UACEE International Journal of Advances in Computer Networks and its Security – IJCNS
Volume 3 : Issue 2 [ISSN 2250 – 3757]
Publication Date : 05 June 2013

# Genetic Algorithm for Reader Network Planning Problem

Hsing-Fang Tsai
Department of Electrical Engineering
Chang Gung University
Taoyuan, Taiwan
m0021014@stmail.cgu.edu.tw

Shin-Yeu Lin
Department of Electrical Engineering &
Green technology Research Center
Chang Gung University
Taoyuan, Taiwan
shinylin@mail.cgu.edu.tw

***Abstract*—Reader network planning *(RNP)* problem of radio frequency identification *(RFID)* system is a combinational optimization problem. In this study, we propose a genetic algorithm *(GA)* to solve this RNP problem. We have tested the proposed GA on several RNP problems and compare with a particle swarm optimization *(PSO)* method by solving the same RNPs. The comparison results demonstrate that the proposed GA outperforms the PSO method.**

***Keywords*—radio frequency identification *(RFID);* RFID network planning *(RNP);* genetic algorithm *(GA);* particle swarm optimization *(PSO)***

## I. Introduction

In recent years, radio frequency identification (RFID) that identifies tagged objects automatically has been widely used in various applications such as warehouse, shopping center, location tracking and healthcare etc. [1-2]. The technology is in the ascendant with a concerted effort to make a further development. RFID systems consist of two types of devices. One is the tag containing a unique identifier and the information associated with the object, and the other is the reader which periodically collecting information from tags. The area to identify RFID devices is called the interrogation zone. Due to the limited interrogation range of the communication between the reader and the tag, the deployment of minimum number of readers to cover all tags in the entire region is known as the RFID network planning (RNP) problem. In addition to tag coverage, the RNP also has to maintain the Quality of Service (QoS). The QoS of RFID arises from the overlapping of interrogation zone of neighboring readers, which may cause the reader collision problem. In general, the RNP problem is a type of resource allocation problems, which is a combinational optimization problem. In this paper, we propose a genetic algorithm (GA) to solve it.

We organized this paper as follows. We present the formulation of the RNP problem in Section $II$. The proposed genetic algorithm will be presented in Section $III$. The test results and the comparison with a particle swarm optimization method are presented in Section $IV$. Finally, we draw a conclusion in Section $V$.

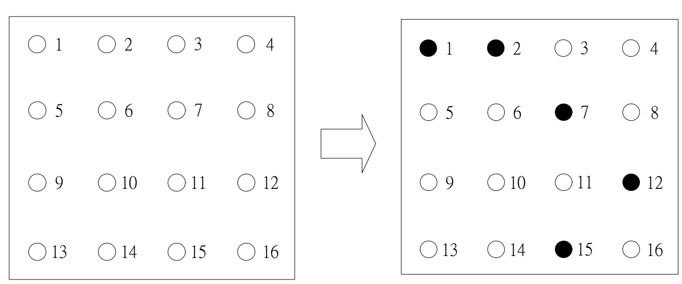
## II. Problem Formulation

### A. *Preliminaries*

We assume the space for placing the tags is rectangular with candidate fixed reader positions, which are designed to covering the complete space. In the sequel, we call the candidate fixed reader position as the candidate reader position (CRP).

Fig. 1(a) presents a simple example of the considered space and sixteen CRPs. The RNP problem is to find the minimum number of CRPs, such that placing readers at those selected CRPs as shown in Fig. 1(b) will cover all tags resided in the space and maintain the QoS. Therefore to formulate the RNP problem, we need to define the terms regarding tag coverage, selected number of CRPs and QoS. The following notations will be used in defining the above mentioned three terms.

$n$ : Number of configurations.
TNT : Total number of tags.
NTC : Number of tags been covered.
TNR : Total number of candidate readers.
OT : Overlapping tags.
UR : Total number of used readers.




(a) (b)

Figure 1. The pattern of space (○ : CRP; ● : selected CRP).

