Farmer Ants Optimization Algorithm: A new meta-heuristic for solving discrete optimization problems

Mahdi Zeinalabedinmalekmian
Shafagh Institute of Higher Education
Ali Asghari
Asghari_ali@aut.ac.ir

Shafagh Institute of Higher Education
Hossein Azgomi
Islamic Azad University
Mahmoud Alimoradi
Shafagh Institute of Higher Education
Shirin Ghaziantafrishi
Ministry of Education

Research Article

Keywords: meta-heuristic algorithms, farmer ants optimization algorithm, optimization, metaheuristic algorithms

Posted Date: April 2nd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4175717/v1

Additional Declarations: No competing interests reported.

Farmer Ants Optimization Algorithm: A new meta-heuristic for solving discrete optimization problems

Mahdi Zeinalabedinmalekmian[a], Ali Asghari[a,*], Hossein Azgomi[b], Mahmoud Alimoradi[a]
Shirin Ghaziantafrishi[c]

[a] Department of Computer Engineering, Shafagh Institute of Higher Education, Tonekabon, Iran
[b] Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
[c] Ministry of Education, Mashhad, Iran

Kulaad@shafagh.ac.ir

Asghari_ali@aut.ac.ir

Azgomi@iaurasht.ac.ir

Mahmoud.alimoradi@shafagh.ac.ir

shiringhazian@gmail.com

Abstract:

Today, some complex problems are known as NP-hard problems. For this category of problems, there is no exact solution or they are not solvable in a reasonable time. For this reason, metaheuristic algorithms have been introduced and developed. These algorithms attempt to find an optimal solution to the problem instead of finding a definite solution. In recent years, these algorithms have gained significant attention from researchers. The major inspiration for metaheuristic algorithms is nature and its laws. An important category of these algorithms is evolutionary algorithms. These algorithms are inspired by the behavior of animals and living organisms that exhibit social and intelligent behavior. However, each metaheuristic algorithm may optimally solve just some types of problems. Therefore, researchers continuously try to introduce new algorithms. In this study, a new metaheuristic algorithm called Farmer Ants Optimization Algorithm (FAOA) is introduced. This algorithm is based on the intelligent life of farmer ants. Farmer ants cultivate mushrooms to provide food for themselves. They also protect them against various pests, and after growth, feed them. These special behaviors of farmer ants, which are based on their social life, are the source of inspiration for the proposed method. Experiments on some engineering and classical problems have shown that FAOA can provide an acceptable solution for discrete optimization problems.

Keywords: meta-heuristic algorithms, farmer ants optimization algorithm, optimization, metaheuristic algorithms

1. Introduction

In the new era, there are problems with different complexity in calculations. Some of these problems are known as non-deterministic polynomial hardness (NP-hard) problems [1]. Np-hard problems include thousands of problems, each of which has many applications in engineering sciences and a definitive solution has not yet been found to solve them [2]. In such problems, it is very unlikely to find a solution with definite polynomial time [3]. With the increase in the amount of data, finding solutions to these issues is more challenging [4]. Therefore, optimization methods are considered an effective solution in this field. Optimization is finding the optimal solution for the parameters of a given system from all values to maximize or minimize the output. Optimization problems can be found in most engineering fields [5]. Because of the drawbacks of some conventional techniques, the possibility of falling into local optima, and the need to expand the search space [6], optimization techniques [7] have been developed over the last two decades [8,9]. With the increasing complexity of the problems, the need for optimization methods is felt more than before [10]. Optimization plays a very important role in industry, the development of science, management, and solving problems that can be modeled in this field. In multidimensional, discontinuous models and data containing noise, which cannot be solved by traditional methods, optimization algorithms can be used as an alternative [11].

Real-world problems in machine learning and artificial intelligence are generally continuous, discrete, bounded, or unbounded [12,13]. Because of these features, it is difficult to find an exact solution for some classes of problems using conventional mathematical methods [14,15]. Several studies have confirmed that these methods are not efficient enough to solve some problems [16]. Meta-heuristic algorithms are an alternative solution to solve such problems. These algorithms are usually inspired by intelligent concepts such as physical rules, social phenomena, animal behavior, and evolution [17].

With the rapid growth of science and industry and the emergence of recent issues, metaheuristic algorithms are deployed more and more [18,19]. However, a specific meta-heuristic algorithm cannot solve all problems. On the other hand, the major challenge of meta-heuristic algorithms is obtaining the solution in the shortest possible time with the highest accuracy. Some algorithms are highly accurate in solving some problems, but their response time is longer than similar algorithms. For this reason, algorithms should approach an acceptable point in terms of accuracy and speed [20]. Today, meta-heuristic algorithms have attracted the attention of many scientists. For example, the Genetic Algorithm (GA) [21] uses the theory of evolution to solve discrete problems. Another algorithm in this field is called the Artificial Immune System [22]. This algorithm is inspired by the principles and processes of the vertebrate immune system and is modeled based on the learning and memory characteristics of the immune system. Ant Colony Optimization (ACO) [23] is one of the most prominent meta-heuristic algorithms. This algorithm is inspired by the search behavior of ants to find food. Particle Swarm Optimization (PSO) [24] is another well-known algorithm inspired by the social movement of birds. The Flower Pollination Algorithm (FPA) [25] is inspired by the growth of plants and the pollination process. Another algorithm in this field is the Firefly Algorithm [26]. This algorithm is based on the blinking of fireflies. The Trees Social Relations Optimization Algorithm (TSR) [27] is one of the new algorithms in this field, which is inspired by the hierarchical and social life of trees. This algorithm can solve discrete and continuous problems. The Water Optimization Algorithm (WOA) [20] is another new algorithm that is inspired by the chemical features of water molecules. The Lion Optimization Algorithm (LOA) [28] is based on the life of lions and their individual and social behavior.

Some meta-heuristic algorithms are inspired by modeling world natural features and adaptability to the environment [29]. Humankind has been using nature's guidance to solve many problems for many years. In recent decades, many efforts have been conducted to develop algorithms derived from nature [30]. Evolutionary algorithms can not optimally solve all problems, and each of them is suitable for solving a certain group of problems. For this reason, researchers try to find new algorithms. These algorithms are used in solving engineering problems and other fields, especially challenging problems [31-33]. Nature-inspired computing has attracted computer scientists for a long time, and popular fields such as neural networks [34], cellular automata [35], molecular computing [36], and evolutionary algorithms have been created [37] [38].

In this paper, a new meta-heuristic algorithm called Farmer Ants Optimization Algorithm (FAOA) is introduced. This algorithm is inspired by the social life of farmer ants to solve discrete NP-Hard problems. Farmer ants have been farming and growing mushrooms for millions of years [39]. They use their products to feed the colony [40]. These ants cultivate a special type of mushrooms for their food. They plant the seeds of these mushrooms in special chambers, feed them, and prevent them from rotting by producing a chemical substance or taking care of them against alien attacks. Knowledge sharing and concurrent behavior are the principal features of the FAOA.

The contribution of the proposed method includes the following:

• Presenting the new Optimization Algorithm based on the life of farmer ants

• The concurrency feature of the algorithm that can solve complex problems with high accuracy and fast speed

• The ability to share local knowledge to reach the best global solution

• Ability to solve discrete NP-hard problems

• Evaluation of the proposed method and comparing it with some state-of-the-art algorithms in solving engineering problems

Other parts of the paper are organized as follows. In section 2, a review of related works has been conducted. Part 3 is related to investigating the life of farmer ants and introducing a new algorithm based on their inspiration. Section 4 is assigned to the evaluation of the proposed method. And finally, the paper is concluded in Section 5.

2. Related works

NP-hard problems are those for which no quick and specific solution has been found. Over the last decade, these issues have become more complex [41]. Metaheuristic algorithms have been developed to address these challenges. These algorithms seek efficient and effective solutions to such problems. Metaheuristic algorithms can be categorized into swarm-based, physic-based, evolutionary-based, and nature-based.

The first category is known as Swarm Based Algorithms. These algorithms use the process of social intelligence and the particles communicate with each other to share their experiences [42]. Intelligent Water Drops (IWD) algorithm is one of these algorithms. This algorithm consists of two distributed memory parts in which the former plays the role of soil edges, and the latter includes smart drops. This algorithm is used to solve continuous problems [43]. One of the most important and widely used algorithms in this group is called Ant Colony Optimization (ACO). This algorithm is inspired by the social behavior of ants to reach the food source [23]. Another commonly used algorithm in this category is the Particle Swarm Optimization (PSO) algorithm. This algorithm is inspired by the social behavior of bird flocks. In PSO, particles move with an initial velocity in the search space. Each particle chooses its next location according to the local and best global experiences, respectively [24]. The Sine Cosine Algorithm (SCA), which is another member of this group, starts with random solutions. Then, using a mathematical model based on sine and cosine functions, each particle moves towards the best solution. Also, several random and adaptive variables are integrated into the search space to help exploration behavior [5]. Other algorithms in this category include the Artificial Fish Swarm Optimization Algorithm (AFSA) [44], the Human Mental Search Algorithm (HMS) [45], and the Trees Social Relation Optimization Algorithm (TSR) [27].

The second category is physics-based algorithms. These algorithms are based on physical phenomena such as gravity, electromagnetism, and temperature. Ray Optimization (RO) algorithm is one of these algorithms.

This algorithm tries to achieve the optimal solution by modeling the transmission of rays from one point to another by using Snell's law of light refraction [46]. Gravitational Search Algorithm (GSA) is another algorithm that originated from physics. This algorithm is based on Newton's law of gravity. According to this law, particles attract each other. The attraction force is proportional to the product of their masses and the square of the distance between them [47]. Archimedes Optimization Algorithm is inspired by Archimedes's principle in physics. According to this principle, when an object sinks into a liquid or gas, the upward buoyancy force is applied to that object, which is equal to the weight of the liquid or gas displaced by the object sinking. [48]. One of the most famous algorithms in this category is the Simulated Annealing (SA) algorithm. This algorithm was inspired by modeling the gradual cooling of molten metal. Because of its high efficiency and simplicity, the SA algorithm is used for searching in large spaces and also for solving discrete problems [49]. Other algorithms that fall into this category are the Charged System Search (CCS) [50], the Memetic Algorithm [51], the Electromagnetism algorithm [52], and The Water Optimization Algorithm (WOA) [20].

Evolutionary Algorithms are inspired by Darwin's theory and based on the reproduction of living organisms. [41]. Genetic Algorithm is the best-known algorithm in this category. This algorithm consists of two major components, chromosome and gene, and two primary operations include crossover and mutation [21]. Plant Growth Optimization (PGO) is another algorithm of this category that is based on plant growth, branching, phototropism, and leaf growth. The major goal of the algorithm is to select the active point by comparing the concentration of morphogen to reach the appropriate solution [53]. another algorithm of this category is called Saplings Growing Up Algorithm (SGA). This algorithm is inspired by the seedling growth process and consists of two phases: planting and growth. The SGA algorithm consists of mating, branching, and vaccination operators that create new candidate solutions by limiting the search space [54]. Other algorithms in this category include the Photosynthetic Algorithm (PA) [55], Differential Evolution Algorithm (DE) [56], Improved Unified Differential Evolution (IUDE) [57], and genetic programming algorithm [58].

The fourth category is called Nature Inspired Algorithms. These algorithms are modeled based on the environment and social life of animals. The Gray Wolf Optimizer (GWO) algorithm is one of these algorithms. This algorithm, which has a hierarchical structure, is inspired by a group of hunting wolves. The GWO algorithm consists of four categories: alpha, beta, delta, and omega in a wolf pack [17]. Another algorithm in this category is called The Whale Optimization Algorithm (WOA). This algorithm is inspired by the way humpback whales hunt and the bubble hunting strategy. [59]. The Red Deer Algorithm (RDA) deals with the behavior of Scottish deers during mating. Males roar and fight to reach their desired mate. The best deer is called a commander and can form its harem [60]. The moth-Flame Optimization Algorithm (MFO) is inspired by the movement of moths. Moths move at night, keeping a constant angle to the moonlight. This mechanism is very effective in long movements on a fixed route. [10]. Other algorithms in this category include the Firefly Algorithm (FF) [26], the Lion Optimization Algorithm (LOA) [28], the Harris Hawks Optimization Algorithm (HHO) [61], and Bat-Inspired Algorithms (BA) [62].

