ARAB ACADEMY FOR SCIENCE, TECHNOLOGY AND MARITIME TRANSPORT (AASTMT)

COLLEGE OF ENGINEERING AND TECHNOLOGY – CAIRO
COMPUTER ENGINEERING DEPARTMENT

A Nested Genetic Algorithm for Ad-hoc Metropolitan Mobile Network Optimization with Fuzzy Fitness

By
Eng. Ahmed Zakaria Nour Mohamed Talha

A thesis submitted to AASTMT in partial
Fulfilment of the requirements for the award of the degree of

MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

Supervised by

Prof. Dr. Ahmed Fahmy Amin
Computer Engineering Dept.
Arab Academy for Science, Technology and Maritime Transport

Prof. Dr. Amr Badr
Department of Computer Science
Faculty of Computers and Artificial Intelligence
Cairo University

2020

ARAB ACADEMY FOR SCIENCE, TECHNOLOGY AND MARITIME TRANSPORT (AASTMT)

COLLEGE OF ENGINEERING AND TECHNOLOGY – CAIRO
COMPUTER ENGINEERING DEPARTMENT

A Nested Genetic Algorithm for Ad-hoc Metropolitan Mobile Network Optimization with Fuzzy Fitness

By

Eng. Ahmed Zakaria Nour Mohamed Talha

A Thesis submitted to AASTMT in partial
Fulfilment of the requirements for the Master's Degree in

COMPUTER ENGINEERING

Prof. Dr. Ahmed Fahmy Amin	**Prof. Dr. Amr Badr**
Supervisor	Supervisor
---	---
Prof. Dr. Abd Elmonem Wahdan	**Prof. Dr. Gamal Selim**
Examiner	Examiner
---	---

DECLARATION

I certify that all the material in this thesis that is not my own work has been identified, and that no material is included for which a degree has previously been conferred on me.

The contents of this thesis reflect my own personal views and are not necessarily endorsed by the University.

(Signature) ....

(Date) ....

ACKNOWLEDGMENTS

In the name of **ALLAH**, the Merciful and prayers and peace be upon His messengers, I thank **ALLAH** the most merciful and compassionate for giving me the power and the desire to finish this thesis and I would like to thank my supervisors, Prof.Dr Ahmed Fahmy Amin and Prof.Dr. Amr Badr, who has given me all the support and guidance I needed as a master's of science student.

This work would not have been possible without the support of my family, friends and my colleagues, also I would like to thank all my teachers from whom I have had the benefit of learning from.

I also would like to express my wholehearted thanks to my family and my friend for their generous support they provided me throughout my entire life and particularly through the process of pursuing the master degree.

ABSTRACT

One of the major culprits that faces Mobile Ad-hoc networks (MANET) is broadcasting, which constitutes a very important part of the infrastructure of such networks. Broadcasting in mobile ad hoc networks (MANETs) is an information dissemination process of sending a message from a source node to all other nodes of the network. Even though it has been studied extensively for wired networks, broadcasting in MANETs poses more challenging problems because of the variable and unpredictable characteristics of its medium as well as the fluctuation of the signal strength and propagation with respect to time and environment. Furthermore, node mobility creates a continuously changing communication topology in which routing paths break and new ones form dynamically.

This thesis presents a nested genetic algorithm (GA) technique with fuzzy logic-based fitness that optimizes the broadcasting capability of such networks. While normally the optimization of broadcasting is considered as a multi-objective problem with various output parameters that require tuning, the proposed system taps another approach that focuses on a single output parameter, which is the network reachability time. This is the time required for the data to reach a certain percentage of connected clients in the network. The time is optimized by tuning different decision parameters of the Delayed Flooding with Cumulative Neighborhood (DFCN) broadcasting protocol. The proposed system is developed and simulated with the help of the Madhoc network simulator and is applied on different realistic real-life scenarios. The results reveal that the reachability time responds well to the suggested system and shows that each scenario responds differently to the tuning of decision parameters.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AODV	Ad-hoc On Demand Distance Vector
API	Application Programming Interface
CPU	Central Processing Unit
DFCN	Delayed Flooding with Cumulative Neighborhood
DSR	Dynamic Source Routing
GA	Genetic Algorithm
MANET	Mobile Ad-hoc Networks
RAD	Random Assessment Delay
RTBD	Record-and-Trust-Based Detection
ZCG	Zone based Routing with Parallel Collision Guided Broadcasting
ZL	Zone Leader

Chapter (I)

INTRODUCTION

CHAPTER I
INTRODUCTION

Mobile Ad-hoc Networks (MANETs) are dynamic types of network consisting of an uncontrolled setup of end-point communication devices known as terminals, which are able of arbitrarily connecting with each other without the need of a base station or a fixed infrastructure [1]. The types of devices that are usually found in MANETs are laptops and smartphones equipped with limited range wireless technologies such as Bluetooth and Wi-Fi (802.11) Fig. I-1 shows the MANET. This, in turn, limits the communication capability of such devices, but allows them to move while communicating.

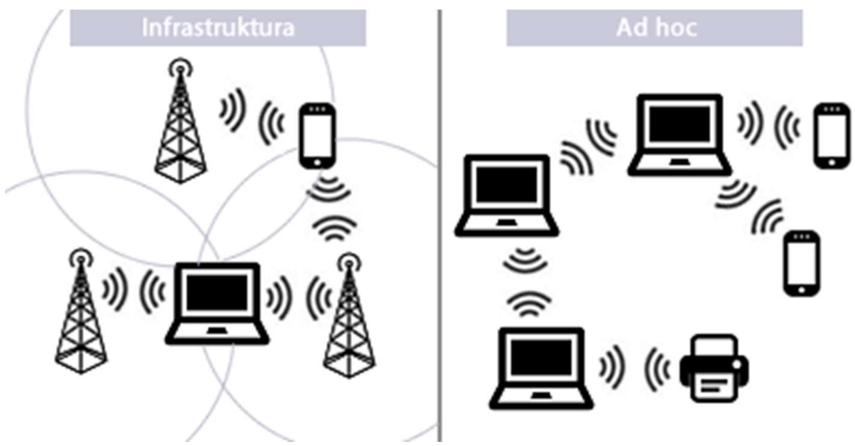

Figure I-1. Infrastructure-based analogy Vs Ad-Hoc analogy

This makes the MANET very unpredictable as it needs to continuously self-reconfigure itself to accommodate these dynamic changes [2]. This is considered a major drawback for the efficiency and effectiveness of the MANETs and, by failing to readjust, link breakage will start to take place and some of the routes can become undiscoverable [3]. For the devices to be able to reach a certain destination, they start sending route discovery requests to their neighboring nodes [4] which, in turn, do the same thing. This results in the network being overwhelmed with an extreme amount of broadcast traffic known as a broadcasting storm [5].

1.1 Broadcasting Definition

Since it is clear that broadcasting plays a very critical role in network discovery and assists the nodes in MANETs in discovering their neighbourhood [6], optimizing it constitutes a major step as it will save both energy and time, especially since most of the devices in the network have limited energy as they are battery powered [7].

Due to the previously mentioned limitations, a key threat known as node 'selfish behaviour' arises in the network, in which the nodes purposely tend to drop the messages that do not target it, in an effort to save its energy [8] [9]. In other words, the nodes are not encouraged to contribute to the forwarding process. This kind of self-regarding behaviour negatively impacts the network because, as already stated, there is no solid infrastructure in MANET and all the nodes rely on the cooperation of other nodes in the network to deliver and forward their messages.

Delayed Flooding with Cumulative Neighbours (DFCN) is a broadcasting protocol that can handle this behaviour and, at the same time, can reduce the number of packets that need forwarding with minimal punitive actions on the final coverage [10] [11] [12]. This is achieved by dropping the forwarded message when enough of the neighbourhood devices have already got it. Also, once a node decides to forward a certain packet, it waits for a specified amount of time before executing this action, which is then cancelled if another node in the network actually forwards the message [13].

1.2 Delayed Flooding with Cumulative Neighbours (DFCN)

DFCN is considered as an event driven technique which is divided into three main parts. The first two parts of the technique are responsible of handling the **outcoming events** like:

(i) Reception of New Messages.

(ii) Detection of New Neighbours

The third part is responsible of the decision making for the emission. The behaviour resulting from receiving a message is known as **"Reactive Behaviour"** and the behaviour resulting from the discovery of new neighbours is known as **"Proactive Behaviour"**.

Let us assume d_1 and d_2 are two nodes within reachable neighbourhoods. When a message is sent from d_1 to d_2, the packet gets attached to the set $N(d_1)$. Then, during reception, d_2 will be able to know that each node inside $N(d_1)$ has received the packet. In this case, the number of nodes that which has probably not yet received the packet, can be calculated by subtraction $N(d_1)$ from $N(d_2)$. If d_2 decides to re-emit the packet, the effective number of newly reached nodes can be maximized by means of the heuristic function:

$$h(d_2, d_1) = |\, N(d_2) - N(d_1)\, |$$

So, in order to minimize the increase in network utilization caused by the possible re-emission of packet, a message will be only forwarded if the number of newly reached nodes is greater than a certain threshold, which is calculated based on the number of nodes in the neighbourhood (local density) of the recipient node d_2. It is denoted as Threshold ($|N(d)|$), and the decision taken by d_2 to re-emit the packet forwarded from d_1 is defined by the boolean function:

$$B(d_2, d_1) = \begin{cases} \boldsymbol{true}, if\ h(d_2, d_1) \geq threshold\ (|N(d_2)|) \\ \boldsymbol{false}, otherwise \end{cases}$$

If this threshold is exceeded, the recipient node d_2 becomes a transmitter in turn. This message is then effectively sent when a random delay (denoted by RAD) expires. The threshold function, which allows the protocol to facilitate the message broadcasting/re-broadcasting depends on the size of the neighbourhood 'n', given by:

$$Threshold\ (n) = \begin{cases} \mathbf{1}\ , & if\ n \leq SafeDensity \\ \boldsymbol{MinGain} * \boldsymbol{n}, & Otherwise \end{cases}$$

Where SafeDensity is the maximum safe density below which, DFCN will always rebroadcast, and MinGain is a parameter of DFCN used to compute the minimum threshold to forward the message (e.g. it is the ratio between the number of neighbours which have not yet received the message and the total number of neighbouring nodes).

Each time a new node **'n'** discovers a new neighbour, the value of **RAD** for all messages is set to 0. This means that the message is set for immediate emission. If **N(d)** is greater than a given threshold, known as **ProD**, this behaviour will be disabled, and therefore, no events or actions will be taken when a new node is discovered.

1.3 Thesis Objective and Motivation

The work proposed in this thesis tackles a specific type of MANET, known as Metropolitan Mobile Ad-Hoc Networks, which is characterized by a disparate density that is continuously changing, whereas highly dense areas can swing from being active to inactive over short periods of time. Because creating a real testbed for this type of network is very costly and challenging, and might also lack the reproducibility factor, it was decided that the best approach to handle it is by means of a simulation framework. The Madhoc simulator has been selected to achieve this [14].

An evolutionary algorithm-based technique that combines nested GA with fuzzy-based fitness is proposed and implemented. The technique integrates the Madhoc simulator in its core and considers DFCN optimization over multiple real-life mobility scenarios.

Another reason for choosing to work with AdHoc network is the prevalence of Internet-of-Things (IoT) and 5G technology, which uses super high frequencies (3 -30 GHz) and extremely high frequencies (30-300GHz). This technology requires a direct line-of-sight in order to deliver the promised the promised Gbps speeds wirelessly. In order to do so, it is expected that multiple micro/small cells will need to be deployed and be arranged in such a way that a direct line-of-sight is maintained.

These cells will eventually act as a adhoc networks and will require optimization. The DFCN protocol is selected because of its highly-scalable nature, where only five parameters require calibration according to different environmental surroundings (e.g. node density, node speed, distance, etc.). Therefore, the cells can be reprogrammed with the newly calculated parameters whenever the surrounding change, without the need to physically modify them.

1.4 Problem Statement

In order to optimize the DFCN protocol, multiple decision parameters need to be considered. These parameters dictate how DFCN operates and they characterize the search space. Since the optimization heavily relies on each specific scenario, an individual optimization trend is expected for each scenario.

The reachability time t_r is the output benchmark that is used to measure the optimization result. It is the amount of time required for the network to reach a certain number of pre-defined nodes. The goal of this research is to optimize the DFCN parameters to decrease the reachability time of the nodes inside the MANET. The problem is formulated as follows:

$\boldsymbol{m}$: $\boldsymbol{instance\ of\ Madhoc\ simulator}$, $\boldsymbol{t_r}$: $\boldsymbol{reachability\ time}$.

$$\boldsymbol{t_r = m(LowerRAD, UpperRAD, ProD, MinGain, SafeDensity)} \qquad \textbf{(1)}$$

$$\boldsymbol{f(LowerRAD, UpperRAD, ProD, MinGain, SafeDensity) = min\,(t_r)}$$

The function f corresponds to the proposed system where the target is to minimize the reachability time t_r for each instance of the simulator *m.* Table I-1 below shows the DFCN parameters along with their respective threshold and domain values.

As already stated, this will be done on three different mobility model scenarios, namely Highway, Mall and Human mobility. The description for these scenarios is shown next.

Table I-1. DFCN PARAMETERS DESCRIPTION

Parameter Name	**Domain**	**Description**	**Unit**	**Threshold Value**
LowerRAD	Real ($\mathbb{R}$)	Minimum time required to rebroadcast.	Second	[0, UpperRAD]
UpperRAD	Real ($\mathbb{R}$)	Maximum time required to rebroadcast	Second	[LowerRAD, 10]
ProD	Integer ($\mathbb{Z}$)	Maximum Density for which it is still required to use proactive behaviour (reacting to new neighbours)	Device	[0, 100]
MinGain	Real ($\mathbb{R}$)	Minimum gain for rebroadcasting.	-	[0, 1]
SafeDensity	Integer ($\mathbb{Z}$)	Maximum density, below which the protocol will always broadcast.	Device	[0, 100]

1.4.1 Highway Scenario

The main feature of the highway mobility model is that the nodes move at significantly higher speeds compared to the other mobility models and the nodes are lower in numbers.

The spot density is also set to one spot per square kilometre, which is very sparse, and the number of spots per simulation area is limited to three. In this scenario, most of the generated traffic comes from nodes moving in opposite directions to simulate cars moving on different and opposing lanes of a highway. Table I-2 below shows the properties of this scenario.

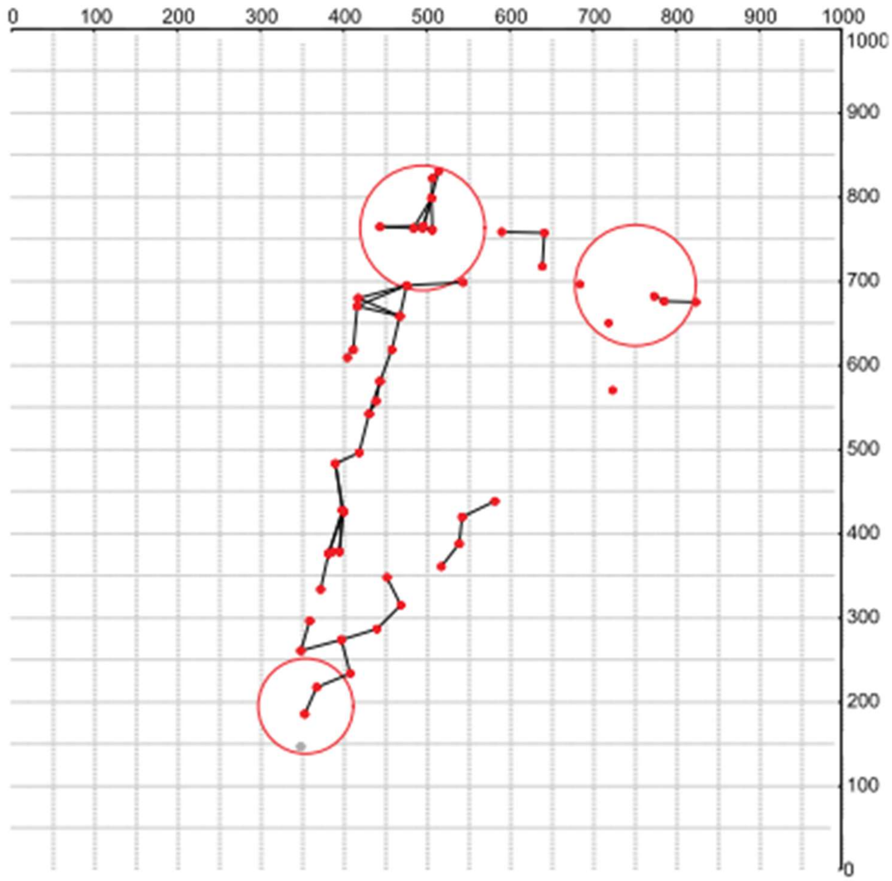

Figure I-2. Highway Scenario Illustration

1.4.2 Mall Scenario

The mall mobility scenario is composed of separate regions connected by relatively narrow areas. It represents a group of shops interconnected using corridors. In this scenario, the surface area is smaller than the highway one and the velocity is much slower.

Also, the nodes move randomly for most of the time with no clear targets, representing humans wandering around and shopping in arbitrary shops. Table I-3 illustrates the different parameters for this scenario.

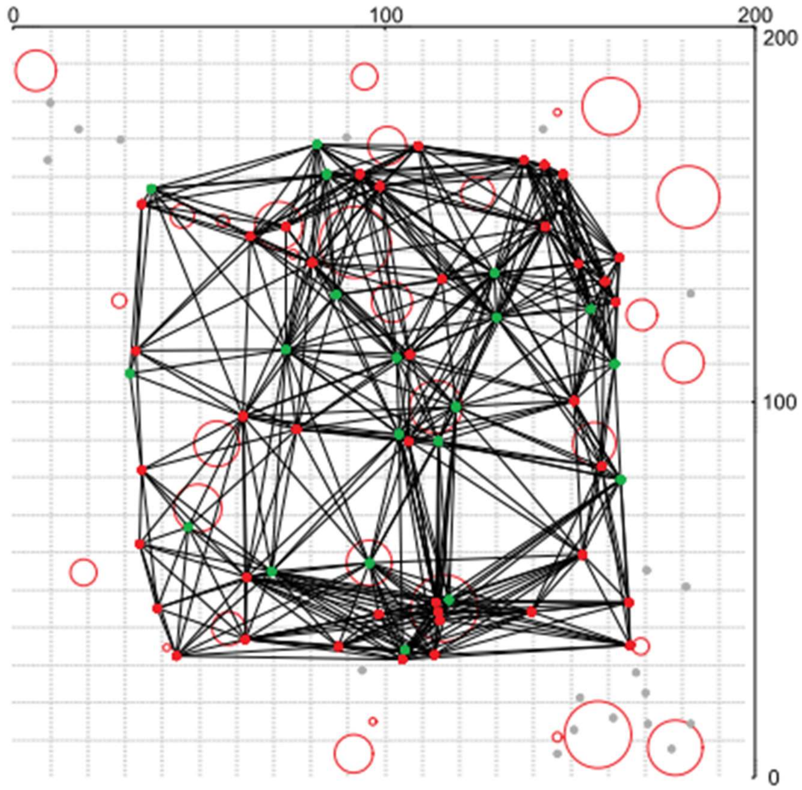

Figure I-3. Mall Scenario Illustration

1.4.3 Human Mobility Scenario

This scenario is more distinctive than the mall one and is considered one of the most daunting models. In this context, the focus is on the human mobility scheme, where the movements are not random, but instead, there is a list of target destinations that each node mostly moves towards. These targets can be far away, as well as a few meters around. Also, the targets can dynamically change with time depending on human behaviour. For instance, a waiter in a restaurant can be regularly moving back and forth between the kitchen and customers' tables.

