A Neural Network-Based Optimization Approach for the RNP Problem in Arbitrary Shaped Workspaces

Nguyen Van Tung, Le Van Hoa, and Vo Viet Minh Nhat

Abstract In order to monitor tags or tagged objects in a workspace, RFID readers need to be arranged so that the RFID reader network can cover most of the tags and simultaneously satisfy some constraints, such as minimized deployment cost, maximized interrogation efficiency, and load balancing among readers. The problem of optimizing the deployment of RFID readers is known as the RFID Network Planning (RNP) problem and is considered NP-hard. Different optimization methods have been proposed, among which nature-inspired approaches often give more impressive results. However, finding an optimal solution in a reasonable time for the multi-objective problem is always challenging. This paper presents a neural network-based approach in which the reader deployment optimization problem is formulated as an energy function and minimized by a Hopfield network. The achieved minimum energy determines the optimal deployment solution. With traditional natural-based optimization methods, discrete populations are initialized and searched in the candidate solution space, often leading to local optima. The Hopfield network-based approach minimizes an energy function of the entire network, thus overcoming the limitation. Furthermore, with the predetermined connection weights and activation threshold, the training phase is omitted, thus shortening the optimization time. Experimental results show that the optimization process converges faster, and the readers' optimal positions are found early.

Keywords RNP · Gridding · Optimization · Hopfield network · Genetic algorithm

N. Van Tung (✉)
University of Sciences, Hue University, Hue, Vietnam
e-mail: nvtung@hueuni.edu.vn

Ho Chi Minh City University of Industry and Trade, Ho Chi Minh City, Vietnam

L. Van Hoa
School of Hospitality and Tourism, Hue University, Hue, Vietnam
e-mail: levanhoa@hueuni.edu.vn

V. V. M. Nhat (✉)
Institute of Educational Testing and Quality Assurance, Hue University, Hue, Vietnam
e-mail: vvmnhat@hueuni.edu.vn

N. T. Nguyen et al. (eds.), *The 14th Conference on Information Technology and its Applications*, Lecture Notes in Networks and Systems 1581,
https://doi.org/10.1007/978-3-032-00972-2_42

1 Introduction

Radio Frequency Identification (RFID) is a wireless communication technology that automatically identifies tags or tagged objects. Along with the rapid development of wireless sensor technology and automatic identification technology in recent years, RFID has been used to build Internet of Things (IoT) systems deployed in fields such as supply chain management, inventory monitoring, or warehouse security [1, 2].

Due to the limited transmission range of antennas, RFID readers need to be networked to be deployed in a large work area to cover most of the tags or tagged objects that need to be monitored. An RFID system is then constructed, consisting of tags and networked RFID readers connected to a server. Constructing an RFID network aims to achieve maximum coverage while satisfying several constraints, such as minimizing the number of readers required, minimizing interference/overlap, and balancing the load among readers. The problem is known as the RFID network planning (RNP) problem, which is NP-hard. RNP is a large-scale, multidimensional, nonlinear optimization problem with many uncertain parameters [3].

Due to the complexity of the RNP problem, traditional optimization methods cannot solve it, and nature-inspired approaches are proposed. A popular approach is formulating the RNP problem as a weighted sum of objective functions and constraints. Then one of the nature-based methods is applied, such as Genetic Algorithm (GA) in [4], Particle Swarm Optimization (PSO) in [5], Cuckoo Search algorithm in [6], Ant Colony Optimization in [7], Bacterial Foraging Optimization in [8], and Chicken Swarm Optimization in [9].

Despite the significant improvement in optimization results compared to traditional optimization methods, pure nature-based optimization methods often get stuck at local optimization positions. In order to improve the optimization level, some combinations of nature-based optimization methods or a nature-based optimization method with other techniques have been proposed. The first combination of GA and PSO was proposed in [10] with the main idea of dividing the single population of PSO into several colonies and use genetic selection and mutation strategies to improve the dynamic rules of the particle swarm. PSO can be combined with a Tentative Reader Elimination (TRE) operator [11] to minimize the number of readers needed by temporarily deleting readers during the PSO process and can restore deleted readers after a few generations if the deletion reduces tag coverage. Another combination of PSO with differential evolution strategies [5], which forms the Differential Evolutionary Particle Swarm Optimization (DEEPSO) model, on the one hand, relies on past information of the optimization process to generate new solutions and replaces individual memory with collective memory, and, on the other hand, has a self-adaptive property thanks to the self-adaptive recombination operator.

With GA, the method can be combined with Simulated Annealing (SA) in a two-stage process [12]. First, GA initiates the optimization process, and the results are the input parameters for SA in the second stage. The GA–SA model has exploited the advantages of the two methods while bypassing their disadvantages. GA can also be integrated with the redundant reader elimination (RRE) technique in a two-stage process to minimize the number of placed readers [13]. GA first finds the optimal position of readers and then RRE eliminates the redundant readers without impacting tag interrogation efficiency.

