

Drone-Base-Station for Next-Generation Internet-of-Things: A Comparison of Swarm Intelligence Approaches

DIMITRIOS PLIATSIOS¹ (Member, IEEE), SOTIRIOS K. GOUDOS² (Senior Member, IEEE), THOMAS LAGKAS³ (Senior Member, IEEE), VASILEIOS ARGYRIOU⁴ (Member, IEEE), ALEXANDROS-APOSTOLOS A. BOULOGEOORGOS⁵ (Senior Member, IEEE), AND PANAGIOTIS SARIGIANNIDIS¹ (Member, IEEE)

¹Department of Electrical and Computer Engineering, University of Western Macedonia, 501 00 Kozani, Greece

²ELEDIA, Department of Physics, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece

³Department of Computer Science, School of Science, International Hellenic University, 570 01 Thessaloniki, Greece

⁴Department of Networks and Digital Media, Kingston University London, Kingston upon Thames KT1 2EE, U.K.

⁵Department of Digital Systems, University of Piraeus, 18534 Piraeus, Greece

CORRESPONDING AUTHOR: S. K. Goudos (e-mail: sgoudo@physics.auth.gr)

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957406 (TERMINET).

ABSTRACT The emergence of next-generation Internet-of-Things (NG-IoT) applications introduces several challenges for the sixth-generation (6G) mobile networks, such as massive connectivity, increased network capacity, and extremely low-latency. To countermeasure the aforementioned challenges, ultra-dense networking has been widely identified as a possible solution. However, the dense deployment of base stations (BSs) is not always possible or cost-efficient. Drone-base-stations (DBSs) can facilitate network expansion and efficiently address the requirements of NG-IoT. In addition, due to their flexibility, they can provide on-demand connectivity in emergency scenarios or address temporary increases in network traffic. Nevertheless, the optimal placement of a DBS is not a straightforward task due to the limited energy reserves and the increased signal quality degradation in air-to-ground links. To this end, swarm intelligence approaches can be attractive solutions for determining the optimal position of the DBS in the three-dimensional (3D) space. In this work, we explore well-known swarm intelligence approaches, namely the Cuckoo Search (CS), Elephant Herd Optimization (EHO), Grey Wolf Optimization (GWO), Monarch Butterfly Optimization (MBO), Salp Swarm Algorithm (SSA), and Particle Swarm Optimization (PSO) and investigate their performance and efficiency in solving the aforementioned problem. In particular, we investigate the performance of three scenarios in the presence of different swarm intelligence approaches. Additionally, we carry out non-parametric statistical tests, namely the Friedman and Wilcoxon tests, in order to compare the different approaches.

INDEX TERMS Drone base station, evolutionary algorithms, mobile communications, optimization methods, swarm intelligence.

I. INTRODUCTION

DRIVEN by the proliferation of the next-generation Internet of Things (NG-IoT), future mobile networks will face an unprecedented increase in demand for computational and networking resources.

NG-IoT is integrated into multiple applications, such as healthcare [1], [2], smart cities [3], and heavy industry [4]. These applications have increased requirements in terms of device connectivity, network capacity, and link latency. Sixth-generation (6G) networks are designed

to efficiently support the heterogeneous networking requirements [5], [6].

A novel concept that can effectively assist in satisfying the increased communication requirements and facilitate network expansion is the drone-base-station (DBS) [7]. DBSs are flexible solutions that aspire to increase user throughput, improve the quality of service (QoS), and expand the coverage of mobile networks. Furthermore, DBSs can provide enhanced network throughput in cases of temporary events, by offloading congestion, or provide connectivity in emergency situations, such as ground BS failures and natural disasters [8]. Finally, DBSs can effectively alleviate the capital and operational expenditure of mobile network operators, as they provide on-demand wireless coverage [9].

As the DBS power is limited, its efficient usage is of utmost importance. Free space pathloss, as well as signal reflection and shadowing, heavily affect the user received signal [10]. To achieve a high QoS, the DBS should be optimally positioned in order to mitigate signal quality degradation. Motivated by this, several researchers have turned their attention to formulating and solving the DBS optimal positioning problems. In particular, Alzenad *et al.* [11] aim to maximize the number of covered users with different QoS requirements. The DBS placement is modeled as a multiple circles placement problem and solved using exhaustive search. The optimal placement of a DBS in a high-rise building scenario is considered in [12]. Due to the intractability of the problem, the Particle Swarm Optimization algorithm was utilized in order to find an efficient DBS position that minimizes the total power that is required to provide coverage to the building users. The authors in [13] formulated the DBS placement problem as a non-linear non-convex combinatorial optimization problem. The problem was decomposed into the DBS placement problem and the joint bandwidth and power allocation problem. The authors developed two heuristic algorithms, namely Dynamic and Fixed DBS placement, based on different placement strategies.

Wang *et al.* [14] proposed an optimal DBS placement method that uses the minimum transmit power to serve a set of ground users. To achieve this, the optimal DBS placement problem was decoupled into two the horizontal and vertical placement sub-problems, respectively. The simulation results showed reduced power consumption for the suburban, urban, and dense urban environments. The joint placement and resource allocation problem is investigated in [15]. The authors decompose the problem into two sub-problems, namely the resource allocation and the DBS placement sub-problem. To find the optimal DBS position, the authors exhaustively search all the possible locations for the one that achieves the highest throughput. The authors in [16] derived analytic expressions for the DBS hovering altitude that optimizes the energy efficiency. Additionally, the horizontal positioning problem was formulated as a circle packing problem for maximum packing density and solved by utilizing the multilevel regular polygon-based placement algorithm. The authors in [17] focused on the energy-efficient

placement optimization problem. To address its complexity, they decoupled it into two separate problems, namely the horizontal placement and the vertical placement problems. A Weiszfeld-based algorithm was leveraged for optimizing the horizontal placement, while the vertical placement was analytically solved using the optimal elevation angle and the radius of the coverage area.

Swarm intelligence approaches have been widely used in solving optimization problems in various wireless applications. The design of design a small-size tag antenna with large gain and reading distance is presented in [18]. Particularly, the design of the tag antenna was generated using the artificial ant colony algorithm. The authors in [19] used the spider monkey optimization approach to optimize a linear antenna array in terms of suppressing the sidelobes and null placement in certain directions. Goudos and Athanasiadou in [20] proposed a modeling method for the prediction of the received signal power at a drone. This proposed method combines the salp swarm algorithm along with five base learners for obtaining accurate predictions. The authors in [21] presented the design of a multiband microstrip patch antenna that was generated with the assistance of the coyote optimization algorithm. A social network optimization method for beam-scanning reflectarray antenna design was presented in [22]. The method was used for optimizing the various antenna radiation patterns, as well as minimizing the difference between the desired and actual radiation direction. Moreover, Luo *et al.* [23] leveraged the Adaptive Opposite Fireworks Algorithm in order to suppress the interference caused by the main lobe in radar systems. In [24], the authors developed a beamforming technique that is able to achieve fast data updates using the conformal phased array radar system. A modified particle swarm optimization algorithm was used to generate the beamforming weights of the antennas to achieve an optimal radiation pattern. Chen *et al.* [25] used the particle swarm optimization method to optimize the parameters of a deep neural network in order to optimally map the configuration parameters of meta-atoms to the antenna reflection coefficients.

The complexity of the DBS placement problem has been established in the aforementioned works. In most of these, the authors aimed to tackle the problem by decoupling it into smaller and more tractable problems in order to obtain a near-optimal solution. Also, in some works, the authors resorted to exhaustive search methods. It is apparent that swarm intelligence approaches can be directly applied to the DBS positioning problem. Therefore, in this work, we aim to evaluate the performance of well-known swarm intelligence approaches in terms of finding the optimal DBS location under various objectives and parameters. In this direction, we consider three scenarios involving single and multiple DBSs providing wireless coverage to a number of IoT devices that are deployed in a particular area. The contributions of this work are summarized as follows:

- We formulate two optimization problems associated with the performance of the system, namely the

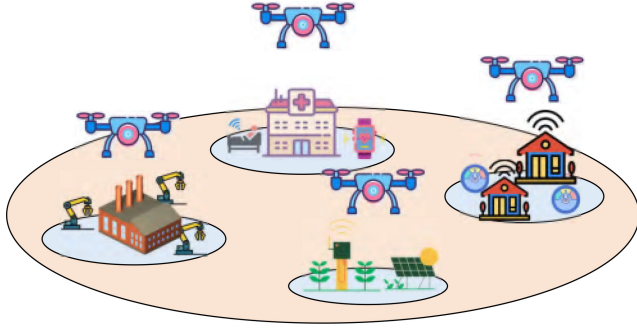


FIGURE 1. Architectural Concept.

minimization of the average pathloss experienced by the devices and the maximization of the device coverage. Due to the complexity and the large search space of the problems, swarm intelligence approaches can prove valuable assets.

- We provide an overview of five swarm intelligence approaches, namely the Cuckoo Search (CS), the Grew Wolf Optimization (GWO), the Monarch Butterfly Optimization (MBO), the Elephant Herd Optimization (EHO), the Salp Swarm Algorithm (SSA), and the Particle Swarm Optimization (PSO). Also, we present the main functionality of each approach.
- We compare the swarm intelligence approaches in terms of performance in finding the optimal DBS position, that optimizes the two optimization problems. The approaches are compared in three evaluation scenarios. In the first two scenarios, we aim to minimize the average pathloss using single and multiple DBSs, respectively. On the other hand, in the third scenario, we aim to maximize the device coverage using multiple DBSs.
- Additionally, we perform non-parametric statistical tests, namely the Friedman and Wilcoxon tests, in order to further evaluate the performance of these approaches. Finally, we rank the swarm intelligence approaches, based on the results of these tests.

The rest of the paper is organized as follows. Section II presents the system model, while Section III provides a background of swarm intelligence approaches. The evaluation of swarm approaches is presented in Section IV, while Section V presents an analysis of the pathloss distribution, along with a discussion of the evaluation results. Finally, Section VI concludes this work.

II. SYSTEM MODEL AND PROBLEM FORMULATION

The considered system model is presented in Fig. 1. A number of DBSs, denoted by $\mathcal{K} = \{1, \dots, K\}$, provide connectivity to a number of IoT devices, denoted by $\mathcal{S} = \{1, \dots, S\}$. The devices are randomly deployed over a geographical area, while the DBSs hover over the devices.

