

Performance Evaluation of Metaheuristic Algorithms for IIoT Device Deployment in Obstacle-Constrained Environments

Samad Najjar-Ghabel* and Shamim Yousefi

Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran

*Corresponding author: s.najjar@uma.ac.ir

Submitted 30 November 2025; Revised 06 January 2026; Accepted 26 January 2026; Available online 23 February 2026.

Copyright © 2026 The Authors.

Abstract: The Industrial Internet of Things (IIoT) has transformed industrial systems through the integration of sensing technologies and automation. Despite these advantages, inadequate device placement leads to coverage holes that degrade system performance in complex environments. Optimizing IIoT device deployment is therefore essential to ensure reliable monitoring and communication. Due to the NP-hard nature of the problem, metaheuristic algorithms have been adopted for exploring complex search spaces. This paper presents a simulation-based performance evaluation of seven metaheuristic algorithms, including Antlion Optimization (ALO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Water Cycle Algorithm (WCA), Bat Algorithm (BA), and PSO with Learning Strategy and Crossover (PSOLC) for IIoT device deployment in obstacle-constrained industrial environments. The experimental environment is designed to simulate the characteristics of a factory layout. All experiments are conducted on a 100×100 m area under two obstacle configurations corresponding to 14% and 44% obstacle density to ensure statistical robustness. Coverage is defined as the percentage of the deployable area covered by the sensing range of at least one device, excluding obstacle regions, and is computed using a grid-based spatial coverage model. Algorithm performance is evaluated in terms of average coverage and the number of function evaluations required for convergence. The results indicate that ALO achieves the highest coverage (91.13%) with rapid convergence, outperforming the others, whereas BA exhibits the weakest performance. The study also outlines future research directions for improving metaheuristic-based IIoT deployment strategies in industrial environments.

Keywords: Coverage optimization; Device deployment; Industrial Internet of Things (IIoT); Metaheuristic algorithms; Obstacle-constrained environments.

1. INTRODUCTION

The Industrial Internet of Things (IIoT) integrates sensing technologies, data analytics and automated systems to enhance efficiency and safety across industrial sectors [1]. Emerging in the early 2010s alongside the rapid growth of internet-connected devices and data-driven strategies, IIoT has evolved from basic process monitoring to sophisticated cyber-physical systems that underpin Industry 4.0 [2], [3]. Today, IIoT enables intelligent industrial ecosystems by supporting applications such as predictive maintenance, smart manufacturing, energy optimization, logistics, and real-time process control [4], [5]. In manufacturing, IIoT facilitates smart factories through interconnected machines and predictive analytics that reduce downtime and improve operational efficiency [6]. In energy and transportation systems, it supports dynamic resource management, smart grids, traffic monitoring, and logistics optimization [7]. IIoT also plays a vital role in healthcare through remote patient monitoring and smart medical devices, improving care quality while reducing system burdens [8]. Additionally, IIoT enhances supply chain visibility, inventory management, and demand forecasting, leading to cost reductions and improved customer satisfaction [9]. Environmental monitoring applications further extend the impact of IIoT by enabling real-time assessment of conditions such as weather, soil, and air quality, supporting sustainable decision-making across industries [10].

Despite its wide range of applications, the large-scale adoption of IIoT faces several critical challenges, as illustrated in Figure 1. Data security and privacy remain major concerns due to the extensive connectivity of devices, which increases system vulnerability to cyberattacks and unauthorized data access [11]. Another significant challenge is the integration of legacy systems with modern IIoT technologies, as many existing industrial infrastructures were not designed for interoperability with interconnected and data-driven systems [12]. The lack of standardization across IIoT protocols further exacerbates interoperability issues, leading to increased system complexity and inefficiencies [2]. Scalability also poses a major challenge, as the rapid growth in connected devices demands robust infrastructures capable of handling large volumes of data without performance degradation [13]. In addition, the massive amount of data generated by IIoT devices creates challenges in data

storage, processing, and analysis, requiring advanced analytics [14]. Finally, deploying IIoT devices in harsh or remote environments introduces coverage and connectivity issues (coverage holes), where maintaining reliable communication and power supply becomes difficult [4]. Overall, these challenges highlight the need for efficient deployment strategies, secure architectures, and scalable solutions to enable reliable IIoT systems.

Coverage hole represents one of the most critical challenges in IIoT systems. It refers to areas within the monitoring environment that lack sufficient device coverage and create gaps in data collection and system visibility [15]. These blind spots reduce the reliability of IIoT applications and lead to incomplete monitoring and suboptimal decision-making. Effective IIoT device deployment is therefore essential to mitigate coverage holes by ensuring continuous coverage across the monitored area [16]. Achieving this requires strategic planning that accounts for device sensing range, environmental obstacles, and application-specific monitoring requirements. However, determining an optimal deployment configuration is an NP-hard problem, making exact solutions computationally infeasible for large-scale environments [17]. NP-hard problems refer to a class of computationally intractable optimization challenges for which no known polynomial-time algorithm exists. In this context, evolutionary optimization algorithms offer a powerful alternative to classical methods, as they are capable of exploring large, complex, and non-linear search spaces to identify near-optimal solutions [18]. Their adaptive mechanisms enable a balanced exploration–exploitation process, allowing the algorithm to avoid local optima while reducing computational overhead. Moreover, evolutionary algorithms are well-suited to handling multi-objective and dynamic deployment scenarios, making them a flexible approach for optimizing IIoT device placement and minimizing coverage holes.

Numerous studies have investigated device deployment in IIoT and sensor-based networks. The authors offer valuable classifications and analytical perspectives [19], [20], [21]. Existing surveys have examined range-based and range-free localisation methods, force-based and geometrical deployment algorithms, and a wide spectrum of optimization techniques, including evolutionary, swarm-intelligence, and metaheuristic approaches [18], [20]. These approaches provide important insights into coverage optimization, connectivity, scalability, and computational complexity in IIoT systems, as well as broader architectural challenges in IoT–Edge–Cloud environments. Other studies have focused on security-aware deployment and localisation techniques, highlighting vulnerabilities and mitigation strategies in IIoT. However, many of these reviews emphasise theoretical classifications, blanket coverage assumptions, or architectural considerations, and do not analyse deployment performance under explicit obstacle-constrained environments or provide comparative evaluations across multiple optimization criteria within a unified simulation framework.

