Analysis of gradient-based cuckoo search for the large scale optimal RFID network planning

K Hasnan, N H Talib and A Nawawi

Faculty of Mech. and Manuf. Eng., Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia

Abstract. Nature inspired algorithms provide an efficient method to resolve the problems of RFID Network Planning optimization that are not possible with the conventional methods. The gradient of the RFID Network Planning objective function was recently used to improve the precision of global optimal solutions. Therefore, this work presents a comparative study between the Gradient-Based Cuckoo Search (GBCS) and state-of-the-art algorithms in a large, complex, and dynamic RNP network. The outcome of this study analyzed the algorithms performance in terms of tag coverage, required number of readers, and interference between reader propagation areas. The present method specifies the combinatorial performance of the reader’s propagation area based on the evaluation of the tag density and location by using the Gradient-Based to manage the input representation of the Cuckoo Search. The results observed high local information for the objective function, which facilitated the choice of complex RFID Network Planning parameters that enabled the algorithm to work with big data in large scale conditions.

1. Introduction
Optimization is an applied science that presents the best values for selected parameters (Armin et al., 2014). Optimization aims to obtain the minimum or maximum values for relevant objective function parameters. The problem optimization design starts with the design of the objective function. Then, it expresses the relationship between the parameters of the problem based on mathematical formula (Civicioglu & Besdok, 2013). Swarm Intelligence (SI) presents the collective behavior of multi-agent systems. It is considered an efficient methodology for solving real-world problems (Yang, 2010). In recent years, a class of RFID Network Planning optimization based on SI algorithms has been developed, including Evolutionary Algorithms (EA) and Swarm Intelligence (SI) (Hasana et al., 2015). SI includes five different algorithms, namely Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Bacterial Foraging Optimization (BFO), and Firefly Algorithm (FA) (Elewe, 2016). Recently, cuckoo search algorithms have been applied in wide function optimization domains such as feature selection, engineering optimization, scheduling, planning, and real-world applications (Jr & Yang, 2014). The Cuckoo Search (CS) is a unique nature-inspired stochastic optimization approach. This method is getting popularity for finding the minimum global result of engineering applications and diverse science problems such as mobile-robot navigation problems (Vijaya Geeta Pentapalli & Kiran Varma, 2007). Fateen et al., in 2014 allowed the CS algorithm to apply the gradient information to improve the efficiency and reliability of the algorithm (Fateen & Bonilla-Petriciolet, 2014). GBCS proved to be a strong algorithm candidate for solving difficult optimization problems. Many researchers applied this algorithm in the RFID field. Jaballah and Meddeb in 2017 presented a Self-Adaptive Cuckoo Search (SACS) algorithm (Jaballah and Meddeb, 2017). The SACS algorithm is a helpful method to solve real RFID network planning
instances. The experimental results observed optimal solutions for the RFID network planning problem (Jaballah, 2017). This paper compares the state of the art PSO developed algorithm, known as the MC-GPSO algorithm, and a firefly algorithm presented by Hasana in 2017 with Gradient-Based Cuckoo Search (GBCS). This study will apply the RNP network planning objective functions based on a large-scale area to evaluate the effectiveness of each algorithm. The aim of this study is to find a proper algorithm to use it in railway stations network planning in order to employ the RFID system in a large and complex area.

2. Gradient-Based Cuckoo Search Algorithm

The aim of this section is to introduce the modified CS algorithm based on the gradient of the objective function. The modification of the algorithm did not change its stochastic nature to avoid negatively affecting its performance.

The CS operation was inspired by the obligate brood parasitism of cuckoo species, who lay their eggs in the nests of host birds. Each egg in a nest is considered as a solution, and a cuckoo egg is considered a new solution. The target is to employ the new better solution (cuckoo) to develop the not so good solutions in the nests. The SC rules are (Valian, Mohanna, & Tavakoli, 2011):

i. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
ii. The best nests with high quality solutions (i.e. eggs) will carry over to the next generation;
iii. Fixed the number of available host nests, the available host can discover an alien egg with a probability of \( \epsilon \in [0, 1] \).

