Modeling and Optimizing RFID Network Planning by using Genetic Algorithms as a Computational Intelligent Technique

Ali Ashkzari

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
July 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Isik Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Alexander Chefranov
2. Asst. Prof. Dr. Sahand Daneshvar
3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

The Radio Frequency Identification (RFID) is a kind of technology, which utilizes radio frequency waves to examine and read transporters or tags. RFID wireless network planning is an emerging automatic device, which has gained increasing popularity in last decades. It is one of the advanced devices, which have many applications in various branches like fraud and counterfeit prevention, military, supply chain and asset management.

In a range of applications, the use of RFID systems has led to RFID network planning (RNP) problem. This problem must be resolved if RFID systems are to be used optimally in a large scale. It is worth mentioning that RNP problem is an arguing issue to resolve. Generally speaking, RNP attempts to optimize certain applications such as load balance, economic efficiency and interference between readers by regulating the control variables of the system such as reader coordinates, reader numbers, aerial parameters and coverage of system, all at the same time. The positions of these readers and tags cannot be designed or preplanned because the position of readers and tags can be changed in various areas due to the tags, which are randomly deployed in the area.

This investigation, studies the modeling and optimizing RFID network planning by using Genetic Algorithms (GA) as a computational intelligent technique, in order to achieve optimal solution for the best number of readers based on maximum coverage of tags by deploying minimum number of readers in the network. The findings showed that the result of GA's is more beneficial than the other methods that mentioned in the references.

Keywords: Radio Frequency Identification (RFID), RFID Network Planning (RNP), Optimization, Genetic Algorithms, Wireless Network Planning.

ÖZ

Radyo Frekansı ile Tanımlama (RFID), taşıyıcıları veya etiketlerini incelemek ve okumak için radyo frekans dalgaları kullanan bir teknoloji türüdür. RFID kullanılarak kablosuz ağ planlaması son yıllarda giderek popülerlik kazanmıştır ve gelişmekte olan otomatik bir cihazdır. Bu cihaz, sahtekarlık ve kalpazanlık önlemede, askeriyede, tedarik zincirinde ve varlık yönetimi gibi çeşitli alanlarda kullanılan gelişmiş cihazlardan biridir.

Farklı uygulama alanlarında RFID sistemlerinin kullanımı RFID ağ planlama (RNP) sorunu da beraberinde getirmiştir. RFID sistemleri büyük ölçekli uygulamalarda etkili şekilde kullanılacaksa RNP sorunu çözülmelidir. RNP sorununu çözme probleminin hala tartışılan bir konu olduğunu belirtmekte yarar var. Genel olarak konuşursak, RNP, okuyucu koordinatları, okuyucu sayıları, ortam parametreleri ve kapsama alanı gibi, sistem kontrol değişkenlerini düzenleyerek, tüm sistemde, okuyucular arasında yük dengesi, ekonomik etkinlik ve girişim gerektiren uygulamaları optimize etmek için çalışır. Bazı uygulamalarda okuyucu ve etiketlerin konumları önceden tasarlanamaz veya planlanamaz, çünkü alana bağlı olarak bu cihazların yerleri farklılık gösterebilir (rastgele seçilebilir).

Bu çalışma, minimum sayıda okuyucunun maksimum kapsama alanı ile hizmet verebilmesini sağlamak için RFID ağ planlamada ve modellemede akıllı hesaplama tekniğini olarak Genetik Algoritma (GA) kullanılması üzerinedir. Elde edilen sonuçlar sorunun çözümünde GA kullanımının referanslarda belirtilen diğer yöntemlerden daha iyi sonuç verdiğini göstermiştir.

Anahtar Kelimeler: Radyo Frekansı ile Tanımlama (RFID), RFID Ağ Planlama (RNP), Optimizasyon, Genetik Algoritmalar, Kablosuz Ağ Planlama.

Dedicated to my wife

ACKNOWLEDGMENT

First and foremost, I offer my sincerest gratitude to my supervisor, dear Dr. Gürcü Öz, who has supported me throughout my thesis with his patience and knowledge. I attribute the level of my Master degree to his encouragement and effort and without her this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

I would like to express my deepest gratitude and appreciation to Assoc. Prof. Dr. Alexander Chefranov and Dr. Sahand Daneshvar for the useful comments, remarks through the learning process of my Master degree. Last, but not least, I am greatly indebted to my lovely wife for her invaluable supports.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

dB	Decibel
dBm	Decibel Milliwatt
EA	Evolutionary Algorithms
GA	Genetic Algorithm
GPSO	Global Particle Swarm Optimization
HF	High Frequency
IC	Integrated Circuit
ITF	Interference
LF	Low Frequency
PSO	Particle Swarm Optimization
QoS	Quality of Service
RFID	Radio Frequency Identification
RNP	RFID Network Planning
SI	Swarm Intelligence
TRE	Tentative Reader Elimination
UHF	Ultra High Frequency
UID	Unique Identification
VNPSO	Von Neumann Particle Swarm Optimization

Chapter 1

INTRODUCTION

1.1 General Information

Nowadays, with the development of wireless network technology, the need for further optimization of these networks is felt. One of these networks is Radio Frequency Identification that called RFID, to be able to get correct and also trusted network planning for the RFID commerce systems, the best place of the readers deployed in the area and also related values of variables must be identified. Each one of this sections need to optimize a collection of aims, like coverage, best number of readers, power that used for communication between tags and readers and interferences among readers and tags [1], [2]. The positioning of RFID systems leads to generate the RFID Network Planning (RNP) problem. The actual positioning of RFID has acquire the (RNP) issue that must to fixed and resolved involving RFID readers within the best style. Nonetheless, RNP is among the nearly all difficult issues that have to satisfy a lot of involves from the RFID method. Generally, the target of RNP is to optimize and improve some objectives such as (coverage, quality of service (QoS), economical effectiveness along with minimum value for interference concerning readers, etc.) at the same time simply by altering your control issues (the coordinates from the readers, how many readers can covered system, and also the readers variables) of the process. As a result, in the large-scale deployment environment, RNP problem is a high-dimensional nonlinear optimization problem with a large number of variables and uncertain parameters.

1.2 Objectives

In this thesis, we attempted to develop a mathematical model for planning RFID networks on the basis of the application of powerful technique for optimization that called Genetic Algorithm. The GA is a stochastic global search method which mimics the metaphor of natural biological evolution. Genetic Algorithm operates on a population of potential solutions applying the principle of survival of the fittest to produce (hopefully) better and better approximations to a solution [3]. The GA are particularly suitable for solving complex optimization problems and for applications that require adaptive problem solving strategies. Here, in this thesis GA is introduced as an intelligent optimization technique.

The algorithms are coded and simulated by MATLAB. The main aim of this study is using genetic algorithms as an intelligent optimization technique to identify the best number of readers based on maximum coverage of tags by deploying minimum number of readers and develop an optimized model of RNP for any environment. According to inputs value that include the size of the area (basis of square meters), number of tags (are randomly placed in the environment), number of segmenting on environment and number of population, that used in genetic algorithms for solving the problem.

1.3 Outline of the Thesis

The remainder of our thesis is prepared in the following ways, Chapter 2 provides the RFID and Genetic Algorithm and literature review of the related work. System overview, system formulations and Genetic Algorithm methods are overviewed in Chapter 3. Chapter 4 describes the conduction of simulation, system setup and results that obtained from optimized RFID Network Planning and Chapter 5 concludes the thesis with a summary of obtained results and the proposals of a further work.

Chapter 2

RFID AND GENETIC ALGORITHM

2.1 Radio Frequency Identification (RFID)

RFID is a kind of technology, which utilizes radio frequency waves to examine and read transporters or tags. Tags possess unique identification numbers (UID) which kept in tag memory in the form of bits. Upon being reading, these numbers sent to the reader. There are also other pieces of information, which kept in the tag memory other than UID. This technology reads the tag as it sees the number. A very simple system of radio frequency identification (RFID) enjoys a reader connected to an aerial and tag with a cable. This system reads the tags [1], [4].

RFID is an emerging automatic device, which has gained increasing popularity in last decades [1]. It is one of the advanced devices, which have many applications in different fields such as fraud and counterfeit prevention, military, and supply chain and asset management. Figure 2.1 illustrates a typical RFID system that consists of three parts. 1) A small device, which reads the items and it called tag. 2) The reader that transmits data to and from the tag through radio signals. 3) A computer system, which analyses the information [1], [5].

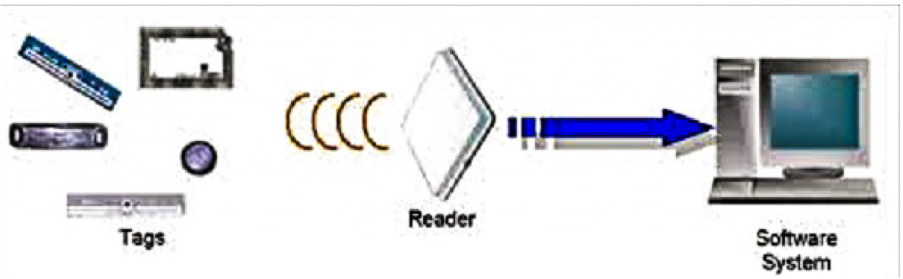


Figure 2.1: The component of RFID [6]

RFID technology is a smart system that can send the data through the tag, these data are then read by RFID reader and analyzed by different application requirements and needs. Recently, the use of RFID technology has grown rapidly and has been modified to meet the needs of different industrial applications such as military, assembly line, supply chain and asset management. The three aforementioned components of RFID system will discussed with reference to their different types [6].

2.1.1 RFID Tag

The first component is as noted earlier is tag which comes in three different types passive, semi - passive (semi - active), and active. Figure 2.2 illustrates samples of RFID tags. These are described in details in the following paragraphs.

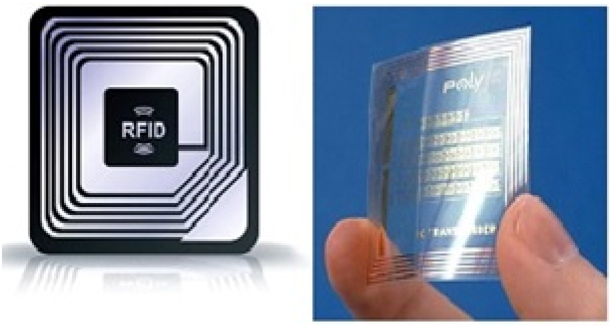


Figure 2.2: Samples of RFID Tags

For functioning and operating, power is needed for RFID tags. In passive tags, readers provide the power. In semi-passive tags, although a battery supplies the power for the responses and ranges, for the operation, the reader still provides the tag with the power.

In active tag one battery or other power sources provide the necessary power, which leaves the reader aside because the tag needs no power from the reader.

a) Passive Tags

Passive tags have wide applications in different fields around the world. This is due to certain features and characteristics of passive tags, which enumerated as follows:

- Its portability and longevity (twenty years of lifespan)
- No need for battery
- Getting the necessary power from the waves sent by the reader.
- A tag read range of a few centimeters/inches to about 12 meters or 37 feet. This makes the device quite useful because by just a few tags per second one can obtain an excellent read rate. It is worth noting that this is possible only with very low loss cables and a high gain circular polarized aerial, which cannot be obtained in European countries because of limited power and bandwidth.

However, one point with regard to the disadvantage of passive tags should be noted here which is its shorter read range. Figure 2.3 shows the sample of passive RFID tag.

Figure 2.3: Sample of passive RFID Tags

b) Semi-Passive Tags

Semi passive tags use batteries to power the IC and has a tag read range of about up to 30 meters or about 100 feet. If they coupled with sensors, they can measure a wide variety of different conditions. However, for its operation, it needs lower reader signal. It also enjoys a mode for saving the battery by putting it on the sleep mode. However, it has some disadvantages such as higher cost, a big and voluminous tag, and battery upkeep [1].

c) Active Tags

As mentioned earlier, active tags need a battery or other power source for activation and operation. A group of them can be connected to large power supply of about 9 Volt, 12 Volt or AC power. They possess a radio transmitter which makes a larger read range of 750 feet or more possible. They are also equipped with an alarm technology can respond in times of failure or cable cuts. Besides, Active tags enjoy a larger memory of about 64 K bytes. Another feature of this tag is its sensor or sensors, which are sensitive to movement, magnetic changes, weather conditions such as pressure and temperature. They can also be set to operate with the reader or with different environmental conditions or can be set to go to the sleep mode in times of inactivation. These devices can also store the information for later use or application. However, they have some disadvantages as well such as its higher cost, bigger and larger size and its heavy weight [7]. Figure 2.4 present two samples of active RFID tags.

Figure 2.4: Sample of Active RFID Tags

2.1.2 RFID Reader

Radio transmissions are used by the reader to send a query to the tag, and by the tag to return an answer, generally containing identifying information. The reader sends the identifying information to a network, or sometimes directly to a host computer, where it can be displayed in human-readable form, or incorporated into a database to track objects and guide the activities of people and machines. Figure 2.5 shows two typical readers, which are used in industry for read the tags.

Figure 2.5: Two typical RFID readers

Another component of RFID Network is software system, that is a generic term used to describe software that resides between the RFID reader and enterprise applications. There are many RFID software system products on the market.

2.2 RFID Network Planning

Since the connection among the readers and tags does not take place properly, RFID systems use a number of readers to resolve the issue. As a result, the extra accessories create their own peculiar sort of problems and questions such as the number of readers, the place of the reader, and the parameter setting for each reader. One of these problems is the RFID network planning (RNP) problem. This problem involves a number of issues such as coverage, cost, and quality. Traditionally, manual trial and error approach was used which time and labor was consuming [1], [5]. Besides, the signals emitted from them were invisible to the naked eye, which made the measurement difficult. However, these problems have been resolved by the use of new technologies such as computer devices [8].

2.2.1 RFID Network

RFID reader network resembles that of a sensor network. There are a lot of sensor nodes placed inside the sensor network. The positions of these sensor nodes cannot be designed or preplanned, because the position makes the use of random nodes possible in different environments or disaster-stricken areas. Another noteworthy point is the ability of sensor networks protocols and algorithms to organize and control themselves. The sensor nodes in sensor networks work well together and are equipped with inbuilt processor [9], [8]. This processor performs the mathematical calculations and sends the necessary data to the nodes, which take care of the fusion tasks. Different researchers have used different methods to resolve the problem of sensor networks which suffered from steady state processes. A sensor network for estimating maximum accuracy was designed by [2] to use the graph theory for the application of process configurations and sensor networks in the design strategy [3], [10]. Nevertheless, none of these algorithms could resolve the problem completely and could not optimize the sensor networks because they only could take one criterion into account [11].

2.2.2 Techniques for RNP Problem

In a range of applications, the use of RFID systems has led to RFID network planning (RNP) problem. This problem must be resolved if RFID systems are to be used optimally in a large scale. It is worth noting that RNP problem is not an easy problem to be resolved. Generally speaking, RNP attempts to optimize certain applications such as load balance and economic efficiency and interference between readers by regulating the control variables of the system such as reader coordinates, reader numbers, aerial parameters all at the same time. Therefore, RNP is still considered a nonlinear optimization problem with its concomitant variables and parameters the large-scale applications [1].

RFID systems use the decibel (dB) to in all hardware applications and specifications for the description of antenna gain, power production, and cable problems. For employing RFID system in different countries certain specifications and regulations should be taken into account for different frequencies. If it is not installed properly, serious damages are inevitable such as legal and health ones. The dB is one tenth of a Bel and is a ratio between two signal strengths. Alexander Graham Bell, the inventor of telephone was the first person to discover this signal and it is named after him. Bel is calculated through a logarithmic scale equation which uses the logarithm of a physical quantity instead of the quantity itself [12], [7].

$$Bel = log\,(P2/P1) \tag{2.1}$$

dB is another logarithmic equation which measures large scale variations in strength of signals. In equation (2.2) easily calculates RFID system gain and losses by adding and subtracting whole numbers.

$$dB = 10 * log\ (P2/P1) \tag{2.2}$$

The dB unit controls signal strength and level variations through simple mathematical formulae. Gain is considered positive while loss is considered negative. A three dB gain/loss amounts to a times signal level increase and decrease, that is, if a cable experience a three dB loss, its signal strength decreases fifty percent as it goes to the other end of the cable. Moreover, a ten dB gain or loss means a signal level increase or decrease of ten times. In other words, the cable loses ninety percent of its signal strength when frequency or signal reaches to its other end. Besides, a twenty dB gain or loss equals to a signal level increase or decrease of a hundred times meaning a ninety nine percent of signal strength loss when the signal travels to the other end of the cable [10].

Watts or dBm are used as units of power level of radio frequency. dBm as an electrical unit is used to describe power in decibel and is equal to one mille watt ($1mW$). It is calculated through as equation (2.3) in the following [1].

$$P(dBm) = 10 * Log\ (P/1mW) \tag{2.3}$$

As you can see in the Table 2.1, 30 dBm is equal to 1.0 Watt, which should be used in our coming calculations [7].

Table 2.1: dBm Value Base on (W)

dBm	**Watt**
0	0.001
10	0.01
20	0.1
30	1.0
40	10.0

2.2.3 Tag and Reader Communication

As we stated earlier, passive tags do not work with batteries and use the power generated from the electromagnetic wave of the reader to transmit information to the reader. Using load modulation in the near field such as LF, HF, and UHF, and using backscatter in the far field such as UHF and Microwave, Passive RFID tags and readers send and receive information between themselves by connecting the transmitter to the receiver [7].

In case of near field communication, the tag uses the electromagnetic inductance to transmit information to and with the reader. The tag aerial, which is in coiled form, acts as a transformer. As a result, the reader makes use of the carrier waves and modifies the frequency, phase, and amplitude [7].

2.2.4 Backscatters

Further, in case of far field communication, backscatters are employed. Backscatters are defined as radio frequency wave reflection upon striking a conductive surface. The energy generated this way varies according to the surface resonance, that is, how well the surface resonates with the frequency on the wave in [1], [4].

Here, we need to clarify what resonance mean. In actuality, resonance is defined as the system oscillation at maximum amplitude and at a certain frequency. This can be resembled to the case of a musical instrument like violin, for example, when you pick its stings and play them, it is not just the strings that generates the sound rather it is the whole instrument. As a result, the tag aerial resonates with the system carrier frequency; this leads to the incapability of the readers of 134.2 kHz to read the 125 kHz tags even at very close frequencies [13].

Furthermore, UHF tags can be designed to resonate according to the material type that they are connected to in far field communication. Therefore, different tags serve a variety of functions for different types of materials such as Pallet tags for wood, plastic tags for plastic, metal tags for metal and etc [7].

UHF tags can be specifically designed in way to make the maximum resonance upon hitting different materials. Therefore, a tag cannot generate correct information unless it is attached to the materials. Passive backscatter serves the task of sending and receiving data between the reader and the tag in the UHF tags. Some of the energy generated from the electromagnetic wave supplies the tag Integrated Circuit (IC) with the necessary power while backscatter transmits the data to and from the reader [7].

As we have noted earlier, RFID technology is advancement for identifications of items and all other applications. In many countries, Multi RFID readers are used in different fields and for various purposes. They also make use of various reader resources. However, RFID-based application is not without flaws and has some defects. Sometimes when the tag is not detected, reads are done incorrectly especially in multi-tag and multi-reader settings. In other words, when readers hit each other incorrect reads happen [5], [7].

2.2.5 Reader Position

Companies can easily chase and follow the inventory movement because of the position of RFID aerials in strategic locations. However, an item can be easily lost when it moves out of the aerial reading range. Labor can be saved and no inventory can be lost if the companies can detect and read those items which move out of the antenna reading range. RFID has become a widely-used technology due to the issuance of mandates from giant companies such as Wal-Mart, Target, and even the US Department of Defense which oblige all the producers and companies to tag their products. More companies will welcome the idea of development of readers which can detect and read the items or facilities to follow their inventory [1], [2].

2.2.6 Coverage of RNP System

Coverage problem is an important issue to be resolved in case of static RNP. It is quite a common problem. This is due to the fact that RFID systems identify different items in particular geographical areas. Extensive studies have been carried out on the issue of coverage problem of RNP. Each area is divided into certain number of grids which includes readers and tags. There are other studies which have focused on continuous working areas where there are no fixed positions for the readers and tags [17], [18], and [22]. Reduction of the number of readers is an important task RFID network planning because the network is quite complicated and the RFID system's price is determined by the number of the readers that used in the network [1].

2.3 Optimization

The word optimization refers to the process of improving the quality of a service or products or to the betterment of something. In our everyday life, we encounter numerous cases of optimization which are solved or resolved by our own efforts. For example, which clothes are better to buy, which university is better to study in while taking into account a variety of other factors such as tuition fees, transportation, and etc. However, in our case, optimization is improving the input of a process, or function or operation of a device to its best possible and maximum state. The inputs are considered as variables while process or function are considered as objective function. In this thesis, cost minimization is going to be studied based on readers that deployed in the network. As a result, the problems and functions are taken into account here as minimization problem [9].