To address this gap, this paper conducts a simulation-based comparative evaluation of seven metaheuristic algorithms, including Antlion Optimization (ALO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Water Cycle Algorithm (WCA), Bat Algorithm (BA), and PSO with Learning Strategy and Crossover (PSOLC) for IIoT device deployment in environments with non-penetrable obstacles. The simulation setup is designed to reflect key spatial constraints of industrial monitoring scenarios while remaining reproducible and algorithm-agnostic. The algorithms are evaluated based on multiple deployment performance criteria, including coverage maximisation, device overlap, overlap with obstacles, and convergence efficiency. This comparative analysis clarifies the strengths of metaheuristic algorithms for obstacle-constrained IIoT deployment and provides a focused benchmark for future research.

The rest of the paper is organized as follows: Section 2 provides the methodology, covering a detailed explanation of the problem and the evolutionary algorithms, with a focus on IIoT device deployment. In Section 3, we present the experimental analysis and results. Section 4 concludes the paper and suggests potential directions for future work.

2. METHODOLOGY

This section formulates the IIoT device deployment problem by defining the objective and key parameters. It then describes the evolutionary algorithms and presents a comparative evaluation of their performance for IIoT monitoring environments.

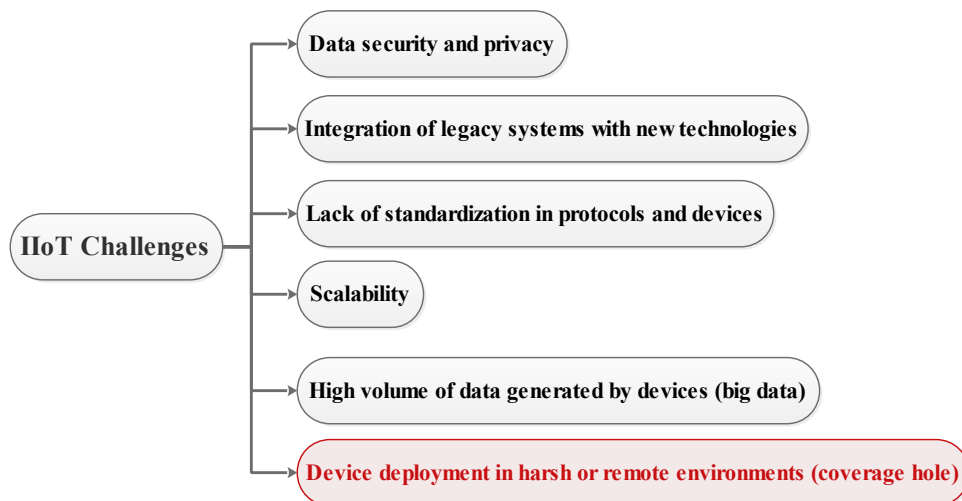


Figure 1. IIoT challenges.

2.1 Problem Statement

The IIoT system is modelled as $IIoT = (D, L)$, where $D = \{D_i\}$ denotes the set of sensing devices, and L represents the communication links. The monitoring region is bounded by a rectangle of size $W \times H$, which encloses the actual monitoring area. Any region inside the bounding box but outside the true monitoring polygon is treated as a non-penetrable obstacle to prevent unnecessary device deployment. Each device D_i is positioned at coordinates (x_i, y_i) with sensing radius C_i , following the Boolean disk model. A point s is considered covered by D_i if $\|s - (x_i, y_i)\| \leq C_i$ and s does not lie within an obstacle. Obstacles O_q are modelled as axis-aligned rectangles defined by the lower-left corner and dimensions (H_{O_i}, W_{O_i}) , and they are excluded from coverage requirements. To initialise the search space, an estimate of the Minimum Number of Devices (MND) required to cover the deployable area is computed using an area-packing approximation, as Equation 1 [22]:

$$MND = \left\lceil \frac{(H \times W) - \sum_{q=1}^M (H_{O_i} \times W_{O_i})}{\pi \times (\min_i C_i)^2} \right\rceil \quad (1)$$

which yields a dimensionless device count and serves as an upper-bound initial population size.

The optimization objective is to maximise spatial coverage while minimising wasted sensing caused by overlaps. Coverage is computed using a grid-based spatial estimator in which the deployable region is discretised into uniform cells; a cell is counted as covered if its centre lies within at least one sensing disk and outside all obstacles. Overall coverage is defined as the ratio of covered deployable cells to total deployable cells. During optimization, candidate solutions are evaluated using a composite fitness function that penalises (i) device-device overlap, (ii) device-obstacle overlap, and (iii) sensing coverage outside the monitoring boundary, weighted by coefficients ω_1 , ω_2 , and ω_3 . The weights are specified to ensure reproducibility (equal weighting ω_1 , ω_2 , and $\omega_3 = \frac{1}{3}$), with the final values reported in the experimental settings. Indeed, our fitness function is a unitless, normalized composite metric, obtained by aggregating the normalized contributions of the penalises, where lower fitness values indicate higher-quality IIoT device deployment solutions. Device-device overlap also follows standard circle-intersection logic with correct containment handling, as Equation (2):

$$overlap(D_i, D_j) = \begin{cases} 0 & \text{if } d_{ij} \geq C_i + C_j \\ \pi \times \min(C_i, C_j)^2 & \text{if } |C_i - C_j| \geq d_{ij} \\ area_{ij} & \text{otherwise} \end{cases} \quad (2)$$

where d_{ij} is the Euclidean distance between devices, and $area_{ij}$ is the two-circle intersection area. For boundary leakage, the overlap between a sensing disk and the area outside the monitoring boundary is computed according to three geometric scenarios illustrated in Figure 2. For each case (Figure 2(a) - (c)), the corresponding overlap area is computed using circular-segment geometry, where angles are obtained via $\arccos(\cdot)$. Numerical safeguards are applied by clamping inputs to $[-1, 1]$, treating tangential contacts as zero-area overlaps, and enforcing $overlap \in [0, \pi C_i^2]$. This deterministic workflow specifies how the scenarios in Figure 2 are selected and ensures robust handling of all edge cases, including near-tangent intersections and devices outside the monitoring boundary [23].

2.2 Metaheuristic Algorithms for IIoT Device Deployment

This section introduces metaheuristic optimization algorithms as effective tools for solving the IIoT device deployment problem by balancing exploration and exploitation in obstacle-constrained environments [24], [25]. It describes seven representative algorithms, including ALO, ABC, PSO, GA, WCA, BA, and PSOLC and explains their mechanisms in the context of coverage maximization and efficient deployment. The section also compares these algorithms in terms of convergence behaviour, solution quality, scalability, computational complexity, and energy efficiency to evaluate their suitability for IIoT applications.

