



# Efficient fog node placement using nature-inspired metaheuristic for IoT applications

Abdenacer Naouri<sup>1</sup> · Nabil Abdelkader Nouri<sup>2</sup> · Amar Khelloufi<sup>1</sup> · Abdelkarim Ben Sada<sup>1</sup> · Huansheng Ning<sup>1</sup> · Sahraoui Dhelim<sup>3</sup>

Received: 14 October 2023 / Revised: 28 February 2024 / Accepted: 29 February 2024  
© The Author(s) 2024

## Abstract

Managing the explosion of data from the edge to the cloud requires intelligent supervision, such as fog node deployments, which is an essential task to assess network operability. To ensure network operability, the deployment process must be carried out effectively regarding two main factors: connectivity and coverage. The network connectivity is based on fog node deployment, which determines the network's physical topology, while the coverage determines the network accessibility. Both have a significant impact on network performance and guarantee the network quality of service. Determining an optimum fog node deployment method that minimizes cost, reduces computation and communication overhead, and provides a high degree of network connection coverage is extremely hard. Therefore, maximizing coverage and preserving network connectivity is a non-trivial problem. In this paper, we propose a fog deployment algorithm that can effectively connect the fog nodes and cover all edge devices. Firstly, we formulate fog deployment as an instance of multi-objective optimization problems with a large search space. Then, we leverage Marine Predator Algorithm (MPA) to tackle the deployment problem and prove that MPA is well-suited for fog node deployment due to its rapid convergence and low computational complexity, compared to other population-based algorithms. Finally, we evaluate the proposed algorithm on a different benchmark of generated instances with various fog scenario configurations. Our algorithm outperforms state-of-the-art methods, providing promising results for optimal fog node deployment. It demonstrates a 50% performance improvement compared to other algorithms, aligning with the No Free Lunch Theorem (NFL Theorem) Theorem's assertion that no algorithm has a universal advantage across all problem domains. This underscores the significance of selecting tailored algorithms based on specific problem characteristics.

**Keywords** Cloud · Intelligent supervision · Fog node deployments · Network operability · Connectivity · Coverage

## 1 Introduction

With the recent advancement of the Internet of Things (IoT), we have witnessed the emergence of many new applications in different domains, such as health care, surveillance, and smart cities. IoT systems can be realized using different computing and communication architectures, where each architecture can provide solutions for data processing, storing, and analyzing according to the available resources [1]. Fog computing and edge computing are considered integral parts of the IoT systems and are proposed to cover the cloud limitation issues arising from the enormous IoT sensor offloaded requests, such as latency response time, which may take a long time to

---

✉ Sahraoui Dhelim  
sahraoui.dhelim@ucd.ie

<sup>1</sup> School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

<sup>2</sup> Department of Mathematics and Computer Science, Ziane Achour University of Djelfa, Djelfa, Algeria

<sup>3</sup> School of Computer Science, University College Dublin, Dublin, Ireland

answer users' requests [2, 3]. Both fog computing and edge computing are regarded as an extension of cloud computing that can provide distributed computing, storage, and networking functions close to the IoT sensors at the edge network to serve the IoT applications needs, which supports low latency, real-time response, location awareness, devices mobility, scalability, heterogeneity, temporary storage, high-speed data dissemination, decentralized computation, and security [4]. The enormous resulting requests from IoT sensors and edge devices could lead to a data explosion through the network, thereby an optimal network deployment is required. In most Ad-hoc networks, the deployment of computing nodes is accomplished either in a pre-planned or ad-hoc method [5]. The pre-planned deployment method is used when the deployment field access is limited, and the deployment cost is not expensive. While the ad-hoc manner is used when the deployment field is large, costly, and consists of many computing nodes [6].

The deployment strategy is constructed on the basis of several considerations, such as the IoT sensor and edge device location, system functionality, and the environments. Also, it may differ from one area to another according to its nature, whereas in an open area, the deployment tries to cover a wider area than the closed area, such as inside a building. There are plenty of deployment possibilities for user coverage and connectivity needs, for example, in critical situations such as natural disasters, oil rigs, mines, battlefield surveillance, high-speed mobile video gaming, or in public transport [7–9]. Finding an optimal deployment for a wide area is a challenge that could suffer from user density and the unregulated deployment surface compared to small areas. Because the entire system's performance may be impacted by the deployment of computing nodes which is considered a crucial concern [10–12]. Therefore, in this work, we concentrate on providing an optimal fog node placement within the edge network to serve the boundary edge devices and IoT sensors in a specific area.

In order to deploy the fog computing nodes effectively, we need a complete understanding of the relation between the network topology and node density, subjected to different factors such as location and density [13]. Hence, some issues should be addressed to enhance the network performance, including connectivity, coverage, reliability, accessibility, etc. If the fog nodes are installed without considering restricting factors of the actual region of interest and the underlying topology, it can cause low network coverage and connectivity. The node deployment problem has been investigated with different networks and in different environments. It remains a challenging problem that has been proven to be NP-Hard [14]. In order to address these issues, meta-heuristic techniques have been

developed, although they typically only provide optimal local solutions [15].

Compared to other approaches, meta-heuristic methods have experienced tremendous success and development in addressing various real-world optimization issues. The computing scenario considered for this work consists of IoT sensors and edge devices located at the edge of the network and fog nodes located within the fog layer. As the problem showed its NP-Hard, we propose an MPO population-based meta-heuristic algorithm to efficiently solve the fog node deployments within the edge network devices and evaluate the network coverage and connectivity effect on the network's performance. Our contributions to this study are summarized as follows:

- To efficiently connect the fog nodes and encompass all edge nodes, we propose a multi-objective aggregate objective function that simultaneously maximizes the coverage of edge devices and enhances network connectivity. This approach ensures seamless connectivity and coverage for all devices and fog nodes, effectively improving network performance and guaranteeing network QoS.
- We present an MPA metaheuristic nature-inspired optimization algorithm, where the objective function aims to maximize network coverage and connectivity. We analyze the impact of different parameters on the network's performance and identify the most effective optimization methods.
- For seamless computation, the proposed algorithm has been implemented and tested in various network settings. The results demonstrate a significant improvement in coverage and connectivity compared to other methods.

The rest of the paper is organized as follows: Sect. 2 reviews existing research on fog node deployment. Section 3 describes the system modeling and problem formulation. Section 4 details the proposed meta-heuristic algorithm for fog node deployment. Section 5 presents the evaluation details and experiment results. We conclude the paper and outline future research directions in Sect. 6.

## 2 Related work

Today's networks require precise computing node placement to ensure reliable measurements and efficient data transmission. Node deployment procedures are vital in this regard. For instance, authors in [16] introduce a fog computing framework and an optimization model for IoT service placement over fog resources, employing a genetic algorithm for improved service execution and fog resource utilization. Similarly, in [17, 18] authors tackle wireless

mesh network deployment challenges by proposing novel objective functions maximizing network connectivity and coverage using meta-heuristic algorithms like Moth-Flame Optimization (MFO) and Accelerated Particle Swarm Optimizer (APSO), respectively, yielding promising outcomes. Furthermore, authors in [19] propose fog node placement techniques utilizing a multi-objective genetic algorithm (MOGA) to minimize deployment cost and network latency, contributing to efficient fog network design. Despite several researchers have been conducted to address this issue, the existing solutions must be improved upon or replaced with new ones regularly due to their limitations. Consequently, it is possible to employ optimization algorithms to find the optimum node locations that meet the specified criteria.

Researchers from both academia and industry have extensively employed heuristic-based optimization algorithms to tackle deployment optimization challenges. In this research, we focus on addressing the Fog Node Deployment Problem (FND) by leveraging heuristic and meta-heuristic methods [20, 21]. These methods have garnered significant attention due to their effectiveness in optimizing various deployment parameters. Specifically, they aim to minimize energy consumption, reduce latency for time-sensitive applications, maximize network bandwidth and throughput, and ensure comprehensive coverage and connectivity of fog computing devices [22–25]. This focus on heuristic and meta-heuristic methods underscores their importance in achieving optimal fog node deployment solutions [25–28]. Authors in [29] initially addressed the coverage problem in WSN and demonstrated that it is NP-hard. They concentrated on the issue of improving the computing node's coverage for a given number of sensors. In [30], the authors suggested an improved Artificial Bee Colony method for optimizing the lifetime of a two-tiered wireless sensor network through optimal relay node deployment. They integrated the problem dimension into the candidate discovery formula and modified the local search strategy based on problem fitness and the number of iterations. These modifications assist in achieving a balance between algorithm exploration and exploitation, enhancing the algorithm's capacity for solving the problem effectively. In [31], the authors proposed a particle swarm optimization (PSO) algorithm to optimize the node network deployment and improve the adaptive ability of the network; however, the algorithm has the disadvantage of falling into the local optimum. In [32], the authors assumed a hexagonal topology for the network and proposed an enhanced virtual spring force algorithm (VSFA) for node deployment within this topology. This approach aims to reduce the vulnerability area. However, it should be noted that the strategy has only been evaluated under perfect circumstances and has not been tested in complex settings.

As in [32], authors in [33] performed their evaluation with an ideal deployment environment presuming that nodes are homogeneous, where an enhanced fish swarm algorithm (AIFS) was adopted to optimize the node's deployment by targeting the coverage rate, which greatly increases the network coverage area and reduces the energy usage. The authors in [34, 35] focused on improving energy utilization, energy balance, and network coverage in wireless sensor networks. In [34], they employed a linear weighted combination to merge the three optimization objectives into a single objective and utilized the whale group algorithm (WGA) to optimize this objective function. However, this approach is computationally time-consuming. On the other hand, in [35], the authors considered the deployment of homogeneous nodes and the presence of obstacles. They employed two versions of a multi-objective evolutionary algorithm, one based on decomposition and the other to jointly optimize the model.