Volume 3 : Issue 2 [ISSN 2250 – 3757]
Publication Date : 05 June 2013

## B. *Tag Coverage*

For a given solution or configuration $n$, we will use the coverage ratio to indicate the degree of its tag coverage. The coverage ratio denoted by $\alpha_n$ is defined as the number of tags been covered divided by the total number of tags, which can be expressed as follows:

$$\alpha_n = \frac{NTC}{TNT} \tag{1}$$

## C. *QoS*

To maintain the QoS, we should avoid reader collisions which arise from the tags located on the overlapping area between neighboring readers' interrogation zones. In other words, we should increase the ratio of non-overlapping tags denoted by $\beta_n$, which can be formulated as follows:

$$\beta_n = \frac{TNT - OT}{TNT} \tag{2}$$

## D. *Cost Efficiency*

Since the cost efficiency of RNP denoted by $\gamma_n$ is inversely proportional to the number of placed readers, we can formulate $\gamma_n$ as follows:

$$\gamma_n = \frac{TNR - UR}{TNR} \tag{3}$$

## E. *Fitness Function*

Any solution or configuration $n$ of the RNP problem is associated with a fitness value denoted by $Fit_n$ that reflects how good it is, comparing with other solutions. We define the fitness of a solution by combining the above three terms linearly as follows:

$$Fit_n = p_1\alpha_n + p_2\beta_n + p_3\gamma_n \tag{4}$$

where the weighting coefficients $p_1$, $p_2$, $p_3$ must satisfy $p_1 + p_2 + p_3 = 1$ and can be varied to account for any specific consideration.

## F. *RNP problem*

Now, we can state the RNP problem in the following:

*The RNP problem is to find a reader placement configuration among all possible ones with largest fitness value.*

# III. Proposed Genetic Algorithm

Using GA to solve this RNP problem, we assume each CRP represents a gene, and a solution of reader configuration represents a chromosome. We employ a binary code to indicate whether a CRP is selected or not, such that 1 represents that the CRP is selected and 0 otherwise. An example chromosome can be formed by starting from the first CRP of the first row in Fig. 1(a), place them as the genes of the chromosome from left to right as shown in Fig. 2. We encode the genes, such that 1 represent that the corresponding CRP is selected and 0 otherwise. $N$ in Fig. 2 represents the total number of CRPs.

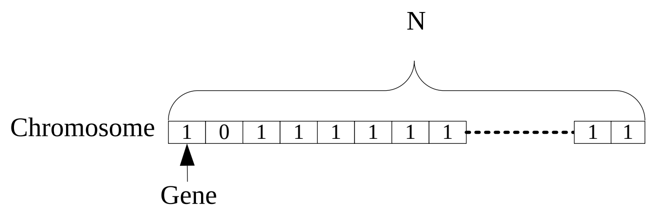



Figure 2. Illustration of GA structure.

The solution process of GA is presented in Fig. 3. We start from initializing the parameters of GA, determine the CRP and randomly assign 0 or 1 to the CRP and form a chromosome. Such a chromosome formation is repeated for $|P|$ times to from the initial population, where $|P|$ denotes the size of the population. We then proceed with the operations of selection, crossover, and mutation. The termination condition is defined based on the fitness value. If the fitness value does not improve, we stop; otherwise we proceed with next generation.

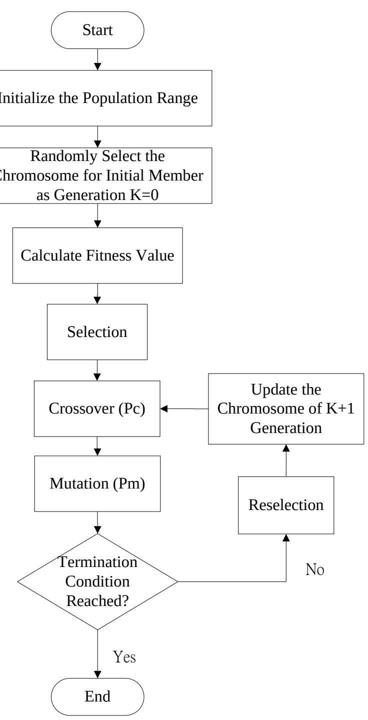



Figure 3. The flowchart of GA.

The three basic operators, selection, crossover and mutation, used by GA are stated in the following:

### A. *Roulette Wheel Selection Method*

Chromosomes with larger fitness occupies larger fraction of roulette wheel whereas chromosomes with smaller fitness occupies smaller piece. Each time, roulette wheel pointer points one chromosome, and that chromosome is placed in the mating pool. A chromosome with a larger fitness is likely to receive more copies than a chromosome with a smaller fitness.

### B. *One point crossover*

The crossover produces new individuals by combining the information contained in the parent chromosomes. Good results can be obtained with a random matching of the individuals. When the random number generated based on a uniform distribution over [0,1] interval is lower than the crossover probability ( $P_c$ ), we execute crossover. The operation of one point crossover can be described as follows. Firstly, one crossover position is randomly selected then the genes situated after this point are exchanged between the parents and the resulting chromosomes and called offspring. The process is illustrated in Fig. 4. For each pair of parents we generate two offspring, which replace their parents in the population.