Due to the hovering altitude of the DBS, the conventional channel models are not sufficient for modeling the links

between the DBS and the ground devices. According to the AtG model presented in [26], the Line of Site (LoS) probability of an AtG link between the k -th DBS and the s -th device is calculated as

$$P(h_k, d_{k,s}) = \frac{1}{1 + \alpha \exp\left[-\beta \left(\arctan\left(\frac{h_k}{d_{k,s}}\right) - \alpha\right)\right]} \quad (1)$$

where α , and β are coefficients based on the environment (e.g., urban, suburban, etc.). h_k denotes the altitude of the DBS, while the horizontal distance between the k -th DBS and the s -th device is denoted by $d_{k,s} = \sqrt{(x_k - x_s)^2 + (y_k - y_s)^2}$, where (x_k, y_k) and (x_s, y_s) are the horizontal locations the DBS and the device, respectively.

Therefore, considering the LoS and Non-LoS (NLoS) probabilities, the pathloss can be obtained by

$$PL(h_k, d_s) = 20 \log\left(\sqrt{h_k^2 + d_{k,s}^2}\right) + AP(h_k, d_{k,s}) + B \quad (2)$$

where $A = \eta_{LoS} - \eta_{NLoS}$ and $B = 20 \log\left(\frac{4\pi f_c}{c}\right) + \eta_{NLoS}$, f_c is the carrier frequency (in Hz), c is the speed of light, while η_{LoS} and η_{NLoS} are the mean additional losses based on the propagation environment.

A. MINIMIZING THE AVERAGE PATHLOSS

According to Shannon's formula, the data rate of a device increases as its respective signal power is increased, assuming that the system bandwidth and noise remain constant. The received power at a ground device depends on the pathloss experienced by the wireless channel can be calculated as

$$P_s^r = P_k^t - PL(h_k, d_s) - P_N \quad (3)$$

where P_k^t is the transmission power of the k -th DBS and P_N the power of the Additive White Gaussian Noise (AWGN). The AWGN power will essentially be the same for all the devices located in a particular area. Therefore, the impact of the noise on determining the optimal D-RRH location will be nominal. Without loss of generality, we assume that $P_N = 0$.

Given a constant transmission power P_k^t , the received power can be increased by minimizing the distance-dependent pathloss between DBS and the devices. In this direction, we optimize the DBS placement so as to minimize the average pathloss experienced by the devices. Assuming that each device is connected to the closest DBS, the optimization problem is formulated as

$$\underset{\{x,y,h\}}{\text{minimize}} \quad \frac{\sum_{k=1}^K \sum_{s=1}^S PL(h_k, d_s)}{S} \quad (4a)$$

$$\text{subject to: } C1: x_{min} \leq x_D^k \leq x_{max}, \quad \forall k \quad (4b)$$

$$C2: y_{min} \leq y_D^k \leq y_{max}, \quad \forall k \quad (4c)$$

$$C3: h_{min} \leq h_D^k \leq h_{max}, \quad \forall k \quad (4d)$$

where vectors $\mathbf{x}, \mathbf{y}, \mathbf{h}$ denote the position of each DBS in the three-dimensional (3D) space, while x_{min}/x_{max} , y_{min}/y_{max} , and z_{min}/z_{max} are the area limits.

TABLE 1. Summary of swarm intelligence approaches.

Approach	Main Principles	Relevant References
Cuckoo Search	Based on the parasitic reproductive behavior of Cuckoos. The search for a nest is based on Levy flights. Maintains a balance between local and global random flights through Levy flight parameters.	[30], [31], [32], [33]
Grey Wolf Optimization	Mimics the leadership hunting mechanisms of grey wolves. Fast convergence due to the limited search randomness. The limited randomness may lead to local optima stagnation.	[34], [35], [36], [37]
Monarch Butterfly Optimization	Mimics the migration of a particular species of butterflies. It uses two operators, namely the migration operator and adjusting operator. The exploration of search space can be adjusted through these operators.	[38], [39], [40], [41]
Elephant Herd Optimization	Inspired by the herding behavior of an elephant group. Based on two operators, namely the clan updating and separating operators. The exploration of search space can be adjusted through these operators.	[42], [43], [44], [45]
Salp Swarm Algorithm	Based on the swarming behavior of salps forming a chain. The first salp in the chain is the leader, while the rest are followers. A coefficient is used to adjust the trade-off between exploration and exploitation.	[46], [47], [48], [49]
Particle Swarm Optimization	Inspired by the flocking behavior of birds in search of food. The position of each particle is updated based on its own knowledge and the swarm's knowledge of the best solution location. Random variables and coefficients are introduced in order to avoid optima stagnation.	[50], [51], [52], [53]

B. MAXIMIZING DEVICE COVERAGE

A device is considered to be under coverage if its QoS can be satisfied. This can be alternatively expressed as the device experiencing pathloss that is lower than a pre-defined threshold, denoted by T . Therefore, the corresponding optimization problem is formulated as

$$\underset{\{x,y,h\}}{\text{maximize}} \sum_{k=1}^K \sum_{S=1}^S C_{k,s} \quad (5a)$$

$$\text{subject to: } C1: x_{min} \leq x_D^k \leq x_{max}, \quad \forall k \quad (5b)$$

$$C2: y_{min} \leq y_D^k \leq y_{max}, \quad \forall k \quad (5c)$$

$$C3: h_{min} \leq h_D^k \leq h_{max}, \quad \forall k \quad (5d)$$

where $C_{k,s}$ is expressed as follows:

$$C_{k,s} = \begin{cases} 1, & \text{PL}(h_k, d_s) \leq T \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The solutions to the aforementioned optimization problems are challenging. To this end, swarm intelligence can be leveraged in order to find an efficient solution.

III. SWARM INTELLIGENCE

Swarm intelligence approaches have received great focus from multiple domains, such as science, engineering, and industry [27], [28], [29]. Swarm intelligence is a branch of computational intelligence concerned with the collective self-organizing behavior of a species' population. In this section, we present well-known swarm intelligence approaches, namely the CS, GWO, MBO, EHO, SSA, and PSO. The main operating principles of these approaches along with the relevant references are summarized in Table 1.

A. CUCKOO SEARCH

The CS algorithm is based on the breeding behavior of certain species of cuckoos [30]. There exist three types of breeding, namely intraspecific brood parasitism, cooperative breeding, and nest takeover. In the first type of breeding, a bird places its eggs on the nest of another bird. If the nest owner discovers that the eggs are not their own, they throw them away or abandon the nest. In addition, some species have evolved and became specialized in the mimicry of the colors and patterns of other birds' eggs. This reduces the probability of their eggs being abandoned increasing, therefore, the survival rate.

The mathematical model of the CS breeding is based on the rules below:

- The number of nests is fixed.
- Each cuckoo lays one egg at a time in a random nest of another bird.
- A foreign egg is discovered with a probability of $p \in [0, 1]$
- The eggs that survive will carry over to the next generation.

Algorithm 1 shows the CS procedure.

B. GREY WOLF OPTIMIZATION

The GWO algorithm mimics the leadership, hierarchy, and hunting mechanism of grey wolves in nature [34]. There exist four types of grey wolves, namely alpha, beta, delta, and omega, that are employed for simulating the leadership hierarchy. Grey wolf hunting involves three main phases, namely a) tracking and approaching the prey, b) pursuing and encircling the prey until it stops moving, c) attacking the prey.

Algorithm 1 Cuckoo Search

```

1: Generate initial population of nests  $x_i (i = 1 \dots n)$ 
2: while  $t < MaxGen$  do
3:   Select a cuckoo randomly using Levy flights [33], [54]
4:   Evaluate its fitness  $F_i$ 
5:   Choose a random nest  $x_j (j = rand(1, n))$ 
6:   if  $F_i > F_j$  then
7:     replace  $j$  by the new solution
8:   end if
9:
10: end while
    
```

The GWO algorithm is mathematically modeled based on the aforementioned hierarchy and hunting method. Therefore, the fittest solution is considered the alpha, while the second and third best solutions are the beta and delta, respectively. The rest of the candidate solutions are assumed to be of the omega type. The alpha, beta, and delta lead the optimization, while the omega follow.

The positions of candidate solutions (wolves) vary during optimization around α , β , and δ as:

$$\vec{D} = \left| \vec{C} \vec{X}_p(t) - \vec{X}(t) \right| \quad (7)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \vec{D} \quad (8)$$

where t indicates the current iteration, \vec{A} and \vec{C} are coefficient vectors, \vec{X}_p is the position vector of the prey, and \vec{X} indicates the position vector of the wolf.

The vectors \vec{A} and \vec{C} are calculated as:

$$\vec{A} = 2\vec{a} \vec{r}_1 - \vec{a} \quad (9)$$

$$\vec{C} = 2\vec{r}_2 \quad (10)$$

where components of \vec{a} are linearly decreased from 2.0 to 0 throughout the iterations, while r_1 , and r_2 are randomly selected in $[0, 1]$.

Grey wolves recognize the location of potential prey and encircle it. This is usually guided by the alpha, while the beta and delta can also participate. However, in a wide search space, the alphas have no clues about the location of the optimum (prey). To mathematically model this hunting behavior, it is assumed that the alpha, beta, and delta have better knowledge about the potential location of prey. Therefore, the first three best solutions are imposed on the other search agents (including the omegas) to update their respective positions according to the position of the best search agents. To this end, the following formulas are derived

$$\begin{aligned} \vec{D}_\alpha &= \left| \vec{C}_1 \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta &= \left| \vec{C}_2 \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta &= \left| \vec{C}_3 \vec{X}_\delta - \vec{X} \right| \\ \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta \end{aligned} \quad (11)$$

Algorithm 2 Grey Wolf Optimization

```

1: Initialize the grey wolf population  $X_i (i = 1, 2, \dots, n)$ 
2: Initialize  $\alpha$ ,  $A$ , and  $C$ 
3: Calculate the fitness of each search agent:
    $X_\alpha$ : the best search agent
    $X_\beta$ : the second best search agent
    $X_\delta$ : the third best search agent
4: while  $t < t^{max}$  do
5:   for  $X_i = 1$  to  $n$  do
6:     Update the position of the search agent using (11)
7:   end for
8:   Update  $\alpha$ ,  $A$ , and  $C$ 
9:   Calculate the fitness of all search agents
10:  Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
11:   $t = t + 1$ 
12: end while
13: return  $X_\alpha$ 
    
```

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \vec{D}_\delta \quad (12)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (13)$$

At the final phase of the hunt, the wolves attack their prey by moving in. Mathematically, this is translated as reducing the value of \vec{a} . \vec{A} is a random value in the range $[-2\alpha, 2\alpha]$, while α is decreased from 2 to 0 throughout the iterations. When the value of \vec{A} is in $[-1, 1]$ range, the next position of the search agent is any position between the current position and the optimal position.

