OPTIMIZATION OF A RFID NETWORK PLANNING (RNP) MODEL USING A MULTI-OBJECTIVE HYBRID EVOLUTIONARY ALGORITHM, NON-DOMINATED DIFFERENTIAL EVOLUTION ALGORITHM (NSDEA)

A Dissertation
Presented to
The Engineering Institute of Technology

by

Doolun, Ian Shivraj

In Partial Fulfilment
of the Requirements for the Degree
Master of Engineering in
INDUSTRIAL AUTOMATION

Date
25th JUNE 2018

COPYRIGHT © 2018 BY IAN SHIVRAJ DOOLUN
ACKNOWLEDGMENTS

I would like to extend my deepest appreciation and gratitude to the supervisor of this project, Dr. Hadi Harb for his valuable contribution and feedback to the successful completion of this project. I would also like to express my sincere thanks to my family and partner, Ayda Emdadian for the continuous moral support.
# Table of Contents

ACKNOWLEDGMENTS ........................................................................................................... ii  
LIST OF TABLES ................................................................................................................... iv  
LIST OF FIGURES ................................................................................................................... v  
ABSTRACT ............................................................................................................................... vi  
CHAPTER 1. INTRODUCTION ................................................................................................. 1  
CHAPTER 2. LITERATURE REVIEW ....................................................................................... 4  
CHAPTER 3. MATHEMATICAL MODELLING .......................................................................... 9  
  3.1 Notations .......................................................................................................................... 9  
  3.2 Constraints ....................................................................................................................... 9  
  3.3 Decision Variables ......................................................................................................... 10  
  3.4 Fitness Functions ........................................................................................................... 10  
  3.5 Data Set .......................................................................................................................... 11  
CHAPTER 4. METHODOLOGY .............................................................................................. 13  
  4.1 Background .................................................................................................................. 13  
    4.1.1 Differential Evolution (DE) ..................................................................................... 13  
    4.1.2 Non-Dominated Sorting Genetic Algorithm (NSGA-II) ......................................... 16  
  4.2 Proposed Hybrid Algorithm .......................................................................................... 20  
    4.2.1 Non-Dominated Differential Evolution (NSDEA) ................................................... 20  
  4.3 Algorithm Parameter Tuning ......................................................................................... 21  
    4.3.1 Phase 1 .................................................................................................................. 21  
    4.3.2 Phase 2 .................................................................................................................. 27  
CHAPTER 5. PERFORMANCE EVALUATION & DISCUSSION ........................................... 34  
  5.1 Single Objective Optimization ....................................................................................... 34  
    5.1.1 Objective 1: Optimum Tag Coverage .................................................................... 34  
    5.1.2 Objective 2: Load Balance ..................................................................................... 36  
  5.2 Multi-objectives Optimization ....................................................................................... 38  
  5.3 Results Comparison ..................................................................................................... 41  
CHAPTER 6. FUTURE WORK ............................................................................................... 43  
CHAPTER 7. CONCLUSION .................................................................................................. 44  
REFERENCES ....................................................................................................................... 45
LIST OF TABLES

Table 1: List of notations for problem definition .................................................................9
Table 2: Notations of decision variables ...........................................................................10
Table 3: Data set comparison .........................................................................................11
Table 4: Example of Non-Dominated Sorting .....................................................................18
Table 5: Parameter tuning results (Phase 1) .....................................................................27
Table 6: Parameter tuning results (Phase 2) .....................................................................33
Table 7: Final values of tag coverage optimization (50 sample runs) ...............................34
Table 8: Final values of load balance optimization (50 sample runs) ..............................36
Table 9: Pareto optimal solutions ....................................................................................39
Table 10: Results comparison with existing research ......................................................41
Table 11: Potential results comparison ..............................................................................42
LIST OF FIGURES

Figure 1: Proposed RFID network topology .................................................................................. 3
Figure 2: Sample of string with decision variables ...................................................................... 10
Figure 3: RFID topology used in reference paper ......................................................................... 12
Figure 4: RFID topology proposed ............................................................................................... 12
Figure 5: Flowchart of DE steps .................................................................................................. 14
Figure 6: Steps of NSGA-II .......................................................................................................... 17
Figure 7: Steps of NSDEA ............................................................................................................ 20
Figure 8: Convergence pattern for G100\_NP10\_F5\_CR5 .............................................................. 22
Figure 9: Convergence pattern for G100\_NP50\_F5\_CR5 .............................................................. 22
Figure 10: Convergence pattern for G500\_NP10\_F5\_CR5 ............................................................ 23
Figure 11: Convergence pattern for G500\_NP50\_F5\_CR5 ............................................................ 23
Figure 12: Convergence pattern for G100\_NP10\_F5\_CR5 ............................................................ 24
Figure 13: Convergence pattern for G200\_NP10\_F5\_CR5 ............................................................ 24
Figure 14: Convergence pattern for G300\_NP10\_F5\_CR5 ............................................................ 25
Figure 15: Convergence pattern for G400\_NP10\_F5\_CR5 ............................................................ 25
Figure 16: Convergence pattern for G500\_NP10\_F5\_CR5 ............................................................ 26
Figure 17: Convergence pattern for G500\_NP20\_F1\_CR1 ............................................................ 28
Figure 18: Convergence pattern for G500\_NP20\_F1\_CR3 ............................................................ 28
Figure 19: Convergence pattern for G500\_NP20\_F9\_CR1 ............................................................ 29
Figure 20: Convergence pattern for G500\_NP20\_F1\_CR9 ............................................................ 30
Figure 21: Convergence pattern for G500\_NP20\_F1\_CR7 ............................................................ 31
Figure 22: Convergence pattern for G500\_NP20\_F5\_CR7 ............................................................ 31
Figure 23: Convergence pattern for G500\_NP20\_F9\_CR7 ............................................................ 32
Figure 24: Example of Tag Coverage convergence pattern .......................................................... 35
Figure 25: Example of RFID topology considering tag coverage only ......................................... 35
Figure 26: Example of Load Balance convergence pattern .......................................................... 36
Figure 27: Example of RFID topology considering load balance only ........................................ 37
Figure 28: Success of algorithm in achieving a balanced load distribution .................................. 38
Figure 29: Theoretical expectation of Pareto graph ...................................................................... 40
Figure 30: Generated Pareto graph after simultaneous optimization .......................................... 41
ABSTRACT

In today's competitive world, asset tracking technology and allocation decisions are one of the most important aspects of supply chain optimization and RFID technology is helping tremendously in this area. This study considers the implementation of evolutionary algorithms to optimize a RFID network planning problem. The decisions considered are the number and location of RFID tags and readers in order to achieve optimal tag coverage and maintain a balanced load distribution among readers. The two objectives are firstly optimized separately, independent of each other using Differential Evolution (DE) algorithm, which is a robust, reliable and consistent optimization tool. A second model is also developed for simultaneous multi-objectives optimization in which the mechanisms of Differential Evolution (DE) algorithm are integrated into the concept of Non-Dominated Sorting Genetic Algorithm (NSGA-II) to form a hybrid algorithm, Non-Dominated Sorting Differential Evolution Algorithm (NSDEA). To examine the effectiveness of NSDEA, extensive simulation is conducted with regards to the decision variables, objective functions and the associated constraints. Computational experiments are able to produce Pareto solutions which are essential for critical decision making in industries.
CHAPTER 1. INTRODUCTION

Radio Frequency Identification (RFID) is a wireless technology which uses RF energy to pass data between a RFID tag and a corresponding reader. Both the tags and readers have an antenna to communicate with each other. There are two types of RFID tags:

- Active tags, which are self-powered. They usually have a battery or an external power supply in order to receive and transmit information from or to a reader.
- Passive tags, which are powered by the signal transmitted by RFID readers. Passive tags have a different electronic circuitry from active tags which allow them to demodulate the signal and turn them on to decode the information. The electronics is relatively simple and does not require to draw a lot of energy from the reader’s RF energy to operate. However, there is usually a minimum amount of power that the tag needs to obtain from the reader in order to operate, namely the minimum threshold power. Passive tags are mainly and widely used due to its cheaper operation cost and maintenance.

