Tran Khanh Dang
Josef Küng
Tai M. Chung (Eds.)

Communications in Computer and Information Science 2708

Future Data and Security Engineering

12th International Conference, FDSE 2025
Ho Chi Minh City, Vietnam, November 27–29, 2025
Proceedings, Part I

Part 1

Springer

FDSE 2025

Improving RFID Network Planning Efficiency with Genetic Algorithm - Hopfield Network Integrated Model

Nguyen Van Tung[1,2](✉), Nguyen Le Kim Thanh[2], Le Van Hoa[3], and Vo Viet Minh Nhat[4]

[1] University of Sciences, Hue University, Hue City, Vietnam
nvtung@hueuni.edu.vn, tungnv@huit.edu.vn
[2] Ho Chi Minh City University of Industry and Trade, Ho Chi Minh City, Vietnam
[3] School of Hospitality and Tourism, Hue University, Hue City, Vietnam
[4] Department of Academic and Students' Affairs, Hue University, Hue City, Vietnam

Abstract. RFID network planning (RNP) is the problem of allocating readers in a work area so that the constructed reader network can cover all tags, while satisfying some constraints such as minimum interference, minimum number of used readers, etc. As a multi-objective optimization problem, RNP is proven to be NP-hard, so natural-based methods are often used to find the best solution in a given period. Approaches based on GA, PSO, Cuckoo Search, and even artificial neural networks have also been proposed to solve this RNP problem. However, a problem for the natural-based methods class is choosing the best hyperparameters. This paper proposes an integration of the genetic algorithm and the Hopfield network, in which the RNP problem is optimized by the Hopfield network based on the hyperparameters determined by the genetic algorithm. Experimental results show that the integrated model achieves impressive results with a given working area in which the tags are evenly distributed. However, in return, the execution time increases significantly due to searching and determining the optimal hyperparameter set.

Keywords: RFID network planning · hyperparameter · optimization · genetic algorithm · Hopfield network

1 Introduction

Radio Frequency Identification (RFID) is a wireless technology essential for IoT applications that uses electromagnetic fields to identify and track tagged objects. RFID systems contain readers that query tags and connect to central servers for data processing. Tags are either active (self-powered) or passive (reader-powered), with passive tags being more common due to lower cost.

RFID Network Planning (RNP) strategically positions readers to maximize coverage while minimizing reader count, interference, and energy usage. This NP-hard optimization problem resembles the Set Covering Problem and requires heuristic methods to find near-optimal solutions.

T. K. Dang et al. (Eds.): FDSE 2025, CCIS 2708, pp. 194–207, 2026.
https://doi.org/10.1007/978-981-95-4721-0_13

This paper presents a novel model combining Genetic Algorithm (GA) with Hopfield Network to optimize RFID reader placement and hyperparameter selection simultaneously. Initial results show improved efficiency in reader distribution despite increased execution time.

Key contributions are:

1. Analysis of Hopfield network optimization with focus on hyperparameter impacts
2. Introduction of the GA-Hopfield model for combined reader placement and hyperparameter optimization
3. Simulation demonstrating the integrated model's effectiveness

The paper structure covers related research **(Sect. 2)**, GA-Hopfield integration **(Sect. 3)**, simulation results **(Sect. 4)**, and conclusions**(Sect. 5)**

2 Related Works

RFID network planning is a critical aspect of deploying RFID applications effectively. This process presents considerable challenges, as it must address a variety of objectives and constraints simultaneously. Optimizing RFID network planning has been identified as an NP-hard problem [1], prompting researchers to explore nature-inspired metaheuristic methods as potential solutions. These methods can be broadly classified into three main categories: swarm intelligence algorithms, evolutionary algorithms, and various hybrid models.

2.1 Swarm Intelligence Approaches

Swarm intelligence algorithms have been widely applied to Reader Node Placement (RNP) problems. Particle Swarm Optimization (PSO) has shown particular promise, with Bhattacharya and Roy [3] demonstrating its effectiveness in optimizing reader placement. Cao, Liu, and Xu [9] enhanced PSO by developing HPSO-RNP, incorporating k-Means clustering and virtual force mechanisms to improve reader quantity, interference management, and energy balance.

Other successful swarm-based approaches include the hierarchical Artificial Bee Colony (ABC) algorithm by Ma et al. [4], the Firefly Algorithm (FA) applied by Bacanin, Tuba, and Jovanovic [5] for multi-objective RNP challenges, and Ant Colony Optimization (ACO) by Zhang and Geng [16], where ants iteratively select optimal reader locations.