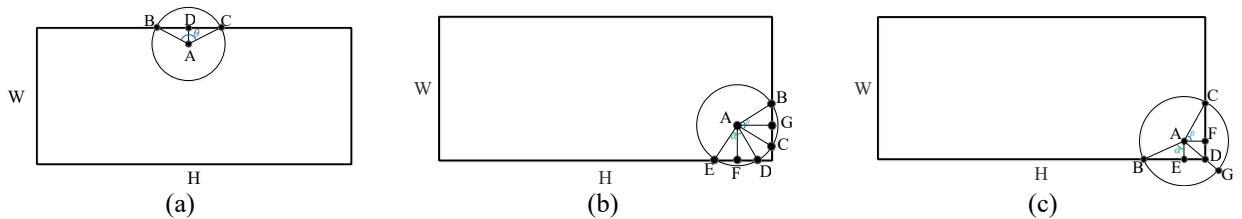


Figure 2. The overlap between the device D_i and the area outside the monitoring environment: (a) intersection of the sensing disk with a single boundary edge, (b) intersection with two boundary edges, and (c) corner intersection involving two boundary edges and their intersection point.

2.2.1 Antlion Optimization (ALO)

ALO, introduced by Mirjalili [26], is a population-based metaheuristic inspired by the hunting mechanism of antlions. In our formulation, each candidate solution (ant or antlion) encodes a deployment configuration as a continuous vector of device coordinates (x_i, y_i) for all deployed devices, representing their spatial positions within the monitoring area. The initial populations of ants and antlions are generated using uniform random sampling over the feasible deployment space. Constraint handling is performed using a repair-based boundary control strategy, whereby any device position that violates spatial boundaries or intersects non-penetrable obstacles is projected back to the nearest feasible location. It ensures valid deployments throughout the optimization process. The fitness of each solution is evaluated using the composite objective function defined in Section 2.1, which promotes coverage maximisation while penalising device–device overlap, device–obstacle overlap, and boundary leakage. During each iteration, ants perform stochastic random walks influenced by selected antlions, simulating the trapping mechanism, while elitism preserves the best-performing deployment configuration found so far.

Algorithm-specific parameters are reported in Table 1, where default values recommended in the ALO formulation are adopted without aggressive tuning to ensure fair comparison. All simulations are executed using fixed random seeds, which are reported to guarantee repeatability. While ALO demonstrates strong global search capability and robustness against local optima, its convergence speed and computational cost are influenced by population size and iteration count, making parameter transparency essential for accurate performance interpretation in IIoT scenarios.

Table 1. ALO parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Population size (Antlions)	N_{AL}	20	Number of antlions (candidate solutions)
Population size (Ants)	N_A	20	Number of ants performing random walks
Maximum iterations	T_{max}	350	Termination criterion ensuring convergence
Search space bounds	—	$[0, W] \times [0, H]$	Deployment region boundaries
Random walk scaling factor	$I(t)$	Adaptive	Linearly reduced over iterations
Elitism	—	Enabled	Best antlion preserved to guide search
Solution encoding	—	Continuous	Device positions encoded as coordinates
Constraint handling	—	Repair-based	Infeasible positions projected to nearest feasible point
Fitness function	—	Composite	Coverage maximisation with overlap penalties (Section 2-1)
Initialization method	—	Uniform random	Initial devices are distributed uniformly in the feasible space
Boundary handling	—	Reflection + repair	Out-of-bound positions are reflected and then repaired
Obstacle handling	—	Exclusion + repair	Positions inside obstacles are relocated
Random seed	—	Fixed	Ensures reproducibility of results
Parameter tuning	—	Default (no tuning)	Default ALO parameters used for fair algorithm comparison

2.2.2 Artificial Bee Colony (ABC)

ABC, introduced by Karaboga [27], is a population-based metaheuristic inspired by the foraging behaviour of honey bees. In our formulation, each food source represents a deployment configuration encoded as a continuous vector of device coordinates (x_i, y_i) , corresponding to the spatial placement of devices. The initial population is generated using uniform random sampling over the feasible deployment space, excluding obstacle regions. During optimization, employed and onlooker bees perform local neighbourhood searches by perturbing device positions to improve coverage while reducing the device as defined by the composite fitness function in Section 2.1. Constraint handling is implemented using a repair-based strategy, whereby infeasible device positions are projected to the nearest feasible location rather than penalised, ensuring valid solutions throughout the search. Scout bees introduce diversification by replacing stagnant solutions after a predefined limit, helping the algorithm escape local optima. Algorithm-specific parameters are offered in Table 2. While ABC provides an effective balance between exploration and exploitation and is well-suited for continuous deployment optimization, its convergence speed and solution quality remain sensitive to colony size and scout activation frequency in large-scale environments.

2.2.3 Particle Swarm Optimization (PSO)

PSO, introduced by Kennedy and Eberhart [28], is a population-based metaheuristic inspired by the collective behaviour of bird flocks and fish schools. In our formulation, each particle encodes a complete deployment configuration as a continuous vector of device coordinates (x_i, y_i) , representing spatial placements within the monitoring region. The swarm is initialised using uniform random sampling over the feasible deployment space. Particle velocities and positions are updated using inertia, cognitive, and social components to promote both exploration and exploitation, with fitness evaluated (defined in Section 2.1), which favours coverage maximisation while penalising device–device overlap, device–obstacle overlap, and boundary leakage. Constraint handling is implemented using a repair-based strategy, whereby positions outside boundaries or inside obstacles are projected to the nearest feasible location rather than penalised. Boundary violations are handled using velocity clamping followed by position repair to maintain swarm stability. Algorithm-specific parameters are brought in Table 2. While PSO exhibits fast convergence and low implementation complexity, its performance is sensitive to parameter settings and suffers from premature convergence.