The range of an RFID reader is mainly dictated by the operating frequency [1]. Different regions in the world have different policies regarding the maximum range of readers but the mostly used technology is Ultra High Frequency (UHF) where the frequency can vary between 860 to 956 MHz.

Radio Frequency Identification (RFID) is rapidly growing into an important technology for object identification and tracking applications. The RFID tags are attached to the items to be tracked and the identification number of the item is stored in the tag using integrated circuits. This gives rise to the most challenging RFID network planning (RNP) problem in the large-scale RFID deployment environment. In traditional warehouse management systems, the focus of RNP is usually on single objective, minimum cost or maximum efficiency among others. But the design, planning and scheduling of real life applications are usually involving several goals simultaneously. Hence real RFID networks are to be optimized considering more than one objective. In many situation, it is necessary to know the following information before the deployment of RFID readers in any environment [2]:

- How many items are required to be tracked and therefore how many readers are needed
- The placement of the readers, also considering the orientation
- The parameter settings of each reader

Metaheuristics is a high level of mathematical optimization procedure that is designed to find, generate or select a lower level procedure that may provide a sufficiently good solution to an optimization problem. The application of evolutionary and swarm intelligence algorithms for solving multi-objective RNP (MORNP) has gained significant attention recently.

The work completed over the last two semesters involves the optimization of a RFID network model by considering two minimization objectives simultaneously. Many of the problems that occur in RNP optimization are combinatorial in nature and picking a set of optimal solutions in the case of multi-objective formulations requires an algorithm that can efficiently search the entire search space with less computational effort. In the first part of this project, Differential Evolution (DE) algorithm, which belongs to the category of evolutionary algorithm, is used to obtain optimal or close to optimal solutions by optimizing two objective functions separately and independently. In the second part, DE is integrated into the classic multi-objective NSGA-II algorithm to form a hybrid algorithm, NSDEA to generate a set of Pareto-optimal solutions by optimizing both objective functions simultaneously. Simulation experiments were conducted in Matlab to evaluate the performance of the proposed algorithm.

The RFID network topology considered in this study is a 30m x 30m grid space with six clustered regions, as depicted in Figure 1. Each cluster consists of 50 RFID tags (red-cross markers) and 5 dedicated RFID readers (blue-circle markers). All the tags are passive and the RFID network is operated under UHF, more precisely a frequency of 915MHz.
Figure 1: Proposed RFID network topology

This report is structured as follows; the next section will cover some prior work in the area of evolutionary algorithms and their applications, as well as previous RFID problems which have been tackled. The mathematical modelling of the problem will then be established. The methodology used will be thoroughly explained in Section 4, which involves the mechanisms of the algorithms used and how the tuning of the algorithm was carried out. Furthermore, the results obtained and a comparison of the results with an already published work [19] are discussed in Section 5. Some future work that can be possibly carried out to extend the complexity of the problem is covered in Section 6 and finally, Section 7 will conclude the report.
CHAPTER 2. LITERATURE REVIEW

This study involves the use of evolutionary algorithms as an optimization method and a means for decision making, as well as the optimization of RFID network planning in order to achieve an efficient topology and other goals. This section deals with prior work related to these areas and will be used as support for the choice of evolutionary algorithms to be used for RFID planning.

Evolutionary algorithm includes stochastic search and optimization algorithms originated from the natural evolution principles. These algorithms are robust, adaptive and have found their application in a wide variety of theoretical and practical problems involving search and optimization tasks. EAs are based on a population of encoded tentative solutions which are processed with some evolutionary operators to find a good acceptable solution if not the global optimum one. The optimization process follows the principle of the survival of the fittest to generate successively better results over generations to finally approximate the optimal solutions [3].

Different methodologies for treating optimization problems are available. DE, invented by Storn and Price [4], is one of the evolutionary algorithms (EAs) used to solve optimization problems in continuous space. It is a population based tool that generates trials by adding the scaled difference of two randomly selected vectors to a third randomly selected vector. Then DE recombinates the trial with its parent with a certain probability to generate its offspring. In addition, DE employs a one-to-one spawning logic which allows replacement of an individual only if the offspring outperforms its corresponding parent.

An overview of the major areas of application of Differential Evolution was presented. The DE algorithm showed its capabilities in handling various difficult problems in diverse areas, such as neural network training, clustering, single and multi-objectives optimization. In all the areas, DE was able to tackle the problems with high dimensions successfully [5].

The different variants of Differential Evolution algorithm were compared to other proven particle swarm optimization methods and it was found that DE simply outperformed the corresponding PSO method it was compared with. Although DE has much less parameters than swarm optimization algorithms, DE
still is able to achieve the same level of performance, if not better. Overall, DE is considered to be one of the most powerful search algorithms, both in terms of ease of use and accuracy [6].

Another optimization problem was investigated using evolutionary algorithms. The study described the application of EAs to the optimization of a simplified supply chain in an integrated production-inventory-distribution system. The performance of four EAs, namely Genetic Algorithm, Evolutionary Programming, Evolution Strategies and Differential Evolution was evaluated with numerical simulations. It was found that DE led to better results [7].

RFID-related optimization problems with different objectives have been tackled in the past using various algorithms. Genetic Algorithm (GA) was used for the optimization of directional antenna beam direction in order to maximize reader accuracy and coverage. The work was considered to be a useful tool in logistics and warehouse management. Simulation was performed and the results demonstrated that the reader coverage can be improved by optimizing the placement of the beam angle. Such work can also be used in logistics and warehouse planning to determine the number of RFID readers needed for optimal coverage [8].

An optimization model was developed for the planning of readers positioning in a RFID network. The model was a multi-swarm particle swarm optimizer, namely PS2O. The proposed algorithm, although simple to implement, has proved to be quite effective in solving complex optimization problems. The simulation results showed that the newly developed algorithm performed better than multi-swarm cooperative PSO (MCPSO), canonical PSO, and two evolutionary algorithms, namely genetic algorithm with elitism (EGA) and self-adaptive evolution strategies (SA-ES) in terms of planning RFID networks [2].

A reader placement technique in a departmental store equipped with RFID network was proposed using Particle Swarm Optimization (PSO). The proposed algorithm was able to find the minimal number of readers along with their position with 100% coverage of tagged items. Simulation results demonstrated the algorithm’s success in achieving the optimal solution [9].
Three parameters of a RFID network were tackled, namely the signal interference of RFID readers, the distribution density of the readers and the load of data transmission. Those optimization targets were optimized using the PSO algorithm. A learning strategy was also introduced into the problem such that the PSO algorithm does not fall into a local optimization region and also to increase the efficiency of the algorithm. The simulation results showed that PSO was able to effectively improve the RFID network layout [10].