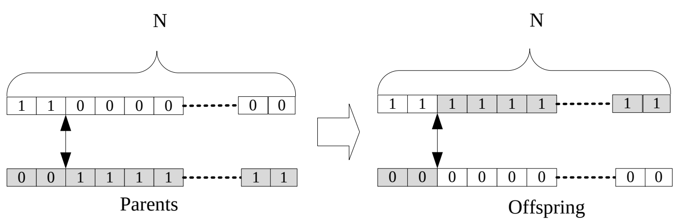



Figure 4. One point crossover (arrow points the crossover point).

### C. *Mutation*

In the later stages of a run, the population may converge in wrong direction and stuck to local optima. Mutation is used to explore new solutions in the search space can be employed to overcome this drawback. The operation of mutation can be stated in the following. For each gene of a chromosome, if the random number generated based on an uniform distribution over [0,1] interval is lower than the mutation probability ( $P_m$ ), the code of the gene will change from 1 to 0 or from 0 to 1. The process is illustrated in Fig 5.

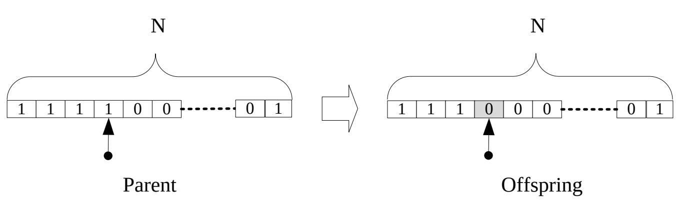



Figure 5. Mutation at the mutation points(be pointed by arrow).

## IV. Simulation Results

We deploy a square space with $N^2$ CRPs, when $N$ is the total number of CRP in one row as shown in Fig. 6. We consider $N$ =7, 8, 9, and10 for cases, in which 441, 576, 729, and 900 tags are randomly placed, respectively. We apply the proposed GA to the corresponding RNP of each case.

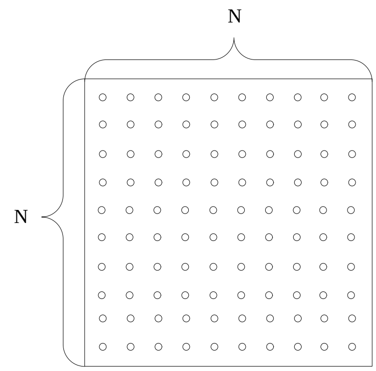



Figure 6. The deploy status of simulation space(N=10).

The parameters used by the proposed GA are presented in Table 1.

TABLE I. PARAMETER USED.

| Parameter | Value |
|---|---|
| Number of Populations | 40 |
| Length of Chromosome | $N^2$ |
| Crossover Probability | 0.5 |
| Mutation Probability | 0.1 |

We set the weighting coefficient of the fitness function (4) to be $p_1 = 0.7$ $p_2 = 0.15$ $p_3 = 0.15$. The progression of the obtained best-so-far fitness with respect to the CPU time consumed by the proposed GA is presented in Figs 7-10., marked by the square points.

We also used the Particle Swarm Optimization (PSO) method [10], which is inspired by the nature phenomenon as fish schooling or birds flocking, to solve the same RNP problems. We also present the test results obtained by PSO method in Figs 7-10.

From these figures, we found that in case 1,when GA obtain the best-so-far fitness value 0.7929 at time 5.79s shown in Fig. 7, the best-so-far fitness value obtained by PSO method is just 0.7517. Similarly, in case 2,we found that when GA obtain the best-so-far fitness value 0.7916 at time 7.65s shown in Fig. 8, the best-so-far fitness value obtained by PSO method is just 0.7815. In case 3, we found that when GA obtain the best-so-far fitness value 0.7907 at time 22.89s shown in Fig. 9, the best-so-far fitness value obtained by PSO method is just 0.7775. In case 4, we found that when GA obtain the best-so-far fitness value 0.79 at time 43.3s shown in Fig. 10, the best-so-far fitness value obtained by PSO method is just 0.7792. The above comparisons demonstrate that the proposed GA outperforms the PSO method.