Table 2. ABC parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Colony size (food sources)	SN	20	Number of candidate deployment solutions
Employed bees	—	20	One employed bee per food source
Onlooker bees	—	20	Equal to the number of employed bees
Scout bees	—	Adaptive	Activated when the food source stagnates
Limit parameter	$limit$	50	Maximum trials before abandonment
Maximum iterations	T_{max}	350	Termination criterion
Search space bounds	—	$[0, W] \times [0, H]$	Deployment region boundaries
Solution encoding	—	Continuous	Device positions
Initialization method	—	Uniform random	Feasible region excluding obstacles
Fitness function	—	Composite	Coverage maximisation + overlap penalties
Constraint handling	—	Repair-based	Infeasible positions projected to the nearest feasible point
Boundary handling	—	Reflection + repair	Prevents boundary violations
Obstacle handling	—	Exclusion + repair	Ensures no device inside obstacles
Random seed	—	Fixed	Guarantees reproducibility
Parameter tuning	—	Default (no tuning)	Original ABC settings used for fairness

Table 3. PSO parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Swarm size	Np	20	Number of particles (deployment candidates)
Maximum iterations	T_{max}	350	Termination criterion
Inertia weight	w	Linearly decreasing (0.9-0.4)	Balances exploration and exploitation
Cognitive coefficient	c_1	2.0	Attraction toward personal best
Social coefficient	c_1	2.0	Attraction toward the global best
Velocity limits	v_{max}	$\pm 0.2 \times \max(W, H)$	Prevents excessive particle movement
Search space bounds	—	$[0, W] \times [0, H]$	Deployment region boundaries
Solution encoding	—	Continuous	Device positions
Initialization method	—	Uniform random	Feasible space excluding obstacles
Fitness function	—	Composite	Coverage maximisation + overlap penalties
Constraint handling	—	Repair-based	Infeasible positions projected to the nearest feasible point
Boundary handling	—	Velocity clamping + repair	Maintains feasibility and stability
Obstacle handling	—	Exclusion + repair	Prevents devices from entering obstacles
Random seed	—	Fixed	Ensures reproducibility
Parameter tuning	—	Default (no tuning)	Standard PSO parameters used for fairness

2.2.4 Genetic Algorithm (GA)

GA is a population-based evolutionary method inspired by natural selection and genetic inheritance [29]. In our formulation, each chromosome encodes a deployment configuration as a continuous vector of device coordinates (x_i, y_i) , representing spatial placements within the monitoring area. The initial population is generated using uniform random sampling over the deployment space while excluding obstacles. Fitness evaluation is performed using the composite objective defined in Section 2.1. Parent selection is implemented using roulette-wheel selection, while two-point crossover is applied to exchange spatial information between parent solutions and generate offspring. Mutation is introduced with a low probability to perturb device coordinates and maintain population diversity. Constraint handling is performed using a repair-based strategy, whereby infeasible device positions are projected to the nearest feasible location, ensuring valid deployment. Boundary violations are handled via coordinate clamping followed by repair. Algorithm-specific parameters are presented in Table 4. While GA provides strong global search and robustness to multimodal search spaces, its convergence speed and computational cost are sensitive to population size and genetic operator settings for large-scale problems.

2.2.5 Water Cycle Algorithm (WCA)

WCA is a nature-inspired metaheuristic that models the hydrological cycle of streams, rivers, and seas [30]. In our formulation, each raindrop represents a complete deployment configuration encoded as a continuous vector of device coordinates (x_i, y_i) . The initial population is generated using uniform random sampling over the feasible deployment region. Based on fitness evaluation, the best solution is designated as the *sea*, a subset of high-quality solutions as *rivers*, and the remaining solutions as *streams*. Streams and rivers move toward better solutions by updating their positions in the direction of the sea, enabling exploitation, while an evaporation-precipitation mechanism introduces diversification by reinitialising stagnant solutions. Fitness evaluation follows the composite objective defined in Section 2.1. Constraint handling is implemented using a repair-based strategy, whereby any position outside boundaries or inside obstacles is projected to the nearest feasible location.

Boundary violations are handled via coordinate clamping followed by repair. Algorithm-specific parameters are outlined in Table 5. While WCA provides strong global search capability and resistance to premature convergence, its computational cost and convergence behaviour are sensitive to population structure and evaporation settings.

Table 4. GA parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Population size	N_{pop}	20	Number of chromosomes (deployment candidates)
Maximum generations	G_{max}	350	Termination criterion
Chromosome encoding	—	Continuous	Device positions
Selection method	—	Roulette-wheel	Fitness-proportional parent selection
Crossover type	—	Two-point	Preserves spatial structure of solutions
Crossover probability	P_c	0.8	Standard GA setting for exploration
Mutation type	—	Uniform	Random perturbation of coordinates
Mutation probability	P_m	0.05	Maintains diversity without instability
Search space bounds	—	$[0, W] \times [0, H]$	Monitoring region boundaries
Initialization method	—	Uniform random	Feasible space excluding obstacles
Fitness function	—	Composite	Coverage maximisation + overlap penalties
Constraint handling	—	Repair-based	Infeasible positions projected to feasible region
Boundary handling	—	Clamping + repair	Prevents boundary violations
Obstacle handling	—	Exclusion + repair	Ensures no device inside obstacles
Elitism	—	Enabled (1 elite)	Preserves best solution
Random seed	—	Fixed	Ensures reproducibility
Parameter tuning	—	Default (no tuning)	Standard GA parameters used for fairness

Table 5. WCA parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Population size (raindrops)	N	20	Number of candidate deployment solutions
Number of rivers	N_r	3	High-quality solutions guiding search
Number of streams	N_s	$(N - N_{r-1})$	Remaining solutions flow toward rivers
Sea	—	1	Best solution in the population
Maximum iterations	T_{max}	350	Termination criterion
Evaporation threshold	d_{max}	1×10^{-5}	Controls reinitialisation of stagnant solutions
Search space bounds	—	$[0, W] \times [0, H]$	Deployment region boundaries
Solution encoding	—	Continuous	Device positions
Initialization method	—	Uniform random	Feasible space excluding obstacles
Fitness function	—	Composite	Coverage maximisation + overlap penalties
Constraint handling	—	Repair-based	Infeasible positions projected to the feasible region
Boundary handling	—	Clamping + repair	Prevents boundary violations
Obstacle handling	—	Exclusion + repair	Ensures no device inside obstacles
Random seed	—	Fixed	Ensures reproducibility
Parameter tuning	—	Default (no tuning)	Original WCA parameters used for fairness

2.2.6 Bat Algorithm (BA)

BA, introduced by Yang [31], is a nature-inspired metaheuristic based on the echolocation behaviour of bats. In our formulation, each bat represents a deployment configuration encoded as a continuous vector of device coordinates (x_i, y_i) , corresponding to spatial placements within the monitoring region. The initial population is generated using uniform random sampling over the feasible deployment space. During optimization, bat positions are updated using frequency-controlled velocity adjustments to balance global exploration and local exploitation, while a local random walk is applied around the current best solution to refine promising deployments. Fitness evaluation follows the composite objective defined in Section 2.1. Constraint handling is implemented using a repair-based strategy, whereby any infeasible device position is projected to the nearest feasible location, ensuring valid solutions. Boundary violations are handled through coordinate clamping followed by repair. Algorithm-specific parameters are described in Table 6. While BA provides a flexible balance between exploration and exploitation, its convergence behaviour and solution quality are sensitive to parameter settings in complex IIoT scenarios with non-penetrable obstacles.