In [36], the authors address fault tolerance and connectivity challenges in wireless sensor networks (WSNs). They specifically focus on connection restoration and propose a simple and efficient algorithm for motion-based  $k$ -connectivity reconstruction. The algorithm categorizes nodes into critical and non-critical clusters based on their failure to achieve a lower  $k$  value. When a critical node fails, the algorithm selects and transfers non-critical nodes to compensate for the loss. In [37] authors presented methods for estimating connectivity in IoT-enabled wireless sensor networks. They introduce a simple and efficient method (PINC) for movement-based  $k$ -connectivity restoration that splits nodes into critical and non-critical groups. When a crucial node fails, the PINC algorithm transfers the non-critical nodes. This approach transfers a non-critical node with the fewest movement costs to the position of the failed node.

Unlike previous works, specific authors concentrated on placing nodes in a discrete grid area, which limits the positions of nodes. Counterpart, our approach gives us high flexibility to place fog nodes within a continuous region of interest. In addition, previous approaches optimized network connectivity and user coverage in two hierarchical ways. In our work, we considered a multi-objective aggregate function to optimize both simultaneously because user coverage may be crucial in some valuable services. Some experimental results will be provided to show the advantage of simultaneous optimization. Our approach allows us to place fog nodes within a continuous region of interest.

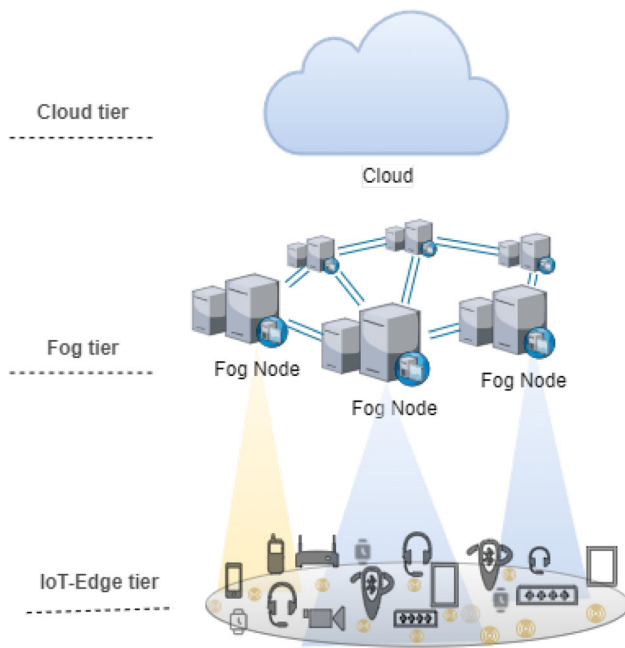


Fig. 1 IoT-edge based fog infrastructure

### 3 System assumptions and problem model formulation

This section introduces a fog deployment model within the IoT-fog-cloud infrastructure. The system under study comprises IoT sensors, devices, and fog nodes, as depicted in Fig. 1. To meet the demands of terminal IoT edge nodes for high throughput and quality of service, while maintaining cost-efficiency, we propose an IoT-edge-based fog infrastructure illustrated in Fig. 1. This model encompasses key concepts such as system architecture, coverage, network connectivity, and the challenges associated with multi-objective fog node deployment.

#### 3.1 Network model description

Figure 1 illustrates three-tier IoT-edge-based Fog infrastructure. The first tier is the IoT-edge tier, encompassing many IoT-edge terminal nodes. These nodes establish connections with the second tier, known as the fog tier, which consists of multiple fog server nodes. The third layer is the cloud layer, where computing nodes can perform intensive data analytics tasks. Let  $U$  represent the set of computing nodes within the infrastructure, given by  $U = FG \cup EN$ . The fog layer consists of  $n$  fog nodes, denoted as  $FG = FG_1, FG_2, FG_3, \dots, FG_n$ , with their corresponding communication ranges represented by  $R(FG_i) = R_1, R_2, R_3, \dots, R_n$ . Similarly, the IoT-edge layer comprises  $m$  edge nodes denoted by

Table 1 mathematical notation used in model development

Symbol	Meaning
$n$	Number of fog nodes
$m$	Number of IoT-edge clients
$l$	Location vector of fog nodes
$l(x_i, y_i)$	Node location
$W$	Region of interest width
$H$	Region of interest height
$FG$	Fog node set
$EN$	IoT-edge nodes set
$EN_i$	The $i$ -th edge node client
$E$	Set of edge links between fog nodes
$G_i$	$i$ -th subgraph component
$ G_i $	Size of the $i$ -th subgraph component
$h$	Number of subgraph components
$\zeta(G)$	Network connectivity
$\Phi(G)$	Terminal IoT-edge node coverage
$\gamma_i$	Boolean coverage function
$X_i$	The $i$ -th prey
$fit_{BEST}$	Optimal fitness value of fog node deployment
$Max_{itr}$	Maximum step movement

$EN = E_1, E_2, E_3, \dots, E_m$ . To facilitate the problem definition, a list of notations and symbols used in the system model is provided in Table 1.

#### 3.2 System model assumptions

To respond to a real network deployment scenario in practice, we investigate a deployment scenario where the fog and IoT-edge nodes are considered static. Each fog node has a different length of communications radius, and fog nodes can communicate with each other via their radio coverage [38]. On the other hand, edge nodes only have the essential functions for network connectivity but do not have the function of gateways or bridges [39]. Hence, edge nodes must go through fog nodes to communicate with other nodes. Moreover, since the locations of fog nodes are determined based on edge nodes' locations, it is necessary to know the locations of edge nodes in advance [40]. Briefly, to ensure network connectivity and coverage, the following assumptions have been established:

- The fog nodes and IoT-edge nodes are deployed uniformly and randomly over the entire network and are considered static inside the region of interest.
- Each fog node within the region of interest has access to the cloud center through cellular networks. The placements of the terminal nodes are considered fixed.

- To address the heterogeneity of terminal nodes in practical settings, we assume that each fog node  $fg_i$  is associated with a distinct communication range  $R_{fg_i}$ .
- A fog node can only be connected to a limited number of edge nodes.

Euclidean distance is a fundamental metric used to evaluate network accessibility and connectivity. Therefore, in order to ensure the desired network performance, it is necessary to satisfy two conditions:

1. An IoT-edge node  $EN_i$  is considered connected, if and only if it is covered by at least a fog node  $fg_i$ , as shown in equation(1):

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R_i \tag{1}$$

2. Two fog nodes,  $fg_i$  and  $fg_j$ , are considered connected if and only if they are within each other’s communication range, as demonstrated in equation (2):

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \min(R_i, R_j) \tag{2}$$

Figure 2 demonstrates an instance of the introduced problem; in this example, nodes consist of two types: fog and IoT-edge nodes, which consist of 10 fog nodes  $n = 10$  and 80 IoT-edge devices  $m = 80$ , located within the deployment area, each fog node have a varied communication range. If two fog nodes are in the communication range of each other, they will be linked via a dark link (e.g., see the dark link between fog nodes 1 and 2). If an IoT-edge node is located within the communication range of a fog node, it will be connected by a thin black link to the nearest fog. Furthermore, the topology graph includes three sub-graph components, the largest of which is 5 (i.e.,  $\zeta(G) = 5$ ), and 68 IoT-edge devices are covered (i.e.,  $\Phi(G) = 68$ ). If we shift specific fog nodes toward the most congested part of the IoT-edge device, as illustrated in Figure 2a, practically all of the sub-graphs will be combined into a single giant graph with size 10 (i.e.,  $\Phi(G) = 10$ ), and 76 IoT-edge devices will be covered (i.e.,  $\zeta(G) = 76$ ). Therefore, edge coverage and network connectivity can be improved by changing the locations of some fog nodes.

### 3.3 Problem formulation

We focus on determining near-optimal fog node placement by placing fog nodes in appropriate positions within the region of interest while ensuring connectivity and maximizing coverage. The network coverage refers to the number of covered edge nodes, while the connectivity refers to connected fog nodes within the network. To address this issue, it is not practical to analyze the whole

network. It is hard or even impossible to provide a connected network that covers all edge nodes due to the network dis-connectivity, which can be treated separately instead. Therefore, in our study, we target the greatest sub-network, i.e. primarily connected sub-network. By assuming  $L$  is the initially given location set of fog nodes  $L = \{L(x_1, y_1), L(x_2, y_2), \dots, L(x_n, y_n)\}$ .

We aim to update the fog nodes’ locations so that the covered edge nodes and the size of the greatest sub-network of connectivity are maximized. We observe that these two targets are in dispute, which means that a large sub-component network does not necessarily indicate a wide coverage of edge nodes. We analyze our scenario problem with a network of  $n$  fog nodes and  $m$  IoT-edge devices deployed in a 2D area. We represent an undirected topology graph  $G = (V, E)$ , where  $V = FG \cup EN$ .  $V$  refers to network nodes which include a set of fog nodes denoted by  $FG$  and a set of edge nodes denoted by  $EN$ , while  $E$  defines the edge node’s connectivities. Two fog nodes are considered connected if an edge  $(fg_i, fg_j) \in E$  exists and  $R_{fg_i} \cap R_{fg_j} \neq \emptyset$ ; an edge node  $EN_j \in E$  are considered covered if and only if fog node  $fg_i \in F$ ,  $L_{EN_j} \in R_j$ , and edge-fog  $edge(en_j, fg_i) \in E$  exists.