In optimization, single objective optimization refers to the case when only one objective function is considered, but in most situations and cases more than one objective function is involved which calls for different methodology. Such cases are called multi-objective optimization. Optimization solving methods have been categorized into two groups of classical methods and evolutionary methods by [9]. These methods will be described in the following sections:

The classical methods make use of single random solution. To reach an optimal solution, these methods are updated after each recursion through a certain procedure. These methods are further categorized into two distinct groups: direct methods and gradient-based methods. In the direct methods, the objective function and the constraints value are used while in the gradient-based methods the first and second offshoot of objective function and constraints are used for direction and optimal solution [11]. The next method as we mentioned earlier is the evolutionary method which as the name suggests.

It is rooted in the human evolutionary process which has been extensively used and been popular since 1960. This method resembles the evolutionary nature principle with a peculiar optimization algorithm. As mentioned in [14], argued that this method could produce much better results than the early classical methods. In each iteration, this method or better to say, this algorithm applies an initial population of random solutions while in the classical methods a single solution was used. To reach optimization, the initial population of each generation is updated after ach iteration. This method could easily solve multi-objective optimization problems since it used a population of optimum solution in a single simulation use. This method is also categorized into three different groups of genetic algorithm, evolutionary programming and evolutionary strategy in [5], however, [2] added another method to the aforementioned categories and that is genetic programming. These methods will be discussed in the next section.

2.3.1 Optimization Techniques

Two important search intensive tasks are placement and routing. Agent objects employ information for decreasing the amount of time on search time, however, still a lot of searching has to be carried out. Optimizing the different components' positioning in the layout requires a great deal of the search time. To resolve this issue in the best possible way, a number of optimization techniques are used which are categorized into three groups [9], [14].

a) Numerical techniques

These techniques also are in turn divided into two methods of direct and indirect. Indirect methods are used to compute the non-linear equations which are resulted from considering the objective function as zero. To reach the likely solutions, search is restricted to points with zero slope in every directions. On the other hand, direct methods search the extremes by moving around the search space and calculating the new point gradients; this leads to the search. These techniques are considered as a hill climbing notion which finds the best spots by going up the steepest permissible gradient. Direct techniques have very limited applications [9], [14].

b) Enumerative techniques:

These techniques searched all the function's domain space points one at a time. These techniques demand accuracy on the part of the implementer and are easy to carry out and are not used for applications which involve large domain spaces. To mention a good example from these techniques one can name dynamic programming [9], [14].

c) Guided random search techniques:

These techniques are considered as enumerative techniques, however, to carry out the search they need extra information. These techniques are also further divided into two categories such as simulated annealing and evolutionary algorithms which are regarded as evolutionary processes. However, it should be noted that for searching minimum energy states, simulated annealing technique employs a thermodynamic evolution process, whereas, Evolutionary algorithms make use of natural selection which evolves throughout different generations. Evolutionary algorithms also use biologically inspired operations to improve potential solutions [9], [14].

A variety of techniques can be employed by the agents for optimizing different locations and positions. Nowadays, the Eval Agent class is the most popular techniques for performing a genetic algorithm. Nevertheless, for carrying out other techniques, other classes can be produced. These other classes can substitute the Eval Agent class without alterations or changes to the program. Although, simulated annealing seems to be the right choice in the coming era, genetic algorithm is the preferred one because it is a new method which produces better solutions [9], [14].

2.4 Genetic Algorithms (GA)

Considering the optimization problems in different fields and areas such as trade, engineering and industry, sometimes some of them are to complex and difficult to handle due to a lot of other intervening factors such as different variables, search spaces, resources and time management. As mentioned in [2], came up with this algorithm but only later when his student used this algorithm to solve a problem in his thesis, this algorithm came to be recognized and widely used [3], [14].

As the saying goes only the fittest survive in the jungle. That is only those which can adapt the environmental conditions will be able to last longer. Evolution takes place through the changes experienced in different species, however, only the species' genetics that carry and store these changes. Evolution is the greatest problem solving machine because through a set of primitive organic molecules, it creates a wonder of different lives with great subtlety and diversity. Each species in the world possess certain characteristics. This is mapped onto an algorithm which can create species that can live in certain environment [3], [14]. Holland proposed this algorithm which comprised of a set of computational models simulating natural evolution to solve a myriad of problems. This algorithm can solve multifaceted optimization problems and applications requiring adaptive problem solving strategies [4] [15].

To resolve optimization problems, a genetic algorithm simulates the process of biological evolution. The genetic algorithm is composed of some individual elements and a number of biologically inspired operators affecting the individual elements. As we mentioned in the previous sections, it is only the fittest who survive and generate offspring and transmit their genetics to the new generations; this is the main idea behind the evolutionary theory. In this theory, a sting of numbers are assigned to each potential solution which is regarded as an individual in the population, then each string represents one individual in turn. When it comes to this point, separation of the individuals from the string becomes the next task. In this regard, genetic algorithm plays an important role by using the potential strings to reach the optimum solution [15]. This algorithm has a cyclical operation which is quite easy to follow:

a) Creating a string population
b) Assessment of the strings
c) Choosing the potential strings
d) Using genetic for producing a new population of strings

In this operation, for each problem, each cycle creates a new generation of individuals or likely solutions. For example, at the beginning, a population of likely solutions is produced. The individuals are assigned a string resembling the chromosome which is used by the genetic operators. Then each individual is assessed and examined by being created from the string description which is its chromosome. After that, its implementation is checked and controlled; which shows the suitability of the individual while performing its duty in the population. The potential and best pairs for genetic manipulation process are chosen according to their fitness criteria [15].

This stage assures the selection of the fittest individuals in the population. Then, the offspring or the new population is created by using the chosen pairs' genetic information. The defining chromosome or strings of each individual keeps the information. In this operation, crossover and mutation operators are employed. Each offspring substitutes the one generation before and the process continues until the desired outcome is reached [15].

2.4.1 The Objective and Fitness Function

This function is applied to evaluate the performance of the individuals in the problem domain. When it comes to minimization problems, the fit individuals have a very low value in the related associated objective functions. This is an indication of the relative individual performance in a GA at an intermediate stage. The fitness function changes the objective function value into a relative fitness measure [3], [15]. Therefore the following equation can be obtained:

$$F(x) = g(f(x)) \tag{2.4}$$

In this equation, f represents the objective function, g changes the objective function value to a positive number, and F is relative fitness measure which is obtained at the end. This equation is used only when the objective function value is reduced to its lower value to show the fitter individuals. In the majority of equations, fitness function value represents the number of offspring created from the individuals in the coming generation. As a result, proportional fitness assignment is used [3], [15].

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N_{ind}} f(x_i)} \tag{2.5}$$

In equation (2.5), N_{ind} shows the population size and xi represents the phenotypic value of individual i. This fitness assignment attempts to give a chance for each individual to be reproduced according to its relative fitness, however, it cannot do anything to reduce the negative objective function values. Therefore, before performing fitness assignment, a linear transformation is carried out that compensate for the objective function:

$$F(x) = af(x) + b \tag{2.6}$$

In this function, a becomes positive when the optimization increases and gets negative when optimization decreases. The offset b makes sure that the outcome becomes positive [3], [15].

2.4.2 Selection Operator Role in Genetic Algorithms

Making more copies of better strings is an important task of selection operator in the population. This is considered as the first method used on population. Selection or reproduction operator chooses the appropriate strings for making a mating pool, that is, why it is also called reproduction operator. As a result, in this operation, those individuals which form successful structures reproduce more than the other individuals. The selection operation makes population generation possible. There are different selection operators available, which will be discussed in the following subsections [16].

a) Rank Selection

In case of Roulette Wheel Selection, some problems occur where the fitness value varies too much, that is, if the most suitable chromosome fitness turns out to be 90% of the entire roulette wheel, then the other chromosomes will have the low likelihood of being selected. Rank selection solves this problem by ranking each chromosome in the population. It starts from the worst chromosome as 1, second worst as 2, and proceeds forward as to n which represents number of chromosomes in population. To calculate the probability of population Pi in rank selection, the following formula is used: $Pi = (Pi\ Rank)\ /\ (Total\ sum\ of\ individual\ Rank)$. In this operator, every chromosome has a likelihood of being chosen; however, it has one problem which is slower convergence, that is, the best chromosomes also have no priority [3], [14] [15].

b) Steady State Selection

This technique is used to save the chromosomes lives and to transfer them to the next generation, but it is not used for selecting parents. It follows a particular procedure which is elucidated as follows:

1. A few suitable chromosomes with a high fitness value is chosen for reproduction.

2. Some unsuitable chromosomes with low fitness value are deleted and are substituted by the new offspring.

When this procedure is followed, the remaining population will survive to new generation.

c) Elitism

Upon applying crossover and mutation technique, there is a possibility of losing the best chromosomes. Elitism arranges the chromosomes from the highest to the lowest in terms of their fitness values. Then, the selection technique is used for each two chromosomes in the arranged set. Here, Genetic Algorithm is used between the strong chromosomes or weak chromosomes meaning that the algorithm cannot be applied through combining weak and strong chromosomes. This method replicates a few best chromosomes in new population and the rest of operation is carried out through classical way. Therefore, it improves GA performance by storing the best found solution [15].

d) Tournament Selection

GA follows a special strategy by selecting individuals from the population and by putting them in a mating pool. Out of the pool, certain individuals are selected to reproduce for the next generation. Since the genes possessed by these individuals are used for the next generation, the best individuals are stored in the mating pools.

Therefore, GA selects the most suitable and the best individuals out of the population to be stored in the mating pool. Tournament selection follows a particular order in carrying out the task:

1. The total no of matches should be equal to the number of teams.

2. No more than two matches are played by each team.

e) Roulette Wheel Selection Methods

Roulette-wheel technique is used for the selection of the probable individuals according to their performance in the population. A value is taken as the total of the individuals' anticipated selection probabilities or the total of raw fitness values in comparison to other individuals in the population. Then, each individual is assigned a contiguous interval in the range. This interval equals the related individual' fitness value. For the selection of each individual, the interval uses a number and hence an individual which corresponds to that number. To reach the expected number of individuals this process is replicated and repeated. The main roulette wheel selection method is regarded as stochastic sampling with replacement (SSR). In this method, the segment size and selection probability are kept constant during the selection and individuals are chosen like the procedures described above. This technique is not flawless, it is bias free but produces a potentially unlimited spread and individuals with segment size bigger than 0 could be used for the creation of next population [15]. Figure 2.6 shows the Roulette Wheel selection. We have used this method in our implementation as selection part of Genetic Algorithm.

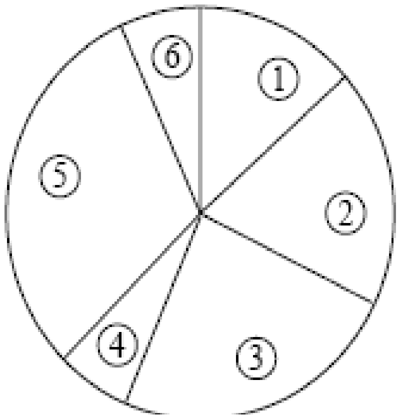


Figure 2.6: Roulette Wheel Selection [15]

2.4.3 Crossover

This technique or method is used to recombine the genetic materials of the individual population. It selects two chromosomes, combines their genetic information and creates new chromosomes. Figure 2.7 shows the process of crossover.

This activity resembles the natural reproduction. When the crossover points are randomly selected, a new offspring son is reproduced through recombination of parents' chromosomes or strings.

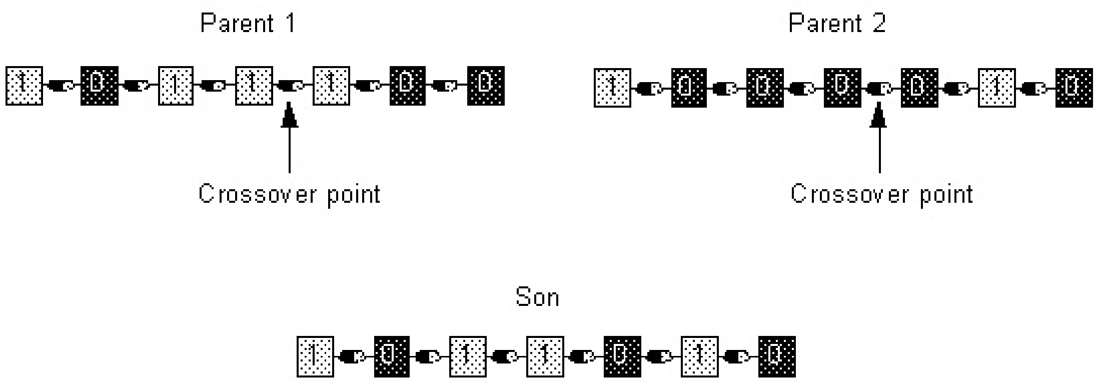


Figure 2.7: Process of Crossover [15]

The process of selection adopted in this technique makes sure that genetic structures which are called building blocks are stored for the next generation. These building blocks show the fit genetic structures in the individuals in population.

a) One-Point Crossover

One-Point Crossover is a method which uses a crossover point in its application from the chromosome of parents. This method then swaps the chromosomes of parents. As a result, two new offspring are created. In what follows, you can see two parents selected for one point crossover application. "|" symbol represents the random selection:

Parent 1: 11001|010

Parent 2: 00100|111

The following offspring are created when the parents' chromosome bits are exchanged and swapped at the crossover point [15].

Offspring1: 11001|111

Offspring2: 00100|010

b) Two-Point Crossover

This is another method in which two crossover points within the parent chromosomes are randomly chosen. Then, during the interval between two points parents' genes are exchanged.

2.4.4 Mutation

The process of recombination cannot search the slots which are not shown in the population's genetic formation and as a result encounter problems around minimum values. Therefore, due to the random nature of this process, different structures associated with other search spaces are created. Mutation only as the name suggests changes bit from chromosome in a random manner [3], [15].

Figure 2.8 shows the application of operator in the fifth element of the chromosome and process of mutation.

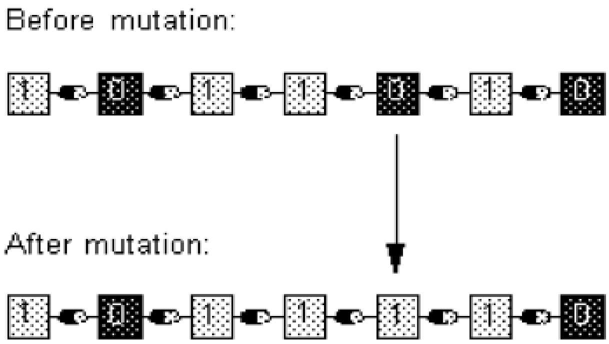


Figure 2.8: Process of Mutation [15]

2.4.5 Problem Dependent Parameters

This is an important feature of genetic algorithms' computational model. It checks and controls the necessary stages for the creation of an algorithm. However, a real-time implementation takes some problem-dependent parameters into consideration during the implementation process. Genetic manipulation produces some offspring which can substitute the entire manipulation or just the less fit members. The best possible options are determined by the problem constraints [3], [15].

If the best possible options are required, then some parameters such as the size of the population, the rate of crossover and mutation as well as evaluation and convergence criteria must be adapted and adjusted.

2.4.6 Encoding

Encoding is an important solution for the optimization problems in the algorithm performance.

In the past, for this purpose only binary encodings including the crossover and mutation operators were employed because of its ease of applicability. However, in case of symbols other than 1 and 0, the crossover and mutation operators have to be adjusted according to the numbers.

2.4.7 Variable Values for Optimization

There are various variables in every optimization problem. Fixed-point integer encoding is a technique which is used to encode these variables in binary forms by applying a certain number of bits. These codes are later combined in the population strings. However, the problem with the binary strings is its Hamming cliffs which are large hamming distances between the codes of neighboring integers [15]. These Hamming cliffs create problems for the algorithm which traditional techniques such as mutation and crossover are not able to resolve.

The robust representations are what encoding desires, that is, in case of random representation change, a possible individual will appear.

2.4.8 Evaluation Step

This step involves optimization of the actual system by the algorithm. This step selects the individual strings of the population and produces the individuals which have to be evaluated. The encoding of the individuals within the strings is related to the optimization parameters and the real structure of likely solutions. However, one point should be taken into account here and that is the size of the resulting strings, because it decreases the speed of the process greatly. Individuals are evaluated after they are produced [15], [16]. The evaluation and scoring of the individuals depend on the real system optimization. In other words, the evaluation is determined by the optimization characteristics and the scoring criteria. For instance, the creation of a certain value for the individual fitness greatly depends on the relative importance of the obtained values [14].

2.5 Former Methods for Optimizing RFID

Former methods for optimizing RFID such as Swarm Intelligence (SI) and Evaluation Computation (EC) methods are kinds of search algorithms based on population encounter problems in changing the readers number in the search process, these algorithms apply fixed representation which makes their search space limited. A limited number of studies have so far embarked on minimization of readers. The studies have focused on certain working areas by selecting a number of reader sites in advance [1].

However, the algorithms proposed in the studies have not been able to optimize the readers as well as radiated of power and the coordinates of readers all at the same time. Therefore, they can only have certain applications.

The working areas that use sequential algorithms cannot reduce the number of readers in the optimization part, hence they cannot determine the selected readers positions in sequential working areas. A number of researchers have pointed to the application of certain readers number in RFID planning [17], [18], however, they have all used a fixed number of readers in their research. A noteworthy point here is the inaccuracy of human beings in determining the readers and number of them.

Moreover, RFID is an NP problem because RFID network makes use of fewer numbers of readers. Previous deterministic algorithms cannot resolve this problem in real times because of the complexity of the problem, although SI method and EC have been used extensively in the last two decades. Using these algorithm have had some advantages for the network such as conceptual simplicity, high efficiency, flexibility, robustness, use of domain information for combination with other techniques and etc. as a result of these benefits, SI and EC have wide applications in industry. A number of EC and SI algorithms have been recently formulated, and expanded to resolve the problems of RNP. Among these algorithm the widely used ones are Particle Swarm Optimization (PSO) algorithms, Genetic Algorithms (GA), Bacterial Foraging Algorithms (BFA) and Differential Evolution (DE) [1].

Chapter 3

SYSTEM OWERVIEW

3.1 System Components

In our network plan, we need to install readers on the network area to coverage all parts for RFID network planning that leads to read all tags, which randomly distributed in the environment according to our RNP problem that the numbers of readers are need to cover the entire perimeter. According to that our problem that is to minimize the number of readers by maximum coverage of system, we must reduce the number of readers in the network to solve RNP problem.

We developed an optimized model of RNP for any environment, any number of tags with optimum number and position of readers that needed in the network to coverage all tags.

For achieve to these objectives we used genetic algorithm as intelligent technique for optimizing RFID network planning to have maximum coverage of system by minimum number of readers and minimum interference between readers and tags and also appropriate power in the network.

For this research, we used passive-tags in the network. Passive RFID tags use the power generated through the signal radiation of the reader to establish backscatter communication and because of this do not require any power source. As a result, the communication between the reader and tag cannot take place unless certain features needed, the power sent to the tag must be larger than the threshold level or value, and power received by the reader must be larger than another threshold. If these features are not present, the reader will not be able to show sensitivity to the signal sent by the tag. The communication that takes place in backscatter has somehow a short range. However, due to cost-efficiency and longevity of passive tags, passive tags are widely used in different RFID systems all around the world [1], [4].

In this study we tried to find the best place for deploy the readers. For example, if we have working area with 50 m^2 and 30 RFID tags, how many readers should be deployed, where these readers must be placed to cover the entire area of RNP. Table 3.1 represents Device Settings that used for modeling an RFID network according to [1] .

Table 3.1: Device Settings [1]

Parameter	**Value**
Operation Frequency	915 MHz with (wavelength = 0.328 m)
Threshold of Tag (T_t)	-14 dBm
Threshold of reader (T_r)	-80 dBm
Gain of Reader (G_r)	6.8 dB
Gain of Tag (G_t)	3.7 dB

The first assumption is that the number of readers and tags, in the first phase are equal and in our planning, the area defined by square meter (m^2). The first step is using mathematical formulas to make a model of RFID system.

3.2 System Formulation

For modeling and implementing RFID network planning and to specify the network, several formulas and recent methods are used. However, a mathematical model is not always perfect and best, with the passage of time the beater and more robust mathematical models will be obtained. Table 3.2 illustrates the notations that used for modeling of RNP in the following formulas [1].