Figure 1 shows the classification of the reviewed algorithms.

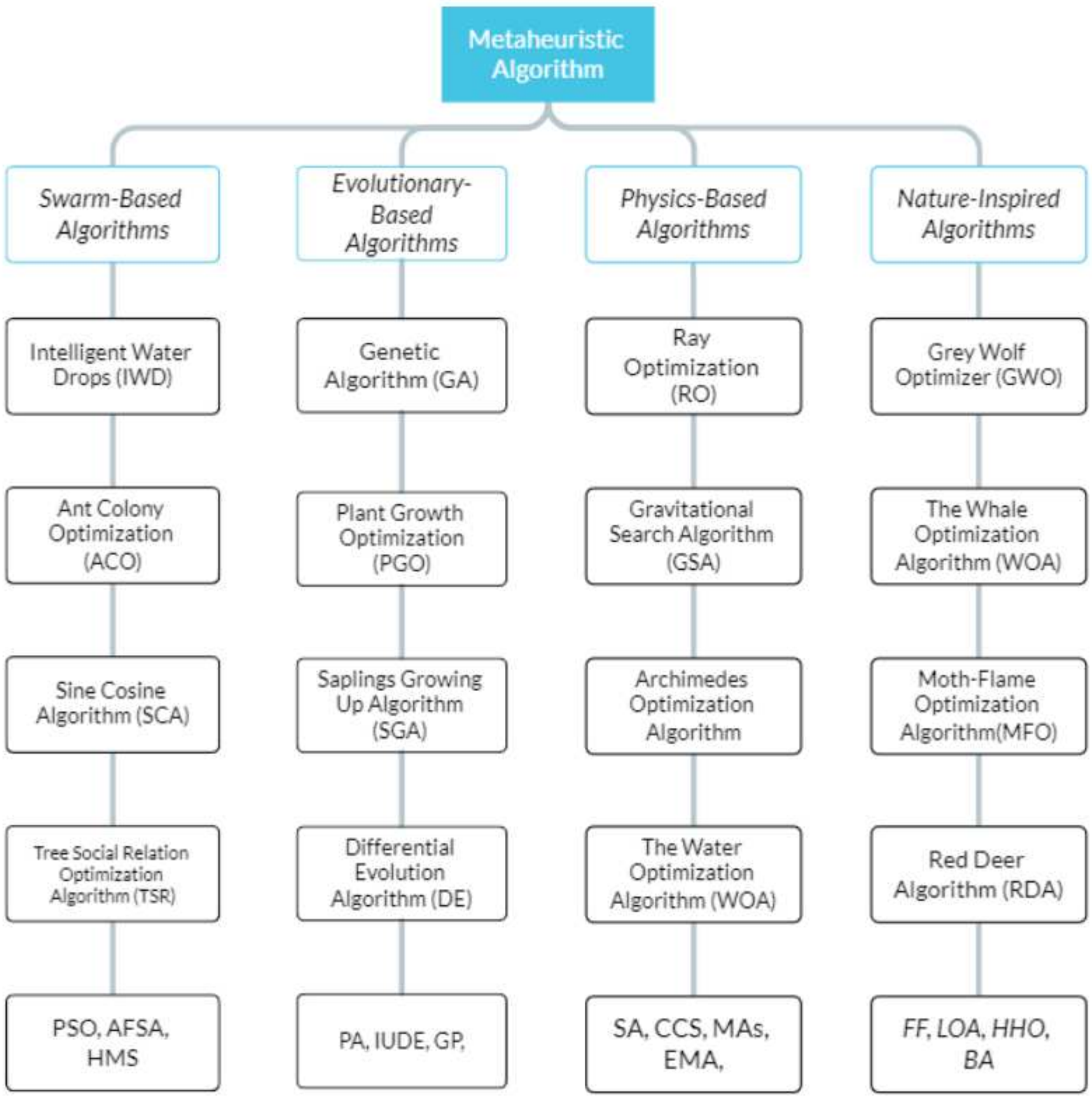

Fig.1 Metaheuristic algorithms classification

3. The proposed method

In this section, the life of the farmer ants is explained first, and then the basics of the proposed algorithm are explained.

3.1 Inspiration

Over the past thousands of years, humans started farming and have domesticated more than 260 plant species, 470 animal species, and 100 mushroom species[63]. However, farmer ants with more than 60 million years of agriculture are known as the first farmers of the world [39]. Attine ants are the first farmers in the world [64]. These ants grow mushrooms [40]. Scientists believe that ants started farming in the humid forests and tropical regions of South America 55 to 60 million years ago [65,66]. A colony of farmer ants is usually formed by a mated female. These colonies are made up of one or more queens. Some of the unmated queens after a while lose their wings and start farming. These ants grow their mushrooms in underground chambers and fertilize gardens with vegetable scraps and dead insects. Farmer ants are compulsively dependent on their mushrooms. Their babies are raised on an exclusively mushroom diet. Farmer ants live in a complex and highly specialized multi-trophic symbiosis. Ants get food for the colony by cultivating a special type of mushroom in their nest, and in return, these ants provide the food needed for the mushrooms and a suitable environment for their growth and cultivation free of parasites. Some leaf-cutter ants use fresh leaves to prepare a suitable environment for the growth of mushrooms [67-71]. Some colonies are smaller and consist of fewer rooms containing hundreds to thousands of ants. Other colonies are made up of populations of up to 8 million ants, with many more rooms and multiple entrances. Agricultural processes are performed by all ants of the colony. The workers search around the nest to find food for mushrooms. These food items can be divided into different parts. They use fresh leaves of trees, flowers, and fruits in humid seasons, and seeds and carcasses of arthropods and insects in cold and dry seasons. Smaller ants clean and chew these materials and turn them into compost for the mushrooms. Ants provide other pieces of fertilizer needed by mushrooms through their excrement. There is also a smaller group of ants that destroy foreign mushrooms that have randomly grown among them [72, 3]. Each colony of farmer ants has a specific smell. Ants recognize these chemical signals, called social profiles, and distinguish between members of their colony and those of other colonies. [74,75]. These colonies live together with numerous ants to plant and develop their fields. Also, the carbon social profile of each colony separates them from other colonies so that the ants of other colonies can be easily identified. This complex and intelligent system is the source of inspiration for the authors of the article to introduce the proposed method [76]. Figure 2 shows the mushroom-growing factors and their features [77,78].

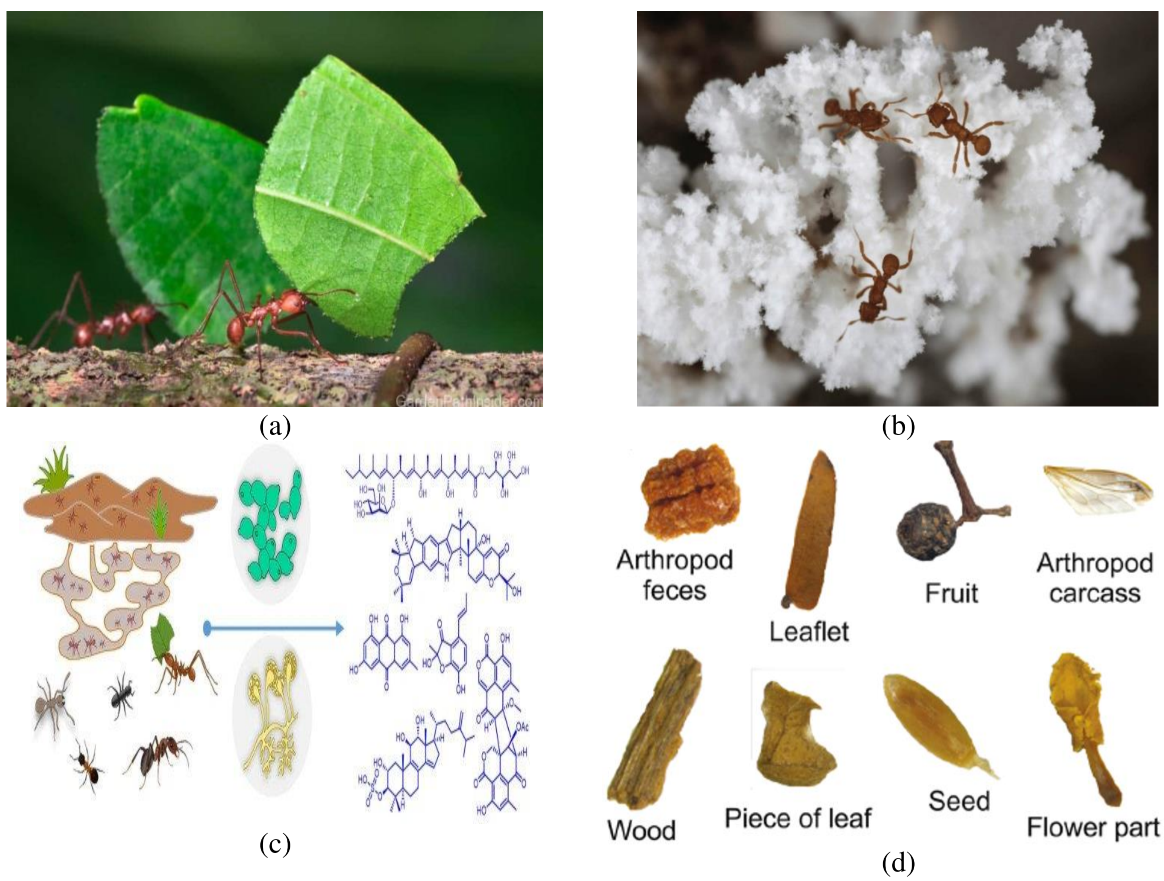

Fig.2 Details of the farmer ant's life; (a) leaf-cutter ants; (b) Cultivation of mushrooms by ants; (c) the structure of the social carbon profile; (d) types of foods to feed mushrooms

3.2 Farmer Ants Optimization Algorithm (FAOA)

In this section, the details of the FAOA algorithm which is inspired by the life of farmer ants are explained. At the beginning of the algorithm, the number of ants (n) that will be responsible for finding food and taking care of mushrooms (p) is determined. Each mushroom will be handled by an ant. The number of ants is greater than or equal to the number of mushrooms. In other words, n≥p. Each colony contains K nests in which the mushrooms are placed for breeding. It is not always the case that the number of mushrooms in the nests is equal. Near the nest, the ants have their food, and each mushroom is assigned to an ant to deliver the desired food. The set of foods (F) consists of M different types. The growth rate of each mushroom depends on the type of food it feeds. This dependency is unclear at the beginning of the algorithm. In the first step, each ant randomly chooses a food and measures the amount of its effect on the growth of the mushroom. At the end of the algorithm, the appropriate food for each mushroom will be determined. The effect of the food assigned to a mushroom on its growth is shown by the parameter C_1According to the social profile of the colony, the ants of each nest have the responsibility of taking care of the mushrooms. The social profile is used to separate the ants from each nest to avoid losing their way and going to the wrong nest. When the value of the social profile is SP=1, it means that ants will never lose their nest, and when SP=0, it means that the behavior of ants in reaching their nest and mushroom is completely random. Changing this factor can determine the exploration behavior of the algorithm, which can be adjusted according to each problem. in other words $0 \leq SP \leq 1$.