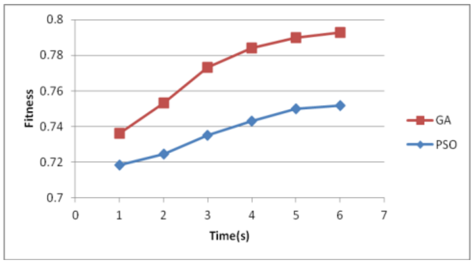



Figure 7. Compare GA with PSO in case 1 of 49 CRPs.

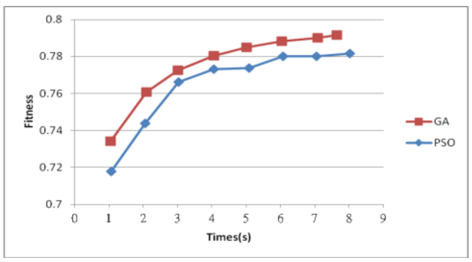



Figure 8. Compare GA with PSO in case 2 of 64 CRPs.

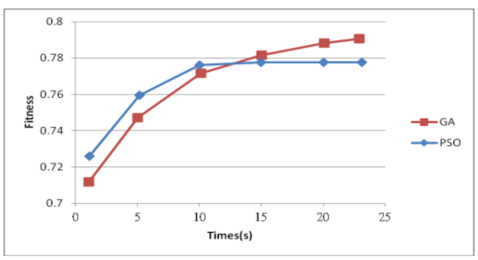



Figure 9. Compare GA with PSO in case 3 of 81 CRPs.

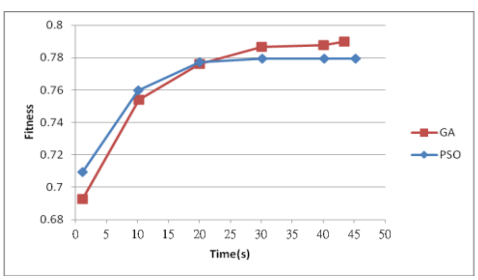



Figure 10. Compare GA with PSO in case 4 of 100 CRPs.

# V. Conclusions

In this paper, we have presented a GA to solve the RNP problems, and tested the propose GA on several RNP problems. We also use the PSO method to solve the same RNP problems, and the test results reveal that the proposed GA outperforms PSO method.

## *Acknowledgment*

This research works in supported by Chang Gung University in Taiwan.

## *References*

[1] E. Bottani and A. Rizzi, "Economical assessment of the impact of RFID technology and EPC system on the fast-moving consumer goods supply chain," Int. J. Production Economics vol.112(2), pp.548–569, April 2008.

[2] A. Oztekin, F. M. Pajouh, D. Delen, L. K. Swim, "An RFID network design methodology for asset tracking in healthcare," Decision Support Systems, vol. 49 pp.100–109, 2010.

[3] B.Carbunar, M.K.Ramanathant, M.Koyuturk, C.Hoffmannt, and A.Grama, "Redundant reader elimination in RFID systems," in Proceeding of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad hoc Communications and Networks, pp. 176–184, September 2005.

[4] Beasley, D. Bull, D. R., and Martin, R. R.,"An overview of genetic algorithms: Part 1,"Fundamentals, University Computing, vol.15(2), 58–69, 1993.

[5] Beasley, D. Bull, D. R., and Martin, R. R.,"An overview of genetic algorithms: Part 2," Research Topics, University Computing, vol.15(4), pp. 170–181, 1993.

[6] L. Guessas and K. Benmahammed, "Adaptive backstepping and PID optimized by genetic algorithm in control of chaotic," International Journal of Innovative Computing, Information and Control, vol. 7(9), pp. 5299–5312, Sept. 2011.

[7] S. M. Sait and H. Youssef, Iterative computer algorithms with applications in engineering: solving combinatorial optimization problem. Los Alamitos, CA: IEEE Comput. Soc., 1999.

[8] B.Carbunar, M.K.Ramanathant, M.Koyuturk, C.Hoffmannt, and A.Grama, "Redundant reader elimination in RFID systems," in Proceeding of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad hoc Communications and Networks (SECON '05), pp. 176–184, September 2005..

[9] Muhammad Amiad Iqbal, Naveed Kazim Khan, Hasan Mujtaba and A. Rauf Baig, "A novel function optimization approach using opposition based genetic algorithm with gene excitation," International Journal of Innovative Computing, Information and Control, vol. 7, no. 7(B), pp. 4263–4276, July 2011.

[10] Kennedy, J. and Bur. of Labor Stat.,"Particle Swarm Optimizer," IEEE International Conference on Neural Network, pp. 1942–1948 ,1995.