Table 3.2: Notations in the Formulation of RNP [1]

Symbols	Descriptions
RS	The set of deployed readers
TS	The set of tags
$PT_{r,t}$	The power received by tag t from reader r
$PR_{t,r}$	The backscatter power received by reader r from tag t
T_t	The threshold value of tag to build reader-to-tag communication
T_r	The threshold value of reader to build tag-to- reader communication
N_t	The number of tags distributed in the working area
N_{max}	The total number of readers which could be deployed in the network
N_{red}	The number of redundant readers found by the algorithm
N_r	The number of readers deployed in the network
PS_r	The amount of power transmitted by reader r

3.2.1 Link Budget

The reader and tag transmit signals to each other through their aerials in backscatter communication. In what follows, the transition process and link budget equation can be seen, where a line-of-sight (LOS) communication is established which works with the power sent to tag. The power is calculated from [1] as follows:

$$P_t\ [dB_m] = P_r\ [dB_m] + G_r\,[dB_i] + G_t[dB_i] - L\,[dB] \qquad (3.1)$$

In equation (3.1), P_r represents the power sent from the reader, G_r shows the reader aerial gain, G_t represents tag aerial gain, and L shows the attenuation factor which is calculated by the following equation called Friis transmission equation from [1].

$$L\,[dB] = 10 \log\left[\left(\frac{4\pi}{\lambda}\right) d^n\right] + \delta\,[dB] \qquad (3.2)$$

Since RFID systems are used for operation in outdoors, consideration of multipath loss should always be made. As a result, in the equation (3.2), λ shows the wavelength, d stands for the distance between two different devices, n also is a number between 1.5 to 4 which varies according to the different geological conditions, and δ represents the impairments incurred of whatsoever such as cable loss, polarization decrease, and etc.

In equation (3.3), The power P_b received by backscatter from the tag is estimated through calculating the reflection coefficient of the Tag, T_{tag}. This is an important coefficient, which shows the reflected power in relation to the power received by the device P_t. As you can see in the previous formula, the communication between tag and reader is established and the power $P_b = (T_{tag})^2 P_t$ (in watt), received by the reader is calculated using the Friis transmission equation in [1].

$$P_r = P_b\ [dB_m] + G_t\,[dB_i] + G_r[dB_i] - 20\log\left(\frac{4\pi\, d}{\lambda}\right) \qquad (3.3)$$

3.2.2 Tag Coverage

One of the most important tasks of RFID network planning is to improve the coverage level. In most of the applications, a full coverage level is required, that is, all the tags need to be covered by at least one reader. RS used to indicate the set of readers used and TS is used to show the set of tags. In the connection among tags and reader, for each tag, that is, $t \in TS$, tag can be covered if one reader $r1 \in RS$ is available which satisfies $PTr1, t \geq Tt$ and a reader $r2 \in RS$ which satisfiesPR_t, r_2≥ T_r. In this formula, $PTr1, t$ represents the power obtained by tag transmitted from $r1$. This relation is calculated by [1] where $PRt, r2$ represents the backscatter signal received by reader r_2 from tag, which is evident in equation (3.5), $T_t$and T_r show the levels of threshold among tags and the reader from [1].

The network coverage rate defined in equation (3.4).

$$COV = \sum_{t \in TS} \frac{Cv\ (t)}{N_t} * 100\ \% \tag{3.4}$$

In the equations (3.4), $N_t = |TS|$ shows the tags number that deployed in our environment [1].

And $Cv\ (t)$ is calculated as equation (3.5):

$$Cv\ (t) = \begin{cases} 1, & if\ \exists\ r_1 r_2 \in RS, PT_{r1,r2} \geq T_t \cap PR_{t,r2} \geq\ T_r \\ 0, & Otherwise \end{cases} \tag{3.5}$$

3.2.3 Readers Number

The number of readers (N_r), used in a working area determines the network expense of the RFID network. However, when the coverage goal is met, the readers number in RNP is detracted. If the total number of readers used is N_{max} and the total redundant readers is N_{red}, then the number of RFID network will be calculated in [1] as such formula (3.6).

$$N_r = |RS| = N_{max} -\ N_{red} \tag{3.6}$$

3.2.4 Interference

When a number of readers review same tag simultaneously, interference happens in the covering area of the used readers. This might misread or lower the QoS in RFID system. As a result, one of the key roles of RNP is interference avoidance. RFID network interference equals the total sum of the interference value of tags [1]. This number calculated in equation (3.7).

$$ITF = \sum_{t \in TS} \gamma\ (t) \tag{3.7}$$

And $\gamma\ (t)$, calculated as follow:

$$\gamma\ (t) = \sum PT_{r,t} - \max\{PT_{r,t}\}, r\ \in RS\ \cap\ PT_{r,t}\ \geq\ T_t \tag{3.8}$$

As one can see, interference level reaches to zero only when tag $\boldsymbol{t}$ connected to one reader. The total amount of interference in an RFID network is defined as the sum of the interference value at each tag.

After modeling RFID network planning with the above formulas, now is the time for RNP optimization by using genetic algorithms. Determining the weight for different objective require more work due to the various and diverse objective units and dimensions. However, these objectives prioritize certain things in specific application despite multiple objectives. As a result, this approach used in this study in the evaluation process of the proposed Genetic Algorithms to take account of the multiple objectives of RNP.

3.3 Genetic Algorithms (GA) for RNP Optimization

GA algorithms are used in this thesis to resolve the problem of RNP on permanent or continuous working area. A program is develop in MATLAB, the first hypothesis of this program is that the number of readers are equal to number of tags in the environment, after that we recognize how many of them have been lead to reject ($N_{\max -} N_{red}$).

When this operation is done, one reader reduced from the network, which may lead to network coverage reduction as well. In case the coverage reaches 100% in some generations with the evolution of GA, reader permanently deleted. For the full coverage of the readers by minimizing them, the algorithm operates Tentative Reader

Elimination (TRE) some times in the search process. A mutation operator also used in this algorithm to prevent from premature convergence.

3.3.1 Tentative Reader Elimination (TRE)

To control the reader-switching vector, methods such as tentative reader elimination are used. The GA keeps the vector $N = [on^1, on^2, on^3, \dots, on^{Nmax}]$. TRE eradicates one reader from each round to decrease the number of used readers, because the tag coverage not affected this way. TRE eradicates the network one reader covering the fewest tags unless full coverage provided. In ON is considered as zero and N_r amount is decreased one. When this happens, network coverage is likely to drop. In the following generations, when coverage reaches 100 percent which is done by decreasing the readers, readers must be eliminated and as a result TRE operator decreases one reader from the network [1], [3].

The first hypothesis of this program is that the number of readers are equal to number of tags in the environment. Therefore, we recognize how many of them have been lead to reject $(N_{\max\,-}\ N_{red})$, that N is number of reader deployed in the network.

In the first stage the number of active (1) or deactivate (0) readers are randomly assigned. By this position of readers we make the chromosome for each line of population in GA and each member of chromosome represented as one reader that have (0 or 1) value which is called Gen. Members of each chromosome equal to the number of readers deployed in the network. We have *n* line population size that each line of this chromosome is a one answer of our problem according to population size that given from input.

$$
papulation\ size: \begin{cases} 1 \\ . \\ . \\ . \\ n \end{cases}
\begin{array}{ccc}
 & & \text{Gen} \\
 & & \downarrow \\
\end{array}
\begin{bmatrix} P1 = r1, r2, \dots, rn & \cdots & 01010 \dots 10101 \\ \vdots & \ddots & \vdots \\ Pn = r1, r2, \dots, rn & \cdots & 00100 \dots 00011 \end{bmatrix} \dashrightarrow \text{Chromosome}
$$

Position of Readers and Papulation (3.9)

We want to determine which line is the best answer from our population, according to fitness function.

In this study, we run the program 100 times to get the best coverage of system, that is calculate according to position of readers and tags, then calculate coverage, and send the answer to Fitness Function. The aim of the fitness function in our system is to have maximum coverage of network by using fitness as:

$$Fitness = \frac{Number\ of\ Runs}{1 + \ (100 - Cov)^2} \tag{3.10}$$

In equation (3.10), the result of fitness function for each iteration is obtained and the sum of this result according to number of runs should be calculated as total value of fitness function.

According to PT and PR we calculate the coverage in equation (3.4) and used in the calculation of fitness function with using equation (3.10). After that by applying crossover, system get one gen from chromosome for crossover as randomly to generate new children from parents.

3.3.2 Using Roulette Wheel for GA

Next step is selection part, fitness value is a decisive parameter since it determines the next population and selects the best possible pairs of chromosomes. In this thesis, efforts have been made to use a selection operator working on the Roulette Wheel Selection (RWS) approach. In this approach, some space of roulette wheel devoted to each individual according to the individual's fitness value. Those individuals, which have better fitness values, occupy a larger space or slot from the roulette wheel. Then, the longer slots chosen first. In our study, we only use two chromosomes and two operations previously mentioned, crossover and mutation are employed for creation of new chromosomes.

A specific percentage of a roulette wheel is devoted to each fitness value where in case of larger values this percentage also increases in turn. Roulette-wheel mechanism copies the string in the mating pool because the wheel circumference computed in reference to a string of fitness values.

This operator follows a particular procedure which described as follows:

Steps of Roulette Wheel Selection

1. Through the use of fitness function, each chromosome's fitness value is determined in the population.

2. The sum of fitness (Sf) value for n population chromosomes are calculated through by equation (3.11) [14].

$$Sf = \sum_{i=1}^{n} fvi \tag{3.11}$$

3. The average fitness (Af) is calculated in the Population as you can see in the formula (3.12).

$$Af = \frac{Sf}{n} \tag{3.12}$$

4. The expected fitness (Ef) is determined for each chromosome in the population as follow equation (3.13).

$$Efi = \frac{fvi}{Af} \tag{3.13}$$

5. The sum of expected fitness ($Sum\ Ef$) are calculated for all the population chromosomes in the following formula (3.14).

$$Sum\ Ef = \sum_{i=1}^{n} Efi \tag{3.14}$$

6. Equation (3.15) shows the generated random number (G) will also be determined in the range [0, Ef] .

$$G = Rnd(\ )mod\ Sum\ Ef \tag{3.15}$$

7. This procedure should be repeated n times, where n stands the population size [14].

3.4 Mutation and Crossover part for GA

As we stated previously, mutation considered as an important operators in genetic algorithm which have been used to prevent untimely convergence and promote performance. Mutation acts both as a performance enhancer and as an inseparable operator in GA algorithm with TRE [1]. This is due to the fact, when the algorithm is introduced for the first time, many readers were used may be additional by this power. This might lead to the elimination of some genes showing the large transmitted power as well as the population in GA. This in turn makes reader deletion difficult, since few readers might not be able to give full network coverage and more power from these readers might be required. As a result, mutation operator is introduced to fill this gap by bring back the deleted genes [3], [14].

Mutation works with a simple procedure. A chromosome i is randomly selected from the population, in each generation of the algorithm as ($i = 1,2, \dots, n$), that n is population size, then a dimension of gen's i position is chosen for the changes from population and according to population size.

After that again we use crossover and mutation, in mutation part we select random number between zero and one for probability of mutation.

$$probability\ of\ mutation = \frac{number\ of\ Gen\ in\ a\ chromosome}{size\ of\ population} \tag{3.16}$$

If the value of probability of mutation was greater than the probability of 0.6 then get a value of 1 and will use in next round and if probability was less than 0.6 , get 0 value and will not use in the next step. Probability of mutation calculated as equation (3.16).

Now we have new population by using the above steps and again we calculate power and coverage of new results to get the best answer among population. Given that the ultimate goal is to reaching maximum coverage of RNP with a minimum number of readers.

The significance of our research is as follows:

1. Application of a new redundant reader elimination technique to minimize the number of readers.
2. Resolving the issue of RNP by considering the number of readers, optimization of coordinates and transmitted power of readers.
3. Consideration of objectives in optimization process and application of genetic algorithm approach for the management of objectives.

Chapter 4

SIMULATIONS AND RESULTS

4.1 Simulation Setup and Parameters

In this chapter, we use MATLAB programming for implementation of modeling and optimizing RFID Network Planning (RNP) Genetic Algorithms is used. This program has the ability to develop an optimized model of RNP for any environment, any number of tags with optimum number and position of readers that are needed in the network for achieve to maximum coverage of system. Table 4.1 is representing Device Settings that are used for modeling our RFID network.

Table 4.1: Device Settings

Parameter	Value
Operation Frequency	915 MHz (with wavelength = 0.328 m)
Threshold of Tag (T_t)	-14 dBm
Threshold of reader (T_r)	-80 dBm
Gain of Reader (G_r)	6.8 dB
Gain of Tag (G_t)	3.7 dB

4.1.1 Input parameters

In this research, number of tags and area size of network being used in RNP, are taken as input from the user. As mentioned in the previous chapter, the first assumption is that the number of tags and readers are equal. Therefore, for each tag, one reader should be considered. On the other hand, in our planning the area is defined as $50m * 50m$.

Population size and number of segmentation area, which used in our program for solving the RNP, also taken as input from the user. The role of segmentation is determining the accuracy of Genetic Algorithms.

In addition, the value of power that used by reader and specifications of reader would be changed as well. In this part power will selected as input and it has entered manually according to the specifications of readers. In our simulation, we used input parameters same as [1] which is provided in Table 4.2.

Table 4.2: Input Parameters

Parameter	Value
X Area	50 m
Y Area	50 m
Tag Number	30
Population	10
Segmentation	8
Power Value	35 dBm

4.1.2 Output Parameters

Here we have measured percentage of coverage, number of readers, interference (ITF) and Fitness for 100 iterations at the end of the simulation. Optimum number of readers and their positions are also displayed in 50m *50m area.

The results of the program after modeling and optimization with GA for percentage of coverage of the system illustrated in Figure 4.1.

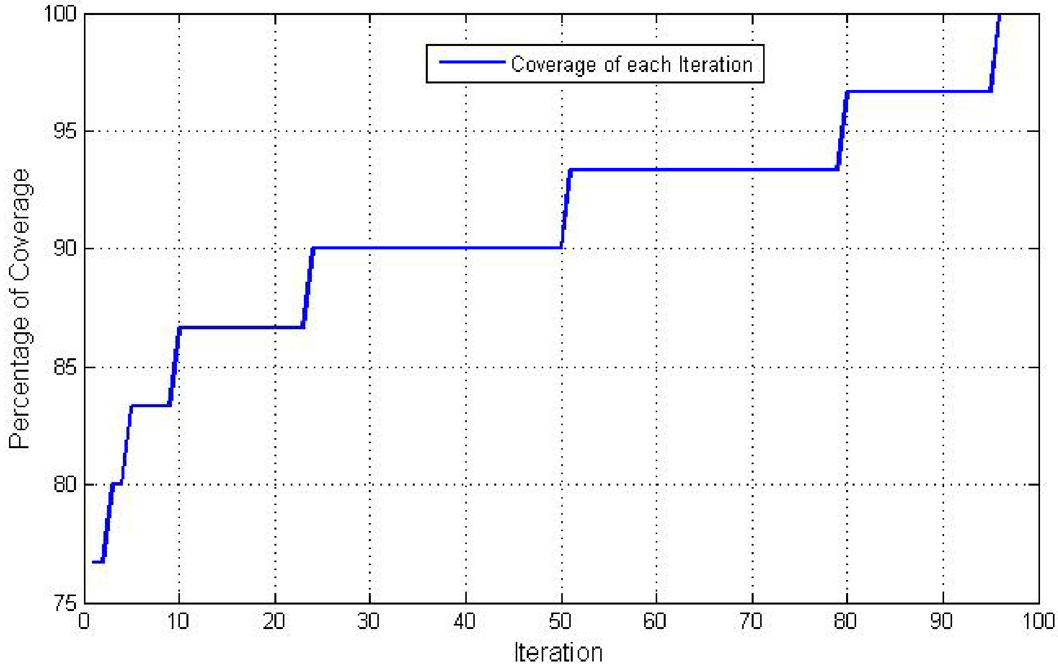


Figure 4.1: Value of Coverage for all iterations

In this particular case, 50 tags randomly distributed in the area. The position and the number of readers are calculated through running GA for 100 iterations with 20 population size. The Figure 4.1 shows coverage in the first iteration started from 77 percent among optimization and reached to 100% in iteration 96.

It is obvious in the Figure 4.1, after 100 iterations, the value of system coverage is 100 percent. Figure 4.2 represents optimum numbers of readers for each iteration. After optimization with three readers, there is 100 percent of coverage in the network. In other words, we have 47 redundant readers out of 50 readers after optimization with GA-RNP in the given area.

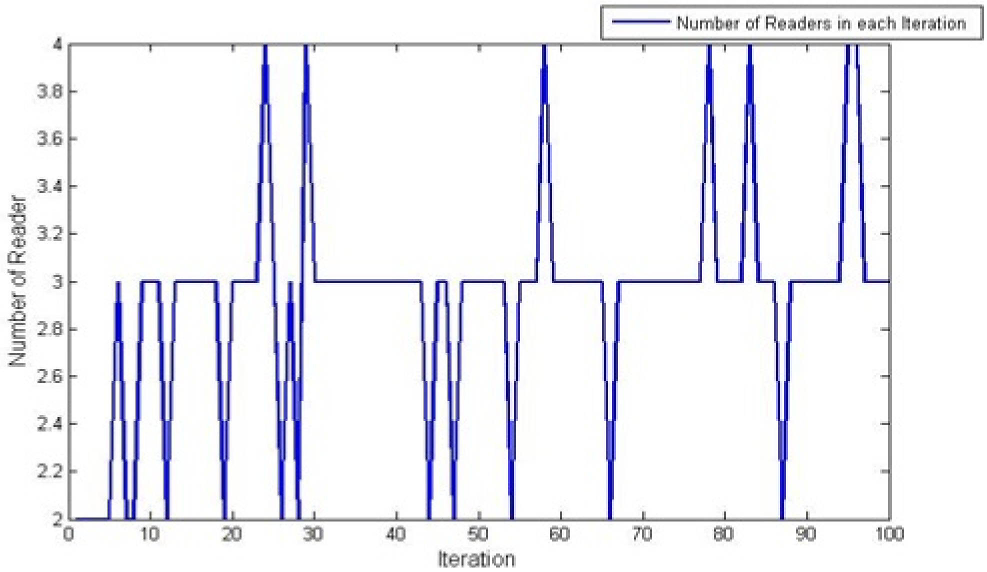


Figure 4.2: Number of Readers for all iterations

Figure 4.3 illustrates the result of interference (ITF) for each iteration. RFID network interference is equal to the sum of the interference value of tags [1]. In this study, the minimum value of interference reduced to 0 dBm after optimization.

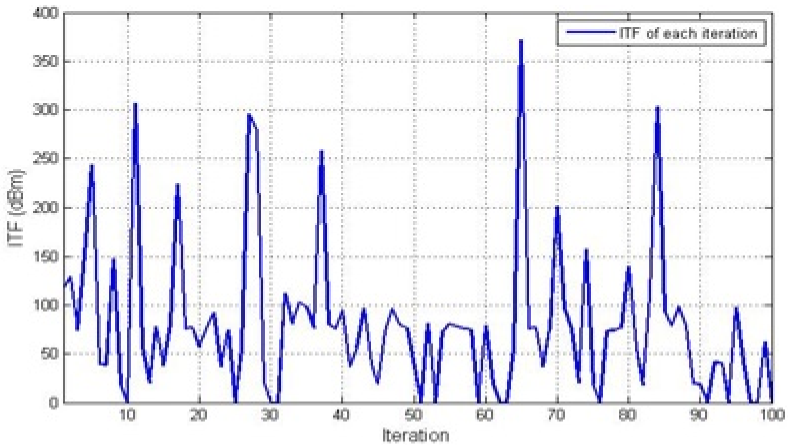


Figure 4.3: Value of Interference for all iterations (ITF)

The total amount of interference in an RFID network is defined as the sum of the interference value at each tag.

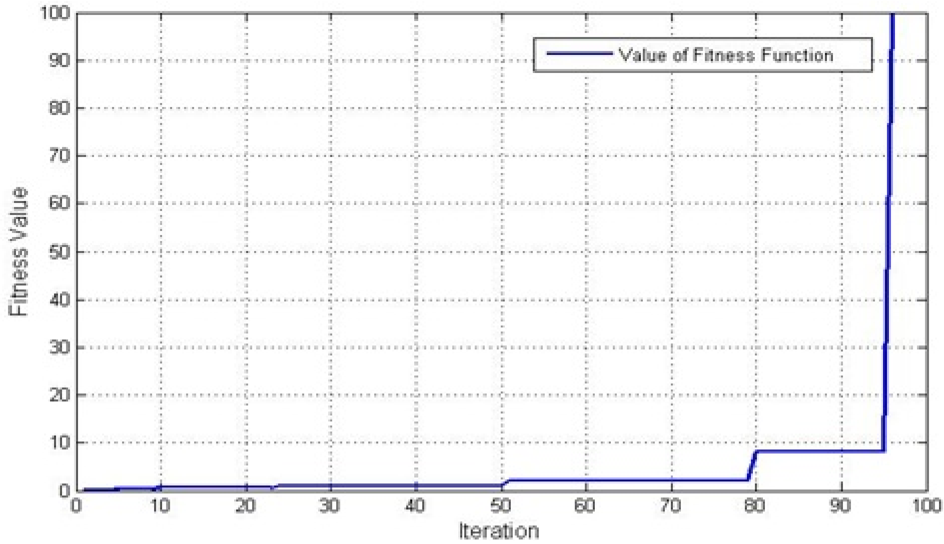


Figure 4.4: Value of Fitness Function in Each Iteration

Figure 4.4 shows the value of fitness function that started from 0 to 100 during optimization. This means that when the solution achieves to the highest value of fitness function, we reached to the optimum solution.

