RFID network topology design based on Genetic Algorithms

Oscar Javier Botero Gomez, Chaouchi Hakima

RFID network topology design based on Genetic Algorithms

Oscar Botero and Hakima Chaouchi
CNRS SAMOVAR, UMR 5157. Telecom SudParis, Evry, France
{oscar.botero, hakima.chaouchi}@it-sudparis.eu

Abstract— **Radio Frequency Identification (RFID) is a well known technology that has entered successfully in the realm of innumerable applications and is considered as one of the principal building blocks for the realization of the Internet of Things concept. However, the majority of RFID applications require the utilization of multiple RFID readers and therefore effective and efficient planning of their networks is a major concern. Network planning is a complex process that involves different steps, one of which is its topology design. In this paper, we focus on the topology design process of a RFID network following an optimization-based approach. More precisely, we propose a multi-objective cost function that is used to evaluate candidate solutions for the position and power levels of the set of RFID readers to be deployed. In order to obtain optimal solutions we applied Genetic Algorithms, a well known heuristic technique. Finally, a developed software tool to assist in the topology design is also presented and processing delay measurements are performed.**

Keywords-Genetic Algorithms; Radio Frequency Identification; Network Planning; Network Topology Design.

I. INTRODUCTION

RFID Technology is one of the leading enabling building blocks of the Internet of Things and it is widely used for tracking and tracing items in production chains and stores as well as for different innovative applications such as citizen's documents, surveillance, public transportation, and building access control [1]. For certain applications i.e. warehouse items tracking, interactive museums, tracing products in supermarkets, among others; it is required to deploy a network of RFID readers in a cost-effective way. This can be achieved by following a network planning procedure. Network planning is an iterative process that involves several steps such as topological design, network synthesis and network realization [2]. This process can be based on different objectives such as coverage area, interference reduction, number of tags detected and low deployment cost. Apparently, for optimally satisfying all these - often opposed - objectives, a non-linear optimization problem with a large search space needs to be solved [8].

In this paper we focused on the topological design of an RFID network in order to obtain the optimal placement and power levels of the RFID readers that are to be deployed. Due to the nature of the problem: a search space with a large number of candidate solutions and different valid local optima, this problem suits to be solved by a well know heuristic technique, Genetic Algorithms (GAs) [3]. GA method is a search technique applied when dealing with huge search spaces and when multiple objectives should be achieved in parallel. GAs are used to obtain optimal solutions for a broad number of problems and are inspired in biological observations linked to the evolution of the organisms.

Consequently, in this work a GAs-based RFID network topology design is proposed. It allows us to obtain RFID readers' location and power levels relying on a multi-objective cost or fitness function that evaluates and rates the solutions provided. In addition, in order to add design flexibility capabilities, we developed a software tool that permits us to vary different parameters such that solutions for diverse test scenarios can be easily proposed.

The rest of the paper is organized as follows. In section II related work is given. Section III provides a brief background on RFID and GA. In section IV the anatomy of the network topology design process is depicted. In section V we describe the software tool developed to assist the topology design. In section VI numerical results related to processing delay measurements are provided and finally the paper concludes in section VII.

II. RELATED WORK

A variety of RFID network issues have been proposed to be solved by using evolutionary techniques including GA. In [4] a new technique for RFID resources allocation based on GA was proposed to deal with the reader-to-reader interference problem. In [5, 6 and 7] GA and binary particle swarm optimization are used to solve multi RFID networks scheduling problem to optimize channels allocation in the system. More related to our work are [8] and [9] where GA are used to propose RFID planning methods.

In our approach the candidate solutions for the required topology design are evaluated by using a linear weighted multi-objective function that considers six parameters: overlapping of the RFID reading area, number of useless readers deployed, number of tags covered, number of readers deployed out of the area to be covered, number of redundant readers deployed and number of tags located inside overlapped reading areas. We also developed a flexible software tool to assist in the topology design process that provides customizable GA parameters, functions to import and export scenarios and parameters settings, a scenario editor and visualization of the solving process and solutions obtained.

III. BACKGROUND

This section provides a brief overview regarding RFID networks and GA.