The human mobility scheme is defined as a round simulation area, where fixed places that act as target spots are scattered and where the distance between two places cannot be less than 10 meters. Table I-4 shows the parameters for the human mobility model.

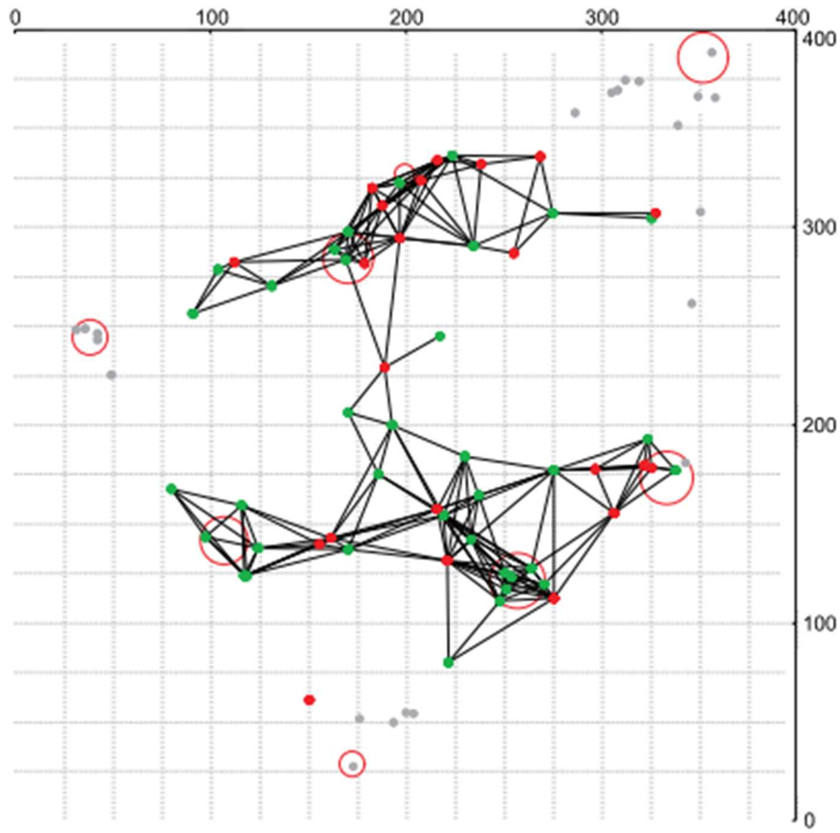

Figure I-4. Human Mobility Scenario Illustration

Table I-2. HIGHWAY MOBILITY SCENARIO PARAMETERS

Parameter	Value	Units
Surface Area	1 x 1	km^2
Nodes Density	80	nodes / km^2
Velocity	[20 40]	$m.s^{-1}$

Table I-3. MALL MOBILITY SCENARIO PARAMETERS

Parameter	Value	Units
Surface Area	0.3 x 0.3	km^2
Nodes Density	6,500	nodes / km^2
Velocity	[0.3 1]	$m.s^{-1}$

Table I-4. HUMAN MOBILITY SCENARIO PARAMETERS

Parameter	Value	Units
Surface Area	0.05 x 0.05	km^2
Nodes Density	80,500	nodes / km^2
Velocity	[0.3 1.5]	$m.s^{-1}$

1.4 Thesis Outline

The rest of this thesis is organized as follows. Chapter II shows an analogy of the genetic algorithm, what is a fuzzy logic system and describes its architecture. Chapter III introduces the Madhoc simulator and gives an insight about its features and the different modes of operations.

In Chapter IV, a review of the related work concerning the optimization of broadcasting techniques in MANETs is presented. Chapter V demonstrates the algorithms and techniques used to solve the problem. Chapter VI shows the obtained results and discusses them. Finally, Chapter VII concludes this work and proposes the potential future work.

Chapter (II)

GENETIC ALGORITHM and FUZZY LOGIC

CHAPTER II
GENETIC ALGORITHM and FUZZY LOGIC

2.1 Genetic Algorithm

Genetic Algorithm is considered as one of the most renowned metaheuristic techniques inspired by natural selection and natural genetics and it is used to reach approximate or true solutions for various optimization problems [15]. GA relies on techniques inspired by evolutionary biology (e.g. inheritance, selection, crossover, and mutation to solve the problem the same way the nature does with the living organisms.

The basic idea of GA is to randomly create an initial population consisting of individual potential solutions to a given problem known as chromosomes, and then evolve this population for a given number of iterations, known as GA generations. For each generation, each chromosome is evaluated using a predetermined fitness function that is selected based on the application.

To reach the next generation, an offspring individual is formed by merging two chromosomes from the current generation. This is done by using a crossover operator which is applied on two chromosomes or by altering a chromosome using what is known as a mutation operator [16] [17].

A new generation is then constructed by means of a selection operation, according to the fitness values of the chromosomes, where the fittest chromosomes have higher probabilities of getting selected [18]. After evolving several generations, the algorithm is expected to reach a better solution, which represents the optimum or sub optimal solution to the problem. This is known as the convergence of the GA [19].

The following are some benefits the GA has got over traditional numerical optimization techniques:

- Does not require derivative information [20].
- Can optimize with both discrete and continuous parameters [21].
- Can operate on big data scale and handle a large number of parameters.

- Can be run on parallel computers [22].
- Can be used to provide a set of solutions to a given problem.

2.1.1 Theory of Genetic Algorithm

GA has been developed by John Holland in 1975 [23]. Back then, it has been widely used in Data Mining applications like data clustering and data classification, etc [24], to provide a randomized and global search methodology to find the optimum solution of a given problem.

GA begins by generating the first initial population, randomly, followed by the evaluation of the fitness function for each individual chromosome, in such a way that the fitter individuals will have a greater chance of survival and pass the information they possess to the newer generation.

The crossover operator allows each potential solution to exchange information in such a way that is similar to that used by natural organisms to create a new biological offspring. Each chromosome might as well be subjected to a mutation operation, in which a random change with a predetermined percentage takes place in the individual chromosome. After the aforementioned operations are applied to the population, a new one emerges and the generations keep evolving until the convergence criteria have been met or a predetermined number of generations have finished. Most of the times, GA algorithms are studied on different data with various sizes to find.

2.1.2 The analogy between nature and genetic algorithm

GAs derives its behaviour from the concept of the natural biological evolution. Table II-1 highlights the analogy between GA and its biological counterpart.

Table II-1. COMPARISON BETWEEN GA AND BIOLOGICAL COUNTERPART

Genetic Algorithm (GA)	Biological Counterpart
Optimization problem	Nature
Feasible solution	Individual living in the nature
Fitness function	Level of adaptation to its surrounding environment
Stochastic operators	Reproduction and mutation in nature evolutionary process
Progressively applying a set of stochastic operators on a set of potential solutions	Evolution of the organisms to suit their environment

2.1.3 Applications of genetic algorithms

As already mentioned, GAs have been used in many applications including data mining, clustering, classification and scheduling and time optimization, in addition to optimizing many real-world problems such as:

- **Financing and Stock**

GA can be used to take investment decisions and to predict future performance in traded stocks and stock markets in general [25].

- **Identifying criminal suspects**

A lot of facial recognition programs use GA to evolve pictures of faces based on the databases of hundreds of individuals features. This methodology can help identify faces, even when the person ages up [26].

It can also be used to do blind prediction where the GA-based programs show randomly generated face images to different witnesses, who pick up the ones that most resembles the suspect, and then the selected faces are mutated and combined using GA-related stochastic operations to create new combinations of facial features, and the process is sustained until an accurate picture of the suspect face is obtained [27].

- **Route Planning**

Shipping and Freight companies can rely on GA to determine the best route to their destination [28].

- **Medical**

GA can help in the developments of new treatments by optimizing drug formulas and improve diagnostics [29].

2.1.4 Canonical genetic algorithm

This section will explain the different operations carried by the GA in brief as follows:

Representation

An initial population is first generated in a random fashion. It represents a random set of chromosomes (population) that corresponds to a possible solution in the form of its genetic structure. Each one of these chromosomes consists of a group of numerical values (int, float, etc.) known as the genes. For some applications, the initial population can be generated according to a certain criterion, instead of being totally random, in such a way that will enhance the final solution of the GA.

Evaluation

Fitness values are calculated for each chromosome based on how likely it might solve the problem and contribute to the desired solution. The fitness function is used to evaluate the fitness of each chromosome. Therefore, the fitness function is designed with special care as it decides which individuals can reproduce and create the next generation of the population.

Selection

The selection operation of the chromosomes that decides whether they will reproduce or live to the next generation mainly depends on the designed fitness function. According to Darwin's' theory, the best individuals are the ones that should survive and be used to create new offspring. This means that chromosomes possessing higher fitness values should be assigned a higher probability of reproducing offspring. After evaluating the fitness of all individuals, the selection operation chooses the fittest individuals for reproduction or recombination.

Selection that is based on the individuals' fitness is usually based on roulette wheel which is one of the techniques used for selecting the best solution for crossover. The roulette wheel selection resembles a roulette wheel in a casino where a certain part of the wheel is assigned to each of the possible outcomes.

Crossover

Crossover operation in a biological system consists of a candidate of solutions combined together to produce an offspring in each iteration which is known as a generation.

The offspring that will survive (yield a higher fitness value) will be considered as one the fittest and its offspring will become a candidate solution in the next generation. Crossover works by combining one or more pairs of chromosomes randomly (based on their fitness) as parents and swapping their segments of genes according to the designed crossover operator. This will yield the offspring for the generations to come. In some cases, parent chromosomes can pass on their bad genes to their offspring, but this can be overcome by employing a powerful selection algorithm. As already highlighted, there are different crossover methods such as single point crossover, two-point crossover and uniform crossover, etc.

Single point crossover

In a single point crossover, a single crossover point is selected, and a binary string from the start of one chromosome to the crossover point is copied from one parent, and the rest is copied from the other parent.

Two-point crossover

In Two-Point crossover method, two different crossover points are randomly selected, and a binary string from the start of the first chromosome to the first crossover point is copied from one parent, the region from the first to the second crossover point is then copied from the second parent to the first region in the first parent.

Uniform crossover

There is no clear separator in this type of crossover. Bits are randomly copied from the first parent or from the second one.

Mutation

The mutation operation is used to modify a chromosome for the next generation. This operation assures the diversity of the population from one generation to another. It does so by randomly mutating one or more genes inside the chromosome. This prevents the GA from stagnating in the earlier phases of evolution.

Therefore, the main aim of mutation in GAs is to introduce diversity and to avoid local minima. The GA performance depends mainly on the crossover and mutation operations. These operations are designed according to the problem or the required application.

Elitism

Sometimes, during the converging process of the GA, the fittest chromosome might get lost and not get transferred to the newer generation. Therefore, the fittest chromosome can be directly transferred to the newer generation to ensure a continuous or a sustained fitness of the population, in such a way that the fitness value for the newer generation will always be either the same or better than the previous one.

As a result of this, elitism can rapidly increase the performance of the GA as it prevents the loss of the already calculated fittest individual. It is worth noting that elitism does not interfere with the normal stochastic operations of the GA. It is done after the whole calculations of the current generation have already finished.

An illustration for the GA is graphically illustrated in the flow chart shown in Fig II-1.

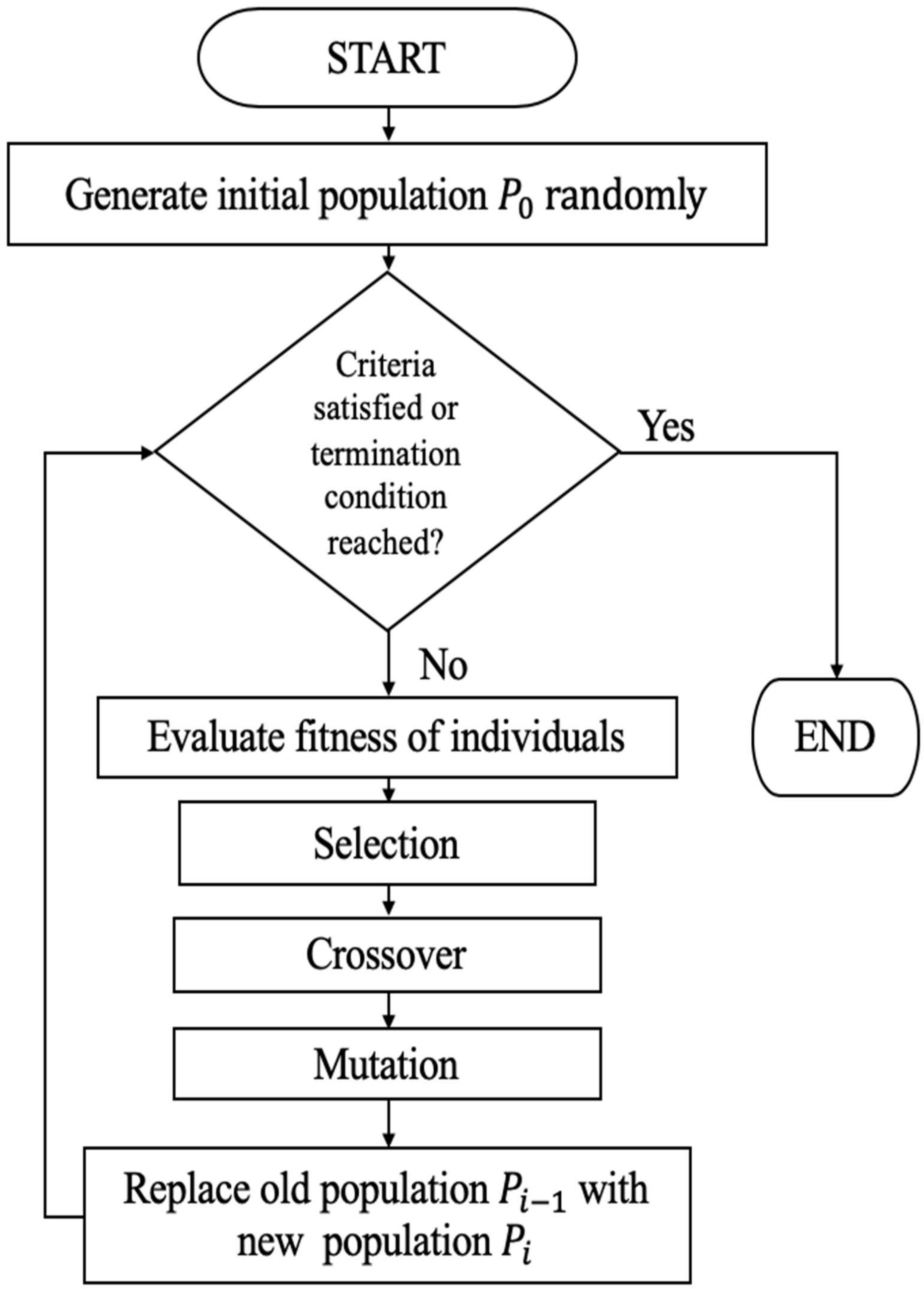

Figure II-1. Illustration of the GA

2.2 FUZZY LOGIC

2.2.1 Introduction to Fuzzy Logic

Fuzzy logic is considered as a type of multi-valued logic system, in which variable reality values can be any real number between zero and one [30]. It is used to address the definition of partial reality, where the value of truth can vary between absolute true and total false. In binary conditional systems, such as the Boolean system, the value of truth for the variables can only either be a zero or a one.

Fuzzy logic is based on the fact that sometimes people make decisions based on non-accurate and non-numerical data. Fuzzy models map ambiguity and imprecise knowledge by means of computational intelligence [31].

The first major part of a fuzzy system is the fuzzification process, which transforms numerical inputs into what is known as fuzzy membership functions, which are then used achieve a fuzzy output membership function and a crisp output value that can then be used as a final decision for the system and for control purposes.

An illustration for the Fuzzy Logic system is shown in Fig. II-1 below [32].

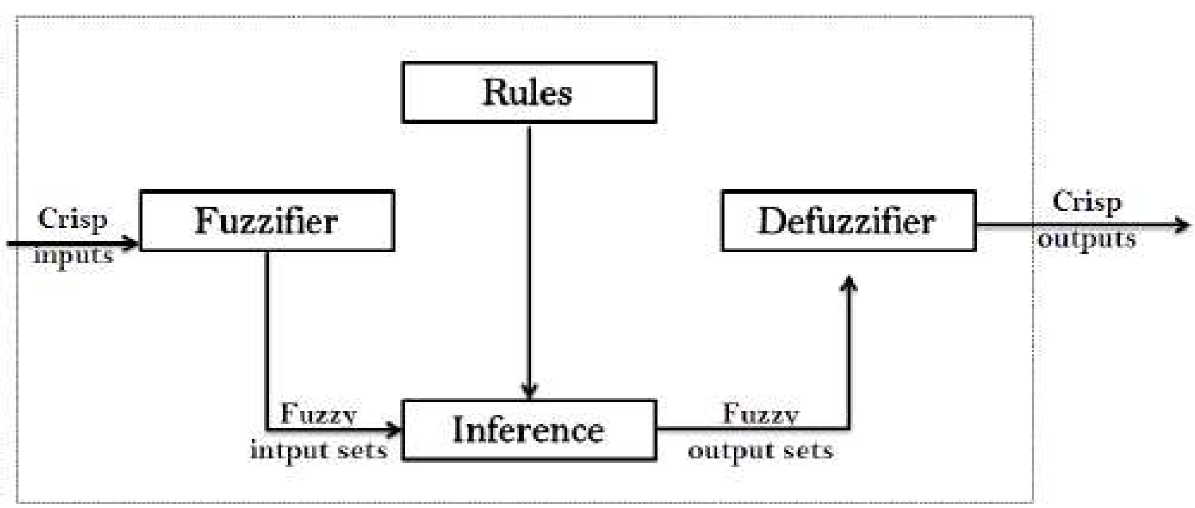

Figure II-2. Fuzzy Logic Illustration

2.2.2 Fuzzy Logic System Architecture

Fuzzy logic systems are divided into four main parts:

- **Fuzzification Module**: It transforms the system inputs, which are crisp numbers, into fuzzy sets. It splits the input values into different regions of various strength (LOW, MED, HIGH, etc…). These regions can vary in numbers, according to the application and the fuzzification model.
- **Knowledge Base**: Fuzzy logic systems are regarded to as AI-based system; therefore, they possess a knowledge base where all of the agent's reference information and data are stored. The rules that govern the Fuzzy Logic system contained in its knowledge base. These rules take the form of an if-else ladder.
- **Inference Engine**: The Inference engine is the Fuzzy Logic System's core. It is analogous to the computer Central Processing Unit (CPU). It is the module where all the information processing takes place within. The main task of the inference system is to derive an acceptable result by interpreting all the data it receives from the unit of fuzzification. This is accomplished with the help of the knowledge base rule. Finally, the output conclusions will be sent to the defuzzification module to get the final meaningful application-dependent output.
- **Defuzzification Module:** This module collects the data received from the Inference Engine and transforms this data into a user-accepted application-dependent form. In other word, it is the process that maps a fuzzy set to a crisp set. A lot of techniques can be used to achieve this. The centre of gravity method is a very common one; in which, the results of the rules are first added together according to certain criteria. The widely used fuzzy set membership function has the shape of a triangle. If this triangle was cut with a straight horizontal line somewhere between its upper and lower part, and its top portion was removed, the remaining part will form a trapezoid. All the trapezoids resulting from different function are then superposed together, forming a single shape

and a fuzzy centroid is formed, where its centre corresponds to the defuzzified value. Fig. III-2 below shows a demonstration of this method.