2.2 Evolutionary and Other Metaheuristic Approaches

Evolutionary algorithms, particularly the Genetic Algorithm (GA), have a long history in solving RNP. The work by Suriya and David Porter [2] provides a foundational GA-based approach for optimizing network planning. Another powerful, though less common, metaheuristic is Simulated Annealing (SA), which is adept at escaping local optima. While often used in hybrid models, standalone SA algorithms have also been developed for RNP, leveraging probabilistic acceptance criteria to explore the solution space broadly

[17]. The Cuckoo Search (CS) algorithm, proposed by Jaballah and Meddeb [6], offers a self-adaptive mechanism to tune its parameters during the search, showing promise in adapting to the problem landscape.

2.3 Hybrid and Novel Approaches

No single optimization algorithm can claim universal superiority, driving researchers toward hybrid models that combine multiple techniques' strengths. Feng and Qi [8] proposed a multi-population GA-PSO approach that divides populations and uses genetic operators to adjust PSO rules for enhanced large-scale system performance dynamically. Similarly, Aghdam et al. [10] developed a two-stage GA-SA model using GA for broad initial searches followed by Simulated Annealing for fine-tuning.

Hybrid optimization has evolved to integrate metaheuristics with machine learning techniques. Maimouni, Majd, and Bouya [12] introduced RAE-NNA, combining Redundant Reader Elimination with Neural Network Algorithms to minimize solution gaps. Shi et al. [7] applied improved Chicken Swarm Optimization for phased array antenna optimization. Yixuan et al. [11] developed GWO-CS, integrating Cuckoo Search with Grey Wolf Optimizer to enhance global search coverage and local search capabilities.

Despite these advances, a significant challenge remains: manual hyperparameter tuning. The efficacy of algorithms like GA and PSO heavily depends on these settings. This paper addresses this gap by proposing an automated model for optimizing hyperparameters in Hopfield Network-based optimizers, aiming to enhance reliability while reducing manual effort and subjectivity in hyperparameter tuning.

3 The Genetic Algorithm - Hopfield Network Integrated Model for Rnp Problem

3.1 Preliminaries

The Hopfield Network for Optimization

A Hopfield Network (HN) is a type of recurrent artificial neural network in which every neuron is interconnected. In a discrete Hopfield network, the neurons operate using binary states (e.g., $+1/-1$ or 1/0). The network begins from an initial state and iteratively updates the states of its neurons until achieving convergence to a stable state. This convergence is assured because the network's dynamics are designed to minimize a quadratic energy function, as defined in Eq. (5). Each stable state corresponds to a local minimum of this energy function, which grants the Hopfield network its effectiveness in addressing combinatorial optimization problems. By strategically mapping the objective function and constraints of an optimization problem onto the structure of the HN's energy function—through the careful definition of synaptic weights w_{ij} and neuron thresholds θ_i—the network's inherent tendency to converge to a low-energy state can be leveraged to identify low-cost, near-optimal solutions to the problem [14].A Hopfield Network (HN) is a form of recurrent artificial neural network in which all neurons are connected to each other. In a discrete Hopfield network, neurons are binary (e.g., $+1/-1$ or 1/0).

The network operates by starting from an initial state and iteratively updating the state of its neurons until it converges to a stable state. This convergence is guaranteed because the network dynamics are designed to minimize a quadratic energy function, defined in Eq. (5). Each stable state of the network corresponds to a local minimum of this energy function. This property makes the Hopfield network a powerful tool for solving combinatorial optimization problems. By carefully mapping the objective function and constraints of an optimization problem onto the structure of the HN's energy function (i.e., by defining the synaptic weights w_{ij} and neuron thresholds θ_i), the network's natural convergence to a low-energy state can be used to find a low-cost (and thus, near-optimal) solution to the problem [14].

The Genetic Algorithm

The Genetic Algorithm (GA) is a powerful metaheuristic technique inspired by the principles of natural selection. This approach is particularly effective for tackling complex optimization and search problems, which often arise in various fields such as engineering, economics, and artificial intelligence. The GA operates on a population of candidate solutions, commonly referred to as chromosomes or individuals. Each chromosome consists of a set of genes that represent the specific parameters of the solution being explored.

The algorithm's strength lies in its fitness function, a crucial component that assesses the quality of each chromosome based on defined criteria. Over successive generations, the GA evolves the population through three main operators: (1) ***Selection***, in which individuals with a higher likelihood of fitness are chosen to produce the next generation of offspring, effectively ensuring that stronger characteristics are passed on; (2) ***Crossover***, where segments from two parent chromosomes are combined to create one or more new offspring, allowing for the sharing of advantageous traits from both parents; and (3) ***Mutation***, the process of introducing random alterations to an individual's genes. This step is vital as it helps maintain genetic diversity within the population and prevents the algorithm from prematurely converging on a suboptimal solution.