The optimum number of readers, the number of tags, positions are illustrated in Figure 4.5, for the given network area.

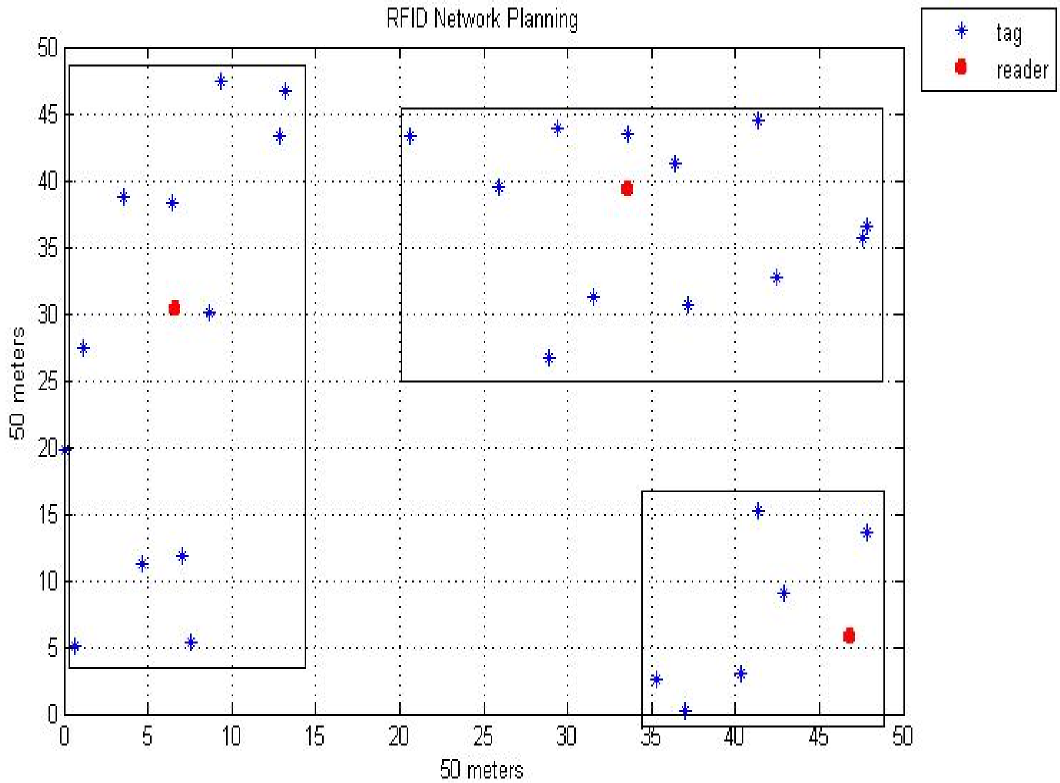


Figure 4.5: Positions of Tags and Optimum Number of Reader in the Given Area

4.2 Simulation Sets

In the following simulation sets, the program is iterated with the same settings. In each set, different set of parameters and power settings were used. The sets are classified based on different values of population size and segmentation used in (GA) according to Table 4.3.

Table 4.3: Parameters for Simulation Sets

Set Number	Population size	Segmentation
Set 1, 2, 3	10	4, 8, 16
Set 4, 5, 6	20	4, 8, 16

Each set has the power value ranging from 25 dBm to 40 dBm and tag numbers that includes 30, 50 and 100. According to Table 4.4 shows the other device settings, which are the same as [1] and used in our simulations as well.

Table 4.4: Device Settings

Parameter	Value
Operation Frequency	915 MHz (with wavelength = 0.328 m)
Threshold of Tag (T_t)	-14 dBm
Threshold of reader (T_r)	-80 dBm
Gain of Reader (G_r)	6.8 dB
Gain of Tag (G_t)	3.7 dB

For each set following performance metrics were measured:

1. The system coverage
2. Optimum number of reader
3. Interference in the network (ITF)
4. Fitness function in the network

4.2.1 Set 1: Population 10, Segmentation 4

Table 4.5 represents input parameters for set 1.

Table 4.5: Input Parameters for Set 1

Population Size	Area Size	Segmentation
10	50m * 50m	4

Table 4.6 shows the result for coverage of the network according to power value that started from 25 dBm to 40 dBm. The results are expressed in terms of percentage after using genetic algorithm for solving RNP in 100 iterations.

Table 4.6: Results of System Coverage in Set1

Power (dBm)	Tags number		
	30	**50**	**100**
25	55 %	65 %	61 %
30	55 %	65 %	61 %
31	89 %	95 %	95 %
32	100 %	100 %	96 %
33	100 %	100 %	98 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

As it is mentioned in Table 4.6 and Figure 4.6, with power value 34 dBm, there is 100 percent coverage in the network with the used number of tags. 100 percent coverage is reached with high power value for 100 tags when we are comparing with 30 tags, so, system can reach 100 percent coverage with high power value when number of tags is increasing.

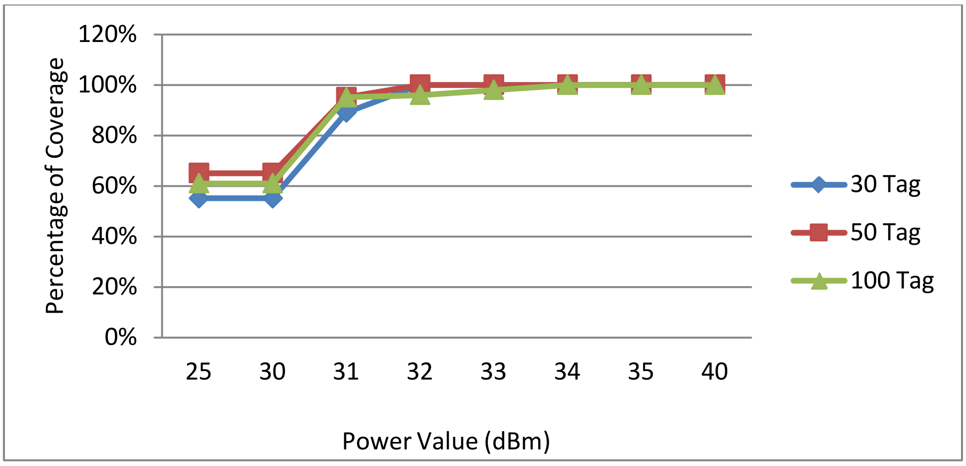


Figure 4.6: Coverage of RFID System in Set 1

Results for optimum number of readers that can cover all system after using genetic algorithm with 100 iterations are shown in Table 4.7. Increasing power of readers leads to decrease the number of readers in the network.

Table 4.7: Results for Best Number of Readers in Set1

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	5	8	12
30	5	8	12
31	4	5	6
32	5	4	6
33	3	4	4
34	3	4	3
35	3	4	3
40	1	2	1

In Figure 4.7, illustrates the best number of readers after optimization with GA in the system. The graph shows that the number of readers is reducing when power of reader is increasing. With 30 tags, when varying power value from 25 dBm to 40 dBm optimum number of readers is changing from 5 to 1 with 100 iterations.

On the other hand, with 100 tags when power value is 25 dBm, number of readers is 12. According to the results with 34 dBm power value, optimum number of readers is same for all tag numbers.

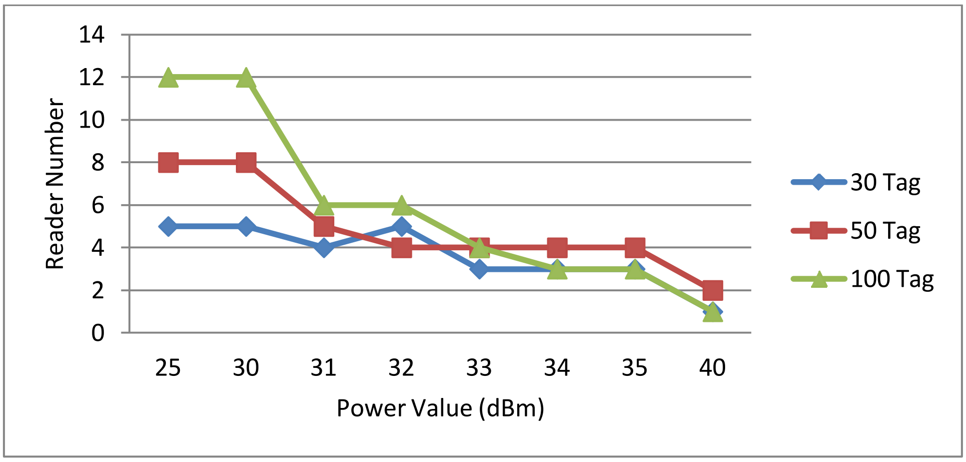


Figure 4.7: Reader Number for Set 1

Table 4.8 shows the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in Set 1. In this table due to the size of area, we do not have any interference for 30 and 50 tags in all powers and for 100 tags we received zero with 34 dBm of power value.

Table 4.8: Results of Interference in the Network

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0	0	50
30	0	0	30
31	0	0	20
32	0	0	20
33	0	0	20
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.9 shows the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value with power of 34 dBm, this means that the results are obtained after optimization with genetic algorithm in 34 dBm value of power was successful and the algorithm is reached to the best answer.

Table 4.9: Results of Fitness Function in Set1

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0.05	0.085	0.065
30	0.7	0.085	6
31	1	3	6
32	100	100	10
33	100	100	20
34	100	100	100
35	100	100	100
40	100	100	100

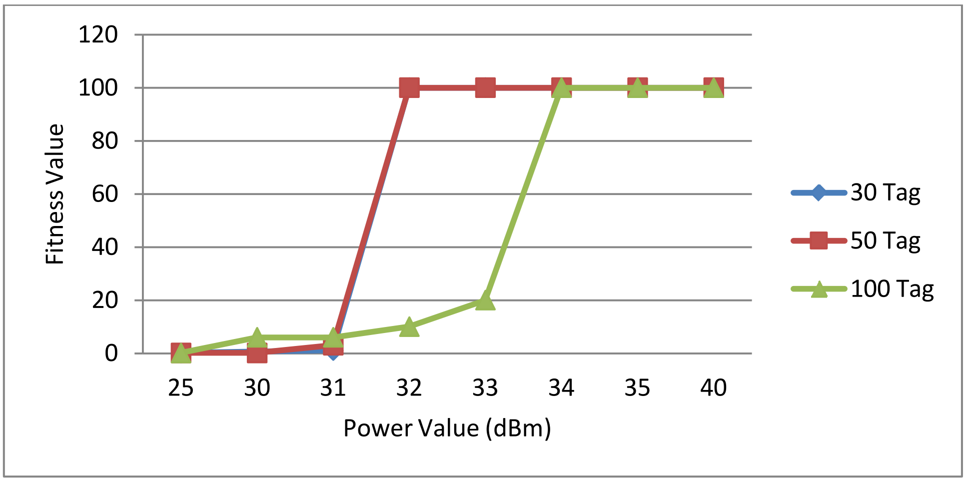


Figure 4.8: Fitness Function of Set 1

The Figure 4.8 shows different fitness values according to power values. As one can see from the Figure, the fitness value for 30 and 50 tags reached to 100 with power value of 32 dBm, compared to 100 tags that reached to 100 fitness value with 34 dBm value of power when fitness value is 100, the algorithm is reached to optimum result.

4.2.2 Set 2: Population 10, Segmentation 8

In this set of simulation we increased segmentation to 8 with the same population size in Set 1. Set 2 simulation parameters are shown in the Table 4.10

Table 4.10: Input Parameters for Set 2

Population Size	**Area Size**	**Segmentation**
10	50m * 50m	8

Table 4.11 shows the result for the network coverage according to power value that started from 25 dBm to 40 dBm. The results are expressed in terms of percentage after 100 iterations illustrated in Figure 4.9.

Table 4.11: Results of System Coverage in Set 2

Power	**Number of Tags**		
	30	50	100
25	60 %	65 %	66 %
30	94 %	94 %	96 %
31	97 %	96 %	96 %
32	100 %	100 %	99 %
33	100 %	100 %	100 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

When we have compared Set 2 results with Set 1 results we can observed that with segmentation 8 in GA, we have same improvement in the system coverage percentage with low power values. With power value 32 dBm, system reach's 100 % coverage with all number of tags, so we have some power save.

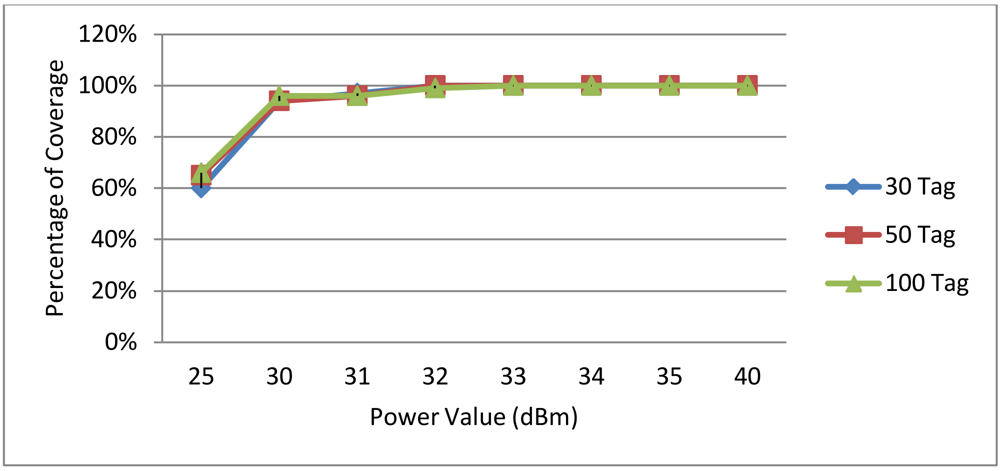


Figure 4.9: Coverage of RFID System Set 2

Results of best number for readers that can cover all system after using genetic algorithm for optimizing the RFID network in 100 iterations are in table 4.12 and illustrated in Figure 4.10.

Table 4.12: Results of Reader Number in Set 2

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	10	10	12
30	7	6	7
31	5	5	7
32	4	5	5
33	4	5	5
34	3	4	4
35	3	3	4
40	1	1	2

However, when we have compared Set 2 results with Set 1 for best number of readers, we can see with 100 tags and segmentation value 8 optimum number of readers is less with low power values (30 dBm and more) .

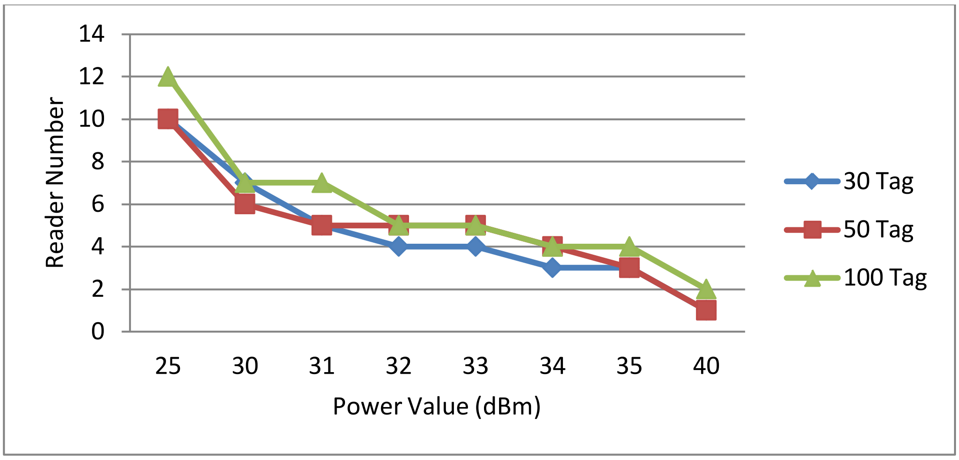


Figure 4.10: Reader Number Set 2

Table 4.13 shows the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in set 2. From the results it is observed with segmentation 8, system reaches 0 interference at low power value (31 dBm) when compared with segmentation 4 in Set 1.

Table 4.13: Results of Interference in the Network

Power (dBm)	Tags number		
	30	**50**	**100**
25	0	0	50
30	0	0	30
31	0	0	0
32	0	0	0
33	0	0	0
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.14 shows the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value in power of 32 dBm with 30 and 50 tags and 33 dBm with 100 tags, this means that after optimization with genetic algorithm with 33 dBm power value, the algorithm reaches to the best answer. There is a slight improvement for fitness value with compared to Set 1 decreased from 34 dBm to 33 dBm. Fitness value results are illustrated in Figure 4.11.

Table 4.14: Results of Fitness Function in Set 2

Power (dBm)	Tags number		
	30	**50**	**100**
25	0.07	0.8	1
30	2.5	3	6
31	9	6	4
32	100	100	50
33	100	100	100
34	100	100	100
35	100	100	100
40	100	100	100

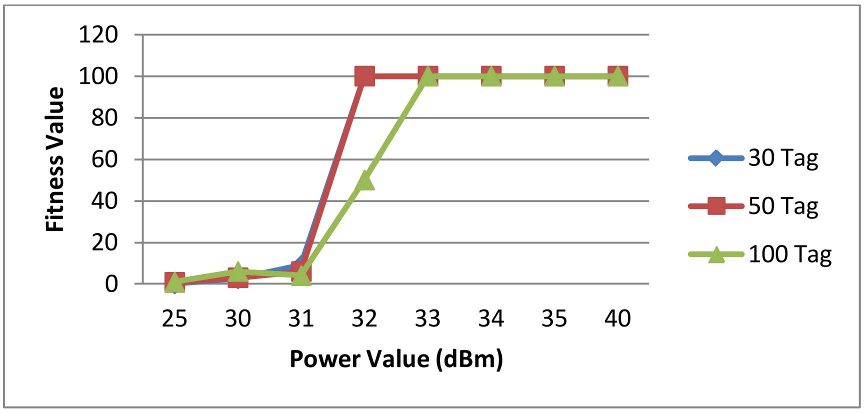


Figure 4.11: Fitness Function Set 2

4.2.3 Set 3: Population 10, Segmentation 16

Fixed parameters used for set3, represented in the Table 4.15.

Table 4.15: Input Parameters for Set 3

Population Size	**Area Size**	**Segmentation**
10	50m * 50m	16

Table 4.16 represents the result for coverage of network according to power value that started from 25 dBm to 40 dBm, the results are expressed in terms of percentage after using Genetic Algorithm for solving RNP in 100 iterations.

Table 4.16: Results of Coverage in Set 3

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	64 %	62 %	69 %
30	100 %	96 %	96 %
31	100 %	98 %	97 %
32	100 %	98 %	99 %
33	100 %	99 %	99 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

As you can see in the Figure 4.12, with increasing power value leads to decrease the number of readers can covered network. On the other hand, the results that obtained for based on tags number shows from power value of 34 dBm, we got 100 percent coverage of network in all 3 types of tags. Results of three graphs is almost identical but in compared with the first set and second set, for 50 and 100 tags we got full coverage in 34 dBm Instead of 32 dBm.

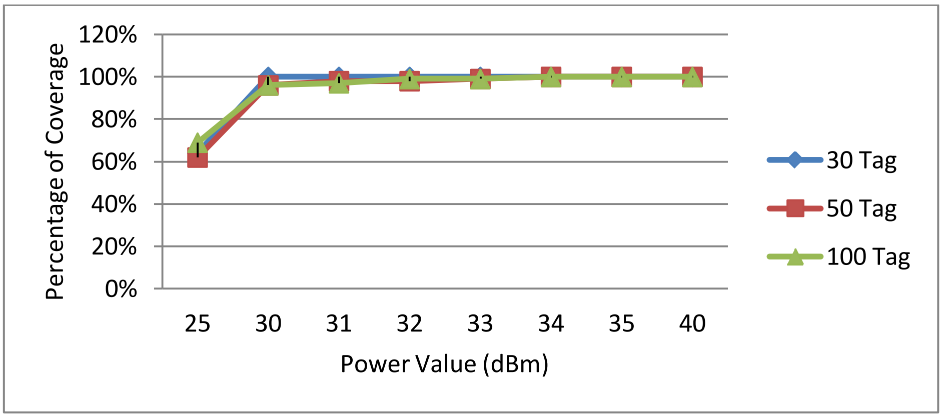


Figure 4.12: Coverage of RFID System Set 3

In Figure 4.12, results are shows in terms of percentage. Result for best number of reader that can cover all system after using genetic algorithm for optimizing the RFID network in 100 iterations in Table 4.17.