As mentioned earlier, It is challenging to find a fully connected and covered graph based on analyzing the entire network due to network disconnectivity; therefore, targeting a large sub-network could be a solution to improve network connectivity. We pointed out that the corresponding graph  $G$  of the target network may not be connected, i.e.,  $G$  may consist of several sub-graph components. However, note that maximizing the network connectivity of fog nodes may not be able to cover all edge nodes. In this situation, we aim to render the size of the most significant sub-graph component as large as possible to maximize the network’s connectivity.

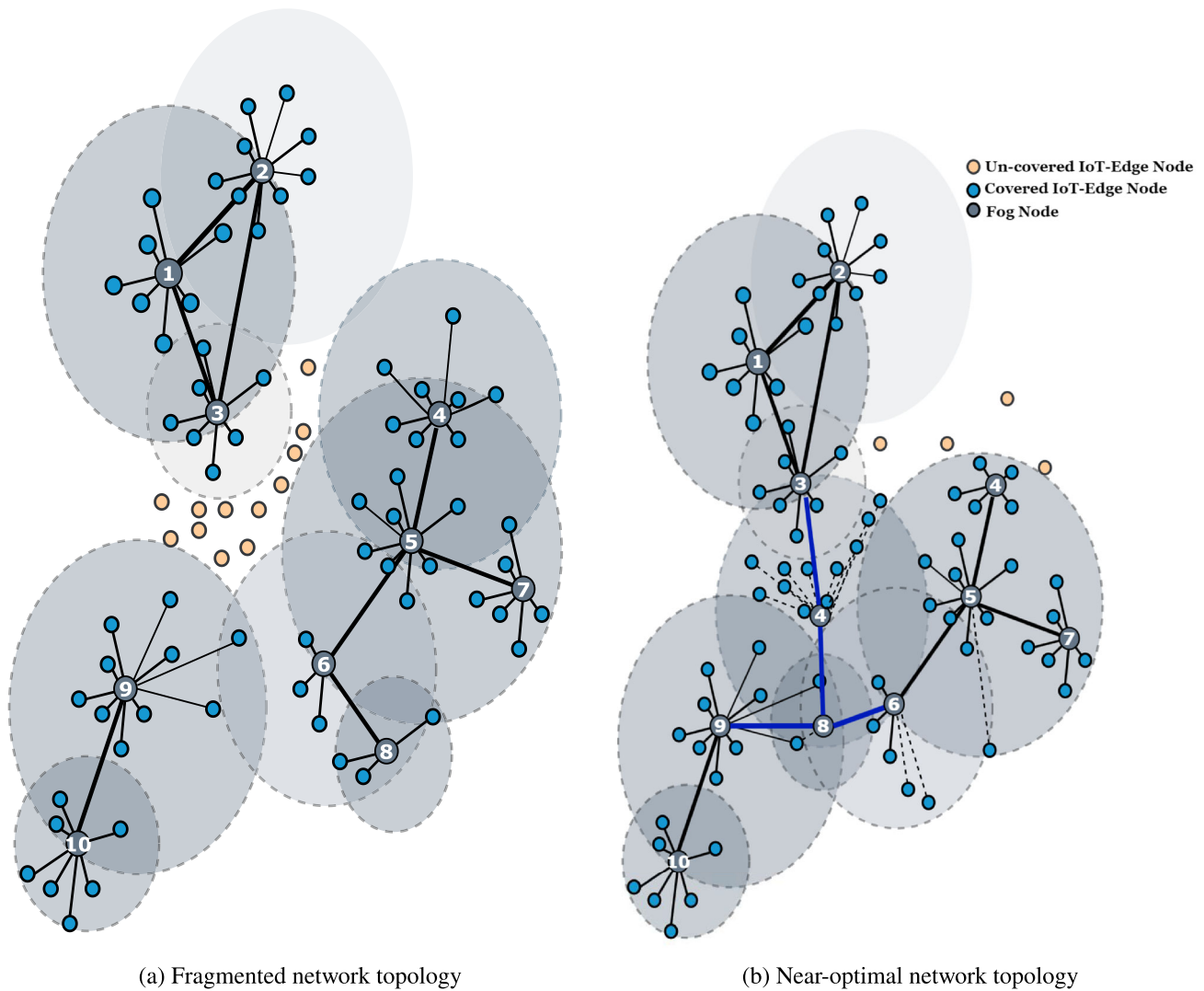
Let us consider a graph  $G$  that comprises  $h$  subgraphs, denoted as  $G_1, \dots, G_h$  in  $G$ , such that  $G$  equals the union of these subgraphs, i.e.,  $G = G_1 \cup G_2 \cup \dots \cup G_h$ , and It is important to note that the intersection of any two subgraphs,  $G_i$  and  $G_j$ , where  $i$  and  $j$  belong to the set  $1, \dots, h$ , is empty. i.e.,  $G_i \cap G_j = \emptyset$ ; for  $i, j \in 1, \dots, h$ . In order to analyze and evaluate the performance of the topology graph  $G$ , the following metrics, network coverage, and connectivity are considered to be optimized.

Network connectivity is defined as the size of the largest connected fog nodes subgraph. Where  $G$  is a graph and  $G_h$  is a subset of  $G$ . The network connectivity is calculated as follows:

$$\zeta(G) = \max_{i \in \{1, \dots, h\}} |G_i| \tag{3}$$

The network coverage function  $\gamma_i$  of an edge node  $i$ , relative to a fog node, is defined by a binary value as follows:





**Fig. 2** Fog connectivity and IoT-edge nodes coverage

$$\gamma_i = \begin{cases} 1 & \text{if the edge node } i \text{ is covered by at least one fog node} \\ 0 & \text{Otherwise} \end{cases}$$

Where

$$\Phi(G) = \sum_{i=0}^m \gamma_i \quad (4)$$

### 3.4 Objective function

To assess the performance of the introduced topology graph, we consider two objectives to optimize. The first is network connectivity, achieved by maximizing the size of the greatest subgraph component  $\zeta(G)$ , defined by equ. 3. The second objective is the node edge coverage  $\Phi(G)$ , defined by equ. 4. Therefore, we use the weighted sum method that transforms the multi-objective problem into a

scalar problem by summing each objective pre-multiplied by a user-provided weight. Our aggregated fitness function  $f(X)$  is defined as follows:

$$f(X) = \omega \cdot \frac{\zeta(G)}{n} + (1 - \omega) \cdot \frac{\Phi(G)}{m} \quad (5)$$

where  $\omega$  is a weighting coefficient between zero and one that describes the ratio in which the objectives are prioritized.

## 4 Marine predators approach

The Marine Predator Algorithm (MPA) has demonstrated remarkable success in diverse research domains, representing an innovative approach introduced in 2020 by [41]. This algorithm stands out for its ability to mimic the behavior of marine predators as they pursue their prey. The

primary motivation for the MPA algorithm is the extensive foraging strategy of ocean predators, namely Levy and Brownian motions [42], and encounter optimal rate policy in biological interactions between predator and prey. In this method, the predator employs a strategic trade-off based on the Brown and Levy model, utilizing the speed ratio between the predator and prey, as illustrated in Fig. 3. MPA adheres to the criteria that naturally regulate optimal foraging strategy and encounter rate policy in marine habitats [43]. According to the MPA definition, the MPA-based optimization algorithm is presented as follow:

In the initiation phase of the optimization process, a portion of the prey is uniformly dispersed within the search area. The mathematical model governing this dispersion is represented by Eq. 6:

$$X_0 = X_{min} + rand \times (X_{max} - X_{min}) \tag{6}$$

In this equation,  $X_0$  signifies the initial position of the prey,  $X_{max}$  and  $X_{min}$  denote the lower and upper bounds of the search space, respectively. The variable *rand* is a random number generated between 0 and 1. Essentially, Eq. 6 outlines the process through which the initial positions of prey within the optimization search space are determined uniformly, ensuring a dispersed and varied starting configuration for the marine predator algorithm.

During the optimization phase, each distributed solution is evaluated using the objective function, and the solution with the best objective value is chosen as the top predator. This latter is used to create a matrix called Elite, also referred to as E, which is based on the notion of the survival of the fittest. The elite matrix is demonstrated as follows:

$$Elite = \begin{bmatrix} X_{1,1}^1 & X_{1,2}^1 & \dots & X_{1,d}^1 \\ X_{2,1}^1 & X_{2,2}^1 & \dots & X_{2,d}^1 \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^1 & X_{n,2}^1 & \dots & X_{n,d}^1 \end{bmatrix}_{n \times d} \tag{7}$$

The variables  $X$ ,  $n$ , and  $d$ , respectively, represent the top

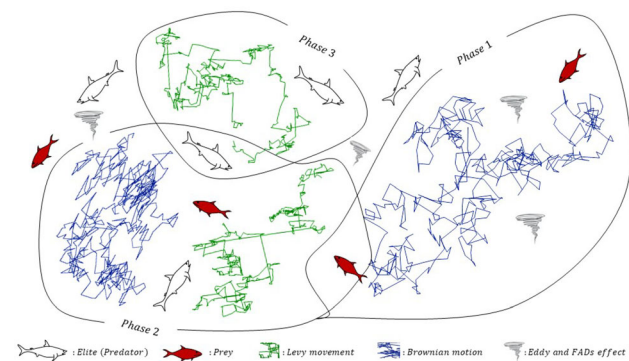


Fig. 3 Examination of the Three Distinct Phases in the Marine Predator Algorithm (MPA)

predator vector repeated  $N$  times to construct the Elite matrix, the number of search agents within the population, and the number of dimensions. The search agents predator and prey both their tasks are based on space exploration to locate food. The Elite matrix is updated at each iteration based on the superior predator compared to the top predator from the previous iteration. Typically, the prey matrix is constructed during the initiation phase when the predators change their positions. Consequently, a matrix called prey, which has the exact dimensions as the Elite matrix, is defined. The fittest prey from the original population is employed to construct the Elite matrix. The formulation of this matrix is as follows:

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \dots & X_{n,d} \end{bmatrix}_{n \times d} \tag{8}$$

$X_{ij}$  denotes the  $j$ th dimension of the  $i$ th prey. The two matrices described previously significantly affect the overall optimization problem. According to the movement modes of predators and prey, the optimization process of the MPA is divided into three phases. The specific operation process is as follows:

### 4.1 Stages

#### 4.1.1 High-velocity ratio (Stage 1)

During this stage, the prey moves at a higher velocity than the predator, prioritizing exploration. This stage typically occurs early in the iteration process and is defined as an exploration phase. Here, the prey employs the Brownian strategy to navigate the search space and identify potential areas that may contain the optimal solution. Mathematically, this stage can be represented as follows:

$$\text{While } t < \frac{1}{3} T_{max} \quad i = 1, 2, \dots, n \tag{9}$$

$$\begin{cases} \text{stepsize}_i = (R_B \otimes (Elite_i - R_B \otimes prey_i)) \\ prey_i = prey_i + P \times R \otimes \text{stepsize}_i \end{cases} \tag{10}$$

where  $t$  represents the current iteration count,  $T_{max}$  represents the algorithm's maximum cycle count,  $\text{stepsize}$  represents the motion scale factor,  $R_B$  represents a Brownian walk random vector that follows the normal distribution,  $Elite_i$  represents the elite matrix formed by the top predator,  $prey_i$  represents the prey matrix,  $\otimes$  stands for elementwise multiplication,  $R$  is a random variable with a range of  $[0, 1]$ , whereas  $P$  is a fixed value set to 0.5.

### 4.1.2 Relative unit velocity (Stage 2)

This stage is based on exploration and exploitation [44]. After completing the first third of the exploration process, predator and prey explore the search space at the same velocity to search for their own food. This is due to the proximity of the potential locations that may contain the ideal solution; therefore, this stage is considered as an intermediate optimization stage. In summary, in this stage, the exploration process will be progressively converted into the exploitation process. Specifically, the predators will be responsible for the exploration, while the prey will be responsible for the exploitation. Hence, to model this stage mathematically, the population is divided into two parts: the first is exploited by prey using a Lévy walk. At the same time, the second is explored by the predator using Brownian motion. The specific mathematical formulations can be observed in Eqs. 12 and 14.

$$\text{While } \frac{1}{3}T_{\max} < t < \frac{2}{3}T_{\max} \quad i = 1, 2, \dots, n/2 \quad (11)$$

$$\begin{cases} \text{stepsize}_i = (R_L \otimes (\text{Elite}_i - R_L \otimes \text{prey}_i)) \\ \text{prey}_i = \text{prey}_i + P \times R \otimes \text{stepsize}_i \end{cases} \quad (12)$$

$$\text{While } \frac{1}{3}T_{\max} < t < \frac{2}{3}T_{\max} \quad i = n/2, \dots, n \quad (13)$$

$$\begin{cases} \text{step size}_i = (R_B \otimes (R_B \otimes \text{Elite}_i - \text{prey}_i)) \\ \text{prey}_i = \text{Elite}_i + P \times CF \otimes \text{step size}_i \end{cases} \quad (14)$$

$R_L$  is the random vector facilitating Lévy's mobility and  $CF$  is an adaptive variable determined by equ. 15; works on controlling the predator's path.

$$CF = \left(1 - \frac{t}{T_{\max}}\right)^{\frac{2t}{T_{\max}}} \quad (15)$$

### 4.1.3 Slow velocity ratio (Stage 3)

In the latter phase of the optimization process, specifically during the final third of the maximum iteration, a significant change takes place in the predator's strategy. At this stage, the predator starts employing Lévy's motion to progressively close in on the prey. This modeling reflects a more accurate representation of the transitional stage as follows:

$$\text{While } t > \frac{2}{3}T_{\max}, \quad i = 1, 2, \dots, n \quad (16)$$

$$\begin{cases} \text{stepsize}_i = (R_L \otimes (R_L \otimes \text{Elite}_i - \text{prey}_i)) \\ \text{prey}_i = \text{Elite}_i + P \times CF \otimes \text{stepsize}_i \end{cases} \quad (17)$$

After progressing through previous stages, the MPA algorithm may encounter the risk of becoming trapped in sub-optimal solutions [45]. Therefore, environmental factors such as Eddy formation and fish aggregation devices (FADs)

have been identified. Eddy formations, characterized by swirling water currents, create areas that can trap and concentrate prey, presenting challenges for predators. Similarly, fish aggregation devices are structures designed to attract fish, impacting the movement patterns of predators. In the context of MPA, these environmental influences are represented in its mathematical model, notably within equ. 18.

$$\text{prey}_i = \begin{cases} \text{if } r \leq P_f \\ \text{prey}_i + CF[X_{\min} + R \otimes (X_{\max} + X_{\min})] \otimes U \\ \text{if } r > P_f \\ \text{prey}_i + [P_f(1-r) + r] \times (\text{prey}_{r1} - \text{prey}_{r2}) \end{cases} \quad (18)$$

Here,  $P_f$  indicates the likelihood that FADs would affect the optimization process;  $U$  denotes a binary vector containing 0 and 1 value;  $r$  a distinct value falling between [0, 1]; and the values  $r1$  and  $r2$ , produced at arbitrary from the broad population range which stands for the random indices of the prey matrix.

In many optimization algorithms, it is common for new solutions to replace older ones without considering their potential value. However, the MPA offers an advantage in data conservation by preserving the prey's previous location. In this approach, the algorithm evaluates both the current solution and the previous one (or solutions) based on their fitness values. If the current solution is deemed superior to the previous one, the algorithm retains it and discards the previous one. Conversely, if the previous solution is found to be better, the algorithm keeps it and discards the current one. By employing this mechanism, the MPA ensures that potentially valuable solutions are not prematurely discarded. The key steps of the Marine Predator Optimization (MPO) method are outlined in Algorithm 1.

In the fog node placement problem, the Marine Predator Algorithm (MPA) can be conceptualized with fog nodes acting as predators and potential positions for fog nodes as prey. Fog nodes, serving as predators, actively seek out optimal positions within the 2D area to maximize network coverage and connectivity. The exploration strategy entails searching for new promising areas within the 2D space where fog nodes could be deployed, representing unexplored prey locations. Concurrently, the exploitation strategy involves exploiting promising regions already discovered by fine-tuning fog node positions iteratively to enhance network coverage and connectivity. Through this iterative process, fog nodes navigate the search space, dynamically balancing exploration and exploitation to achieve near-optimal solutions for fog node placement, ultimately ensuring comprehensive coverage and robust connectivity in fog computing environments. In summary, MPA iteratively adjusts fog node positions based on prey encounters, balancing exploration and exploitation to improve both metrics. Termination criteria guide the



algorithm's convergence, ensuring the final fog node positions represent a near-optimal placement guaranteeing comprehensive coverage and connectivity. Through rigorous analysis, we aim to evaluate MPA's efficacy in addressing the fog node placement problem and further refine the optimization process.

#### 4.1.4 Time complexity evaluation of the marine predator algorithm

The time complexity of the Marine Predator Algorithm (MPA) is primarily determined by two key operations within each iteration: calculating the fitness values of individuals and updating the prey positions. These operations scale linearly with the size of the population ( $N$ ), the dimensionality of the search space ( $d$ ), and the maximum number of iterations ( $Max\_Iter$ ). As a result, the overall time complexity of the MPA can be briefly represented as  $O(N \times d \times Max\_Iter)$ . This characterization provides a clear understanding of how the algorithm's computational requirements grow with changes in input parameters, facilitating analysis and optimization efforts for practitioners seeking to deploy the MPA in various optimization tasks.

**Algorithm 1** Pseudo code of Marine Predator Optimization algorithm

---

**Input:** Search agents (Prey) population  
 $X_i (i = 1, 2, 3, \dots, n)$

**Output:** Optimal fitness value of fog node deployment  
 $fit_{best}$

```

1 while iter < MaxIter do
2   Calculate the fitness value of each
    $py_i | i = 0, 1, 2, 3, \dots, N$ 
3   Build the elite Matrix E and accomplish memory
   saving
4   if iter <  $\frac{MaxIter}{3}$  then
5     Update prey based on equation 10.
6   end
7   if  $\frac{MaxIter}{3} < iter < \frac{2 \cdot MaxIter}{3}$  then
8     For the first half of the populations
       ( $i = 1, \dots, n/2$ )
9     Update prey based on equation 12.
10    For the other half of the populations
       ( $i = n/2, \dots, n$ )
11    Update prey based on equation 14.
12  end
13  if iter >  $\frac{2 \cdot MaxIter}{3}$  then
14    Update prey based on equation 17.
15  end
16  Accomplish memory saving and Elite update
17  Applying FADs effect and update based on
   equation 18.
18  Iter = Iter + 1
19 end

```

---

## 5 Experimental and results

In this section, we explore a scenario set in a smart city environment where a static network is established to support a range of IoT applications, including traffic monitoring, environmental sensing, and smart lighting. The network comprises static fog servers distributed throughout the city, with a particular focus on enhancing public safety through intelligent surveillance. These fog servers are strategically located in high-traffic areas, public spaces, and critical infrastructure points such as transportation hubs and governmental buildings. By deploying fog nodes in these critical locations, the network can effectively cover and process real-time video feeds from surveillance cameras, analyze data locally, and respond swiftly to potential security threats or emergencies. This scenario underscores the importance of leveraging fog computing to support urban security measures and ensure the safety of residents and visitors within the smart city ecosystem. Furthermore, to evaluate our work, we implement the Marine Predator Algorithm (MPA) within our proposed architecture, which is structured as a three-tier IoT-edge fog-based system. In this architecture, IoT-edge devices in the first tier generate data, while the second tier is comprised of fog servers, and the final tier integrates a resource-abundant cloud. This hierarchical design optimizes resource deployment and utilization by defining the responsibilities for each tier. Subsequently, we conduct several scenarios with varying settings and compare the results against different algorithms. This comparative analysis allows us to assess the effectiveness and efficiency of the MPA algorithm within the given architecture.