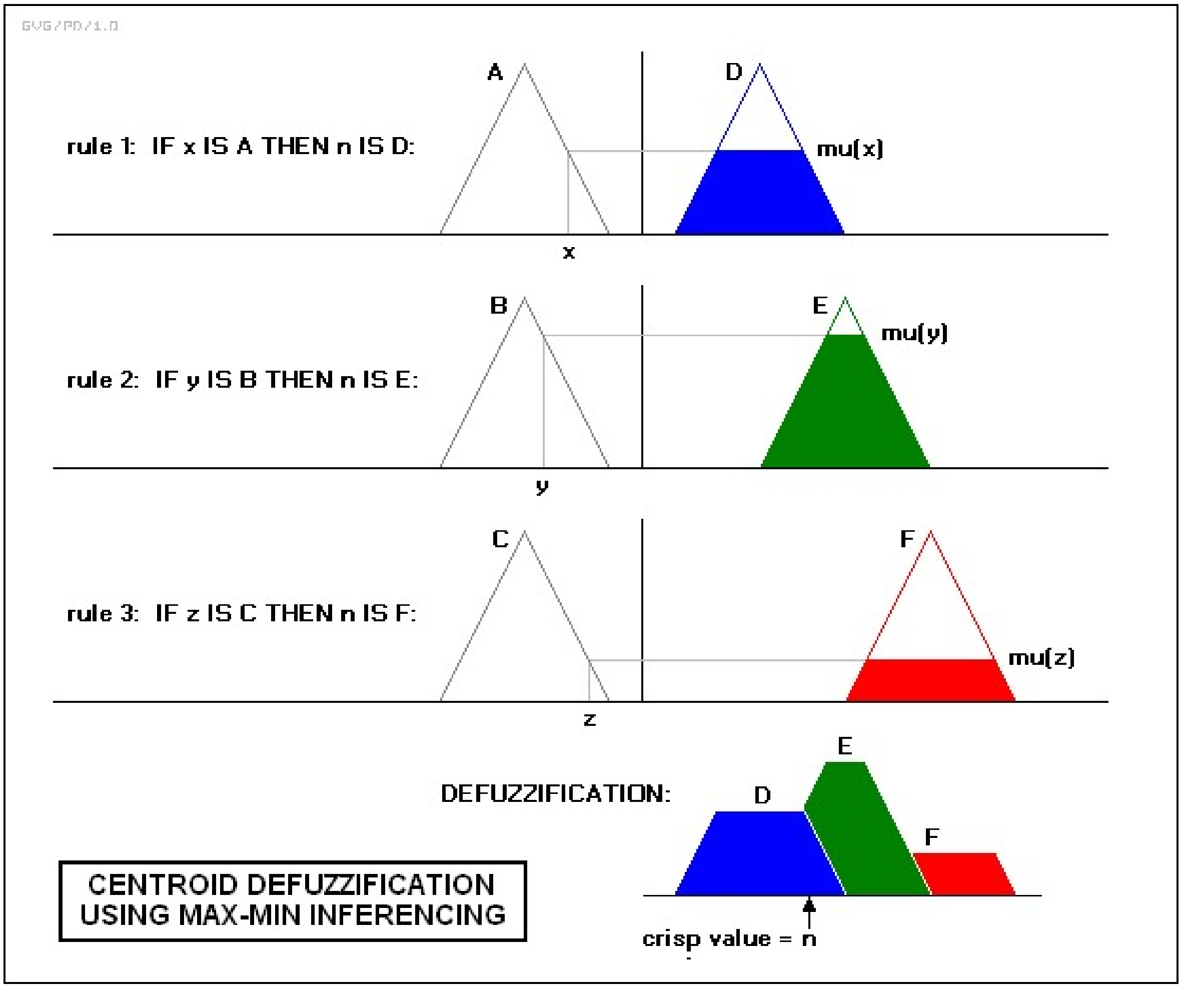

Figure II-3. Example of a Defuzzification module

Chapter (III)

MADHOC SIMULATOR

CHAPTER III
MadHoc Simulator

Madhoc is a metropolitan MANET simulator completely written in Java and available to use publicly [33] on the author’s website [34]. The simulator provides the ability to simulate MANET using different parameters and real-life constraints such as working area size, mobility speed, wall thickness, etc. It also supports many different wireless technologies (e.g. Wi-Fi, Bluetooth, GSM, etc.). Most importantly, it implements the full DFCN broadcasting protocol with all the required decision parameters to optimize it. Madhoc can be executed as a standalone application, as shown in Fig. III-1, or as an Application Programming Interface (API).

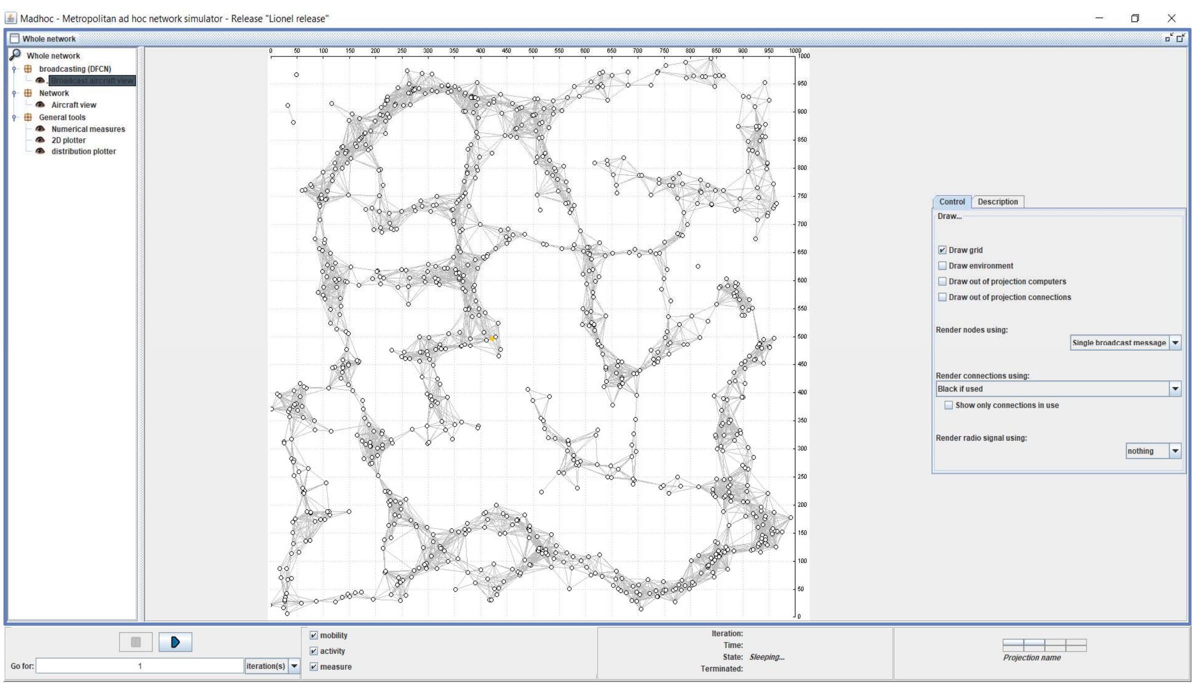

Figure III-1. Madhoc Standalone Application llustration

To be able to collect the required statistics and results, a Madhoc monitor class is used. A monitor is not a part of the physical network and does not have an instance in real networks and is regarded to as an abstraction entity that only exists at simulation level. It mainly aims at maintaining a global perspective on all nodes and for carrying out the required operations such as node deployment and initialization. It mainly serves as an observer of the Ad-hoc decentralized process.

Fig. III-2 shows an illustration for such Monitor. Another major attribute of the Madhoc simulator is that it does not use an event-driven simulation architecture, but instead, the simulator's kernel iterates upon a discrete time domain, where the distance between two intervals is known as *the resolution.*

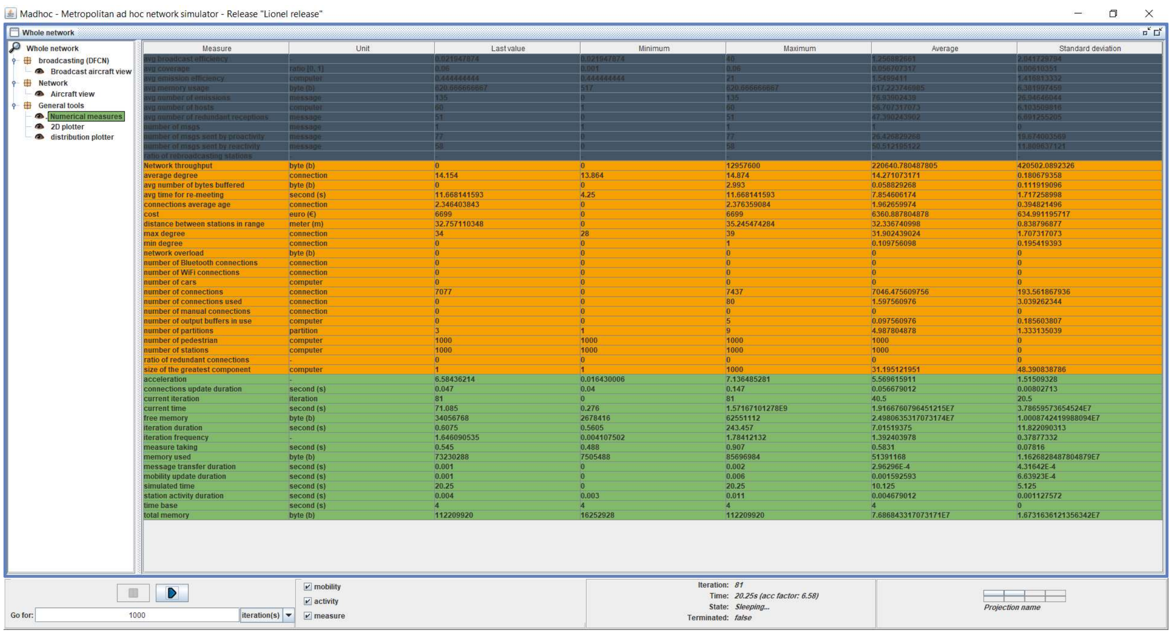

Figure III-2. Madhoc Numerical Measures

This parameter is defined by the user and should be fixed throughout all the related applications to guarantee comparable and consistent results. The higher this value is, the less accurate the simulation will become. This value should be carefully used according to the required application. In the case of DFCN, this value must be at least twice lower than the maximum RAD, otherwise the benefits of using RAD will be completely lost.

Another important factor to consider while choosing the resolution is the mobility scheme of the nodes, the resolution must be small enough to make sure that the nodes move in reasonable steps, otherwise, some connections that could have taken place in real life would not be simulated.

Chapter (IV)

RELATED WORK

CHAPTER V
Related Work

In the literature, most research has been dedicated to solving the broadcasting issues by using a multitude of different methods. Evolutionary multi-objective approaches have been proven to be effective in solving broadcasting problems [35], however, they suffer from time and performance issues [36]. Other methods focus on combinatory numerical models but most of them fail to adequately reduce the routing overhead with highly scalable networks, which is a main feature of MANET.

Those who focused on the DFCN protocol did not formulate a trending mobility model for optimizing the decision parameters. Some of the researchers directly focused on detecting the selfish nodes in the network and avoiding them to increase the efficiency of the broadcasting protocols, the most notable work in this regard is by S. Subramaniyan et al. [37], where a Record-and-Trust-Based Detection (RTBD) technique was simulated that can efficiently detect selfish nodes in MANET.

The main focus of this work was to accelerate the detection of misbehaving selfish nodes. The suggested technique consists of a packet dropping detection mechanism and a selfish node reduction mechanism. The selfish node generates a trust report with each neighbour, which in turn, reports its previous communication reports to the neighbouring node. Based on that info, the neighbouring node can detect whether the selfish node has dropped packets or not. Then, the neighbouring node gathers the trust report to detect misreporting and finds out which node has dropped any packets. Also, a selfish node may report a false record to hide the packet drops from being detected.

The proposed method managed to diminish the overhead, latency and overhead ratio which improved the broadcasting performance of the MANET. However, the authors did not demonstrate how the acquired security could be transferred to the neighbouring nodes in the network so that they could avoid being compromised by the selfish nodes detected by RTBD, meaning that the technique is not scalable on larger networks and the performance will be degraded. Another key focus in the literature is intelligent rebroadcasting techniques that reduce the overhead by estimating the usefulness of rebroadcasts and the probability of causing a collision.

S. S. Basurra et al. [38] discussed a Zone based Routing with Parallel Collision Guided Broadcasting Protocol (ZCG) to reduce redundant broadcasting and to accelerate the path discovery process. The authors compared ZCG with two other techniques, Dynamic Source Routing (DSR) and Ad-hoc On Demand Distance Vector Routing (AODV). It was concluded that ZCG can speed up the routing process in MANET due to its on-demand parallel collision guided broadcasting. ZCG uses a one hop clustering algorithm that divides the network into different group or zones led by reliable leader nodes that are mostly static and have plentiful power resources.

The ZCG protocol depends on the decomposition of the network into contiguous zones, with one node being selected from a group of nodes to be considered the zone leader, known as ZL, which is selected based on a predetermine fitness criteria, such as high battery power and/or zero/low mobility. Eventually, the ZL(s) establish connectivity among themselves, either directly or via reliable intermediate nodes that are situated in the overlap of two or more zone coverage areas.

However, the proposed method lacked distribution fairness among the nodes and did not protect zone members from selfish behaviour attributed to the Zone Leader. Another interesting finding in the literature is the clustering of MANETs as a mean to reduce the complexity of the routing table.

M. Ahmad et al. [39] provided a comprehensive survey about the different clustering algorithms that address this issue. It concluded that the effectiveness of the clustering algorithms depends on a set of specific parameters, which are the nodes remaining power, the relative mobility, the overhead data, the trust value, and the node reputation.

Raziel Carvajal-Gomez et al. [40] proposed and designed an emergent overlay technique for efficient and reliable broadcast in heterogeneous MANETs. The proposed technique allows the devices in the network to automatically switch from a controlled flooding broadcasting scheme, to the use of an overlay. The authors tested the proposed technique on 600 mobile nodes using a full-stack simulation in OMNeT++ over an area of 90*45m.

The achieved results have shown that the adaption of emergent overlays reduced the overall energy consumption and have improved the total coverage compared to other protocols. It has also decreased the collision rate significantly. The proposed system tackled the poor performance of the controlled flooding broadcasting, where the nodes density is very large, and a broadcasting storm is likely to arise. In the proposed work, the nodes in very dense areas autonomously decide to switch from controlled flooding to the use of an overlay. The authors also presented a mechanism for the autonomous adaption of overlay where the nodes collect observables about their surrounding and the environment and then an adaption policy triggers the creation of an overlay based on predefined thresholds over the collected observables.

The suggested protocol was compared with two other adaptive protocols (ACF [41] and S-H Flooding [42]) reached a reach-value of 99.99% outside points of interest (POI) regions and 100% reach-value in POI regions, compared to values of 100%, 98.9% and 90.99%, 96.34% for the other algorithms respectively. The most challenging aspect about the proposed system was the estimating of the threshold values that need to be applied over the collected observables, therefore an automated technique that can calculate and estimate these thresholds would significantly add to the proposed work.

Chapter (V)

PROPOSED SYSTEM

CHAPTER V
Proposed System

The proposed technique consists of a nested GA with fuzzy-based fitness. The aim is to optimize the DFCN decision parameters according to the reachability time and to find certain trends for each of the different scenarios.

The benchmark used is the reachability time for 10% of the nodes, which is the time required so that 10% of the nodes in the network successfully deliver their messages. The outer GA contains the DFCN parameters and the to-be-calculated output from the simulator. The inner GA evolves a set of rules for the fuzzy system, where each chromosome represents a complete fuzzy set and the inference output represents the inner fitness.

The final inner fitness value that is calculated after the convergence has completed sets the fitness value of the outer GA. The proposed system is developed using C# language on Microsoft Visual Studio 2017 under 64-bit Windows 10 with 8GB of RAM and an Intel Core i5-6500 CPU. Because the proposed system is built using C# and the Madhoc simulator operates fully in Java, a mechanism that interfaces them was required. To be able to accomplish this, each time the simulator is required to calculate the reachability time, it is executed by the developed application as a command line program running inside a virtual sandbox process, where all the standard inputs and outputs are redirected to the application.

