4.40e−1	4.17e−1	**3.98e−1**	**3.98e−1**	**3.98e−1**	**3.98e−1**
	STD	**7.37e−17**	4.40e−3	5.09e−2	1.08e−1	4.04e−2	2.04e−1	3.32e−2	1.03e−11	2.30e−13	6.48e−13	8.88e−15
F18	AVG	**3.00e+0**	4.00e+0	4.13e+0	3.63e+0	3.93e+0	7.36e+0	6.71e+0	**3.00e+0**	**3.00e+0**	**3.00e+0**	**3.00e+0**
	STD	**0.00e+0**	5.67e+0	2.53e+0	1.40e+0	3.81e−1	2.13e+1	8.72e+0	1.37e−15	3.67e−9	5.47e−13	3.03e−15
F19	AVG	**−3.35e+0**	−3.00e−1	−3.00e−1	−3.01e−1	−3.01e−1	−3.00e−1	−3.01e−1	−3.01e−1	−3.01e−1	−3.01e−1	−3.01e−1
	STD	**6.64e−17**	2.35e−2	2.26e−16	2.26e−16	2.26e−16	3.10e−3	2.26e−16	2.26e−16	2.26e−16	2.26e−16	3.50e−8
F20	AVG	**−3.32e+0**	−2.73e+0	−2.99e+0	−3.20e+0	−3.20e+0	−3.18e+0	−3.20e+0	−3.25e+0	−3.20e+0	**−3.32e+0**	−3.20e+0
	STD	**1.58e−15**	3.22e−1	3.70e−1	1.00e−3	5.80e−3	9.31e−2	6.45e−4	6.30e−2	1.15e−2	6.71e−5	1.87e−15
F21	AVG	**−1.02e+1**	−4.94e+0	−2.59e+0	−4.62e+0	−6.37e+0	−3.07e+0	−1.17e+0	−7.78e+0	−8.12e+0	**−1.02e+1**	−6.37e+0
	STD	1.16e+0	4.92e−1	1.39e+0	9.84e−1	2.86e+0	2.42e+0	1.04e−1	2.59e+0	2.53e+0	**2.22e−3**	2.36e+0
F22	AVG	**−1.04e+1**	−5.02e+0	−2.08e+0	−7.27e+0	−6.20e+0	−7.31e+0	−2.96e+0	−8.01e+0	−9.52e+0	−1.02e+1	−7.19e+0
	STD	1.35e+0	4.09e−1	4.99e−1	**3.01e−11**	4.57e−1	3.08e+0	1.00e+0	3.07e+0	2.01e+0	9.26e−1	3.75e+0
F23	AVG	**−1.06e+1**	−5.05e+0	−4.43e+0	−1.05e+1	−8.59e+0	−3.82e+0	−1.77e+0	−1.05e+1	−9.10e+0	−1.05e+1	−1.03e+1
	STD	1.35e+0	4.28e−1	3.40e+0	2.49e−2	3.06e+0	2.73e+0	4.28e−1	**4.18e−11**	2.42e+0	4.10e−4	1.22e+0