### 5.1 Experimental setup

In the experimental setting, we implement a practical application using Matlab 2019Ra. This application permits the calculations to be conducted as fast as possible. Based on the pseudo-code presented in algorithm 1, we conducted our experiment in a rectangular area of 1000 m  $\times$  1000 m with normal/uniform distribution to fog nodes using an AMD Ryzen 7 5700U with 8 cores with a 4.3 GHz clock speed, 8GB of memory, and a Windows 10 environment. The experimental setup enables us to repeat the experiment under various conditions and restrictions, allowing us to conduct the experiment in the control environment with the desired set of settings. Table 2 provides a list of parameters and their corresponding values derived and established by various initial experiments.

**Table 2** Network Parameters

Parameter	Value	Initial value
Fog nodes density	[10, 120]	45
Edge nodes density	[30, 200]	120
Communication range	[90, 200]	100 m
Area width	1000 m	1000 m
Area height	1000 m	1000 m

## 5.2 Experimental results

We have conducted various trials to evaluate the connectivity and coverage of the proposed algorithm compared to the following baselines: harris hawks optimization (HHO) [46], particle swarm optimization (PSO) [47], and sine cosine algorithm (SCA) [48] algorithms. Since our target function involves two competing priorities and the network may experience inadequate connectivity and coverage as illustrated in Figs. 2a, 2b. Therefore, the conducted tests have been carried out under different network parameters to establish the best weight  $\omega$  of the defined objective function for comparing the performance of the proposed algorithm to other methods.

### 5.3 Fitness function study under different $\omega$ Values

To determine an appropriate weight-coefficient  $\omega$ , we analyze the user coverage and fog nodes connectivity simultaneously with different values of  $\omega$ , and we employ the convergence curves, which are considered the most critical analysis tool to understand the algorithm's behaviors while developing an optimal solution using meta-heuristics algorithms. Therefore, several iterations were conducted to identify the optimum weight-coefficient  $\omega$  that satisfies the termination criteria outlined in equ. 5. We execute the experiments 10 times, each of which has 1000 iterations. Furthermore, each experiment was associated with different values of  $\omega$ . In each experiment, 1000 instances are employed to determine the effect of changing the weight-coefficient  $\omega$  value. We observed that when using a larger value of  $\omega$ , the user coverage is low, which means fog nodes tend to crowd in a small area, leaving many IoT-edge nodes uncovered. In contrast, when using a smaller value, the user coverage is high, which means the fog nodes are dispersed throughout the area and can cover the majority of IoT-edge nodes independently but cannot make up a larger component (i.e., High connectivity). Therefore, we can conclude to some extent that the weight

coefficient  $\omega$  plays a crucial role in balancing our two objectives, namely  $\zeta$  and  $\Phi$ . As a result, it becomes necessary to determine an appropriate value for  $\omega$  that can effectively satisfy both criteria simultaneously. Through extensive testing, Fig. 4 demonstrates that a value of  $\omega = 0.5$  achieves an excellent trade-off, yielding favorable results.

#### 5.3.1 Comparison of algorithms convergence

Fig. 5 illustrates the results obtained from the studied systems concerning various network configuration values specified in Table 3. As observed at the beginning of the experiment, all algorithms, including MPA, undergo significant changes in their fitness values as iterations progress. However, what sets MPA apart is its resilience against premature convergence, a common issue faced by other algorithms like HHO, PSO, and SCA. Premature convergence can hinder the ability of algorithms to escape local optima, ultimately limiting their effectiveness in finding globally optimal solutions. In contrast, MPA's unique search strategy allows it to maintain a balance between exploration and exploitation, preventing premature convergence and facilitating the discovery of high-quality solutions. An essential aspect of an algorithm's overall performance is its convergence speed, which directly impacts its efficiency and effectiveness in finding optimal solutions. As depicted in Fig. 5, our experiments demonstrate that MPA exhibits exceptional potential and achieves faster convergence compared to other algorithms. This faster convergence, coupled with the ability to find optimal fitness function values, underscores the efficacy of MPA in addressing the complexities of fog node deployment. By leveraging MPA, we can effectively optimize fog computing environments, ensuring robust network coverage, connectivity and deployment costs.

#### 5.3.2 Comparison of algorithms terms of connectivity and coverage

To evaluate the performance of our proposed algorithm in comparison to PSO, HHO, and SCA algorithms for the given objective function  $f$  (as defined in equ. 5), we assess both the network coverage and user coverage. Figure 6 illustrates how our algorithm achieves a higher coverage of IoT-edge nodes using fewer fog nodes. In Fig. 6a, it is evident that our algorithm achieves a connectivity rate of 40%, with more than 90% of the target population being covered, as shown in Fig. 6b. In contrast, PSO, HHO, and SCA algorithms only achieve connectivity rates of (45, 45, 41)% for fog servers, covering (67, 67, 79)% of terminal IoT-edge devices, respectively. These results indicate that

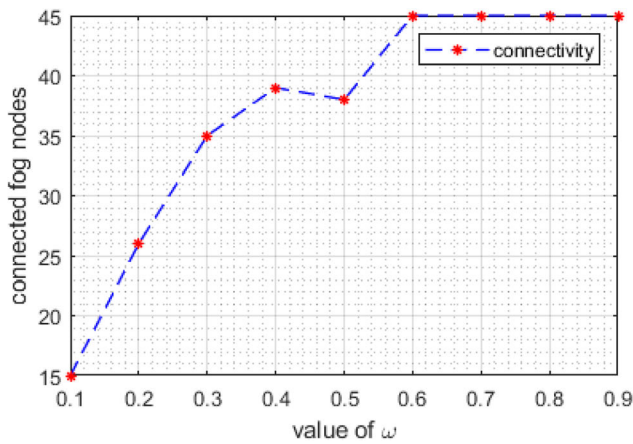


Fig. 4 Analysis of fitness functions across varying ω Parameters

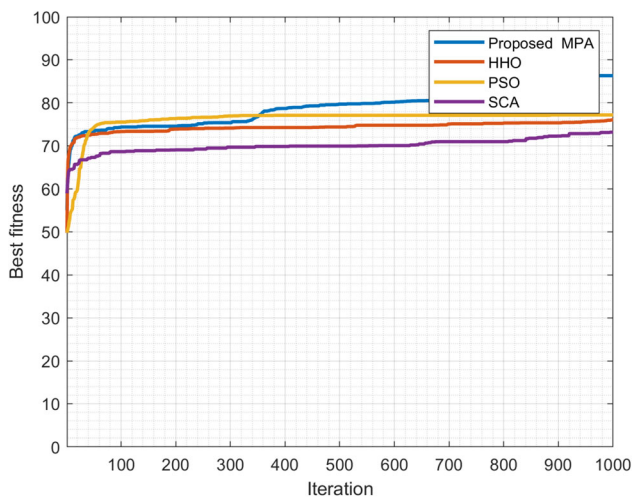
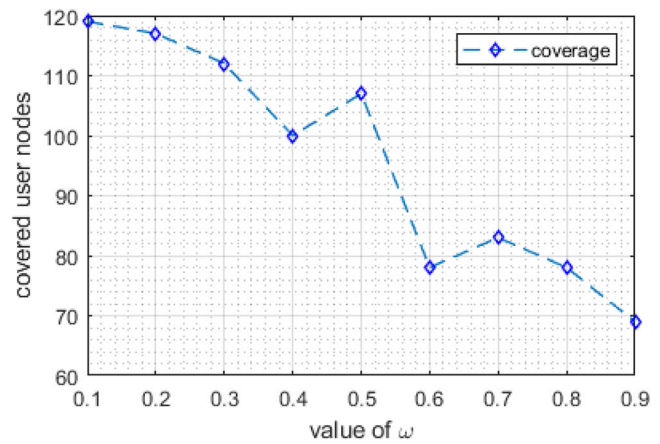


Fig. 5 Convergence behavior of the studied algorithm

Table 3 Algorithm Parameters Configurations

Algorithms	Parameters Configurations
MPA	$FAD_s = 0.1, P = 0.5$
SCA	$a = 2$
PSO	$V_{Max} = 6, W_{Max} = 0.9, W_{Min} = 0.2, c1 = c2 = 2$
HHO	No parameters for control

our proposed MPA algorithm outperforms the other methods.

### 5.4 Evaluation criterion

In order to have a better understanding of the adjustments that have been made to the optimization objectives, we have evaluated the performance of the proposed marine predator’s optimization algorithm against the previous

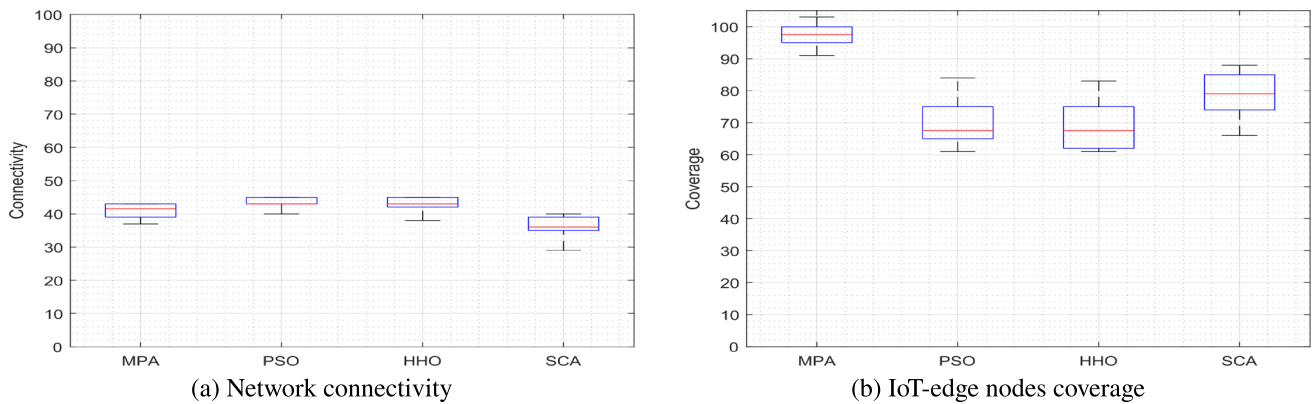
methods in terms of the following evaluation metrics, namely; Network connectivity ( $\Phi$ ), user coverage ( $\zeta$ ), and objective function value  $f$ , we assess our experiment by varying the number of edge nodes, the number of fog nodes, and the fog nodes’ communication range.