A grey wolf's search space is dominated by the positions of alpha, beta, and delta. These diverge from each other in order to search for prey and converge when they attack it. This divergence is modeled by utilizing $\vec{A} \in (-\infty, -1) \cup (1, \infty)$ in order to force the search agent to diverge from the prey. Additionally, the exploration is also imposed by the \vec{C} term. Specifically, this term provides random weights so that the impact of the prey during the search is emphasized ($C > 1$) or de-emphasized ($C < 1$). This forces the GWO to feature a more random behavior that favors exploration instead of converging to potential local optima.

Algorithm 1 summarizes the GWO process. The search starts with generating a random population of grey wolves (candidate solutions). Throughout the iterations alpha, beta, and delta wolves estimate the probable position of the prey (optimum solution). The parameter α is decreased from 2 to 0 in order to emphasize exploration. Candidate solutions tend to diverge from the prey if $|\vec{A}| > 1$ or converge towards the prey if $|\vec{A}| < 1$. The GWO process is terminated when the maximum number of iterations is reached.

C. MONARCH BUTTERFLY OPTIMIZATION

MBO is a meta-heuristic algorithm that mimics the migration of butterflies from the northern United States to Mexico. MBO consists of two main processes [39]. The first process mimics the way a number of butterflies move from their

current to a new position based on the migration operator. In the second process, the algorithm tunes the position of the rest of the butterflies by adjusting the migration operator.

To model the behavior of the monarch butterflies, the following rules are established:

- The whole population is divided into two locations, namely land A and land B.
- A child butterfly is born by the migration operator from a parent either in land A or land B.
- A newborn butterfly will take the place of its parent if it has better fitness.
- The butterflies featuring the best fitness will carry over to the next generation, and cannot be modified by any operators. This guarantees that the butterfly population will not be eliminated after many generations.

The number of butterflies in each region is denoted by:

$$NP_1 = \lceil pNP_{tot} \rceil \quad (14)$$

$$NP_2 = NP_{tot} - NP_1 \quad (15)$$

where NP_1 and NP_2 are the sub-populations in land A and B, respectively, while NP_{total} is the total population number and p is the ratio of monarch butterflies in land A.

The value $r = 1.2rand()$ is introduced, where $rand()$ denotes a uniform random number. If $r \leq p$, a new butterfly is generated using (16).

$$x_{i,k}^{t+1} = x_{r_1,k}^t \quad (16)$$

where $x_{i,k}^{t+1}$ indicates the k -th element of x_i at generation $t + 1$ that corresponds to the position of the butterfly i . Similarly, $x_{r_1,k}^t$ indicates the k -th element of x_{r_1} that is the new position of butterfly r_1 . Butterfly r_1 is randomly selected from sub-population A (i.e., the one located in land A).

If $r > p$, a butterfly is generated using (17).

$$x_{i,k}^{t+1} = x_{r_2,k}^t \quad (17)$$

where $x_{r_2,k}^t$ indicates the k -th element of x_{r_2} at generation t that is a newly generated position r_2 .

It is apparent that the p ratio affects the direction of the migration operator. If p is large, more butterflies in land A will be selected, and thus, more butterflies from sub-population A will carry over to the next generations. Similarly, if p is small, more butterflies in land B will be selected and sub-population B will mainly affect the population of new butterflies. In this work, p is set to 5/12. Algorithm 3 represents the migration operator process.

Apart from the migration operator, the positions of butterflies can be also updated by the adjusting operator, as follows. For all elements in monarch butterfly j , if $rand() \leq p$, the butterfly is updated as:

$$x_{j,k}^{t+1} = x_{best,k}^t \quad (18)$$

where $x_{best,k}^t$ represents the k -th element of x_{best} that is the best butterfly in land A and land B, at generation t .

Algorithm 3 MBO Migration Operator

```

1: for  $i = 1$  to  $NP_1$  do
2:   for  $k = 1$  to  $D$  do
3:     Randomly generate a random number uniformly
       distributed:
        $r = rand()period$ 
4:     if  $r \leq p$  then
5:       Randomly select a monarch butterfly in sub-
       population A
6:       Generate the  $k$ -th element of  $x_i^{t+1}$  using (16)
7:     else
8:       Randomly select a monarch butterfly in sub-
       population B
9:       Generate the  $k$ -th element of  $x_i^{t+1}$  using (17)
10:    end if
11:  end for
12: end for
13: return  $x_i^{t+1}$ 

```

On the other hand, if $rand() > p$, the butterfly is updated as:

$$x_{j,k}^{t+1} = x_{r_3,k}^t \quad (19)$$

where $x_{r_3,k}^t$ indicates the k -th element of x_{r_3} that is randomly selected in land B (i.e., $r_3 \in \{1, 2, \dots, NP_2\}$).

Under this condition, if $rand() < BAR$, where BAR is the butterfly adjusting rate, it can be further updated as:

$$x_{j,k}^{t+1} = x_{j,k}^{t+1} + \alpha \times (d_{x_k} - 0.5) \quad (20)$$

where d_x is the walk step of butterfly j that can be calculated by the Levy flight [33], [54] and α is the weighting factor given by:

$$\alpha = \frac{S_{max}}{l^2} \quad (21)$$

where S_{max} is the maximum walk step. Therefore, a bigger α corresponds to a longer search step and, thus, increases the exploration process. On the other hand, a smaller α decreases the step search and encourages the exploitation process. The process of the adjusting operator is shown in Algorithm 4.

D. ELEPHANT HERD OPTIMIZATION

EHO is a swarm-based meta-heuristic search approach, inspired by the herding behavior of an elephant group [42]. Male elephants often live in isolation, while females live in family groups. An elephant group is a complicated social structure that consists of multiple clans under the leadership of the oldest female elephant.

The elephant behavior is modeled using two operators, namely the clan updating operator and the separating operator. The global optimum is achieved through the optimization of these two operators. The following assumptions are established:

- The whole population consists of several clans, where each clan has a fixed number of elephants.

Algorithm 4 MBO Adjusting Operator

```

1: for  $j = 1$  to  $NP_2$  do
2:   Calculate the walk step  $d_x$  using the Levy flight
   method
3:   Calculate the weighting factor using (21)
4:   for  $k = 1$  to  $D$  do
5:     Randomly generate a random number uniformly
     distributed:
      $r = rand()period$ 
6:     if  $r \leq p$  then
7:       Randomly select a monarch butterfly in sub-
       population A
8:       Generate the  $k$ -th element of  $x_j^{t+1}$  using (18)
9:     else
10:      Randomly select a monarch butterfly in sub-
      population B
11:      Generate the  $k$ -th element of  $x_j^{t+1}$  using (19)
12:      if  $rand() > BAR$  then
13:         $x_{j,k}^{t+1} = x_{j,k}^{t+1} + \alpha \times (d_{x_k} - 0.5)$ 
14:      end if
15:    end if
16:  end for
17: end for
18: return  $x_i^{t+1}$ 
    
```

- At each generation, a fixed number of male elephants will leave their group and live in solitary
- The elephants in each clan live together under the leadership of the oldest female.

The first operator, namely the clan updating operator, is focused on updating the position of each elephant in the clan, depending on the position of the leader. Therefore, the position of that elephant is denoted by:

$$x_{new,ci,j} = x_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \quad (22)$$

where $x_{new,ci,j}$ and $x_{ci,j}$ are the new and old position of elephant j in clan ci , respectively. $\alpha \in [0, 1]$ is a scale factor that determines the leader influence. Finally, r is a uniform random number in $[0, 1]$.

The fittest elephant in each clan is updated by:

$$x_{new,ci,j} = \beta \times x_{center,ci} \quad (23)$$

where $\beta \in [0, 1]$ is a factor that determines the influence of $x_{center,ci}$ on $x_{new,ci,j}$. $x_{center,ci}$ is considered the center of clan ci and is calculated as:

$$x_{center,ci} = \frac{1}{n_{ci}} \sum_{j=1}^{n_{ci}} x_{ci,j,d} \quad (24)$$

where d indicates the dimension, n_{ci} is the number of elephants in clan ci , and $x_{ci,j,d}$ is the position of elephant j in the d -dimension.

The clan updating operator is summarized in Algorithm 5.

The male elephants will leave their group and live in isolation. It is assumed, that all elephants featuring the worst

Algorithm 5 EHO Clan Updating Operator

```

1: for  $ci = 1$  to  $nClan$  (for all clans in the population)
   do
2:   for  $j = 1$  to  $n_{ci}$  do
3:     Update  $x_{ci,j}$  and generate  $x_{new,ci,j}$  using (22)
4:     if  $x_{ci,j} = x_{best,ci}$  then
5:       Update  $x_{ci,j}$  and generate  $x_{new,ci,j}$  using (23)
6:     end if
7:   end for
8: end for
    
```

Algorithm 6 EHO Separating Operator

```

1: for  $ci = 1$  to  $nClan$  (for all clans in the population)
   do
2:   Replace the elephant with the worst fitness using (25)
3: end for
    
```

Algorithm 7 Elephant Herd Optimization

```

1: Initialization:
2: Set generation counter  $t = 1$ 
3: Initialize population
4: Initialize maximum number of generations  $MaxGen$ 
5: Main:
6: while  $t < MaxGen$  do
7:   Sort the elephants according to their fitness
8:   Perform clan updating operator (Algorithm 5)
9:   Perform separating operator (Algorithm 6)
10:  Evaluate population using the updated positions
11:   $t = t + 1$ 
12: end while
    
```

fitness at each generation will leave their respective groups. This behavior is modeled by the separating operator as:

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times r \quad (25)$$

where x_{max} and x_{min} are, respectively, the upper and lower bound of the elephant's position, while $x_{worst,ci}$ is the elephant with the worst fitness in clan ci . Finally, r is a uniform random number in $[0, 1]$.