Fig. V-1 shows a pseudo code for the system. Fig. V-2 shows the class diagram of the designed system. The class Diagram shows the relation and the dependencies between the different classes inside the code. and Fig. V-3 shows an illustration for the system. Figure V-4 shows an example for running the system on one outer chromosome.

```
FUNCTION RunInnerGA(innerGenerationsCount, oChromosome)
    i ← 0;
    P_i ← InitializeInnerPopulation(innerPopulationSize, keyMin, keyMax );
    WHILE ( i < innerGenerationsCount -1 )
        FL ← BuildFuzzySystem_VariableSets( oChromosome );
        P_i+1 ← NULL;
        j ← 0;
        WHILE ( j < innerPopulationSize)
          FL ← InitializeFuzzySystemLinguistics( P_i[ j ] );
          Fitness( P_i[ j ] ) ← FuzzySystem_InferenceResult( );
          j ← j + 1;
        END WHILE
        j ← 0;
        WHILE ( j < innerPopulationSize / 2 - 1)
            parents = RouletteSelect( P_i );
            offspring[0,1] = Crossover(parents, innerGACrossoverProbability);
            P_i+1 ← P_i+1 + offspring[0,1 ];
        END WHILE
        j ← 0;
        WHILE ( j < innerPopulationSize)
            P_i[ j ] = Mutate( P_i[j ], innerGAMutationProbability);
            j ← j + 1;
        END WHILE
        P_i+1 ← P_i+1 + GetFittest( P_i );
        i ← i + 1;
    END WHILE
    RETURN GetHighestFitnessValue( Pi );
END FUNCTION

FUNCTION RunOuterGA(outerGenerationsCount): MAIN
    i ← 0;
    P_i ← InitializePopulation(outerGenerationsCount, thresholds Values[ ]);
    WHILE ( i < outerGenerationsCount -1 )
        P_i+1 ← NULL;
        j ← 0;
        WHILE( j < outerPopulationSize)
            Output( P_i[ j ]) ← GetMadhocOutput( P_i[ j ]);
            Fitness( P_i[ j ] ) ← RunInnerGA ( P_i[ j ] );
            j ← j+1;
        END WHILE
        j ← 0;
        WHILE( j < OP_ outerPopulationSize  / 2 - 1)
            parents = RouletteSelect(P_i) ;
            offspring[0,1] = Crossover(parents, outerGACrossoverProbability);
            P_i+1 ← P_i+1 + offspring[0,1];
        END WHILE
        j ← 0;
        WHILE ( j < outerPopulationSize)
            P_i[ j ] = Mutate( P_i[ j ], outerGAMutationProbability);
            j ← j + 1;
        END WHILE
        P_i+1 ← P_i+1 + GetFittest( P_i );
        i ← i + 1;
        ExtractParametersAndOutput( P_i );
    END WHILE
END FUNCTION
```

Figure V-1. Pseudo-Code of the Proposed System

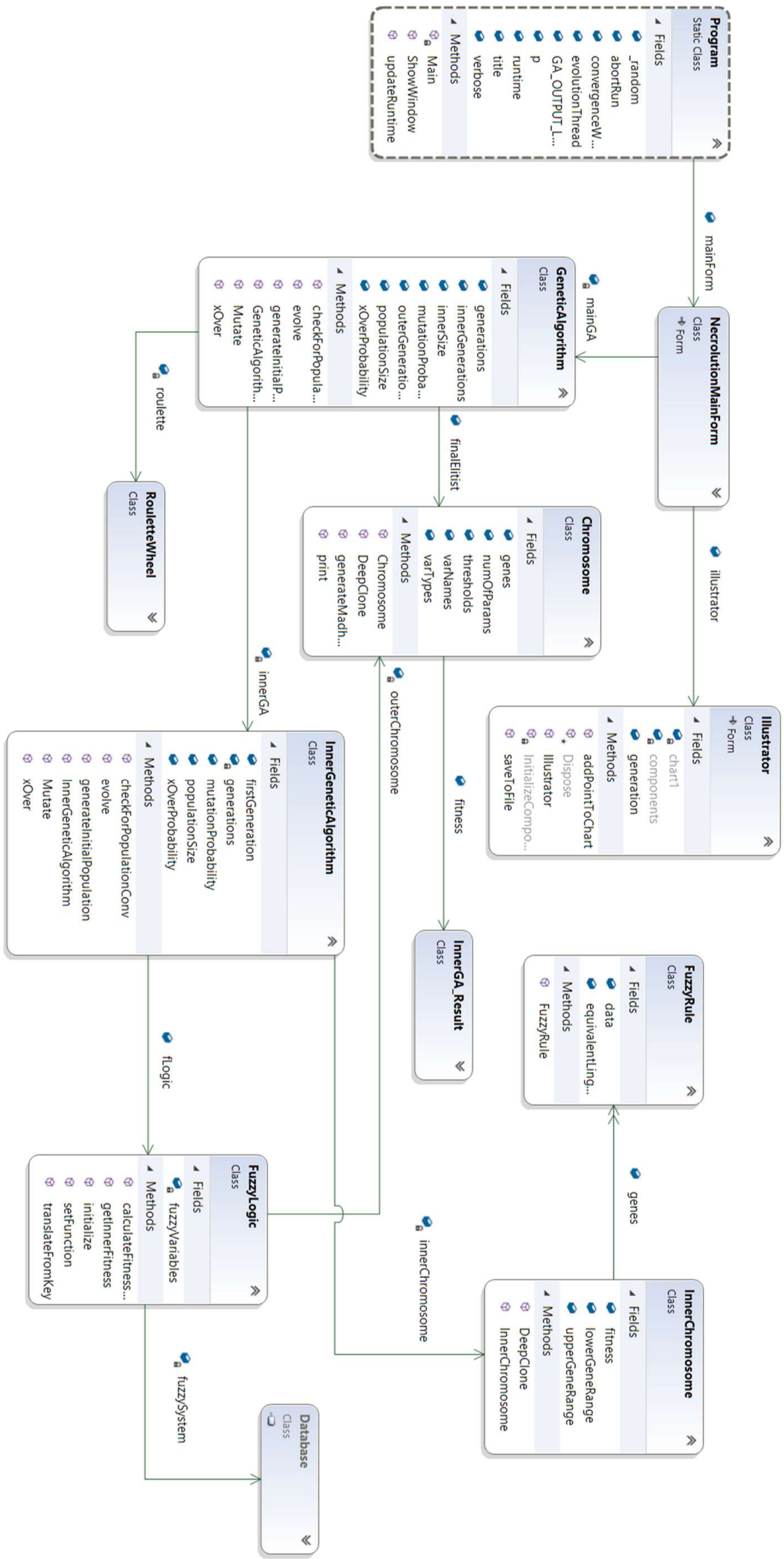

Figure V-2. Class Diagram for The Proposed System

The *RunOuterGA* function is the entry point of the program. The *InitializeInnerPopulation* function creates the initial population with random fuzzy logic keys that correspond to the linguistic strings. The *oChromosome* variable is the outer chromosome passed from the outer GA to the inner one, per generation. The *keyMin* and *keyMax* variables represent the range for the allowed number of keys per chromosome. At *line 5*, the fuzzy logic system is initialized and the fuzzy sets are created using the *oChromsome* genes, then at *line 9*, the linguistics are generated using the inner chromosome $P_i[j]$, and finally at *line 10*, the fitness is calculated by getting the inference result for the developed fuzzy logic system. The *ExtractParametersAndOutput* function is called per each outer GA generation to extract the current values of the decision parameters and the output from the fittest chromosome.

5.1 The Fuzzy Logic System

The fuzzy system is used to calculate the fitness for the inner GA. Each chromosome from the inner GA will act as complete fuzzy set. Each DFCN parameter will act as a linguistic variable with LOW, MED and HIGH as values.

All of the variables have a triangular membership function that is equally divided over the maximum threshold of the respective parameters it represents. The rules for the fuzzy set are generated and optimized using the inner GA, which will be highlighted later. In order to accomplish this, the inner chromosome is decoded from a numerical form to equivalent linguistic strings, according to Table V-1.

To get the output values, the inference system uses a centroid defuzzifier with an interval of 1000. The interval represents the number of segments that the linguistic universe will be split into to perform the numerical approximation of the area center.

Table V-1. NUMERICAL TO LINGUISTIC CONVERSION TABLE

Value	Equivalent Linguistic
1	LOW
2	MED
3	HIGH
-1	NOT LOW
-2	NOT MED
-3	NOT HIGH
0	NOT APPLICABLE

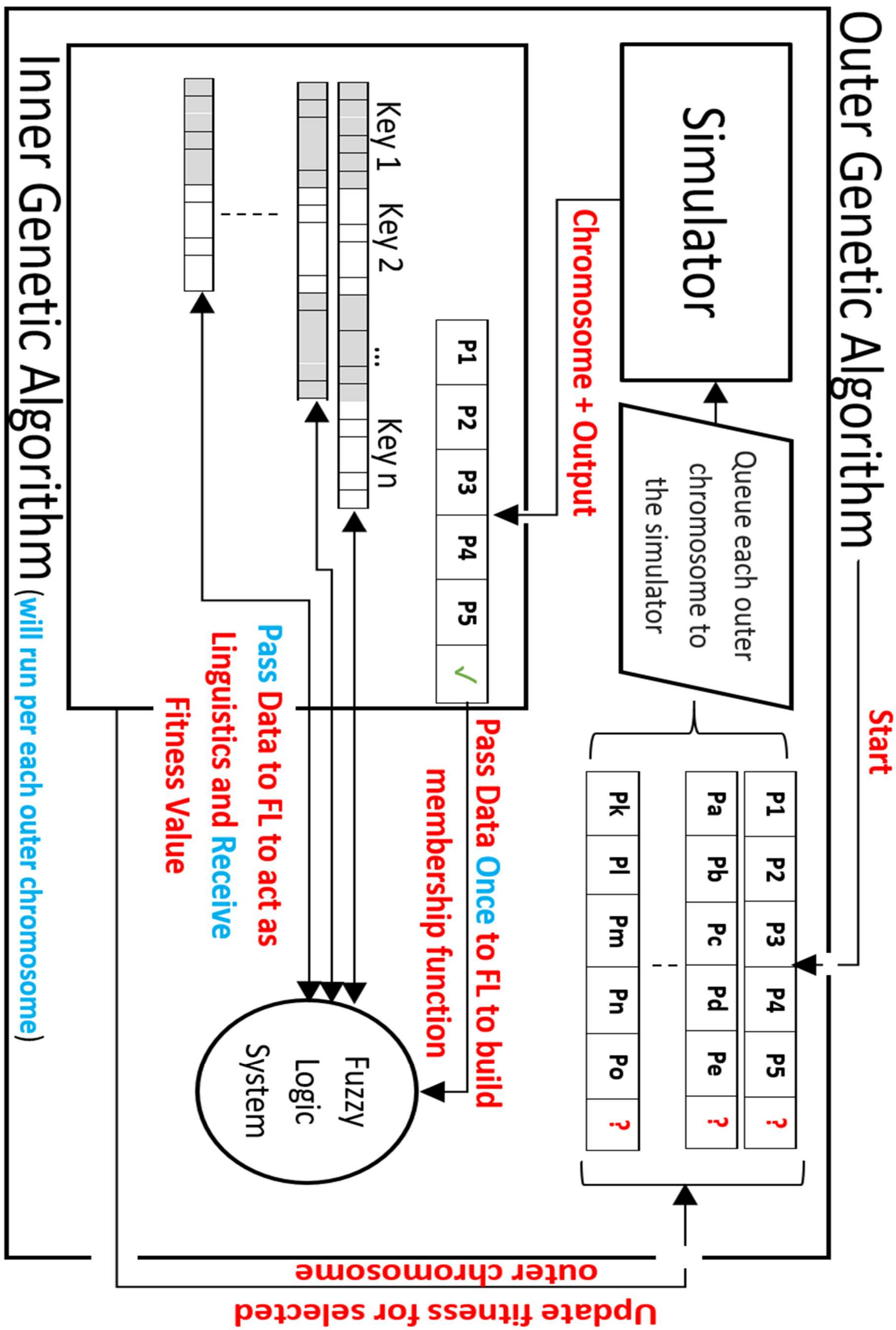

Figure V-3. Proposed System Illustration

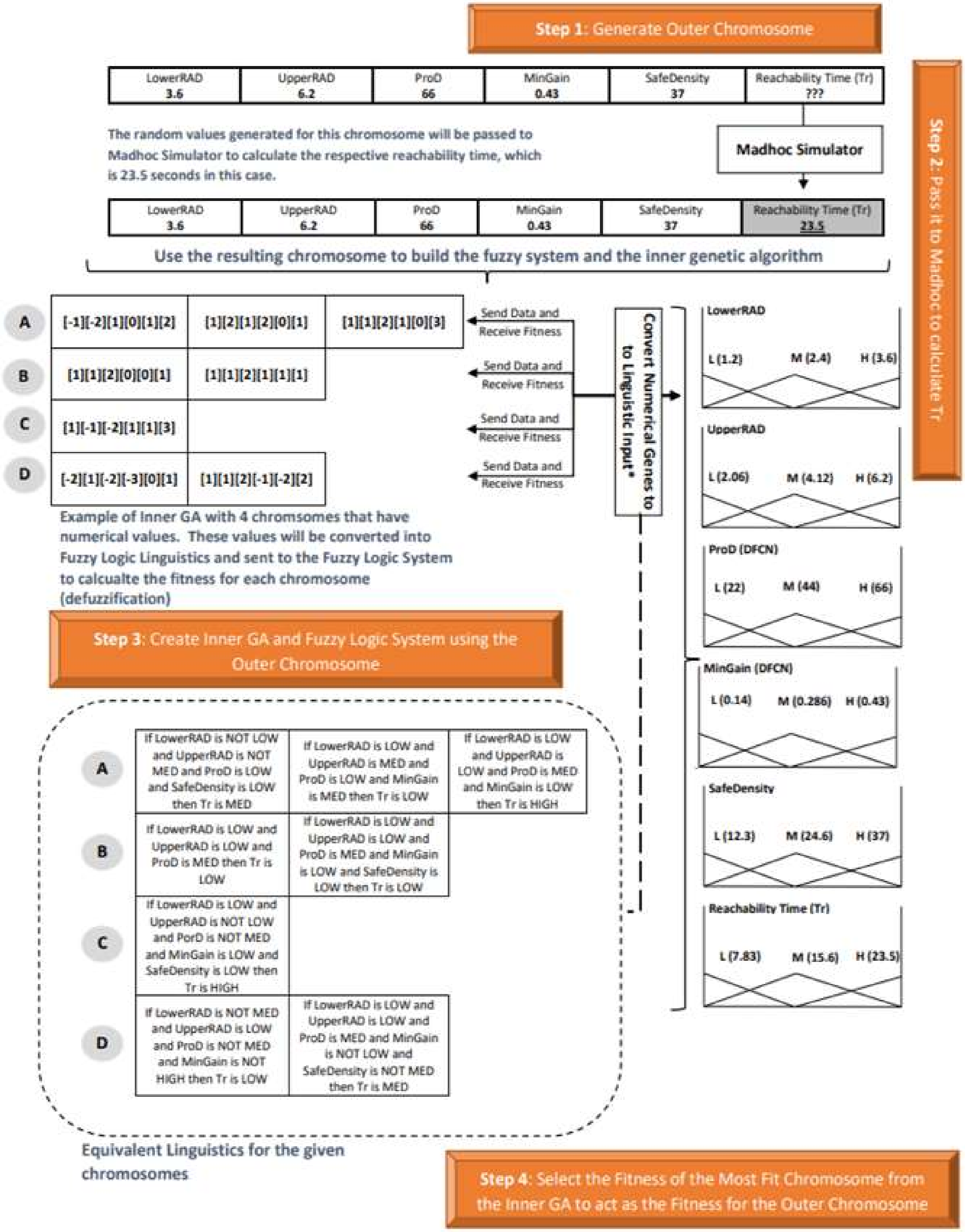

Figure V-4. Example Run for The Proposed System

5.2 Outer Genetic Algorithm

The chromosome structure for the outer GA contains a hybrid of floating-point and integer values that correspond to the DFCN parameters, and also contain the output parameter which corresponds to the reachability time that will be calculated using the Madhoc simulator.

The chromosome size for the outer GA has a fixed length of six genes. Fig.V-5 illustrates the chromosome structure. The crossover is a standard single-point operator that takes into consideration the gene placement to make sure the swapped parameters are still compatible and are within the specified thresholds.

The mutation is performed through a non-uniform operator, which can be used to limit the lower and upper boundaries for the genes - which is crucial to avoid out-of-boundaries parameters - and also because it prevents the population from stagnating during the early evolution stages. The outer population size is fixed at 100 chromosomes and runs for a maximum of 300 generations. The crossover and mutation probabilities are fixed at 30% and 10% respectively.

The selection is done through a traditional Roulette-Wheel operator. It is worth noting that the last gene (reachability time) is excluded from the evolution process and is stored inside the chromosome and passed later to the fuzzy system. All of the other aforementioned decision parameters are randomly generated within the threshold.

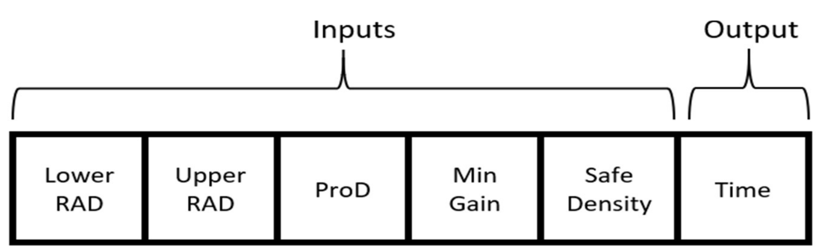

Figure V-5 Outer Chromosome Structure (Size=6)

5.3 Inner Genetic Algorithm

The inner GA uses the same operators as the outer one. However, the chromosome structure is different. It consists of a variable number of genes ranging from 3 to 15. Each gene represents a key that encodes a linguistic string into numerical values as shown previously. This had to be done in order to be able to evolve the rules using the GA. Each key has a fixed length of 6 which corresponds to the number of input parameters and the output parameter.

The population size for the inner GA is set to 50 and the maximum number of generations is 100. Fig. VI-6. illustrates a sample inner GA with a population size of 7 and random chromosome sizes, denoted with S_n, where n is the chromosome number inside the population.