#### 5.4.1 Scalability in terms of network size

To analyze the network coverage and connectivity, we establish a fog network by deploying a variety of network sizes (N), varying from 10 to 120 fog nodes as shown in Figs. 7, 8 and 9. As can be seen in Fig. 7a, the number of connected fog nodes increases as additional fog nodes are added to the network, which reflects the network connectivity. As predicted, increasing the number of fog nodes increases network connectivity. Hence, this is due to the fact that by adding additional fog nodes to the network, the isolated network connects to the rest of the network. As a direct consequence of this, a wider network will be formed. While in terms of coverage, Fig. 7b displays the relationship between the number of fog nodes and the total number of covered IoT-edge nodes. The results show that our proposed algorithm finds optimum coverage, with more IoT-edge nodes covered as predicted. Also, It demonstrates the scalability of the fog network architecture by showing how adding more fog nodes improves network connectivity and expands coverage, indicating that the system can scale up effectively to accommodate a larger number of fog nodes.

#### 5.4.2 Scalability with increasing number of IoT-edge devices

Figure 8 shows the network coverage and connectivity results by varying the total number of IoT-edge nodes from 30 to 200. From figure 8a, it is clear that the number of connected fog nodes is always somewhere between 70%



**Fig. 6** Assessing connectivity and coverage across investigated algorithms

and 90% because our algorithm tries to connect all accessible fog nodes to encompass additional edge nodes. Furthermore, figure 8b shows the network's overall coverage, which remains stable between 70% and 93%. We observed that when the number of IoT-edge nodes increased, the percentage of coverage and connectivity remained consistent because the IoT-edge node distribution was uniform, which implies that the likelihood of an IoT node falling into a covered or uncovered region is the same. This analysis illustrates that, despite fluctuations in the number of edge nodes, the network's coverage and connectivity remain consistent. This highlights the scalability of the system to handle varying densities of edge nodes while maintaining stable performance.

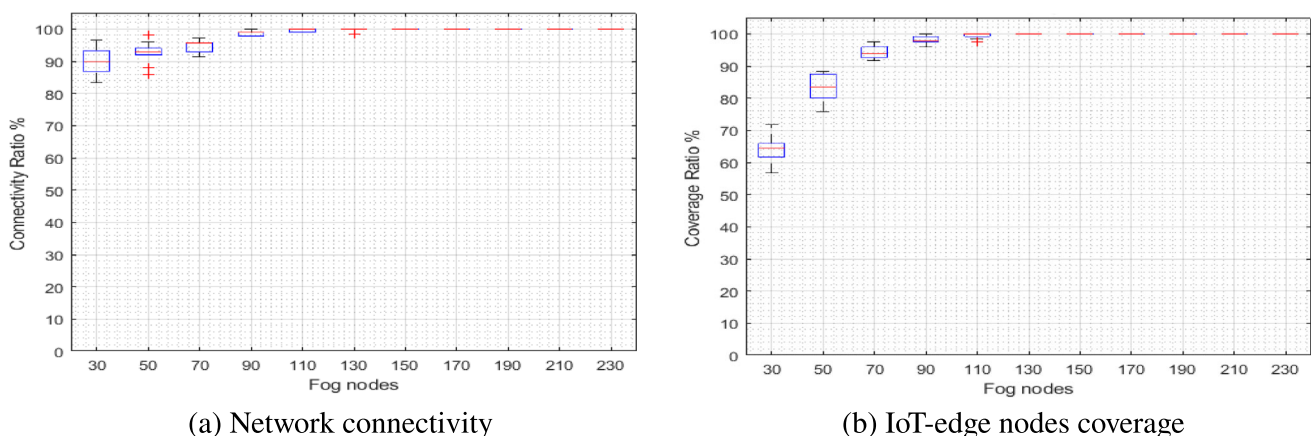
#### 5.4.3 The impact of communication range

Fig. 9, shows the impact of communication range adjustments on network coverage and connectivity performance across multiple experimental runs by adjusting the range of the communication from 90 ms up to 200 ms, respectively. The result in Fig. 9a shows that once the communication

range exceeds 140 ms, all fog nodes start attempting to communicate with each other because almost isolated sub-networks connect to the rest of the network, forming a single sizeable giant network. Figure 9b shows the impact of communication range changes on IoT-edge nodes coverage. Results proved that the number of covered edge nodes grows proportionally with the communication range. According to Fig. 9b, when the communication range goes over and above 140 ms, almost terminal nodes of the network are covered. Hence, 140 m is considered a crucial communication range.

#### 5.4.4 Experimental objective function impact

This part shows the impact of user coverage and network connectivity suggested in the objective function 5 in the performance of the proposed algorithm. Using the same experimental parameters as before, tables 4, 5, and 6 outline accurate numerical results of the objective function obtained by the proposed algorithm. Results in table 4, show that when increasing the number of fog nodes from 30 to 230, the covered IoT-edge nodes percentage and



**Fig. 7** Effect of fog nodes density on coverage and connectivity



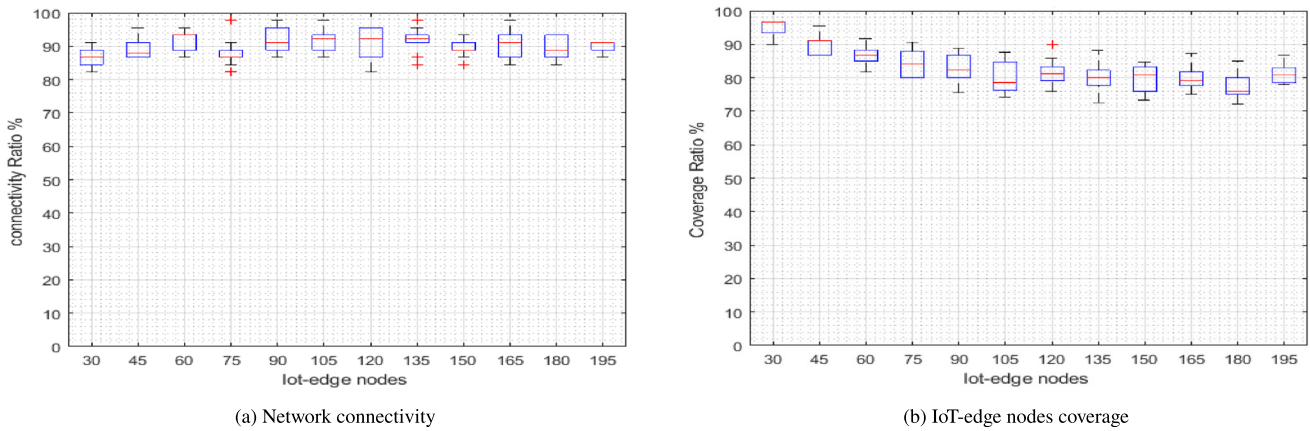


Fig. 8 Effect of IoT-edge nodes density on coverage and connectivity

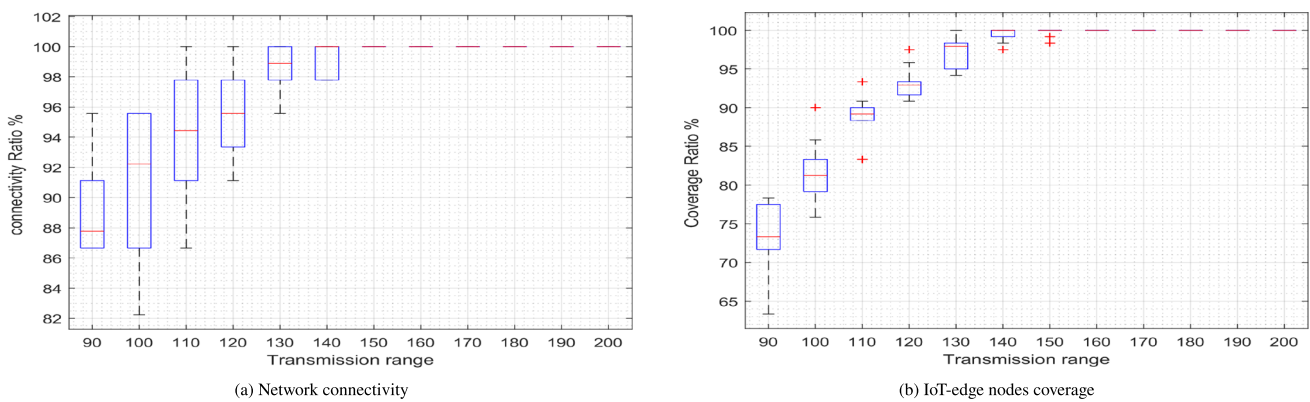


Fig. 9 Investigating Coverage and Connectivity Dynamics under Different Communication Ranges

value of objective function rise correspondingly. In addition, we obtained a connected network topology nearly all the time. However, upon adding more fog nodes to the network, our proposed algorithm gives more excellent IoT-edge node coverage and the best fitness value  $f$ . According to the statistics that are provided in table 4, when the number of fog nodes changed from 30 to 230, both the percentage of covered edge nodes and the value of the objective function increased proportionally. Quite notably, our network structure is nearly always interconnected. However, the obtained results affirmed that the proposed approach provides better network coverage and better value to the fitness function whenever additional fog nodes are added to the network.