3.2 Problem Formulations

The deployment of RFID readers in workspaces can be utilized across various applications, including warehouse and inventory monitoring, as well as tracking medical devices within healthcare facilities. The monitored areas may take on diverse shapes, but for the sake of clarity, this paper focuses on a rectangular workspace measuring X × Y square meters, illustrated in Fig. 1. Within this area, m RFID tags (represented by blue x's) are distributed uniformly and randomly. The objective is to determine an optimal configuration for n RFID readers (depicted as red triangles) that maximizes coverage, defined as the number of tags detected, while adhering to specific constraints.

Given geographical coordinates in two-dimensional space, infinite RFID reader placement locations exist. To streamline selection, the workspace is divided into a grid where each cell represents a feasible placement option. Cell size creates a trade-off: larger cells reduce placement possibilities and accelerate solution time but may compromise optimality. In comparison, smaller cells increase potential locations, requiring more computation time but significantly enhancing solution quality.

Therefore, it is essential to strike a balance between optimality and the time needed to determine the placement configuration. This study employs a quadrilateral mesh (see Fig. 2), ensuring that the cell size is a divisor of the normalized reader coverage diameter

(r_n). With an RFID reader coverage radius of r = 3.69 m, the corresponding normalized coverage diameter is calculated to be d_n= 2 × 3.69 × cos(45°) ≈ 5.2 m. The cell sizes examined in this paper are 3.9 m, 2.6 m, and 1.3 m.

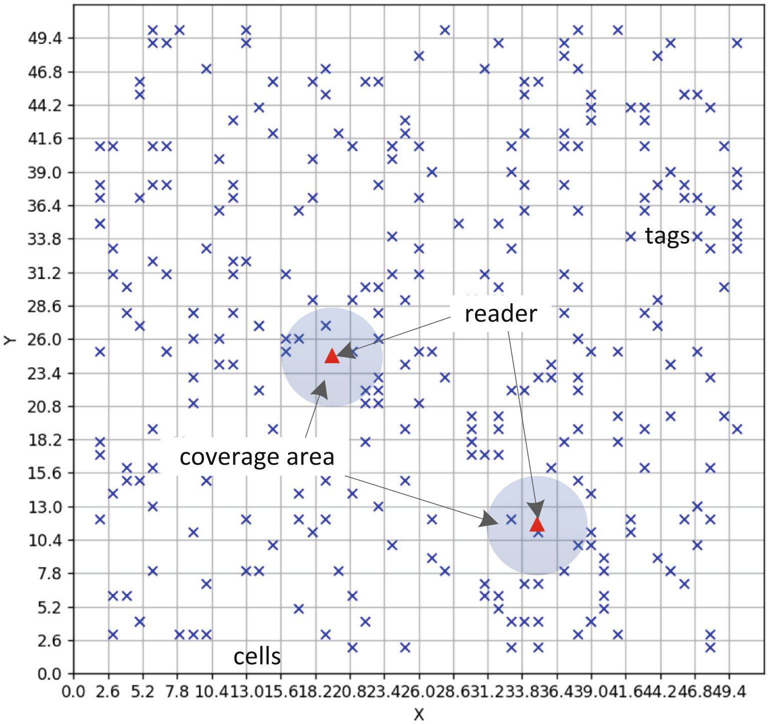


Fig. 1. Quadrilateral meshed workspace

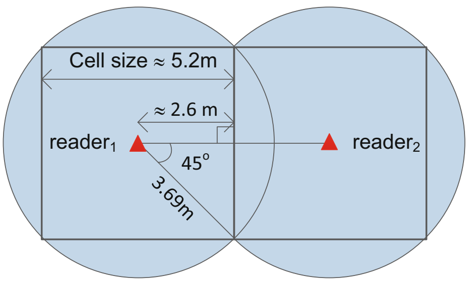


Fig. 2. With a coverage radius of 3.69 m, quadrilateral meshing will create mesh sizes of 3.9, 2.6 and 1.3 m

With the cell size defined, the number of cell rows and columns of the grid workspace is N = Y/d_n and M = X/d_n respectively. The grid workspace becomes a 2-dimensional cell matrix (as shown in Fig. 1.), where the state of each cell (*i, j*), $i = 1,\ldots,$ N and $j = 1,\ldots,$ M, is represented by the variable s_{ij}; $s_{ij} = 1$ if the cell is selected for reader placement and $s_{ij} = 0$ otherwise.