Table 6. BA parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Population size (bats)	N_b	20	Number of candidate deployment solutions
Maximum iterations	T_{max}	350	Termination criterion
Initial loudness	A_0	1.0	Controls exploration at early stages
Minimum loudness	A_{min}	0.1	Limits the exploitation phase
Pulse emission rate	r_0	0.5	Initial local search probability
Loudness reduction factor	α	0.9	Gradual transition from exploration to exploitation
Pulse rate increase factor	γ	0.9	Increases local search over iterations
Frequency minimum	f_{min}	0	Lower bound of frequency
Frequency maximum	f_{max}	2	Upper bound controlling step size
Velocity limits	v_{max}	$\pm 0.2 \times \max(W, H)$	Prevents excessive movement
Search space bounds	—	$[0, W] \times [0, H]$	Monitoring region boundaries
Solution encoding	—	Continuous	Device positions
Initialization method	—	Uniform random	Feasible space excluding obstacles
Fitness function	—	Composite	Coverage maximisation + overlap penalties
Constraint handling	—	Repair-based	Infeasible positions projected to the feasible region
Boundary handling	—	Clamping + repair	Ensures valid deployment positions
Obstacle handling	—	Exclusion + repair	Prevents devices from entering obstacles
Random seed	—	Fixed	Ensures reproducibility
Parameter tuning	—	Default (no tuning)	Standard BA parameters used for fairness

Table 7. PSOLC parameters for IIoT device deployment.

Parameter	Symbol	Value	Description / Rationale
Learning probability	P_{learn}	0.3	Probability of learning from top-performing particles
Elite set size	K	5	Number of top particles used for learning
Crossover type	—	Two-point	Exchanges coordinate segments
Crossover probability	P_c	0.6	Controls diversity introduction

2.2.7 Particle Swarm Optimization with a Learning strategy and Crossover Operator (PSOLC)

PSOLC is an enhanced PSO that integrates population learning and genetic crossover mechanisms to improve exploration and exploitation [23]. In our formulation, each particle encodes a complete deployment configuration as a continuous vector of device coordinates (x_i, y_i) , representing spatial placements within the monitoring area. The initial swarm is generated using uniform random sampling over the deployment space while excluding obstacle regions, and initial velocities are assigned randomly within predefined limits. Fitness evaluation follows the composite objective defined in Section 2.1. In addition to the standard PSO velocity update, PSOLC employs a learning strategy in which particles probabilistically learn from a subset of top-performing particles rather than only the global best, thereby enhancing population diversity. A crossover operator is applied to selected particle pairs to exchange segments of position vectors and generate new candidate deployments. Constraint handling is implemented using a repair-based strategy, whereby positions outside boundaries or inside obstacles are projected to the nearest feasible location, ensuring valid deployments throughout optimization. Boundary violations are handled via velocity clamping followed by coordinate repair. The additional parameters introduced by PSOLC, beyond the standard PSO, are presented in Table 7. While PSOLC improves convergence speed and robustness against premature convergence compared to standard PSO, its additional learning and crossover operations increase computational cost and parameter sensitivity.

2.2.8 Comparative Overview

Table 8 presents a comparative analysis of the evaluated algorithms across key performance criteria relevant to IIoT device deployment. While all methods exhibit distinct strengths, ALO emerges as a strong approach, combining effective exploration with the ability to identify near-optimal solutions in obstacle-constrained environments. Unlike PSO, which converges rapidly but may stagnate in local optima if not carefully tuned, ALO maintains diversity in the search process and demonstrates robust global search capability, albeit with some sensitivity to parameter settings. GA offers strong exploration and high solution quality in diverse search spaces, but its convergence is slower, and the computational cost is higher due to genetic operators. ABC provides a reasonable balance but tends to prioritise exploitation, which can limit its effectiveness in complex regions. WCA and BA also exhibit balanced behaviour, with WCA showing good adaptability and scalability in large-scale deployments and BA offering flexible parameter control, although both require careful configuration to avoid premature convergence. PSOLC improves upon PSO by incorporating learning and crossover mechanisms, achieving faster convergence and high solution quality; however, this comes at the expense of increased algorithmic complexity. Overall, ALO stands out as a competitive and reliable solution for IIoT device deployment, offering strong exploration, high coverage performance, and favourable energy efficiency without the additional structural complexity introduced by hybrid methods.

Table 8. Comparative overview of the algorithms.

Criteria	PSO	GA	ABC	ALO	WCA	BA	PSOLC
Exploration vs. Exploitation	Good balance, but gets stuck in local optima	Strong in exploration, needs tuning for exploitation	Good balance, but exploitation dominates	Strong exploration, but requires parameter tuning	Good balance with adaptable exploration	Balanced with adjustable parameters	Enhanced balance with learning and crossover
Convergence Speed	Fast but stagnant	Slower	Moderate speed	Moderate	Fast	Fast, but needs tuning	Fast with better global exploration
Solution Quality	High, but dependent on initial settings	High in diverse search spaces	High, with a tendency towards good exploitation	High, with the potential to find near-optimal solutions	High with the correct water cycle strategy	High with strong exploration features	Very high due to additional learning and crossover
Scalability	Good, but performance drops with size	Good, but computational cost increases with size	Good, but slows down with large swarms	Good, handles complex, large-scale problems	Very scalable	Good, but requires parameter tuning	Highly scalable
Complexity	Moderate, easy to implement	High	Moderate complexity, simple to implement	Moderate complexity, requires parameter tuning	Moderate	Moderate and simple structure, but needs tuning	High complexity
Energy Efficiency	Moderate	High	Moderate to high	High	High	High	Very high

3. EXPERIMENTAL ANALYSIS AND RESULTS

This section presents a simulation-based comparative analysis of the considered metaheuristic algorithms for IIoT device deployment, focusing on two performance metrics: coverage and Number of Function Evaluations (NFE). All experiments are conducted in a synthetic monitoring environment implemented in MATLAB R2018a, designed to emulate the spatial characteristics of an industrial layout rather than representing a physical factory. The monitored area is a fixed 100×100 m region containing rectangular, non-penetrable obstacles arranged in two configurations, as illustrated in Figure 3. The first configuration (Figure 3(a)) includes five obstacles covering 14%, while the second configuration (Figure 3(b)) consists of four obstacles covering 44% of the area. These configurations are selected to represent low- and high-obstacle-density scenarios, and all results and conclusions in this study are limited to these tested cases.