Another PSO algorithm was developed to optimize a RFID network planning problem. Since it is always difficult to estimate how many readers are needed for a RFID network, the algorithm was designed such that the number of deployed readers in the environment was also optimized in the process. Hence, the algorithm can start with a high number of readers to guarantee full network coverage and end with a lower amount of readers to improve the efficiency and cost of operation. The algorithm took into account four objectives to be optimized, namely the minimization of deployed readers, the maximization of tag coverage, the total power transmitted and interference. The experimental results showed the effectiveness of the proposed algorithm and also proved to be a powerful tool for RFID network planning [11].

A 3D RFID network planning problem was presented and a genetic algorithm with a correction strategy was proposed to solve the problem. The proposed genetic algorithm was tested with network planning problems with different size. The performance of the algorithm was compared to a previously developed algorithm known as micro GA and the results showed that the proposed GA outperformed mGA in the quality of the solutions [12].

Another paper was presented in which a RFID network layout was designed based on Genetic Algorithms. The algorithm produced flexible reader deployment topologies and the location and power level of the RFID readers were optimized. Other factors were also taken into consideration, such as reader interference, the number of tags and reader redundancy [13].

A multi-objective algorithm was implemented to tackle a real-world RFID antenna design problem. The objectives to be optimized were to maximize efficiency and minimize resonant frequency. The algorithm was subjected to simulation experiments and the results were compared to results obtained from other
algorithms such as Ant Colony Optimization (ACO) and Differential Evolution (DE). The comparison indicated that the proposed algorithm was able to obtain competitive results, particularly in the generation of readers with high efficiency [14].

It is important for industrial organisations to have KPIs which reflect their ability to match the customer’s requirements. The industry of automation is rapidly growing and the end users are expecting high-end products to be delivered in a timely fashion and in a professional manner. Therefore, it is critical that multiple objectives are optimized simultaneously during the phase of manufacturing and storage among others. Some algorithms have been developed to tackle multi-objectives and the Non-Dominated Sorting Genetic Algorithm (NSGA), proposed by Srinivas and Deb [15] was one of the first EAs to be developed. The results of NSGA suggested that it can be successfully used to find multiple Pareto-optimal solutions which could be useful for designers and decision makers. However, over the years, the main criticisms of the NSGA approach have been the high computational complexity of non-dominated sorting, lack of elitism and the need for specifying a sharing parameter. Deb et. Al [16] then introduced the concept of Non-Dominated Sorting Genetic Algorithm (NSGA-II), where they addressed the shortcomings of the previously developed algorithm. Also, a selection operator was presented that created a mating pool by combining the parent and offspring populations and selecting the best (with respect to fitness and spread) solutions. Simulation results on difficult test problems showed that the proposed NSGA-II, in most problems, is able to find much better spread of solutions and better convergence near the true Pareto-optimal front.

A paper was presented in which NSGA-II was implemented in order to find better spread of solutions and better convergence close to optimal Pareto fronts. Several benchmark functions were tested using the algorithm to demonstrate its effectiveness and the simulated results showed the high quality and performance of the proposed algorithm [17].

A cooperative multi-objective artificial colony algorithm called CMOABC was proposed to find all the Pareto optimal solutions and to achieve the optimal planning solutions by optimizing four conflicting objectives simultaneously in multi-objective RFID network planning (MORNP). The paper presented a
comparison of the proposed CMOABC and two successful multi-objective techniques, namely the recently developed multi-objective artificial bee colony algorithm (MOABC) and non-dominated sorting genetic algorithm II (NSGA-II). Simulation results showed that CMOABC provided competitive results for planning RFID networks compared to NSGA-II and MOABC in terms of optimization accuracy and computation robustness [18].

Another optimization model was presented for the position planning of RFID readers. Multiple objectives functions were selected for optimization, such as reader interference and economic efficiency among others. All the objectives were then combined using a mathematical equation with each objective being weighed and the combined measure was optimized. The Hierarchical Artificial Bee Colony algorithm (HABC) was proposed to solve this RNP problem and the simulation results obtained were compared to other algorithms. The algorithm was tested with two different RFID topologies, one with a clustered distribution of 100 tags and the other with a uniform distribution of 500 tags. It was found that HABC was able to generate suitable results for solving high-dimensional RNP problem [19].

This work focuses primarily on the implementation of an evolutionary algorithm in order to optimize a RFID network topology. A significant amount of work has been done in the past from fellow researchers regarding the application of evolutionary algorithms in various industrial areas and RFID optimization problems as mentioned in this section. From the literature covered, it is clear that Differential Evolution (DE) and NSGA-II are robust and reliable algorithms for optimization problems. Therefore, a hybrid multi-objective algorithm is proposed in this study to optimize a RFID network planning problem.
CHAPTER 3. MATHEMATICAL MODELLING

In this section, a mathematical optimization model for the RNP problem based on RFID is proposed. All the notations and parameters used for the mathematical programming model are defined in Tables 1 & 2.

The optimization problem involves decision variables and constraints that affect the overall behaviour of the algorithm.

3.1 Notations

Table 1: List of notations for problem definition

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c$</td>
<td>Fitness function denoting optimum tag coverage</td>
</tr>
<tr>
<td>$f_b$</td>
<td>Fitness function denoting load balance</td>
</tr>
<tr>
<td>$i$</td>
<td>Tag index</td>
</tr>
<tr>
<td>$j$</td>
<td>Reader index</td>
</tr>
<tr>
<td>$TS$</td>
<td>Tag set</td>
</tr>
<tr>
<td>$RS_i$</td>
<td>Set of readers which has the tag $i$ in its interrogation region</td>
</tr>
<tr>
<td>$P_{i,j}$</td>
<td>Received power of each tag $i$ in the interrogation region of reader $j$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Minimum power threshold for a tag to be activated</td>
</tr>
<tr>
<td>$P_{rx}$</td>
<td>Received power by a tag</td>
</tr>
<tr>
<td>$P_{tx}$</td>
<td>Transmitted power by a reader</td>
</tr>
<tr>
<td>$A_e$</td>
<td>Effective aperture of the reader</td>
</tr>
<tr>
<td>$r$</td>
<td>The interrogation range of a reader</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength of RFID technology used (based on frequency)</td>
</tr>
<tr>
<td>$C_j$</td>
<td>Assigned tags number to reader $j$ in the working area</td>
</tr>
</tbody>
</table>

3.2 Constraints

- The grid space is restricted to a dimension of 30m x 30m
- All the tags in the working space must be covered
- A minimum tag power threshold, $\delta$ of -10dBm must be established for each tag for it to be covered by a reader
3.3 Decision Variables

Decision variables are variables within an optimization model that one can control. The variables involved in this model only consists of the RFID readers’ parameters and each reader will be characterized by three parameters, its X-coordinate, its Y-coordinate and the radiated power by the reader. Each parameter is randomly generated within the constraints and stored in strings/vectors, as illustrated in Figure 2. Since there are 30 readers and each reader has 3 parameters, the total number of decision variables per string is 90.

<table>
<thead>
<tr>
<th>X_1</th>
<th>Y_1</th>
<th>P_1</th>
<th>X_2</th>
<th>Y_2</th>
<th>P_2</th>
<th>...</th>
<th>X_{30}</th>
<th>Y_{30}</th>
<th>P_{30}</th>
</tr>
</thead>
</table>

Figure 2: Sample of string with decision variables

Table 2: Notations of decision variables

<table>
<thead>
<tr>
<th>X_j</th>
<th>X coordinate of reader j</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y_j</td>
<td>Y coordinate of reader j</td>
</tr>
<tr>
<td>P_j</td>
<td>Power radiated by reader j</td>
</tr>
</tbody>
</table>

3.4 Fitness Functions

The objectives of this optimization problem are formulated through Equations 1-4.