With the cell size defined, the grid workspace's cell rows and columns are $\mathrm{N} = \mathrm{Y}/d_n$ and $\mathrm{M} = \mathrm{X}/d_n$, respectively. The grid workspace becomes a 2-dimensional cell matrix (as shown in Fig. 1), where the state of each cell (*i, j*), $i = 1,\ldots,$ N and $j = 1,\ldots,$ M, is represented by the variable s_{ij}; $s_{ij} = 1$ if the cell is selected for reader placement and $s_{ij} = 0$ otherwise.

Objective

The main objective of the RNP problem is to cover the maximum number of tags in a working area, that is, the ratio of covered tags to the total number of tags reaches a maximum.

$$\text{maximize} f_1 = \sum_{i,j=1}^{N,M} \sum_{k \neq i, h \neq j}^{N,M} \frac{|T_{ij} \cup T_{kh}|}{m^2} s_{ij} s_{kh} \tag{1}$$

where T_{ij} is the set of covered tags if the reader is installed at cell (i, *j*); $|T_{ij}|$ is the number of tags the reader covers.

Constraints

The primary consideration for optimizing reader placement is energy efficiency. In an RFID network, the bulk of the energy consumption arises from the readers' activities. This includes the energy required to send interrogation commands to the tags, the energy used to maintain the downlink network (which allows the tags to gather energy for their responses), and the energy needed to receive and read the data from the tags [13]. As a result, the energy consumption of an RFID network is directly proportional to the number of tags being covered. In a work area, readers may have overlapping coverage zones, which can lead to some tags being picked up by multiple readers. This redundant interrogation of tags consumes unnecessary energy. To improve energy efficiency, it's important to minimize the number of tags in these overlapping areas.

$$\text{minimize} f_2 = \sum_{i,j=1}^{N,M} \sum_{k \neq i, h \neq j}^{N,M} \frac{|T_{ij} \cap T_{kh}|}{m^2} S_{ij} S_{kh} \tag{2}$$

where $T_{ij} \cap T_{kh}$ is the number of tags overlapping between two readers located at cells (i, j) and cells (*k, h*).

Multiple readers may be required to ensure coverage of most or all tags in the working area, increasing the size of the overlap area, leading to an increase in the number of tags in the overlap area and resulting in redundant interrogations. Therefore, the second constraint is minimizing the number of placed readers.

$$\text{minimize} f_3 = \sum_{i,j=1}^{N,M} \frac{S_{ij}}{n} \tag{3}$$

From Eqs. (1), (2), and (3), the RNP optimization problem can be expressed as optimizing a weighted sum of component objectives and constraints.

$$\text{minimize} f = -a_1 \sum_{i,j=1}^{N,M} \sum_{k \neq i, h \neq j}^{N,M} \frac{|T_{ij} \cup T_{kh}|}{m^2} s_{ij} s_{kh} + a_2 \sum_{i,j=1}^{N,M} \sum_{k \neq i, h \neq j}^{N,M} \frac{|T_{ij} \cap T_{kh}|}{m^2} s_{ij} s_{kh} + a_3 \sum_{i,j=1}^{N,M} \frac{1}{n} s_{ij} \tag{4}$$

There are two additional constraints that are not included in Eq. (4) but are addressed through the placement region restriction technique used during the optimization process:

- Readers positioned at the edge of the working area can lead to external coverage. This not only wastes power consumption but also creates security vulnerabilities that can be exploited in network attacks. Thus, minimizing external coverage is considered the third constraint of the RNP problem.
- When choosing a location for a new reader, it is essential to select a spot that is sufficiently distant from existing readers to minimize overlap. Each installed reader establishes a restricted placement zone around it. The placement zone restriction technique ensures that new readers cannot be placed within this designated area. Implementing Hopfield network to solve the RNP problem

A Hopfield network is a type of recurrent neural network made up of a layer of fully connected neurons. It is often used for optimization tasks, which is why we implement it to address the RNP problem in this paper. Consider a Hopfield network consisting of N × M neurons. Its energy function is

$$E = -\frac{1}{2} \sum_{i,j=1}^{N,M} w_{ij} v_i v_j + \sum_{i=1}^{N} \theta_i v_i \tag{5}$$

where v_i and v_j are the activation values of neurons i and j, and w_{ij} is the weight of the connection between neurons i and j.

After each execution cycle, the variance of the energy is

$$\Delta E = -\left(\frac{1}{2} \sum_{j=1}^{M} w_{ij} v_j + \theta_i \right) \times \Delta v_i \tag{6}$$

As shown in [14], ΔE is never positive, meaning E never increases, regardless of any variation of Δv_{i}. That is the Hopfield-based optimization principle, which is "*minimum energy, optimal solution*".