As mentioned earlier, the growth rate of a mushroom depends on the proper food it feeds on. But on the other hand, for its more effective growth, it also depends on the bacteria that the ants provide to take care of them. These bacteria protect the mushroom from the pest. Each mushroom growth rate is proportional to the type of bacteria produced by a given ant. For this reason, this factor can also be effective in mushroom growth. Bacteria can include several types, which are called Bacteria types or BT, and must be determined at the beginning of the algorithm. We determine the effectiveness of the bacteria in protecting the mushrooms with a coefficient C_2 .This coefficient can be adjusted according to the problem. The type of bacteria belonging to each ant is not changed. In order not to limit this issue, some other ants randomly deliver their bacteria to the mushrooms to vary the distribution of bacteria on the mushrooms. Another important factor that is effective in the growth of mushrooms is pests. Pests can interfere with mushroom growth and cause its destruction or weight loss. Pests are beyond the control of ants and are assumed to act randomly on mushrooms. PF or pest factor is considered a parameter with a negative impact on the growth of mushrooms, but its effect can be reduced by choosing the right bacteria for the mushroom. The effectiveness of PF is also determined by the coefficient C_3 The following equations show the effect of all the mentioned factors on the mushroom growth in each nest k.

$$W_k = \sum_{for\ all\ ants\ and\ mushrooms} W_0 + (C_1 \times f_m + C_2 \times B_i) \times W_0 - C_3 \times PF \times W_0 \tag{1}$$

Where W_0 is the initial weight of the mushroom. C_1, C_2, C_3 are learning parameters and. f_m is the value of food quality for food *m*. PF is also determined by equation 2.

$$PF = Is^{\alpha} \tag{2}$$

Where I is the negative impact value of the pest, s is the volume of the pest, and α is the spread parameter of the pest on mushrooms, which can be adjusted according to the problem.

The effect of bacteria on mushroom growth is also calculated by equation 3, where e is the positive effectiveness parameter of the bacteria, v is the volume of bacteria used, t is the lifetime of the bacteria, and β is the regulating parameter.

$$B = e \times v \times t^{\beta} \tag{3}$$

To increase the exploration capability of the algorithm, some random solutions can be considered. By doing this, more variety of solutions can lead to finding better solutions and avoiding falling into the local optimum. Equation 4 shows the global phases in finding the solution. At this stage, *1-FP* percentage of ants have random behavior and go to other nests to take care of the mushroom.

$$W_k = \sum_{for\ all\ ants\ and\ mushrooms} W_0 + (r_1 \times f_m + r_2 \times B_i) \times W_0 - r_3 \times PF \times W_0 \tag{4}$$

In this regard, the coefficients r_1, r_2, r_3, which are random numbers between zero and one, replace the coefficients C_1, C_2, C_3. In this case, the behavior of the algorithm is more random, and a global search is performed. Figure 3 shows the problem model. The weight of mushrooms produced in each nest k is calculated from the equation 1 and 4.

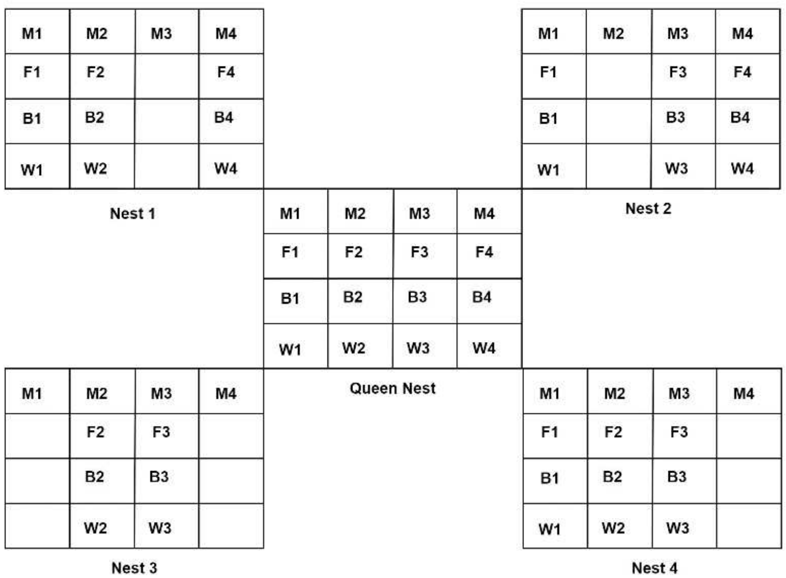

Fig.3. Ants Colony Model

In this figure, five nests are considered, one of which is the nest of the queen. M, F, B, and W represent mushrooms, food, bacteria, and weight, respectively. As this figure shows, not all nests contain all mushrooms. For example, there are no M3 mushrooms in Nest 1, or there are only M2 and M3 mushrooms in Nest 3. Calculations and operations on the nests are sent to the queen's nest, so the queen's nest has a complete structure and will contain all types of mushrooms. The weight of the queen's nest mushrooms is equal to the total weight of the nests. We define equation 5 as follows.

$$W = \sum_{k=1}^{k} W_k \tag{5}$$

Where *W* is the total weight of the mushrooms in the entire colony. Mushroom breeding nests send their local experiences to the queen nest to achieve global solutions. In this regard, some local and global operations should be carried out. Figure 4 shows the exchange of the partial solutions to construct a complete solution in the queen nest.

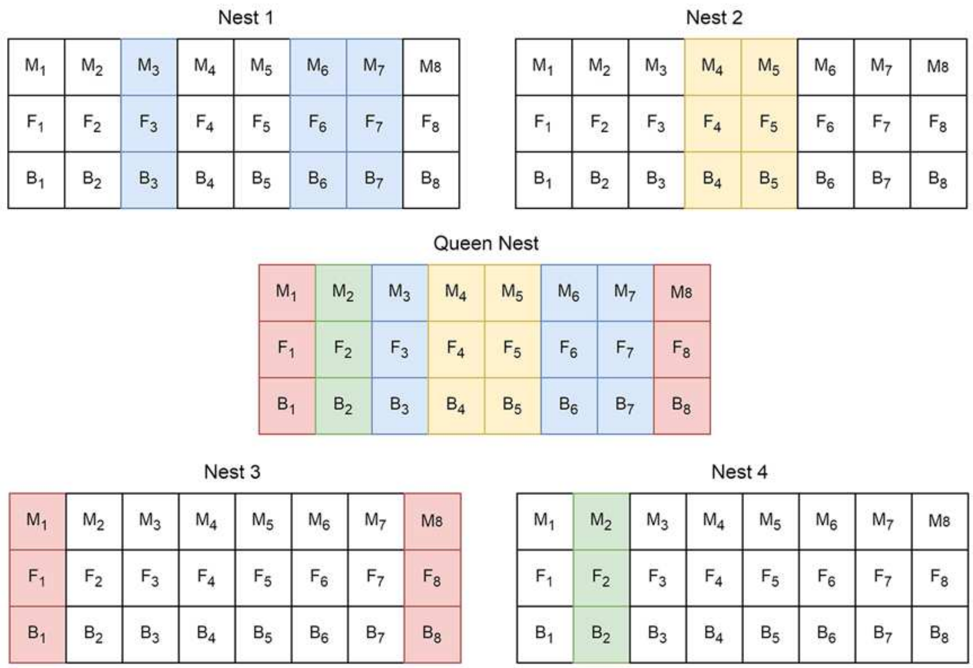

Fig.4. Exchange of Information

Food change operation:

At this stage, while keeping the ant constant(no bacteria change), the food of the mushrooms is changed to determine its impact on mushroom growth. To achieve this, two solution vectors are combined to create two new solution vectors (offspring). Subsequently, the fitness or quality of these new solution vectors is evaluated and added to the existing population. Figure 5 shows the food change operation.

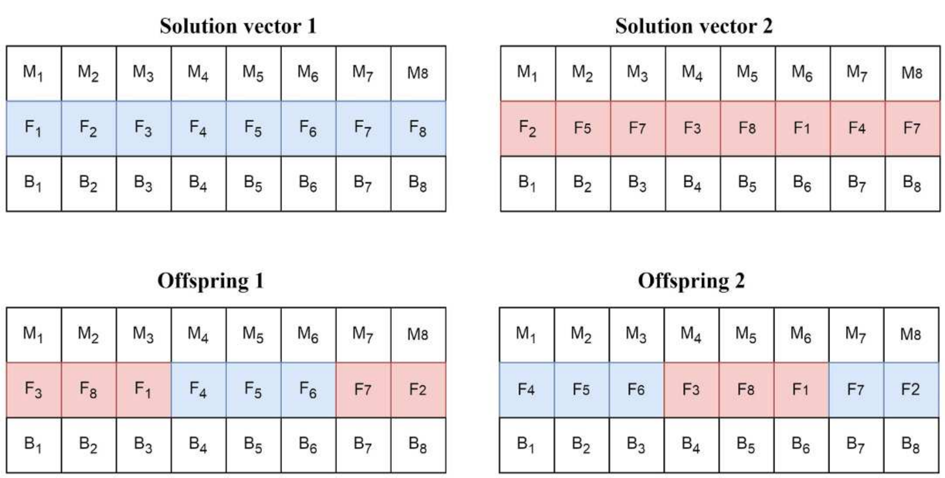

Fig.5. Food change operation

Bacteria change operation:

In the food change operation, the ant assigned to each mushroom is constant, and only the type of food is changed. Each ant has its bacteria type that cannot be changed. On the other hand, some bacteria types may not be suitable for the growth of some mushrooms. For this reason, it is necessary to change the bacteria to check its effectiveness for each mushroom. To do this, some ants outside the nest that have lost their way randomly move to the other nests, and bacteria change operations occur. This expands the problem search space and increases the probability of getting better solutions. Figure 6 shows the Bacteria change operation. In this figure, Bacteria B_9 and B_{10} replace B_4 and B_7 respectively and new offspring is generated.

Solution vector 3

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M8
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8

Offspring 3

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M8
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
B_1	B_2	B_3	**B_9**	B_5	B_6	**B_{10}**	B_8

Fig.6. Bacteria change operation

The proposed method is carried out in the following steps.

Step 1: Algorithm initialization including the number of ants, number of nests, types of mushrooms, foods, and bacteria

Step 2: Random distribution of mushrooms in nests and assigning ants to mushrooms

Step 3: Calculate the total weight of mushrooms in each nest using equations 1 to 4

Step 4: Do food change operations

Step 5: Do Bacterial change operation on some mushrooms according to SP

Step 6: Participate 1- SP percentage of ants in the global behavior of the algorithm

Step 7: Send the best relative pattern to the queen nest

Step 8: Calculate W_K or the total weight of mushrooms in each nest using relations 1 to 4

Step 9: Calculate the total weight of mushrooms in the entire colony or W using equation 5

Step 10: Remove weak solutions

Step 11: Repeat the algorithm until the stop condition is reached

Figure 7. shows the flowcharts of the FAOA

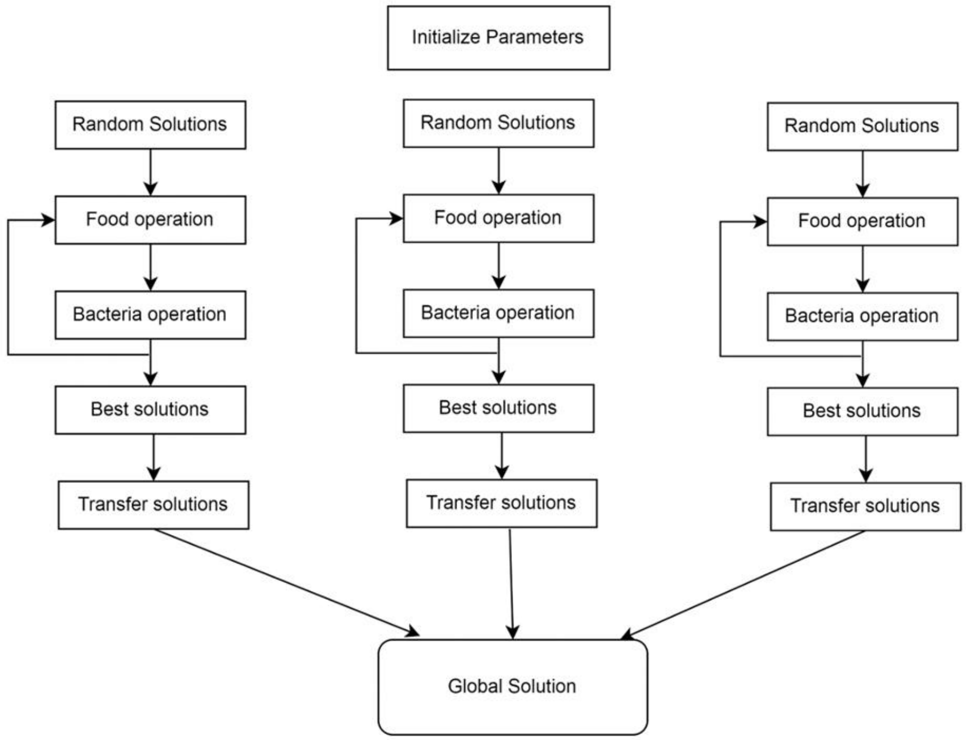

Fig.7 Flowchart of FAOA

Algorithm 1: Farmer's Side Process
Initialize parameters 1. for each nest K 2. Generate the initial population 3. Randomly Assign ants to mushrooms 4. $For\ each\ S_k$ 5. Compute W_K $by\ Equations\ 1\ to\ 4$ 6. do food operation 7. for 1- SP percent of mushrooms 8. Do bacteria operation 9. compute W_K for new solutions 10. Generate a random number $r \in [0,1]$ 11. if $r < p$ 12. Transfer the best partial solution into the Queen nest by $Equation\ 1$ 13 Else transfer the random partial solution into the queen nest by $Equation\ 4$ 14 Until the last iteration

Once all nests have sent their partial solutions to the queen's nest for aggregation, it is time to evaluate the complete solution. Algorithm 2 shows this step.