Objective 1: Optimum Tag Coverage (minimization function) [2]

\[ \text{Min } f_c = \sum_{i\in TS} \sum_{j\in RS_i} (P_i^j - \delta) \]  

(1)

TS and RS are the tag and reader set that are deployed in the working area respectively, and RS_i represents the set of readers which has the tag i in its interrogation region.

\( \delta \) is the tag power threshold, taken as -10dBm [20].

The function is formulated as the sum of the difference between the desired power level \( \delta \) and the actual received power \( P_i^j \) of each tag i in the interrogation region of reader j.

The follow equation [20] relates the power received by a tag (Prx) to the power transmitted by a reader (Ptx) and the distance between them, also known as the interrogation range (r) of the reader.

\[ P_{rx} = \frac{P_{tx} A_e}{4\pi r^2}, \text{ where } A_e = \frac{\lambda^2}{4\pi} \]  

(2)
Assumptions:

- The transmitting antenna radiates in all directions with the same power density
- The antenna gains are unity

Since the radiated power which is randomly generated in the strings is in Watts, it is important to convert the value into dBm before using Equation 2. The conversion of power from W to dBm and vice versa can be done using Equation 3 below.

\[ P_{dBm} = 10\log(P_{mW}) \]  

(3)

Objective 2: Load Balance (minimization function) [2]

\[ \text{Min } f_b = \prod_{j=1}^{M} \left( \frac{1}{C_j} \right) \]  

(4)

In the process of the optimization algorithm, the numbers of tags served by jth reader changes as the function of the position and radiated power of the readers.

3.5 Data Set

An initial data set was taken as reference from [2]. However, after further investigation into the origin of the data, some discrepancies were found and therefore the reference data set was considered as being unreliable and unrealistic. Hence, a new data set was established to simulate a more realistic environment.

Table 3: Data set comparison

<table>
<thead>
<tr>
<th></th>
<th>Reference Data Set [2]</th>
<th>New Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of readers</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>Number of tags</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>Reader radiated power range</td>
<td>0.1-2W</td>
<td>1-2W</td>
</tr>
<tr>
<td>Working space</td>
<td>30x30m</td>
<td>30x30m</td>
</tr>
<tr>
<td>Tag Distribution</td>
<td>Uniform</td>
<td>Clustered</td>
</tr>
</tbody>
</table>
Figure 3: RFID topology used in reference paper

Figure 4: RFID topology proposed
CHAPTER 4. METHODOLOGY

As mentioned above, a hybrid algorithm consisting mainly of Differential Evolution (DE) algorithm will be implemented as the solution to the placement of readers in a RFID network. DE is a heuristic optimization method used to solve many optimization problems in real-valued search space. The optimization problem involves decision variables and constraints that affect the overall behaviour of the algorithm. Although metaheuristics such as DE do not guarantee an optimal solution to be found, it consists of a set of defined steps in order to obtain the best solution that is closest to the optimal solution. For an optimal solution to be found, many assumptions need to be made. Unfortunately, real life RFID networks are affected by so many factors, causing some of the theoretical assumptions to be invalid. Hence, the optimal solution is practically impossible to implement in real life. However, the solution closest to the optimal solution is used for optimization and DE fits perfectly in that category. Furthermore, the concept of Non-Dominated Sorting Genetic Algorithm (NSGA-II) will also be used since NSGA-II allows faster convergence to be achieved without compromising the quality of the search process. Finally, the aforementioned algorithms will be combined to form a hybrid and this algorithm will be used for this thesis.

4.1 Background

4.1.1 Differential Evolution (DE)

DE starts with the initialization of a random population, composed of target vectors. Target vectors undergo a mutation process whereby a new set of vectors are formed, called donor vectors. A crossover operation is then carried out by taking into account the target vectors with their corresponding donor vectors in order to generate trial vectors. A round of selection then takes place, where the target vectors are compared to the corresponding trial vectors to determine which one of them will go to the next generation. The three processes are repeated until a final solution is obtained by meeting a termination criterion. Figure 5 shows how classical DE is performed.
The individual steps involved in DE are as follows:

4.1.1.1 Population Initialization

In the first generation, target vectors are initialized by randomly assigning a value to each element/dimension in the vector within boundaries.

Example: Consider a population of NP by NVAR (where NP is the population size and NVAR is the number of elements) dimensional vector and values between 0 and 5 are to be randomly assigned to each element

\[
x_{1,G} = \{1.4, 4.6, 3.1, 2.0, 1.2\}
\]
\[
x_{2,G} = \{3.6, 4.2, 3.3, 0.7, 1.8\}
\]
\[
x_{3,G} = \{1.3, 4.5, 2.4, 1.0, 0.9\}
\]
\[
x_{4,G} = \{0.1, 0.8, 4.9, 5.0, 4.7\}
\]
\[
x_{5,G} = \{1.8, 0.6, 3.7, 0.8, 2.6\}
\]
4.1.1.2 **Mutation**

This step is critical in the generation of mutated strings, known as donor vectors. They are formed by adding the difference of two randomly selected target vectors to a third randomly selected target vector, as shown in Equation 5 [4].

$$y_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}), \quad \text{where } i = 1, \ldots, 5$$  \hspace{1cm} (5)

$F$ is known as the mutation scale factor.

**Example:** Considering a mutation scale factor of 0.5

$$x_{r1,G} = x_{4,G} = \{0.1, 0.8, 4.9, 5.0, 4.7\}$$
$$x_{r2,G} = x_{2,G} = \{3.6, 4.2, 3.3, 0.7, 1.8\}$$
$$x_{r3,G} = x_{1,G} = \{1.4, 4.6, 3.1, 2.0, 1.2\}$$

$$y_{1,G} = x_{4,G} + 0.5(x_{2,G} - x_{1,G}) = \{1.2, 0.6, 5.0, 4.35, 5.0\}$$

4.1.1.3 **Crossover**

The mixture of elements of donor vectors and target vectors to create trial vectors. Crossover can be categorized into binomial crossover and exponential crossover.

In this study, only binomial crossover is implemented. Binomial crossover is an element-by-element operator, as shown in Equation 6 [4].

$$z_{i,k,G} = \begin{cases} y_{i,k,G} & \text{if } R_k \leq C_R \\ x_{i,k,G} & \text{if } R_k > C_R \end{cases}, \quad \text{where } i = 1, \ldots, 5 \text{ & } k = 1, \ldots, 5$$  \hspace{1cm} (6)

$R$ is random number generated between 0 and 1. $C_R$ is known as the crossover rate.