In both configurations, heterogeneous IIoT devices with sensing radii of 5, 6, 8, and 10 m are deployed, with proportions of 25%, 20%, 15%, and 10%, respectively, based on the estimated MND in Equation (1). The remaining devices are allocated to ensure full coverage of the deployable area. For all algorithms, the population size and maximum number of iterations are fixed at 20 and 350, respectively, to ensure a fair comparison. Each algorithm is executed 20 independent runs, and the reported results correspond to the average performance, which reduces the influence of stochastic variability inherent in metaheuristic optimisation. While no formal statistical significance tests are reported, all algorithms are evaluated under identical conditions, and performance comparisons are therefore interpreted relatively, consistent with exploratory benchmarking studies. The main simulation parameters are summarised in Table 9.

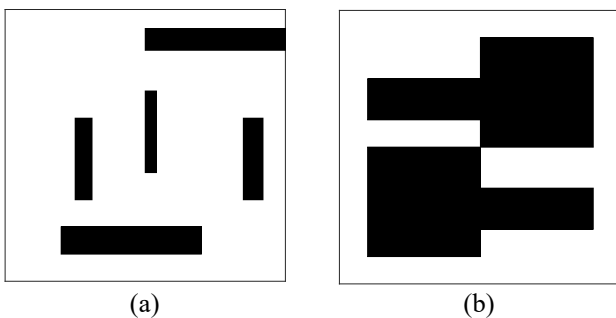


Figure 3. Obstacle distribution in the monitoring environment: (a) Five obstacles covering 14%; (b) Four obstacles covering 40% of the area.

Table 9. Simulation parameters and values.

Notations	Description	Values	
H	Height of the monitoring area	100 m	
W	Width of the monitoring area	100 m	
C_i	Sensing radius of IIoT devices	5, 6, 8, 10 m	
M	Number of obstacles in the environment	4 and 5	
MND	Estimated minimum number of devices required for full coverage	Conf 1	78
		Conf 2	52

Conf: Configuration

3.1 Number of Function Evaluations (NFE)

Figure 4 illustrates the convergence behaviour of the evaluated algorithms in terms of fitness versus NFE for the first configuration with 14% obstacle coverage, where lower fitness values correspond to better IIoT device deployment solutions. Overall, ALO and PSO demonstrate the most effective performance in this scenario. ALO begins with a fitness value of approximately 0.2 and shows a consistent reduction, reaching a stable value close to 0.1 after around 1500 NFEs, which indicates a good balance between exploration and exploitation and fast convergence. PSO exhibits strong optimisation capability, with a rapid decrease in fitness within the first 500 NFEs and stabilisation around 0.1 - 0.11, highlighting its efficiency in early-stage search. ABC shows a noticeable improvement during the initial 1000 NFEs and converges to a fitness value of about 0.13 after approximately 2500 NFEs, demonstrating stable but slower convergence compared to ALO and PSO. GA and WCA both display more gradual convergence trends; GA stabilises around 0.13 - 0.14 after roughly 4000 NFEs, while WCA reaches a similar fitness level after a comparable number of evaluations, indicating moderate efficiency. In contrast, BA starts with a high fitness value near 0.7 and, despite a rapid initial decrease, stagnates at around 0.22, showing limited optimisation capability. PSOLC follows a similar initial pattern to BA, rapidly reducing fitness early on but stabilising at approximately 0.2, outperforming BA but remaining less effective than ALO, PSO, and ABC. Overall, the results indicate that ALO and PSO achieve the fastest convergence and lowest fitness values, making them the most suitable algorithms for IIoT device deployment in environments with low obstacle density, while BA and PSOLC exhibit weaker performance in this scenario.

Figure 5 also shows the fitness–NFE curve for the second configuration with 44% obstacle coverage. Among all algorithms, ALO demonstrates the strongest performance, starting from a fitness value of approximately 0.2 and achieving a rapid reduction during the first 1000 NFEs. It converges to a stable fitness value slightly below 0.1 by around 1500 NFEs and maintains this level throughout the remaining evaluations, indicating both fast convergence and superior solution quality in the constrained environments. PSO also performs well, beginning near 0.3 and showing a sharp fitness reduction within the first 500 NFEs before stabilising around 0.1, making it the second-best algorithm in terms of convergence speed and final fitness. ABC exhibits reasonable performance, with notable improvement in the early iterations but slower convergence and a higher final fitness compared to ALO and PSO. GA and WCA display more gradual convergence trends, stabilising at fitness values around 0.14 after approximately 3000 NFEs, which reflects moderate optimisation capability but lower efficiency. In contrast, BA starts with a high fitness value close to 0.7 and, despite a rapid initial decrease, stagnates around 0.22, showing limited optimisation potential. PSOLC follows a similar pattern to BA, rapidly reducing fitness early on but stabilising at a relatively high level, indicating suboptimal convergence behaviour. Overall, the results confirm that ALO is the most effective algorithm for device deployment under high obstacle density, achieving the fastest convergence and lowest final fitness values, while BA and PSOLC exhibit the weakest performance in this scenario.

3.2 Coverage

Figure 6 compares the coverage performance of the algorithms in the first configuration with 14% obstacle coverage, where coverage represents the percentage of the monitoring area covered by devices. The results indicate that WCA achieves the highest coverage at 93%, demonstrating strong capability in arranging devices to maximise spatial coverage in low obstacle-density environments. PSO follows with about 92% coverage, confirming its robustness in obstacle-constrained deployment scenarios. ALO ranks third with a coverage of 91.57%, which is marginally lower than WCA and PSO, and still reflects a high level of efficiency in covering the monitoring region. In contrast, PSOLC and GA exhibit moderate performance, achieving coverage levels of around 77.23% and 76.44%, respectively, which are lower than the leading algorithms and indicate reduced effectiveness under this configuration. BA and ABC perform the weakest, both achieving coverage values just above 73%, suggesting difficulties in device placement in the presence of obstacles. Overall, WCA, PSO, and ALO are the most reliable algorithms for maximising coverage in IIoT device deployment under low obstacle density.