Table 24
Results of benchmark functions (F24–F31) with 150 search agents (Best results are shown in Bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F24	AVG	**4.66e+2**	6.41e+2	1.82e+4	7.27e+3	2.81e+3	4.66e+2	4.03e+4	1.22e+7	2.93e+7	9.68e+6	7.88e+6
	STD	2.12e+3	3.71e+3	1.48e+4	1.12e+3	**8.91e+2**	1.54e+3	2.82e+5	5.35e+6	1.37e+7	4.33e+6	3.07e+6
F25	AVG	**1.03e+0**	6.46e+0	2.20+1	4.33e+0	4.01e+2	1.43e+0	2.88e+0	1.08e+4	4.09e+4	5.74e+3	1.60e+4
	STD	5.12e+0	3.17e+0	2.64e+1	1.45e+1	**1.49e+0**	8.06e+0	1.63e+1	2.91e+3	1.11e+4	1.94e+3	5.10e+3
F26	AVG	**4.73e+0**	7.10e+0	1.03e+2	4.05e+1	7.36e+1	1.67e+1	3.36e+1	2.28e+3	1.93e+3	1.05e+6	2.23e+3
	STD	**3.59e+1**	3.61e+1	8.94e+1	1.01e+2	1.53e+2	5.71e+1	4.06e+1	3.77e+3	3.70e+3	2.41e+6	3.82e+3
F27	AVG	**3.92e+0**	3.94e+1	3.98e+1	3.95e+1	3.90e+1	3.92e+1	3.92e+1	4.70e+3	4.70e+3	4.70e+3	5.92e+3
	STD	8.62e+0	3.01e+3	3.22e+0	3.24e+0	3.23e+0	2.07e−1	2.15e−2	**1.91e−7**	1.09e−4	2.59e−2	2.27e+2
F28	AVG	**2.01e+1**	2.11e+1	2.89e+2	2.03e+1	2.06e+1	2.11e+1	2.11e+1	2.11e+1	2.11e+1	2.10e+1	2.11e+1
	STD	4.53e−2	9.88e−2	1.72e+0	5.45e−2	1.18e−1	7.19e−2	5.67e−2	5.57e−2	6.01e−2	**4.45e−2**	5.11e−2
F29	AVG	**4.22e+0**	4.13e+2	2.89e+2	1.40e+2	1.49e+2	3.11e+2	1.10e+2	6.46e+1	4.73e+1	8.38e+1	2.03e+2
	STD	**1.80e−3**	2.11e+1	1.03e+2	3.66e−2	3.99e+1	5.64e+1	2.87e+1	1.42e+1	1.65e+1	5.15e+1	4.53e+1
F30	AVG	**2.20e+1**	4.54e+1	4.05e+1	2.29e+1	3.12e+1	4.23e+1	3.08e+1	2.32e+1	2.56e+1	3.62e+1	3.06e+1
	STD	**9.59e−1**	1.79e+0	5.75e+0	1.10e+1	3.16e+0	2.05e+0	4.07e+0	3.53e+0	2.50e+0	8.67e+0	3.33e+0
F31	AVG	**3.59e+0**	4.48e+1	3.47e+2	5.14e+0	1.04e+1	3.84e+2	1.01e+1	5.06e+0	6.02e+0	1.18e+1	9.37e+0
	STD	6.07e+0	9.76e+1	4.64e+2	1.07e+0	2.98e+0	3.58e+2	4.17e+0	1.48e+0	**1.06e+0**	4.16e+0	3.12e+0

Table 25
Results of benchmark functions (F32–F41) with population size of 150 (Best results are shown in bold).