In this condition, the host bird can be operate either throw the egg away or abandon the nest to generate a completely new nest in a new location. For simplicity, the last assumption in the process can be approximated as a fraction \( p_a \) of the \( n \) nests being changed by new nests, having new random solutions. Based on the present-mentioned rules, the essential steps of the CS can be summarized as the pseudo code in Figure 1 as follows (Valian et al., 2011):

Begin

Objective function \( f(x), X=(x_1,\ldots,x_d)^T \)

Generate initial population of

N number of host nests \( x_i (i=1,2,\ldots,n) \)

While (\( t< \text{Max Generation} \) or (stop criterion))

Get a cuckoo randomly by Lévy flights
Evaluate its quality/fitness \( F_i \)
Choose a nest among \( n \) randomly

If (\( F_i > F_j \)),
Replace \( j \) by the new solution;
End if

A fraction (\( p_a \)) of worse nests
Are abandoned and new ones are built;

Keep the optimal solutions
(Nests with quality solutions);
Rank the solutions and find the current best

End while

Figure 1. The CS pseudo code

The new solutions of \( x_i (t+1) \) for the \( i \)th cuckoo will apply the following Lévy flight formula:
\[ X_i(t + 1) = x_i(t) + \alpha \otimes \text{Lévy}(\lambda) \]  

(1)

Where \( \alpha > 0 \) is the step size that is related to the scale of the problem. In the present formula, the product (\( \otimes \)) means entry-wise multiplications. The Lévy flight is considered as step-lengths based on the following probability distribution:

\[ \text{Lévy } u = t^{-\lambda}, 1 < \lambda \leq 3 \]  

(2)

The real application of this algorithm observed the cuckoo’s behavior. It shows that if a cuckoo’s egg is similar to a host’s eggs, the random walk is less likely to discover the cuckoo’s egg. In this case, fitness should be related to the difference in solutions (Valian et al., 2011). The modification presented by Fateen and Petriciolet in 2014 used the local random walk based on the fraction (1-pa) of the replaced nests (Fateen & Bonilla-Petriciolet, 2014). A fraction (1-pa) of the nest selected at random is abandoned and changed by new ones at new positions via local random walks. The local random walk can be written as:

\[ x_{i+1} = x_i + \alpha (x_j - x_k) \]  

(3)

Where \( x_j \) and \( x_k \) are two different solutions generated and selected randomly by random permutation and the factor \( \alpha \) is a random number drawn from a uniform distribution. In the original algorithm, the value and the direction of the step are both random in walk depends on the new nests when they are generated from the replaced nests. The modification in the present algorithm, the researcher resaved the randomness of the magnitude of the step. However, direction is calculated based on the gradient sign of the objective function. When the gradient is negative, the direction of step will be positive. If the gradient is positive, the direction of step will be negative. Based on the present sequence, new nests will be generated randomly from the worse nests but in the direction of the minimum number of old nests. Thus, (Eq. 1) is replaced by

\[ \text{step}_i = \alpha (x_j^i - x_k^i), x_{i+1} = x_i + \text{step}_i \otimes \text{sign} \left( \frac{\text{step}_i}{df_i} \right) \]  

(4)

where sign function involves the sign of its argument and \( df_i \) is the objective function gradient at each variable, that is, \( \frac{df}{d_i} \) (Fateen & Bonilla-Petriciolet, 2014).

3. RNP Optimization Methods

RFID Network Planning (RNP) defined as a process of network-synthesis based on a RFID multi-objective. The target from RNP is to ensure that the network service can satisfy the needs of the subscriber and operator. RFID Network Planning optimization helps to maximize network structure performance. Therefore, RNP is enhanced by employing different algorithms. The Particle Swarm Optimization (PSO) algorithm is one of the state-of-art algorithms used in this field. PSO algorithm considered a population based stochastic optimization technique. This algorithm is inspired by the social behavior of birds and fish. PSO is a fast operation speed algorithm based on few parameters that need to be adjusted. To overcome Particle Swarm Optimization problems, the researchers suggested a new development for tuning the parameters and hybridized the algorithm. Chen et al. in 2011 presented the PS20 algorithm, which extends the single population and enhances PSO dynamic update equations. The researchers presented the idea of a reader collision problem. They observed a good optimization effect for weight and speed (Elewe, 2016). Gong et al. in 2012 improved the PSO algorithm by omitting readers through the search process with the Tentative Reader Elimination (TRE) operator. The proposed method improved overall RFID network performance for number of readers by minimizing the number of deployed readers correlated with interference and maximizing tag coverage (Elewe, Hasnan, & Nawawi; 2017). Feng and Qi (2013) developed a novel optimization algorithm using ellipse propagation patterns to solve the problems of multi-
objective nonlinear optimization of complicated large scale RFID network planning. The multi-community GA-PSO Algorithm uses the mutation strategy and genetic selection to improve particle swarm dynamic rules for RPN applications. The algorithm working process is to partition the single population of the canonical PSO into a multi-swarm (Feng & Qi, 2013). Gong et al. in 2013 used the adaptive small-world network model to develop a novel local topology method named (ASWPSO). This method interacts each swarm particle with its cohesive neighbors, and by chance communicates via small-world randomization with some distant particles. The ASWPSO method was tested using thirteen benchmark functions, and the results verified the robustness and high efficiency of the proposed adaptive small-world topology (Hasnan, 2017).