A. RFID Technology

RFID stands for radio-frequency identification and it is mainly used for tracking and tracing objects, animals or persons. Its major advantage over its predecessor; the barcode is that the identification is stored electronically and it can be retrieved wirelessly via an interrogator or reader with no line of

sight requirement [10]. The basic hardware implementation of this technology (Fig. 1) is built upon three elements: *transponders* or *tags* that store an identifier (ID) related to a specific object, *readers* or *interrogators* who obtain the ID stored in the tags, and the RFID middleware that performs processing of collected data following a certain purpose. The most used frequencies are 860–960 MHz named Ultra High Frequency band (UHF) used also used by the Electronic Product Code (EPC) Gen II/ISO 18000–6c standards and 13.56 MHz named High Frequency band (HF) used by ISO 18000-3 standard [11].

Fig. 1. Basic RFID architecture.

A RFID network requires the interaction of multiple readers and a planning strategy should be followed in order to reduce interference problems, maximize the number of tags covered, reduce the number of deployed readers and reduce the cost of the network.

B. *Genetic Algorithms (GAs)*

GAs were initially proposed in the 60's by John Holland [3] and at the beginning they were applied to study natural adaptation and how to bring them into computer models. In fact, they try to emulate as an abstraction what happens in nature regarding how organisms reproduce and survive based on evolution. GAs process populations constituted of several *individuals*. An individual is conformed of one or more *chromosomes*. The chromosome is the set of *genes* that encode particular properties of the individual. The genes can have different possible values called *alleles*. The genes are located in a particular *locus* or position inside the chromosome. In GAs the chromosome refers to a candidate solution to a problem. Generally, a chromosome is represented as a binary string that encodes it. The genes are bits whose alleles can be either one or zero. However, there are different representations for chromosomes that include real numbers, alphabet characters, etc.

GAs operators mimic what nature does regarding organisms' evolution. The main operators are: *Selection*, *Crossover* and *Mutation*. Selection obtains the fittest individuals (best performing solution) in order to make them exchange their genetic information to produce a new *offspring*. Crossover performs the combination of the selected individuals to produce new solutions and finally, mutation flips alleles to add variety and reduce local optima convergence. GAs search through a huge number of possible solutions where every chromosome represents one point in the space search. The solutions are evaluated based on a fitness function that provides a measurement which indicates how well the individual solves the current problem. A simple GAs workflow is as follows:

1. A random population is generated.
2. The population is evaluated by using a fitness function.
3. Through *Selection*, *Crossover* and *Mutation* a new population is created which is called next generation that is actually a next iteration.
4. Repeat from step 2 until a solution is obtained or other limitations are reached (execution time, memory capacity, number of generations, etc.).

Other search methods like Hill Climbing, Simulated Annealing and tabu search as well as GAs share the same principle that is they generate a set of candidate solutions, evaluate them according to a fitness function, discard or keep solutions based on fitness and produce variants of the solutions. What differentiates GAs from the rest is that they use stochastic selection, crossover and mutation at the same time.

There is no rigorous answer to the question: why and when to use GAs over other search methods. Intuitively, the applicability of GAs suits when the search space is "large" and the problem does not require a global optimum to be obtained. GAs are also prone to noisy fitness evaluation (involving error measurement of real world processes) because they accumulate fitness statistics over many generations and behave robustly in the presence of small amounts of noise. The success of GAs also depends on the way the problem is coded, the operator functions, the parameters and the evaluation function.

IV. RFID Network Topology Modeling

A. *Overview*

As we stated previously, we aim to obtain the optimal position and power level of the RFID readers to be deployed by following a certain criteria. The main goal of the topology design solution is that it needs to minimize the interference effect, maximize coverage that will be directly translated into a maximum number of tags detected as well as to reduce network cost by discarding redundancy and useless readers' deployment. In order to rate candidate solutions, we define a multi-objective fitness function that considers different factors that impact the RFID network topology design and that are described in Section B. Furthermore, we chose a two-dimension squared area as our test region and model the RFID reader coverage as a circular area with the reader located at the center. We assume RFID passive tags to be deployed. In the following section the multi-objective fitness function used to evaluate the candidate solutions is described.