Based on the optimization principle, the function in Eq. (4) is rewritten as

$$f = \sum_{i,j=1}^{N,M} \sum_{k \neq i,k,k,j}^{N,M} \frac{-a_1 |T_{ij} \cup T_{kh}| + a_2 |T_{ij} \cap T_{kh}|}{m^2} s_{kh} s_{ij} + \sum_{i,j=1}^{N,M} \frac{a_3}{n} s_{ij} - -\frac{1}{2} \sum_{i,j=1}^{N,M} \sum_{k=1,h=1}^{N,M} 2 \frac{a_1 |T_{ij} \cup T_{kh}| - a_2 |T_{ij} \cap T_{kh}|}{m^2} C_{ik} C_{jh} s_{kh} s_{ij} + \sum_{i,j=1}^{N,M} \frac{a_3}{n} s_{ij} \tag{7}$$

where C is a constant square matrix, $C_{ik=}$ 0 if $i = k$, otherwise $C_{ik=}$ 1.

By comparing Eqs. (7) and (5), the weights and thresholds are obtained as follows

$$w_{ijkh} = 2\frac{a_1|T_{ij} \cup T_{kh}| - a_2|T_{ij} \cap T_{kh}|}{m^2}C_{ik}C_{jh} \text{ and } \theta_{ij} = \frac{a_3}{n} \quad (8)$$

3.3 The Integrated GA-HNEE-RNP Model

The proposed model integrates a Genetic Algorithm with a Hopfield Network-based Energy-Efficient (HNEE) optimizer. The core principle is to leverage the GA's global search capabilities to find the optimal set of hyperparameters for the HN, which in turn solves the RNP optimization problem. The overall architecture, depicted in Fig. 3, consists of an outer loop managed by the GA and an inner loop where the HN performs the reader placement optimization.

Chromosome Representation and Fitness Evaluation

In the GA, each individual (chromosome) represents a candidate set of hyperparameters for the weighted objective function in Eq. (4). A chromosome is a vector of three real-valued genes, $A = \{a_{1,}\ a_{2,}\ a_3\}$, corresponding to the weights for tag coverage, tag overlap, and the number of readers, respectively. These genes are initialized randomly, subject to the constraint that their sum must equal 1 ($a_1 + a_2 + a_3 = 1$) to ensure a normalized weighting scheme.

The fitness of each chromosome A is determined by its performance in solving the RNP problem. For each individual A in the GA population, the HNEE optimizer is executed. The genes $\{ a_1 + a_2 + a_3 \}$ are used to calculate the HN's weights and thresholds as per Eq. (8). The HN then runs until it converges to a stable reader placement solution. The resulting values for tag coverage, overlap, and number of readers are then used to calculate a fitness score. The fitness function is identical to the objective function f in Eq. (4), where the goal is minimization. A lower value of f indicates a better solution and therefore a higher fitness.

Genetic Algorithm Operators

The GA employs specific operators chosen for their effectiveness with real-valued chromosomes:

- ***Selection***: Tournament selection is used to choose parents for breeding. In this method, a small subset of individuals (a "tournament") is randomly selected from the population, and the fittest individual from this subset is chosen as a parent. This process is repeated to select the second parent. Tournament selection was chosen as it reduces stochastic noise and is not susceptible to premature convergence caused by a single, dominant "super-individual," which can be a problem with methods like roulette wheel selection.
- ***Crossover***: Blend Crossover (**BLX-α**) is applied to create offspring from the selected parents. This method is well-suited for real-valued genes, as it generates new genes that lie within an expanded range defined by the parent genes, promoting exploration of the solution space. An alpha coefficient of 0.5 was used.

- ***Mutation***: Gaussian mutation is applied to each gene with a set probability. This operator adds a random value from a Gaussian distribution (with mean 0 and a specified standard deviation) to the gene. This is appropriate for continuous parameters as it allows for fine-tuning of the hyperparameter values.

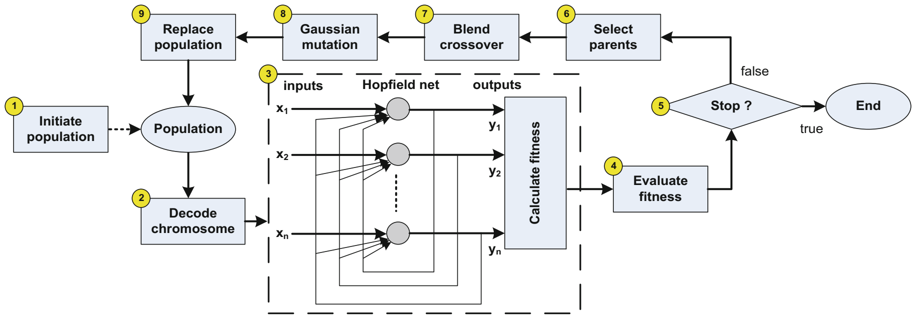


Fig. 3. The Genetic algorithm - Hopfield network integrated model

3.4 Computational Complexity Analysis

Integrating a genetic algorithm with the Hopfield network (HNEE) to optimize weights (a_1, a_2, and a_3) enhances the distribution efficiency of RFID readers. However, this integration does come with a significant increase in computational complexity.