Algorithm 2: Queen side Process

Repeat
    For each iteration
        Receive partial solutions for all nests
        Compute W by Equation.5
        Do global food operation
        Do global bacteria operation
        Add solutions to the population
        Keep the P_s best solutions
    End For
Until the last iteration
Return the final solution

4. Evaluation and results

In this section, the effectiveness of FAOA in solving some engineering problems and benchmark functions is evaluated and compared with some state-of-the-art algorithms.

4.1 Problems and compared algorithms

In this section, some engineering problems have been deployed to evaluate the performance of FAOA. These problems include Antennal Locatio‹ Arable Field Planning [79], SLP [80], Embedded System [81], Fog Computing Systems Cloud [82], Knapsack [83], Truss Structures with Static Constraints [84], TSP [85], and UCAV Three-Dimension Path Planning [86].

Additionally, the algorithms compared with FAOA include ACO [23], BWO [87], CRO [89], GA [21], GP [58], GWO [17], ICA [88], SA [49], TSR [27], and Tabu [90]. Table 1 shows the parameters utilized in these algorithms.

Table. 1 Parameter values of algorithms.

Algorithms	**Parameters**	Values
Trees Social Relations Optimization Algorithm (TSR)	Population size	50
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100
Genetic Algorithm (GA)	Population size	50
	Number of generation of 10,000 cities	1000
	Number of generations for 1000	100
Gray Wolf Optimizer (GWO)	Control Parameter a	[2,0]
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100
	Number of particles	50
Imperialist Competitive Optimization (ICO)	Number of countries	50
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100

	Number of nimps	10
Simulated Annealing (SA)	Population size	50
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100
	Number of neighbors	10
Ant Colony Optimization (ACO)	Population size	50
	Number of generations	100
	Conversion ratio fitness	100
Taboo (Tabu)	Population size	50
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100
Genetic Programming (GP)	Population size	50
	Number of generation of 10,000 cities	1000
	Number of generations for 1000	100
CRO	Population size	50
	Number of generations of 10,000 cities	1000
	Number of generations for 1000	100
	Number of reefs	10
Gray Wolf Optimizer (GWO)	Control Parameter a	[0,2]
	Number of generations	1000
	Search Agent	50
Binary Whale Optimization Algorithm (BWO)	Number of generations	1000
	Parameter b	1
	Initial population	100

4.2 Discrete problems

Discrete optimization is a field of optimization in applied math and computer science. Unlike continuous optimization, some or all variables in a discrete optimization problem are restricted to discrete values, like integers. There are various discrete problems in scientific and engineering fields and several approaches are used to solve them. The proposed algorithm will also be used to solve these problems.

4.2.1 The TSP issue

The traveling salesman problem or TSP is one of the classic optimization problems in computer science. In this problem, a seller has a list of cities and must travel to each of them and choose a route in such a way that the total travel distance is minimized. He must pass through each city only once and return to his starting city. This problem is known as an NP-hard problem [85]. In this study, we utilized two scenarios to assess the proposed approach for addressing this issue. The initial scenario involved two hundred cities, with results displayed in Figure 8-a, while the second scenario comprised a thousand cities, with results shown in Figure 8-b. Initially, the algorithm's number of iterations is set to 100 and 1000, respectively. The FAOA algorithm in the first scenario is relatively satisfactory for this problem [91]. In the second scenario, when the number of cities is increased to 1000, the proposed method demonstrates significantly improved performance. This superiority is attributed to the parallel feature of the method and its powerful operators, which effectively break down complex problems into smaller sub-problems for more appropriate solutions.

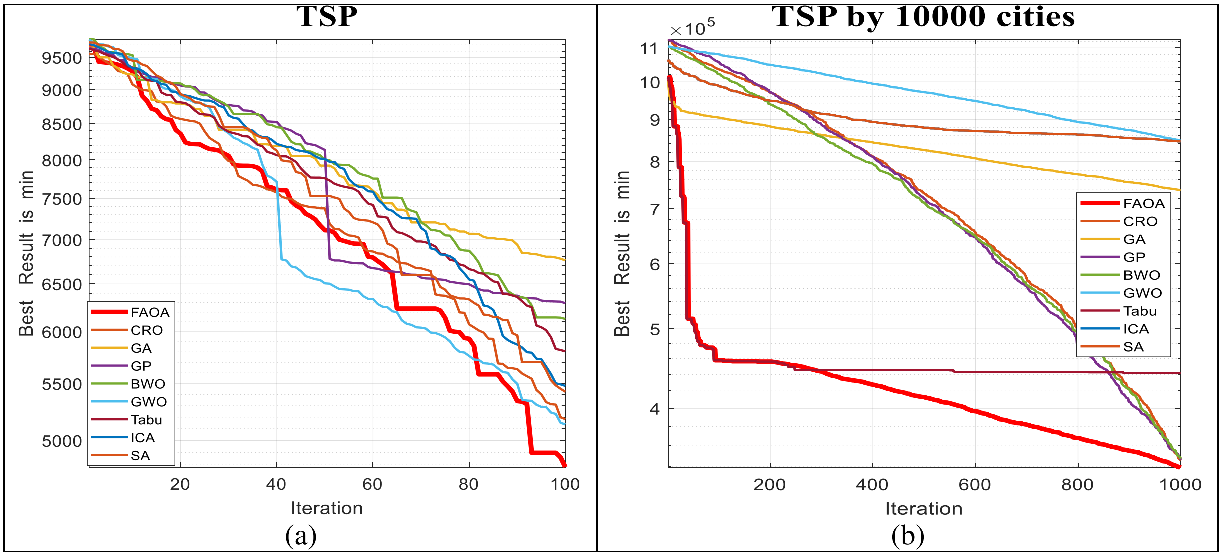

Fig.8 Comparison of FAOA with Other Discrete Algorithms for TSP: (a) 200 cities ; (b) 1000 cities

4.2.2 Knapsack problem

The knapsack problem is a classic discrete optimization problem. Given items with weights and profits, the goal is to maximize the total value by selecting items to put in the knapsack. For instance, if the knapsack can hold 20 kg, we need to choose items with a total weight of less than or equal to 20 kg while maximizing profit. The knapsack problem is an important optimization and NP-hard problem. In our experiments, we consider the number of iterations as 100. Additionally, the maximum allowed number of objects is 5. The objective of this problem is to maximize the weight of the Knapsack. Based on Figure 9, it is evident that the FAOA algorithm has yielded better results. Despite initially performing poorly in the early iterations, the algorithm's performance steadily improved after the 80th iteration.

4.2.3 Server placement problem

Proper placement of edge servers is crucial in mobile computing networks. This affects network response time, optimizes server load balance, and reduces server energy consumption. To evaluate the FAOA in server placement problem, we consider 300 antennas in the network area and aim to place 100 servers in optimal locations [92-94]. Figure 10 compares the performance of the FAOA algorithm with other algorithms. As this figure shows, the FAOA performs similarly to other algorithms between the 400th and 850th iteration but outperforms them thereafter.

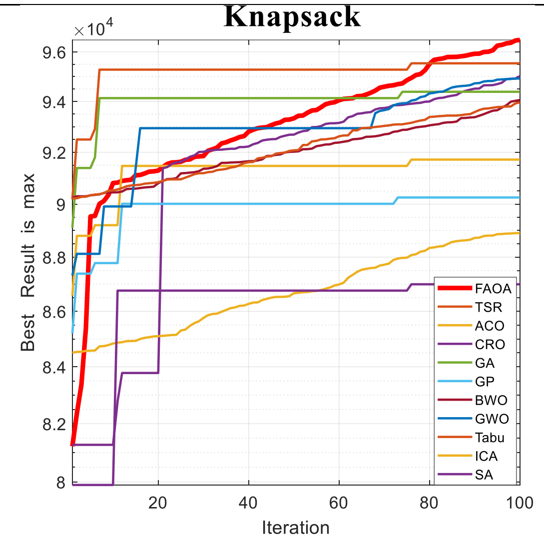

Fig.9 Results of Knapsack problem

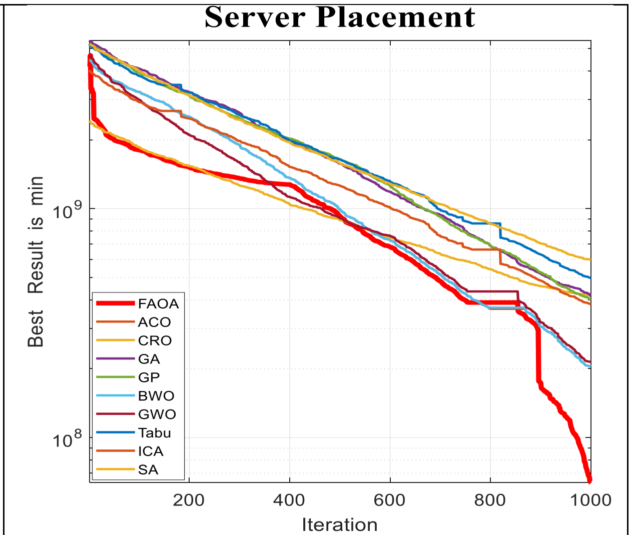

Fig.10 Results of server placement problem

4.2.4 Construction site layout planning (CSLP)

Construction site layout planning has always been a concern for clients, contractors, and consultants. The major goal is to arrange facilities like offices, warehouses, and residences in a way that optimizes the transfer of materials, information, and staff. A well-planned layout can enhance safety and efficiency, lower transport costs, and avoid bottlenecks and obstructions during material and equipment transfer, especially in large-scale projects. The site layout can be designed based on the preferences of decision-makers and various criteria [95]. We consider the number of rounds in this problem to be 1000. Figure 11 shows the performance of FAOA and its comparison with other methods. As this figure shows, the proposed method performed better than the other methods.

4.2.5 AFP (Arable Field Problem)

AFP concerns the act of preparing the land to a specific size and making it suitable for cultivation with the aid of machinery and tools. This issue aims to determine the most efficient and cost-effective method for each device's performance. For this purpose, 8 trucks have been used to solve the problem using the FAOA algorithm [96]. The number of iterations is considered to be 1000. As Figure 12 shows, FAOA results in better solutions compared to other algorithms.

4.2.6 ES (Emblem System for Intelligent Vehicles)

ES is one of the current issues in vehicle engineering. The goal of this issue is to enhance the efficiency of intelligent systems in automobiles. Vehicles, with all-wheel drive, offer greater mobility and flexibility compared with other vehicles. These types of omnidirectional mobile robots are very useful in some applications, like smart wheelchairs, industrial robots, and nursing robots [97]. However, there is a redundancy issue in this four-wheeled robotic system, as having more than three controllable degrees of freedom for mobile vehicles creates a redundant system [98]. optimization algorithms are considered a good solution in this field [98]. Figure 13 shows the performance of FAOA in solving this problem and comparing it with other algorithms. As this figure shows, FAOA performs better response than other algorithms.