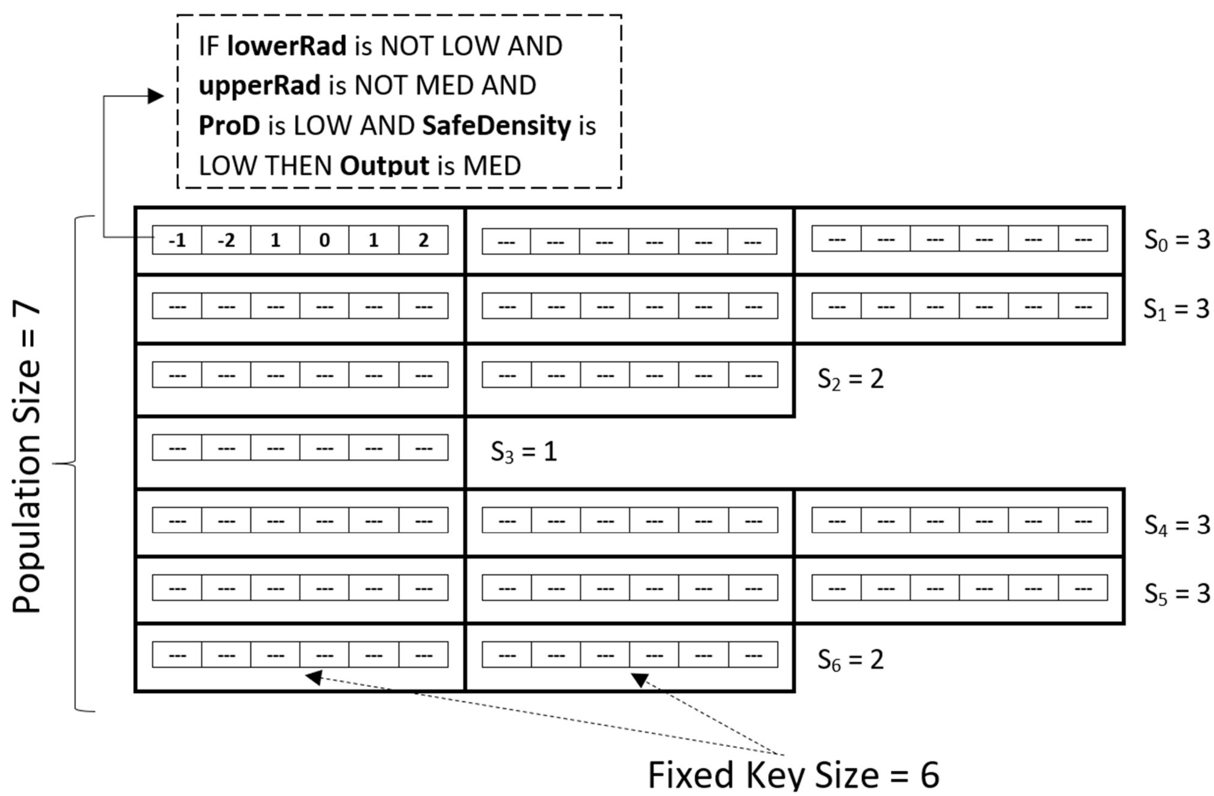

Figure V-6. Inner Genetic Algorithm Illustration

It also shows an example of how the key is decoded into a linguistic string. The inner GA makes a complete run of 50 generations for each outer chromosome. The target is to diversify the linguistics of the fuzzy logic to reach the best possible output.

The defuzzified output value represents the fitness of the outer chromosome. After doing this for all the outer GA chromosomes, the best one is chosen and the outer GA transits into the next generation.

Chapter (VI)

RESULTS AND DISCUSSION

CHAPTER VI
Results and Discussion

The experiments are run five times and the results are averaged. The results show the convergence of decision parameters and the output (solid black line). The calculated logarithmic trendline (red dotted line) provides a mathematical model for the parameters.

The trend line is calculated using a curve fitting software called “Curve Expert” since the high number of points of the 300 generations for each scenario, the “Curve Expert” software calculate a non-linear regression “Curve Fit” for the points. The software is implemented using “R” language which is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity, as of November 2019, R ranks 16th in the TIOBE index, a measure of popularity of programming languages.

5.1 Results for Highway Mobility Model

Fig. VI-(1-7) shows the results for the highway mobility environment. The time required to reach the destination decreased from 26.44 to 23.41 seconds, which amounts to 11.45%. Given that the number of nodes in this network is 80, the average time for a node to deliver a message decreased from 3.3 to 2.92 seconds. the average time for a node to deliver a message decreased from 3.3 to 2.92 seconds. The average time for each node is calculated by dividing the reachability time (tr) by the number of nodes per kilo meter square “km^2”.

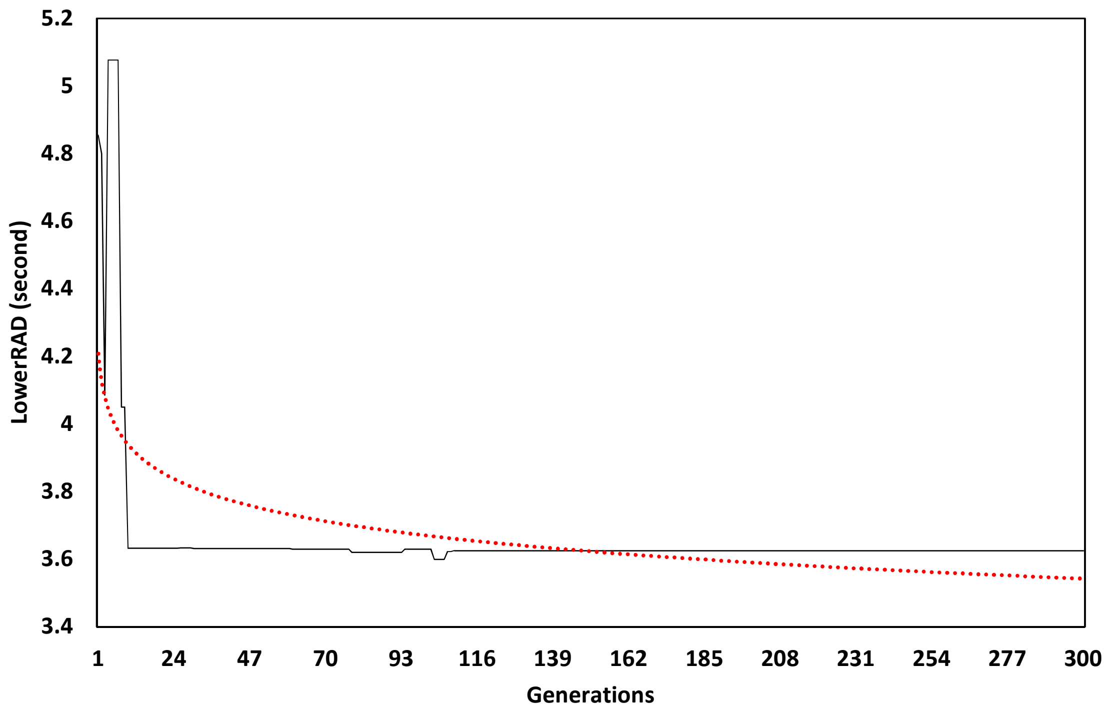

Figure VI-1. LowerRAD Convergence for The Highway Mobility Model

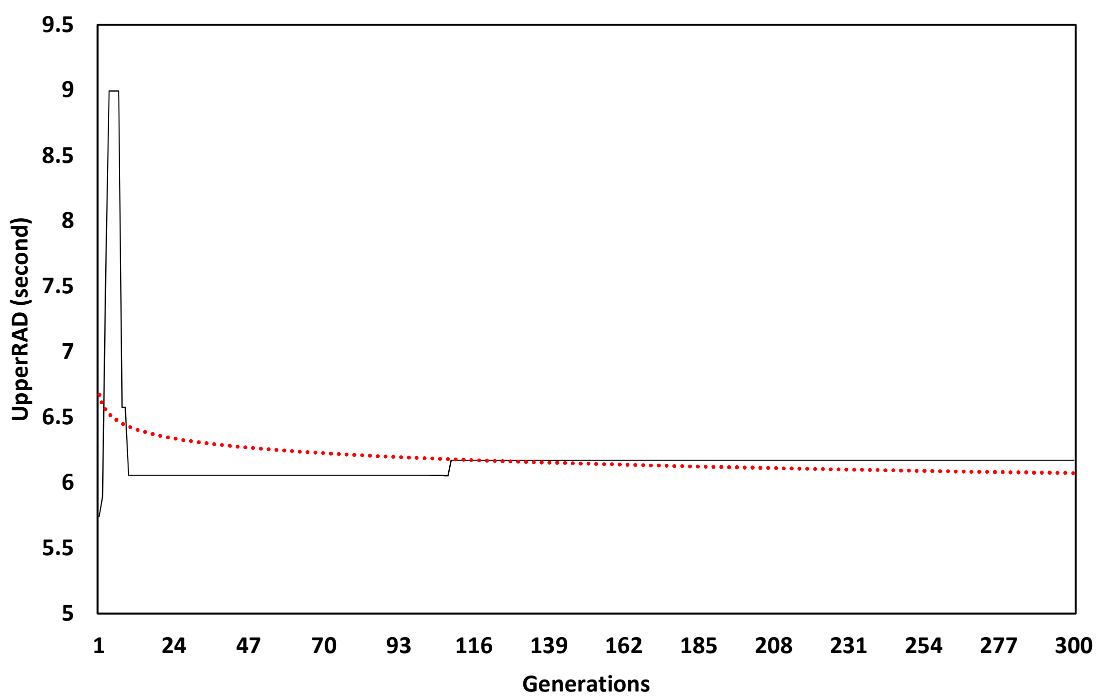

Figure VI-2. UpperRAD Convergence for The Highway Mobility Model

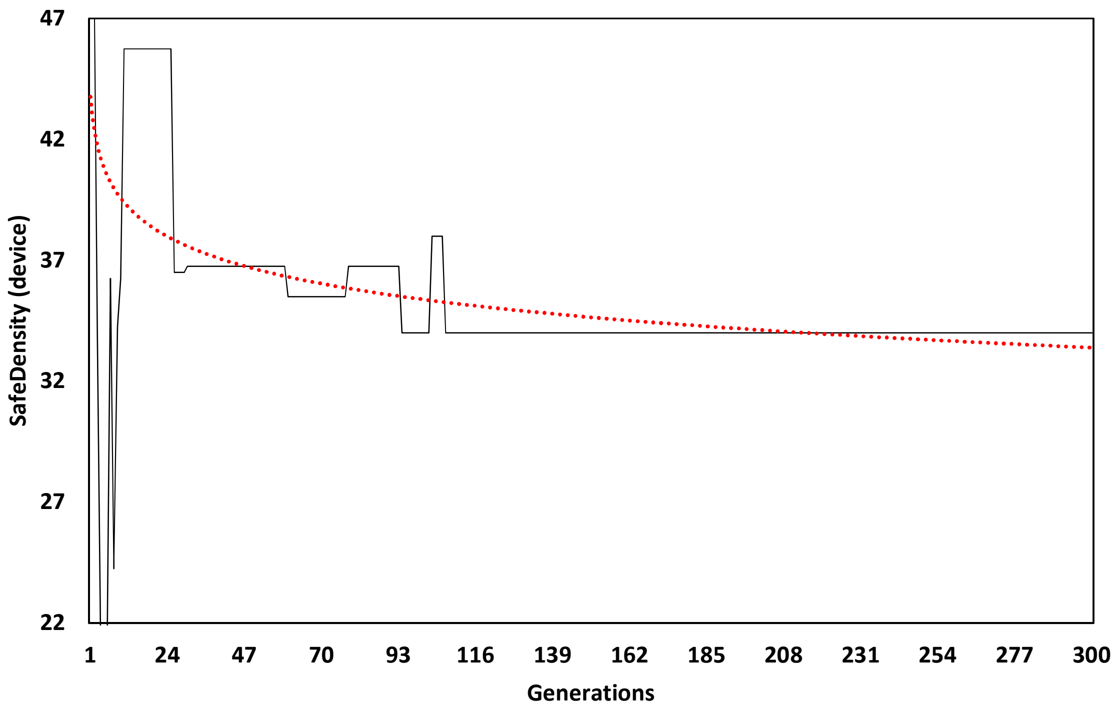

Figure VI-3. SafeDensity Convergence for The Highway Mobility Model

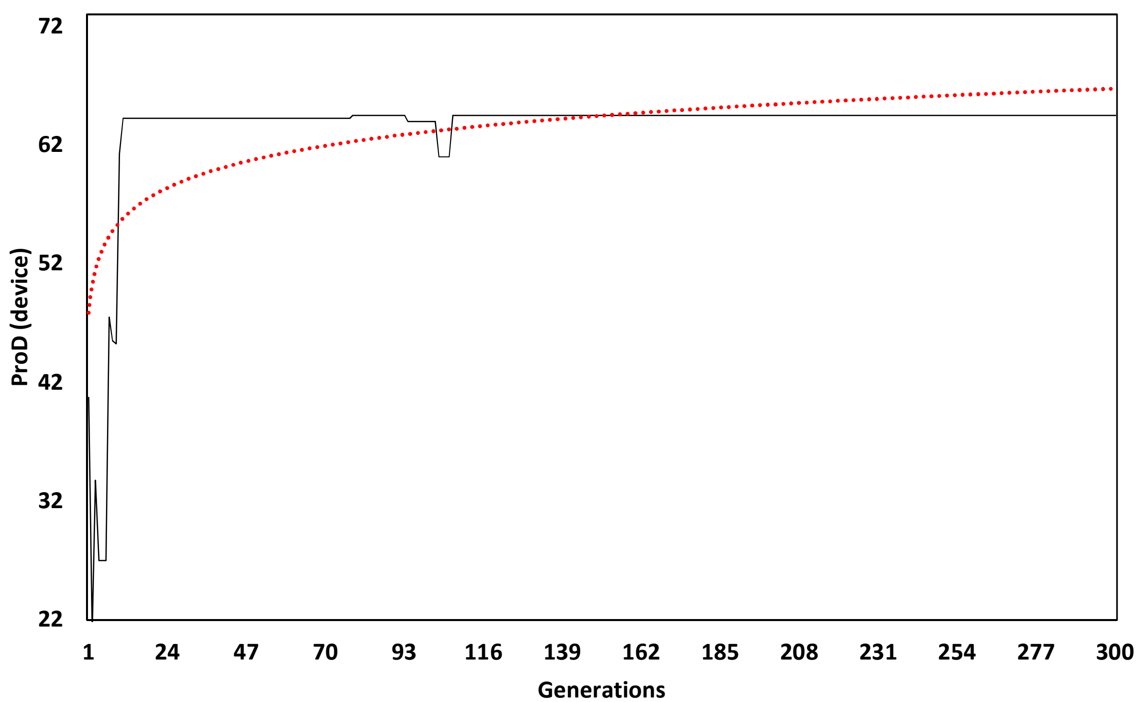

Figure VI-4. ProD Convergence for The Highway Mobility Model

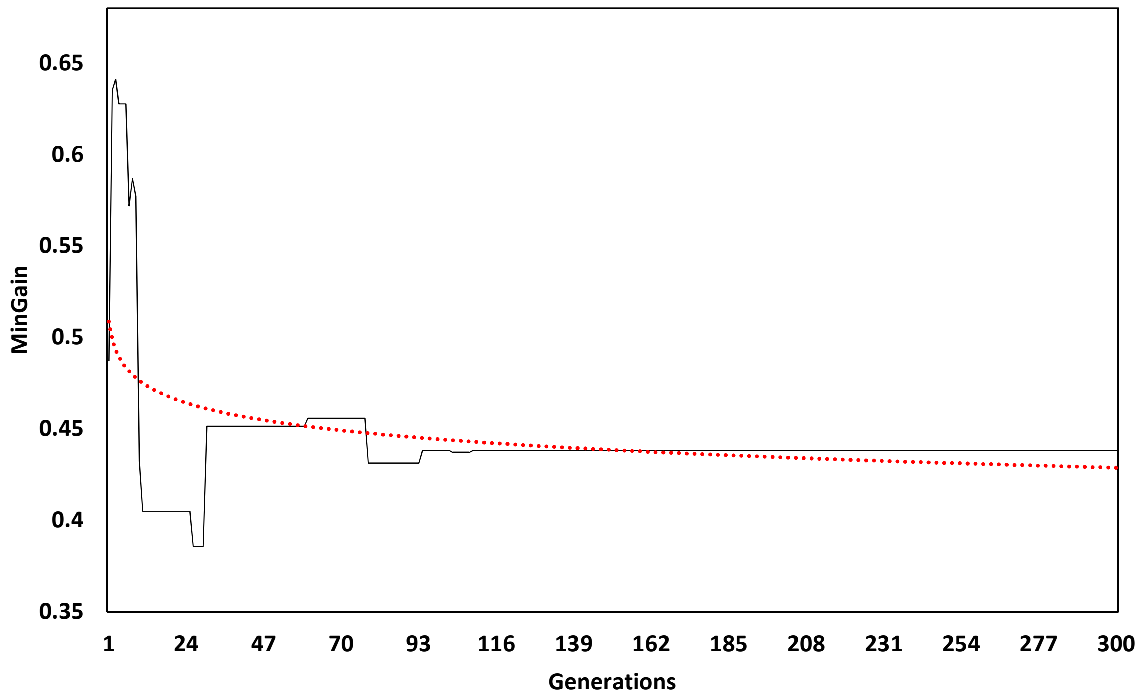

Figure VI-5. MinGain Convergence for The Highway Mobility Model

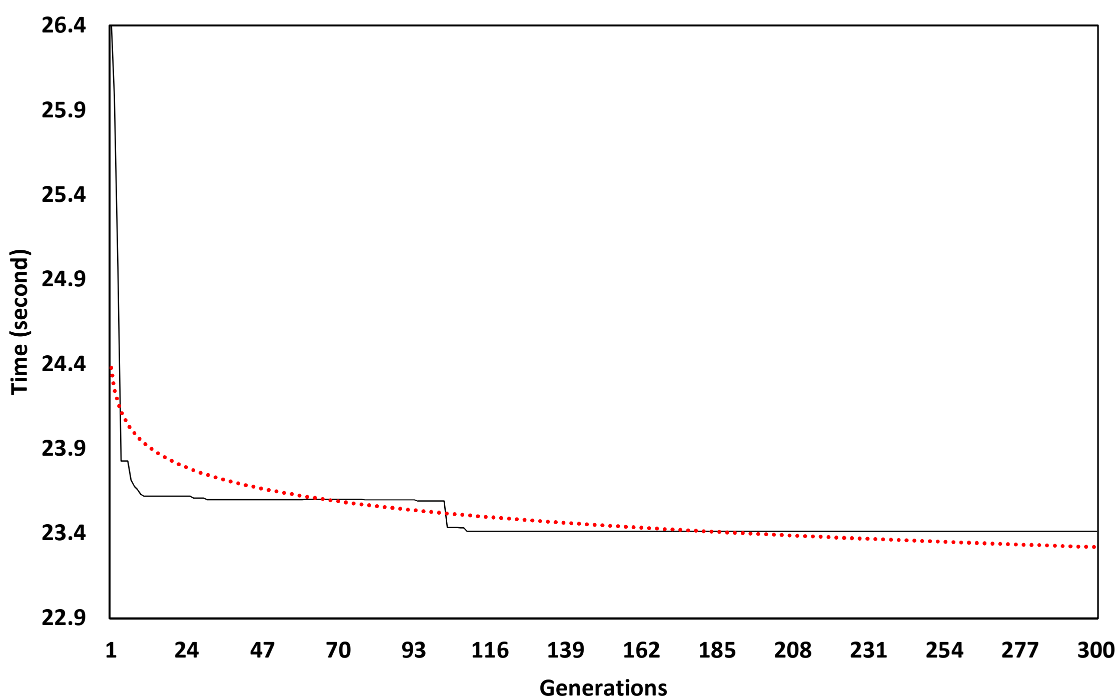

Figure VI-6. Time Convergence for The Highway Mobility Model

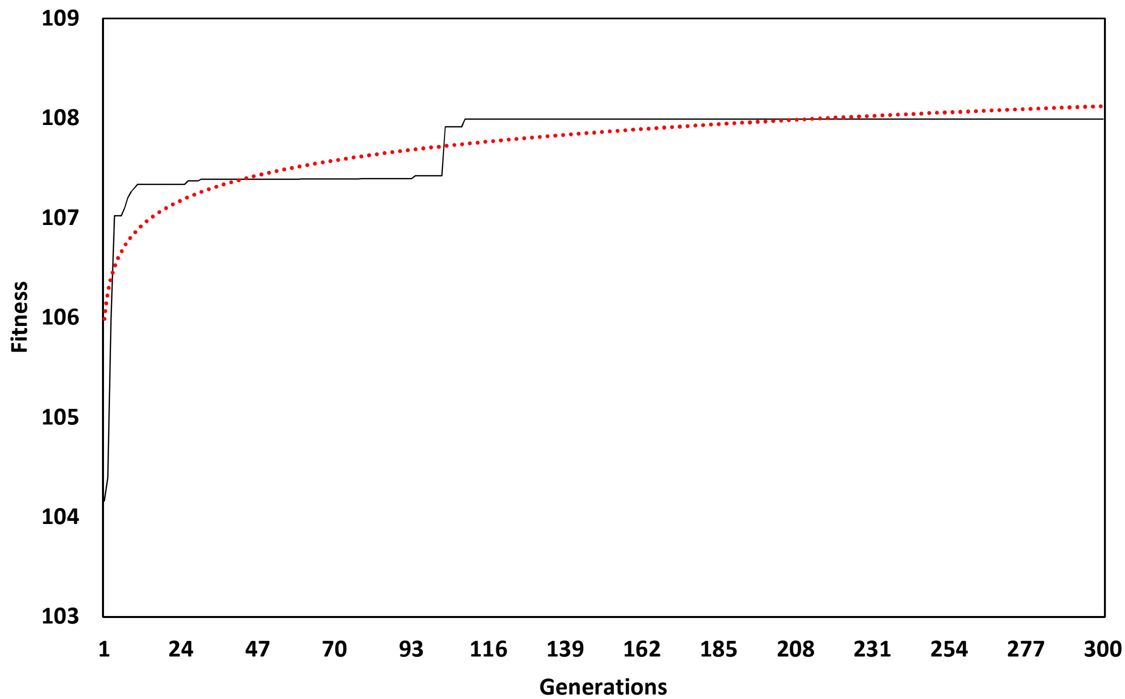

Figure VI-7. Fitness Convergence for The Highway Mobility Model

Table VI-1 shows the output trendline for each decision parameter and the equivalent logarithmic regression expressions.