In Figure 4.13, the results are shows for the best number of readers after optimization with GA. The graph shows reducing the number of readers in the network by increasing the power value. The number of readers for coverage of RFID system in 100 iterations received from 10 readers to 2 reader for 30 and 50 tags, and 14 readers to 2 reader for 100 tags number in the same area.

Table 4.17: Results of Reader Number in Set 3

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	10	10	14
30	6	6	6
31	4	5	6
32	5	4	5
33	4	4	4
34	4	4	4
35	3	3	4
40	2	2	2

However, in compared with the first and second set, the number of readers that can covered the network are increased due to the high levels of accuracy or increasing the segmentation.

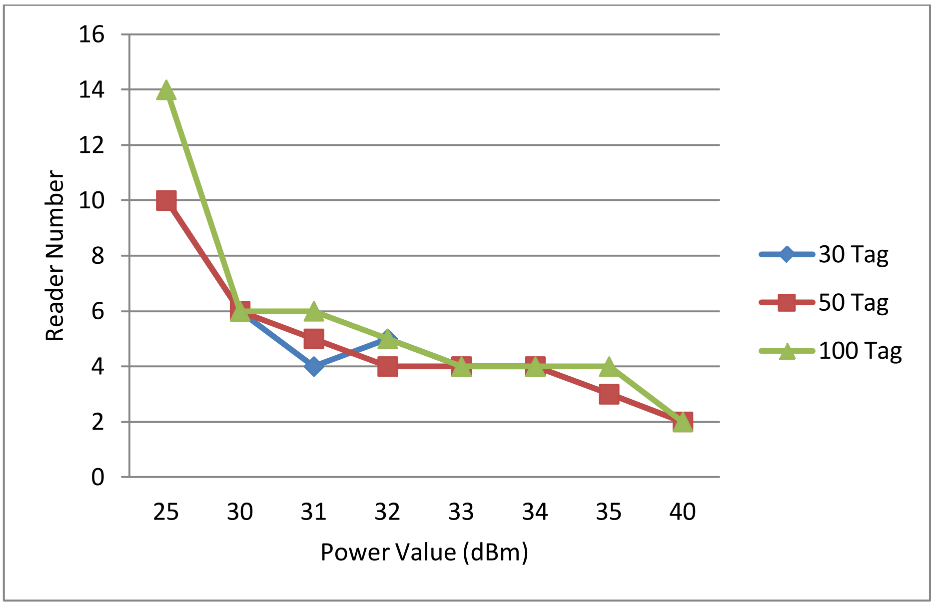


Figure 4.13: Reader Number Set 3

Table 4.18 discuses about the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in set 3. In this table due to the size of area, we do not have any interference for 30 and 50 tags in all powers and for 100 tags, we received zero after 33 dBm of power value. In addition, as a result the value of ITF in set 3 in comparison with Set 1 and 2, approximately remain unchanged.

Table 4.18: Results of ITF in Set 3

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0	6	200
30	0	0	73
31	0	0	20
32	0	0	5
33	0	0	0
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.19 presents the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value in power of 32 dBm for 30 and 50 tags number, this means that the results are obtained after optimization with genetic algorithm in 30 dBm value of power for 30 tags and for 50 and 100 tags with 34 dBm is successful and the algorithm is reached to the best answer. For fitness value in compared with Set 1 and Set 2, the results almost are weaker.

Table 4.19: Results of Fitness Function in Set 3

power	Tags number		
	30	50	100
25	0.08	0.07	0.10
30	100	3.55	6
31	100	6	10
32	100	20	50
33	100	20	50
34	100	100	100
35	100	100	100
40	100	100	100

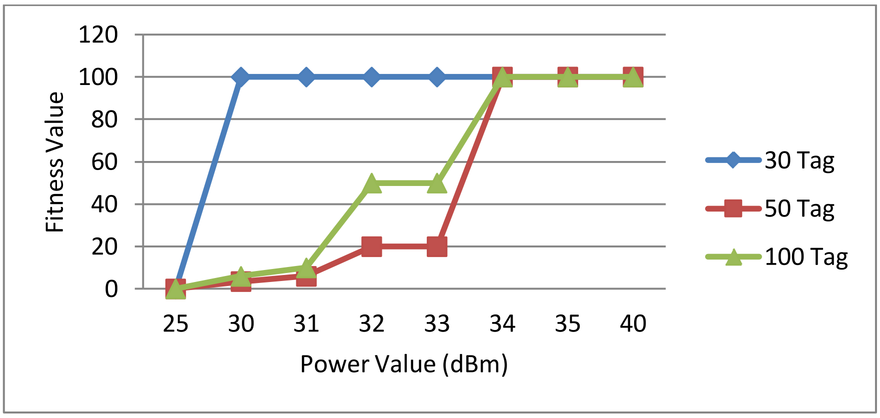


Figure 4.14: Fitness Function Set 3

4.2.4 Set 4: Population 20, Segmentation 4

Fixed parameters used for set 4, represented in the Table 4.20. We change population size from 10 to 20 from this set compared to the previous sets.

Table 4.20: Input Parameters for Set 4

Population Size	**Area Size**	**Segmentation**
20	50m * 50m	4

Table 4.21 shows the result for coverage of network according to power value that started from 25 dBm to 40 dBm, the results are expressed in terms of percentage after using genetic algorithm for solving RNP in 100 iterations.

Table 4.21: Results of Coverage in Set 4

Power (dbm)	**Tags number**		
	30	**50**	**100**
25	63 %	65 %	70 %
30	88 %	90 %	93 %
31	100 %	95 %	98 %
32	100 %	100 %	99 %
33	100 %	100 %	99 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

Figure 4.15 illustrate with increasing power value leads to reduce the number of readers can cover network. On the other hand, the results that obtained for based on tags number shows in power value of 32 (dBm), we got 100 percent coverage of network in 30 and 50 tags number also for 100 tags we have full coverage in 34 dBm. Results of three graphs are almost identical and in compared with the first set, the results are exactly same.

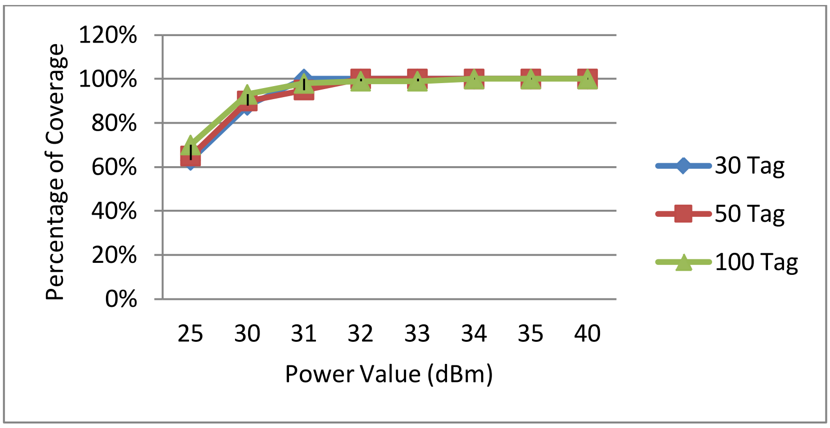


Figure 4.15: Coverage of RFID System Set 4

In Figure 4.15, the results presented in terms of percentage. Result for best number of reader that can cover all system after using genetic algorithm for optimizing the RFID network in 100 iterations in Table 4.22.

Table 4.22: Results of Reader Number in Set 4

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	7	9	13
30	6	7	7
31	7	5	6
32	5	5	5
33	4	5	5
34	4	4	4
35	3	4	4
40	1	1	1

In Figure 4.16, the results are shows for the best number of readers after optimization with GA. The graph shows reducing the number of readers in the network by increasing the power value. The number of readers for coverage of RFID system in 100 iterations received from 7 readers to 1 reader for 30 and 50 tags, and 13 readers to 1 reader for 100 tags and in compared with the first set, The results are approximately the same.

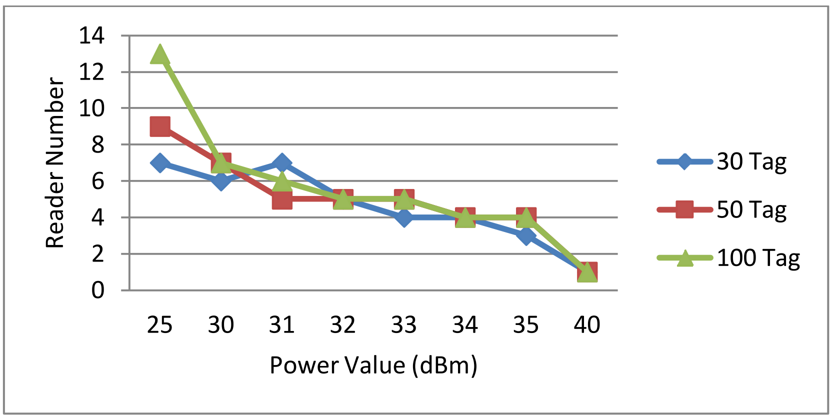


Figure 4.16: Reader Number Set 4

However, in compared with the first set with 10 population size, the number of readers that can covered the network are decreased due to the population that increased in this Set.

Table 4.23 shows the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in set 4. In this table due to the size of area, we do not have any interference for 30 and 50 tags in all powers and for 100 tags, we received zero after 34 dBm of power value and in compared with ITF in Set 1 the results are approximately the same.

Table 4.23: Results of ITF in Set 4

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0	6	50
30	0	0	20
31	0	0	20
32	0	0	5
33	0	0	5
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.24 shows the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value in power of 32 dBm for 30 and 50 tags number, this means that the results are obtained after optimization with genetic algorithm in 33 dBm value of power and for 100 tags with 34 dBm is successful and the algorithm is reached to the best answer. For fitness value in compared with set 1, the results almost remain unchanged.

Figure 4.17 shows the graph of fitness function results after optimization with GA.

Table 4.24: Results of Fitness Function in Set 4

Power (dBm)	Tags number		
	30	**50**	**100**
25	0.08	0.08	0.12
30	0.7	1	1.6
31	10	3	20
32	10	20	50
33	100	100	90
34	100	100	100
35	100	100	100
40	100	100	100

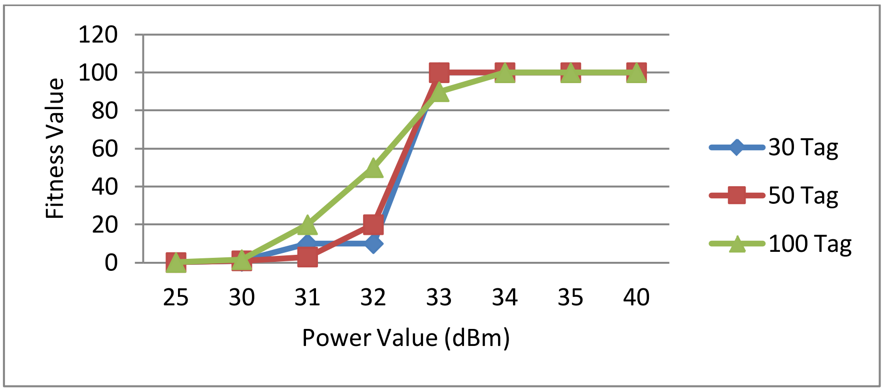


Figure 4.17: Fitness Function Set 4

4.2.5 Set 5: Population 20, Segmentation 8

Fixed parameters used for set5, represented in the Table 4.25.

Table 4.25: Input Parameters for Set 5

Population Size	Area Size	Segmentation
20	50m * 50m	8

Table 4.26 shows the result for coverage of network according to power value that started from 25 dBm to 40 dBm, the results are expressed in terms of percentage after using genetic algorithm for solving RNP in 100 iterations.

Table 4.26: Results of Coverage in Set 5

Power (dBm)	Tags number		
	30	**50**	**100**
25	56 %	66 %	65 %
30	85 %	89 %	93 %
31	90 %	98 %	96 %
32	99 %	99 %	97 %
33	100 %	100 %	100 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

As you can see in the Figure 4.18, increasing power of value leads to decrease the number of readers can covered network. On the other hand, the results obtained for 3 types of tags number shows from power value of 33 (dBm), we got 100 percent coverage of network in all 3 types of tags. Results of three graphs is almost identical.

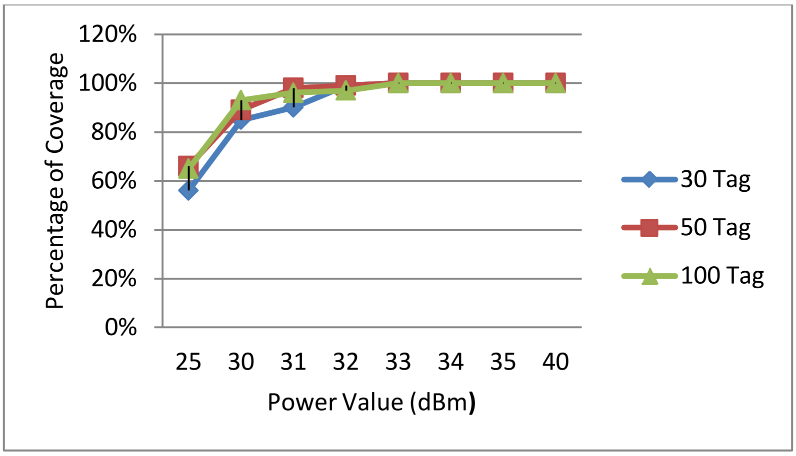


Figure 4.18: Coverage of RFID System Set 5

In Figure 4.18, the results are presented in terms of percentage. Result for best number of reader that can cover all system after using genetic algorithm for optimizing the RFID network in 100 iterations in Table 4.27.

Table 4.27: Results of Reader Number in Set 5

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	7	10	14
30	6	8	8
31	6	6	6
32	5	6	5
33	4	5	4
34	3	4	4
35	2	3	3
40	1	2	2

In Figure 4.19, the results are shown for the best number of readers after optimization with GA. The graph shows reducing the number of readers in the network by increasing the power value. The number of readers for coverage of RFID system in 100 iterations received from 7 readers to 1 reader for 30 tags, for 50 tags received from 10 to 1 and 14 readers to 1 reader for 100 tags number in the same area.

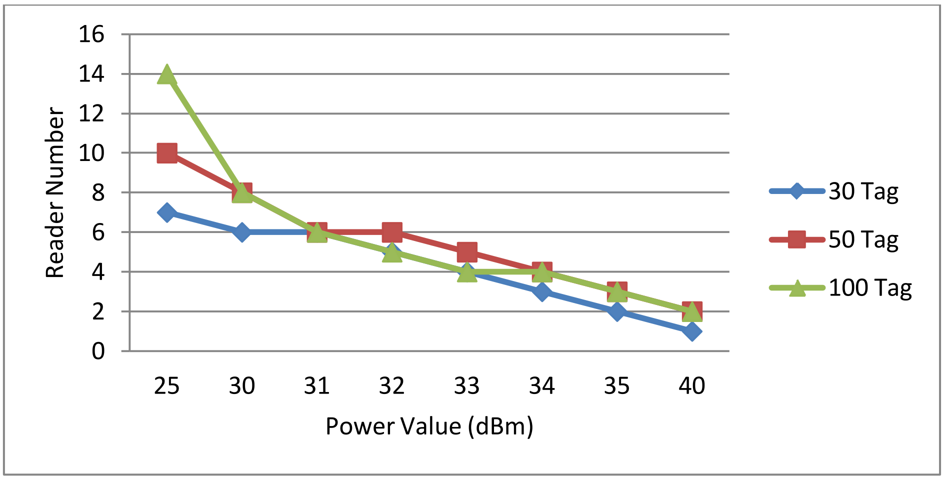


Figure 4.19: Reader Number Set 5

Table 4.28 shows the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in set 5. In this table due to the size of area, we do not have any interference for 30 and 50 tags in all powers and for 100 tags, we received zero after 33 dBm of power value and in compared with ITF with previous sets the results are approximately the same.

Table 4.28: Results of ITF in Set 5

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0	6	200
30	0	0	50
31	0	0	10
32	0	0	10
33	0	0	0
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.29 shows the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value in power of 33 dBm, this means that the results are obtained after optimization with genetic algorithm in 33 dBm value of power was successful and the algorithm is reached to the best answer. For fitness value in compared with set 2, the results almost remain unchanged.

Table 4.29: Results of Fitness Function in Set 5

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	0.05	0.1	0.85
30	0.4	3	2
31	1	6	6
32	10	20	10
33	100	100	100
34	100	100	100
35	100	100	100
40	100	100	100

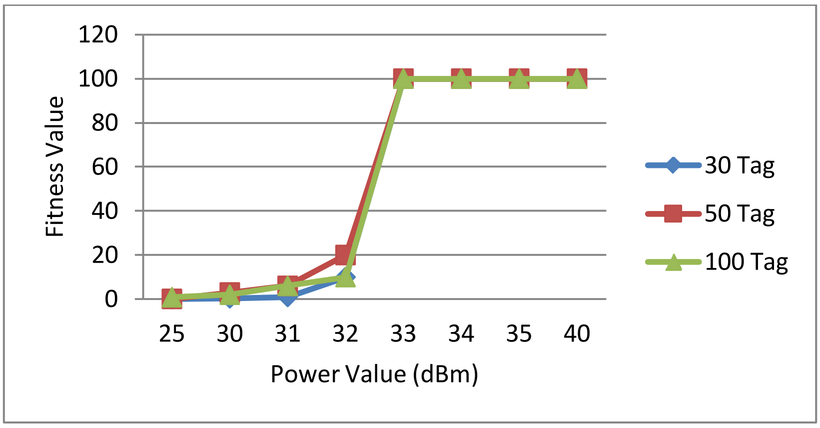


Figure 4.20: Fitness Function Set 5

The Figure 4.20 shows that, the results of set 5 from power values against fitness function. As you can see in this figure, the results for 30, 50 and 100 tags reached to 100 in power value of 33 dBm value of power was successful and the GA algorithm is reached to the best answer. Due to the similar results for 30, 50 and 100 tag number, as you can see line graphs shows in the same line.

4.2.6 Set 6: Population 20, Segmentation 16

Fixed parameters used for set 6, represented in the Table 4.30.

Table 4.30: Input parameters for Set 6

Population Size	Area Size	Segmentation
10	50m * 50m	16

Table 4.31 shows the result for coverage of network according to power value that started from 25 dBm to 40 dBm, the results are expressed in terms of percentage after using genetic algorithm for solving RNP in 100 iterations.

Table 4.31: Results of Coverage in Set 6

Power (dBm)	Tags number		
	30	**50**	**100**
25	75 %	68 %	65 %
30	90 %	90 %	94 %
31	90 %	96 %	97 %
32	97 %	100 %	100 %
33	100 %	100 %	100 %
34	100 %	100 %	100 %
35	100 %	100 %	100 %
40	100 %	100 %	100 %

As you can see in the Figure 4.21, increasing power of value leads to decrease the number of readers can covered network. On the other hand, the results obtained for 3 types of tags number shows from power value of 32 (dBm), we got 100 percent coverage of network in all 3 types of tags. Results of three graphs is almost identical and in compared with the other set, for 100 tags we got better results.

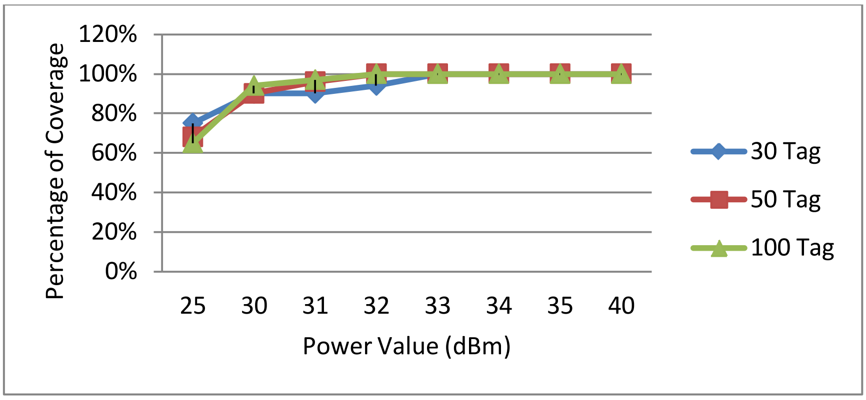


Figure 4.21: Coverage of RFID System Set 6

In Figure 4.21, the results are presented in terms of percentage. Result for best number of reader that can cover all system after using genetic algorithm for optimizing the RFID network in 100 iterations in table 4.32

Table 4.32: Results of Reader Number in Set 6

Power (dBm)	**Tags number**		
	30	**50**	**100**
25	9	10	12
30	5	7	8
31	5	5	7
32	5	5	6
33	4	5	5
34	4	4	4
35	3	3	3
40	1	1	1

In Figure 4.22, the results are shown for the best number of readers after optimization with GA. The graph shows reducing the number of readers in the network by increasing the power value. The number of readers for coverage of RFID system in 100 iterations received from 9 readers to 1 reader for 30 tags, for 50 tags received from 10 to 1 and 12 readers to 1 reader for 100 tags number in the same area.