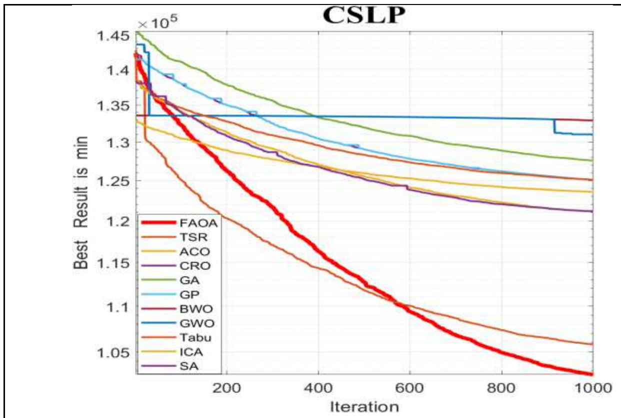

Fig.11 Results of CSLP problem

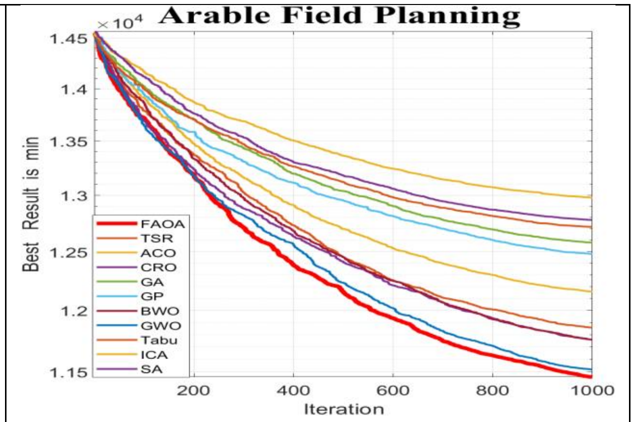

Fig.12 Results Of Arable Field planning problem

4.2.7 FCS (Fog Computing System-cloud problem)

Fog computing involves distributed computing on fog nodes, incorporating various devices (like sensors and home appliances), and has the potential to enhance the performance of Internet of Things (IoT) environments [99]. Fog nodes are usually deployed between low-level devices and high-level cloud computing platforms. Since the fog node deployment strategy affects the cost and performance of the fog computing system, determining an appropriate and efficient deployment strategy has become an optimization problem. The main objective of this issue is the fair allocation of computing resources [100]. Figure 14 shows the performance of FAOA in this problem compared to other algorithms. The results of experiments indicate the superiority of FAOA over the compared methods.

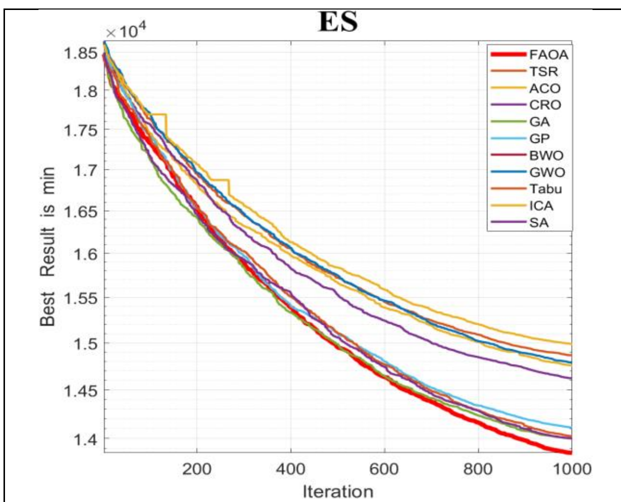

Fig.13. Results Of the ES problem

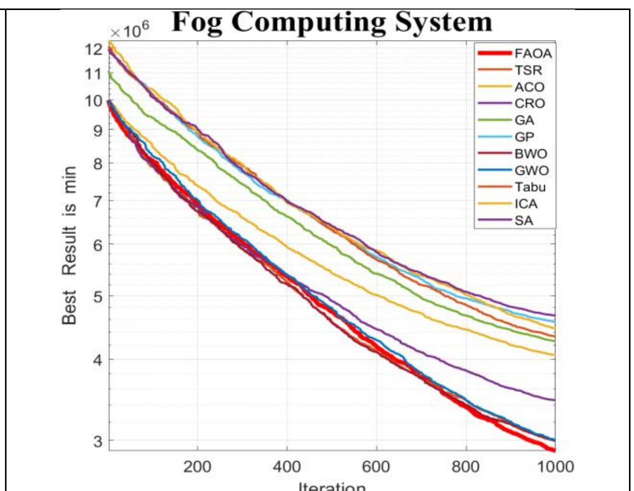

Fig.14. Results Of Fog Computing System Problem

4.2.8 Truss Structures problem

Aluminum and steel are commonly used in engineering [101]. Trusses are widely used in structural engineering. However, as the number of truss structures in a project increases, the design and analysis become more complex. This leads to an increase in the number of solutions for these structures. One of the primary objectives is to create the most cost-effective structure that meets specific loading conditions. This aims to prevent the unnecessary use of materials in response to the growing demand for raw materials by ensuring the optimal design of truss structures. The objective of this issue is to minimize the weight of the trusses while meeting movement and stress [102]. Optimization methods can be considered as a solution in this field. The performance of FAOA to solve this problem is shown in Figure 15. The results of experiments and comparisons with other algorithms show the superiority of the proposed method in solving this problem.

4.2.9 UCAV problem (UCAV Three-Dimension Path Planning)

The unmanned combat aerial vehicle (UCAV) is a complex optimization problem that focuses on flight path optimization, considering various types of constraints in complex battlefield environments. Unmanned aerial vehicles are either remotely piloted or self-piloted aircraft capable of carrying various accessories, like cameras, sensors, and communication equipment. They have diverse applications in both civilian and military domains. Their popularity stems from their low cost, compact size, and extensive maneuverability [103]. Path generation and planning are key technologies in countering UCAV [104]. Optimization methods can be considered as a solution in this field. Figure 16 depicts the performance of FAOA in this scenario. It illustrates that FAOA initially has average performance, but starting from round 93, the algorithm consistently achieves the best results.

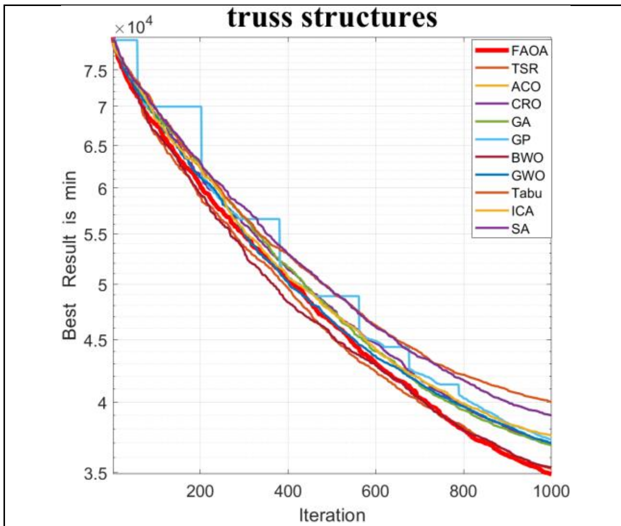

Fig 15 Results Of Truss Structures problem

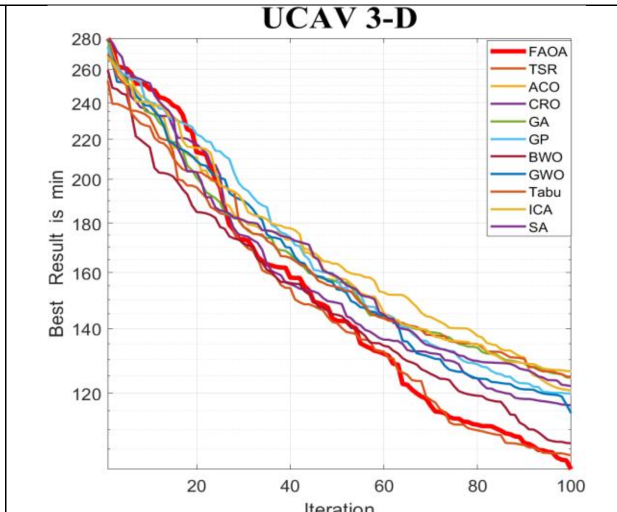

Fig.16 Results of UCAV 3-D problem

4.3 Benchmark Functions

In this section, 22 standard benchmark functions with various characteristics are utilized to evaluate the performance of FAOA and other algorithms. These functions consist of seven unimodal functions, nine multimodal functions, and six combined functions. Tables 2 to 4 display the unimodal, multimodal, and combined functions, respectively. Additionally, Table 4 presents the performance results of these algorithms [87]. The results of these tables show that, in most cases, the efficiency of the proposed method is superior to other algorithms.

Table. 2 Unimodal Functions			
Function	*Range*	*Dim*	*Minimum*
$f1=\sum_{i=1}^{n} x_i^2 \|$	[-100, 100]	30	0
$f2=\sum_{i=1}^{n} \| x_i \| + \prod_{i=1}^{n} \| x_i \|$	[-10, 10]	5	0
$f3=\sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	[-100, 100]	5	0
$f4=\max_i \{\| x_i \|, 1 \le i \le n\}$	[-100, 100]	5	0
$f5=\sum_{i=1}^{n-1} [100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	[-30, 30]	5	0
$f6=\sum_{i=1}^{n} ([x_i+0.5])^2$	[-100, 100]	5	0
$f7=\sum_{i=1}^{n} ix_i^4 + random[0,1]$	[-1.28, 1.28]	5	0

Table. 3 Multimodal Functions			
Function	*Range*	*Dim*	*Minimum*
$f8=\sum_{j=1}^{n} -Z_j \sin(\sqrt{\| Z_j \|})$	[-500, 500]	30	-418.98 ×5
$f9=\sum_{i=1}^{n} [z_j^2 - 10\cos(2\pi z_j)+10]$	[-5.12, 5.12]	30	0
$f10=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} z_j^2}) - \exp(\frac{1}{n}\sum_{j=1}^{n}\cos(2\pi Z_j))+20+e$	[-32, 32]	30	0
$f11=\frac{1}{4000}\sum_{j=1}^{n} z_j^2 - \prod_{j=1}^{n}\cos(z_j/\sqrt{j})+1$	[-600, 600]	30	0
$f12=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{j=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{j+1})+(y_n-1)^2\}$ $+\sum_{j=1}^{n} u(x_j,10,100,4),$ $y_j=1+\frac{x_j+1}{4}, u(x_j,a,k,m)=\begin{cases} k(x_j-a)^m & x_j>a \\ 0 & -a>x_j>a \\ k(-x_j-a)^m & -a>x_j \end{cases}$	[-50, 50]	30	0
$f13=0.1\{\sin^2 3\pi z_1)+\sum_{j=1}^{n}(z_j-1^2[1+\sin^2(3\pi z_j+1)]+(z_n-1)^2$ $[1+\sin^2(2\pi z_n)]\}+\sum_{j=1}^{n} u(z_j,5,100,4)$	[-50, 50]	30	0
$f14=-\sum_{i=1}^{n}\sin(x_i)\times(\sin(i.x_i^2/\pi))^{2m}, m=10$	[-100, 100]	2	-1
$f15=[\exp(-\sum_{i=1}^{n}(x_i/\beta)^{2m}-2\exp(-\sum_{i=1}^{n}x_i^2)]\times\prod_{i=1}^{n}\cos^2 x_i, m=5$	[-100, 100]	2	0
$f16=\{[\sum_{i}^{n}\sin^2(x_j)]-\exp(-\sum_{i=1}^{n}x_i^2)\}\times\exp[-\sum_{i=1}^{n}\sin^2\sqrt{\| x_i \|}]$	[-5, 5]	2	-1.0316