Algorithm 6 describes the separating operator procedure, while Algorithm 7 summarizes the whole EHO process.

E. SALP SWARM ALGORITHM

Salps belong to the family of Salpidae and are very similar to jellyfishes. In deep oceans, salps form a swarm called a salp chain. SSA mimics the swarming behavior of Salps [46].

To mathematically model a salp chain, it is assumed that the whole population consists of two parts, namely the leading salp and the following salps. The leader is the salp at the top of the chain and guides the swarm, while the rest of the salps compose the followers.

The n variables of an optimization problem correspond to the positions of salps in an n -dimensional search space. The swarm's target is to search for the food source, denoted

Algorithm 8 Salp Swarm Algorithm

```

1: Initialize the salp population  $x_i (i = 1 \dots n)$  considering
    $ub$  and  $lb$ 
2: while termination condition not met do
3:   Calculate the fitness of each search agent (salp)
4:    $F$  = the best search agent
5:   Update  $c_1$  using (27)
6:   for  $x_i = 1$  to  $n$  do
7:     if  $i = 1$  then
8:       Update the position of leading salp using (26)
9:     else
10:      Update the position of the follower salp using
        (28)
11:    end if
12:  end for
13:  Fix the salp positions based on the lower and upper
   bounds
14: end while
15: return  $F$ 

```

by F . The position update of the leader is determined by:

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 > 0 \end{cases} \quad (26)$$

where x_j^1 denotes the position of the leader in the j -dimension, while F_j denotes the position of the food source in the j -dimension. ub_j and lb_j indicate the upper and lower bound of j -dimension, respectively. Finally, c_1 , c_2 , and c_3 are random numbers.

Particularly, coefficient c_1 is the most important parameter, since it balances exploration and exploitation, and is calculated as:

$$c_1 = 2 \exp\left(-\left(\frac{4l}{L}\right)^2\right) \quad (27)$$

where l and L correspond to the current and maximum iterations, respectively. Parameters c_2 and c_3 are random numbers uniformly distributed in $[0, 1]$. c_2 is the step size, while c_3 dictates if the next position is towards positives or negatives.

The positions of each follower salp i in the j -dimension is calculated by:

$$x_j^i = \frac{1}{2}at^2 + v_0t, \forall i \geq 2 \quad (28)$$

where t is the time, v_0 is the initial speed, and $a = \frac{v_{final}}{v_0}$.

Throughout the optimization, the time corresponds to the iterations, therefore, considering $v_0 = 0$ the above equation can be expressed as:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}), \forall i \geq 2 \quad (29)$$

The whole SSA is summarized in Algorithm 8.

F. PARTICLE SWARM OPTIMIZATION

PSO is an optimization approach based on the flocking behavior of birds in search of food [50]. A swarm of particles

Algorithm 9 Particle Swarm Optimization

```

1: Initialization:
2: initialize each particle's position and calculate  $x_{best}^i$ 
3: calculate the swarm's knowledge of the best solution
   location  $x_{best}^{global}$ 
4: Set generation counter  $j = 1$ 
   Main:
5: while  $j < maxGen$  do
6:   for each particle do
7:     update the particle position (30)
8:     evaluate the fitness function in that position
9:     update the particle's knowledge of the best solution
       location  $x_{best}^i$ 
10:    update the swarm's knowledge of the best solution
       location  $x_{best}^{global}$ 
11:    Set  $j = j + 1$ 
12:   end for
13: end while
14: return  $x_{best}^{global}$ 

```

is placed at random initial positions. These particles move in the search-space and exchange their knowledge about the optimal solution.

In each generation, the position of each particle is updated according to two variables. The first variable corresponds to the particle's own knowledge about the location of the solution, while the second variable corresponds to the swarm's knowledge of the solution location.

According to the PSO operation, a particle's new position can be obtained as

$$x_i^{j+1} = x_i^j + wv_i^j + c_1r_1(x_{best}^i - x_i^j) + c_2r_2(x_{best}^{global} - x_i^j) \quad (30)$$

where i denotes the particle, while j denotes the generation. Additionally, w is the inertia parameters, while c_1 and c_2 are the coefficients adjusting the exploration. Moreover, r_1 and r_2 are random values in range $[0, 1]$, while x_{best}^i and x_{best}^{global} denote the particle's and swarm's knowledge of the solution location, respectively. The PSO algorithm is summarized in Algorithm 9.

IV. NUMERICAL RESULTS

To evaluate the performance of each of the aforementioned algorithms, we carried out extensive Monte Carlo simulations and the final results are generated by averaging 1000 simulations. Four propagation environments are considered namely the suburban, urban, dense urban, and high-rise urban environments. The propagation parameters of each environment are presented in Table 2.

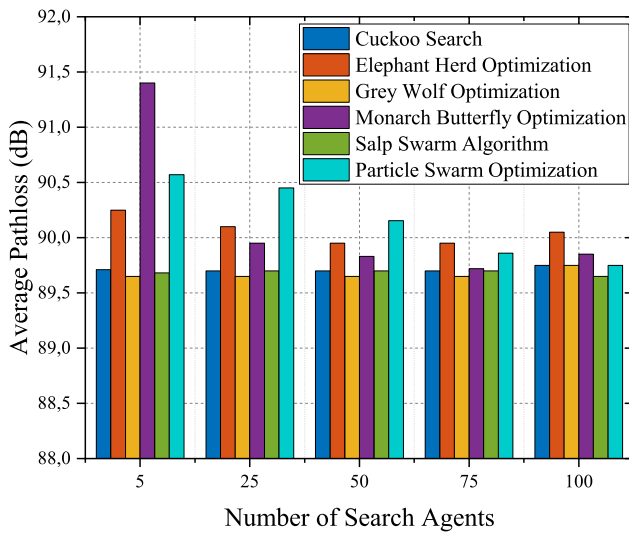
We consider three scenarios for the evaluation of the swarm intelligence algorithms. In the first scenario, we assume that a single DBS is hovering over the devices and we evaluate the performance of the algorithms in terms of minimizing the average pathloss under various parameters, such

TABLE 2. Propagation parameters in different environments.

Environment	α	β	η_{LoS}	η_{NLoS}
Urban	9.61	0.16	1	20
Suburban	4.88	0.43	0.1	21
Dense Urban	12.08	0.11	1.6	23
High-rise Urban	27.23	0.08	2.3	34

TABLE 3. Scenario 1 simulation parameters.

Parameter	Value
Area	1 km ²
Number of devices	10, 20, 30, 40, 50
Search agents	5, 25, 50, 75, 100
Number of generations	10, 50, 100, 200, 500
Carrier Frequency	2.0 GHz
Environments	Urban, Suburban, Dense Urban, High-rise Urban

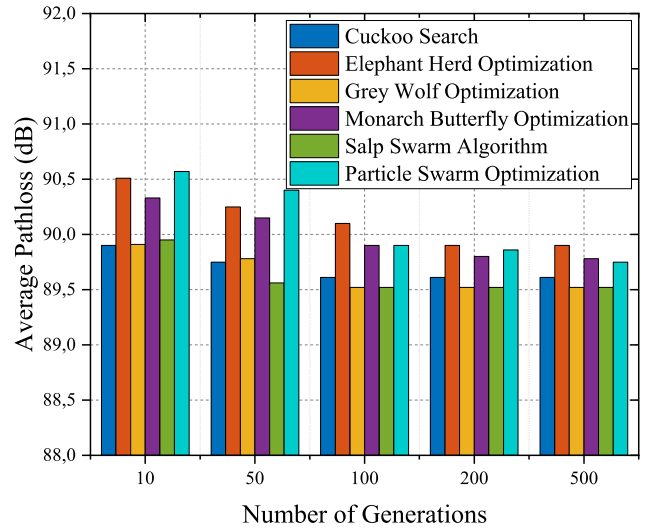

FIGURE 2. Average pathloss as a function of number of search agents.

as the number of devices, number of search agents, number of generations, and the propagation environment. In the second and third scenarios, we assume that multiple DBSs are hovering over a number of devices deployed throughout a large area. Particularly, in the second scenario we evaluate the swarm intelligence algorithms towards minimizing the average pathloss for various numbers of DBSs, while in the third scenario, we evaluate these algorithms towards maximizing the coverage for various numbers of DBSs.

A. SCENARIO 1: AVERAGE PATHLOSS USING A SINGLE DBS

In this scenario, a single DBS is hovering a number of devices in a 1 km² area, while the number of uniformly distributed ground devices is set to 10, 20, 30, 40, and 50. The number of search agents (population) and maximum generations in each algorithm are respectively set to 5, 25, 50, 75, 100 and 10, 50, 100, 200, 500. The simulation parameters for the first scenario are summarized in Table 3.

Fig. 2 shows the average pathloss as a function of the number of search agents when there are 20 devices deployed


FIGURE 3. Average pathloss as a function of the maximum number of generations.

in an urban environment, while the maximum number of generations is 100. CS, EHO, GWO, and SSA feature almost the same respective performance, regardless of the number of search agents. Specifically, CS, GWO, and SSA feature the best performance, while EHO features a slightly worse performance. On the other hand, the number of search agents affects the performance of the MBO and PSO approaches. In particular, MBO performs better when the number of search agents is increased.

Fig. 3 shows the average pathloss as a function of the maximum generations, when the numbers of search agents and devices are 25 and 20, respectively, while an urban environment is considered. According to the results, when the number of generations is 100, 200, and 500, the algorithms feature a similar performance. On the other hand, the algorithms perform worse when there are only 10 and 50 generations. This is expected, as the optimum is more closely approximated when more generations are used. Overall, CS, GWO, and SSA have the same performance, while PSO and MBO have the worst and second worst, respectively.

The average pathloss in various propagation environments is shown in Fig. 4. The numbers of maximum generations and search agents are 100 and 25, respectively. An urban area with 20 devices is considered as the deployment area. It is apparent that the propagation environment has a significant impact on the pathloss experienced by the devices. This is expected due to the denser existence of buildings in the deployment area. Specifically, the suburban environment features the lowest pathloss as there are very few buildings. On the other hand, the high-rise urban and dense urban environments feature the highest and second highest pathloss, respectively. In addition, CS, GWO, and SSA feature a similar overall performance in all cases.