Nawawi et al. in 2015 developed PSO using Design of Experiment (DOE) analysis. The present method was able to specify the optimum setting of Particle Swarm parameters. They managed high quality results in overall RNP (Nawawi, 2015). Also, in 2015, Hasnun et al. developed a robust approach for RFID Network Planning using a Multi-Colony Global Particle Swarm Optimization (MC-GPSO) approach. The process in this algorithm is to participate the swarm into multi-colonies in order to detect the minimum number of readers and the minimum interference effect that covers all tags on a large-scale basis. This algorithm presented a robust and efficient RNP technique (Hasnan, 2015). Therefore, it was compared with the Firefly Algorithm (FA) in 2017 based on large scale RNP (Hasnan, 2017). All the presented algorithms were applied in a specific area that not exceed 80 meters square. But in real applications, the use of larger area such as in railway stations requires more efficient methods. This guides researchers to test the Gradient-Based Cuckoo Search (GBCS) Algorithm for this purpose.

4. PNR Mathematical Model
The Mathematic Model was developed under the assumption that the attached antenna to the RFID reader generates a circular coverage area. The objective function parameters of the RFID Network Planned (RNP) problem were identified to improve the solution quality. The values of these parameters are identified in Table 1.

| Parameter | Description |
|-----------|-------------|
| $P_r$     | Power input at receiving antenna |
| $P_t$     | Power output at transmitting antenna |
| $G_t$     | Transmitting antenna gain |
| $G_r$     | Receiving antenna gain |
| $\lambda$| Wavelength |
| $r_{max}$| Maximum propagation distance |
| $L_m$     | Path loss in meter |
| $L_{db}$  | Path loss in decibels |
| $c$       | Speed of light (299792458 m/s) |
| $f$       | Frequency (Hz) |
| $N_t$     | Total Tags |
| $\text{Coverage Rate of First Group of Tags}$ | $\text{Coverage Rate of Second Group of Tags}$ |
| $\text{Interference Range of the Readers}$ | $\text{Position of the}$ |

---

Table 1. Simulation Parameters

| Parameter | Description |
|-----------|-------------|
| $P_r$ | Power input at receiving antenna | Coordinate of tags |
| $P_t$ | Power output at transmitting antenna | Coordinate of reader |
| $G_t$ | Transmitting antenna gain | $r_{max}$ Maximum propagation distance |
| $G_r$ | Receiving antenna gain | $r_{td}$ Distance between tag and reader center |
| $\lambda$ | Wavelength | $Cov$ Coverage of tags group in TS range |
| $r_{max}$ | Maximum propagation distance | $N_i$ Readers number |
| $L_m$ | Path loss in meter | $t$ Tag |
| $L_{db}$ | Path loss in decibels | $RS$ The propagation domain |
| $c$ | Speed of light (299792458 m/s) | $TS$ The working domain |
| $f$ | Frequency (Hz) | $k$ Number of groups |
| $N_t$ | Total Tags | $Cov_i$ Coverage rate of first group of tags |
| $\text{Coverage Rate of First Group of Tags}$ | $\text{Coverage Rate of Second Group of Tags}$ |
| $\text{Interference Range of the Readers}$ | $\text{Position of the}$ |
The values of these parameters were identified as follows: RFID Reader System Operating Frequency UHF band was (915 MHz). RFID Reader adjustable Transmitting power range was [20; 33 Dbm] (i.e., 0.1 to 2 watts). Sensitivity thresholds of tags (Tt) was (-14dBm). Sensitivity thresholds of Readers (Tr) was (~70dBm). RFID Reader Antenna Gain (Gr) was (7.3 dBi). RFID Tag Antenna Gain (Gt) was (3.7 dBi) (Elewe et al., 2017). The task of RFID network planning is to employ RFID readers to cover the tags. The main parameter is the propagation range of the reader (rmax). The propagation range of the reader can be calculated from the set of equations below (Paper & Botero, 2016):

\[ L_{db} = P_t + G_t + G_r - P_r \]  
\[ L_m = 10^{(L_{db}/10)} \]  
\[ r_{max} = \frac{\sqrt{L_{m} \lambda}}{4\pi} \]  