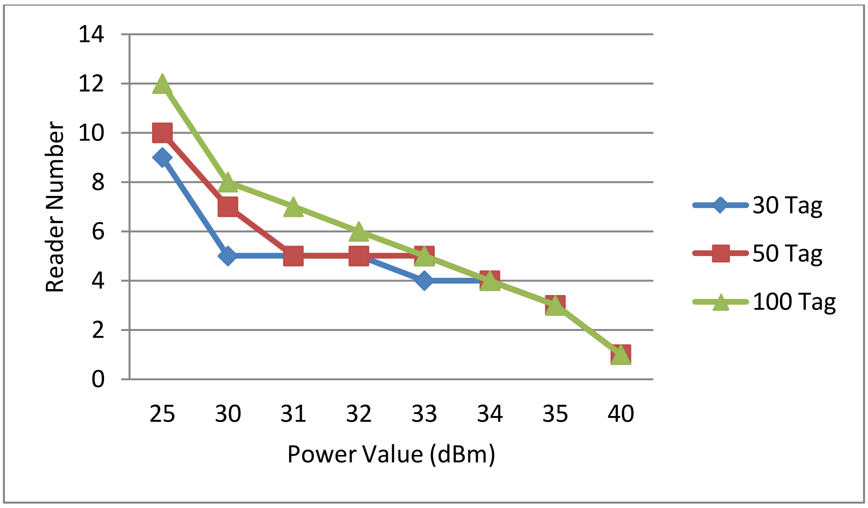


Figure 4.22: Reader Number Set 6

Table 4.33 represents the result for minimum number of interference (ITF) between readers and tags after optimizing RFID readers in set 6. In this table due to the size of area, we do not have any interference for 30 and 50 tags in all powers and for 100 tags, we received zero after 33 dBm of power value and in compared with ITF in previous sets, the results are approximately the same.

Table 4.33: Results of ITF in Set 6

Power (dBm)	Tags number		
	30	50	100
25	0	20	150
30	0	3	50
31	0	0	30
32	0	0	20
33	0	0	0
34	0	0	0
35	0	0	0
40	0	0	0

Table 4.34 shows the result for fitness function base on genetic algorithm that used for optimizing RNP. In this set, the fitness function has reached full value in power of 33 dBm for 30 tags and full value in power of 32 dBm for 50 and 100 tags, this means that the results are obtained after optimization with genetic algorithm in 32 dBm value of power was successful and the algorithm is reached to the best answer. For fitness value in compared with set 3, the results for 50 and 100 tags we got better results.

Table 4.34: Results of Fitness Function in Set 6

Power (dBm)	Tags number		
	30	50	100
25	0.16	0.1	0.78
30	1	1	2.7
31	1	6	10
32	2.5	100	100
33	100	100	100
34	100	100	100
35	100	100	100
40	100	100	100

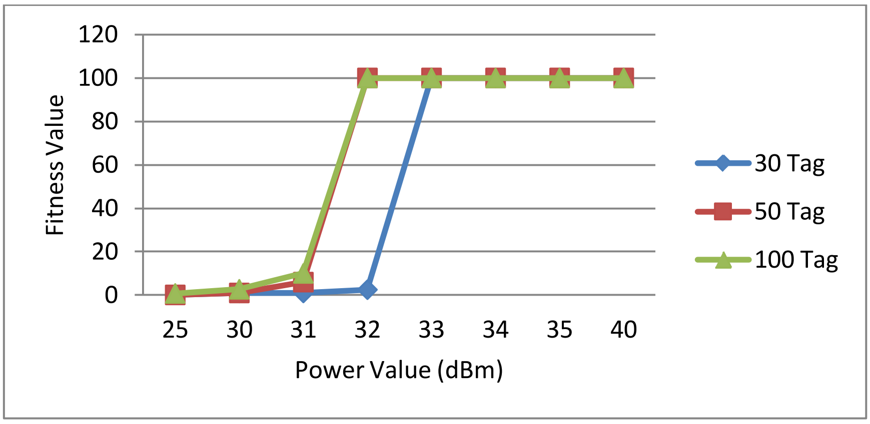


Figure 4.23: Fitness Function Set 6

The Figure 4.23 shows that, the results of set 6 from power values against fitness function. As you can see in this Figure, the results for 30 tags reached to 100 in power value of 33 dBm and the results for 50 and 100 tags reached to 100 in power value of 32 dBm value of power was successful and the GA is reached to the best answer. Due to the similar results for 50 and 100 tag number, as you can see line graphs shows in the same line.

After comparing the obtained results, we conclude that with higher accuracy and population in the program, the results have more effect that is significant and in comparison with major reference with the similar setting, we got the more beneficial reply. Finally, according to the above sets we can say with the increasing in population size also the accuracy of problem (segmentation) we achieve a better answer of RNP for more tags in our research.

4.3 Comparison of GA with Other Methods

In this study Genetic algorithm has been used to optimize RFID network planning with the parameters given in Table 4.35, which are taken from [1]. Given that tags randomly distributed in our program, for provide same condition and comparison with same configuration in the similar work, we run the program 50 times for each tags number and calculate average of these runs as result for Genetic Algorithms (GA) in Table 4.36. The genetic algorithm results in comparison with the results obtained by the four deferent algorithms, that used for optimizing RFID network planning (RNP) with the same input parameters value and settings that used in the [1] based on 50 m^2 working area is shows in Table 4.35.

Table 4.35: Input Parameters

Parameter	Value
Operation Frequency	915 MHz with (wavelength = 0.328 m)
Threshold of Tag (T_t)	-14 dBm
Threshold of reader (T_r)	-80 dBm
Gain of Reader (G_r)	6.8 dB
Gain of Tag (G_t)	3.7 dB
X Area	50 m
Y Area	50 m
Tag numbers	30, 50, 100

Table 4.36: Average of GA-RNP in 50 Randomly Runs

Tags Number	Average of Coverage %	Average of Reader Number	Average of Interference	Power (dBm)
30	100 %	2.92	0.0052	35
50	100 %	2.26	0.0022	40
100	100 %	1.62	0.0028	40

In this experiment, 50 times the program are tested GA-RNP with 30, 50 and 100 tags in the working area. The results are compared based on the number of tags. Tables 4.37 – 4.39 present our results with Genetic Algorithm (GA-RNP) and the other algorithms given in [1] such as Global Particle Swarm Optimization (GPSO), Von Neumann Particle Swarm Optimization (VNPSO) and accordingly, GPSO-RNP and VNPSO-RNP are the version of the proposed PSO embedded with TRE and mutation. The result of GA-RNP in the Tables represent according to the best results from Table 4.36 for compared to other methods.

Table 4.37: Comparison of the Results for 30 Tags

Algorithm	Reader Number	Interference	Power (dBm)	System Coverage (in %)
GPSO	6	0	38.8	100 %
VNPSO	6	0	38.65	100 %
GPSO-RNP	6	0	39.26	100 %
VNPSO-RNP	6	0	39.57	100 %
GA-RNP	3	0	35	100 %

Table 4.38: Comparison of the Results for 50 Tags

Algorithm	Reader Number	Interference	Power (dBm)	System Coverage (in %)
GPSO	6	0	40.5	98 %
VNPSO	6	0	39.6	98 %
GPSO-RNP	7	0	40.31	100 %
VNPSO-RNP	7	0	40.1	100 %
GA-RNP	2	0	40	100 %

Table 4.39: Comparison of the Results for 100 Tags

Algorithm	Reader Number	Interference	Power (dBm)	System Coverage (in %)
GPSO	6	0	40.1	95 %
VNPSO	6	0	40.6	97 %
GPSO-RNP	8	0	40.9	100 %
VNPSO-RNP	8	0	41	100 %
GA-RNP	2	0	40	100 %

All results were presented in form of graphs in Figures 4.24 – 4.26. From table 4.37 it is observed that when number of is less (e.g. 30) using GA optimum number of reader is less of the other algorithms with less power value (35 dBm), and 100 percent coverage in the system.

From Tables 4.38 and 4.39 it is observed that for 50 and 100 tags number of optimum number of readers is much less 2 with power value 40 dBm with 100 percent coverage in the system.

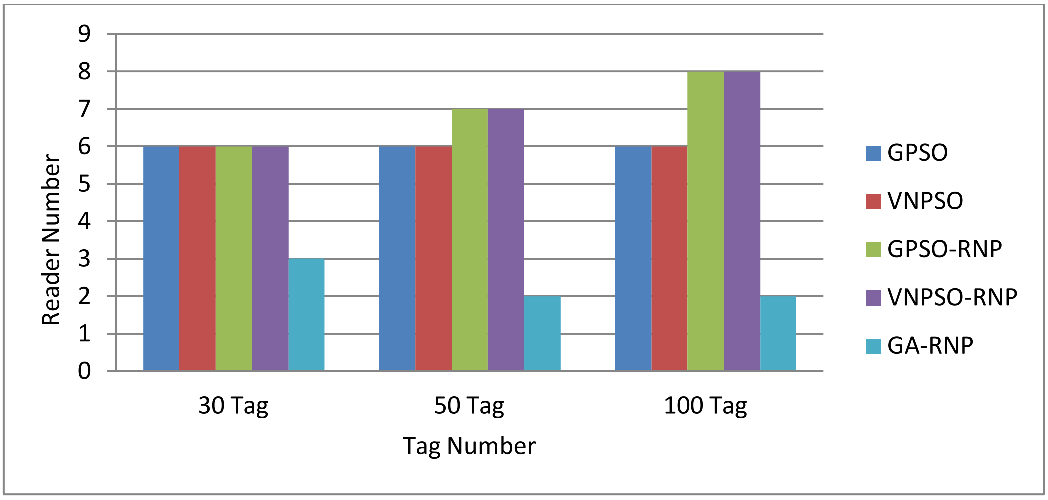


Figure 4.24: Result of Reader in Five Algorithms

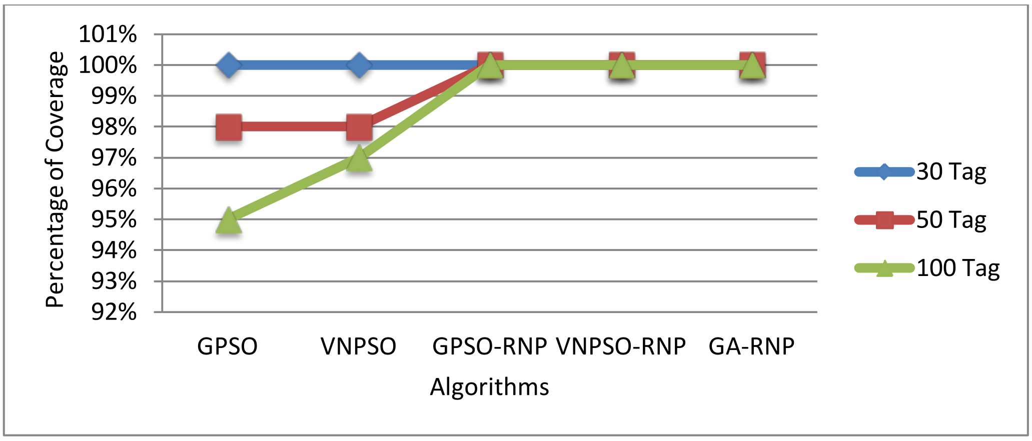


Figure 4.25: Result of Coverage in Five Algorithms

We can see immediately that there are enormous differences in use of GA proportion to other algorithms. As you can see in Figure 4.25, the result of GA-RNP is a much better than other algorithms that they used for RFID network planning. On the other hand, the results that obtain from power in these five Algorithms in Figure 4.26, clearly shown the fact that the power supply for the GA-RNP is lower than other Algorithms that used for RNP.

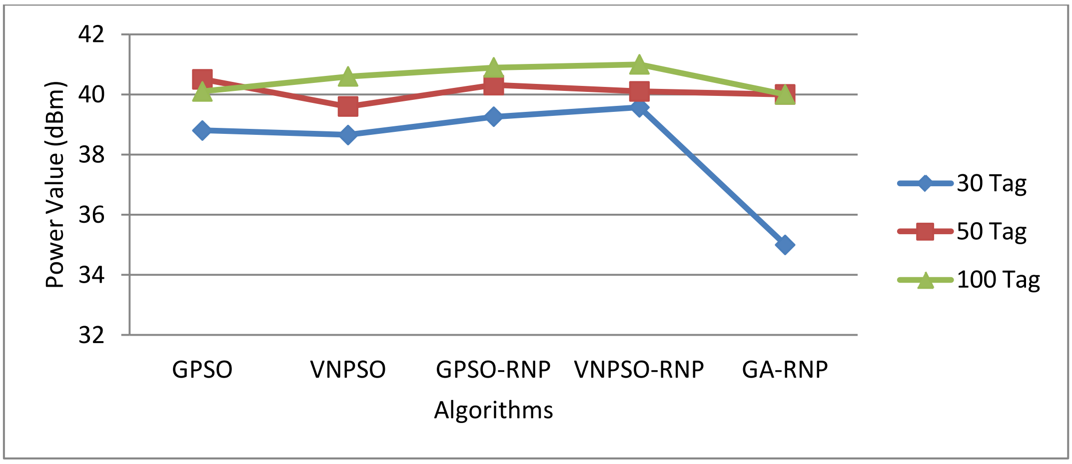


Figure 4.26: Result of Power in Five Algorithms

Chapter 5

CONCLUSION

Conclusively, RNP problem is a high-dimensional nonlinear optimization problem with a large number of variables and uncertain parameters in the large-scale deployment environment. The positioning of RFID systems leads to generate the RFID Network Planning (RNP) problem. The actual position of RFID has acquired the RNP issue which was modeled in this thesis by mathematical formulas of RFID and optimized by Genetic Algorithm as a computational intelligent technique**.**

The findings of this research explored the effectiveness of Genetic Algorithm (GA) for the resolution of the RFID network planning and optimizing this network. This study may be lead to rise up the use of Genetic Algorithm for optimizing RNP in the industry.

After implementing the proposed algorithm, the obtained results of simulation and the results of tested in sets with different inputs in Chapter four from this thesis, clearly have shown a significant decrease in the number of readers that used in the network in compare to other algorithms such as Particle Swarm (PSO) or VNPSO-RNP that used for redundant readers Eliminator [1].

In fact, our proposed method (GA-RNP) is appropriate for scheduling the large scale of radio frequency identification readers that should be deployed in the network.

Finally, the results illuminate an optimized model of RNP for any environment by every number of tags with the optimum number and the position of readers that are needed in the network to receive maximum coverage of RFID system based on genetic algorithms.

As we are living in this modern world, nothing is absolute. Hence, no mathematic model is absolute and perfect. Every moment, better models and new methods and models will be designed and for further research in the future, I try to resolve the modeling and optimizing RFID network planning by using other methods and algorithms like neural networks and fuzzy logic optimization for the optimizing systems and RFID readers to resolve the RNP problem.

REFERENCES

[1] Y. Jiao, M. Shen & J. Zhang, "Optimizing RFID Network Planning by Using a Particle Swar Optimization Algorithm With Redundant Reader Elimination," *IEEE Transaction on Industrial Information*, Vol. 8, No. 4, pp. 900-912, 2012.

[2] Y. Yahui, Y.Wu, M.Xia & Z.Qin, "A RFID Network Planning Based on Genetic Algorithm, "*IEEE, International Conference on Network Security, Wireless Communication and Trusted Computing*, Kansas, USA, Vol. 1, pp. 533-537, 2009.

[3] N. Irfan, E. Yagoub & K.Hettak, "Genetic Algorithm Based Efficient Tag Detection in RFID Reader Networks, "*Computational Intelligence for Measurement Systems and Applications (CIMSA)*, pp. 1-4, 2011.

[4] B. Oscar & H. Chaouchi, "RFID Network Topology Design Based on Genetic Algorithms, "*IEEE International Conference on. IEEE*, Ho Chi Minh, Vietnam, pp. 300-305, 2011.

[5] J. Landt, "The history of RFID," *IEEE Potentials Magazine,* Vol. 24, no. 4, pp. 8-11, 2005.

[6] S. Rao, V. Pavel, F. Sander & F. Lam, "Antenna Design for UHF RFID Tags A Review and a Practical Application," *IEEE Transactions on Antennas and*

Propagation, Vol. 35, no. 12, pp. 3870-3876, 2005.

[7] C.K.M. Lee & T.M. Chan, "Development of RFID Based Reverse Logistics System," *Elsevier International Journal of Expert Systems with Applications,* vol. 36, No. 5, pp. 9299–9307, 2009.

[8] M. Adickes, R. Billo, B. Norman & S. Baneljee, "Optimization of Indoor Wireless Communication Network Layouts," *Springer IIE Transactions,* Vol. 34, No. 9, pp. 823-836, 2002.

[9] H. Chen & Y. Zhu, " RFID Networks Planning Using Evolutionary," in *In Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2008.

[10] C. Chui-Yu, K.Y. Chen & K. Cheng-Hsin, "Optimal RFID Networks Scheduling Using Genetic Algorithm and Swarm intelligence, "*Conference on Systems, Man, and Cybernetics*, San Antonio, TX, pp. 1201-1208, 2009.

[11] GAUKLER & M. Gary, "Item-level RFID in a Retail Supply Chain with Stock-Out-Based Substitution," *IEEE Trans. Ind. Inf,* Vol. 7, No. 2, pp. 362–370, 2011.

[12] P. Nikitin, S. Rao, S. Lam, V. Pillai, R. Martinez & H. Heinrich, "Power Reflection Coefficient Analysis for Complex Impedances in RFID Tag Design, " *IEEE Transactions on Microwave Theory and Techniques,* Vol. 35, No. 9, pp.

2721-2725, 2005.

[13] Y. Chen, S. Horng, R. Run & J. Lai, "A Novel Anti-Collision Algorithm in RFID Systems for Identifying Passive Tags," *IEEE Trans. Ind. Inf,* Vol. 6, No. 1, pp. 105–121, 2010.

[14] L. Painton & J. Campbell, "Genetic Algorithms in Optimization of System Reliability," *lEEE Transactions on Reliability,* Vol. 44, No. 2, pp. 172-178, 1995.

[15] A. Ashkzari & A. Azizi, "Introducing Genetic Algorithm as an Intelligent Optimization Technique," *Applied Mechanics and Materials,* Vol. 568, pp. 793-797, 2014.

[16] F. Alabsi & R. Naoum, "Comparison of Selection Methods and Crossover Operations Using Steady State Genetic Based Intrusion Detection System," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 3, no. 7, pp. 1053-1058, 2012.

[17] S. Gandhi, D. Khan & V. Singh Solanki, "A Comparative Analysis of Selection Scheme," *International Journal of Soft Computing and Engineering (IJSCE),* Vol. 2, No. 4, pp. 131-134, 2012.

[18] H. Chen, Y. Zhu & K. Hu, "Multi-colony Bacteria Foraging Optimization with Cell-to-cell Communication for RFID Network Planning," *Applied Soft*

Computing, Vol. 10, No. 2, pp. 539-547, 2010.

[19] T. Choi, "Coordination & Risk Analysis of VMI Supply Chains with RFID Technology," *IEEE Trans. Ind. Inf,* Vol. 7, No. 3, pp. 497–504, 2011.

[20] V. Alonso, B. Delgado, E. Lopez, C. F. J & A. Gonzalez, "Multiframe Maximum-likelihood Tag Estimation for RFID Anticollision Rotocols," *IEEE Trans. Ind. Inf,* Vol. 7, No. 3, pp. 487–496, 2011.

[21] Q. Guan, Y. Liu, Y. Yang & W. Yu, "Genetic Approach for Network Planning in the RFID Systems," *International Conference on Intelligent*, Jinan - China, Vol. 2, pp. 567-572, 2006.

APPENDICES

Appendix A: MATLAB Code

First part is defining the variables and input parameters:

In this part, we get input parameters from user that include number of tags, room length of our area, segmentation of area according to accuracy of Genetic Algorithm and population size. Other hand, the fixed value of parameters for modeling an RFID network planning such as Gain of readers and tags, landa, power value of reader, Threshold of Tag and reader according to Table 7.1.