Fig. 5 depicts the average pathloss as a function of the number of devices in the deployment area. The numbers of maximum generations and search agents are 100 and 25,

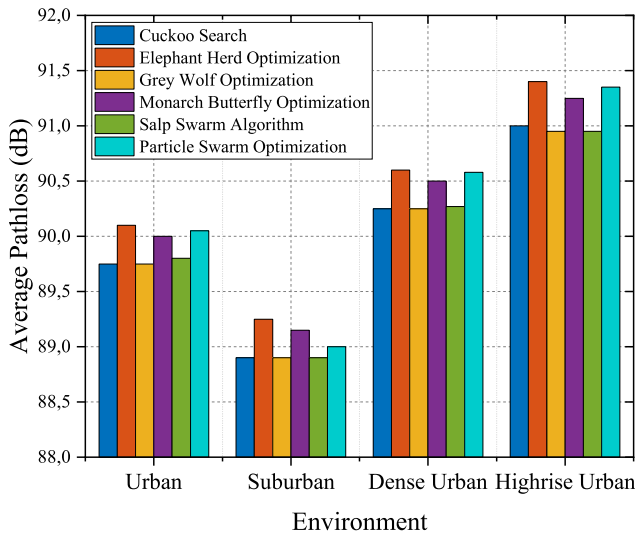


FIGURE 4. Average pathloss in various propagation environments.

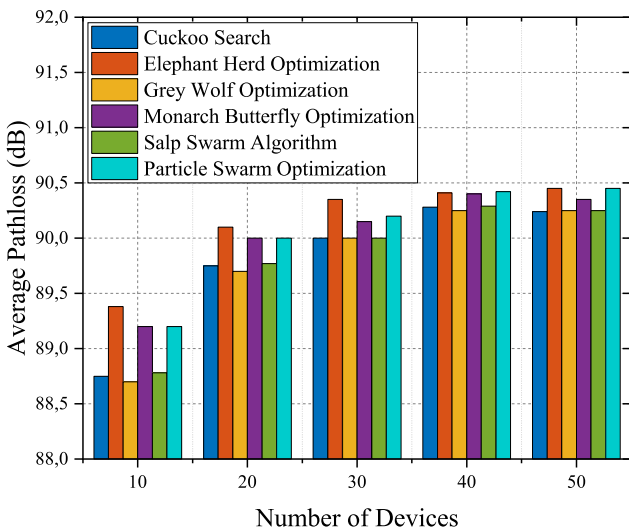


FIGURE 5. Average pathloss as a function of the number of devices.

respectively. An urban area is considered as the deployment area. According to the results, as the number of devices increases, the pathloss is increased because the close DBS deployment to devices is more complicated. CS, GWO, and SSA feature a similar overall performance in all cases.

B. SCENARIO 2: AVERAGE PATHLOSS USING MULTIPLE DBSS

In this scenario, we evaluate the coverage probability over a number of devices deployed in a larger area, while multiple DBSS provide connectivity. A device is considered under coverage if its experienced pathloss is lower than a threshold. The area size is 5 km^2 , the number of devices is 100, and the number of DBSS ranges from 1 to 10. Also, the threshold is set to 90, 100, 110, and 120 dB. Moreover, the number of search agents (population) and maximum generations in

TABLE 4. Scenario 2 simulation parameters.

Parameter	Value
Area	5 km^2
Number of devices	100
Search agents	25
Number of generations	100
Carrier frequency	2.0 GHz
Environments	Urban, Suburban, Dense Urban, High-rise Urban

TABLE 5. Scenario 3 simulation parameters.

Parameter	Value
Area	5 km^2
Number of devices	100
Threshold	90, 100, 110, 120 dB
Search agents	25
Number of generations	100
Carrier frequency	2.0 GHz
Environments	Urban, Suburban, Dense Urban, High-rise Urban

each algorithm are 25 and 100, respectively. The simulation parameters for the second scenario are summarized in Table 4.

The average pathloss as a function of the number of DBSS for various environments is depicted in Fig. 6. In all cases, when the number of DBSS is increased, the average pathloss is decreased. This is expected as when there are more DBSS, they can be deployed closer to the devices, thus, reducing the experienced pathloss. According to the results, GWO has the best overall performance, while MBO features the worst overall performance in all cases. EHO, CS, SSA, and PSO feature a similar performance. Furthermore, as also deduced in the first scenario, the suburban and high-rise urban environments have the lowest and highest average pathloss, respectively.

C. SCENARIO 3: COVERAGE PROBABILITY USING MULTIPLE DBSS

In this scenario, we evaluate the coverage probability over a number of devices deployed in a larger area, while multiple DBSS provide connectivity. A device is considered under coverage if its experienced pathloss is lower than a threshold. The area size is 5 km^2 , the number of devices is 100, and the number of DBSS ranges from 1 to 10. Also, the threshold is set to 90, 100, 110, and 120 dB. Moreover, the number of search agents (population) and maximum generations in each algorithm are 25 and 100, respectively. The simulation parameters for the third scenario are summarized in Table 5.

The coverage probability as a function of the number of DBSS for various thresholds is presented in Fig. 7–Fig. 10. As expected, when the number of DBSS is increased, the coverage probability is also increased, as the DBSS can be spread throughout the area to cover more devices. As far as the threshold is concerned, lower threshold values correspond to lower coverage probability. This is expected as the devices that are closer to the DBSS will experience a lower pathloss lower than the threshold. Overall, when the threshold is set to

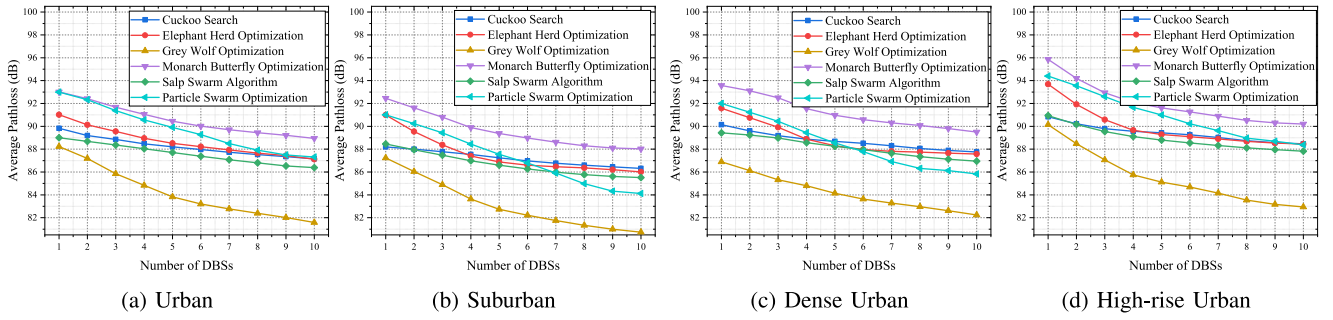


FIGURE 6. Average pathloss as a function of the number of DBSs for various environments.

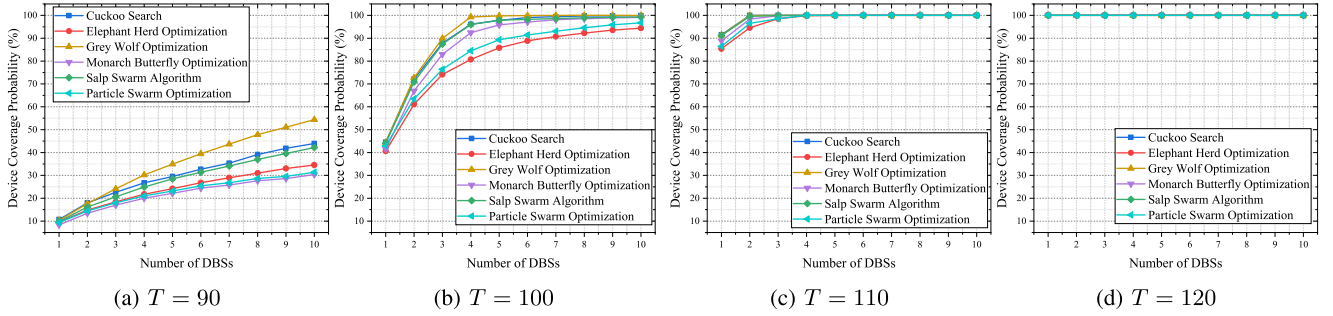


FIGURE 7. Coverage probability as a function of the number of DBSs for various thresholds in the urban environment.

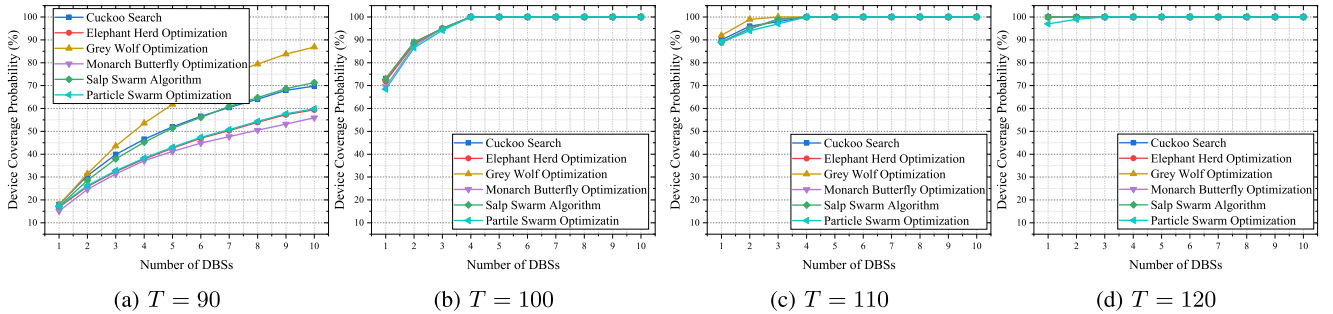


FIGURE 8. Coverage probability as a function of the number of DBSs for various thresholds in the suburban environment.