B. *Multi-objective fitness function*

A linear weighted fitness function (f_T) to evaluate each possible solution is defined (Equation (1)). It provides a measureable indication on how well each evaluated individual performs at solving the current problem. The coefficients (w_i) can be set empirically, stressing different simulation requirements for giving more importance to some objectives than others. This function is composed of the following six objectives:

$$f_T = \sum_{i=1}^{6} w_i * f_i \ , \ \text{where:} \ \sum_{i=1}^{6} w_i = 1 \qquad (1)$$

1) Overlapping of the reading area

The tags located inside the reading area will be normally detected but if any other reader interferes there might be collisions (Fig.2). Thus, in order select only solutions with reduced reading area interference, we measured the overlapped area caused by all the deployed readers and we defined an acceptance threshold. Equations (2) to (5) show the definition of our first objective function (f_1).

$$f_1=\frac{1}{1+\varepsilon^2} \quad (2)$$

$$\varepsilon=\text{Target-Total OverlappingArea} \quad (3)$$

$$\text{Target}=\sum_{\text{all readers}} \text{CArea*CRatio} \quad (4)$$

$$\text{Total Overlapping Area}=\sum_{\text{all readers}} \text{Overlapping Area} \quad (5)$$

The Coverage ratio (CRatio) specifies the amount of overlapping allowed on each reader. We observe that values from 0.10 to 0.25 require less iterations in order to obtain a suitable solution. If Cratio is equal to zero, the GAs will take more time to obtain a total no-overlapping deployment layout. This implies that the number of readers required will increase since enough readers need to be packed inside a square perimeter to cover the entire area [12].

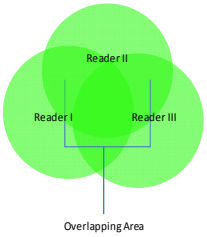

Fig. 2. Overlapping of the reading area.

2) Number of useless readers

It is to notice that even if a layout complies with the overlapping parameter it is possible to obtain readers that cover no tags and they need to be discarded from the optimal solutions (Fig. 3). Therefore, we measure and minimize the number of useless readers (Equations (6) to (8)).

$$f_2=\frac{1}{1+\varepsilon^2} \quad (6)$$

$$\varepsilon=\text{Target-Total Useless Readers} \quad (7)$$

$$\text{Target}=0 \quad (8)$$

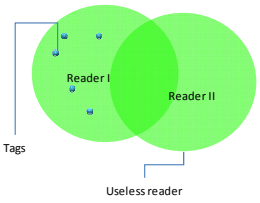

Fig. 3. Useless Reader.

3) Number of tags covered

One of the main objectives of a RFID deployment is to be able to detect and obtain the IDs of the totality of the deployed tags. We defined a target of 100% of tags detected (Equations (9) to (11)).

$$f_3=\frac{1}{1+\varepsilon^2} \quad (9)$$

$$\varepsilon=\text{Target-Total Covered Tags} \quad (10)$$

$$\text{Target}=100\% \text{ of the deployed tags} \quad (11)$$

We assume that if the tags are covered they will be detected but in practical applications it will not always hold, thus we provide an error reference value determined by defining three cases. In the first case, we associated a uniform probability distribution to the event of a tag being detected. The second and third cases define concentric circular areas inside the reading region with variable probabilities of detecting tags. For the two regions case we defined 90% and 10% as the probability detection rates, whereas 90%, 70% and 10% for the three regions case (Fig. 4). Finally, the parameter obtained is a reference value that indicates the number of tags that might not be detected.

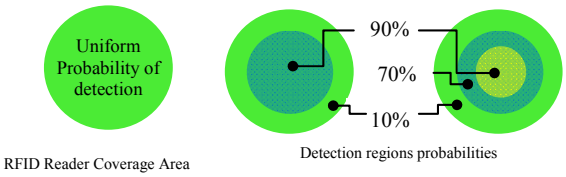

Fig. 4. Zones with different tag detection probabilities.