The consideration is fixed as the length and width of the area that contain the distributed tags. The propagation range is calculated in equation 3.13 and the present calculations are subject to the boundary conditions (Elewe et al., 2017):

\[ r_{max} \geq r_{td} \]  

From the present formulas, the distance between reader and tag can be found from the formula below:

\[ r_{td} = \sqrt{(x - xt)^2 - (y - yt)^2} \]  

In order to calculate the required number of readers and the position of each reader initially the following equations are used (Elewe et al., 2017):

\[ Covi = \sum_{t=RS}^{max}(r_{max} - r_{td}) \]  
\[ N_i = \sum_{t=RS}^{k} Cov_i \]  

The rate of tags coverage in the specific network, which represents a very important and basic objective function of all RFID systems is defined as:

\[ f(Cov) = \frac{\text{detected tags}}{\text{total tags}} = \sum_{t=RS}^{\max} \frac{Covi}{N_t} \]  
\[ Cov = \begin{cases} 1 & \text{if } r_{max} \geq r_{td} \\ 0 & \text{otherwise} \end{cases} \]  

The interference between RFID readers can be solved by the following mathematical equation (Hasnan et al., 2015):
where dist(.) is function that computes the distance between readers. The interference rate can be calculated by the formula:

\[ f(l) = \frac{\text{Interfered Tags}}{\text{TotalTags}} = \frac{\sum_{i,j} \text{Cov}_i \cap \text{Cov}_j}{N_t} \] (14)

Based on the present set of equations, a RNP performance comparison was made in different RNP environments in order to investigate the algorithms reliability in large scale conditions.

5. Simulation results

The experimental results of the Cuckoo algorithm observe a good solution for large scale areas (80m* 80m). A comparison with MC-GPSO and FA algorithms was done based on the solution quality of the maximum tag coverage, minimum interference between readers, and required number of readers. The scenarios of the 50 and 100 tags dataset were taken from the comparative study presented by (Hasnan, 2017). Figure 2 show the solution quality for RNP large scale area.

![Figure 2. GBCS algorithm simulation results, (a)50 tags, (b) 100 tags](image)

It was seen from the results that the GBCS algorithm observes robust solutions due to the required number of readers, especially in large data. In case of 100 tags, it is used 6 readers while the firefly algorithm used 8 readers. Table 2 present the numerical results of the three algorithms.

| No. of readers | Interference | Tags Coverage |
|----------------|--------------|---------------|
| 50 tags |
| GBCS          | 7            | 0.0015        | 92%           |
| FA            | 8            | 0.002         | 98%           |
| MC-GPSO       | 9            | 0.053         | 95.09%        |
| 100 tags |
| GBCS          | 6            | 0.034         | 93%           |
They indicate that RFID reader positions were distributed in order to cover all the tags denoted as plus signs “+”. The coordinates of the readers are shown as red stars “∗”, and their interrogation range as a red circle of dashed lines. The algorithm was able to employ 7 readers for 50 tags cases. That indicates to significant reliability of the GBCS algorithm in large scale area with big tags data. Comparative analysis observes the weakness in tag coverage. The GBCS algorithm was able to achieve 92% coverage for 50 tags, while FA covered 98%. It can be seen that the firefly algorithm and MC-GPSO were better than the GBCS algorithm in tag coverage. The case of 100 tags present 93% coverage, while the MC-GPSO covered 94.68%.

| Algorithm | Readers | Coverage |
|-----------|---------|----------|
| FA        | 8       | 97%      |
| MC-GPSO   | 10      | 94.68%   |

![Figure 3. GBCS simulation results for cluster data of 50 tags](image)

The same indicator can be seen. This fact guides to new test procedure in order to investigate the algorithm effect based on clustered tag conditions using the same number of tags (i.e. 50 and 100). The results shown in Figure 3 and 4 observe a high performance algorithm.
Figure 4. GBCS simulation results for cluster data of 100 tags

It can be seen from the results that the GBCS algorithm covered all the tags using six readers for 100 tags and only five readers for 50 tags. Table 3 presents the numerical results of the GBCS algorithms. These results indicate that this algorithm can be used efficiently in large scale conditions with clustered big data. Therefore, it can be a useful solution for RFID network planning in railway stations.