**Example:** Considering a crossover rate of 0.5

$$x_{1,G} = \{1.4, 4.6, 3.1, 2.0, 1.2\}$$
$$y_{1,G} = \{1.2, 0.6, 5.0, 4.35, 5.0\}$$
$$R_1 = 0.6 > 0.5, \quad z_{1,1} = 1.4$$
$$R_2 = 0.4 < 0.5, \quad z_{1,2} = 0.6$$
$$R_3 = 0.8 > 0.5, \quad z_{1,3} = 3.1$$
$$R_4 = 0.7 > 0.5, \quad z_{1,4} = 2.0$$
$$R_5 = 0.2 < 0.5, \quad z_{1,5} = 5.0$$
\[ z_{1,G} = \{1.4, 0.6, 3.1, 2.0, 5.0\} \]

4.1.1.4 Selection

Each target vector competes with the corresponding trial vector to be carried on to the next generation/iteration. The vector with the lowest function value (depending on the objective functions) is copied to the next generation [4].

\[
x_{i,G+1} = \begin{cases} 
  z_{i,G} \text{ if } f(z_{i,G}) > f(x_{i,G}) \\
  x_{i,G} \text{ otherwise}
\end{cases}
\]  \hspace{1cm} (7)

4.1.2 Non-Dominated Sorting Genetic Algorithm (NSGA-II)

Non-Dominated Sorting Genetic Algorithm (NSGA-II) is a fast and elitist multi-objective evolutionary algorithm which uses non dominated sorting. NSGA-II starts with the initialization of a parent population of size \( N \) comprised of decision variables generated randomly within boundaries. The parent population then undergoes any feasible mechanism to generate an offspring population of the same size as the parent population. The parent and offspring populations are merged together to form a population of size \( 2N \). Non-dominated sorting is then performed on the merged population to generate a number of fronts. The fronts, after undergoing a crowding distance sorting, are carried to the next parent population. Basic Genetic Algorithm (GA), consisting of selection, crossover and mutation, is applied to the new parent population to generate a new offspring population. The steps involved in NSGA-II are depicted in Figure 6.
4.1.2.1 Parent Population Initialization

In the first generation, decision variables are randomly generated within their respective boundaries to form a population, $P_t$ of parent vectors.

**Example:**

Consider a population of size NP and vectors of length NV. Assuming NP is 5 and NV is 5 and that all decision variables should be integers generated between 0 and 5 inclusive.

\[
P_{1,G} = \{1,5,3,0,3\} \\
P_{2,G} = \{2,3,4,5,0\} \\
P_{3,G} = \{2,3,3,4,3\} \\
P_{4,G} = \{5,1,1,0,4\} \\
P_{5,G} = \{2,0,0,3,5\}
\]

After the parent population is successfully generated, an offspring population, $Q_t$ is created by disturbing the original parent population. The decision variables of the offspring vectors need to be checked if they are within boundaries.

\[
Q_{1,G} = \{4,0,5,2,2\} \\
Q_{2,G} = \{3,2,1,0,5\} \\
Q_{3,G} = \{0,5,2,3,4\} \\
Q_{4,G} = \{1,1,2,3,0\}
\]
\[ Q_{5,G} = \{1,4,5,0,3\} \]

### 4.1.2.2 Non-Dominated Sorting

The parent and offspring populations are grouped together to form another population, \( R_t \) of size 2NP.

\[
\begin{align*}
R_{1,G} &= \{1,5,3,0,3\} \\
R_{2,G} &= \{2,3,4,5,0\} \\
R_{3,G} &= \{2,3,3,4,3\} \\
R_{4,G} &= \{5,1,1,0,4\} \\
R_{5,G} &= \{2,0,0,3,5\} \\
R_{6,G} &= \{4,0,5,2,2\} \\
R_{7,G} &= \{3,2,1,0,5\} \\
R_{8,G} &= \{0,5,2,3,4\} \\
R_{9,G} &= \{1,1,2,3,0\} \\
R_{10,G} &= \{1,4,5,0,3\}
\end{align*}
\]

Depending on the mathematical formulation and nature of the objective functions to be optimized, the fitness values are calculated for each vector in population \( R_t \).

Let us assume that two objectives are to be optimized simultaneously and the two fitness values have already been evaluated. Objective function F1 needs to be minimized while objective function F2 needs to be maximized. The fitness values are shown in the table below. The figures in the table are only for illustration purposes.

<table>
<thead>
<tr>
<th>Vector</th>
<th>F1 (Minimize)</th>
<th>F2 (Maximize)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>
A vector is said to be dominant if and only if both fitness values of that particular vector are better than the corresponding fitness values of all other vectors. For example, from the table above, it can be observed that Vector 3 dominates all other vectors in the population.

Equal dominance can also occur if there is a trade-off between the fitness values of any two vectors. For example, Vector 2 and Vector 5 can be classified as being equally dominant. F1 of Vector 2 is better than that of Vector 5 whereas F2 of Vector 2 is worse than that of Vector 5.

Knowing these concepts, a sorting process can be carried out among the vectors of the population $R_t$. The result on non-dominated sorting is called fronts. After evaluating the dominance of all possible pairs of vectors in the population, the most dominant vectors are discarded from the population and are placed in a front. The process of non-dominated sorting is repeated until all vectors are assigned a front.

**4.1.2.3 Crowding Distance Sorting**

If a front cannot be accommodated in the next parent population, a mechanism known as crowding distance sorting is performed on the solutions in that particular front.

Example:

Consider a population $R_t$ having 20 vectors, meaning that the next parent population will consist of only 10 vectors.

Front 1 has 3 solution vectors: $3 < 10$ → All solutions of Front 1 are carried to the next parent population

Front 2 has 3 solution vectors: $3+3 < 10$ → All solutions of Front 2 are carried to the next parent population

Front 3 has 6 solution vectors: $3+3+6 > 10$ → Crowding distance sorting → Only best 4 solutions of Front 3 will be carried to the next parent population. Front 3 will therefore have to undergo a crowding distance sorting.

The resulting population of crowding distance sorting will then be the parent population for the next iteration.
4.1.2.4 Genetic Algorithm

For the next iterations until the algorithm meets the stopping criteria, basic GA consisting of selection, crossover and mutation in order is applied to the parent populations in order to generate offspring populations.

4.2 Proposed Hybrid Algorithm

An innovative multi-objective hybrid algorithm is proposed in this study to find the optimal placement of RFID readers in a constrained space. The previously discussed DE is integrated into the concept of NSGA-II to form NSDEA in order to obtain Pareto-optimal solutions.

4.2.1 Non-Dominated Differential Evolution (NSDEA)

Figure 7 is a detailed illustration of the steps involved in the hybrid algorithm.

![Figure 7: Steps of NSDEA](image)

The steps included in the classic NSGA-II to form NSDEA are simply the following:

- DE is carried out on the resulting population of crowding distance sorting. This is done to increase the probability of getting better solution vectors at an earlier stage, as well as increasing diversity.
- The mechanism to produce offspring populations from parent populations is a simple mutation, which corresponds to the same mutation variant used in the DE at the later stage.
4.3 Algorithm Parameter Tuning

Optimum performance of the DE algorithm is sensitive to its control parameters. Fine tuning of the parameters is carried out to obtain the best combination of control parameters. The parameters considered for fine tuning are: population size (NP), number of generations (G), mutation factor (F) and crossover rate (CR).

The tuning for this study was carried out in two phases. The criteria for choosing the parameters can be computation time or convergence pattern that is whether the algorithm is exploring the search space gradually or reaching premature convergence.

The parameter tuning was done based on the tag coverage function only since it has a higher priority over load balance.

Each combination of parameters would generate a convergence graph and the graphs were each assigned a tag for easy interpretation. For example a tag such as G100_NP10_F1_CR3 represents a number of generations of 100, a population size of 10, a mutation factor of 0.1 and a crossover rate of 0.3.

4.3.1 Phase 1

The mutation factor F and crossover rate CR were fixed to 0.5. The number of generations G and population size NP were then varied in the range of 100-500 (increments of 100) and 10-50 (increments of 10) respectively.