4) Number of readers located out of the deployment area

Since we are allowing a huge number of solutions to be evaluated we need to restrict as valid ones all of those whose readers are only located inside the area to be covered. For that purpose we measure the number of readers located out of the established bounds and we eliminate such solutions from the valid ones (Equations (12) to (14)).

$$f_4=\frac{1}{1+\varepsilon^2} \quad (12)$$

$$\varepsilon=\text{Target-Total Readers out of bounds} \quad (13)$$

$$\text{Target}=0 \quad (14)$$

5) Number of redundant readers

Some solutions might include redundant readers (Fig. 5), which happens when two or more readers cover the same or a subset of tags. Such solutions must be discarded because they imply interference, no useful operation and additional cost. In order to measure the redundant readers, we observe the set of tags that each reader might detect and we increase the count each time we find a redundant one. The target is set to zero redundant readers (Equations (15) to (17)).

$$f_5=\frac{1}{1+\varepsilon^2} \quad (15)$$

$$\varepsilon=\text{Target-Total Redundant Readers} \quad (16)$$

$$\text{Target}=0 \quad (17)$$

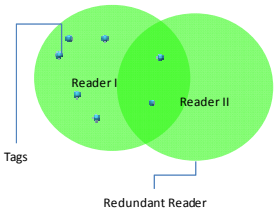

Fig. 5. Redundant Reader.

6) Number of tags located in overlapped reading areas

Finally, we observe that for certain geometrical considerations readers will tend to overlap in some areas. Additionally, it is possible to obtain solutions that comply with all the previous five objectives but that may present tags located inside overlapped areas (Fig. 6). Consequently, reading collisions might occur because several readers can be performing a reading process at the same time. Hence, we defined our last objective function to take into account and prevent this problem (Equations (18) to (20)). The target is set to obtain zero tags in the overlapped areas.

$$f_6=\frac{1}{1+\varepsilon^2} \quad (18)$$

$$\varepsilon=\text{Target-Total Tags in overlapped area} \quad (19)$$

$$\text{Target}=0 \quad (20)$$

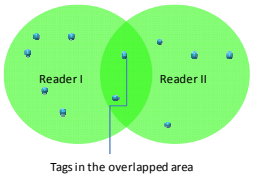

Fig. 6. Tags located in the overlapped area.

C. Problem Coding

The topology design problem is coded by using traditional binary coding. A reader's position and its power level are represented with 21 bits. The binary string is divided as shown in Fig. 7. The first 3 bits provide a 7 power step parameter. The following power steps are used: 0 (reader OFF), 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, and 1 Watt (maximum power). Then the 2D position of the reader is defined by a X-Y Cartesian coordinate system with a 10 cm resolution. By allocating 5 bits for the integer part of the position we can obtain 31 different values and by using 4 bits for the decimal part we code values from 0 to 9. Readers are placed between the range of X [0, 31.9] and Y [0,31.9]. For bigger areas the string needs to be modified in order to allow more values.

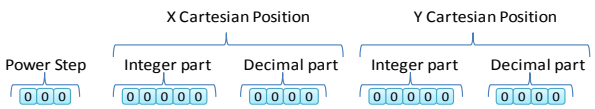

Fig. 7. Genes representing a RFID Reader.

If we define a maximum of 10 readers to be deployed, each candidate solution is composed by 10 x 21 = 210 bits. This gives us a huge search space of 2^{210} possibilities where GAs perform successfully in order to obtain an optimal solution.

D. Propagation Models

We used the Friis equation [13] in order to estimate the coverage area of each reader as a reference value (Equation (21)).

$$Pr=Pt\frac{GtGr\lambda^2}{(4\pi r)^2} \quad (21)$$

Where:

Pr: Received power, Pt: Transmitted power, Gr: Receiving antenna gain

Gt: Transmitting antenna gain, λ: wavelength of the frequency used

r: reading range

For representing indoor scenarios, we use the ITU indoor propagation model. The ITU indoor path loss model is expressed in equation (22).

$$L=20\log f+N\log d+Pf(n)-28 \quad (22)$$

Where:

L = the total path loss (dB). f = Frequency of transmission (MHz). d = Distance (m).

N = the distance power loss coefficient. n = Number of floors between the transmitter and receiver.

$P_f(n)$ = the floor loss penetration factor.

E. Genetic Algorithm Implementation

The GA method proposed has the following operators implemented:

1) Selection

Three selection operators are implemented: *Roulette* Wheel, *Tournament* and *Random Selection*. In the first method, each individual is assigned a slice of a "roulette wheel" where the size of each division is related to the fitness they have. The roulette will spin as many times as the size of the new required population. The higher the fitness value, the higher the probabilities for an individual to be selected. In the Tournament method we select "k" individuals from the population. "k" is known as the tournament size. The individual that has the highest fitness among the rest is the one to be selected. Finally Random Selection chooses an individual among the total population randomly. We also implemented Elitism that keeps the best solution obtained through all the iterations performed.