		AAM	HHO	GOA	SSA	ALO	DE	PSO	EO	SMA	IGWO	AFT
F32	AVG	**1.20e+1**	1.39e+1	1.39e+1	1.29e+1	1.32e+1	1.41e+1	1.32e+1	1.21e+1	1.30e+1	1.28e+1	1.30e+1
	STD	**1.21e−1**	1.22e−1	1.17e+0	5.20e−1	3.59e−1	2.09e−1	3.39e−1	6.22e−1	4.59e−1	3.87e−1	3.32e−1
F33	AVG	**1.18e+4**	1.22e+10	2.39e+11	1.72e+8	7.28e+9	1.20e+11	1.91e+9	2.14e+7	1.04e+5	1.03e+8	4.14e+7
	STD	**5.00e+4**	1.04e+11	3.80e+11	6.78e+7	5.35e+9	1.80e+11	1.42e+9	8.01e+7	9.02e+4	1.47e+8	6.84e+7
F34	AVG	**1.73e+1**	2.37e+2	4.52e+3	**1.73e+1**	**1.73e+1**	4.69e+2	**1.73e+1**	**1.73e+1**	**1.73e+1**	**1.73e+1**	**1.73e+1**
	STD	9.18e−5	1.25e+3	3.68e+3	9.33e−5	8.20e−3	1.41e+3	3.29e−8	**7.50e−14**	2.63e−4	2.11e−4	6.66e−9
F35	AVG	**1.27e+1**	**1.27e+1**	1.28e+1	**1.27e+1**	**1.27e+1**	1.29e+1	1.28e+1	**1.27e+1**	**1.27e+1**	**1.27e+1**	**1.27e+1**
	STD	**2.35e−15**	9.46e−4	1.04e+0	3.61e−15	9.65e−15	3.00e−4	1.57e−6	5.09e−8	4.38e−5	9.86e−8	3.69e−15
F36	AVG	**1.10e+1**	1.04e+4	4.96e+3	1.69e+1	4.84e+1	1.62e+3	4.20e+1	1.13e+1	2.79e+1	4.04e+1	2.92e+1
	STD	7.32e+0	9.46e+2	5.77e+3	**5.76e−7**	3.91e+1	3.47e+3	7.04e+1	5.66e+0	1.43e+1	7.27e+0	1.39e+1
F37	AVG	**1.04e+0**	3.51e+0	2.94e+0	1.41e+0	1.19e+0	2.11e+0	1.14e+0	1.06e+0	1.29e+0	1.52e+0	1.16e+0
	STD	**2.60e−11**	1.23e−1	1.26e+0	4.87e−11	8.09e−2	6.06e−1	8.10e−2	3.77e−2	1.25e−1	1.28e−1	8.24e−2
F38	AVG	**2.53e+0**	9.31e+0	1.18e+0	2.60e+0	3.16e+0	1.12e+1	8.04e+0	9.16e+0	5.97e+0	9.32e+0	6.91e+0
	STD	5.46e−1	1.17e−1	1.09e+0	**2.80e−2**	9.81e−1	1.14e+0	2.57e+0	4.23e−1	9.08e−1	9.25e−2	8.43e−1
F39	AVG	**1.34e+2**	1.19e+3	1.01e+3	7.91e+2	3.08e+2	6.63e+2	1.36e+2	1.44e+2	1.36e+2	5.09e+2	2.10e+2
	STD	**1.32e−2**	2.03e+2	2.63e+2	1.40e−2	2.53e+2	3.25e+2	1.36e+2	1.59e+2	2.08e+2	1.44e+2	1.72e+2
F40	AVG	**2.33e+0**	1.38e+3	1.56e+3	2.35e+0	2.47e+0	4.01e+2	2.37e+0	2.37e+0	2.55e+0	2.46e+0	2.37e+0
	STD	4.05e−1	3.07e+2	1.69e+3	2.93e−2	8.96e−2	6.45e+2	1.24e−2	**1.53e−2**	1.73e−1	6.22e−2	1.84e−2
F41	AVG	**2.03e+1**	2.06e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1	2.04e+1
	STD	7.16e−2	1.08e−1	1.68e+0	**1.73e−9**	7.81e−2	1.13e−1	1.03e−1	1.19e−1	3.65e+0	5.83e+0	1.25e−1

Table 26
Results of benchmark functions (F1–F13) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
$\alpha_1 = 0.5$	F1	AVG	**3.14e−2**	1.50e−1	1.79e−1
		STD	**9.08e−2**	5.41e−1	6.78e−1
	F2	AVG	**2.30e−2**	3.02e−2	3.60e−2
		STD	5.07e−2	9.70e−2	**1.96e−2**
	F3	AVG	2.28e−1	3.81e−1	2.64e−1
		STD	9.11e−1	2.08e+0	5.23e−1
	F4	AVG	2.95e−1	2.40e−1	1.84e−1
		STD	3.32e−1	4.69e−1	1.33e−1
	F5	AVG	2.91e−1	**1.21e−1**	2.24e−1
		STD	**1.61e−1**	5.41e−1	9.44e−1
	F6	AVG	3.05e−1	**1.08e−1**	1.25e−1
		STD	1.68e+0	3.03e−1	**2.82e−1**
	F7	AVG	**6.68e−4**	7.93e−4	1.80e−3
		STD	**1.90e−3**	2.50e−3	2.20e−3
	F8	AVG	**−8.92e+2**	−8.37e+2	−1.08e+2
		STD	5.73e+2	**6.41e+0**	6.15e+1
	F9	AVG	8.06e−2	1.05e−1	1.52e−1
		STD	**1.45e−1**	4.14e−1	2.60e−1
	F10	AVG	2.83e−1	4.34e−1	4.93e−1
		STD	8.20e−1	7.67e−1	5.97e−1
	F11	AVG	1.03e−1	7.01e−2	1.37e−1
		STD	1.10e−1	1.46e−1	2.09e−1
	F12	AVG	1.06e−1	**2.62e−2**	9.63e−2
		STD	4.57e−1	**6.37e−2**	4.92e−1
	F13	AVG	2.34e−2	**1.08e−2**	1.17e−2
		STD	5.53e−2	2.23e−2	2.66e−2