Table 7.1: Device Settings

Parameter	Value
Operation Frequency	915 MHz (with wavelength = 0.328 m)
Threshold of Tag (T_t)	-14 dBm
Threshold of reader (T_r)	-80 dBm
Gain of Reader (G_r)	6.8 dB
Gain of Tag (G_t)	3.7 dB

```
clc
clear all
tags_number=input('tags number=');
readers_number=tags_number;
room_lentgh=input('room lentgh (m)=');
room_mesh=input('how many sections room should be?=');
population_size=input('population size=');
P1=input('Power value of Readers(dBm)=');
migirim ta dar crossover moshkel nabashad%%
n=(ceil((log(room_mesh))/(log(2))))*2*(readers_number);
n1=(ceil((log(room_mesh*room_lentgh))/(log(2))));
%(ceil((log(room_mesh))/(log(2))));
first_population=round(rand(population_size,n));
readers_activity=ones(population_size,readers_number);
first_population(1:n,n+1:n+readers_number)=readers_activity
(1:end,1:end);
first_populationR=first_population;
```

We deployed tags, based on randomly coordinates of Tag (X,Y)

```
tags_x=(1-rand(1,tags_number))*room_lentgh;
tags_y=(1-rand(1,tags_number))*room_lentgh;
Gt=3.7;
Gr=6.7;
landa=0.328;
delta=0;
```

```
gamma=.5;
N=2;
Tt=-14;
Tr=-80;
ddd=1;
plot_population=first_population;
```

We have 100 iterations for get the best and optimum result among these iterations and calculate coverage, interference and fitness function for each iteration.

```
for nn=1:100
for i=1:population_size
N_redinitial=0;
ITF_tag=0;
m=1;
m1=0;
cv=zeros(1,tags_number);
for i1=1:n-n1
      if m1<(readers_number*2)
readers_cordinate(i,m1+1)=binvec2dec(first_population(i,m:m
+n1-1))*((room_lentgh)/((2^(n1))-1));
      m1=m1+1;
      m=i1+n1;
      end
end
      for m2=1:readers_number
             if first_population(i,n+m2)==1
             readers_x=readers_cordinate(i,2*m2-1);
             readers_y=readers_cordinate(i,2*m2);
             for k1=1:tags_number
                 if cv(1,k1)==0;
                       tagreaders_distance=(((readers_x-
                       tags_x(k1))^2)+((readers_y-
                       tags_y(k1))^2))^.5;
                       d=tagreaders_distance;
                       L=10*log10(((4*pi/landa)^2)*d^N)+delt
                       a;
                       Pt=P1+Gr+Gt-L;
                 if Pt>=Tt
                       Pb=gamma^2*Pt;
                       for i26=1:readers_number
                              if
                              first_population(i,n+i26)==1
                              %%%baraye readerhaye roshan%%%
readers_x=readers_cordinate(i,2*i26-1);
readers_y=readers_cordinate(i,2*i26);
                              d1=(((readers_x-
                              tags_x(k1))^2)+((readers_y-
                              tags_y(k1))^2))^.5;
                              Pr=Pb+Gt+Gr-
                              (20*log10((4*pi*d1/landa)));
                                  if Pr>=Tr
                                  cv(1,k1)=1;
```

```
                                        end
                                    end
                              end
                         end
                         end
                     end
                     end
               end
               coverage_tag_darhar_fard(i,1)=sum(cv(:));
               coverage(i,1)=coverage_tag_darhar_fard(i,1)*100/tags_n
               umber;

               %%%%mohasebeye tedade redundent reader va tedad reader
               mofid%%%
               N_max(i,1)=sum(first_population(i,n+1:n+readers_number
               ));
               for i27=1:readers_number
                     cv111=zeros(1,tags_number);
                     if  first_population(i,n+i27)==1 %%%baraye
                     readerhaye roshan%%%
                     first_population(i,n+i27)=0;
                     for i28=1:readers_number
                         if  first_population(i,n+i28)==1 %%%baraye
                         readerhaye roshan%%%
                              readers_x1=readers_cordinate(i,2*i28-
                              1);
                              readers_y1=readers_cordinate(i,2*i28)
                              ;
                              for i29=1:tags_number
                                    if cv111(i29)==0
                                    tagreaders_distance1=(((reader
                                    s_x1-
                                    tags_x(i29))^2)+((readers_y1-
                                    tags_y(i29))^2))^.5;
                                    d2=tagreaders_distance1;
```

Calculating $[dB] = 10 \log \left[\left(\frac{4\pi}{\lambda}\right) d^n \right] + \delta\, [dB]$:

```
   L1=10*log10(((4*pi/landa)^2)*d2^N)+delta;
                                    Pt1=P1+Gr+Gt-L1;
                                    if Pt1>=Tt
                                    Pb1=gamma^2*Pt1;
                                    for i30=1:readers_number
                                        if
                                        first_population(i,n+i30)=
                                        =1 %%%baraye readerhaye
                                        roshan%%%
   readers_x11=readers_cordinate(i,2*i30-1);
   readers_y11=readers_cordinate(i,2*i30);
                                             d11=(((readers_x11-
                                             tags_x(i29))^2)+((rea
                                             ders_y11-
                                             tags_y(i29))^2))^.5;
                                             Pr1=Pb1+Gt+Gr-
                                             (20*log10((4*pi*d11/l
                                             anda)));
```

```
                                if Pr1>=Tr
                                cv111(i29)=1;
                                end
                            end
                        end
                        end
                        end
                    end
                end
            end
```

Calculating coverage formula as: $COV = \sum_{t \in TS} \frac{Cv\,(t)}{N_t} * 100\,\%$ and Cv as:

$$Cv\,(t) = \begin{cases} 1, & if\; \exists\, r_1 r_2 \in RS, PT_{r1,r2} \geq T_t \cap PR_{t,r2} \geq\; T_r \\ 0, & Otherwise \end{cases}$$

```
            coverage1(i,1)=sum(cv111(:))*100/tags_number;
            if coverage1(i,1)<coverage(i,1)
                first_population(i,n+i27)=1;
            end
            end
        end
        for m22=1:readers_number
            if first_population(i,n+m22)==1 %%%baraye
            readerhaye roshan: yani agar reader roshan
            bashad%%%
            readers_x=readers_cordinate(i,2*m22-1);
            readers_y=readers_cordinate(i,2*m22);
            for k11=1:tags_number
                    tagreaders_distance11=(((readers_x-
                    tags_x(k11))^2)+((readers_y-
                    tags_y(k11))^2))^.5;
                    d111=tagreaders_distance11;
                    L111=10*log10(((4*pi/landa)^2)*d111^N
                    )+delta;
                    Pt111=P1+Gr+Gt-L111;
                if Pt111>=Tt
                    ITF1(m22,k11)=Pt;
                else
                    ITF1(m22,k11)=0;
                end
            end
            else
            ITF1(m22,1:tags_number)=0;
            end
        end
        for m3=1:tags_number
            ITF_tag(m3,1)=sum(abs(ITF1(1:end,m3)))-
            max(abs(ITF1(1:end,m3)));
        end
        ITF(i,1)=sum(ITF_tag(:));
```

```
        N_r(i,1)=sum(first_population(i,n+1:n+readers_number))
        ;
        N_red(i,1)=N_max(i,1)-N_r(i,1);
        Fi(i,1)=(100/(1+(100-coverage(i,1))^2));
end
sigmaFi=sum(Fi(1:end,1));
if ddd==1
Fi_first=Fi;
N_r_first=N_r;
coverage_first=coverage;
N_max_first=N_max;
N_red_first=N_red;
ddd=ddd+1;
end

[value index]=max(Fi(:));
Fi_n(nn,1)=Fi(index,1);
N_red_n(nn,1)=N_red(index,1);
N_r_n(nn,1)=N_r(index,1);
ITF_n(nn,1)=ITF(index,1);
coverage_n(nn,1)=coverage(index,1);
plot_population(nn,1:end)=first_population(index,1:end);
```

Mutation and Crossover in Genetic Algorithm

```
q_initial=0;
for j=1:population_size
q(j,1)=q_initial+(Fi(j,1)/sigmaFi);
q_initial=q(j,1);
end
%%crossover%%
random_number1=rand(population_size,1);
cross_population=first_population* random_number1;
```

Roulette Wheel selection in GA:

```
for j2=1:population_size   %%roulet weel : be andazeye
jamiat micharhanim hade bala ra entekhab mikonim%%
condition1=0;
for j1=1:population_size
        if condition1==0;
                if random_number1(j2)<=q(j1,1)
cross_population(j2,1:end)=first_population(j1,1:end);
                condition1=1;
                end
        end
end

end                %%%ta inja roulet weel%%%

random_number2=rand(population_size,1);
p_cross=0.6;
matingpool_population1=cross_population* random_number2;
j3=0;
shomareye_satr=0;
for j4=1:population_size
```

```
if random_number2(j4,1)<=p_cross
matingpool_population1(j3+1,1:end)=cross_population(j4,1:en
d);
      shomareye_satr(1,j3+1)=j4;
      j3=j3+1;
end
end
jj=size(shomareye_satr,2)+1;
if rem(jj,2)==0
shomareye_satr(1,jj)=shomareye_satr(1,jj-1);
end
%%%%baraye tashkhis shomare satrhai ke hame 0 hastand dar
matingpool_population va hamchenin baraye az emkane fard
boodan dar avardane tedade satrha%%
condition2=0;
matingpool_population=matingpool_population1;
j6=0;
for j5=1:population_size
if condition2==0
if sum(matingpool_population1(j5,1:end))==0
      if rem(j5,2)==0 %%yani shomarey satr zoj mibashad%%%
matingpool_population(j5,1:end)=matingpool_population1(j5-
1,1:end);
             j6=j5;
      end
      condition2=1;
else
      j6=j5;
end
end
end
j7=size(first_population,2);
randum_number3=randi(j7,1,j6/2);%%%mahale cross: be tetade
jamiate mating pool beine 1 va tedade jhenhaye yek
coromozim adad tolid mikonim baraye mahale cross%
j9=1;
cross_matingpool_population=matingpool_population(1:j6,1:en
d);
for j8=0:(j6/2)-2
cross_matingpool_population(1+2*j8,randum_number3(j9)+1:end
)=matingpool_population(2*j8+2,randum_number3(j9)+1:end);
cross_matingpool_population(2+2*j8,randum_number3(j9)+1:end
)=matingpool_population(2*j8+1,randum_number3(j9)+1:end);
j9=j9+1;
end
aftercross_population=cross_population;
j11=0;
for j10=1:j6
j11=shomareye_satr(1,j10);
aftercross_population(j11,1:end)=cross_matingpool_populatio
n(j10,1:end);
end
%%%%payane crossover%%%%

%%%mutation%%%%%%%%%
P_mutation=(size(first_population,2))/1000;
```

```
randum_number4=rand(population_size,size(first_population,2
));
aftermutation_population=aftercross_population;
for j12=1:population_size
for j13=1:size(first_population,2)
      if randum_number4(j12,j13)<=P_mutation
aftermutation_population(j12,j13)=not(aftercross_population
(j12,j13));
      end
end
end
new_population=aftermutation_population;
Last_population=first_population;
first_population=new_population;

end
```

Output parts

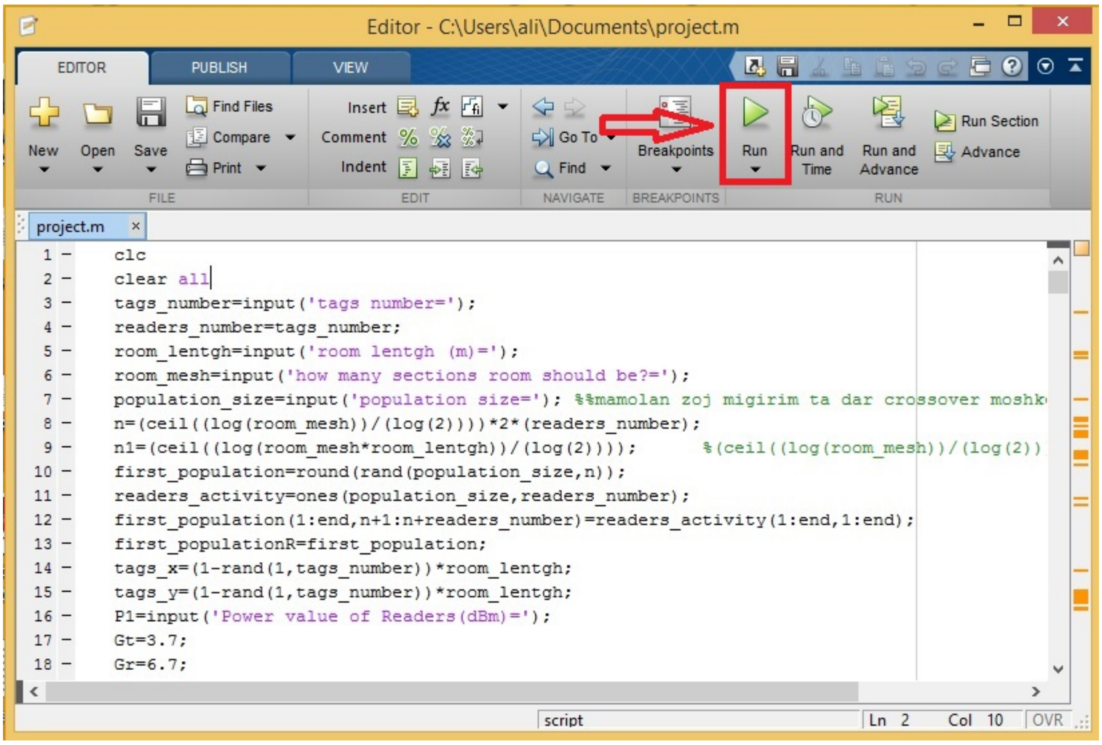
In this part the result that obtained in our problem shows as five datagrams for reader number, fitness function, coverage of system, interference and working area with tags and number of optimum readers after optimization with genetic algorithms.

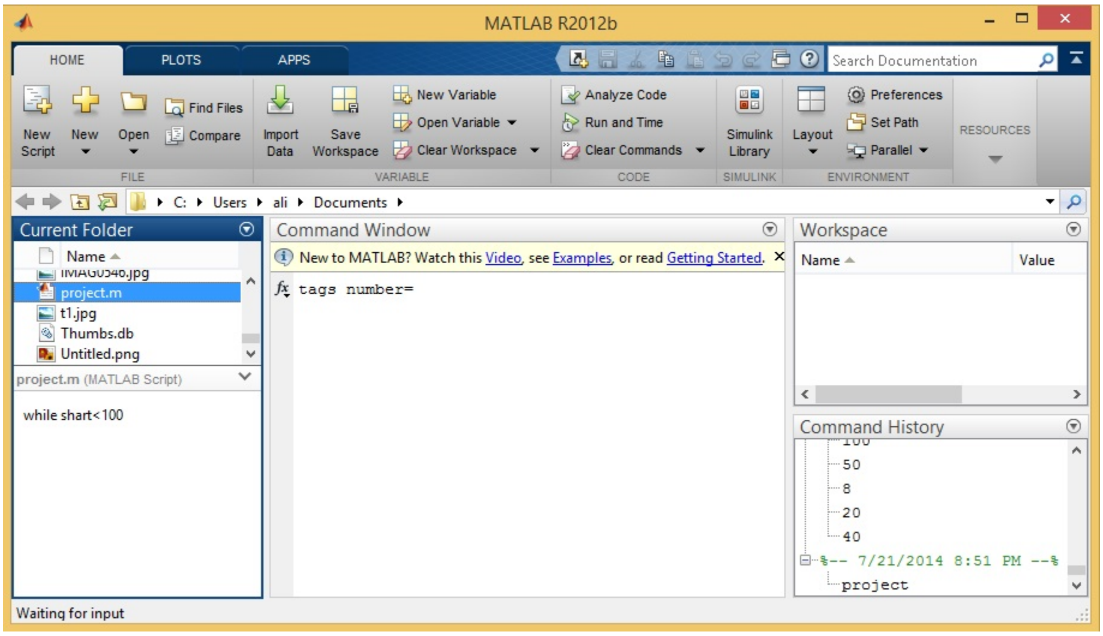
```
      [val idx]=max(coverage_n);
ali=1;
for aydin=1:readers_number
             if plot_population(idx,n+aydin)==1 %%%baraye
             readerhaye roshan: yani agar reader roshan
             bashad%%%
             readers_x(ali)=readers_cordinate(idx,2*aydin-
             1);
             readers_y(ali)=readers_cordinate(idx,2*aydin);
             ali=ali+1;
             end
end
Fi_plot=0;
      coverage_plot=0;
      N_r_plot=0;
      N_red_plot=0;
      ITF_plot=0;
      Fi_nn=Fi_n;
for hh=1:100
      [val ixx]=max(Fi_nn);
      Fi_plot(101-hh)=Fi_nn(ixx);
      Fi_nn(ixx)=0;
      coverage_plot(101-hh)=coverage_n(ixx);
      N_r_plot(hh)=N_r_n(ixx);
      N_red_plot(hh)=N_red_n(ixx);
```

```
      ITF_plot(hh)=ITF_n(ixx);
end
Fi_best=(Fi_plot);
N_red_best=(N_red_plot);
N_r_best=(N_r_plot);
ITF_best=(ITF_plot);
coverage_best=(coverage_plot);
for zz=1:100
N_r_1(zz)=N_r_plot(101-zz);
      N_red_1(zz)=N_red_plot(101-zz);
end
N_r_plot=N_r_1;
      N_red_plot=N_red_1;
      N_red_best=(N_red_plot);
N_r_best=(N_r_plot);
figure(1);
plot(N_r_best);
figure(2);
plot(N_red_best);
figure(3);
plot(Fi_best);
figure(4);
plot(coverage_best);
figure(5);
plot(ITF_best);
figure(6);
plot(tags_x,tags_y,'*')
hold on
plot(readers_x,readers_y,'+')
```

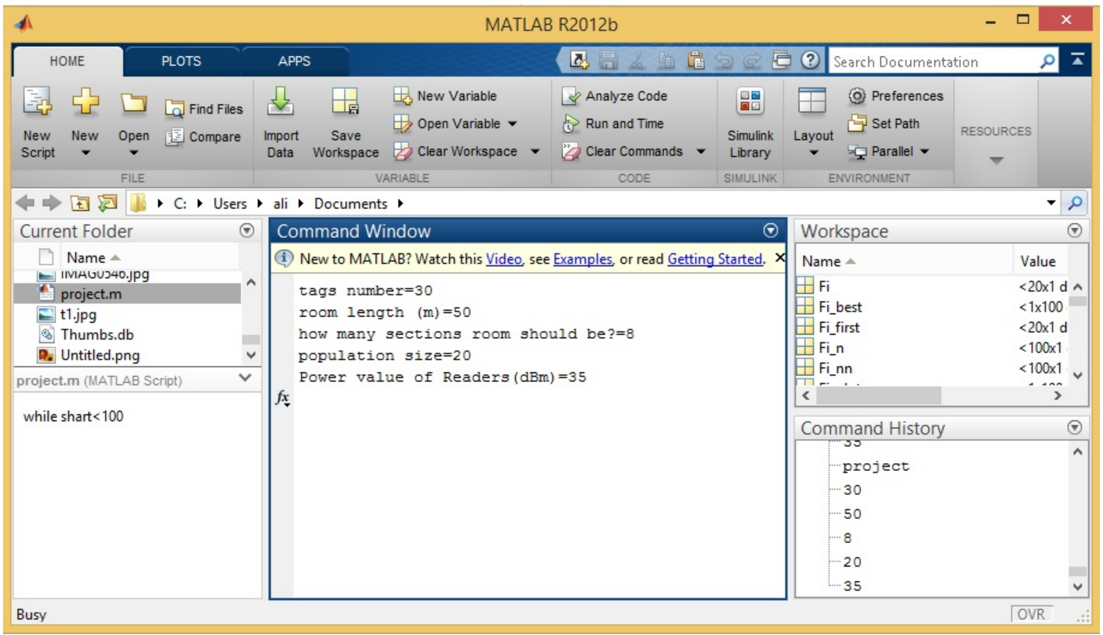
Appendix B: User Guide

In this section, we propose to help the user how run the program in MATLAB. To this aim firstly user should be open the file project.m in the document folder to run the program in the MATLAB space.