Some other combinations of different nature-inspired algorithms have also been proposed. Typically, a hybrid grey wolf-optimized cuckoo search (GWO-CS) algorithm is proposed in [14], which uses an input representation based on a random grey wolf search and evaluates tag density and location to determine the combined performance of the reader's propagation area. A combination of redundant antenna elimination (RAE) and neural network algorithm (NNA) is introduced in [15], in which RAE focuses on optimizing RNP by eliminating redundant antennas and NNA, minimizing the difference between the target solution and the forecasted solutions.

However, the combination approach only uses an addition technique to refine or improve the optimal result found by a natural-based optimization algorithm. The approach is challenging to achieve global optimization because there is no action to escape from the local optimal position. The paper proposes an approach based on the Hopfield network, in which the objective function and constraints of RNP are formulated as an energy function. The connection weights and activation thresholds are predetermined instead of going through the training process like other neural networks. A Hopfield network is then constructed to minimize the objective function. Our approach affects the entire network, so it not only succeeds in finding the global optimal solution but also increases the convergence speed of the optimization process.

The main contributions of the paper include:

- Formulating the RNP optimization problem as minimizing a Hopfield network energy function and constructing a suitable Hopfield network to minimize this energy function;
- Proposing the techniques for stimulating neural inputs, fine-tuning neuron activation, and constructing deployment constraint regions. Combining these techniques has reduced the computational volume and algorithm execution time and, as a result, increased the convergence speed of the optimization process;
- Implementing an improved Hopfield network-based optimization method for an arbitrary shaped workspace: Experimental results show that the proposed method has many superior advantages, such as coverage efficiency, deployment efficiency, interrogation efficiency, and load balancing, with several scenarios.

The following sections of the paper are organized as follows: Sect. 2 describes the formulation of the RNP problem as minimizing an energy function, which is minimized by a Hopfield network. The contents of this section include an introduction to a basic RFID system, the formulation of the RNP problem as a multi-objective function, the optimization principle based on the Hopfield network, and the RNP optimization algorithm based on the Hopfield network. Section 3 describes and analyzes the implementation results of Hopfield Network-based RNP optimization with several different scenarios of tag density, comparing it with Genetic Algorithm-based RNP optimization. The conclusion belongs to Sect. 4.

2 RFID Network Planning Formulation

2.1 An RFID System and Its Components

An RFID system consists of tags, readers, and a server to receive and process the collected data (Fig. 1). RFID tags consist of two parts: a chip and an antenna. The chip is a small system that stores the chip's ID string. The chip uses the antenna to receive power to respond to its ID to the reader. Tags can be active, passive, or semi-passive. Readers can be deployed either fixed or mobile. Depending on the implementation, the connection between readers and the server can be wired, wireless, or both. The paper considers passive tags and fixed readers.

The RFID reader reads the tag if the tag is within the reader's interrogation zone. The reader initiates the communication by sending a command and maintaining a carrier pulse. The passive tag harvests energy from the pulse to send back its ID string. The communication between the reader and the tag is affected by several parameters expressed in the Friis transmission equation [16]. In practice, the received signal is affected by environmental interference (metallic materials, obstacles, etc.) within the reader's interrogation zone.

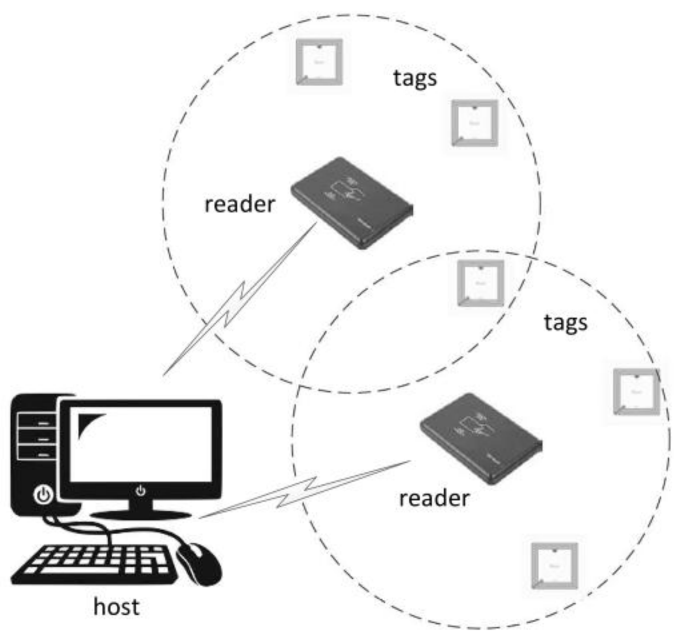


Fig. 1 An RFID system and its components

2.2 Formulating the RNP Problem in Terms of Multi-objective Optimization

We consider an arbitrary shaped workspace. In order to reduce the infinite number of possible deployment locations, the workspace is gridded (Fig. 2), where the cell size can be adjusted to reduce the number of possible deployment locations. If the cell size is small, the number of candidate locations is large, and the time needed to find an optimal deployment is significant. In contrast, if the cell size increases, the optimization time is shortened due to the reduction in candidate locations. Choosing an appropriate grid size compromises optimal deployment locations and running time.