Table VI-1. TRENDLINE PARAMETERS FOR THE HIGHWAY SCENARIO

Parameter	Trendline	Expression
LowerRAD	↓	$-0.117 * ln(G) + 4.2076$
UpperRAD	↓	$-0.105 * ln(G) + 6.6723$
ProD	↑	$3.3079 * ln(G) + 47.885$
MinGain	↓	$-0.014 * ln(G) + 0.5087$
SafeDensity	↓	$-1.818 * ln(G) + 43.76$

5.2 Results for Mall Mobility Model

Fig. VI-(8-14) shows the results for the Mall mobility scenario and Table VI-2 shows the trendline for the decision parameters. The time to reach the nodes decreased from 4.98

seconds to 3.57 seconds which amounts to 28.3%, which brings down the average required time to deliver a message from 7.6ms to 5.49ms.

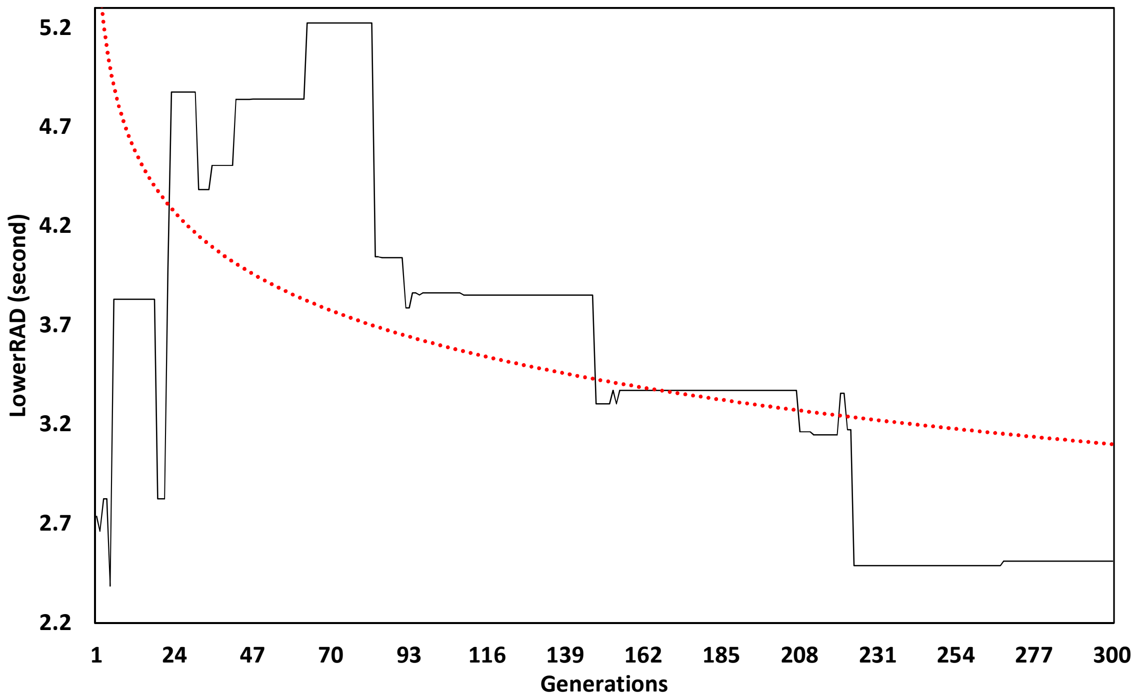

Figure VI-8. LowerRAD Convergence for The Mall Mobility Model

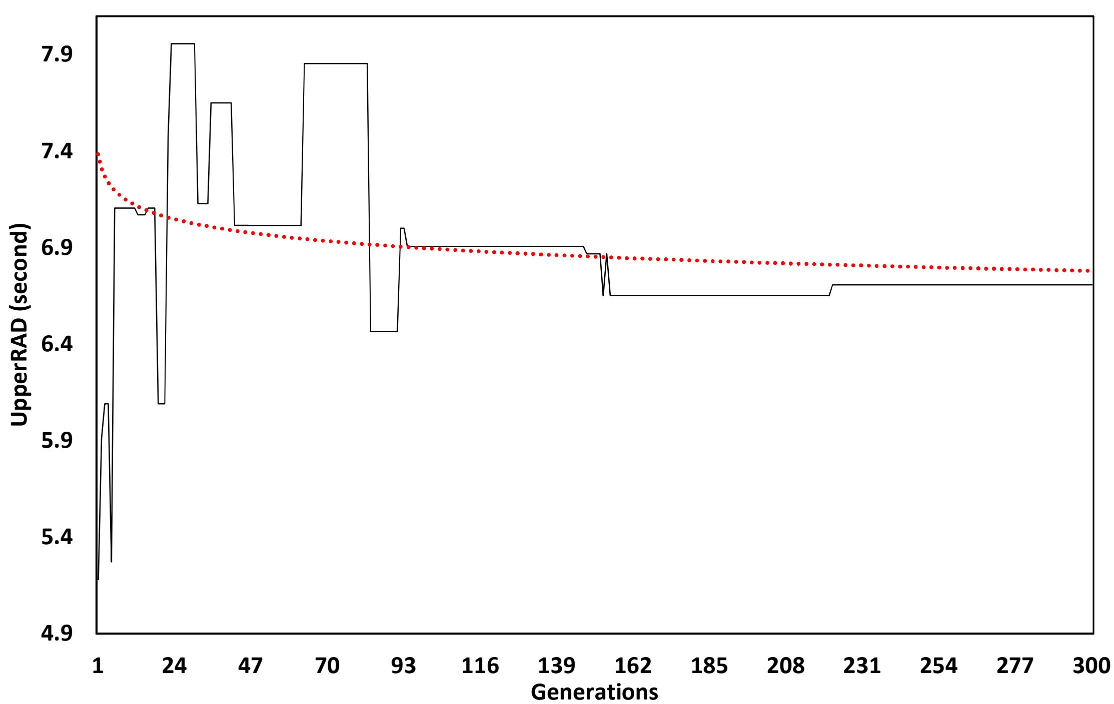

Figure VI-9. UpperRAD Convergence for The Mall Mobility Model

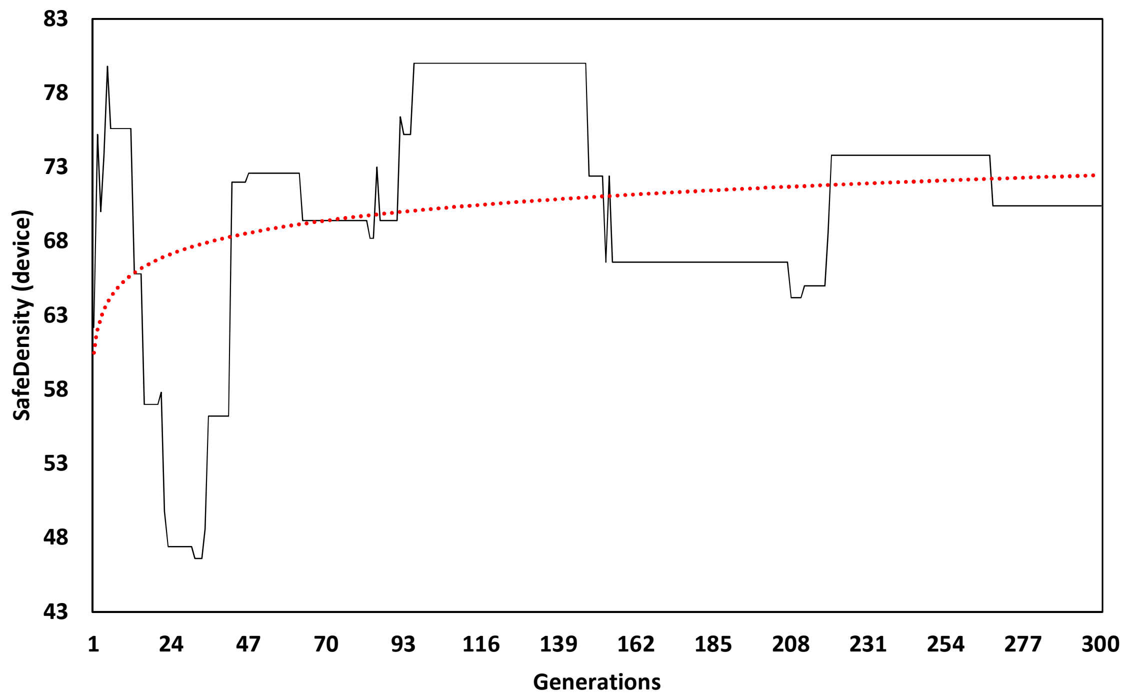

Figure VI-10. SafeDensity Convergence for The Mall Mobility Model

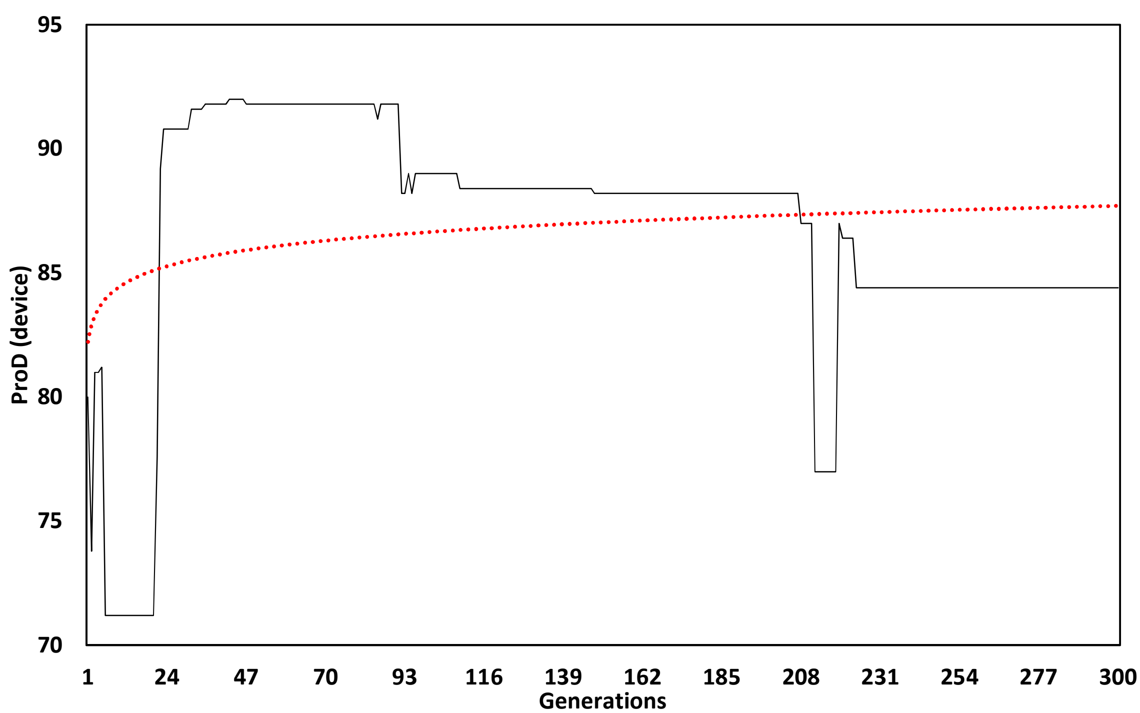

Figure VI-11. ProD Convergence for The Mall Mobility Model

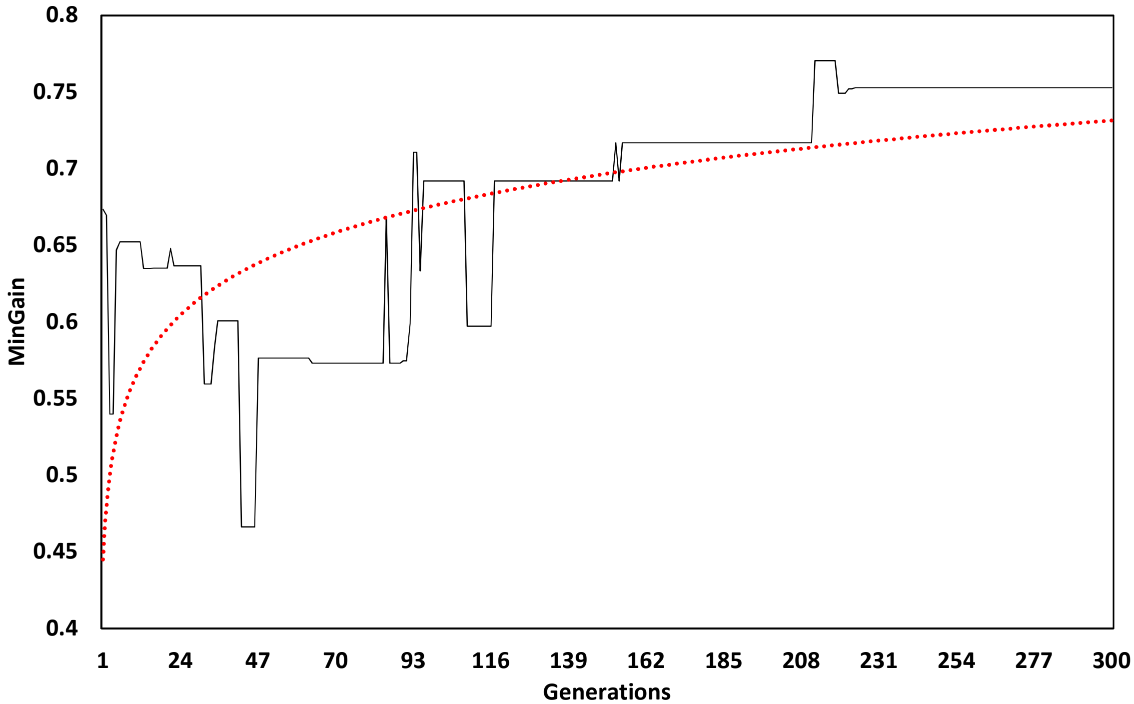

Figure VI-12. MinGain Convergence for The Mall Mobility Model

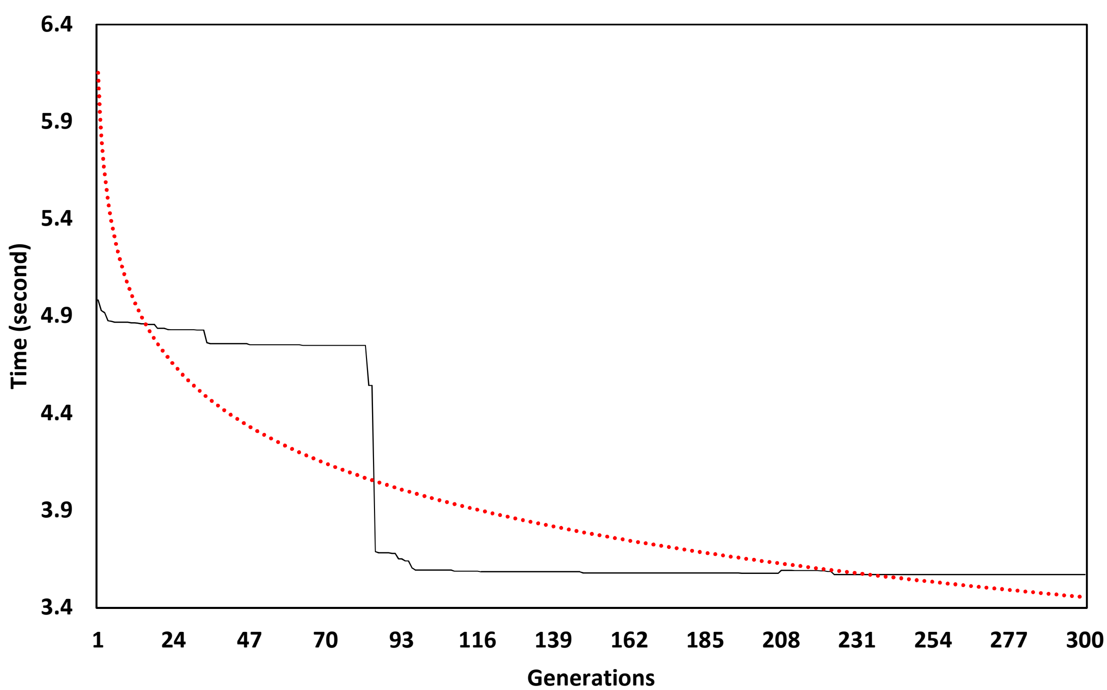

Figure VI-13. Time Convergence for The Mall Mobility Model

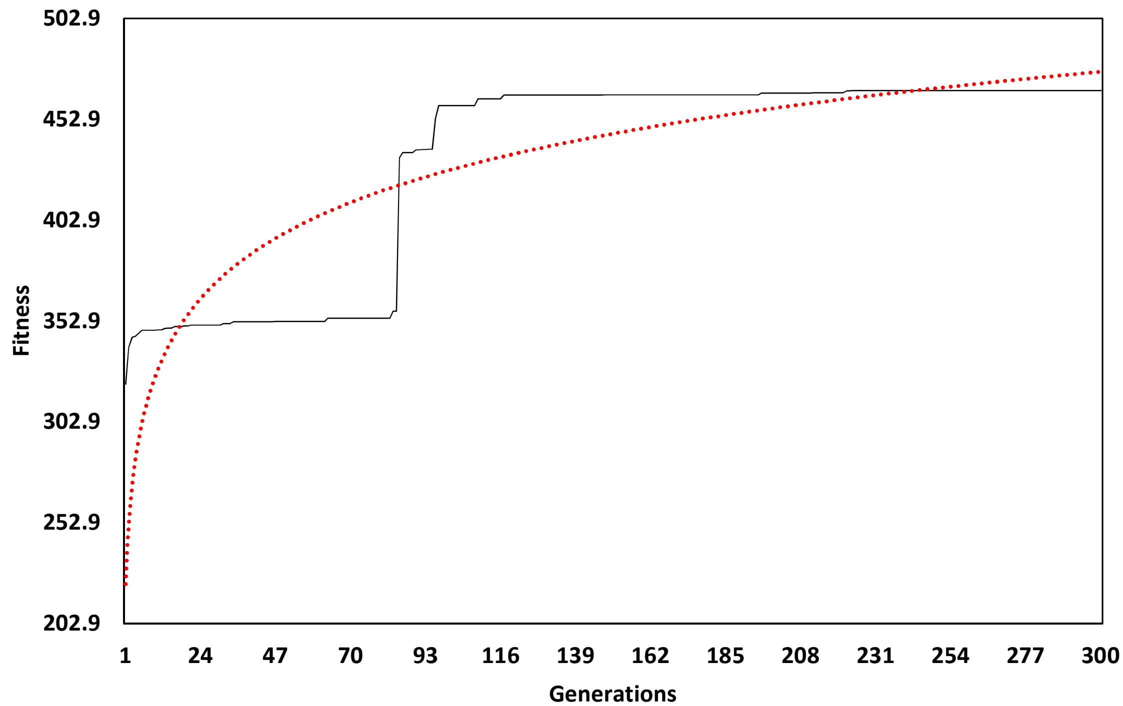

Figure VI-14. Fitness Convergence for The Mall Mobility Model

Table VI-2. TRENDLINE PARAMETERS FOR MALL SCENARIO

Parameter	Trendline	Expression
LowerRAD	↓	$-0.463 * ln(x) + 5.7409$
UpperRAD	↓	$-0.106 * ln(x) + 7.3855$
ProD	↑	$0.9608 * ln(x) + 82.223$
MinGain	↑	$0.0502 * ln(x) + 0.445$
SafeDensity	↑	$2.0984 * ln(x) + 60.488$

5.3 Results for Human Mobility Model

The results for the human mobility model are shown in Fig. VI-(15-21) and the respective trendline parameters are shown in Table VI-3. As for the human mobility scenario, the time to deliver the messages to their respective destinations decreased from 4.07 to 3.78 seconds, which amounts to 7.12%. The average time to deliver a message decreased from 5ms to 4.6ms.