Trends:

- For same G, the computation time increases as NP increases since more individuals are required to undergo mutation and crossover and satisfy the objective constraints. No significant change in the convergence value as NP increases with a constant G, as illustrated in Figures 8 & 9 for a G value of 100 and in Figures 10 & 11 for a G value of 500.
Figure 8: Convergence pattern for G100_NP10_F5_CR5

Figure 9: Convergence pattern for G100_NP50_F5_CR5
For same NP, the convergence value gets lower as G increases consistently, as illustrated in Figures 12 through 16.
Figure 12: Convergence pattern for G100_NP10_F5_CR5

Figure 13: Convergence pattern for G200_NP10_F5_CR5
Figure 14: Convergence pattern for G300_NP10_F5_CR5

Figure 15: Convergence pattern for G400_NP10_F5_CR5
Additional criterion: Convergence value must be settled for the last 25% of the total number of generations (375/500), therefore indicating a reliable convergence.

Since NP does not seem to affect the convergence value for a fixed G value, the highest G value of 500 is chosen since it provides lower convergence value and a value of 20 for NP is chosen based on computation time and the above criterion.
Table 5: Parameter tuning results (Phase 1)

<table>
<thead>
<tr>
<th>Tag</th>
<th>Average Convergence Value (based on 5 sample runs)</th>
<th>Average Computation Time (sec) (based on 5 sample runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G100_NP10_F5_CR5</td>
<td>13594.51719</td>
<td>7.565669313</td>
</tr>
<tr>
<td>G100_NP20_F5_CR5</td>
<td>14002.94321</td>
<td>13.27482689</td>
</tr>
<tr>
<td>G100_NP30_F5_CR5</td>
<td>13905.83629</td>
<td>19.16665548</td>
</tr>
<tr>
<td>G100_NP40_F5_CR5</td>
<td>13737.09705</td>
<td>24.49236966</td>
</tr>
<tr>
<td>G100_NP50_F5_CR5</td>
<td>13814.97451</td>
<td>29.66328818</td>
</tr>
<tr>
<td>G200_NP10_F5_CR5</td>
<td>12969.30302</td>
<td>14.17375286</td>
</tr>
<tr>
<td>G200_NP20_F5_CR5</td>
<td>12688.68306</td>
<td>25.41315008</td>
</tr>
<tr>
<td>G200_NP30_F5_CR5</td>
<td>13094.71048</td>
<td>37.31423619</td>
</tr>
<tr>
<td>G200_NP40_F5_CR5</td>
<td>12734.27122</td>
<td>46.85208559</td>
</tr>
<tr>
<td>G200_NP50_F5_CR5</td>
<td>12895.80346</td>
<td>57.6696674</td>
</tr>
<tr>
<td>G300_NP10_F5_CR5</td>
<td>12393.52752</td>
<td>25.00389784</td>
</tr>
<tr>
<td>G300_NP20_F5_CR5</td>
<td>12615.16351</td>
<td>42.1920255</td>
</tr>
<tr>
<td>G300_NP30_F5_CR5</td>
<td>12650.08631</td>
<td>57.63716941</td>
</tr>
<tr>
<td>G300_NP40_F5_CR5</td>
<td>12616.66424</td>
<td>74.15662641</td>
</tr>
<tr>
<td>G300_NP50_F5_CR5</td>
<td>12739.53927</td>
<td>91.00371086</td>
</tr>
<tr>
<td>G400_NP10_F5_CR5</td>
<td>11410.52857</td>
<td>31.48190645</td>
</tr>
<tr>
<td>G400_NP20_F5_CR5</td>
<td>12436.61296</td>
<td>58.40135816</td>
</tr>
<tr>
<td>G400_NP30_F5_CR5</td>
<td>12270.71299</td>
<td>80.37304739</td>
</tr>
<tr>
<td>G400_NP40_F5_CR5</td>
<td>12508.59729</td>
<td>101.9797844</td>
</tr>
<tr>
<td>G400_NP50_F5_CR5</td>
<td>12373.28006</td>
<td>148.3992203</td>
</tr>
<tr>
<td>G500_NP10_F5_CR5</td>
<td>11344.58405</td>
<td>45.30377388</td>
</tr>
<tr>
<td>G500_NP20_F5_CR5</td>
<td>12074.51494</td>
<td>66.83059823</td>
</tr>
<tr>
<td>G500_NP30_F5_CR5</td>
<td>12041.66737</td>
<td>106.0471037</td>
</tr>
<tr>
<td>G500_NP40_F5_CR5</td>
<td>11947.85475</td>
<td>133.7942141</td>
</tr>
<tr>
<td>G500_NP50_F5_CR5</td>
<td>12257.6593</td>
<td>140.4339762</td>
</tr>
</tbody>
</table>

4.3.2 Phase 2

After fixing NP and G to suitable values, both F and CR were varied in the range of 0.1-0.9 (increments of 0.2).

Trends:

- Low values of CR, in the range of 0.1-0.3, are able to provide lower fitness values. The search process is a very slow one but smooth. The search space is explored extensively and the slow process is reflected in the computation time. However, low values of CR do not converge for enough number of generations. Since a G value of 500 was chosen from the first phase, low values of CR are not recommended. Low CR works with very high values of G but this would
increase the computation time drastically. Figures 17, 18 & 19 illustrate the low convergence values but poor convergence pattern.

Figure 17: Convergence pattern for G500_NP20_F1_CR1

Figure 18: Convergence pattern for G500_NP20_F1_CR3
High values of CR tend to indicate premature convergence since the fitness value starts to stagnate and the search process ends at a very low number of generations, as shown in Figure 20.
Figure 20: Convergence pattern for G500_NP20_F1_CR9

- As the value of F goes up, for a constant CR, the gap between fitness values for 2 consecutive generations gets bigger. High values of F do not explore the search space gradually, as depicted in Figures 21, 22 & 23.
Figure 21: Convergence pattern for G500_NP20_F1_CR7

Figure 22: Convergence pattern for G500_NP20_F5_CR7
Figure 23: Convergence pattern for G500_NP20_F9_CR7

- There is no significant difference in the computation time for any pair of F and CR, therefore not a deciding factor in this second phase.
### Table 6: Parameter tuning results (Phase 2)