To perform the gridding, we consider a rectangle, with size $M \times N$, containing an arbitrary shaped workspace (Fig. 2). Each cell is a square with an edge length of d. The number of cell rows and columns are $m = M/d$ and $n = N/d$, respectively. Assume that m and n are integers; if m and n are not integers, M and N can be adjusted so that m and n are integers and the grid $M \times N$ still covers the entire work area.

As in Fig. 2, cells not in the workspace (white cell) are not considered for reader deployment, while dark cells are candidate locations. Considering that the cell size d is always smaller than the coverage radius of each reader antenna ($r = 3.69$ m), cell locations at the edge of the workspace are not of concern because the coverage area will extend beyond, causing waste and security risks.

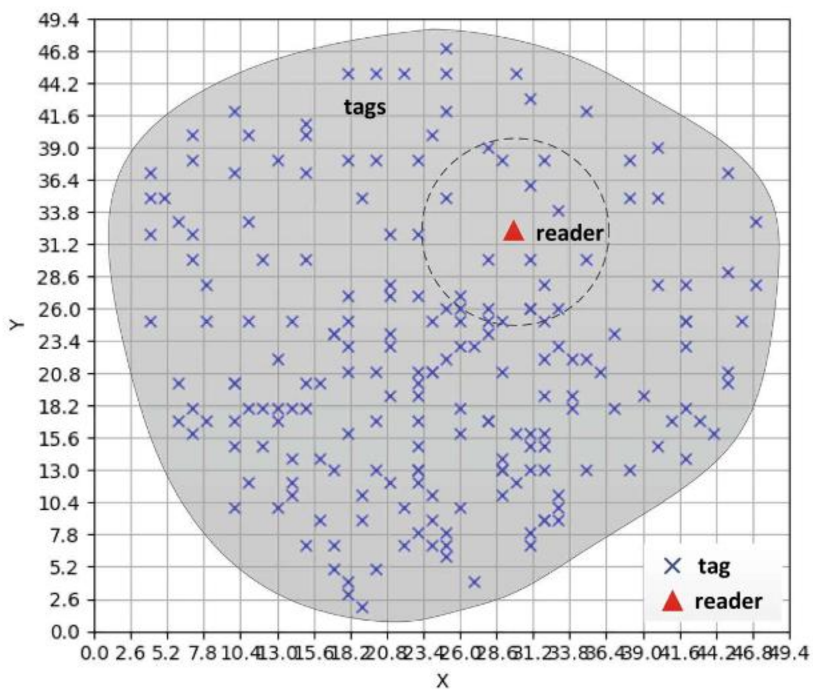


Fig. 2 The workspace is enclosed by a gridded rectangle

Each reader, if deployed at a cell c_{ij}, $i = 1..m$, $j = 1..n$, covers a set of tags T_{ij}. A set of N_r readers is available, but only n_r ($n_r \leq N_r$) readers are deployed. The covered tags (n_t) are usually less than or equal to the total tags in the workspace N_t ($N_t \geq n_t$).

Deploying readers aims to maximize the number of tags covered by the readers' network, which means maximizing the following objective function:

$$f_1 = \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cup T_{kh}\right|}{N_t} c_{ij} c_{kh} \tag{1}$$

where, $c_{ij} = 1$ if a reader is installed at cell (i,j); $c_{ij} = 0$ if no reader is installed.

Maximizing coverage also comes with some constraints. The following are the constraints of RNP optimization.

1. Deployment efficiency: achieved by minimizing the number of readers deployed:

$$f_2 = \sum_{i,j=1}^{m,n} C_r c_{ij} \tag{2}$$

   where C_r is the deployment cost per reader, normalized to the interval [0,1].
2. Interrogation efficiency: The main factor affecting interrogation efficiency is interference. Tags in interference areas must be interrogated many times before being recognized. Repeated interrogations not only consume the reader's energy but also consume interrogation bandwidth. Therefore, increasing interrogation efficiency is to reduce the number of tags in the overlapping area of the readers.

$$f_3 = \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cap T_{kh}\right|}{N_t} c_{ij} c_{kh} \tag{3}$$

3. Load balancing among readers: A reader responsible for reading too many tags consumes much energy, exhausts its processing capacity, and reduces the system's lifetime. In a wireless network, a node that dies early can disrupt the communication structure of the network and reduce the tag coverage. Load balancing among readers is therefore interpreted as balancing the number of tags each reader is responsible for.

$$f_4 = \sum\nolimits_{i,j=1}^{m,n} \sum\nolimits_{j=1}^{n} \left(\frac{\left|T_{ij}\right| - N_a}{N_a} \right)^2 c_{ij} \tag{4}$$

   where $N_a = N_t / N_r$ is the average number of tags each reader is responsible for.