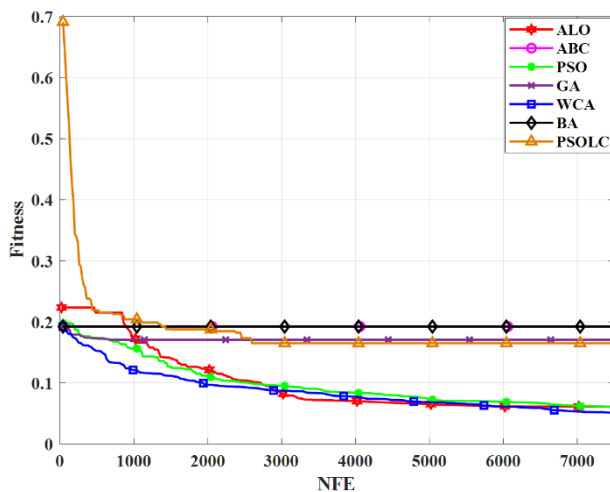


Figure 4. NFE of algorithms in the first configuration (14% obstacle coverage).

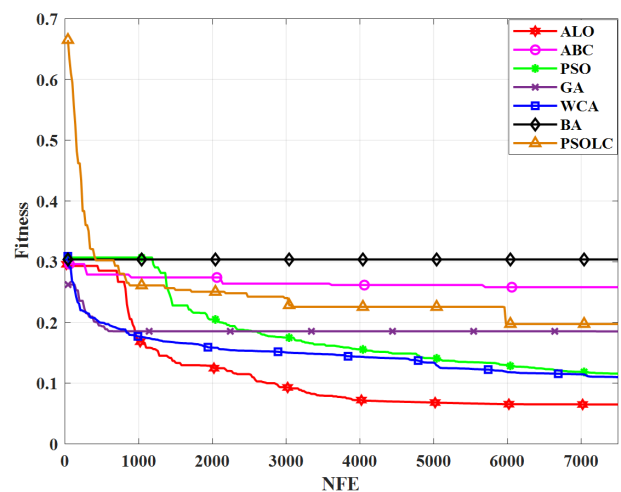


Figure 5. NFE of algorithms in the second configuration (44% obstacle coverage).

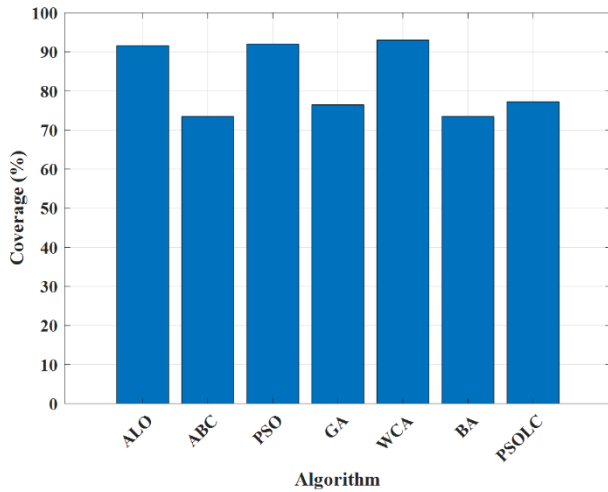


Figure 6. Coverage of algorithms in the first configuration (14% obstacle coverage).

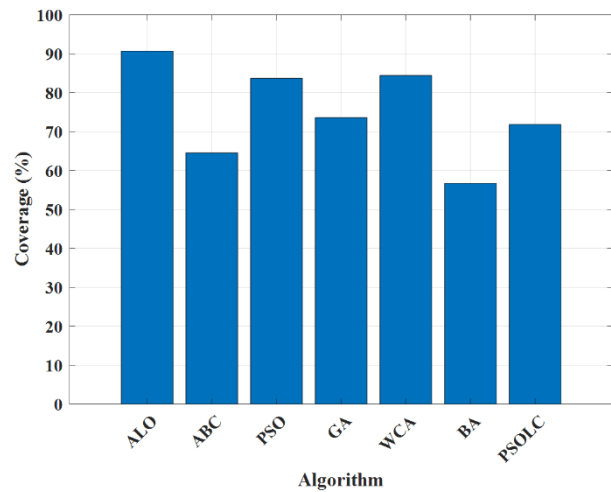


Figure 7. Coverage of algorithms in the second configuration (44% obstacle coverage).

Figure 7 also presents the coverage of the algorithms in the second configuration with 44% obstacle coverage, representing a constrained and challenging environment. The results show that ALO achieves the highest coverage at 90.73%, confirming its superior ability to handle dense obstacle conditions and to maintain near-complete monitoring coverage. This strong performance highlights its robustness and global search capability in complex environments. WCA ranks second with a coverage of 84.45%, followed by PSO at 83.71%, indicating that both algorithms remain competitive under high obstacle density, although they fall short of the performance of ALO. GA achieves a moderate coverage of 73.6%, reflecting reduced effectiveness in obstructed scenarios, while PSOLC performs lower at 71.87%, suggesting that the learning and crossover mechanisms do not compensate for the increased environmental complexity. ABC exhibits a further decline in performance, reaching only 64.56% coverage, and BA performs the weakest with 56.75%, indicating significant difficulty in device placement under dense obstacle conditions. Overall, ALO outperforms other algorithms in the 44% configuration, making it the most reliable approach for IIoT device deployment in obstacle-constrained environments, with WCA and PSO serving as secondary alternatives and the remaining algorithms showing limited suitability in such challenging scenarios.

4. CONCLUSION

This paper provides a comparative evaluation of several metaheuristic algorithms, including ALO, PSO, WCA, GA, PSOLC, ABC, and BA for IIoT device deployment in obstacle-constrained environments. The results demonstrate that ALO achieves the best overall performance, with an average coverage of 91.13%, fast convergence, and robust behaviour across both low (14%) and high (44%) obstacle-density configurations. PSO and WCA also perform strongly, delivering high coverage and efficient convergence, particularly in less complex scenarios. In contrast, GA and PSOLC show moderate effectiveness, while ABC and BA struggle to achieve competitive coverage as obstacle density increases. It is important to emphasise that all findings are derived from controlled MATLAB-based simulations, and conclusions are therefore strictly limited to the tested configurations rather than real-world deployments.

Further analysis indicates that algorithm performance is influenced by failure modes and sensitivities. For example, PSO and PSOLC tend to suffer from premature convergence and diversity collapse in constrained environments, while GA and WCA exhibit slower convergence due to population diversity and parameter dependence. ABC and BA are sensitive to penalty dominance and boundary effects, which limit their effectiveness under dense obstacle conditions. The superior performance of ALO is attributed to its strong global exploration mechanism and resilience to local optima, although it remains sensitive to parameter tuning in extreme cases. Future research should therefore investigate systematic parameter sensitivity analysis, adaptive parameter control, and hybridisation strategies to mitigate these limitations. In addition, extending the framework to large-scale scenarios, as well as integrating with learning-based or data-driven models, could further enhance adaptability and long-term performance in practical IIoT systems.

ACKNOWLEDGEMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

- [1] Y. Hu, Q. Jia, Y. Yao, Y. Lee, M. Lee, C. Wang, X. Zhou, R. Xie and F. R. Yu, Industrial internet of things intelligence empowering smart manufacturing: A literature review, *IEEE Internet of Things Journal*, 11, 2024, 19143-19167.

- [2] E. Dritsas and M. Trigka, A survey on the applications of cloud computing in the industrial internet of things, *Big Data and Cognitive Computing*, 9, 2025, 44.
- [3] S. Yousefi, H. Karimipour and F. Derakhshan, Data aggregation mechanisms on the internet of things: a systematic literature review, *Internet of Things*, 15, 2021, 100427.
- [4] S. Khan, P. A. Ferreira Lopes Martins, B. Sousa and V. Pereira, A comprehensive review on lightweight cryptographic mechanisms for industrial internet of things systems, *ACM Computing Surveys*, 58, 2026, 1-37.
- [5] S. Yousefi, F. Derakhshan and H. Karimipour, Applications of big data analytics and machine learning in the internet of things, *Handbook of Big Data Privacy*, Cham: Springer International Publishing, 2020, 77-108.
- [6] B. X. Chai, M. Gunaratne, M. Ravandi, J. Wang, T. Dharmawickrema, A. Di Pietro, J. Jin and D. Georgakopoulos, Smart industrial internet of things framework for composites manufacturing, *Sensors*, 24, 2024, 4852.
- [7] X. Fernando and G. Lăzăroiu, Energy-efficient industrial internet of things in green 6G networks, *Applied Sciences*, 14, 2024, 8558.
- [8] M. Adil, M. K. Khan, N. Kumar, M. Attique, A. Farouk, M. Guizani and Z. Jin, Healthcare internet of things: Security threats, challenges, and future research directions, *IEEE Internet of Things Journal*, 11, 2024, 19046-19069.
- [9] M. Soori, F. K. G. Jough, R. Dastres and B. Arezoo, Blockchains for Industrial internet of things in sustainable supply chain management of industry 4.0: A review, *Sustainable Manufacturing and Service Economics*, 3, 2024, 100026.
- [10] S. Najjar-Ghabel, S. Yousefi and H. Karimipour, Blockchain applications in the industrial internet of things, *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*, Cham: Springer International Publishing, 2021, 41-76.
- [11] V. Demertzi, S. Demertzis and K. Demertzis, An overview of privacy dimensions on the industrial internet of things (IIoT), *Algorithms*, 16, 2023, 378.
- [12] S. Yousefi, F. Derakhshan, H. S. Aghdasi and H. Karimipour, An energy-efficient artificial bee colony-based clustering in the internet of things, *Computers & Electrical Engineering*, 86, 2020, 106733.
- [13] A. Adel, The convergence of intelligent tutoring, robotics, and IoT in smart education for the transition from industry 4.0 to 5.0, *Smart Cities*, 7, 2024, 325-369.
- [14] S. Yousefi, F. Derakhshan, H. Karimipour and H. S. Aghdasi, An Efficient route planning model for mobile agents on the internet of things using Markov decision process, *Ad Hoc Networks*, 98, 2020, 102053.
- [15] C. Qiu, H. Shen and K. Chen, An energy-efficient and distributed cooperation mechanism for coverage hole detection and healing in wireless sensor networks, *IEEE Transactions on Mobile Computing*, 17, 2018, 1247-1259.
- [16] C. -F. Cheng, Y. -C. Dong, M. -H. Hsu and H. -Y. Chang, The carrier-based sensor deployment problem in industrial internet of things with mesh topologies, *IEEE Internet of Things Journal*, 11, 2024, 41137-41150.
- [17] F. Medina, H. Ruiz, J. Espíndola and E. Avendaño, Deploying industrial internet of things systems for long-term planning in underground mining: A focus on the monitoring of explosive atmospheres, *Applied Sciences*, 14, 2024, 1116.
- [18] A. Chai, L. Wang, C. Guo, M. Li, W. Yin and Z. Fang, AS-MPCA: An adaptive scheduling algorithm for industrial internet of things based on multi-population co-evolution, *Internet of Things*, 31, 2025, 101596.
- [19] S. Ghorpade, M. Zennaro and B. Chaudhari, Survey of localization for internet of things nodes: Approaches, challenges and open issues, *Future Internet*, 13, 2021, 210.
- [20] M. Sadeghi Ghahroudi, A. Shahrabi, S. M. Ghoreyshi and F. A. Alfouzan, Distributed node deployment algorithms in mobile wireless sensor networks: survey and challenges, *ACM Transactions on Sensor Networks*, 19, 2023, 1-26.
- [21] M. Almasabi, A. B. Alkhodre, M. Khemakhem, F. Eassa, A. A. Abi Sen and A. Harbaoui, Internet of things-based anomaly detection hybrid framework simulation integration of deep learning and blockchain, *Information*, 16, 2025, 406.
- [22] M. Adawy, O. Almomani, M. Tahboush and A. Althunibat, Integrated boolean sensing and event radius model on data aggregation in wireless sensor networks, *Proceedings of the 2023 International Conference on Information Technology*, Amman, Jordan, 2023, 613-619.
- [23] S. Molaei, H. Moazen, S. Najjar-Ghabel and L. Farzinvash, Particle swarm optimization with an enhanced learning strategy and crossover operator, *Knowledge-Based Systems*, 215, 2021, 106768.
- [24] T. Kaur, J. Singh and M. Singh, Optimization-driven localization in wireless sensor networks: A comprehensive review of single and hybrid metaheuristic approaches, *International Journal of Communication Systems*, 38, 2025.
- [25] F. Kiani, A multi-objective metaheuristic method for node placement in dynamic internet of things environments, *Discover Internet of Things*, 5, 2025, 60.
- [26] S. Mirjalili, The ant lion optimizer, *Advances in Engineering Software*, 83, 2015, 80-98.
- [27] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Ph.D. Dissertation, Erciyes University, Kayseri, Turkey, 2005.
- [28] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of the International Conference on Neural Networks*, Perth, Australia, 1995, 1942-1948.
- [29] S. Mirjalili, Genetic algorithm, *Evolutionary Algorithms and Neural Networks*, Cham: Springer, 2019, 43-55.
- [30] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, 110-111, 2012, 151-166.
- [31] X. -S. Yang, A new metaheuristic bat-inspired algorithm, *Nature-Inspired Metaheuristic Algorithms*, Springer, 2010, 65-74.