Table 27
Results of benchmark functions (F1–F13) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
$\alpha_1 = 1$	F1	AVG	7.89e−2	7.06e−2	2.59e−1
		STD	3.34e−1	9.21e−2	9.95e−1
	F2	AVG	5.86e−2	3.84e−2	3.16e−2
		STD	1.12e−1	5.06e−2	6.70e−2
	F3	AVG	3.70e−1	**1.18e−1**	1.22e−1
		STD	1.14e+0	4.51e−1	**3.80e−1**
	F4	AVG	2.18e−1	2.43e−1	2.42e−1
		STD	2.48e−1	3.55e−1	1.97e−1
	F5	AVG	5.15e−1	5.63e−1	5.98e−1
		STD	1.51e+0	3.34e+0	2.97e+0
	F6	AVG	6.18e−1	1.13e+0	6.17e−1
		STD	6.87e+0	5.05e+0	4.47e+0
	F7	AVG	2.00e−3	1.70e−3	2.10e−3
		STD	**1.90e−3**	4.20e−3	2.50e−3
	F8	AVG	−1.92e+2	−8.36e+2	−4.24e+2
		STD	1.10e+2	1.17e+1	2.64e+2
	F9	AVG	1.09e−1	1.30e−1	**6.71e−2**
		STD	3.57e−1	3.84e−1	1.64e−1
	F10	AVG	2.82e−1	4.96e−1	**1.79e−1**
		STD	5.79e−1	**3.62e−1**	6.15e−1
	F11	AVG	**4.76e−2**	8.28e−2	1.37e−1
		STD	**8.55e−2**	8.63e−2	1.53e−1
	F12	AVG	7.89e−2	9.77e−2	4.73e−2
		STD	3.54e−1	6.912e−1	1.80e−1
	F13	AVG	2.24e−2	2.12e−2	2.39e−2
		STD	4.13e−2	**1.44e−2**	4.58e−2