Finally, the objective function of the RNP optimization problem is the weighted sum of the component objectives and constraints from (1–4):

$$\begin{aligned} f &= -a_1 f_1 + a_2 f_2 + a_3 f_3 + a_4 f_4 \\ &= -a_1 \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cup T_{kh}\right|}{N_t} c_{ij} c_{kh} + a_2 \sum_{i,j=1}^{m,n} C_r c_{ij} \\ &\quad + a_3 \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cap T_{kh}\right|}{N_t} c_{ij} c_{kh} + a_4 \sum_{i,j=1}^{m,n} \left(\frac{\left|T_{ij}\right| - N_a}{N_a} \right)^2 c_{ij} \end{aligned} \tag{5}$$

2.3 Hopfield Network-Based RNP Optimization

This paper considers the use of a Hopfield network for RNP optimization. A Hopfield network is a recurrent neural network consisting of a layer of neurons where each neuron is connected to all the remaining neurons. A prominent feature of a Hopfield network is that its energy function (essentially, it is an error function) does not increase during execution, so it is used for optimization tasks. If an optimization problem can be formulated regarding the Hopfield network energy function, then this network can be used to find the optimal solution [17].

Consider a Hopfield network consisting of $X \times Y$ neurons; its energy function is

$$E = \frac{1}{2} \sum_{i,j=1}^{X,Y} w_{ij} x_i x_j + \sum_{i=1}^{X} \theta_i x_i \tag{6}$$

where x_i and x_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

$$\Delta E_i = -\left(\frac{1}{2} \sum_{j=1}^{Y} w_{ij} x_j + \theta_i \right) \Delta x_i \tag{7}$$

As proved in [18], ΔE_i is never positive, meaning that E never increases, regardless of the variation of Δx_i. This is the Hopfield-based optimization principle: "*The less the energy, the more optimal the solution.*" Eq. (5) is therefore rewritten as:

$$\begin{aligned} f = &-a_1 \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cup T_{kh}\right|}{N_t} c_{ij} c_{kh} + a_2 \sum_{i,j=1}^{m,n} C_r c_{ij} + a_3 \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{\left|T_{ij} \cap T_{kh}\right|}{N_t} c_{ij} c_{kh} \\ &+ a_4 \sum_{i,j=1}^{m,n} \left(\frac{\left|T_{ij}\right|}{N_a} \right)^2 C_{ik} C_{jh} c_{ij} c_{kh} - 2a_4 \sum_{i,j=1}^{m,n} \frac{\left|T_{ij}\right|}{N_a} c_{ij} + a_4 \end{aligned}$$

$$= -a_1 \sum_{i,j=1}^{m,n} \sum_{k,h=1}^{m,n} \frac{1}{N_t} \left(-a_1 \left| T_{ij} \cup T_{kh} \right| + a_3 \left| T_{ij} \cap T_{kh} \right| + a_4 \left(\frac{\left| T_{ij} \right|}{N_a} \right)^2 C_{ik} C_{jh} \right) c_{ij} c_{kh}$$

$$+ a_2 \sum_{i,j=1}^{m,n} \left(a_2 C_r - 2a_4 \frac{\left| T_{ij} \right|}{N_a} \right) c_{ij} + a_4 \tag{8}$$

where, a_i, ($i = 1..4$ and $\sum_1^4 a_i = 1$); C is a constant matrix, such that $C_{ik} = 1$ if $i = k$, and $C_{ik} = 0$, otherwise.

Compared with the Hopfield network energy Eq. (6), the weights and activation thresholds of the Hopfield network are determined as

$$W_{ijkh} = \frac{1}{N_t} \left(-a_1 \left| T_{ij} \cup T_{kh} \right| + a_3 \left| T_{ij} \cap T_{kh} \right| + a_4 \left(\frac{\left| T_{ij} \right|}{N_a} \right)^2 C_{ik} C_{jh} \right) \tag{9}$$

$$\theta_{ij} = a_2 C_r - 2a_4 \frac{\left| T_{ij} \right|}{N_a} \tag{10}$$

The proposed Hopfield network for RNP optimization is shown in Fig. 3, where each neuron corresponds to a cell in the workspace. The connection weights between neurons and the activation threshold of each neuron have been determined by Eqs. (6) and (7). To perform the optimization process, a neuron, say (i,j), is randomly selected to stimulate. The stimulation at this neuron will create an activation effect on all the remaining neurons, say (k,h), through the weighted sum of the connections between neurons (i,j) and neurons (k,h). The neuron with the highest activation is selected, and the cell corresponding to this neuron is the location where a new reader can be deployed. The process is repeated until all readers have been deployed or the tags have been covered by the deployed readers. In case only a certain percentage of covered tags is required, a threshold value is predefined, which is one of the stopping conditions of the optimization process.