2) Crossover

We implemented two types of crossover: one point crossover with two parents, and two points' crossover with three parents [14]. The idea of the crossover is to combine genes from the selected parents to obtain a new solution that theoretically should perform better. The cutting points are selected randomly and then the genes are combined to form new individuals. Once the new individuals are generated, the algorithm chooses the best solution (highest fitness) among the new individuals as the new offspring.

3) *Mutation*

Three mutation operators were implemented. The simplest one flips one randomly selected gen from 1 to 0 or vice versa. The other two operators are: Cataclysmic mutation and Migration [15]. The first performs intensive bits flips randomly in the whole population (except on the best solution kept if Elitism is used). The second method introduces randomly generated individuals. Mutation augments the diversity and reduces convergence to local optima.

V. SOFTWARE TOOL DESIGN

The purpose of developing a software tool is to provide a flexible framework that permits the experimentation with several parameters variation but also for observing the solutions generated and providing a graphical interface for specifying scenario problems. JAVA SE [16] was used to develop the application on the Netbeans 6.9 IDE [17]. The tool provides several customizable parameters that can be easily saved and recalled. The block diagram of the application is represented in the following figure (Fig. 8).

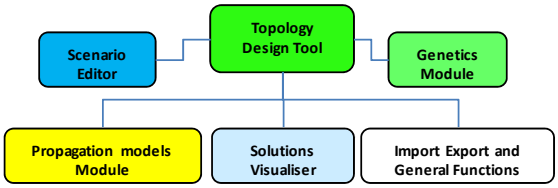

Fig. 8. RFID Topology design Tool modular structure.

1) *Customizable Parameters*

The tool has a Graphical user Interface (GUI) that allows setting and modifying a variety of parameters. For the GA we can define the following parameters: Population size, number of genes per individual, mutation and crossover rates, cataclysmic and migration counters, type of crossover, type of selection and Elitism. The set of RFID parameters that can be modified are: Frequency operation, number of deployed tags, number of readers, transmission antenna gain, tag antenna gain and the power threshold to detect tags. An option for testing that allows us to generate random deployed tags scenarios is also included. Finally, the set of weights of the cost function the size of the area to be covered and the interference ratio can be tuned. The screenshots are presented in Fig. 9.

2) *Import-Export and other functions*

The tool provides saving and loading functionalities of all the parameters involved. Additionally, for testing purposes a function to generate, save and load random scenarios was included as well as a Scenario Editor for defining deployment problems (Fig.10 right side). The GA can be executed in a step-by-step fashion to observe the evolution of the solutions. The results and the execution of the GA can be seen in a graphical way (Fig.10 left side). Moreover, we can register the time and the number of generations used to find a solution into a text file which can be utilized for statistics processing. We can also generate and export the solution report into pdf format. The button shortcuts of all the functions are shown in Fig.11.

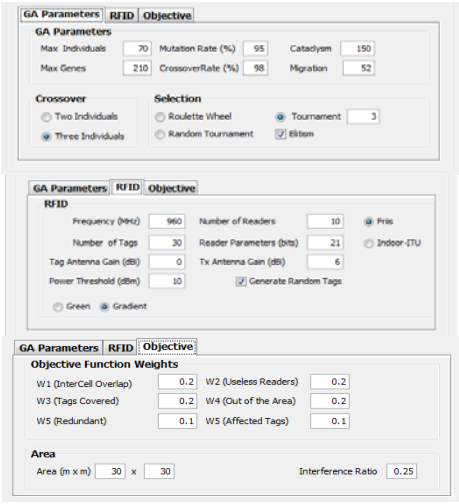

Fig. 9. Topology design tool customizable parameters.