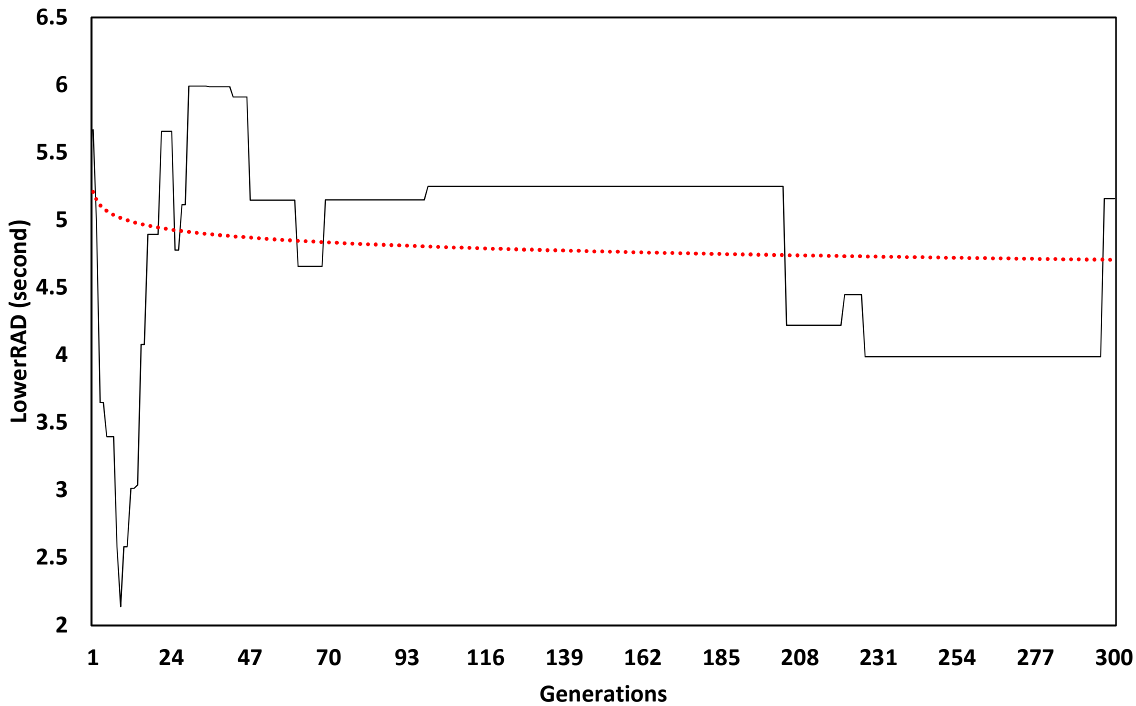

Figure VI-15. LowerRAD Convergence for The Human Mobility Model

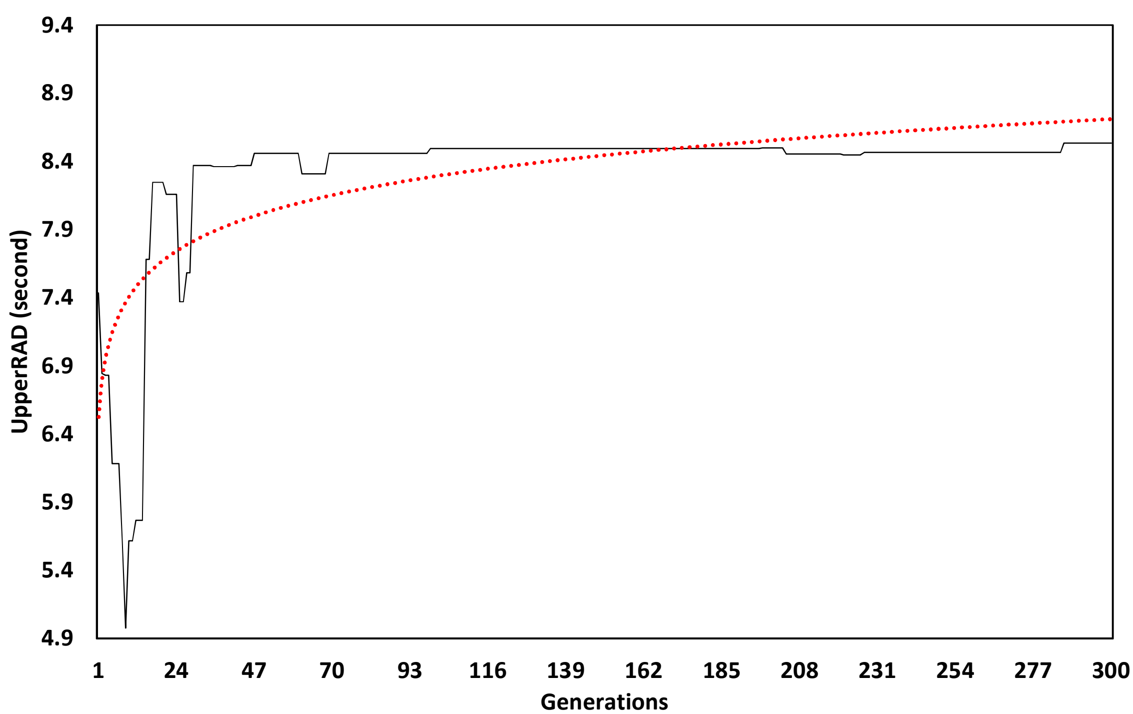

Figure VI-16. UpperRAD Convergence for The Human Mobility Model

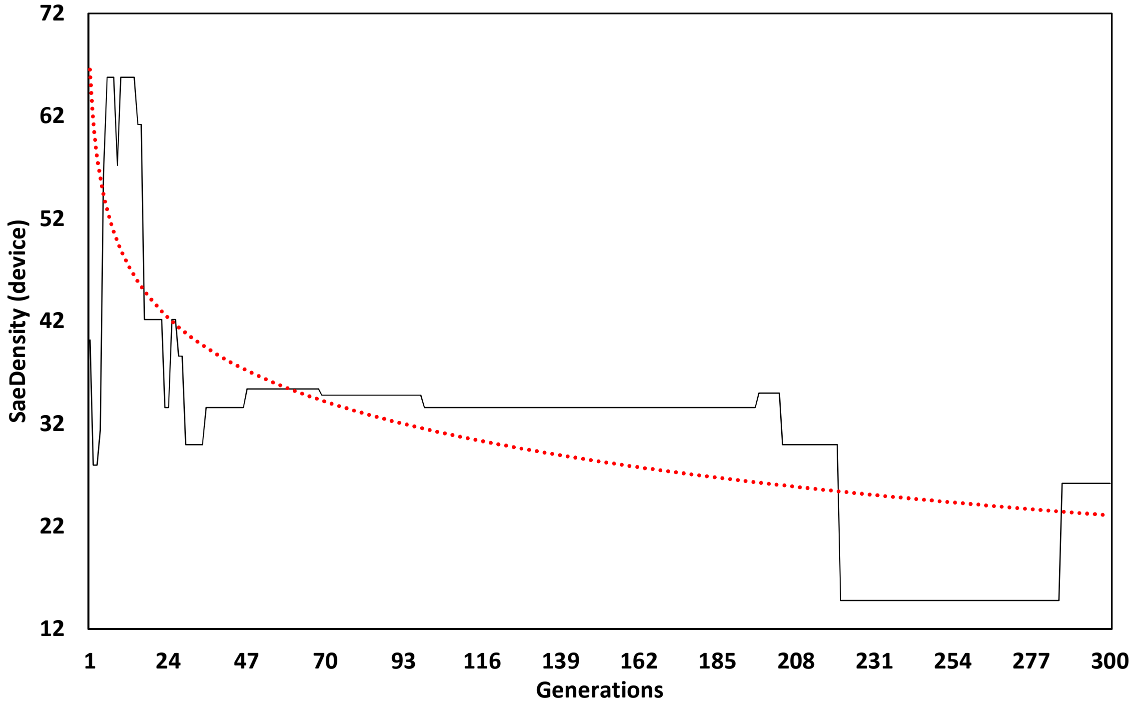

Figure VI-17. SafeDensity Convergence for The Human Mobility Model

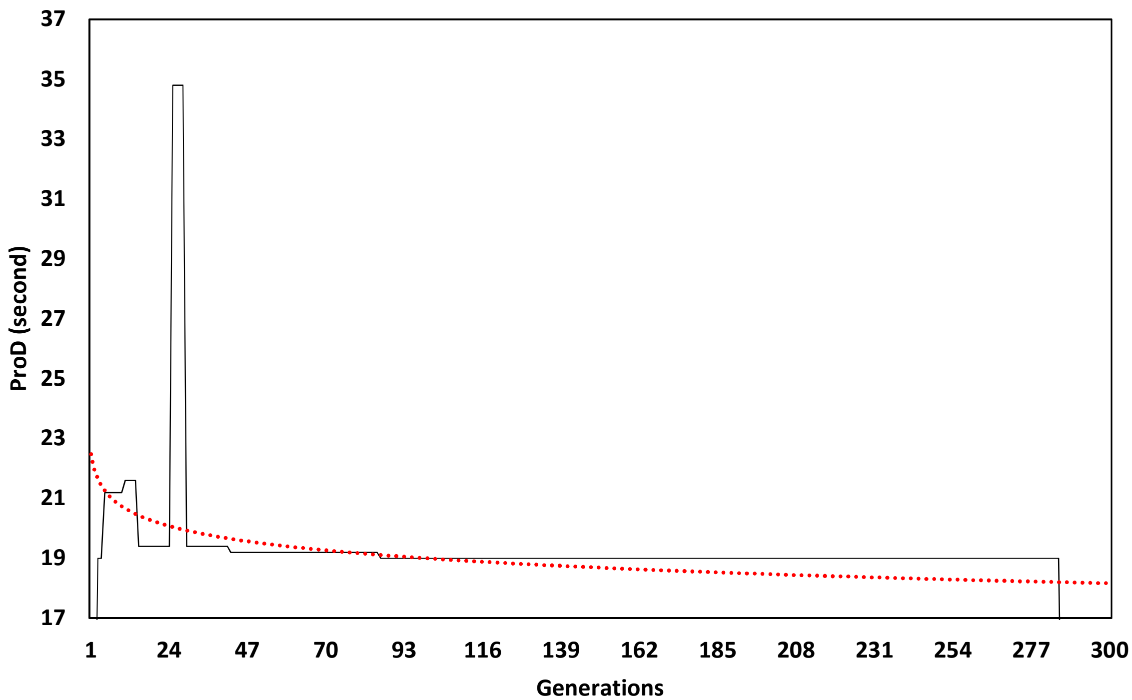

Figure VI-18. ProD Convergence for The Human Mobility Model

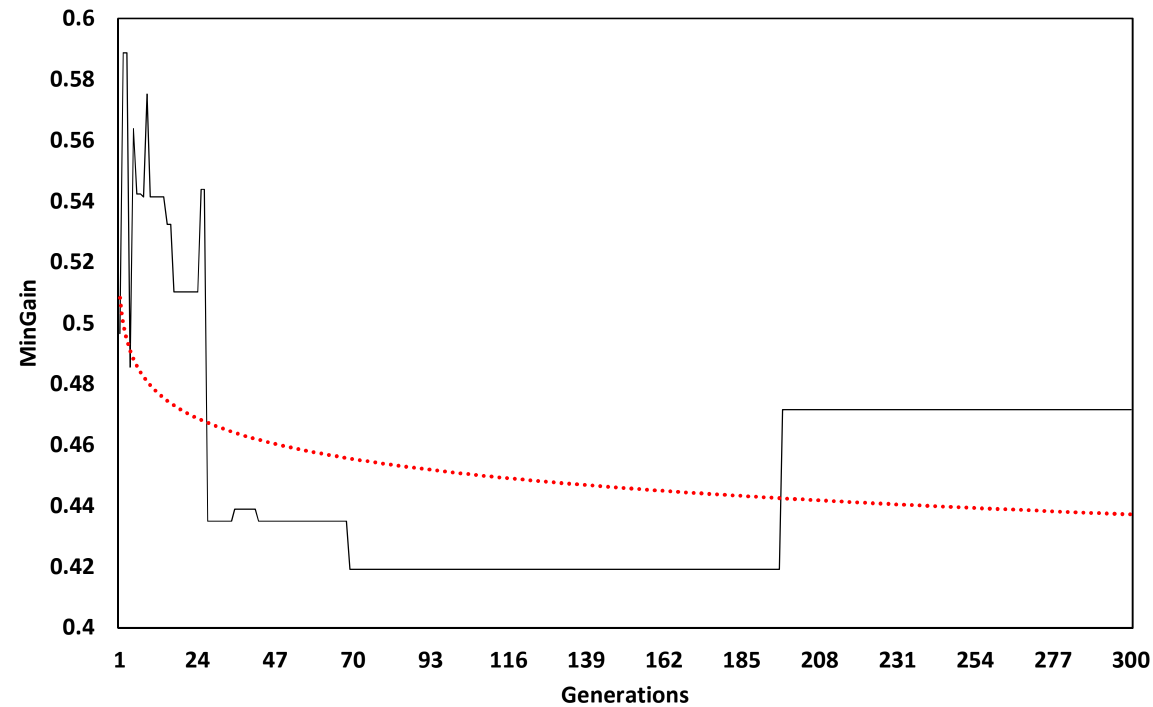

Figure VI-19. MinGain Convergence for The Human Mobility Model

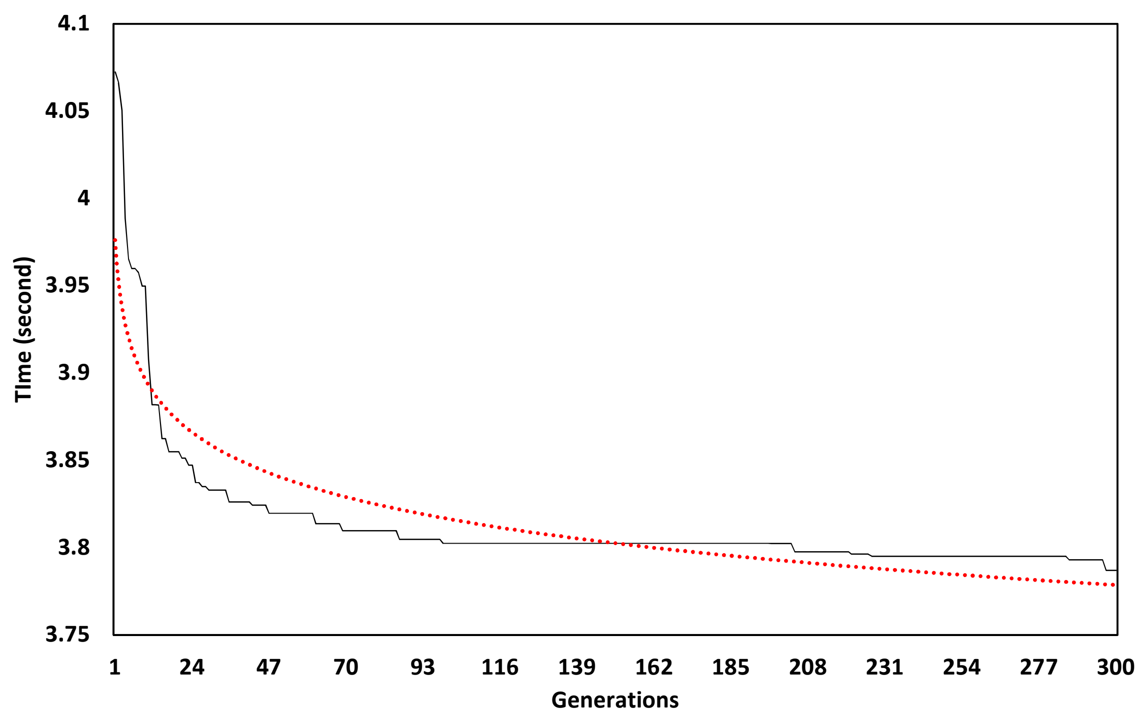

Figure VI-20. Time Convergence for The Human Mobility Model

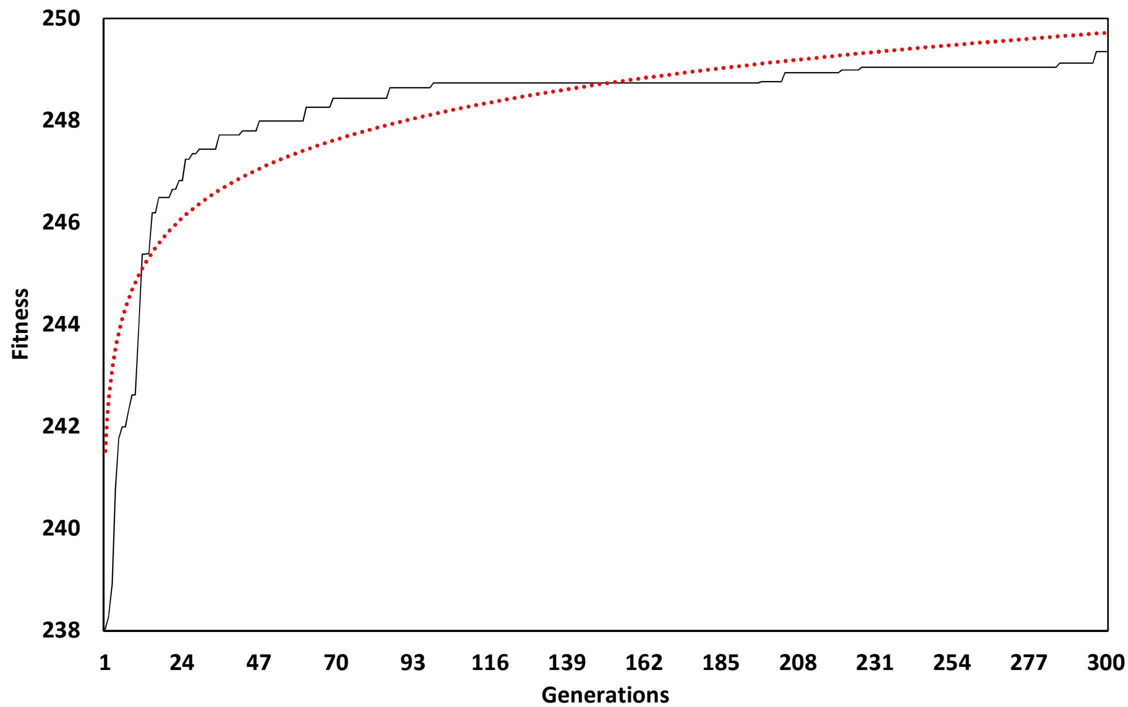

Figure VI-21. Fitness Convergence for The Human Mobility Model

Table VI-3. TRENDLINE PARAMETERS FOR HUMAN MOBILITY SCENARIO

Parameter	Trendline	Expression
LowerRAD	↓	$-0.088 * ln(x) + 5.2098$
UpperRAD	↑	$0.3832 * ln(x) + 6.526$
ProD	↓	$-0.757 * ln(x) + 22.479$
MinGain	↓	$-0.012 * ln(x) + 0.5084$
SafeDensity	↓	$-7.621 * ln(x) + 66.544$

5.4 Discussion

By inspecting all the previous results, it appears that the mall mobility model benefited the most from the optimization of the DFCN decision parameters and the human mobility model benefited the least. While these two models have very close features, the major difference between them, as stated previously, is the randomness of the movements. The human mobility model is governed by human intentions of moving between a dynamic list of targets while the mall one is governed by random motion of shoppers moving between random shops. Also, by inspecting the highway scenario, it seems that the lack of enough nodes has significantly raised the average delivery time six times (6x) the delivery time in other scenarios.

As for the highway scenario, one of the most notable outcomes, is the range for the RAD parameters. Both the UpperRAD and the LowerRAD are uniformly decreasing throughout the whole GA evolution process without fluctuating. This can be attributed to the fact that the nodes are sparse (situated far apart) and are moving relatively fast.

The LowerRAD assures that the minimum time for rebroadcasting is as low as possible to avoid missing the chances of delivering a message to a nearby fast-moving node. It can also be seen that the ProD parameter has increased in the same manner. This is responsible for the network sensitivity or awareness towards newly introduced neighbours. This optimization model can decrease the probabilities of a broadcasting storm from happening.

To demonstrate the consistency and statistical validity of the achieved results, a 5% confidence interval for the final reachability time is calculated and is shown in Table VI-4.

Table VI-4. 5% CONFIDENCE INTERVAL FOR THE FINAL TIME

Mobility Model	**5% Confidence Interval (seconds)**
Highway	23.4 ± 0.75
Mall	3.57 ± 0.34
Human	3.78 ± 0.02

Chapter (VII)

CONCLUSION AND FUTURE WORK

CHAPTER VII
Conclusion and Future Work

The proposed system managed to decrease the message delivery time for the three real-life scenarios (the highway, the mall and the human mobility models) by optimizing the decision parameters for the DFCN protocol.

The mall mobility model benefited the most from the optimization of the DFCN parameters, which is mainly attributed to the randomness of the mobility, since the human mobility model also shares very close parameters but only differs in the movement intention.

In the human mobility model, the mobility is governed by the intentions of the humans to reach a certain dynamic list of destinations and, therefore, the randomness significantly decreases. Also, the highway mobility model yielded the highest average message delivery time, which is attributed to the lack of nodes and the very high mobility speed, and since the DFCN protocol relies on 1-hop neighbours to deliver the messages to their destinations, this scenario severely affects it.

In the future, a mathematical model that is based on the calculated trendlines can be established and tested. This will help achieve faster results, instead of relying solely on metaheuristic techniques, which require a significant amount of time to converge to the optimal solutions. Another key-point to consider is to test the proposed system with a higher number of reachable nodes, other than the specified 10%. This will help understand how the tuned parameters of the DFCN protocol respond to the increased number of nodes for the same scenarios.

This mathematical model can also help with real time broadcasting in MANETs and will allow the network to dynamically readjust the broadcasting parameters as the network density and structure change.

Also, Genetic Programming (GP) can be experimented with, in order to evolve programs and expressions dedicated to each one of the three aforementioned scenarios. The resulting programs can then be used as a rigid optimization model without the need to repeat the evolution process each time like GA.

References

[1] V. Rishiwal, S. K. Agarwal and M. Yadav, “Performance of AODV protocol for H-MANETs,” in *International Conference on Advances in Computing, Communication, & Automation*, Dehradun, 2016.

[2] L. J. G. Villalba, J. G. Matesanz, A. L. S. Orozco and J. D. M. Díaz, “Auto-Configuration Protocols in Mobile Ad Hoc Networks,” *Sensors (Basel),* vol. 11, no. 4, pp. 3652-3666, 2011.

[3] C. Dhakad and A. S. Bisen, “Efficient route selection by using link failure factor in MANET,” in *International Conference on Electrical, Electronics, and Optimization Techniques*, Chennai, 2016.

[4] P.-J. Chuang, P.-H. Yen and T.-Y. Chu, “Efficient Route Discovery and Repair in Mobile Ad-hoc Networks,” in *IEEE 26th International Conference on Advanced Information Networking and Applications*, Fukuoka, 2012.

[5] V. Sharma and A. Vij, “Broadcasting methods in mobile ad-hoc networks,” in *International Conference on Computing, Communication and Automation*, Greater Noida, India, 2017.

[6] M. Bakhouya, “Broadcasting approaches for Mobile Ad hoc Networks,” in *International Conference on High Performance Computing & Simulation*, Helsinki, 2013.

[7] M. Vanjale, J. S. Chitode and S. Gaikwad, “Residual Battery Capacity Based Routing Protocol for Extending Lifetime of Mobile Ad Hoc Network,” in *International Conference On Advances in Communication and Computing Technology*, Sangamner, India, 2018.

[8] H. Yadav and H. K. Pati, “A Survey on Selfish Node Detection in MANET,” in *International Conference on Advances in Computing, Communication Control and Networking*, Greater Noida (UP), 2018.

[9] N. Ramya and S. Rathi, “Detection of selfish Nodes in MANET - a survey,” in *International Conference on Advanced Computing and Communication Systems*, Coimbatore, 2016.

[10] L. Hogie, P. Bouvry, M. Seredynski and F. Guinand, “A Bandwidth-Efficient Broadcasting Protocol for Mobile Multi-hop Ad hoc Networks,” in *InternationInternational Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, Morne, 2006.

[11] K. Kamran, S. Afzal, M. Yaqoob and M. Sharif, “A Comparative Survey on Vehicular Ad-hoc Network (VANET) Routing Protocol using Heuristic and Optimistic Techniques,” *Research Journal of Information Technology,* vol. 6, no. 2, pp. 14-24, 2015.

[12] R. Li and H. Asaeda, "A community-oriented route coordination using information centric networking approach," in *38th Annual IEEE Conference on Local Computer Networks*, Sydney, NSW, Australia, 2014.