Table. 4 Composite Functions				
No.	*Function*	*Range*	*Dim*	*Minimum*
F17	$f_1, f_2, f_3, \ldots, f_{10}$ Sphere function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [1,1,1,…1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ [5/100, 5/100, 5/100,…, 5/100]	[-100, 100]	10	0
F18	$f_1, f_2, f_3, \ldots, f_{10}$ Griewank's function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [1,1,1,…1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ [5/100, 5/100, 5/100,…, 5/100]	[-10, 10]	10	0
F19	$f_1, f_2, f_3, \ldots, f_{10}$ Griewank's function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [1,1,1,…1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ [1,1,1,…,1]	[-100, 100]	10	0
F20	$f_1, f_2 =$ Ackley's function f_3, f_4 Rastrigin's function f_5, f_6 Weierstrass's function f_7, f_8 Griewank's function f_{9}, f_{10} Sphere function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [1,1,1,…1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]	[-100, 100]	10	0
F21	$f_1, f_2 =$ Rastrigin's function f_3, f_4 Weierstrass's function f_5, f_6 Griewank's function f_7, f_8 Ackley's function f_{9}, f_{10} Sphere function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [1,1,1,…1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]	[-30, 30]	10	0
F22	$f_1, f_2 =$ Rastrigin's function f_3, f_4 Weierstrass's function f_5, f_6 Griewank's function f_7, f_8 Ackley's function f_{9}, f_{10} Sphere function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ 0.1 × 1/5, 0.2 × 1/5, 0.3 × 5/0.5, 0.4× 5/0.5, 0.5 × 5/100, 0.6 × 5/100,	[-100, 100]	10	0

	0.7 × 5/32, 0.8 × 5/32, 0.9 × 5/100, 1 × 5/100			

Table 5. Comparison results for benchmark functions

No.	*Measure*	*FAOA*	*CRO*	*GA*	*GP*	*BWO*	*GWO*	*Tabu*	*ICA*	*SA*	*ACO*	*TSR*
F1	Worst	**1.2226**	1.6567	1.6664	1.6664	1.6533	1.3074	1.3143	1.3066	1.3143	1.8590	1.3305
	Best	0.6405	0.7291	0.7739	0.8129	0.8290	**0.4884**	0.6468	0.6692	0.6156	0.8744	0.7902
	Med	**0.8168**	1.1465	1.1977	1.2719	1.1355	0.8993	1.2389	0.9187	0.8564	1.1923	0.9168
F2	Worst	**10.9587**	11.3290	12.0828	12.4438	13.3660	12.7284	11.9477	12.7284	11.9585	13.3573	10.9977
	Best	5.3574	**5.1686**	5.7474	6.1975	5.9946	6.2801	5.2693	6.2801	5.6281	6.7918	5.2679
	Med	7.6834	7.7909	7.9445	9.8799	8.4458	8.8632	7.8180	8.8632	7.9842	9.0852	**7.5280**
F3	Worst	10.9587	11.3290	12.0828	11.3641	13.3660	12.7284	10.9477	12.7284	11.9585	13.3573	**9.9656**
	Best	**5.3574**	5.3686	5.7474	6.1975	5.9946	6.8331	5.5673	6.2801	5.6281	6.7918	5.5673
	Med	8.6220	7.7909	7.9445	9.8799	8.4458	8.9541	**7.3180**	8.8632	7.9842	9.0852	7.4260
F4	Worst	**0.8994**	1.0002	1.0070	1.3677	0.9981	1.0074	1.3143	1.1092	1.0843	1.0100	0.9892
	Best	0.4730	0.4993	0.4929	0.5203	0.4972	0.4884	0.6468	0.5506	0.5316	**0.4701**	0.6673
	Med	**0.6402**	0.7080	0.6871	0.7116	0.6902	0.6993	1.2389	0.8861	0.7818	0.6745	0.6415
F5	Worst	**3.1211**	3.9431	4.1744	3.9554	4.6525	4.0957	3.8584	3.8806	3.9971	4.6902	4.3219
	Best	**1.7522**	2.2766	1.9454	1.7936	2.2766	1.8504	1.9857	1.8863	1.7857	2.2492	1.8659
	Med	2.80982	3.2650	3.002	2.8007	3.1870	2.6767	2.6913	**2.6070**	2.6988	3.0873	2.8098
F6	Worst	2.0063	2.0262	2.1124	2.0513	2.1310	**1.8285**	2.2780	2.0669	2.0324	2.1038	1.9954
	Best	**0.9854**	1.0366	1.0150	1.2310	1.1456	1.0366	1.0513	1.3321	1.0224	0.9692	0.9942
	Med	1.4084	1.4639	1.4642	1.7264	1.5722	1.4639	1.5325	1.3991	1.3987	**1.3658**	1.3668
F7	Worst	25.1211	21.9253	22.0878	22.0878	22.0424	21.9731	21.3654	21.9373	**21.0684**	21.8900	25.2519
	Best	**10.2678**	10.3068	10.6314	11.6009	10.8434	10.7293	11.1668	10.9716	11.1868	10.8703	11.9538
	Med	15.9337	14.7396	15.0474	17.6634	14.9337	15.2530	14.7483	**14.1574**	16.5789	14.7572	16.9507
F8	Worst	-4.5334	0.4053	0.7774	0.7730	-9.0650	-9.2331	-18.2535	-15.2535	**-64.666**	-2.2731	-5.5324
	Best	**-607.59**	-438.91	-479.62	-573.01	-526.75	-544.52	-477.41	-477.41	-480.42	-475.46	-588.17
	Med	**-360.89**	-259.36	-314.60	-302.37	-316.94	-345.38	-297.29	-302.33	-300.59	-317.55	-371.12
F9	Worst	**20.9378**	22.0333	21.6217	21.6335	22.0750	21.9096	21.9830	21.7611	21.9830	22.2724	21.7434
	Best	**10.2199**	10.8661	11.0292	13.4440	10.8661	10.7896	10.8292	11.0831	11.3330	10.9488	11.0184
	Med	**14.5628**	15.2704	15.5599	17.2383	15.2704	14.7859	14.6263	15.0769	14.7723	15.2834	14.9042
F10	Worst	1.9797	1.6567	1.6664	1.9747	1.8862	**1.0075**	1.14453	1.3066	1.3143	1.8630	1.9446
	Best	0.9532	0.7291	0.8129	0.9327	0.9004	**0.4884**	0.6469	0.6693	0.6156	0.8532	0.9972
	Med	1.3296	1.1465	1.1593	1.3179	1.2712	**0.6993**	1.2389	0.9187	0.8564	1.2511	1.3318
F11	Worst	**11.5569**	13.7779	13.7837	12.7673	13.8091	12.8441	12.1889	12.8139	12.9898	12.5548	12.8139
	Best	0.9532	0.7291	0.8129	0.9327	0.9004	**0.4884**	0.6469	0.6693	0.6156	08532	0.9532
	Med	1.0296	1.1465	1.1593	1.3179	1.2712	**0.6993**	1.2389	0.9187	0.8564	1.2511	1.3296
F12	Worst	**17.8449**	20.8558	19.9719	19.9802	19.9472	19.6997	20.7629	19.6568	19.5704	20.9049	21.5133
	Best	10.1063	10.2402	9.3473	9.6638	9.8094	9.6014	10.3482	9.5607	**9.1808**	9.7175	9.9035
	Med	**13.4438**	14.2109	13.6702	13.6702	13.8010	13.9259	14.0935	13.5037	13.6570	13.6363	14.6050
F13	Worst	31.8301	30.5145	**30.4432**	32.4154	35.4449	31.5799	30.9385	33.9556	34.7625	32.7230	43.9484
	Best	**12.9900**	15.0983	**13.1165**	16.0333	14.8222	14.5822	19.8335	21.3236	22.4064	15.3866	21.6129
	Med	22.8353	20.9827	**19.1762**	22.2618	22.2631	21.0264	27.2229	27.0829	28.0355	21.5134	30.9913
F14	Worst	39.3650	58.8548	49.6172	49.6172	52.4141	54.8990	38.5144	38.6164	**38.5144**	48.8547	52.0969
	Best	20.3952	25.6390	24.9847	24.3202	25.3664	27.3500	**18.0410**	18.3310	19.9596	24.2191	26.5638
	Med	**24.5581**	40.5171	34.5394	41.4929	37.5636	36.4024	27.6067	26.6771	27.5412	33.2746	36.7144
F15	Worst	**27.5214**	34.4763	32.4191	31.6964	31.0444	29.2296	27.8899	31.3618	30.4080	31.0444	38.6854
	Best	**14.2108**	18.2131	18.9884	14.7338	16.3008	14.7999	15.499	15.4935	15.4647	16.2108	18.7600
	Med	19.4930	24.1748	27.1994	21.4537	25.3031	21.4487	20.6888	**18.6888**	22.3304	26.6939	25.6250

F16	Worst	147.695	148.143	138.968	130.987	137.410	128.2279	128.2250	128.197	**127.805**	146.083	147.990
	Best	**55.630**	65.864	64.498	66.907	69.659	58.576	59.8529	58.576	62.131	69.777	67.449
	Med	100.913	95.735	131.242	105.476	99.322	**80.650**	83.4341	123.008	90.170	99.6067	95.988
F17	Worst	1.2797	1.6567	1.6663	1.9746	1.8999	**1.0074**	1.3903	1.3066	1.3143	1.8630	1.9445
	Best	**0.3531**	0.7290	0.8129	0.9327	0.9003	0.4884	0.6468	0.6692	0.6156	0.8532	0.9531
	Med	**0.5296**	1.1464	1.1592	1.3179	1.2711	0.6993	1.2389	0.9187	0.8564	1.6004	1.3296
F18	Worst	88.065	89.672	99.672	97.302	98.360	111.545	**81.339**	89.608	89.786	97.932	117.736
	Best	**40.778**	43.001	62.503	60.683	45.543	45.543	42.754	44.930	44.812	47.233	56.591
	Med	**49.580**	60.250	57.737	60.683	65.191	65.191	54.845	65.226	64.332	66.505	80.496
F19	Worst	1.3797	1.6567	1.6664	1.9747	1.8863	**1.0075**	1.3143	1.3066	1.3711	1.8630	1.9446
	Best	0.5532	0.7291	0.8129	0.9327	0.9004	**0.4884**	06469	0.6693	0.6156	0.8532	0.9532
	Med	**0.7296**	1.1465	1.1593	1.3179	1.2712	0.8993	1.2389	0.9187	0.8564	1.2511	1.3296
F20	Worst	**14.9639**	21.8677	20.0637	22.7771	20.5455	17.9166	17.7144	15.7328	17.9709	21.7651	17.6642
	Best	**7.3037**	10.5102	9.6211	9.7146	11.4424	8.5933	8.8341	7.8003	8.0778	10.4934	8.8085
	Med	**11.9414**	14.3246	13.4553	15.7196	15.4901	12.2676	12.8070	12.8921	12.5988	14.4777	12.9201
F21	Worst	**17.5639**	21.9898	20.0637	22.7771	23.0112	17.9166	17.7144	18.7328	17.9709	21.7651	17.6642
	Best	8.3037	10.5102	9.2611	9.6211	11.4424	8.5933	8.8341	**7.8003**	8.0778	10.4934	8.8085
	Med	**11.9414**	14.3246	13.4553	15.7196	15.4901	12.2676	12.8070	12.8921	12.5988	14.4777	12.9201
F22	Worst	2.9258	4.9365	4.9832	**1.9477**	4.9631	5.9998	5.9865	4.9542	5.0002	5.000	5.8980
	Best	0.9911	2.3619	2.4346	**0.9327**	2.5401	2.6400	2.9500	2.4413	2.9500	2.5401	2.8357
	Med	**1.1071**	3.2982	3.3766	1.3179	3.6015	3.3513	4.0263	3.711	4.1572	3.6015	4.0007

4.4 Discussion

In this article, the farmer ants optimization algorithm (FAOA), as a new meta-heuristic algorithm, was explained. This algorithm, which is suitable for solving discrete problems, has unique features that distinguish it from other similar algorithms. In most similar algorithms, negative impact parameters and problem constraints are not considered. This will reduce their efficiency and accuracy. The proposed method considers the factors that enhance the solution to the problem, while also taking into account the limiting factors that can significantly affect the solution. For example, in the performance of a processor, factors such as processor speed and RAM capacity are considered. On the other hand, for example, the heat of the processor at higher speeds, which can negatively affect the calculation speed, is less considered. The impact of processor speed and RAM capacity varies for different applications and differs for each specific problem. Some applications are processor-intensive and others require more memory. The proposed algorithm with its operators and unique nature determines the effects of positive and negative factors in the learning process, which can help make the problem-solving process more realistic. On the other hand, another important feature of the proposed algorithm is its parallelization capability. In most cases, population-based algorithms require many iterations to get the optimal solution. Breaking the problem into smaller components that lead to local processing can reduce the complexity of the problem and convergence can be achieved in fewer iterations. The local and global features of the proposed algorithm expand the search space and increase the chances of reaching a global solution.