The Hopfield network-based optimization algorithm contains some random elements, such as randomly selecting neurons to stimulate and randomly selecting one of the neurons with the largest activation, which reduces the optimization direction and slows down the algorithm's convergence. To improve these issues, this paper proposes a method to adjust the restricted region, including the following techniques:

- Each reader deployed determines its coverage area, which consists of its neighboring cells, called the restricted region. The next reader deployed should avoid the restricted region to reduce the overlap and thus reduce the tag interference. Therefore, when randomly selecting a neuron to stimulate, the neurons corresponding to the cells in the restricted region (briefly, neurons in the restricted region) are not selected.
- With the same idea of avoiding the restricted region above, a neuron with the largest activation may also belong to the restricted region, and the corresponding

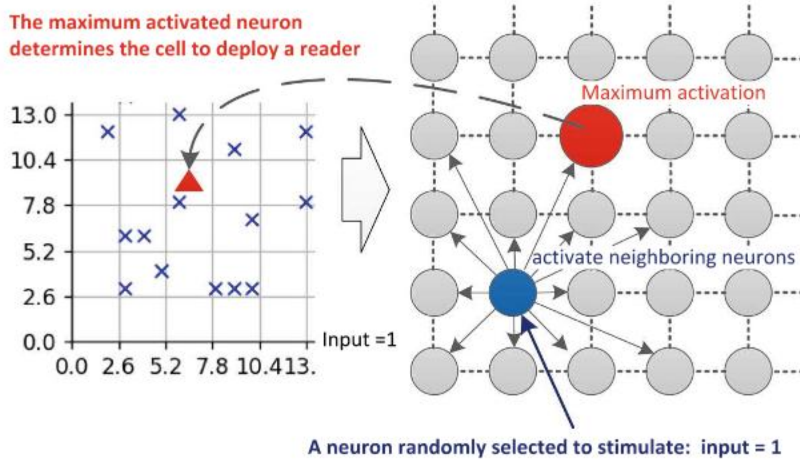


Fig. 3 The Hopfield network deployed corresponds to the gridded workspace

cell should not be selected for reader deployment. This is reflected in the optimization algorithm, which states that neurons in the restricted region are not calculated for activation, and the selected neuron with the largest activation is not in the restricted region.

- A cell for deploying a new reader is not available if the number of tags covered by the reader is in the overlapping area with other deployed readers. Therefore, if at least two additional covered tags are not in the overlapping area with other deployed readers, the cell is selected for deploying a new reader.
- For an arbitrary shaped workspace, cells not part of the workspace are also added to the restricted region. This ensures that deployed readers are not outside the workspace.

These improvements promise to reduce the computational load, the running time, and thus the complexity of the Hopfield network-based optimization algorithm.

2.4 Hopfield Network Based RNP Optimization Algorithm

The Hopfield network-based RNP optimization is performed by minimizing an energy function by the Hopfield network. Therefore, the Hopfield network-based RNP optimization algorithm is called Hopfield Network-based RNP Optimization, HN-RNPO. The minimization process is run as follows.

Algorithm 1. HN-RNPO algorithm

Input: - N_r // maximum number of readers that can be deployed
- N_t // number of tags in the workspace

Output: - c^* // list of cells selected to deploy readers

Step 1. Initialize the Hopfield network with $n \times m$ neurons, corresponding to the matrix of $n \times m$ cells (c); the inputs are initialized to 0; the connection weights and the activation threshold are determined by Eqs. (9) and (10), respectively;
Step 2. Initialize the restricted region, initially the cells outside the workspace;
Step 3. While (there are still available readers) and (there is a suitable cell to place a new reader) do
Step 4. Randomly select a non-restricted neuron and excite it with input = 1;
Step 5. Calculate the activations of the non-restricted neurons and determine the neuron with the largest activation;
Step 6. The cell corresponding to the largest activation neuron is selected for deploying a new reader; the list of cells selected for deploying reader c^* is updated, and the restricted region g^* is expanded.
Step 7. Repeat the steps from step 3–6 until the stopping condition is achieved.

The complexity of the HN-RNPO algorithm mainly comes from calculating the activation levels of the neurons. For a Hopfield network with $n \times m$ neurons, the complexity of calculating the activation level is $O(n^2 \times m^2)$. Accordingly, the algorithm's complexity will increase significantly when the number of cells in the workspace is large. However, because the restricted region increases, the calculation time decreases significantly.