In a typical Hopfield network run, the most time-consuming part is the neuron update loop. For a grid of size N × M, there are N × M neurons, and updating each neuron involves summing inputs from all other neurons. This results in a complexity of O((N × M)2). If the network requires K iterations to reach convergence, the total complexity for one Hopfield network execution becomes O(K × N2 × M2).

In this integrated model, the genetic algorithm (GA) acts as an outer loop, running the HNEE optimizer for each individual in the population across multiple generations. Consequently, the overall complexity for the combined GA-HNEE-RNP model is O(P × S × K × N2 × M2), where P represents the number of GA generations and S stands for the population size.

Despite the increased computational cost, the GA significantly improves the results in terms of the number of required readers and the level of tag coverage. This represents a reasonable trade-off between computational efficiency and solution quality, especially in offline planning tasks where the runtime is less critical than the quality of the final network deployment.

4 Simulation and Analysis of Results

The simulation is performed in Python, on an Intel(R) Core(TM) i5, 2.40 GHz, 8GB RAM computer. The workspace is assumed to be a 52 × 52 m square, where the space is gridded with different cell sizes and the minimum distance between readers (r2r) is

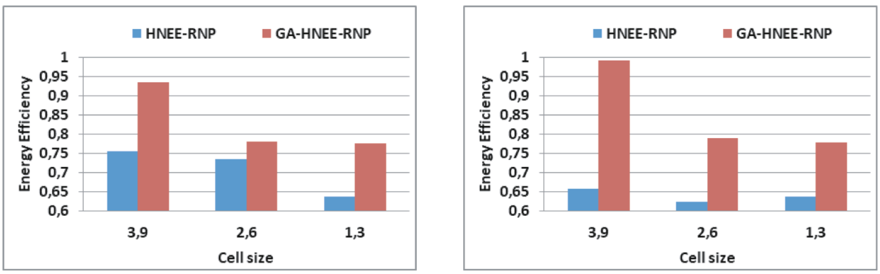
2.6 m. Two scenarios are set for tag distribution: uniform and non-uniform distribution, where tags are densely packed in the lower left corner and sparsely packed in the remaining corners. The goal of the simulation is to evaluate the performance of the proposed integrated model, called GA-HNEE-RNP, compared with HNEE-RNP in [15], in terms of the following criteria: tag coverage ratio (calculated as the number of covered tags divided by the total number of tags in the workspace), energy efficiency (defined as in [15]), algorithm runtimes and number of readers required. Details of simulation parameters are shown in Table 1.

Table 1. Simulation parameters

No.	Parameters	Value
1	Work area size (m^2)	52 × 52
2	Number of tags in the workspace	300
3	Grid size (m)	3.9, 2.6, 1.3
4	Minimum distance between readers (r2r) (m)	2.6
5	Objective function weights and initial constraints (a_0, a_1, a_2)	0.5, 0.3, 0.2
6	Population size (S) of GA	20
7	Number of generations (P) of the GA	20
8	Probability of crossbreeding	0.5
9	Mutation probability	0.2

4.1 Energy Efficiency

The energy efficiency of GA-HNEE-RNP is significantly improved compared to HNEE-RNP, as shown in Fig. 4 with even distribution, the improvement rate of GA-HNEE-RNP is about 17%, but with uneven distribution, this value increases to 33%.



(a) When the tags are evenly distributed (b) When the tags are unevenly distributed

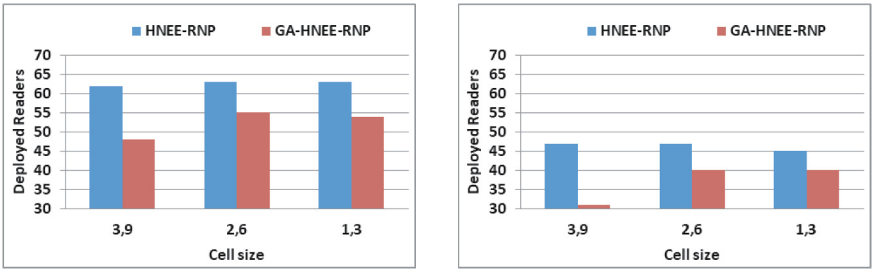
Fig. 4. Energy efficiency comparison between GA-HNEE-RNP and HNEE-RNP

Section 3.1 examines energy efficiency optimization by strategically reducing distributed readers and minimizing overlapping and external coverage areas. The framework uses Hopfield network-based placement constraints with genetic algorithms to optimize reader deployment while maintaining coverage effectiveness.