Table 28
Results of benchmark functions (F14–F31) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
	F14	AVG	1.03e+0	1.06e+0	1.18e+0
		STD	1.79e−1	2.36e−1	5.11e−1
	F15	AVG	2.90e−3	2.30e−3	1.70e−3
		STD	1.80e−3	2.10e−3	1.60e−3
	F16	AVG	**−1.03e+0**	−1.03e+0	−1.03e+0
		STD	2.61e−2	2.98e−2	2.90e−2
	F17	AVG	4.14e−1	4.01e−1	3.99e−1
		STD	2.86e−2	1.18e−2	2.50e−3
	F18	AVG	3.42e+0	3.78e+0	3.63e+0
		STD	2.00e+0	3.17e+0	3.00+0
	F19	AVG	1.10e−3	**7.15e−4**	6.10e−3
		STD	**0.00e+0**	5.50e−3	1.36e−2
	F20	AVG	2.09e−1	**2.65e−2**	2.97e−1
		STD	7.47e−2	**3.90e−16**	8.03e−2
	F21	AVG	−6.07e+0	−5.05e+0	**−6.39e+0**
		STD	1.73+0	1.20e+0	1.09e+0
	F22	AVG	**−7.51e+0**	−6.10e+0	−5.17e+0
$\alpha_1 = 0.5$		STD	1.66e+0	1.76e+0	1.34e+0
	F23	AVG	**−7.11e+0**	−5.43e+0	−3.90e+0
		STD	1.04e+0	1.22e+0	5.92e−1
	F24	AVG	4.83e+2	1.65e+3	6.70e+2
		STD	2.36e+3	1.06e+4	1.52e+3
	F25	AVG	1.68e+0	8.59e−1	**7.77e−1**
		STD	9.53e+0	**1.47e+0**	3.59e+0
	F26	AVG	6.52e+0	7.89e+0	1.83e+1
		STD	1.12e+1	5.07e+1	1.41e+2
	F27	AVG	2.99e+0	1.92e+1	**2.55e+0**
		STD	7.07e+0	**5.10e+0**	5.57e+0
	F28	AVG	**5.08e+0**	8.02e+0	7.06e+0
		STD	4.03e+0	3.66e+0	3.45e+0
	F29	AVG	**7.98e−2**	1.1e−1	1.33e−1
		STD	3.44e−1	2.99e−1	**2.31e−1**
	F30	AVG	**1.10−1**	2.10e−1	1.72e−1
		STD	**1.77e−2**	1.27e−1	6.86e−2
	F31	AVG	**1.70e−3**	1.80e−3	9.00e−3
		STD	**1.77e−2**	2.06e−2	2.42e−2

Table 29
Results of benchmark functions (F14–F31) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
$\alpha_1 = 1$	F14	AVG	1.11e+0	1.12e+0	**1.03e+0**
		STD	8.54e−1	5.57e−1	**2.30e−1**
	F15	AVG	1.40e−3	**1.30e−3**	1.70e−3
		STD	3.70e−3	1.70e−3	**1.20e−3**
	F16	AVG	−1.03e+0	−1.03e+0	−1.03e+0
		STD	**2.44e−2**	3.21e−2	2.59e−2
	F17	AVG	4.05e−1	4.02e−1	**3.98e−1**
		STD	4.62e−2	2.14e−2	**8.93e−5**
	F18	AVG	**3.18e+0**	3.93e+0	7.42e+0
		STD	**5.38e−1**	5.32e+0	2.98e+1
	F19	AVG	5.70e−3	4.15e−2	2.40e−3
		STD	1.05e−2	8.70e−3	1.48e−2
	F20	AVG	2.19e−1	2.93e−1	8.53e−2
		STD	2.09e−2	1.58e−1	1.13e−1
	F21	AVG	−5.35e+0	−5.37e+0	−4.70e+0
		STD	1.04e+0	**8.44e−1**	1.59e+0
	F22	AVG	−6.16e+0	−6.58e+0	−3.73e+0
		STD	1.76e+0	1.25e+0	**7.43e−1**
	F23	AVG	−4.10e+0	−4.73e+0	−4.80e+0
		STD	4.33e−1	**3.01e−1**	8.80e−1
	F24	AVG	1.10e+3	5.70e+2	**4.34e+2**
		STD	3.55e+3	5.15e+3	**5.77e+2**
	F25	AVG	1.09e+0	8.16e−1	1.09e+0
		STD	6.87e+0	5.84e+0	5.01e+0
	F26	AVG	**3.10e+0**	4.76e+0	7.46e+0
		STD	**5.38e+0**	1.59e+1	4.06e+1
	F27	AVG	2.87e+1	1.21e+1	3.83e+0
		STD	4.84e+1	6.43e+0	9.09e+0
	F28	AVG	5.79e+0	7.44e+0	6.11e+0
		STD	1.95e+0	**1.94e+0**	2.54e+0
	F29	AVG	8.01e−2	4.82e−1	8.27e−2
		STD	3.69e−1	5.54e−1	1.77e−1
	F30	AVG	1.70e−1	1.17e−1	1.34e−1
		STD	8.66e−2	8.44e−2	1.32e−1
	F31	AVG	1.51e−2	3.60e−3	1.41e−2
		STD	2.50e−2	1.80e−2	4.55e−2