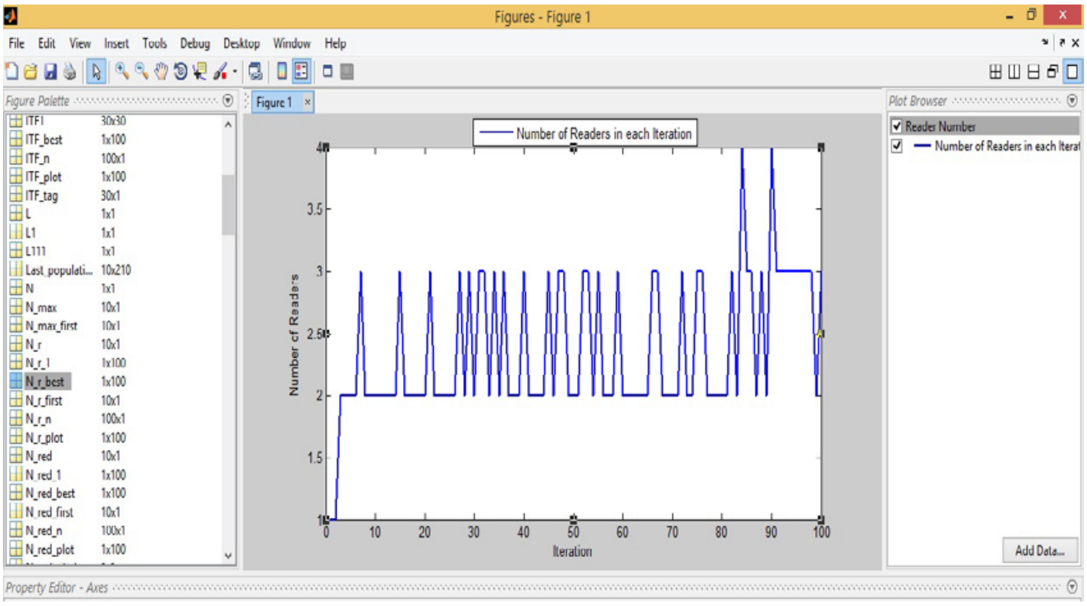


After opening project.m by pressing Run button or F10 in the keyboard the program will be implemented, with define inputs such as tags number, room length (in m^2), power of readers, segmentation, population size that used in genetic algorithm to run the program for solving problem of RNP. Then user should press Enter key to start the program. In this part we have example as followed picture by using 30 tags number, 50 m^2 size of area, 8 segmentation that used for accuracy of genetic algorithm, 20 population size and 35 dBm value for power of readers.

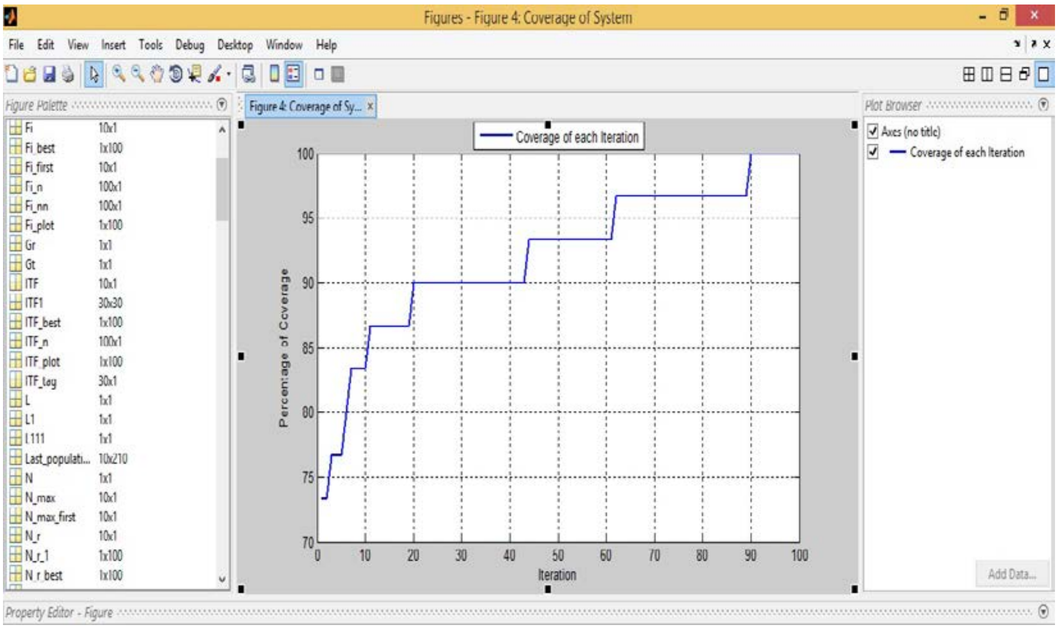


After Enter the inputs value, result that obtained in our problem shows as five datagrams for reader number, fitness function, coverage of system, interference and working area with tags and number of optimum readers after optimization with genetic algorithms as followed pictures.

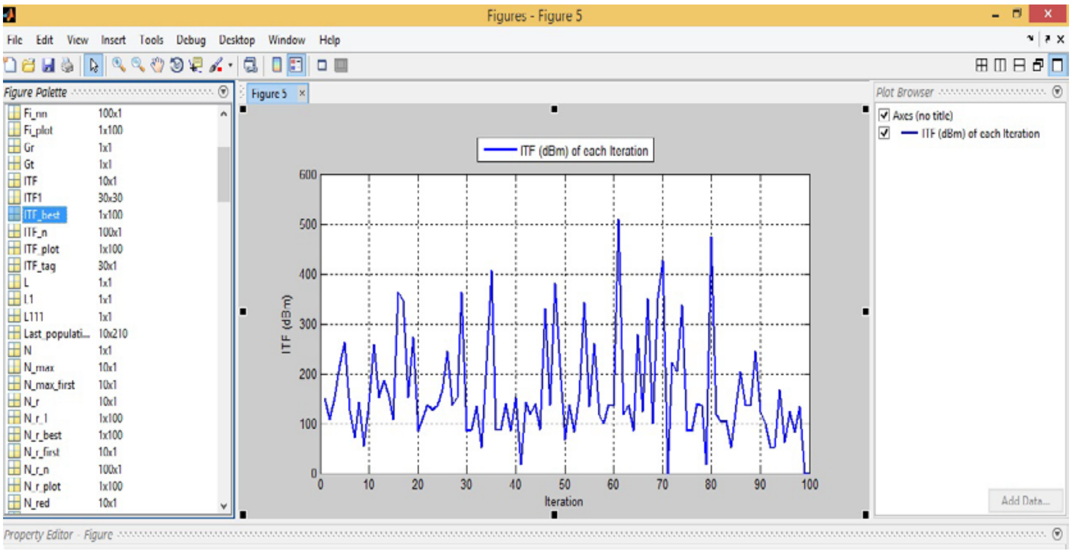
1. Reader Number



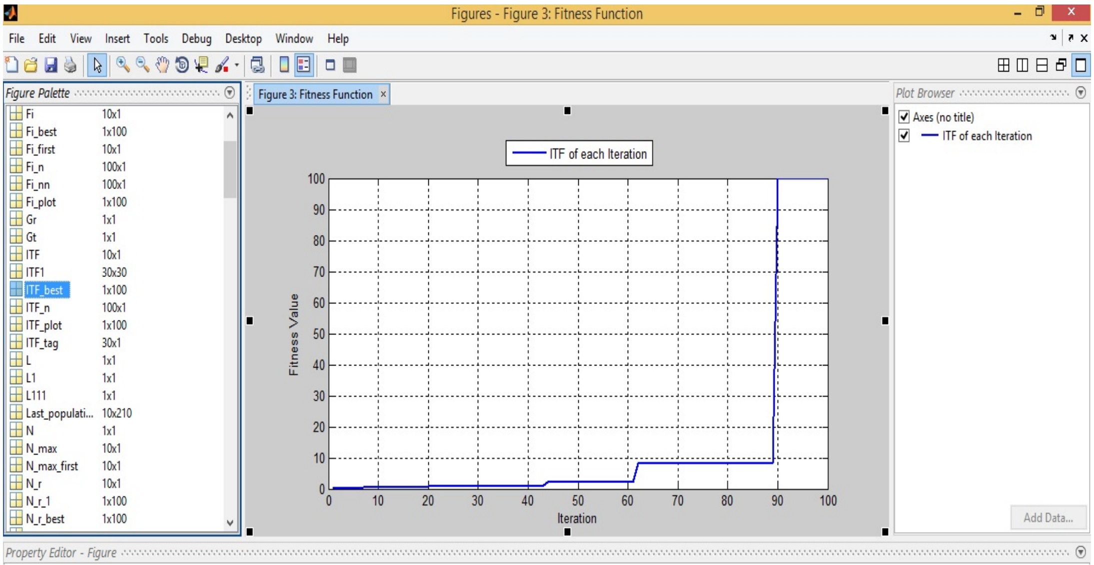
2. Coverage of System



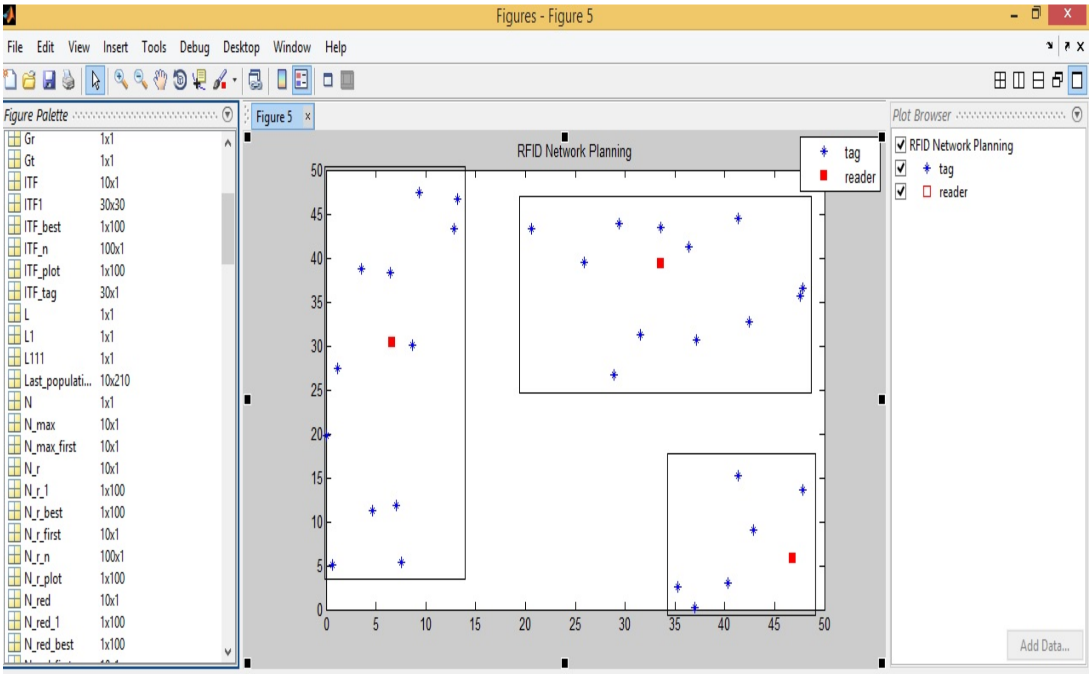
3. Interference (ITF)



4. Fitness Function



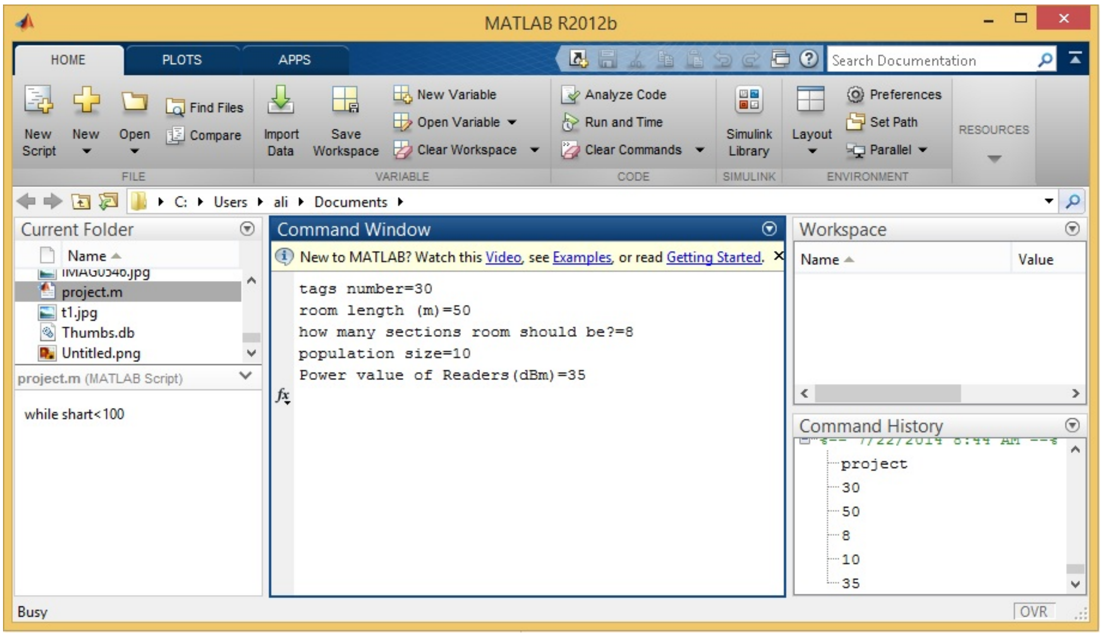
5. Position of Tag and Readers in Optimized Mode



Appendix C: Result of Row Data for One Example

This example represent with 30 number of tags, 50 m^2 of area size, 8 segmentation, 10 population size, 35 dBm power of Readers. This settings and input values given from Set 2, simulation Sets in Chapter 4, Table 4.10-4.14 with 35 dBm value of power and Results of row data for this example present as following.

Input Value:



First population values based on 10 size of population:

```
first_population: 10

  Columns 1 through 10

     0     1     1     0     1     0     0     0     0     0
     0     1     1     1     1     0     1     1     1     0
     1     1     1     1     0     1     0     1     0     1
     0     0     1     0     1     1     0     1     0     1
     0     0     1     0     0     0     1     1     1     1
     0     0     0     0     1     0     1     1     0     0
     1     1     1     1     1     0     1     1     1     1
     0     0     0     0     0     0     1     1     1     1
     0     0     1     0     0     0     0     0     0     0
     0     1     0     1     0     0     0     0     1     0
```

Tag coordinate that are randomly distributed in the area (x , y):

```
tags_x=(1-rand(1,tags_number))*room_lentgh;
```

```
tags_x: 1x30 double =

  Columns 1 through 6

   40.0889   40.2464   33.6580    5.9831   26.4449   29.8015

  Columns 7 through 12

   41.0384    1.5538   29.6272    7.7756   19.2337   31.1694

  Columns 13 through 18

    6.1409   10.7574   26.7523    9.3012    5.0778   28.5381

  Columns 19 through 24

   33.2835   20.1676    4.9005   14.8967   31.1272   13.2522

  Columns 25 through 30

    2.2949   22.8593   22.9947   34.4445   46.4383   40.9010
```

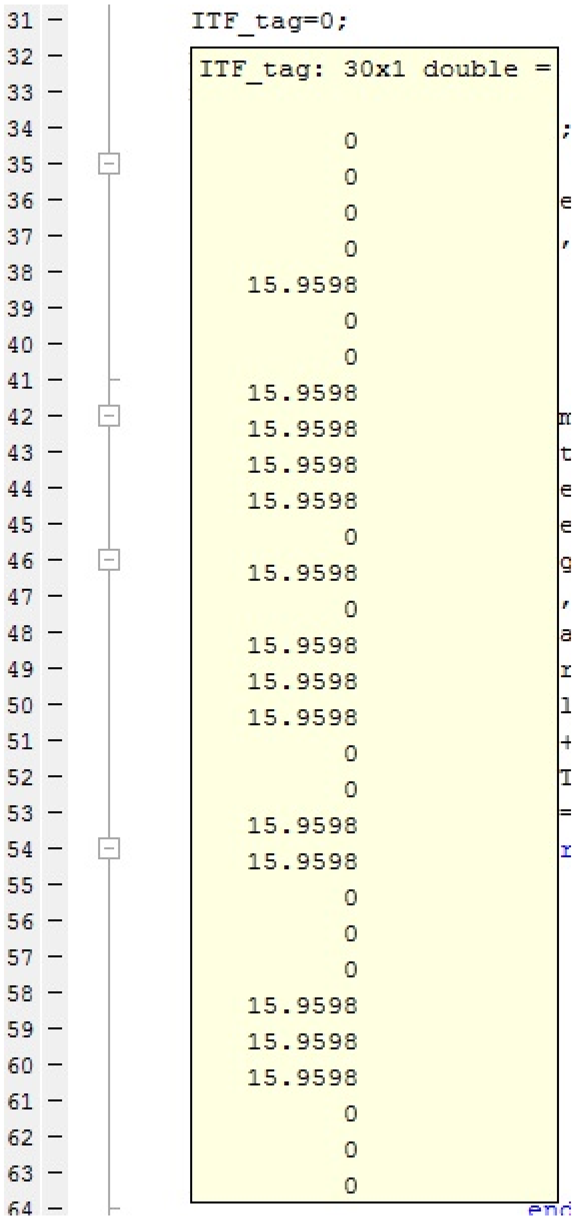
```
tags_y=(1-rand(1,tags_number))*room_lentgh;
```

```
tags_y: 1x30 double =

  Columns 1 through 6

   45.3506   26.8255   49.5334    4.2487   17.8629   49.9290

  Columns 7 through 12

   48.4807   39.5765   27.2517   43.6367   49.5676   13.6460

  Columns 13 through 18

   32.2942   10.9777   28.1672   28.1723   47.5393   47.5184

  Columns 19 through 24

   45.4450   20.2981   37.9458    7.9315    7.1394    1.8194

  Columns 25 through 30

   25.5550   38.9845   38.6896   23.1606   11.8945   32.6216
```

First coverage value:

```
cv=zeros(1,tags_number);
cv: 1x30 double =

  Columns 1 through 12

     1     1     1     1     1     0     0     1     1     1     1     1

  Columns 13 through 24

     1     1     1     1     1     0     0     1     1     1     1     1

  Columns 25 through 30

     1     1     1     1     1     0
```

First value of interference (ITF):



Probability of mutation and selection with Roulette Wheel:

```
random_number2=rand(population_size,1);

random_number2: 10x1 double =

    0.3609
    0.6442
    0.0679
    0.2079
    0.0396
    0.4694
    0.1501
    0.9913
    0.4271
    0.9554
```

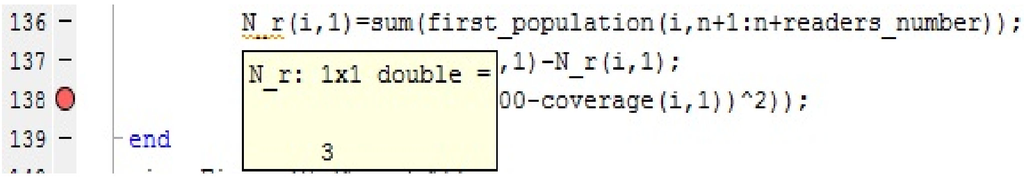
Last population:

```
plot_population: 10

  Columns 1 through 10

     1     0     1     0     0     0     1     0     0     1
     0     1     1     0     0     1     0     1     1     1
     0     0     0     0     1     0     1     0     1     1
     0     0     1     1     0     0     1     0     1     0
     0     0     1     0     0     0     0     0     0     0
     0     0     0     1     1     1     0     0     0     0
     0     0     0     1     1     1     0     1     0     0
     1     1     0     1     0     1     1     1     1     0
     1     0     1     0     0     0     0     0     0     1
     0     1     0     1     1     0     0     1     1     1
```

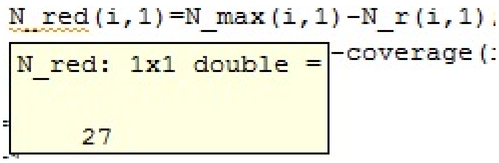
Finally coverage for each iteration, after optimization with GA:

```
coverage_best=(coverage_plot);
coverage_best: 1x100 double =

  Columns 1 through 7

   73.3333   73.3333   76.6667   76.6667   76.6667   80.0000   83.3333

  Columns 8 through 14

   83.3333   83.3333   83.3333   86.6667   86.6667   86.6667   86.6667

  Columns 15 through 21

   86.6667   86.6667   86.6667   86.6667   86.6667   90.0000   90.0000

  Columns 22 through 28

   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000

  Columns 29 through 35

   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000

  Columns 36 through 42

   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000   90.0000

  Columns 43 through 49

   90.0000   93.3333   93.3333   93.3333   93.3333   93.3333   93.3333

  Columns 50 through 56

   93.3333   93.3333   93.3333   93.3333   93.3333   93.3333   93.3333

  Columns 57 through 63

   93.3333   93.3333   93.3333   93.3333   93.3333   96.6667   96.6667
```

Number of optimum reader in the network:



Number of redundant



The results of this example obtained after modeling and optimization, present in Chapter 4, simulation Set 2, Table 4.11-4.14.