3 Simulation and Analysis

The simulation is performed in an arbitrary shaped workspace, which can be assumed to be enclosed by a square of size 50×50 m^2. The tags are randomly distributed and different tag density cases are considered: from sparse of 50 tags to dense of 200 tags in the workspace. The maximum number of available readers can be determined based on the coverage radius of each reader in the square gridded work area, and the maximum allowable overlap ratio of $r - r \times \cos(45°)/r = 47\%$.

We implement the HN-RNPO algorithm and compare it with the GA-based RNP optimization algorithm, GA-RNPO. Simulation parameters are described in Table 1.

Simulation evaluation metrics include:

- Coverage efficiency: is determined as the ratio of covered tags to total tags:
- Deployment efficiency: is the ratio of covered tags to deployed readers:
- Interrogation efficiency: is evaluated by arguing that the lower the number of tags in the overlapping area, the higher the interrogation efficiency. Therefore, this metric is the ratio of non-overlapping covered tags to covered tags:

$$\text{Interrogation efficiency} = \frac{\text{covered tags} - \text{overlapped tags}}{\text{covered tags}};$$

- Load balancing: is the balance of the number of tags interrogated per reader. That is, the number of tags interrogated by each reader (interrogated-per-reader

Table 1 Simulation parameters

Parameters	Value
Grid sizes	1.6 m
Population	8
Selection/Crossover operators	Roulette wheel/Single-point and two-point
Mutation operator (with probability)	One gen (0.05)
New population	30% elite parents, 70% best offspring
Weights of objective function and constraints	$w_1 = 0.4$; $w_2 = 0.2$, $w_3 = 0.2$ and $w_4 = 0.2$
Radius (r)	3.69 m
Stopping conditions	100 generations or the fitness unchanged over 5 generations

tags) should approach the average number of tags a reader is responsible for (average interrogated tags). Therefore, this metric can be determined by the square of the ratio of the difference between interrogated-per-reader tags and average interrogated tags to average interrogated tags:

$$\text{Load balancing} = \left(\frac{\text{interrogated-per-reader tags} - \text{average interrogated tags}}{\text{average interrogated tags}}\right)^2;$$

- Running time.

3.1 Coverage Efficiency and Deployment Efficiency

Figures 4 and 5 show the coverage efficiency and deployment efficiency of the two algorithms, HN-RNPO and GA-RNPO, when considering tag density and the number of tags varying from 50 to 200 tags. The curves show that the efficiency increases as the tag density increases, reflecting that the readers are used efficiently in covering the tags. For both evaluation metrics, HN-RNPO consistently outperforms GA-RNPO due to its ability to exceed local optima and find global optima.

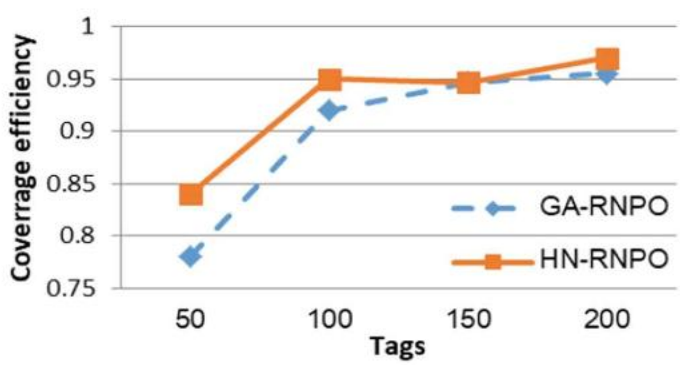
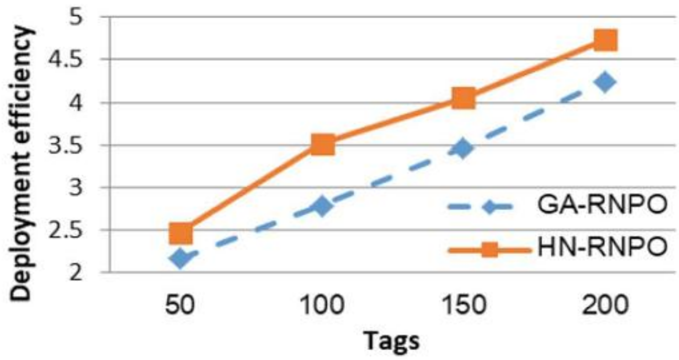