The GA-HNEE-RNP model demonstrates superior performance by combining Hopfield networks and genetic algorithms for resource allocation in energy-constrained applications. This approach validates the importance of sophisticated optimization methods for developing sustainable, energy-efficient distributed reader networks across diverse technological domains.

4.2 Number of Readers Used

As shown in Fig. 5, the GA-HNEE-RNP algorithm reduces reader requirements by 17% in uniform distribution scenarios and 20% in uneven distribution cases compared to HNEE-RNP. This reduction streamlines operations, lowers energy consumption, and minimizes overlapping coverage areas for more efficient resource utilization.



(a) When the tags are evenly distributed (b) When the tags are unevenly distributed

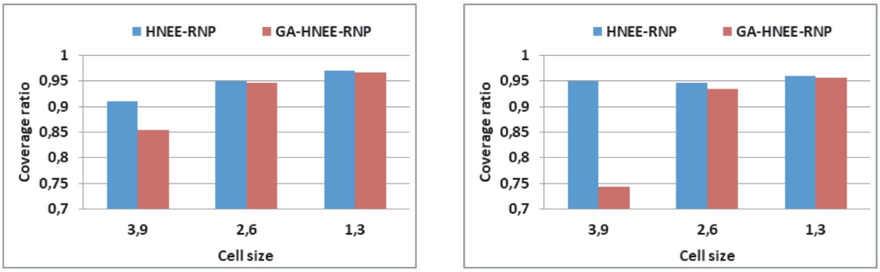
Fig. 5. Comparison of the number of readers required to be distributed between GA-HNEE-RNP and HNEE-RNP

The improved reader deployment maximizes performance while reducing ecological impact through cost savings and decreased energy expenditure. Enhanced efficiency eliminates unnecessary overlaps, enabling more targeted coverage where needed most.

4.3 Tag Coverage Ratio

While GA-HNEE-RNP reduces reader requirements, it slightly decreases coverage compared to HNEE-RNP: 2% in even distribution and 8% in uneven distribution scenarios. These trade-offs highlight the balance between optimizing reader count and maintaining coverage.

For applications requiring 100% coverage, adjusting weighting parameters by prioritizing specific weights can enhance performance and meet coverage targets. This strategic adjustment compensates for reduced coverage ratios while ensuring efficient tag distribution for real-world applications (Fig. 6).



(a) When the tags are evenly distributed (b) When the tags are unevenly distributed

Fig. 6. Comparison of coverage ratio between GA-HNEE-RNP and HNEE-RNP

4.4 Running Time

A significant limitation of the GA-HNEE-RNP model lies in the substantial computational overhead required for parameter optimization, as evidenced in Table 2. This challenge reflects a fundamental trade-off inherent in complex optimization systems: balancing solution quality against computational efficiency and available processing resources. At the same time, modern computing advances have considerably accelerated execution times for computationally intensive problems, and integrated hybrid models such as GA-HNEE-RNP demand further optimization to achieve practical deployment viability.

Table 2. Comparison of running time (ms) between GA-HNEE-RNP and HNEE-RNP

Cell size	Evenly distributed tags		Unevenly distributed tags	
	HNEE-RNP	GA-HNEE-RNP	HNEE-RNP	GA-HNEE-RNP
3.9	3.6	300	3.5	312
2.6	18	1010	20	1005
1.3	200	5600	198	5786

Experimental results demonstrate the GA-HNEE-RNP model's effectiveness, achieving up to 33% energy efficiency improvement and 20% reduction in required readers through automated hyperparameter optimization. The GA effectively balances minimizing overlap and reader count for more cost-effective, energy-conscious solutions than manually-set baseline weights.

The reduced coverage (2–8%) is a trade-off from using fewer readers, with some peripheral tags potentially missed. While execution time increases substantially, this computational investment is justified for RFID network planning. In this one-time, offloading task, upfront processing costs are offset by multi-year savings in hardware and energy consumption.

5 Conclusion

This study addresses optimization challenges in NP-hard problems using nature-inspired methods, specifically proposing the GA-HNEE-RNP algorithm that integrates genetic algorithms (GA) with Hopfield networks (HN). By embedding HN within GA, the algorithm streamlines the identification of optimal solutions from GA-generated parameters.