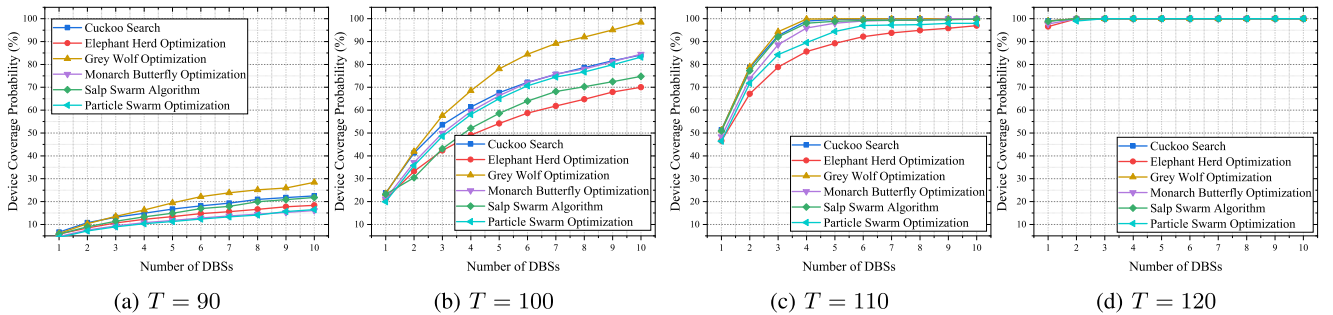


FIGURE 9. Coverage probability as a function of the number of DBSs for various thresholds in the dense urban environment.

90 dB, more DBSs are required to maintain a high coverage probability.

With respect to the propagation environment, we can notice that, in the suburban environment, the least number of DBSs is required to maintain a high coverage probability. On the other hand, in the high-rise urban environment, even with a threshold value

of 120 dB and 10 DBSs, the coverage probability is about 75%.

Finally, all swarm intelligence approaches feature a similar performance for higher numbers of DBSs and threshold values. Nevertheless, for lower threshold values, GWO and MBO have best and worst overall performance, respectively, while the rest approaches feature a relatively similar performance.

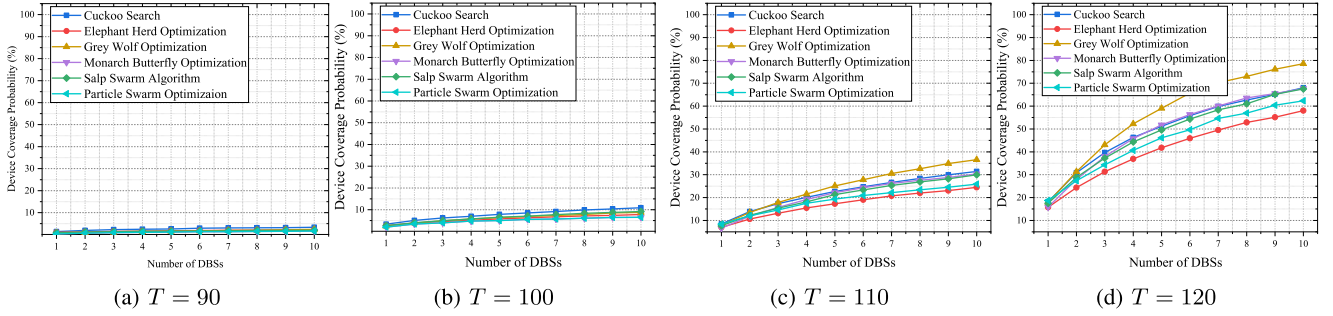


FIGURE 10. Coverage probability as a function of the number of DBSs for various thresholds in the high-rise urban environment.

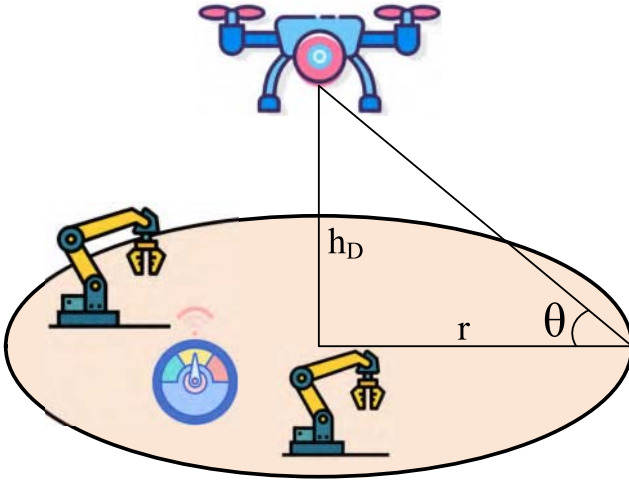


FIGURE 11. Coverage area of a DBS.

V. PATHLOSS ANALYSIS AND RESULTS DISCUSSION

After the DBSs have been deployed in the optimal location, we present an analysis of the pathloss distribution for a single DBS providing coverage to the devices. Fig. 11 depicts the coverage area of a DBS, where h_d denotes the hovering altitude, r the radius of the coverage area, and θ denotes the elevation angle. Since the devices are uniformly distributed over the whole area, their deployment inside the coverage will also follow a uniform distribution.

The derivative of (2) with respect to the elevation angle is obtained by [26]

$$\frac{\partial \text{PL}}{\partial \theta} = \frac{\pi}{9 \ln(10)} \tan(\theta) + \frac{abe^{-b(\theta-a)}}{[ae^{-b(\theta-a)} + 1]^2} \quad (31)$$

By setting the equation above equal to zero and solving for θ , the optimal elevation angles are found for the various environments, as follows:

$$\theta_{opt} = \begin{cases} 42.44^\circ & \text{for urban environment} \\ 20.34^\circ & \text{for suburban environment} \\ 54.62^\circ & \text{for dense urban environment} \\ 75.52^\circ & \text{for high-rise environment} \end{cases} \quad (32)$$

As stated in the previous subsection, a device is considered to be covered if the experienced pathloss is lower than a pre-defined threshold. Therefore, the coverage probability can be

expressed as

$$\mathbb{P}_{cov} = \Pr[\text{PL} \leq T] = \Pr\left[r \leq \cos(\theta_{opt}) 10^{\frac{Q}{10}}\right] \quad (33)$$

where

$$Q = \frac{1}{20} \left[-B + A \left(-1 + \frac{a}{a + e^{b(\theta-a)}} + \text{PL} \right) \right] \quad (34)$$

The probability in the rightmost side of (33) is also the cumulative distribution function of the radius. By defining R to be the maximum radius of the coverage area (i.e., the radius where all devices experience pathloss equal to PL), the probability density function (PDF) of r is obtained by

$$f_r(r) = \frac{2\pi r}{\pi R^2} \quad (35)$$

By integrating the PDF, the cumulative density function (CDF) is obtained by

$$F_r(r) = \left(\frac{r}{R}\right)^2 \quad (36)$$

Therefore, by combining (33) and (36), the CDF of the pathloss can be expressed as

$$F_{\text{PL}}(r) = \left[\frac{\cos(\theta_{opt}) 10^{\frac{Q}{10}}}{R|_{\text{PL}=T}} \right]^2 \quad (37)$$

The CDF of the pathloss is depicted in Fig. 12. As the maximum threshold is increased, the coverage radius also increases as more devices located further from the DBS can be covered. Also, based on the results, we can notice that the convergence rate is similar for all cases of T_{max} . Of note. The CDF is the same for all environments since the propagation parameters will also affect the maximum coverage radius, thus resulting in the same CDF.

In order to further evaluate the comparison results, we perform non-parametric statistical tests [55], [56]. Firstly, we apply the Friedman test using the data as presented in Fig. 2–Fig. 10. Through the simulations, 99 results for each approach were generated in total. Based on each result, the approaches were ranked from 1st to 6th and the average rank of each approach was determined. Table 6 presents the average ranking of the algorithms obtained by the Friedman test. According to the results, GWO is ranked as the highest performing approach in optimizing the DBS position in terms of maximizing the network performance. On the other hand, MBO features the lowest rank.

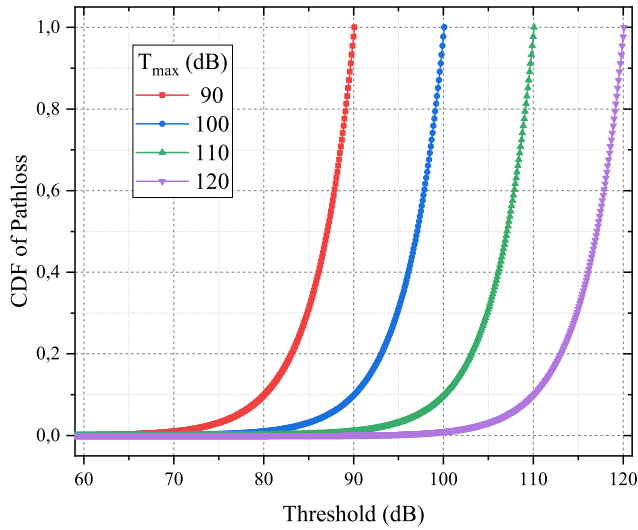

FIGURE 12. CDF of the pathloss for various T_{max} .

TABLE 6. Average ranking of the approaches.

Approach	Average Rank	Normalized Values	Rank
Cuckoo Search	2.545	2.227	2
Elephant Herd Optimization	4.796	4.196	4
Grey Wolf Optimization	1.143	1.000	1
Monarch Butterfly Optimization	6.844	6.005	6
Salp Swarm Algorithm	3.740	3.272	3
Particle Swarm Optimization	5.814	5.087	5

TABLE 7. Wilcoxon significance test.

Grey Wolf Optimization vs.	<i>P</i> -Value
Cuckoo Search	7.86E-28
Elephant Herd Optimization	7.41E-35
Monarch Butterfly Optimization	1.41E-36
Salp Swarm Algorithm	1.40E-36
Particle Swarm Optimization	1.41E-36

Furthermore, to test if GWO is significantly better than all the rest approaches, we perform the Wilcoxon signed-rank test with a significance level of 0.05. The respective results are summarized in Table 7. Based on the results, GWO is significantly better than the other approaches.

However, this conclusion cannot be generalized to all optimization problems in electromagnetics. An important theorem in evolutionary optimization is the ‘‘No Free Lunch’’ (NFL) theorem [57]. This theorem is about the average behavior of optimization algorithms over given spaces of optimization problems. NFL states that when averaged over all possible optimization problems defined over some search space X , no algorithm has a performance advantage over any other. Thus, there exists no optimization algorithm that is able to solve every problem effectively and efficiently. There are different optimizers that have the capabilities for obtaining a solution in different types of optimization problems. Therefore, it is worthwhile to try to find the best suited optimizer for a specific optimization problem.