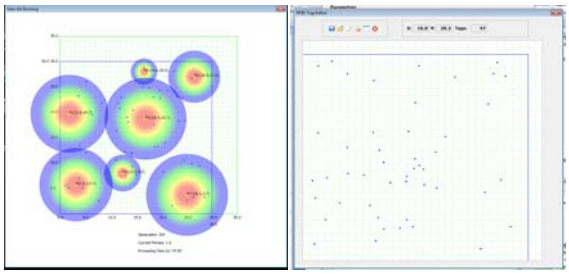

Fig. 10. Solutions Pane (Left) and Scenario Editor (Right).

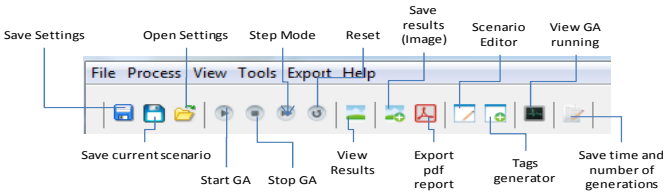

Fig. 11. Tool functions menu shortcuts.

VI. EXPERIMENT AND RESULTS

The objective of our experiment is to measure the time and the number of generations that were employed for different cases. We rely on the assumptions that the current or approximate position of the RFID tags can be obtained and RFID Tags' antennae are never at 90° with respect to the readers, thus they can always be detected.

A. *Processing Time and number of iterations*

We measured the number of iterations and the time required to solve a scenario composed of 30 tags into a square area of 20 m x 20 m. The coefficients for the objective function were set empirically to 0.2 for w_1 to w_4 and to 0.1 for w_5 and w_6. We allowed an interference ratio of 0.25. The GA was set to run with a three individual crossover, roulette wheel selection and elitism activated. We ran 100 times each experiment. The complete list of parameters is presented in the Table I.

TABLE I. Simulation Parameters.

Parameters			
Number of Individuals	70	Cataclysmic counter	150
Num genes per individual	210	Number of Readers	10
Mutation percentage	95	Number of Tags	30
Croosover percentage	98	Readers parameters (bits)	21
Elitism	TRUE	Interference Ratio	0.25
X Area	20	Migration counter	52
Y Area	20	Operation Frequency (MHz)	960
W1 Intercell Overlap	0.2	Power Threshold (Rx Tags dBm)	10
W2 Useless Readers	0.2	Tx antena gain (dBi)	6
W3 Tags Covered	0.2	Tag antena gain (dBi)	0
W4 Out of Area	0.2	Three individual crossover	TRUE
W5 Redundant Readers	0.1	Roulette Selection	TRUE
W6 Tags in overlapped	0.1	Friis and ITU models	

The first measurement involved the comparison of Friis and ITU propagation model in terms of number of iterations and processing time. The results are expressed in Fig. 12. We observe that the ITU model presents less coverage range than the Friis, consequently, it implies that more iterations and processing time are required for the ITU in order to obtain an optimal solution in contrast with the Friis model.

Secondly, we compared the variation of the crossover choice in the GA. We used two and three individual methods for both ITU and Friis models. The results are presented in Fig. 13. We observe that three individual crossover allows the convergence of the solution in less number of iterations though more processing time per iteration is required. This is due to the complexity of the three-individual crossover function where more candidate solutions need to be evaluated in order to obtain the final individual. However, this increase in processing time is compensated by the smaller number of iterations required to obtain a solution with fitness equals to one.

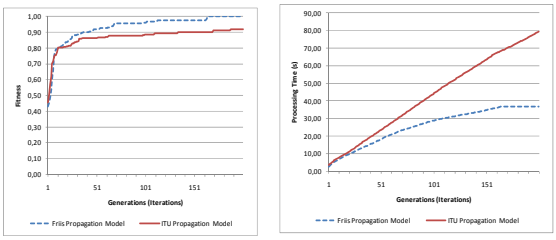

Fig. 12. Number of generations vs. fitness and vs. processing time for Friis and ITU propagation models.