4.5 time complexity

In this section, we analyze the complexity of the FAOA. In the proposed algorithm, we start with a population of p, which is divided into k sub-rooms. Each sub-room contains *m* mushrooms, and each mushroom is assigned to an ant. By dividing solutions into k nests, the required calculations in each sub-section will be reduced by n/k and will be of logarithmic type. If we consider the number of algorithm iterations (maxiter) as *n* and also consider the three operations of addition and subtraction performed, the total number of necessary calculations equals:

$$O((\text{maxiter}) * \frac{\text{p}}{\text{k}} * (W_0 \ + (C_1 \times f_m + C_2 \times B_i) \times W_0 - \ C_3 \times PF \times W_0))= \tag{6}$$
$$O\left(n * \frac{\text{p}}{\text{k}} + 3\right) = O(n * (\log_2 n + 3)) = O(n * \log_2 n)$$

5. Conclusion

In this paper, a new meta-heuristic algorithm named the farmer ants optimization algorithm (FAOA) was introduced. FAOA is inspired by the life of a group of farmer ants who grow mushrooms. In this algorithm, some aspects of the life of these ants, including mushroom cultivation, feeding, and caring for them, have been used. This algorithm, which is suitable for solving discrete optimization problems, can solve NP-hard problems and provide optimal solutions, especially for large-scale problems. The important feature of this algorithm includes considering the positive and negative impact of problem parameters in reaching the solution and its parallelization capability, which is suitable for problems with a discrete nature, especially with high dimensions. Experiments performed on some classic and new engineering problems, as well as on some benchmark functions, show that the proposed method is efficient in solving optimization problems. As a future work, we plan to develop this algorithm to solve problems with a continuous nature, so that it can solve a wider range of problems.

Acknowledgments:

Competing Interests: The authors declare that they have no competing interests.

Authors contribution statement: A. Asghari and M. Alimoradi carried out the experiments. M. Zeinalabedinmalekmian wrote the manuscript with support from H.Azgomi. S.Ghaziantafrishi translated the manuscript into English and supervised the project.

Ethical and informed consent for data used: All information taken from the study will be coded to protect each subject's name. No names or other identifying information will be used when discussing or reporting data. The Authors) will safely keep all files and data collected in a secured locked place.

Data availability and access: Datasets are available from the corresponding author upon reasonable request.

References

[1] K. ChandraBora and B. Kalita, *"Exact Polynomial-time Algorithm for the Clique Problem and P= NP for Clique Problem,"* in IJCA, 2013, vol. 73, no. 8, pp. 19–23
[2] G. J. Woeginger, *"Exact algorithms for NP-hard problems: A survey,"* in Combinatorial optimization—eureka, you shrink!, Springer, 2003, pp. 185–207.
[3] Feng-Tse Lin, Cheng-Yan Kao, and Ching-Chi *Hsu "Applying the Genetic Approach to Simulated Annealing in Solving Some NP-Hard Problems",* IEEE Transactions on Systems, Man, and Cybernetics, pp 1752 - 1767 (1993)
[4] Gerhard J. Woeginger *"Exact Algorithms for NP-Hard Problems: A Survey",* Combinatorial Optimization Eureka, You Shrink! pp 185–207 (2003)
[5] Seyedali Mirjalili *"SCA: A Sine Cosine Algorithm for Solving Optimization Problems",* Knowledge-Based Systems Volume 96, 15 March 2016, pp 120-133
[6] A. R. Simpson, G. C. Dandy, and L. J. Murphy, *"Genetic algorithms compared to other techniques for pipe optimization,"* Journal of water resources planning and management, vol. 120, pp 423-443, 1994.
[7] C. James, *"Introduction to Stochastics Search and Optimization,"* ed: Wiley-Interscience, New Jersey, 2003.
[8] I. Boussaïd, J. Lepagnot, and P. Siarry, *"A survey on optimization metaheuristics,"* Information Sciences, vol. 237, pp. 82-117, 2013.

[9] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez, *"Metaheuristic optimization frameworks: a survey and benchmarking,"* Soft Computing, vol. 16, pp. 527-561, 2012.
[10] Seyedali Mirjalili *"Moth-Flame Optimization Algorithm: A Novel Nature-inspired Heuristic Paradigm",* Knowledge-Based Systems Volume 89, November 2015, pp 228-249
[11] Wu, Guohua, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. *"Ensemble strategies for population-based optimization algorithms–A survey."* Swarm and evolutionary computation 44 (2019): 695-711.
[12] Warren S. Mcculloch, Walter Pitts *"A Logical Calculus Of The Ideas Immanent In Nervous Activity",* Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133.
[13] J. Von Neumann, A. W. Burks, *"Theory of self-reproducing automata",* IEEE Transactions on Neural Networks, 1966
[14] J. Nocedal, S. J. Wright, *"Numerical optimization 2nd",* 2006.
[15] G. Wu, *"Across neighborhood search for numerical optimization",* Information Sciences 329 (2016) 597–618.
[16] G. Wu, W. Pedrycz, P. N. Suganthan, R. Mallipeddi, *"A variable reduction strategy for evolutionary algorithms handling equality constraints",* Applied Soft Computing 37 (2015) 774–786.
[17] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis *"Grey Wolf Optimizer",* Advances in Engineering Software Volume 69, March 2014, pp 46-61
[18] Reeves, Colin R. Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc., 1993.
[19] Blum, Christian, Jakob Puchinger, Günther R. Raidl, and Andrea Roli. *"Hybrid metaheuristics in combinatorial optimization: A survey*." Applied Soft Computing 11, no. 6 (2011): 4135-4151
[20] Arman Daliri, Ali Asghari, Hossein Azgomi & Mahmoud Alimoradi*, "The water optimization algorithm: a novel metaheuristic for solving optimization problems",* Applied Intelligence volume 52, pp17990-18029 (2022)
[21] Holland, John H. *"Genetic algorithms",* Scientific american 267, no. 1 (1992): 66-73.
[22] Castro, Leandro Nunes, Leandro Nunes De Castro, and Jonathan Timmis. *"Artificial immune systems: a new computational intelligence approach",* Springer Science & Business Media, 2002.
[23] Marco Dorigo, Mauro Birattari, Thomas Stutzle *"Ant Colony Optimization",*IEEE Computational Intelligence Magazine (Volume: 1, Issue: 4, November 2006), pp 28 – 39
[24] Kennedy, J., and R. Eberhart. *"Particle swarm optimization (PSO)."* In Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948. 1995.
[25] Xin-She Yang *"Flower Pollination Algorithm for Global Optimization"* Unconventional Computation and Natural Computation, pp 240–249
[26] Xin-She Yang *"Firefly Algorithms for Multimodal Optimization",* SAGA 2009: Stochastic Algorithms: Foundations and Applications pp 169–178
[27] Mahmoud Alimoradi, Hossein Azgomi, Ali Asghari*, "Trees Social Relations Optimization Algorithm: A new Swarm-Based metaheuristic technique to solve continuous and discrete optimization problems",* Mathematics and Computers in Simulation Volume 194, April 2022, pp 629-664
[28] Maziar Yazdani, Fariborz Jolai *"Lion Optimization Algorithm (LOA): A NatureInspired Metaheuristic Algorithm",* Journal of Computational Design and Engineering, Volume 3, January 2016, pp 24-36, https://doi.org/10.1016/j.jcde.2015.06.003
[29] Blum, Christian, and Andrea Roli. *"Metaheuristics in combinatorial optimization: Overview and conceptual comparison."* ACM computing surveys (CSUR) 35, no. 3 (2003): 268-308.
[30] R. Oftadeh, M.J. Mahjoob, M. Shariatpanahi *"A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search",* Computers & Mathematics with Applications Volume 60, Issue 7, October 2010, pp 2087-2098
[31] Xin-She Yang, Nilanjan Dey, Simon Fong*, "Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications",* Nature-inspired Metaheuristic Algorithms 2010
[32] D. H. Wolpert and W. G. Macready, *"No free lunch theorems for optimization,"* IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, 1997.
[33] Wolpert, David H., and William G. Macready. *"No free lunch theorems for optimization."* IEEE transactions on evolutionary computation 1, no. 1 (1997): 67-82.
[34] Warren S. Mcculloch, Walter Pitts *"A Logical Calculus Of The Ideas Immanent In Nervous Activity",* Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133.
[35] J. Von Neumann, A. W. Burks, *"Theory of self-reproducing automata",* IEEE Transactions on Neural Networks, 1966
[36] L. Ardleman*, "Molecular computation of solutions to combinatorial problems",* SCIENCE 1994 Vol 266, pp. 1021-1024
[37] K. A. De Jong, *"Evolutionary computation a unified approach",* GECCO Companion '15: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, July 2015, pp 21–35, https://doi.org/10.1145/2739482.2756576
[38] Yu-Jun Zheng "*Water wave optimization: A new natureinspired metaheuristic",* Computers & Operations Research Volume 55, March 2015, pp 1-11
[39] Jonathan Z. Shik, Pepijn W. Kooij, David A. Donoso , Juan C. Santos, Ernesto B. Gomez, Mariana Franco, Antonin J. J. Crumiere , Xavier Arnan, Jack Howe, William T. Wcislo, Jacobus J. Boomsma *"Nutritional niches reveal fundamental domestication trade-offs in fungus-farming ants",* Nature Ecology & Evolution volume 5, pp 122-134 (2021)