In light of the aforementioned results, the following remarks can be established:

- The number of search agents does not have a significant impact on the performance of each algorithm, except for the MBO and PSO algorithms.
- After a particular number of generations (i.e., 100 according to Fig. 3), the performance of each algorithm is not affected. On the other hand, a lower number of generations leads to worse performance.
- The density, as well as the height of buildings, considerably affects final results. Nevertheless, the differences among the performance of each of the algorithms remain the same in all cases.
- The number of devices has a limited impact on the average pathloss results. Apart from the case of 10 devices, where CS, EHO, and SSA perform considerably better, the rest differences in each algorithm’s performance are similar.
- The number of DBSs has a considerable impact on the average pathloss. When the number of DBSs is high, they can be placed closer to the user, thus, reducing the average pathloss.
- The coverage probability is heavily dependent on the threshold value since the users that are placed farther from the DBSs cannot be covered when the threshold value is low. Of note, in the high-rise urban environment, higher threshold values and number of DBSs are required in order to maintain a high coverage probability.
- According to the Friedman ranking test, GWO is the best performing approach in finding the optimal DBS position, while MBO is the worst performing one. Also, as verified by the Wilcoxon significance test, GWO has a significantly better performance compared to the rest approaches.
- Finally, the visual representations of the results (i.e., Fig. 2–Fig. 10) are verified by the statistical results of the Friedman and Wilcoxon tests.

VI. CONCLUSION

Motivated by the emergence of NG-*IoT* and the associated challenges in terms of massive connectivity, we investigated the efficiency of swarm intelligence approaches on optimizing the DBS deployment. In this direction, we designed three evaluation scenarios related to the overall network performance and evaluated the performance of five swarm intelligence approaches, namely the CS, EHO, GWO, MBO, SSA, and PSO. In particular, these approaches were evaluated and compared in terms of finding the optimal DBSs positions using various parameters, such as the number of search agents, number of maximum generations, propagation environment, number of devices, and area size. Additionally, we performed non-parametric statistical tests, namely the Friedman and Wilcoxon tests, in order to further evaluate the comparison results. The comparison results showed that GWO and MBO rank highest and lowest in the Friedman

test, respectively. Furthermore, according to the Wilcoxon test, GWO performs significantly better than the rest of the approaches. Therefore, we deduce that GWO features the best performance in finding the DBS position in the 3D space so that the network performance is optimized.

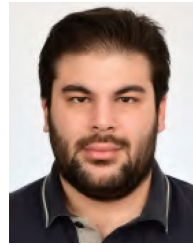
As a future extension of this work, we aim to modify the swarm intelligence approaches in order to consider multiple DBSs. By grouping the devices and assigning different DBSs, the overall performance of the network can be further increased. To further improve the network performance, non-orthogonal multiple access (NOMA) schemes can be utilized to schedule radio and power resources to the devices. In this direction, we aim to integrate this work with our previous ones in [58] and [59].

REFERENCES

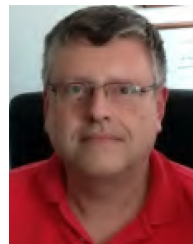
- [1] V. Kelli, P. Sarigiannidis, V. Argyriou, T. Lagkas, and V. Vitsas, "A cyber resilience framework for NG-IoT healthcare using machine learning and blockchain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.
- [2] I. Siniosoglou, P. Sarigiannidis, V. Argyriou, T. Lagkas, S. K. Goudos, and M. Poveda, "Federated intrusion detection in NG-IoT healthcare systems: An adversarial approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.
- [3] C. W. Chen, "Internet of video things: Next-generation IoT with visual sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, Aug. 2020.
- [4] M. J. Beliatas, K. Jensen, L. Ellegaard, A. Aagaard, and M. Presser, "Next generation industrial IoT digitalization for traceability in metal manufacturing industry: A case study of industry 4.0," *Electronics*, vol. 10, no. 5, p. 628, Mar. 2021.
- [5] R. Bajracharya, R. Shrestha, R. Ali, A. Musaddiq, and S. W. Kim, "LWA in 5G: State-of-the-art architecture, opportunities, and research challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 134–141, Oct. 2018.
- [6] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things," *IEEE Netw.*, vol. 33, no. 5, pp. 12–19, Sep./Oct. 2019.
- [7] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G With UAVs: Foundations of a 3D wireless cellular network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 357–372, Jan. 2019.
- [8] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5G and resilient public safety networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 36–42, Jan. 2018.
- [9] N. Sharma, M. Magarini, D. N. K. Jayakody, V. Sharma, and J. Li, "On-demand ultra-dense cloud drone networks: Opportunities, challenges and benefits," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 85–91, Aug. 2018.
- [10] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovács, J. Wigard, and T. B. Sorensen, "Radio channel modeling for UAV communication over cellular networks," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 514–517, Aug. 2017.
- [11] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 38–41, Feb. 2018.
- [12] H. Shakhateh, A. Khreishah, A. Alsarhan, I. Khalil, A. Sawalmeh, and N. S. Othman, "Efficient 3D placement of a UAV using particle swarm optimization," in *Proc. 8th Int. Conf. Inf. Commun. Syst. (ICICS)*, Irbid, Jordan, Apr. 2017, pp. 258–263.
- [13] L. Zhang, Q. Fan, and N. Ansari, "3-D drone-base-station placement with in-band full-duplex communications," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1902–1905, Sep. 2018.
- [14] L. Wang, B. Hu, and S. Chen, "Energy efficient placement of a drone base station for minimum required transmit power," *IEEE Wireless Commun. Lett.*, vol. 9, no. 12, pp. 2010–2014, Dec. 2020.
- [15] L. Zhang and N. Ansari, "Approximate algorithms for 3-D placement of IBFD enabled drone-mounted base stations," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7715–7722, Aug. 2019.
- [16] N. Babu, C. B. Papadias, and P. Popovski, "Energy-efficient 3-D deployment of aerial access points in a UAV communication system," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2883–2887, Dec. 2020.
- [17] D. Pliatsios, P. Sarigiannidis, S. K. Goudos, and K. Psannis, "3D placement of drone-mounted remote radio head for minimum transmission power under connectivity constraints," *IEEE Access*, vol. 8, pp. 200338–200350, 2020.
- [18] S. K. Goudos, K. Siakavara, and J. N. Sahalos, "Novel spiral antenna design using artificial bee colony optimization for UHF RFID applications," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 528–531, 2014.
- [19] A. A. Al-Azza, A. A. Al-Jodah, and F. J. Harackiewicz, "Spider monkey optimization: A novel technique for antenna optimization," *IEEE Antennas Wireless Propag. Lett.*, vol. 15, pp. 1016–1019, 2016.
- [20] S. K. Goudos and G. Athanasiadou, "Application of an ensemble method to UAV power modeling for cellular communications," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, pp. 2340–2344, 2019.
- [21] A. D. Boursianis *et al.*, "Multiband patch antenna design using nature-inspired optimization method," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 151–162, 2021.
- [22] A. Niccolai, M. Beccaria, R. E. Zich, A. Massaccesi, and P. Pirinoli, "Social network optimization based procedure for beam-scanning reflectarray antenna design," *IEEE Open J. Antennas Propag.*, vol. 1, pp. 500–512, 2020.
- [23] W. Luo, H. Jin, H. Li, and K. Duan, "Radar main-lobe jamming suppression based on adaptive opposite fireworks algorithm," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 138–150, 2021.
- [24] M.-H. Golbon-Haghighi, M. Mirmozafari, H. Saeidi-Manesh, and G. Zhang, "Design of a cylindrical crossed dipole phased array antenna for weather surveillance radars," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 402–411, 2021.
- [25] J. Chen *et al.*, "Absorption and diffusion enabled ultrathin broadband metamaterial absorber designed by deep neural network and PSO," *IEEE Antennas Wireless Propag. Lett.*, vol. 20, pp. 1993–1997, 2021.
- [26] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
- [27] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications—Swarm intelligence algorithms," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018.
- [28] M. Elbes, S. Alzubi, T. Kanan, A. Al-Fuqaha, and B. Hawashin, "A survey on particle swarm optimization with emphasis on engineering and network applications," *Evol. Intell.*, vol. 12, no. 2, pp. 113–129, Jun. 2019.
- [29] S. K. Goudos, *Emerging Evolutionary Algorithms for Antennas and Wireless Communications* (Electromagnetic Waves). Stevenage, U.K.: SciTech Publ. Incorp., 2021.
- [30] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nat. Biol. Inspired Comput. (NaBIC)*, Coimbatore, India, 2009, pp. 210–214.
- [31] M. Mareli and B. Twala, "An adaptive cuckoo search algorithm for optimisation," *Appl. Comput. Informat.*, vol. 14, no. 2, pp. 107–115, Jul. 2018.
- [32] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search optimization—A review," *Materialstoday Proc.*, vol. 4, no. 8, pp. 7262–7269, 2017.
- [33] S. Roy and S. S. Chaudhuri, "Cuckoo search algorithm using Lévy Flight: A review," *Int. J. Mod. Educ. Comput. Sci.*, vol. 5, no. 12, pp. 10–15, Dec. 2013.
- [34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [35] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. D. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Syst. Appl.*, vol. 47, pp. 106–119, Apr. 2016.
- [36] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm Evol. Comput.*, vol. 44, pp. 101–112, Feb. 2019.
- [37] S. K. Goudos, A. Boursianis, M. Salucci, and P. Rocca, "Dualband patch antenna design using binary grey wolf optimizer," in *Proc. IEEE Int. Symp. Antennas Propag. North Amer. Radio Sci. Meeting*, Montreal, QC, Canada, Jul. 2020, pp. 1777–1778.