By varying the number of terminal IoT-edge nodes from 30 to 195, results in table 5 show that the percentage of user coverage and network connectivity, as well as the objective function value  $f$ , almost remained practically unchanged, and this occurs because node placement distribution is randomly affected. Since the whole network is almost connected for all IoT-edge node values, it is remarkable that even with the addition of terminal nodes, the coverage value will remain stable, resulting in the same

Table 4 Coverage, connectivity, and fitness value vs Fog node density

Fog	Connectivity(%)	Coverage (%)	Function
30	90,00	64,25	77,13
50	92,60	83,50	88,05
70	94,86	94,00	94,43
90	98,67	98,00	98,33
110	99,73	99,50	99,61
130	99,85	100,00	99,92
150	100,00	100,00	100,00
170	100,00	100,00	100,00
190	100,00	100,00	100,00
210	100,00	100,00	100,00
230	100,00	100,00	100,00

proportion of all edge nodes. Similarly, statistics analysis demonstrates that the proposed algorithm performs better than others and achieves better coverage. The samples in table 6 illustrate how the communication range of fog nodes affects both coverage and network connectivity.



**Table 5** Coverage, connectivity, and fitness value vs IoT-edge node density

IoT-edge	Connectivity(%)	Coverage (%)	Function
30	86,7	95,0	90,83
45	89,1	90,4	89,78
60	91,8	86,8	89,31
75	88,0	84,5	86,27
90	91,8	82,7	87,22
105	92,2	80,3	86,25
120	90,9	81,8	86,36
135	91,8	80,4	86,07
150	89,6	79,8	84,68
165	90,9	79,7	85,29
180	89,3	77,2	83,25
195	90,0	81,1	85,54

**Table 6** Coverage, connectivity, and fitness value vs communication range

Range	Connectivity (%)	Coverage (%)	Function
90	88,89	73,42	79,61
100	90,89	81,83	85,46
110	94,00	88,50	90,70
120	95,56	93,17	94,12
130	98,44	97,00	97,58
140	99,33	99,50	99,43
150	100,00	99,75	99,85
160	100,00	100,00	100,00
170	100,00	100,00	100,00
180	100,00	100,00	100,00
190	100,00	100,00	100,00
200	100,00	100,00	100,00

Simply put, an increase in communication range will lead to an increase in coverage, which in turn will improve the fitness value. Moreover, the fitness value that can be attained using the proposed algorithm is more significant than that which can be obtained using the other methods.

## 6 Conclusion

To optimize network performance and ensure seamless user experiences at the network's edge, it's crucial to strategically place data and resources, especially considering the limitations of IoT edge devices and user

requirements. In this paper, we propose a three-tier IoT edge-fog-cloud integration architecture and leverages metaheuristic algorithms to facilitate resource deployment. Our approach revolves around defining an aggregated bi-objective function with the dual aim of maximizing network connectivity, as measured by the number of connected fog nodes, and enhancing the coverage of IoT-edge nodes. To tackle this challenge, we introduce the Multi-Objective Particle Swarm Optimization (MPA) algorithm, designed to efficiently address these objectives.

Noting the No Free Lunch Theorem (NFL Theorem), which asserts that no single algorithm universally excels across all problem domains, we empirically demonstrate the superiority of the MPA algorithm over alternative algorithms. Our findings show a notable improvement, with the MPA algorithm outperforming other algorithms by approximately 50%. This underscores the importance of selecting algorithms tailored to the specific characteristics of the problem at hand. Furthermore, our experimental results underscore the efficacy of our proposed algorithm in establishing network connections, consistently achieving complete network connectivity.

While our empirical findings clearly demonstrate the superior performance of the MPA algorithm compared to alternative methods, we recognize the importance of further exploration and enhancement in various areas. One promising direction for future research involves examining how the algorithm can adapt to dynamically changing IoT environments, evaluating its effectiveness in maintaining optimal performance, such as deploying UAV nodes in affected areas to provide services for users. Additionally, there is potential to improve the MPA algorithm to be tailored to the emerging IoT technologies, which could significantly enhance its capabilities and broaden its applicability.

**Acknowledgements** This work was supported by The National Natural Science Foundation of China under Grant No. 61872038.

**Author Contributions** AN, NN and SD: designed the system, HSN: supervised the study and received the funding. all authors implemented the experiment, validated the results and wrote/revised the manuscript.

**Funding** Open Access funding provided by the IReL Consortium.

## Declaration

**Conflict of interest** The authors declare no Conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this

article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

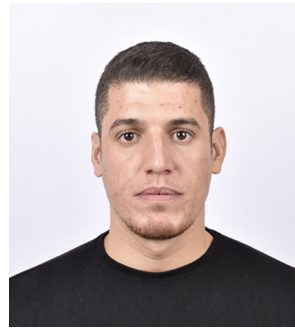
- Chen, H., Huang, S., Zhang, D., Xiao, M., Skoglund, M., Poor, H.V.: Federated learning over wireless iot networks with optimized communication and resources. *IEEE Int. Things J.* **9**(17), 6592–16605 (2022)
- Chakraborty, C., Mishra, K., Majhi, S.K., Bhuyan, H.K.: Intelligent latency-aware tasks prioritization and offloading strategy in distributed fog-cloud of things. *IEEE Trans. Ind. Inform.* **19**(2), 2099–2106 (2022)
- Hu, P., Dhelim, S., Ning, H., Qiu, T.: Survey on fog computing: architecture, key technologies, applications and open issues. *J. Netw. Comput. Appl.* **98**, 27–42 (2017)
- Naouri, A., Wu, H., Nouri, N.A., Dhelim, S., Ning, H.: A novel framework for mobile-edge computing by optimizing task offloading. *IEEE Int. Things J.* **8**(16), 13065–13076 (2021)
- Abdenacer N., Abdelkader N. N., Qammar A., Shi F., Ning H., Dhelim S.: Task Offloading for Smart Glasses in Healthcare: Enhancing Detection of Elevated Body Temperature, in: 2023 IEEE International Conference on Smart Internet of Things (SmartIoT), IEEE, pp. 243–250. (2023) <https://doi.org/10.1109/SmartIoT58732.2023.00044> <https://ieeexplore.ieee.org/document/10296320/>
- Dhelim, S., Aung, N., Kechadi, M.T., Ning, H., Chen, L., Lakas, A.: Trust2Vec: large-scale IoT Trust management system based on signed network embeddings. *IEEE Int. Things J.* **10**(1), 553–562 (2023)
- Islam M. M., Ramezani F., Lu H. Y., Naderpour M.: Optimal placement of applications in the fog environment: A systematic literature review, *Journal of Parallel and Distributed Computing* (2022). <https://www.sciencedirect.com/science/article/pii/S0743731522002465>
- Aung N., Kechadi T., Zhu T., Zerdoumi S., Guerbouz T., Dhelim S.: Blockchain Application on the Internet of Vehicles (IoV), 2022 IEEE 7th International Conference on Intelligent Transportation Engineering (ICITE) (2022) 586–591 <https://doi.org/10.1109/ICITE56321.2022.10101404>. <https://ieeexplore.ieee.org/document/10101404/>
- Ben, Sada A., Naouri, A., Khelloufi, A., Dhelim, S., Ning, H.: A context-aware edge computing framework for smart internet of things. *Future Int.* **15**(5), 154 (2023)
- Wang, J., Li, D., Hu, Y.: Fog nodes deployment based on space-time characteristics in smart factory. *IEEE Trans. Ind. Inform.* **17**(5), 3534–3543 (2020)
- Aung, N., Dhelim, S., Chen, L., Lakas, A., Zhang, W., Ning, H., Chaib, S., Kechadi, M.T.: VeSoNet: traffic-aware content caching for vehicular social networks using deep reinforcement learning. *IEEE Trans. Intell. Transport. Syst.* **24**(8), 8638–8649 (2023)
- Aung, N., Dhelim, S., Chen, L., Ning, H., Atzori, L., Kechadi, T.: Edge-enabled metaverse: the convergence of metaverse and mobile edge computing. *Tsinghua Sci. Technol.* **29**(3), 795–805 (2024)
- Maiti, P., Apat, H.K., Sahoo, B., Turuk, A.K.: An effective approach of latency-aware fog smart gateways deployment for iot services. *Int. Things* **8**, 100091 (2019)
- Kumar, D., Baranwal, G., Shankar, Y., Vidyarthi, D.P.: A survey on nature-inspired techniques for computation offloading and service placement in emerging edge technologies. *World Wide Web* **25**(5), 2049–2107 (2022)
- Natesha, B., Guddeti, R.M.R.: Meta-heuristic based hybrid service placement strategies for two-level fog computing architecture. *J. Netw. Syst. Manag.* **30**(3), 47 (2022)
- Huangpeng, Q., Yahya, R.O.: Distributed iot services placement in fog environment using optimization-based evolutionary approaches. *Exp. Syst. Appl.* **237**, 121501 (2024)
- Nouri, N.A., Aliouat, Z., Naouri, A., Hassak, S., a.: An efficient mesh router nodes placement in wireless mesh networks based on moth-flame optimization algorithm. *Int. J. Commun. Syst.* **36**(8), e5468 (2023)
- Nouri, N.A., Aliouat, Z., Naouri, A., Hassak, S.A.: Accelerated pso algorithm applied to clients coverage and routers connectivity in wireless mesh networks. *J. Ambient Intell. Human. Comput.* **14**(1), 207–221 (2023)
- Singh, S., Vidyarthi, D.P.: Fog node placement using multi-objective genetic algorithm. *Int. J. Inform. Technol.* **16**(2), 713–719 (2023)
- Wang, W., Ning, H., Shi, F., Dhelim, S., Zhang, W., Chen, L.: A survey of hybrid human-artificial intelligence for social computing. *IEEE Trans. Human-Mach. Syst.* **52**(3), 468–480 (2021)
- Khelloufi A., Ning H., Naouri A., Sada A. B., Qammar A., Khalil A., Mao L., Dhelim S.: A Multimodal Latent-Features-Based Service Recommendation System for the Social Internet of Things, *IEEE Transactions on Computational Social Systems* 1–16 (2024) <https://doi.org/10.1109/TCSS.2024.3360518> <https://ieeexplore.ieee.org/document/10440644/>
- Nouri, N.A., Naouri, A., Dhelim, S.: Accurate range-based distributed localization of wireless sensor nodes using grey wolf optimizer. *J. Eng. Exact Sci.* **9**(4), 15920 (2023)
- Zhang, Y., Liu, Z., Bi, Y.: Node deployment optimization of underwater wireless sensor networks using intelligent optimization algorithm and robot collaboration. *Sci. Rep.* **13**(1), 15920 (2023)
- Dhelim, S., Huansheng, N., Cui, S., Jianhua, M., Huang, R., Wang, K.I.-K.: Cyberentity and its consistency in the cyber-physical-social-thinking hyperspace. *Comput. Electr. Eng.* **81**, 106506 (2020)
- Wang, J., Luo, D., Peng, F., Chen, W., Liu, J., Zhang, H.: Wireless sensor deployment optimisation based on cost, coverage, connectivity, and load balancing. *Int. J. Sens. Netw.* **41**(2), 126–135 (2023)
- Tang, C., Zhu, C., Guizani, M.: Coverage optimization based on airborne fog computing for internet of medical things. *IEEE Syst. J.* **17**(3), 4348–4359 (2023)
- ChinaTelecomSichuanBranch L.: Internet of things deployment based on fog computing systems: Security approach (2024)
- Pallewatta, S., Kostakos, V., Buyya, R.: Microfog: a framework for scalable placement of microservices-based iot applications in federated fog environments. *J. Syst. Softw.* **209**, 111910 (2024)
- Yoon, Y., Kim, Y.-H.: An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. *IEEE Trans. Cybern.* **43**, 1473–1483 (2013)
- Yu, W., Li, X., Zeng, Z., Luo, M.: Problem characteristics and dynamic search balance-based artificial bee colony for the optimization of two-tiered wsn lifetime with relay nodes deployment. *Sensors* (2022). <https://doi.org/10.3390/s22228916>
- Cong C.: A coverage algorithm for wsn based on the improved pso, in: 2015 International Conference on Intelligent Transportation, Big Data and Smart City, pp. 12–15. (2015) <https://doi.org/10.1109/ICITBS.2015.9>
- Deng, X., Yu, Z., Tang, R., Qian, X., Yuan, K., Liu, S.: An optimized node deployment solution based on a virtual spring