[40] Henrik H. De Fine Licht, Morten Schi□tt, Ulrich G. Mueller, Jacobus J. Boomsma *"EVOLUTIONARY TRANSITIONS IN ENZYME ACTIVITY OF ANT FUNGUS GARDENS",* Evolution, Volume 64, 1 July 2010, pp 2055-2069
[41] Gaurav Dhiman, Vijay Kumar*"Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications",* Advances in Engineering Software Volume 114, December 2017, pp 48-70
[42] Rafael Stubs Parpinelli, Heitor Silvério Lopes *"New inspirations in swarm intelligence: A survey",* International Journal of Bio-Inspired ComputationVol. 3, No. 1, February 2011, pp 1-16, https://doi.org/10.1504/IJBIC.2011.0387
[43] Hamed Shah-Hosseini *"Optimization with the Nature-Inspired Intelligent Water Drops Algorithm"*, Evolutionary Computation, pp 297-338
[44] B. LIU and Y. ZHOU, *"Artificial fish swarm optimization algorithm based on genetic algorithm",* Computer engineering and design, 2008
[45] Aboul-Ella Hassanien, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa *"Enabling AI Applications in Data Science",* Studies in Computational Intelligence (SCI, volume 911), 2021
[46] A. Kaveh, M. Khayatazad *"A new meta-heuristic method: Ray Optimization",* Computers & Structures Volumes 112–113, December 2012, pp 283-294
[47] Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi *"GSA: A Gravitational Search Algorithm",* Information Sciences Volume 179, Issue 13, 13 June 2009, pp 2232-2248
[48] Fatma A. Hashim, Kashif Hussain, Essam H. Houssein, Mai S. Mabrouk, Walid Al-Atabany *"Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems",* Applied Intelligence volume 51, pp 1531–1551 (2021)
[49] Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. *"Optimization by simulated annealing.",* SCIENCE, 13 May 1983, Vol 220, pp. 671-680
[50] A. Kaveh, S. Talatahari *"A novel heuristic optimization method: charged system search",* Acta Mechanica volume 213, pp 267-289 (2010)
[51] oshua D. Knowles and David W. Corne, M-PAES: *"A Memetic Algorithm for Multiobjective Optimization",* Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)
[52] S. ILKER BIRBIL, SHU-CHERNG FANG "*An Electromagnetism-like Mechanism for Global Optimization",* Journal of Global Optimization volume 25, pp 263-282 (2003)
[53] Wei Cai, Weiwei Yang, Xiaoqian Chen *"A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization",* 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), 28 October 2008
[54] Ali Karci *"Theory of Saplings Growing Up Algorithm",* Adaptive and Natural Computing Algorithms, pp 450-460
[55] Alok Kumar Shukla1, Diwakar Tripathi, B.Ramachandra Reddy, D.Chandramohan *"A study on metaheuristics approaches for gene selection in microarray data: algorithms, applications and open challenges",* Evolutionary Intelligence volume 13, pp 309-329 (2020)
[56] A. K. Qin, V. L. Huang, P. N. Suganthan *"Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization",* IEEE Transactions on Evolutionary Computation (Volume: 13, April 2009), pp 398 □ 417
[57] Anupam Trivedi, Krishnendu Sanyal, Pranjal Verma, Dipti Srinivasan *"A Unified Differential Evolution Algorithm for Constrained Optimization Problems",* IEEE Congress on Evolutionary Computation (CEC), 07 July 2017
[58] Roohollah Shirani Faradonbeh, Masoud Monjezi1, Danial Jahed Armaghani *"Genetic programing and non-linear multiple regression techniques to predict backbreak in blasting operation",* Engineering with Computers volume 32, pp 123-133 (2016)
[59] Seyedali Mirjalili, Andrew Lewis *"The Whale Optimization Algorithm",* Advances in Engineering Software Volume 95, May 2016, pp 51-67
[60] Amir M. Fathollahi-Fard, Mostafa Hajiaghaei-Keshteli, Reza Tavakkoli-Moghaddam *"Red deer algorithm (RDA): a new nature-inspired meta-heuristic",* Soft Computing volume 24, pp 14637-14665 (2020)
[61] A.A. Heidari, S. Mirjalili, H.Faris et al. *"Harris hawks optimization: Algorithm and applications",* Future Generation Computer Systems Volume 97, August 2019, pp 849-872
[62] Xin-She Yang *"A New Metaheuristic Bat-Inspired Algorithm",* Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp 65-74
[63] Jared Diamond *"Evolution, consequences and future of plant and animal domestication",* Nature volume 418, pp 700-707 (2002)
[64] Natasha J. MEHDIABADI & Ted R. SCHULTZ *"Natural history and phylogeny of the fungus-farming ants (Hymenoptera: Formicidae: Myrmicinae: Attini)",* Myrmecological News volume 13, pp 37-55 (2009)
[65] Sanne Nygaard, Haofu Hu, Cai Li, Morten Schiøtt, Zhensheng Chen, Zhikai Yang, Qiaolin Xie, Chunyu Ma, Yuan Deng, Rebecca B. Dikow, Christian Rabeling, David R. Nash, William T. Wcislo, Seán G. Brady, Ted R. Schultz, Guojie Zhang, Jacobus J. Boomsma *"Reciprocal genomic evolution in the ant–fungus agricultural symbiosis",* Nature Communications volume 7, Article number: 12233 (2016)
[66] Else C. Vellinga *"Ecology and Distribution of Lepiotaceous Fungi (Agaricaceae)",* Nova Hedwigia 78, pp 273-299 (2004)

[67] ULRICH G. MUELLER, STEPHEN A. REHNER, TED R. SCHULTZ "*The Evolution of Agriculture in Ants",* SCIENCE 25 Sep 1998 Vol 281, pp 2034-2038
[68] Ulrich G. Mueller, Nicole M. Gerardo, Duur K. Aanen, Diana L. Six, Ted R. *Schultz "The Evolution of Agriculture in Insects",* Annual Review of Ecology, Evolution, and Systematics Volume 36, pp 563-595 (2005)
[69] Bert Hölldobler, Edward O. Wilson *"The Leafcutter Ants: Civilization by Instinct",* 2011
[70] C. R. Currie, A. N. M. Bot, J. J. Boomsma *"Experimental evidence of a tripartite mutualism: bacteria protect ant fungus gardens from specialized parasites",* 23 April 2003 https://doi.org/10.1034/j.1600-0706.2003.12036.x
[71] Steven Van Borm, Johan Billen, Jacobus J Boomsma *"The diversity of microorganisms associated with Acromyrmex leafcutter ants",* BMC Evolutionary Biology volume 2, Article number: 9 (2002)
[72] Leah Ariniello *"Protecting Paradise: Fungus-farming ants ensure crop survival with surprising strategies and partnerships",* BioScience, Volume 49, October 1999, Pages 760–763,
[73] M. U. V. Ronque, R. M. Feitosa, P. S. *Oliveira "Natural history and ecology of fungus-farming ants: a field study in Atlantic rainforest",* Insectes Sociaux volume 66, pp 375–387 (2019)
[74] Freddie-Jeanne Richard, Michael Poulsen, Falko Drijfhout, Graeme Jones, Jacobus J. Boomsma *"Specificity in Chemical Profiles of Workers, Brood and Mutualistic Fungi in Atta, Acromyrmex, and Sericomyrmex Fungus-growing Ants",* Journal of Chemical Ecology volume 33, pp 2281–2292 (2007)
[75] MICHAEL POULSEN JACOBUS J. BOOMSMA *"Mutualistic Fungi Control Crop Diversity in Fungus-Growing Ants",* SCIENCE 4 Vol 307, Feb 2005, pp. 741-744
[76] Wilson E.O. *"The Insect Societies",* Insect Societies 1971
[77] William O. H. Hughes, Jørgen Eilenberg, Jacobus J. Boomsma *"Trade-offs in group living: transmission and disease resistance in leaf-cutting ants",* 07 September 2002 https://doi.org/10.1098/rspb.2002.2113
[78] Ángel Sahid Aguilar-Colorado, José Rivera-Cháve "Ants/Nest-Associated Fungi and Their Specialized Metabolites: Taxonomy, Chemistry, and Bioactivity", Revista Brasileira de Farmacognosia, 17 July 2023, pp 901-923
[79] Erfan Khosravani Moghadam, Mahdi Vahdanjoo, Allan Leck Jensen, Mohammad Sharifi, Claus Aage Grøn Sørensen *"An Arable Field for Benchmarking of Metaheuristic Algorithms for Capacitated Coverage Path Planning Problems"*, Agronomy Volume 10, 23 September 2020
[80] Ali Kaveh, Yasin Vazirinia *"Construction Site Layout Planning Problem Using Metaheuristic Algorithms: A Comparative Study"*, Iran J Sci Technol Trans Civ Eng (2019) 43:105–115, https://doi.org/10.1007/s40996-018-0148-6
[81] Hsu-Chih Huang *"A Hybrid Metaheuristic Embedded System for Intelligent Vehicles Using Hypermutated Firefly Algorithm Optimized Radial Basis Function Neural Network"* , IEEE Transactions on Industrial Informatics Volume 15, February 2019
[82] Huan Chen, Wei-Yan Chang, Tai-Lin Chiu, Ming-Chao Chiang, Chun-Wei Tsai *"SEFSD: an effective deployment algorithm for fog computing systems",* Journal of Cloud Computing, 15 July 2023 https://doi.org/10.1186/s13677-023-00475-x
[83] Leguizamon, Guillermo, and Zbigniew Michalewicz. "*A new version of ant system for subset problems.",* In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 2, pp. 1459-1464. IEEE, 1999
[84] Nima Khodadadi, Aybike Özyüksel Çiftçioğlu, Seyedali Mirjalili, Antonio Nanni *"A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints",* Decision Analytics Journal Volume 8, September 2023 https://doi.org/10.1016/j.dajour.2023.100266
[85] Reinelt, Gerhard. *"The traveling salesman: computational solutions for TSP applications."*, Springer-Verlag, 1994.
[86] GaigeWang, Lihong Guo, Hong Duan, HeqiWang, Luo Liu, Mingzhen Shao *"A HybridMetaheuristic DE/CS Algorithm for UCAV Three-Dimension Path Planning"*, The ScientificWorld Journal Volume 2012, Article ID 583973, 11 pages doi:10.1100/2012/583973
[87] Abdelazim G. Hussien, Aboul Ella Hassanien, Essam H. Houssein, Mohamed Amin & Ahmad Taher Azar *"New binary whale optimization algorithm for discrete optimization problems",* Engineering Optimization Volume 52, 2020, pp 945-959
[88] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López, J. A. Portilla-Figueras *"The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems",* Research Article Volume 2014, https://doi.org/10.1155/2014/739768
[89] Smaeil Atashpaz-Gargari, Caro Lucas, *"Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition",* 2007 IEEE Congress on Evolutionary Computation, 10.1109/CEC.2007.4425083
[90] R.A. Gallego, R. Romero, A.J. Monticelli *"Tabu search algorithm for network synthesis",* IEEE Transactions on Power SystemsVolume: 15, May 2000, pp 490-495, 10.1109/59.867130
[91] http://www.math.uwaterloo.ca/tsp/world/index.html
[92] Asghari, Ali, and Mohammad Karim Sohrabi. "Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet." *Computer Science Review* 51 (2024): 100616.
[93] Asghari, Ali, Hossein Azgomi, and Zahra Darvishmofarahi. "Multi-Objective edge server placement using the whale optimization algorithm and Game theory." *Soft Computing* (2023): 1-15.
[94] Asghari, Ali, and Mohammad Karim Sohrabi. "Multiobjective edge server placement in mobile-edge computing using a combination of multiagent deep q-network and coral reefs optimization." *IEEE Internet of Things Journal* 9, no. 18 (2022): 17503-17512.

[95] Cheung SO, Tong TKL, *"Site pre-cast yard layout arrangement through genetic algorithms*", Tam CM (2002), Autom Constr 11:35–46. https://doi.org/10.1016/S0926-5805(01)00044-9
[96] Erfan Khosravani Moghadam, Mahdi Vahdanjoo , Allan Leck Jensen, Mohammad Sharifi and Claus Aage Grøn Sørensen "An Arable Field for Benchmarking of Metaheuristic Algorithms for Capacitated Coverage Path Planning Problems", Argonomy, 23 September 2020
[97] R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, *"Introduction to autonomous mobile robots,"* 2nd Edition, MIT Press, London, 2011.
[98] Hsu-Chih Huang, Shao-Kang Lin *"A Hybrid Metaheuristic Embedded System for Intelligent Vehicles Using Hypermutated Firefly Algorithm Optimized Radial Basis Function Neural Network",* IEEE Transactions on Industrial Informatics, Volume: 15, February 2019, pp 1062-1069
[99] Bonomi F, Milito R, Zhu J *"Fog computing and its role in the internet of things. In: Proceedings of Mobile Cloud Computing Workshop",* Addepalli S (2012) , ACM, pp 13–16
[100] Huan Chen, Wei-Yan Chang, Tai-Lin Chiu, Ming-Chao, Chiang, Chun-Wei Tsai *"SEFSD: an effective deployment algorithm for fog computing systems",* Journal of Cloud Computing (2023), https://doi.org/10.1186/s13677-023-00475-x
[101] A. Kaveh, S. Talatahari, N. Khodadadi, *"The hybrid invasive weed optimizationshuffled frog-leaping algorithm applied to optimal design of frame structures",* Period. Polytech. Civ. Eng. 63 (3) (2019) 882–897.
[102] Nima Khodadadi, Aybike Özyüksel Çiftçioğlu, Seyedali Mirjalili, Antonio Nanni "A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints", Decision Analytics Journal, volume 8, September 2023
[103] S. Kurnaz, O. Cetin, and O. Kaynak*, "Adaptive neurofuzzy inference system based autonomous flight control of unmanned air vehicles,"* Expert Systems with Applications, vol. 37, no. 2, pp. 1229–1234, 2010.
[104] Gaige Wang, Lihong Guo, Hong Duan, Heqi Wang, Luo Liu, Mingzhen Shao *"A HybridMetaheuristic DE/CS Algorithm for UCAV Three-Dimension Path Planning",* The ScientificWorld Journal volume 2012, 21 October 2012