- [38] S. Chen, R. Chen, and J. Gao, "A monarch butterfly optimization for the dynamic vehicle routing problem," *Algorithms*, vol. 10, no. 3, p. 107, Sep. 2017.
- [39] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, Jul. 2019.
- [40] S. K. Goudos, "Cognitive radio engine design for IoT using monarch butterfly optimization and fuzzy decision making," in *Internet of Things*. Cham, Switzerland: Springer Int., 2020, pp. 81–100.
- [41] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: A comprehensive review," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114418.
- [42] G.-G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proc. 3rd Int. Symp. Comput. Bus. Intell. (ISCBI)*, Bali, Indonesia, Dec. 2015, pp. 1–5.
- [43] I. Strumberger, M. Beko, M. Tuba, M. Minovic, and N. Bacanin, "Elephant herding optimization algorithm for wireless sensor network localization problem," in *IFIP Advances in Information and Communication Technology*. Cham, Switzerland: Springer Int., 2018, pp. 175–184.
- [44] W. Li, G.-G. Wang, and A. H. Alavi, "Learning-based elephant herding optimization algorithm for solving numerical optimization problems," *Knowl. Based Syst.*, vol. 195, May 2020, Art. no. 105675.
- [45] J. Li, H. Lei, A. H. Alavi, and G.-G. Wang, "Elephant herding optimization: Variants, hybrids, and applications," *Mathematics*, vol. 8, no. 9, p. 1415, Aug. 2020.
- [46] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [47] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, Nov. 2019.
- [48] H. M. Kanoosh, E. H. Houssein, and M. M. Selim, "Salp swarm algorithm for node localization in wireless sensor networks," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–12, Feb. 2019.
- [49] R. Salgotra, U. Singh, S. Singh, G. Singh, and N. Mittal, "Self-adaptive salp swarm algorithm for engineering optimization problems," *Appl. Math. Model.*, vol. 89, pp. 188–207, Jan. 2021.
- [50] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, 1995, pp. 33–57.
- [51] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyper-parameter selection in deep neural networks," in *Proc. Genet. Evol. Comput. Conf.*, Jul. 2017, pp. 481–488.
- [52] M. H. Ali, B. A. D. A. Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.
- [53] C. E. Garcia, M. R. Camana, I. Koo, and M. A. Rahman, "Particle swarm optimization-based power allocation scheme for secrecy sum rate maximization in NOMA with cooperative relaying," in *Intelligent Computing Theories and Application*. Cham, Switzerland: Springer Int., 2019, pp. 1–12.
- [54] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *J. Comput. Phys.*, vol. 226, no. 2, pp. 1830–1844, Oct. 2007.
- [55] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, Dec. 2009.
- [56] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.
- [57] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [58] D. Pliatsios and P. Sarigiannidis, "Power allocation in downlink non-orthogonal multiple access IoT-enabled systems: A particle swarm optimization approach," in *Proc. 15th Int. Conf. Distrib. Comput. Sens. Syst. (DCOSS)*, Santorini Island, Greece, May 2019, pp. 416–422. [Online]. Available: <https://ieeexplore.ieee.org/document/8804731/>

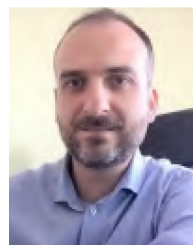
- [59] D. Pliatsios and P. Sarigiannidis, "Resource allocation combining heuristic matching and particle swarm optimization approaches: The case of downlink non-orthogonal multiple access," *Information*, vol. 10, no. 11, p. 336, Oct. 2019.



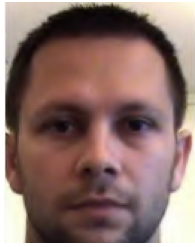
DIMITRIOS PLIATSIOS (Member, IEEE) received the Diploma degree from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece. His Ph.D. research is funded by the Greek State Scholarship Foundation. His research interests lie in the areas of wireless communications, mobile networks, and computer & network security. He has participated as a Research Associate with the University of Western Macedonia in European and National research projects. He is a member of the Technical Chamber of Greece.



SOTIRIOS K. GOUDOS (Senior Member, IEEE) received the B.Sc. degree in physics, the M.Sc. degree in electronics, and the Ph.D. degree in physics from the Aristotle University of Thessaloniki in 1991, 1994, and 2001, respectively, the master's degree in information systems from the University of Macedonia, Greece, in 2005, and the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki in 2011. He joined the Department of Physics, Aristotle University of Thessaloniki in 2013, where he is currently an Associate Professor. He is the Director of the ELEDIA@AUTH Lab Member with the ELEDIA Research Center Network. He has authored the book *Emerging Evolutionary Algorithms for Antennas and Wireless Communications* (Institution of Engineering & Technology, 2021). His research interests include antenna and microwave structures design, evolutionary algorithms, machine learning, wireless communications, and semantic Web technologies. He is currently serving as a Chapter/AG Coordinator for IEEE Greece Section. He is the founding Editor-in-Chief of the *Telecom* (MDPI). He is currently serving as an Associate Editor for *IEEE ACCESS* and *IEEE OPEN JOURNAL OF THE COMMUNICATION SOCIETY*. He is also a Member of the Editorial Board of the *International Journal of Antennas and Propagation*, the *EURASIP Journal on Wireless Communications and Networking*, and the *International Journal on Advances on Intelligent Systems*. He is also a member of the topic board of the Electronics, IEICE, the Greek Physics Society, the Technical Chamber of Greece, and the Greek Computer Society.



THOMAS LAGKAS (Senior Member, IEEE) received the Ph.D. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2006. He has been a Lecturer and then a Senior Lecturer with the University of Sheffield International Faculty—CITY College, from 2012 to 2019. He also served as the Research Director of the Computer Science Department, CITY College and the Leader of the ICT Track, South-East European Research Centre. His research interests are in the areas of IoT communications and distributed architectures, communication security, wireless communication networks, QoS in medium access control, mobile multimedia communications, power saving/fairness ensure for resource allocation in wireless sensor-cooperative broadband networks as well as in hybrid fiber-wireless networks, e-health data monitoring, 5G and beyond systems, flying ad hoc networks, and computer-based educational technologies with more than 90 publications at a number of widely recognized international scientific journals and conferences. He also participates in the Editorial Boards of the *IEEE INTERNET OF THINGS JOURNAL*, *Computer Networks* (Elsevier), the *Telecommunication Systems* (Springer), and the *EURASIP Journal on Wireless Communications and Networking* (Springer). He actively participates in drafting research funding proposals, as well as in the implementation of the corresponding EU projects (namely, ELECTRON, TERMINET, SDN-microSENSE, SmartROOT, JAUNTY, FuseGI, and ARRANGE-ICT).



VASILEIOS ARGYRIOU (Member, IEEE) received the B.Sc. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2001, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Surrey in 2003 and 2006, respectively, working on registration. From 2001 to 2002, he held a research position with the AIIA Lab, Aristotle University, working on image and video watermarking. He joined the Communications and Signal Processing Department, Imperial College, London, in 2007,

where he was a Research Fellow working on 3D object reconstruction. He is currently a Professor with Kingston University, working on computer vision and AI for crowd and human behavior analysis, computer games, entertainment, and medical applications. His research is also conducted on educational games and on HCI for augmented and virtual reality (AR/VR) systems.

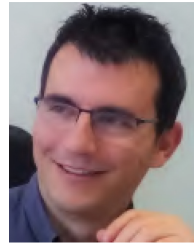


ALEXANDROS-APOSTOLOS

A. BOULOGEOGOS (Senior Member, IEEE) was born in Trikala, Greece, in 1988. He received the Diploma degree in electrical and computer engineering and the Ph.D. degree in wireless communications from the Aristotle University of Thessaloniki (AUTH) in 2012 and 2016, respectively.

In 2012, he joined the Wireless Communications Division, AUTH, as a Research Assistant/Project Engineer in various telecommunications and networks projects. In 2017, he joined the Information Technologies Institute. In November 2017, he also joined the Department of Digital Systems, University of Piraeus, where he conducts research in the area of wireless communications. From October 2012 to September 2016, he was a Teaching Assistant with the Department of ECE, AUTH, and since February 2017, he has been serving as an Adjunct Professor with the Department of ECE, University of Western Macedonia, and as an Visiting Lecturer with the Department of Computer Science and Biomedical Informatics, University of Thessaly. He has authored and coauthored more than 65 technical papers, which were published in scientific journals and presented at prestigious international conferences. He has submitted two (one national and one European) patents. His current research interests spans in the area of wireless communications and networks with emphasis in high frequency communications, optical wireless communications, and communications for biomedical applications.

Dr. Boulougorgos was awarded the Distinction Scholarship Award from the Research Committee of AUTH in 2014, and was recognized as an Exemplary Reviewer for IEEE COMMUNICATION LETTERS in 2016 (top 3% of reviewers). He served as a reviewer in various IEEE journals and conferences. He is currently an Editor for IEEE COMMUNICATIONS LETTERS, and an Associate Editor for the *Frontier in Communications and Networks*. He was named a Top Peer Reviewer (top 1% of reviewers) in Cross-Field and Computer Science in the Global Peer Review Awards 2019, which was presented by the Web of Science and Publons. He has been involved as a member of technical program committees in several IEEE and non-IEEE conferences. He is a member of the Technical Chamber of Greece.



PANAGIOTIS SARIGIANNIDIS (Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. He has published over 200 papers in international journals, conferences, and book chapters, including IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE INTERNET OF THINGS, IEEE TRANSACTIONS ON BROADCASTING, IEEE

SYSTEMS JOURNAL, *IEEE Wireless Communications Magazine*, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY, IEEE ACCESS, and *Computer Networks*. He has been involved in several national, European, and international projects. He is currently the Project Coordinator of three H2020 projects, namely, a) H2020-DS-SC7-2017 (DS-07-2017), SPEAR: Secure and PrivatE smArt gRid; b) H2020-LC-SC3-EE-2020-1 (LC-SC3-EC-4-2020), EVIDENT: bEhaVioral Insigths and Effective eNergy policy aCTions; and c) H2020-ICT-2020-1 (ICT-56-2020), TERMINET: nexT gEneRation sMART INterconnectEd IoT, while he coordinates the Operational Program MARS: sMART fARming with dRoneS (Competitiveness, Entrepreneurship, and Innovation) and the Erasmus+ KA2 ARRANGE-ICT: SmartROOT: Smart faRming innOvatiOn Training. He also serves as a Principal Investigator in the H2020-SU-DS-2018 (SU-DS04-2018), SDN-microSENSE: SDN-microgrid reSilient Electrical eNergy SystEm and in three Erasmus+ KA2: a) ARRANGE-ICT: pArtnErship foR AddressiNG mEgatrends in ICT; b) JAUNTY: Joint undergAduate coURses for smart eNergy managemenT sYstems; and c) STRONG: advanced firST ResPONders traininG (Cooperation for Innovation and the Exchange of Good Practices). His research interests include telecommunication networks, Internet of Things, and network security. He participates in the Editorial Boards of various journals, including *International Journal of Communication Systems* and *EURASIP Journal on Wireless Communications and Networking*.