Table 30
Results of benchmark functions (F32–F41) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
$\alpha_1 = 0.5$	F32	AVG	3.01e−2	3.33e−2	3.76e−2
		STD	4.86e−2	3.95e−2	5.18e−2
	F33	AVG	**1.09e+4**	1.36e+4	1.97e+4
		STD	**5.38e+5**	6.92e+5	6.92e+5
	F34	AVG	3.42e+1	4.34e+1	4.10e+1
		STD	6.62e−4	1.41e−1	1.40e−1
	F35	AVG	**1.27e+1**	**1.27e+1**	**1.27e+1**
		STD	3.00e−4	2.20e−6	2.58e−8
	F36	AVG	6.67e+1	7.37e+1	9.21e+1
		STD	1.43e+1	1.62e+1	2.11e+1
	F37	AVG	3.21e+0	3.67e+0	3.46e+0
		STD	3.41e−1	2.93e−1	**2.25e−1**
	F38	AVG	**9.60e+0**	1.04e+1	1.03e+1
		STD	8.30e−1	5.10e−1	4.28e−1
	F39	AVG	7.12e+2	8.09e+2	7.24e+2
		STD	1.47e+2	1.52e+2	1.57e+2
	F40	AVG	1.36e+1	2.05e+0	2.96e+0
		STD	4.40e+1	5.72e−1	**3.38e−1**
	F41	AVG	2.03e+1	2.03e+1	2.04e+1
		STD	4.86e−2	**7.88e−3**	6.39e−2

Table 31
Results of benchmark functions (F32–F41) for different values of α_1 and α_2 (Best results are shown in Bold).

	Functions	$\alpha_2 = 0.5$		$\alpha_2 = 1$	$\alpha_2 = 2$
$\alpha_1 = 1$	F32	AVG	3.57e−2	**1.19e−2**	3.70e−2
		STD	4.41e−2	**1.43e−2**	7.29e−2
	F33	AVG	2.23e+4	1.86e+4	1.68e+4
		STD	9.37e+5	8.47e+5	6.81e+5
	F34	AVG	3.92e+1	4.39e+1	**1.72e+1**
		STD	1.48e−1	1.30e−1	**1.18e−6**
	F35	AVG	**1.27e+1**	**1.27e+1**	**1.27e+1**
		STD	2.30e−9	2.25e−10	**3.13e−15**
	F36	AVG	8.18e+1	9.32e+1	**1.43e+1**
		STD	**1.51e+0**	2.45e+0	1.48e+1
	F37	AVG	3.68e+0	3.64e+0	**1.04e+0**
		STD	4.02e−1	3.41e−1	2.52e−1
	F38	AVG	1.05e+1	1.00e+1	1.03e+1
		STD	**2.98e−1**	7.96e−1	5.15e−1
	F39	AVG	7.06e+2	6.85e+2	**1.62e+2**
		STD	7.01e+1	1.56e+2	**1.51e+0**
	F40	AVG	2.47e+0	2.68e+0	**1.98e+0**
		STD	4.86e−1	6.35e−1	4.80e−1
	F41	AVG	2.04e+1	2.04e+1	**1.79e+1**
		STD	5.19e−2	1.25e−1	8.62e−2

References

[1] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, Int. J. Intell. Syst. 36 (10) (2021) http://dx.doi.org/10.1002/int.22535.

[2] B. Al-Khateeb, K. Ahmed, M. Mahmood, D.N. Le, Rock hyraxes swarm optimization: A new nature-inspired metaheuristic optimization algorithm, Comput. Mater. Contin. 68 (1) (2021) http://dx.doi.org/10.32604/cmc.2021.013648.

[3] W.J. Al-Kubaisy, M. Yousif, B. Al-Khateeb, M. Mahmood, D. Le, The red colobuses monkey: A new nature–inspired metaheuristic optimization algorithm, Int. J. Comput. Intell. Syst. 14 (1) (2021) http://dx.doi.org/10.2991/IJCIS.D.210301.004.