- force algorithm for wireless sensor network applications. *Sensors* (2019). <https://doi.org/10.3390/s19081817>
33. Qin, N., Xu, J.: An adaptive fish swarm-based mobile coverage in wsn. *Wireless Commun. Mobile Comput.* **2018**, 1–8 (2018). <https://doi.org/10.1155/2018/7815257>
  34. Wang, L., Wu, W., Qi, J., pu Jia Z.: Wireless sensor network coverage optimization based on whale group algorithm. *Comput. Sci. Inf. Syst.* **15**, 569–583 (2018)
  35. Xu Y., Ding O., Qu R., Li K.: Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization, *Applied Soft Computing* 68 (2018) 268–282. <https://doi.org/10.1016/j.asoc.2018.03.053><https://www.sciencedirect.com/science/article/pii/S1568494618301868>
  36. Akusta Dagdeviren Z.: Akram, Connectivity estimation approaches for internet of things-enabled wireless sensor networks, *Emerging Trends in IoT and Integration with Data Science, Cloud Computing, and Big Data Analytics* 22 (2022)
  37. Khalilpour, Akram V., Akusta, Dagdeviren Z., Dagdeviren, O., Challenger, M.: Pinc: pickup non-critical node based k-connectivity restoration in wireless sensor networks. *Sensors* (2021). <https://doi.org/10.3390/s21196418>
  38. Peng, M., Quek, T.Q., Mao, G., Ding, Z., Wang, C.: Artificial-intelligence-driven fog radio access networks: recent advances and future trends. *IEEE Wireless Commun.* **27**(2), 12–13 (2020)
  39. Ghosh, A., Mukherjee, A., Misra, S.: Sega: secured edge gateway microservices architecture for iiot-based machine monitoring. *IEEE Trans. Ind. Inform.* **18**(3), 1949–1956 (2021)
  40. Gilbert, G.M., Shililiandumi, N., Kimaro, H.: Evolutionary approaches to fog node placement in lv distribution networks. *Int. J. Smart Grid-ijSmartGrid* **5**(1), 1–14 (2021)
  41. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: a nature-inspired metaheuristic. *Exp. Syst. Appl.* **152**, 113377 (2020)
  42. Humphries, N., Queiroz, N., Dyer, J., Pade, N., Musyl, M., Schaefer, K., Fuller, D., Brunnschweiler, J., Doyle, T., Houghton, J., Hays, G., Jones, C., Noble, L., Wearmouth, V., Southall, E., Sims, D.: Environmental context explains lévy and brownian movement patterns of marine predators. *Nature* **465**, 1066–9 (2010). <https://doi.org/10.1038/nature09116>
  43. Bartumeus, F., Catalan, J., Fulco, U., Lyra, M., Viswanathan, G.: Optimizing the encounter rate in biological interactions: Lévy versus brownian strategies. *Phys. Rev. Lett.* **88**, 097901 (2002). <https://doi.org/10.1103/PhysRevLett.88.097901>
  44. Hussain, K., Salleh, M.N.M., Cheng, S., Shi, Y.: On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Comput. Appl.* **31**, 7665–7683 (2019)
  45. Zhong, K., Zhou, G., Deng, W., Zhou, Y., Luo, Q.: Mompa: multi-objective marine predator algorithm. *Comput. Methods Appl. Mech. Eng.* **385**, 114029 (2021)
  46. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. *Future Gen. Comput. Syst.* **97**, 849–872 (2019)
  47. Bonyadi, M.R., Michalewicz, Z.: Particle swarm optimization for single objective continuous space problems: a review. *Evolut. Comput.* **25**(1), 1–54 (2017). [https://doi.org/10.1162/EVCO\\_r\\_00180](https://doi.org/10.1162/EVCO_r_00180)
  48. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the

accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



**Abdenacer Naouri** is currently a Ph.D. candidate at the University of Science and Technology Beijing China, Beijing, China. He received his B.S. degree in computer science from the University of Djelfa Algeria, in 2011, and the M.Sc. degree in networking and distributed systems from the University of Laghouat Algeria, Laghouat, Algeria, in 2016. His current research interests include Cloud computing, Smart communication, machine learning,

Internet of vehicles and Internet of Things.



**Nabil Abdelkader Nouri** is an associate professor in university of Djelfa, Algeria. Received his BS's degree in computer sciences from the university of Laghouat, Algeria, in 2003 and his magister degree in networking and distributed systems from the university of Bejaia, Algeria, in 2007. His current research interests include Wireless Networking Design, Internet of Things, Performance Evaluation, Fog Computing and Optimization.



**Amar Khelloufi** received the B.S. degree (Hons.) in computer science from the Faculty of Sciences and Technology, Ziane Achour University of Djelfa, Djelfa, Algeria, in 2012, and the M.S. degree in distributed information systems from the Faculty of Sciences, University of Boumerdès, Boumerdès, Algeria, in 2014. He is currently pursuing the Ph.D. degree with the School of Computer and Communication Engineering, University of Science and



**Abdelkarim Ben Sada** received his B.Sc. in Computer Science in 2014 from the University of Djelfa Algeria, and his M.Sc. degree in 2016 majoring in Networking and Distributed Systems from the University of Laghouat Algeria. He is currently pursuing his Ph.D. degree at the University of Science and Technology Beijing China. His research interests include Computer Vision, Machine Learning and Internet of Things.



**Huansheng Ning** received his B.S. degree from Anhui University in 1996 and his Ph.D. degree from Beihang University in 2001. Now, he is a professor and vice dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research focuses on the Internet of Things and general cyberspace. He has presided many research projects including Natural Science Foundation of China,

National High Technology Research and Development Program of China (863 Project). He has published more than 150 journal/conference papers, and authored 5 books. He serves as an associate editor of IEEE Systems Journal (2013-Now), IEEE Internet of Things Journal (2014-2018), and as steering committee member of IEEE Internet of Things Journal (2016-Now).



**Sahraoui Dhelim** is a senior postdoctoral researcher at University College Dublin, Ireland. He was a visiting researcher at Ulster University, UK (2020-2021). He obtained his Ph.D. degree in Computer Science and Technology from the University of Science and Technology Beijing, China, in 2020. And a Master's degree in Networking and Distributed Systems from the University of Laghouat, Algeria, in 2014. He serves as a regular guest editor for several reputable journals. His research interests include Social Computing, Digital Agriculture, Deep-learning, Recommendation Systems and Intelligent Transportation Systems.