The PSO and FA algorithms are working effectively in random data due to use either uniform distributions or Gaussian to generate new explorative moves. Also, they use exhaustive tuning; these two features make them working efficiently with random data. While the GBCS algorithm used L’evy flights to find the nests or solution which is difficult to find in random data. But in large clustered data, the GBCS observe better performance because the ease of creating the nests and keeping the best nests or solutions.

| Tags    | No. of readers | Interference | Coverage |
|---------|----------------|--------------|----------|
| 50 tags | GBCS           | 5            | 0.00     | 100%     |
| 100 tags| GBCS           | 6            | 0.017    | 100%     |

6. Conclusion

GBCS algorithm is used to solve the continuous problems of RFID network planning tools based on objective functions. The results were compared with the best state of the art algorithms based on large scale conditions. The test results observed an efficient reader distribution, but there were weaknesses in tag coverage when the tags were randomly distributed. Therefore, the algorithm was tested in clustered tag
positions, which observed superior results. The optimization results indicate a reduction in required readers and the high reliability of this method for complex tags clustered in a large scale area. This results obtained due to enhancement of input representation features due to the Gradient-Based. This method specify the combinatorial performance of the objective function in the specified RNP network. Therefore, its enable the Cuckoo Search algorithm to deal with the large and complex area.

7. REFERENCES
[1] Armin, F., Geibler, B., Gollmer, R., Hayn, C., Henrion, R., Hiller, B., … Schmidt, M. (2014). Mathematical optimization for challenging network planning problems in unbundled liberalized gas markets. Energy Systems, 5(March 2013), 449–473.
[2] Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artificial Intelligence Review, 39(4), 315–346. https://doi.org/10.1007/s10462-011-9276-0
[3] Elewe, A. M. (2016). Review of rfid optimal tag coverage algorithms, 11(12), 7706–7711.
[4] Elewe, A. M., Hasnan, K. Bin, & Nawawi, A. Bin. (2017). HYBRIDIZED FIREFLY ALGORITHM FOR
[5] Fateen, S. E. K., & Bonilla-Petriciolet, A. (2014). Gradient-based cuckoo search for global optimization. Mathematical Problems in Engineering, 2014. https://doi.org/10.1155/2014/493740
[6] Feng, H., & Qi, J. (2013). Planning Using a New Hybrid Evolutionary Algorithm, 2(1), 179–188.
[7] Hasnan, K. (2017). Comparative evaluation of Firefly Algorithm and MC-GPSO for Optimal RFID Network Planning, 2017.
[8] Hasnan, K., Ahmed, A., Bakhsh, Q., Hussain, K., Latif, K., Tun, U., … Pahat, B. (2015). A Novel Optimal RFID Network Planning by MC-GPSO, 8(Augus
[9] Jaballah, A. (2017). Self Adaptive Cuckoo Search Algorithm for RFID Network Planning, 122–127.
[10] Jaballah, A., & Meddeb, A. (2017). A new variant of cuckoo search algorithm with self adaptive parameters to solve complex RFID network planning problem. Wireless Networks. https://doi.org/10.1007/s11276-017-1616-9
[11] Jr, I. F., & Yang, X. (n.d.). Cuckoo Search: A Brief Literature Review, 1–13.
[12] Nawawi, A. B. I. N., Access, F., Permai, T., & Pahat, B. (2015). UNIVERSITI TUN HUSSEIN ONN MALAYSIA STATUS CONFIRMATION FOR DOCTORAL THESIS A MODIFIED TECHNIQUE IN RFID NETWORK PLANNING AND OPTIMIZATION ACADEMIC SESSION : 2014 / 2015 ii This thesis has been examined on date 20 th August 2014 and is sufficient in fulfi.
[13] Paper, C., & Botero, O. (2016). RFID network topology design based on Genetic Algorithms RFID network topology design based on Genetic Algorithms, (October 2011). https://doi.org/10.1109/RFID-TA.2011.6068653
[14] Armin, F., Geibler, B., Gollmer, R., Hayn, C., Henrion, R., Hiller, B., … Schmidt, M. (2014). Mathematical optimization for challenging network planning problems in unbundled liberalized gas markets. Energy Systems, 5(March 2013), 449–473.
[15] Vijaya Geeta Pentapalli, V., & Kiran Varma, R. P. (2007). Cuckoo Search Optimization and its Applications: A Review. International Journal of Advanced Research in Computer and Communication Engineering ISO, 3297(11), 556–562. https://doi.org/10.17148/IJARCE.2016.511119
[16] Yang, X. (2010). Nature-Inspired Metaheuristic Algorithms Second Edition. 834–840