[4] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007.
[5] M. Braik, M.H. Ryalat, H. Al-Zoubi, A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves, Neural Comput. Appl. 34 (1) (2022) http://dx.doi.org/10.1007/s00521-021-06392-x.
[6] M. Braik, A. Sheta, H. Al-Hiary, A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm, Neural Comput. Appl. 33 (7) (2021) http://dx.doi.org/10.1007/s00521-020-05145-6.
[7] Y.H. Chou, S.Y. Kuo, L.S. Yang, C.Y. Yang, Next generation metaheuristic: Jaguar algorithm, IEEE Access 6 (2018) http://dx.doi.org/10.1109/ACCESS.2018.2797059.
[8] M. Cikan, B. Kekezoglu, Comparison of metaheuristic optimization techniques including equilibrium optimizer algorithm in power distribution network reconfiguration, Alexandria Eng. J. 61 (2) (2022) http://dx.doi.org/10.1016/j.aej.2021.06.079.
[9] E. Cuevas, M. Cienfuegos, D. Zaldívar, M. Pérez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Syst. Appl. 40 (16) (2013) http://dx.doi.org/10.1016/j.eswa.2013.05.041.
[10] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) http://dx.doi.org/10.1109/TEVC.2010.2059031.
[11] M. Dehghani, Š. Hubálovský, P. Trojovský, Cat and mouse based optimizer: A new nature-inspired optimization algorithm, Sensors 21 (15) (2021) http://dx.doi.org/10.3390/s21155214.
[12] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, Theoret. Comput. Sci. 344 (2–3) (2005) http://dx.doi.org/10.1016/j.tcs.2005.05.020.
[13] A.E. Ezugwu, D. Prayogo, Symbiotic organisms search algorithm: theory, recent advances and applications, Expert Syst. Appl. (2019) http://dx.doi.org/10.1016/j.eswa.2018.10.045.
[14] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) http://dx.doi.org/10.1016/j.knosys.2019.105190.
[15] J. García-Algarra, J. Galeano, J.M. Pastor, J.M. Iriondo, J.J. Ramasco, Rethinking the logistic approach for population dynamics of mutualistic interactions, J. Theoret. Biol. 363 (2014) http://dx.doi.org/10.1016/j.jtbi.2014.08.039.
[16] C.K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) http://dx.doi.org/10.1109/TEVC.2008.920671.
[17] D.M. Gordon, The ecology of collective behavior in ants, Annu. Rev. Entomol. (2019) http://dx.doi.org/10.1146/annurev-ento-011118-111923.
[18] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: New metaheuristic algorithm for solving optimization problems, Math. Comput. Simulation 192 (2022) http://dx.doi.org/10.1016/j.matcom.2021.08.013.
[19] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) http://dx.doi.org/10.1016/j.future.2019.02.028.
[20] A. Kaveh, Advances in Metaheuristic Algorithms for Optimal Design of Structures, second ed., 2016.
[21] Y.-H. Kuo, Wu Shu-Yu, Chen Ching-Hsuan, Chou Cheng-Chun, A novel metaheuristic: Fast jaguar algorithm, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022.
[22] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) http://dx.doi.org/10.1016/j.future.2020.03.055.
[23] N. Li, L. Wang, Bare-bones based Sine cosine algorithm for global optimization, J. Comput. Sci. 47 (2020) http://dx.doi.org/10.1016/j.jocs.2020.101219.
[24] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer with local search, in: 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings, Vol. 1, 2005.
[25] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, Z. Zhu, A survey on cooperative co-evolutionary algorithms, IEEE Trans. Evol. Comput. 23 (3) (2019) http://dx.doi.org/10.1109/TEVC.2018.2868770.
[26] Z. Meng, G. Li, X. Wang, S.M. Sait, A.R. Yıldız, A comparative study of metaheuristic algorithms for reliability-based design optimization problems, Arch. Comput. Methods Eng. 28 (3) (2021) http://dx.doi.org/10.1007/s11831-020-09443-z.
[27] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) http://dx.doi.org/10.1016/j.advengsoft.2015.01.010.
[28] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.
[29] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.
[30] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Syst. Appl. 166 (2021) http://dx.doi.org/10.1016/j.eswa.2020.113917.
[31] R. Nafidi, El Azri Ahmed, Gutiérrez Sánchez Abdenbi, The stochastic modified lundqvist-korf diffusion process: statistical and computational aspects and application to modeling of the CO2 emission in Morocco, Stoch. Environ. Res. Risk Assess. 36 (2022) 1163–1176, http://dx.doi.org/10.1007/s00477-021-02089-8.
[32] B. Niu, Y. Zhu, X. He, H. Wu, MCPSO: A multi-swarm cooperative particle swarm optimizer, Appl. Math. Comput. 185 (2) (2007) http://dx.doi.org/10.1016/j.amc.2006.07.026.
[33] T.H. Oliver, The ecology and evolution of ant-aphid interactions thesis, 2008.
[34] A. Osaba, Del Ser Eneko, Martinez Javier, Aritz D. Hussain, Evolutionary multitask optimization: a methodological overview, challenges, and future research directions, Cognit. Comput. (2022) http://dx.doi.org/10.1007/s12559-022-10012-8.
[35] K. Rajendran, N. Shankar, Čep Ganesh, R.C. Robert, Pal Narayanan, Kalita Subham, A conceptual comparison of six nature-inspired metaheuristic algorithms in process optimization, Processes 10 (197) (2022) 1–20.
[36] A. Rezaee Jordehi, J. Jasni, Parameter selection in particle swarm optimisation: A survey, J. Exp. Theor. Artif. Intell. 25 (4) (2013) http://dx.doi.org/10.1080/0952813X.2013.782348.