<table>
<thead>
<tr>
<th>Tag</th>
<th>Average Convergence Value (based on 5 sample runs)</th>
<th>Average Computation Time (sec) (based on 5 sample runs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G500_NP20_F1_CR1</td>
<td>6166.844856</td>
<td>65.49035806</td>
</tr>
<tr>
<td>G500_NP20_F1_CR3</td>
<td>6186.069763</td>
<td>64.75776383</td>
</tr>
<tr>
<td>G500_NP20_F1_CR5</td>
<td>6104.308725</td>
<td>62.69080036</td>
</tr>
<tr>
<td>G500_NP20_F1_CR7</td>
<td>7742.292381</td>
<td>55.07291935</td>
</tr>
<tr>
<td>G500_NP20_F1_CR9</td>
<td>11707.78897</td>
<td>59.97807661</td>
</tr>
<tr>
<td>G500_NP20_F3_CR1</td>
<td>7378.540962</td>
<td>59.41536836</td>
</tr>
<tr>
<td>G500_NP20_F3_CR3</td>
<td>9189.489552</td>
<td>62.22393468</td>
</tr>
<tr>
<td>G500_NP20_F3_CR5</td>
<td>9553.238919</td>
<td>63.10745559</td>
</tr>
<tr>
<td>G500_NP20_F3_CR7</td>
<td>9912.784495</td>
<td>63.84543163</td>
</tr>
<tr>
<td>G500_NP20_F3_CR9</td>
<td>8591.577788</td>
<td>61.70658482</td>
</tr>
<tr>
<td>G500_NP20_F5_CR1</td>
<td>8192.381065</td>
<td>66.63138556</td>
</tr>
<tr>
<td>G500_NP20_F5_CR3</td>
<td>10604.18524</td>
<td>68.2111933</td>
</tr>
<tr>
<td>G500_NP20_F5_CR5</td>
<td>12058.46895</td>
<td>70.57605459</td>
</tr>
<tr>
<td>G500_NP20_F5_CR7</td>
<td>12889.10518</td>
<td>71.45259532</td>
</tr>
<tr>
<td>G500_NP20_F5_CR9</td>
<td>12637.0898</td>
<td>72.27529871</td>
</tr>
<tr>
<td>G500_NP20_F7_CR1</td>
<td>8313.555697</td>
<td>64.61633678</td>
</tr>
<tr>
<td>G500_NP20_F7_CR3</td>
<td>11106.0974</td>
<td>68.91106995</td>
</tr>
<tr>
<td>G500_NP20_F7_CR5</td>
<td>12592.25769</td>
<td>77.35097644</td>
</tr>
<tr>
<td>G500_NP20_F7_CR7</td>
<td>13726.28865</td>
<td>78.81357614</td>
</tr>
<tr>
<td>G500_NP20_F7_CR9</td>
<td>14151.59036</td>
<td>79.7658729</td>
</tr>
<tr>
<td>G500_NP20_F9_CR1</td>
<td>8227.041197</td>
<td>69.41567571</td>
</tr>
<tr>
<td>G500_NP20_F9_CR3</td>
<td>11035.74708</td>
<td>69.13010608</td>
</tr>
<tr>
<td>G500_NP20_F9_CR5</td>
<td>12415.06002</td>
<td>71.65691505</td>
</tr>
<tr>
<td>G500_NP20_F9_CR7</td>
<td>13727.32082</td>
<td>73.28352193</td>
</tr>
<tr>
<td>G500_NP20_F9_CR9</td>
<td>14572.35208</td>
<td>80.02281115</td>
</tr>
</tbody>
</table>

Based on this round of tuning, a compromised value of 0.5 was chosen for both F and CR.
CHAPTER 5. PERFORMANCE EVALUATION & DISCUSSION

This section will cover how the algorithms performed with respect to the fitness functions. The DE parameters used are: number of generations = 500, number of vectors per population = 20, mutation scale factor = 0.5 and crossover rate = 0.5. To assess the quality of the algorithms, a series of computational simulations were conducted. The computational experiments were conducted using a laptop with a 64-bit Windows 10 OS and Intel Core i7-6820HQ processor running at 2.7GHz and using 8GB RAM under Matlab environment.

5.1 Single Objective Optimization

The RFID network was firstly optimized by considering each objective function separately and independently. The DE algorithm was used for the single objective optimization and 50 simulation samples were obtained for each function. For each sample, the final convergence value and the computation time were recorded. The values will be compared to an existing work in a later section of this report.

5.1.1 Objective 1: Optimum Tag Coverage

A RFID network can be considered to be working efficiently when all the tags in the working space are covered at all times and the readers are strategically placed in order to minimize power consumption.

The Differential Evolution (DE) algorithm has been used with a particular goal of achieving an efficient network topology. As previously mentioned, 50 sample runs were conducted and the final convergence values are tabulated in Table 7 below. The average computation time for the algorithm to reach the convergence value over 50 runs was recorded as 60.94 seconds.

Table 7: Final values of tag coverage optimization (50 sample runs)

<table>
<thead>
<tr>
<th>Tag Coverage Optimization</th>
<th>2523.071</th>
<th>2507.806</th>
<th>2492.563</th>
<th>2540.524</th>
<th>2637.566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Coverage Optimization</td>
<td>2599.242</td>
<td>2613.401</td>
<td>2477.206</td>
<td>2434.097</td>
<td>2590.674</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2570.047</td>
<td>2513.401</td>
<td>2494.671</td>
<td>2561.868</td>
<td>2499.962</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2504.201</td>
<td>2509.617</td>
<td>2554.524</td>
<td>2385.097</td>
<td>2527.813</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2577.993</td>
<td>2424.985</td>
<td>2514.852</td>
<td>2515.759</td>
<td>2404.634</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2563.265</td>
<td>2526.485</td>
<td>2564.563</td>
<td>2584.742</td>
<td>2444.524</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2518.111</td>
<td>2494.098</td>
<td>2443.671</td>
<td>2421.662</td>
<td>2564.605</td>
</tr>
<tr>
<td>Tag Coverage Optimization</td>
<td>2484.239</td>
<td>2580.831</td>
<td>2518.747</td>
<td>2520.716</td>
<td>2518.492</td>
</tr>
</tbody>
</table>
Figure 24: Example of Tag Coverage convergence pattern

Figure 25: Example of RFID topology considering tag coverage only
5.1.2 Objective 2: Load Balance

A network with a homogeneous distribution of reader cost can give a better performance than an unbalanced configuration [2]. Thus, in large-scale RFID system, the set of tags to be monitored needs to be properly balanced among all readers.

Similar to objective 1, 50 sample runs were done and the final convergence values are tabulated in Table 8 below. The average computation time for the algorithm to reach the convergence value over 50 runs was recorded as 52.56 seconds.

Table 8: Final values of load balance optimization (50 sample runs)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02E-43</td>
<td>8.88E-44</td>
<td>6.11E-44</td>
<td>1.76E-43</td>
<td>1.01E-43</td>
</tr>
<tr>
<td>1.83E-43</td>
<td>9.93E-44</td>
<td>1.11E-43</td>
<td>2.12E-44</td>
<td>5.31E-44</td>
</tr>
<tr>
<td>6.28E-44</td>
<td>4.01E-44</td>
<td>9.09E-44</td>
<td>4.30E-44</td>
<td>1.96E-44</td>
</tr>
<tr>
<td>8.84E-44</td>
<td>1.14E-43</td>
<td>9.95E-44</td>
<td>7.89E-44</td>
<td>1.15E-43</td>
</tr>
<tr>
<td>1.55E-43</td>
<td>6.24E-44</td>
<td>1.14E-43</td>
<td>1.86E-44</td>
<td>7.64E-44</td>
</tr>
<tr>
<td>5.08E-44</td>
<td>1.34E-43</td>
<td>1.02E-43</td>
<td>2.72E-44</td>
<td>4.94E-44</td>
</tr>
<tr>
<td>1.77E-43</td>
<td>2.20E-44</td>
<td>1.04E-43</td>
<td>1.04E-43</td>
<td>4.51E-44</td>
</tr>
<tr>
<td>6.43E-44</td>
<td>5.55E-44</td>
<td>6.19E-44</td>
<td>9.22E-44</td>
<td>9.81E-44</td>
</tr>
<tr>
<td>1.63E-44</td>
<td>2.88E-44</td>
<td>9.00E-44</td>
<td>5.04E-44</td>
<td>6.17E-44</td>
</tr>
<tr>
<td>1.26E-43</td>
<td>7.08E-44</td>
<td>4.44E-44</td>
<td>5.43E-44</td>
<td>1.04E-43</td>
</tr>
</tbody>
</table>

Figure 26: Example of Load Balance convergence pattern
Figure 27: Example of RFID topology considering load balance only

Figure 28 below demonstrates the success of the algorithm in its capability to optimize the load balance.