Results show GA-HNEE-RNP outperforms traditional HNEE-RNP in energy efficiency and reader reduction, though with trade-offs including slightly lower coverage rates and increased execution time. These limitations could be addressed through enhanced computational resources or model refinement.

Study limitations include testing only in simple, obstacle-free square environments with uniform circular coverage models. Future research should implement the model in complex environments with physical obstacles and RF signal propagation challenges, incorporate advanced propagation models like ray-tracing for realistic simulations, and expand GA capabilities to optimize additional parameters such as grid cell size and diverse reader models. Additionally, comparative analysis against contemporary hybrid models would provide valuable performance benchmarking and ensure competitiveness in the evolving optimization field.

References

1. Chen, H., Zhu, Y.: RFID networks planning using evolutionary algorithms and swarm intelligence. In: 4th International Conference on Wireless Communications. Networking and Mobile Computing, pp. 1–4. IEEE, Dalian (2008)
2. Suriya, A., Porter, J.D.: Genetic algorithm-based approach for RFID network planning. In: TENCON 2014 - 2014 IEEE Region 10 Conference, pp. 1–5. IEEE, Bangkok (2014)
3. Bhattacharya, I., Roy, U.: Optimal placement of readers in an RFID network using particle swarm optimization. Int. J. Comput. Netw. Commun. **2**(6), 203–215 (2010)
4. Ma, L., Chen, H., Hu, K., Zhu, Y.: Hierarchical artificial bee colony algorithm for RFID network planning optimization. Sci. World J. **2014**, 941532 (2014)
5. Bacanin, N., Tuba, M., Jovanovic, R.: Hierarchical multiobjective RFID network planning using firefly algorithm. In: International Conference on Information and Communication Technology Research (ICTRC), pp. 282–285. IEEE, Abu Dhabi (2015)
6. Jaballah, A., Meddeb, A.: Self adaptive cuckoo search algorithm for RFID network planning. In: Internet Technologies and Applications (ITA), pp. 122–127. IEEE, Wrexham (2017)
7. Shi, W., et al.: Optimal implementation of phased array antennas for RFID network planning based on an improved chicken swarm optimization. IEEE Internet Things J. **8**(19), 14572–14588 (2021)
8. Feng, H., Qi, J.: Optimal RFID networks planning using a hybrid evolutionary algorithm and swarm intelligence with multi-community population structure. In: 14th International Conference on Advanced Communication Technology (ICACT), pp. 1063–1068. IEEE, PyeongChang (2012)
9. Cao, Y., Liu, J., Xu, Z.: A hybrid particle swarm optimization algorithm for RFID network planning. Soft. Comput. **25**(8), 5747–5761 (2021)
10. Aghdam, A.S., Kazemi, M.A.A., Eshlaghy, A.T.: A hybrid GA–SA multiobjective optimization and simulation for RFID network planning problem. J. Appl. Res. Indust. Eng. **8**(1), 1–25 (2021)

11. Yixuan, Q., Jiali, Z., Xiaode, X., Zihan, L., Wencong, L.: Hybrid gray wolf optimization-cuckoo search algorithm for RFID network planning. J. China Univ. Posts Telecommun. **28**(6), 91–102 (2021)
12. Maimouni, M., Majd, B.A.E., Bouya, M.: RFID network planning using a new hybrid ANNs-based approach. Connect. Sci. **34**(1), 2265–2290 (2022)
13. Golsorkhtabar, M., Issazadeh, N., Pouresfehani, N., Mohammadialamoti, M., Hosseinzadehsadati, S.: Comparison of energy consumption for reader anti-collision protocols in dense RFID networks. Wireless Netw. **25**(6), 3159–3171 (2019)
14. Vo, V.M.N.: An application of neural networks in the connection admission control of ATM networks. In: IEEE-RIVF International Conference on Computing and Communication Technologies, pp. 1–5. IEEE, Danang (2009)
15. Hoa, L.V., Tung, N.V., Nhat, V.V.M.: An approach to hopfield network-based energy-efficient RFID network planning. Cybern. Inform. Technol. **24**(2), 50–66 (2024)
16. Zhang, Y., Geng, L.: An ant colony optimization algorithm for solving RFID network planning problem. Procedia Eng. **15**, 2119–2123 (2011)
17. Zhang, C., Wang, Y., Liu, Z.: An improved simulated annealing algorithm for RFID network planning. In: 2010 International Conference on Communications and Mobile Computing, pp. 238–242. IEEE, Shenzhen (2010)