[37] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) http://dx.doi.org/10.1016/j.advengsoft.2017.01.004.

[38] A.S. Shamsaldin, T.A. Rashid, R.A. Al-Rashid Agha, N.K. Al-Salihi, M. Mohammadi, Donkey and smuggler optimization algorithm: A collaborative working approach to path finding, J. Comput. Des. Eng. 6 (4) (2019) http://dx.doi.org/10.1016/j.jcde.2019.04.004.

[39] B. Stadler, A.F.G. Dixon, Ecology and evolution of aphid-ant interactions, Annu. Rev. Ecol. Evol. Syst. (2005) http://dx.doi.org/10.1146/annurev.ecolsys.36.091704.175531.

[40] J.E. Stewart, The direction of evolution: The rise of cooperative organization, BioSystems 123 (2014) http://dx.doi.org/10.1016/j.biosystems.2014.05.006.

[41] S. Sumathi, T. Hamsapriya, P. Surekha, Evolutionary intelligence: An introduction to theory nd applications with matlab, 2008.

[42] T. Tashtoush, J. Ahmed, V. Arce, H. Dominguez, K. Estrada, W. Montes, A. Paredez, P. Salce, A. Serna, M. Zarazua, Developing a radiating L-shaped search algorithm for NASA swarm robots, Int. J. Adv. Comput. Sci. Appl. 11 (8) (2020) http://dx.doi.org/10.14569/IJACSA.2020.0110802.

[43] M. Trojovský, Dehghani Pavel, Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications, Sensors 22 (855) (2022) 1–34.

[44] I. Yao, Costs and constraints in aphid-ant mutualism, Ecol. Res. 29 (3) (2014) http://dx.doi.org/10.1007/s11284-014-1151-4.

[45] Z. Zhang, C. Huang, K. Dong, H. Huang, Birds foraging search: a novel population-based algorithm for global optimization, Memet. Comput. 11 (3) (2019) http://dx.doi.org/10.1007/s12293-019-00286-1.

[46] H.W. Zhao, L.W. Tian, Cooperative artificial fish swarm optimization, Appl. Mech. Mater. 741 (2015) http://dx.doi.org/10.4028/www.scientific.net/amm.741.359.