From Figure 28, two important conclusions can be drawn:

1. The number of tags covered by a particular reader increased in the process, which can be useful in terms of reader redundancy
2. The distribution of tags to each reader is more balanced at the end of the algorithm’s process, compared to the inefficient distribution at the beginning of the algorithm
5.2 Multi-objectives Optimization

In this second part of the problem, a hybrid approach was formed by integrating NSGA-II with DE to form NSDEA in order to obtain Pareto-optimal solutions. A Pareto solution is one which involves a compromise between two or more objectives and this type of solution is often found in industries where several goals must be achieved at the same time. The hybrid NSDEA has the basic components of DE, which are mutation and crossover. NSGA-II was used to handle selection operation, since there are multiple objectives to handle. Target and trial populations are merged together and NSGA-II operations were carried out (sorting and crowding distance). The results of NSDEA are shown below in Table 9. It can be noted that all solutions are equally dominant, that is no one solution is better than any other when considering both objective functions. This set of results is called a Pareto set, which is critical for decision making.
Table 9: Pareto optimal solutions

<table>
<thead>
<tr>
<th>Tag Coverage</th>
<th>Load Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3635.163</td>
<td>2.06E-42</td>
</tr>
<tr>
<td>2685.027</td>
<td>5.88E-38</td>
</tr>
<tr>
<td>3158.318</td>
<td>1.85E-41</td>
</tr>
<tr>
<td>2774.992</td>
<td>5.38E-39</td>
</tr>
<tr>
<td>2723.522</td>
<td>1.40E-38</td>
</tr>
<tr>
<td>3049.114</td>
<td>9.76E-41</td>
</tr>
<tr>
<td>3152.223</td>
<td>5.92E-41</td>
</tr>
<tr>
<td>2687.313</td>
<td>5.46E-38</td>
</tr>
<tr>
<td>3112.787</td>
<td>6.17E-41</td>
</tr>
<tr>
<td>3275.944</td>
<td>1.75E-41</td>
</tr>
<tr>
<td>2861.191</td>
<td>4.91E-40</td>
</tr>
<tr>
<td>2961.661</td>
<td>2.40E-40</td>
</tr>
<tr>
<td>3413.546</td>
<td>8.44E-42</td>
</tr>
<tr>
<td>2828.692</td>
<td>1.46E-39</td>
</tr>
<tr>
<td>2741.42</td>
<td>9.41E-39</td>
</tr>
<tr>
<td>3037.577</td>
<td>1.29E-40</td>
</tr>
<tr>
<td>3499.032</td>
<td>2.92E-42</td>
</tr>
<tr>
<td>3704.458</td>
<td>7.50E-43</td>
</tr>
<tr>
<td>3440.418</td>
<td>6.92E-42</td>
</tr>
<tr>
<td>2649.675</td>
<td>9.80E-38</td>
</tr>
<tr>
<td>3311.578</td>
<td>1.01E-41</td>
</tr>
<tr>
<td>2802.657</td>
<td>3.05E-39</td>
</tr>
<tr>
<td>2633.914</td>
<td>1.35E-37</td>
</tr>
<tr>
<td>4390.61</td>
<td>6.09E-44</td>
</tr>
<tr>
<td>3872.828</td>
<td>3.34E-43</td>
</tr>
<tr>
<td>2712.245</td>
<td>2.99E-38</td>
</tr>
</tbody>
</table>
Since both objective functions are minimization functions, the expected Pareto graph is as shown in Figure 29. All the points lying on the outer bold line represent the non-dominated and equally dominant Pareto solutions and the rest are the dominated solutions.

![Figure 29: Theoretical expectation of Pareto graph](image)

The application of the hybrid NSDEA was a success, as shown in Figure 30. The generated graph matched the theoretical expectation. The points with red markers represent the Pareto solutions found in Table 9.
5.3 Results Comparison

Based on the results obtained in Tables 7 & 8, the best, worst and mean values for each objective function were noted and tabulated below in Table 10. The generated values for the current problem were compared to an existing research [19], where another evolutionary algorithm, HABC was implemented to solve similar objective functions.

Table 10: Results comparison with existing research

<table>
<thead>
<tr>
<th>Objective</th>
<th>HABC [19]</th>
<th>NSDEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>218.2832</td>
<td>2385.097</td>
</tr>
<tr>
<td>Worst</td>
<td>247.1265</td>
<td>2637.566</td>
</tr>
<tr>
<td>Mean</td>
<td>221.0934</td>
<td>2520.551</td>
</tr>
<tr>
<td>$f_b$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>5.24E-20</td>
<td>1.63E-44</td>
</tr>
<tr>
<td>Worst</td>
<td>3.25E-19</td>
<td>1.83E-43</td>
</tr>
<tr>
<td>Mean</td>
<td>1.39E-19</td>
<td>8.02E-44</td>
</tr>
</tbody>
</table>

It can be argued that HABC produced better results than NSDEA. However, HABC was used to optimize a RFID network topology consisting of 100 tags and 10 readers. In this case, NSDEA is used to optimize a topology of 300 tags and 30 readers, which is 3 times the amount of tags and 3 times the amount of
readers therefore leading to a factor of 9. Table 11 below shows a better value comparison for the tag coverage function if the values obtained from NSDEA are divided by 9. As for the load balance function, the calculation is more complex since it involves a reciprocal function but the value will definitely be able to match the results of HABC.

NSDEA can clearly provide competitive results compared to HABC. One way to guarantee lower and even better function values would be to carry out further tuning of the DE parameters and using a higher number of generations and varying the mutation factor and crossover rate.

Table 11: Potential results comparison

<table>
<thead>
<tr>
<th>Objective</th>
<th>HABC [19]</th>
<th>NSDEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>218.2832</td>
<td>265.011</td>
</tr>
<tr>
<td>Worst</td>
<td>247.1265</td>
<td>293.063</td>
</tr>
<tr>
<td>Mean</td>
<td>221.0934</td>
<td>280.061</td>
</tr>
</tbody>
</table>

It can be safely stated that NSDEA was able to produce remarkable results even though the complexity of the problem was increased to reflect a more realistic approach in terms of the number of RFID tags and readers.
CHAPTER 6. FUTURE WORK

The complexity of the problem can be further increased at a later stage by considering the optimization of more objectives simultaneously with regards to the decision variables and the associated constraints. This can be used as a decision support system for the location of RFID readers in department stores and warehouses. Other objectives could be the minimization of interference, maximization of efficiency and even minimization of total operating cost. The parameters of the DE algorithm can be further tuned to obtain even better fitness values, as well as implementing more than one DE variant. Moreover, a real-world application or operation can be used as a case study to validate the algorithms in terms of applicability.
CHAPTER 7. CONCLUSION

The project is concerned with the optimal placement of tag readers in a RFID network. This report discussed about the work that has been carried out and the methodology used to achieve the objectives. The Differential Evolution (DE) algorithm has once again proved to be one of the best optimization algorithms for combinatorial problems. The performance of the DE algorithm for single-objective optimization was a success for two fitness functions, namely the optimal tag coverage and load balance. A hybrid algorithm, NSDEA was also implemented to optimize both objective functions simultaneously and the choice of the algorithm was derived from the integration of Differential Evolution (DE) into one of the most effective and proven multi-objective algorithms, namely Non-Dominated Sorting Genetic Algorithm (NSGA-II). NSDEA was able to offer the best trade-off solutions between optimal tag coverage and load balance. The hybrid algorithm was able to generate results which matched the theoretical expectations and also comparable to an existing tested algorithm, HABC. It was noted that the generated results were competitive and that better results can definitely be achieved if further tuning of NSDEA parameters is carried out. The aims of the thesis were successfully met in the given timeframe and using minimum resources. Future work seems very promising in the area of evolutionary algorithms and a real-world application can be used as a case study to test the applicability of the algorithms.
REFERENCES