Fig. 4 A comparison between HN-RNPO and GA-RNPO on coverage efficiency

Fig. 5 A comparison between HN-RNPO and GA-RNPO on deployment efficiency

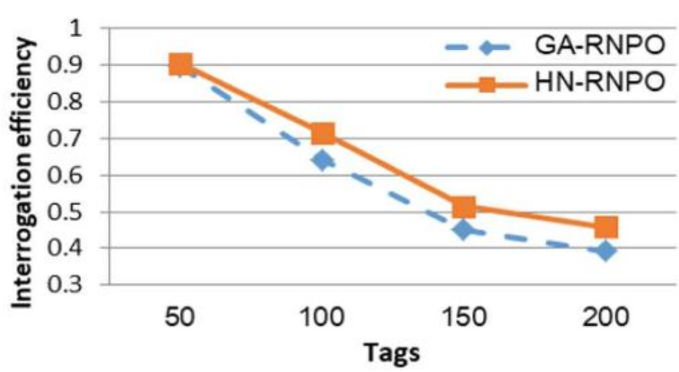


3.2 Interrogation Efficiency and Load Balancing

The coverage and deployment efficiencies increase as the number of tags increases, but increasing the density reduces the interrogation efficiency (Fig. 6). According to the metric of interrogation efficiency, the more the number of tags is in the overlapping area, the lower the interrogation efficiency is. As the tag density increases, the overlapping area inevitably increases when deploying readers, and the number of tags appearing in this overlapping area increases. Despite the reduction in interrogation efficiency, HN-RNPO consistently outperforms GA-RNPO thanks to fine-tuning the neural activation during optimization, avoiding readers being deployed too close together and causing large overlaps.

Increasing tag density also has a significant impact on load balancing among readers. When tag density is sparse, selecting a location to deploy each reader with load balancing is easy to achieve, but when tag density increases, it becomes more difficult to both reduce interference (number of tags in overlapping areas) and balance the number of tags in each reader's reading area. As a result, load balancing decreases as tag density increases. As shown in Fig. 7, HN-RNPO consistently outperforms GA-RNPO, thanks to the fine-tuning technique of the activated neurons during the optimization process when selecting a location to deploy a new reader.

Fig. 6 A comparison between HN-RNPO and GA-RNPO on interrogation efficiency



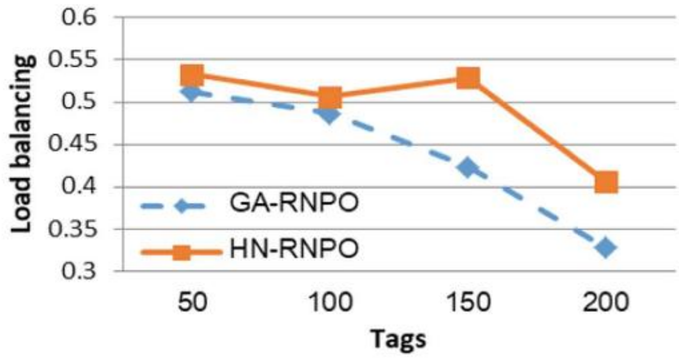


Fig. 7 A comparison between HN-RNPO and GA-RNPO on load balancing

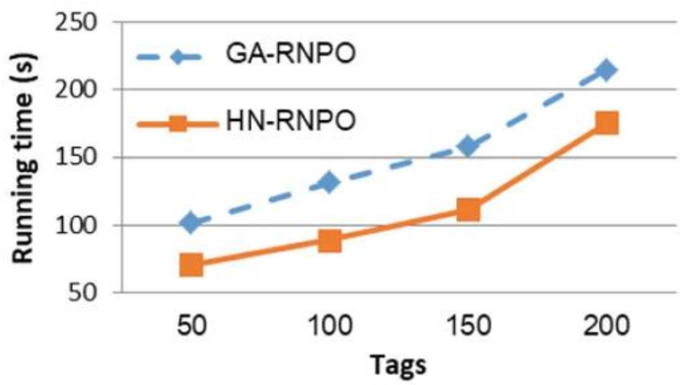


Fig. 8 A comparison between HN-RNPO and GA-RNPO on running time

3.3 Running Time

The running time of HN-RNPO is indeed more efficient than GA-RNPO. In fact, with the Hopfield network-based optimization approach, as described in Sect. 2.3, the connection weights between neurons and the activation threshold of each neuron are determined on the fly instead of having to go through a training period. When performing the optimization process, based on the determined weights and thresholds, the optimization process only minimizes the energy function by trying to deploy each reader according to the suggestion of the largest activated neuron. Thus, The algorithm helps find the optimal deployment solution and accelerates the convergence speed. As shown in Fig. 8, the running time of HN-RNPO keeps a significant gap with that of GA-RNPO when considering different tag densities.