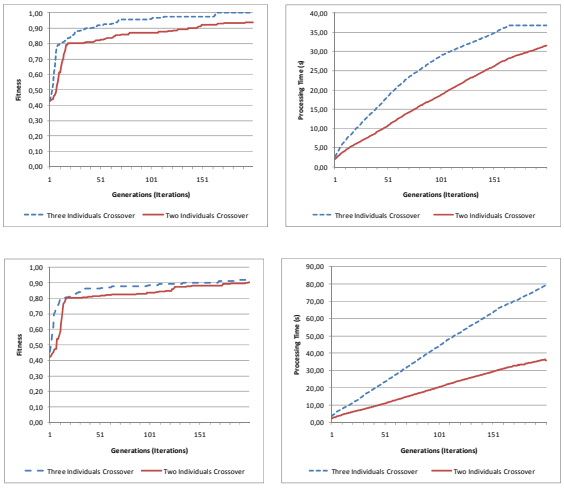

Fig. 13. Number of generations vs. fitness and vs. processing time for Friis and ITU propagation models using 2 and 3 individual crossover.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a RFID network topology design based on Genetic Algorithms. This approach produces flexible RFID reader deployment layouts that optimize the location and power level of the readers to be installed. It takes into account factors such as interference, number of tags to be detected and reader redundancy. We also developed a customizable software tool to assist in the design and we performed measurements that enable the performance comparison of all the processes involved in the topology design. As future work, we are planning to study other optimization methods for more dynamic network configurations in order to improve the processing time.

REFERENCES

[1] Wiebking L., Metz G., Korpela M., Nikkanen M, Penttila K., A Roadmap for RFID Applications and Technologies, Published on Internet by "Coordinating European Efforts for Promoting the European RFID Value Chain" (CE RFID). August 12, 2008. http://www.rfid-in-action.eu/public/results

[2] Penttinen A., *Chapter 10 – Network Planning and Dimensioning, Lecture Notes: S-38.145 - Introduction to Teletraffic Theory*, Helsinki University of Technology, Fall 1999.

[3] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge.

[4] Hyunsik Seo; Chaewoo Lee; , "A New GA-Based Resource Allocation Scheme for a Reader-to-Reader Interference Problem in RFID Systems," Communications (ICC), 2010 IEEE International Conference on , vol., no., pp.1-5, 23-27 May 2010.

[5] Qiang Guan; Yu Liu; Yiping Yang; Wenshneg Yu; , "Genetic Approach for Network Planning in the RFID Systems," *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on* , vol.2, no., pp.567-572, 16-18 Oct. 2006.

[6] Chiu Chui-Yu; Ke Cheng-Hsin; Chen, K.Y.; , "Optimal RFID networks scheduling using genetic algorithm and swarm intelligence," *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* , vol., no., pp.1201-1208, 11-14 Oct. 2009.

[7] Ben Niu; Wong, E.C.; Yujuan Chai; Li Li; , "RFID Network Planning Based on MCPSO Alogorithm," *Information Science and Engineering (ISISE), 2009 Second International Symposium on* , vol., no., pp.8-12, 26-28 Dec. 2009.

[8] Yahui Yang; Yujie Wu; Min Xia; Zhijing Qin; , "A RFID Network Planning Method Based on Genetic Algorithm," *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on* , vol.1, no., pp.534-537, 25-26 April 2009.

[9] Hanning Chen; Yunlong Zhu; Kunyuan Hu; , "RFID networks planning using a multi-swarm optimizer," *Control and Decision Conference, 2009. CCDC '09. Chinese* , vol., no., pp.3548-3552, 17-19 June 2009.

[10] K. Finkenzeller. RFID handbook - Second Edition. JOHN WILEYSONS, 2003.

[11] Chaouchi Hakima, The Internet of Things: Connecting Objects, Wiley-ISTE,2010.Chapter 5 by Oscar Botero and Hakima Chaouchi.

[12] Lubachevsky B., Graham R: "Minimum perimeter rectangles that enclose congruent non-overlapping circles". Journal Discrete Mathematics 309 (2009) 1947–1962 Elsevier.

[13] Syed Ahson and Mohammad Ilyas, RFID handbook : applications, technology, security, and privacy / edited by. CRC press 2008. pag. 76.

[14] Eiben, A. and Raué, P. and Ruttkay, Zs., Parallel Problem Solving from Nature PPSN III, Springer Berlin / Heidelberg, vol. 866, pages 78-8, 1994.

[15] Darrell Whitley, "A Genetic Algorithm Tutorial", "*Statistics and Computing Journal 1994*". vol. no. 4, pages 65-85.

[16] http://www.oracle.com/technetwork/java/javase/overview/index.html

[17] http://netbeans.org/