[13] B. Dorronsoro, P. Ruiz, G. Danoy, Y. Pigné and P. Bouvry, "BROADCASTING PROTOCOL," in *Evolutionary Algorithms for Mobile Ad Hoc Networks*, John Wiley & Sons, Inc., 2014, pp. 135-138.

[14] L. Hogie, P. Bouvry and F. Guinand, "An Overview of MANETs Simulation," *Electronic Notes in Theoretical Computer Science,* vol. 150, no. 1, pp. 81-101, 2006.

[15] S. Mirjalili, Evolutionary Algorithms and Neural Networks, Springer International Publishing, 2019.

[16] I. Boussaïd, J. Lepagnot and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences,* vol. 10, no. 237, pp. 82-117, 2013.

[17] S. Desale, A. Rasool, S. Andhale and P. Rane, "Heuristic and Meta-Heuristic Algorithms and their Relevance to the Real World: A Survey," *International Journal of Computer Engineering in Research Trends,* vol. 2, no. 5, pp. 296-304, 2015.

[18] M. Mitchell, An Introduction to Genetic Algorithms, 1996.

[19] J. Jiang, L. D. Meng and X. M. Xu, "The Study on Convergence and Convergence Rate of Genetic Algorithm Based on an Absorbing Markov Chain," *Applied Mechanics and Materials,* Vols. 239-240, pp. 1511-1515, 2012.

[20] J. Lee, H. Jeong and S. Kang, "Derivative and GA-based methods in metamodeling of back-propagation neural networks for constrained approximate optimization," *Structural and Multidisciplinary Optimization,* vol. 35, no. 1, pp. 29-40, 2007.

[21] A. Sehgal, H. La, S. Louis and H. Nguyen, "Deep Reinforcement Learning Using Genetic Algorithm for Parameter Optimization," in *Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, 2019.

[22] R. Serrano, J. Tapia, O. Montiel, R. Sepúlveda and P. Melin, "High Performance Parallel Programming of a GA Using Multi-core Technology," in *Soft Computing for Hybrid Intelligent Systems* , Springer, 2008, pp. 307-314.

[23] J. Sadeghi, S. Sadeghi and S. T. A. Niaki, Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm, ELSEVIER, 2014.

[24] H. H. Örkcü and H. Bal, Comparing performances of backpropagation and genetic algorithms in the data classification, vol. 38, Elsevier, 2011.

[25] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," *Journal of Financial Economics,* vol. 51, no. 2, pp. 245-271, 1999.

[26] G. G. Yen and N. Nithianandan, "Facial feature extraction using genetic algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, Honolulu, 2002.

[27] Y. Zhao, S. Wu, L. Reynolds and S. Azenkot, “A Face Recognition Application for People with Visual Impairments: Understanding Use Beyond the Lab,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC, Canada, 2018.

[28] S. Chien, Z. Yang and E. Hou, “Genetic Algorithm Approach for Transit Route Planning and Design,” *Journal of Transportation Engineering,* vol. 127, no. 3, 2001.

[29] A. Ghaffari, H. Abdollahi, M. Khoshayand, I. S. Bozchalooi, A. Dadgar and M. Rafiee-Tehrani, “Performance comparison of neural network training algorithms in modeling of bimodal drug delivery,” *International Journal of Pharmaceutics,* vol. 327, no. 1-2, pp. 126-138, 2006.

[30] L. Zadeh, “Fuzzy logic,” *Computer,* vol. 21, no. 4, pp. 83-93, 1988.

[31] J. Alcala-Fdez and J. M. Alonso, “A Survey of Fuzzy Systems Software: Taxonomy, Current Research Trends, and Prospects,” in *IEEE Transactions on Fuzzy Systems,* , 2016.

[32] E. Cox, “Fuzzy fundamentals,” *IEEE Spectrum,* vol. 29, no. 10, pp. 58-61, 1992.

[33] L. Hogie, F. Guinand and P. Bouvry, *The Madhoc metropolitan ad hoc network simulator,* France: Luxembourg University and Le Havre University, 2006.

[34] L. Hogie, “Madhoc Metropolitan ad hoc network simulator,” 2006. [Online]. Available: http://www.i3s.unice.fr/~hogie/madhoc/. [Accessed 25 January 2019].

[35] R. M. Chintalapalli and V. R. Ananthula, “M-LionWhale: multi-objective optimisation model for secure routing in mobilead-hocnetwork,” *IET Communications,* vol. 12, no. 12, pp. 1406 - 1415, 2018.

[36] E. Alba, B. Dorronsoro, F. Luna and P. Bouvry, “A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs,” in *IEEE International Parallel and Distributed Processing Symposium*, Denver, CO, USA, 2005.

[37] S. Subramaniyan, W. Johnson and K. Subramaniyan, “A distributed framework for detecting selfish nodes in MANET using Record- and Trust-Based Detection (RTBD) technique,” *EURASIP Journal on Wireless Communications and Networkingvolume,* p. Article 205, 2014.

[38] S. S. Basurra, M. D. Vos, J. Padget, Y. Ji, T. Lewis and S. Armou, “Energy Efficient Zone based Routing Protocol for MANETs,” *Ad Hoc Networks,* vol. 25, pp. 16-37, 2015.

[39] M. Ahmad, A. Hameed, A. A. Ikram and I. Wahid, “State-of-the-Art Clustering Schemes in Mobile Ad Hoc Networks: Objectives, Challenges, and Future Directions,” *IEEE ACCESS,* vol. 7, pp. 17067 - 17081, 2019.

[40] R. Carvajal-Gomez, Y.-D. Bromberg, Y. Elkhatib, L. Reveillere and E. Riviere, “Emergent Overlays for Adaptive MANET Broadcast,” in *Symposium on Reliable Distributed Systems*, Lyon France, 2019.

[41] Y.-C. Tseng, S.-Y. Ni and E.-Y. Shih, “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network,” *IEEE Transactions on Computers,* vol. 52, no. 5, pp. 545 - 557, 2003.

[42] K. Viswanath and K. Obrazcka, “An Adaptive Approach to Group Communications in Multi Hop Ad Hoc Networks,” in *Proceeding ISCC '02 Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, Washington, DC, USA, 2002.

Appendices

Publication based on thesis

- **N. S. Eissa, A. Z. Talha, A. F. Amin and A. Badr, "A Nested Genetic Algorithm for Mobile Ad-Hoc Network Optimization with Fuzzy Fitness," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 10, no. 9, pp. 222-228, 2019**

Arabic Abstract
الملخص العربي

احد المشاكل التى تواجه شبكة الجوال الارتجالية هى البث لجميع من فى الشبكة, و التي تشكل جزءا هاما جدا من البنية التحتية لهذه الشبكات. تقدم هذه الرسالة خوارزمية جينية متداخلة مع جودة فازية لتحسين وصول رسالة البث لتلك الشبكة. بينما عادةً ما يعتبر تحسين البث مشكلة متعددة الأهداف مع معلمات الإخراج المختلفة التي تتطلب توليف, النظام المقترح يطرق نهج آخر يركز على معلمة الإخراج الواحد, و هو زمن الوصول لرسالة البث فى هذه الشبكة.

هذا هو الوقت اللازم للوصول إلى البيانات بالنسبة إلى نسبة معينة من العملاء المتصلين في الشبكة. تم تحسين الوقت من خلال ضبط معلمات القرار المختلفة لبروتوكول البث المتأخر للفيضانات مع الحي التراكمي (DFCN).تم تطوير النظام المقترح ومحاكاته بمساعدة محاكي شبكة Madhoc وتطبيقه على سيناريوهات واقعية مختلفة.

تكشف النتائج أن وقت إمكانية الوصول يستجيب جيدًا للنظام المقترح ويوضح أن كل سيناريو يستجيب بشكل مختلف لضبط معلمات القرار.

الأكاديمية العربية للعلوم و التكنولوجيا و النقل البحري

كلية الهندسة و التكنولوجيا – القاهرة
قسم هندسة الحاسبات

خوارزمية جينية متداخلة لأمثلية شبكة الجوال الارتجالية بجودة فازية

إعداد

مهندس/ احمد زكريا نور محمد طلحه

رسالة مقدمة للأكاديمية العربية للعلوم والتكنولوجيا والنقل البحرى
لإستكمال متطلبات نيل درجة *الماجستير فى*

هندسة الحاسبات

الأستاذ الدكتور/ أحمد فهمى أمين
مشرف

الأستاذ الدكتور/ عمرو أنور بدر
مشرف

الأستاذ الدكتور/ عبد المنعم عبد الظاهر وهدان
ممتحن

الأستاذ الدكتور/ جمال ابراهيم سليم
ممتحن

2020

الأكاديمية العربية للعلوم و التكنولوجيا و النقل البحري

كلية الهندسة و التكنولوجيا – القاهرة
قسم هندسة الحاسبات

خوارزمية جينية متداخلة لأمثلية شبكة الجوال الارتجالية بجودة فازية

إعداد

مهندس/ احمد زكريا نور محمد طلحه

رسالة مقدمة للأكاديمية العربية للعلوم والتكنولوجيا والنقل البحرى لإستكمال متطلبات نيل درجة

الماجستير
فى
هندسة الحاسبات

تحت إشراف

الاستاذ الدكتور/ أحمد فهمى أمين
أستاذ دكتور هندسة الحاسبات
الاكاديمية العربية للعلوم و التكنولوجيا و النقل البحري (فرع القاهره)

الاستاذ الدكتور/ عمرو أنور بدر
أستاذ دكتور كلية الحاسبات و الذكاء الاصطناعى
جامعة القاهرة

2020