4 Conclusion

RNP optimization is known to be an NP-hard problem, and natural-based methods are often used to find optimal solutions. Most found optimal solutions are local, rather than global, because the approach starts from some discrete populations and searches the entire candidate solution space. This approach requires much time to check all possible candidate solutions. The approach proposed in the paper is to formulate RNP as minimizing an energy function (error function) and use a Hopfield network to perform the optimization process. By pre-determining the connection weights and activation thresholds of neurons, the Hopfield network-based optimization converges faster, and the global optimization result is found sooner. Compared with GA, one of the most effective natural optimization methods, our proposed Hopfield network-based RNP optimization and improved neuron activation fine-tuning have shown remarkable advantages. However, the randomness of operators in the Hopfield-based method is still unavoidable, and this feature needs further investigation.

Acknowledgements This work was partially supported by Hue University under the Core Research Program, Grant No. NCTB.DHH.2025.09.

Disclosure of Interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Elbasani E, Siriporn P, Choi JS (2020) A survey on RFID in Industry 4.0. In: EAI/Springer innovations in communication and computing, pp 1–16. https://doi.org/10.1007/978-3-030-32530-5_1
2. Landaluce H, Arjona L, Perallos A, Falcone F, Angulo I, Muralter F (2020) A review of IoT sensing applications and challenges using RFID and wireless sensor networks. Sensors (Switzerland). 20:1–18. https://doi.org/10.3390/s20092495
3. Azizi A (2019) RFID network planning. Sensors 19–25. https://doi.org/10.3390/s21093138
4. Suriya A, David Porter J (2014) Genetic algorithm based approach for RFID network planning. In: TENCON 2014—2014 IEEE Region 10 conference. IEEE, USA, pp 1–5. https://doi.org/10.1109/TENCON.2014.7022427
5. Kong H, Yu B (2019) Modeling and optimization of RFID networks planning problem. Wirel Commun Mob Comput 2019:1–7. https://doi.org/10.1155/2019/2745160
6. Jaballah A, Meddeb A (2017) Self adaptive cuckoo search algorithm for RFID network planning. In: 2017 Internet technologies and applications (ITA). IEEE, USA, pp 122–127. https://doi.org/10.1109/ITECHA.2017.8101922
7. Tsai CY, Chang HT, Kuo RJ (2017) An ant colony based optimization for RFID reader deployment in theme parks under service level consideration. Tour Manag 58:1–14. https://doi.org/10.1016/j.tourman.2016.10.003
8. Chen H, Zhu Y, Hu K (2010) Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning. Appl Soft Comput J 10:539–547. https://doi.org/10.1016/j.asoc.2009.08.023
9. Shi W, Wang W, Yu Y, Zhang S, Cao Y, Yan S, Gao J (2021) Optimal deployment of phased array antennas for RFID network planning based on an improved chicken swarm optimization. IEEE Internet Things J 8:14572–14588. https://doi.org/10.1109/JIOT.2021.3067013
10. Feng H, Qi J (2012) Optimal RFID networks planning using a hybrid evolutionary algorithm and swarm intelligence with multi-community population structure. In: International conference on advanced communication technology, ICACT, pp 1063–1068
11. Gong Y-J, Shen M, Zhang J, Kaynak O, Chen W-N, Zhan Z-H (2012) Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant reader elimination. IEEE Trans Ind Inform 8:900–912. https://doi.org/10.1109/TII.2012.2205390

12. Sanagooy Aghdam A, Afshar Kazemi MA, Toloie Eshlaghy A (2001) A hybrid GA–SA multi-objective optimization and simulation for RFID network planning problem. J Appl Res Ind Eng 8:1–25. https://doi.org/10.22105/jarie.2021.295762.13571
13. Van Hoa L, Chuong DT, Quoc NH, Nhat VVM (2023) An integrated model of placement optimization and redundancy elimination in RFID network planning. In: Lecture notes in networks and systems (LNNS 734), pp 342–354. https://doi.org/10.1007/978-3-031-36886-8_29
14. Yixuan Q, Jiali Z, Xiaode X, Zihan L, Wencong L (2021) Hybrid gray wolf optimization-cuckoo search algorithm for RFID network planning. J China Univ Posts Telecommun 28:91–102. https://doi.org/10.19682/j.cnki.1005-8885.2021.1012
15. Maimouni M, Abou El Majd B, Bouya M (2022) RFID network planning using a new hybrid ANNs-based approach. Conn Sci 34:2265–2290. https://doi.org/10.1080/09540091.2022.2115011
16. Curty J, Declercq M, Dehollain C (2007) Design and optimization of passive UHF RFID systems
17. Atencia M, Joya G (2015) Hopfield networks: from optimization to adaptive control. In: Proceedings of international joint conference on neural networks. https://doi.org/10.1109/IJCNN.2015.7280368
18. Nhat VVM (2009) An application of neural networks in the connection admission control of ATM networks. In: 2009 IEEE-RIVF international conference on computing and communication technologies. IEEE, USA, pp 1–5. https://doi.org/10.1109/RIVF.2009.5174618