EVALUATION OF HEURISTIC ALGORITHMS FOR TRAFFIC SIGNAL OPTIMIZATION ALONG A CORRIDOR

by

Morteza Ahmadpur

B.S., Civil Engineering, Tabriz Azad University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Civil Engineering

Boğaziçi University

2016

EVALUATION OF HEURISTIC ALGORITHMS FOR TRAFFIC SIGNAL OPTIMIZATION ALONG A CORRIDOR

APPROVED BY:

Assist. Prof. Ilgın Gökaşar (Thesis Supervisor)	……..
Assoc. Prof. Nilüfer Özyurt Zihnioğlu	……..
Sinan Işık, Ph.D.	……..

DATE OF APPROVAL: 12.01.2016

DEDICATION

For Sanam

ACKNOWLEDGEMENTS

My sincere thanks also go to Assist. Prof. Ilgın Gökaşar who provided me an opportunity to join her research team, and who gave access to BOUN Intelligent Transportation Laboratory and research facilities. Without her precious support, it would not be possible to conduct this research.

I would like to thank my thesis committee: Assoc. Prof. Nilüfer Özyurt Zihnioğlu and Dr. Sinan Işık, for their insightful comments and encouragement.

I would like to thank my wife Sanam. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past year of my life has been built. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love.

I would like to thank my family: my parents Ali and Ruqeye, for giving birth to me at the first place and supporting me spiritually throughout my life.

A special thanks to my wife's family. Words cannot express how grateful I am to my mother-in-law, father-in-law, sister-in-law and brother-in-law. I thank for all of the sacrifices that you've made on my Behalf.

Your prayer for me was what sustained me thus far also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

ABSTRACT

EVALUATION OF HEURISTIC ALGORITHMS FOR TRAFFIC SIGNAL OPTIMIZATION ALONG A CORRIDOR

Traffic signals are one of the most important parts of road networks. They play a significant role in decreasing congestion and providing better service for the society in urban areas. The aim of this thesis is to determine the best traffic signal timing plan using heuristic algorithms for a corridor consisting of several intersections with a microscopic traffic simulator based on a car following theory, namely Intelligent Driver Model. The measure of effectiveness used in this study is the duration of stopped vehicles. All available signal timing plans for the corridor are evaluated and the results illustrate that determining the best traffic timing plan can effectively reduce the congestion in corridors. Based on the evaluation of the performances of heuristic algorithms (Simulated Annealing, Hill Climbing, Genetic Algorithm), Genetic Algorithm provides the global optima for all scenarios for this specific corridor. In this study, it is also determined that Hill Climbing Algorithm works 28 times faster than Genetic Algorithm, but it fails to reach global optima in the predetermined time duration. Genetic Algorithm's search process takes much more time than the Hill Climbing and Simulated Annealing; however, its performance for determining global optima in the given number of generations makes it the best option in this study.

OZET

BİR KORİDOR BOYUNCA TRAFİK SİNYAL OPTİMİZASYONU İÇİN SEZGİSEL ALGORİTMALARIN DEĞERLENDİRİLMESİ

Trafik işaretleri yol ağının en önemli parçalarından biridir. Trafik işaretleri kentsel alanlarda, tıkanıklık azaltılmasında ve topluma daha iyi hizmet verilmesinde önemli bir rol oynamaktadır. Bu tezin amacı, çeşitli kavşaklardan oluşan bir koridor için, Akıllı Sürücü Modeli araba-takip teorisine dayanan bir mikroskobik trafik benzetimi kullanılarak sezgisel algoritmalar ile en iyi trafik sinyal zamanlama planını belirlemektir. Bu çalışmada kullanılan başarım ölçütü duran araç süresidir. Koridor için tüm mevcut trafik sinyal zamanlama planları değerlendirildi. En iyi trafik sinyal zamanlama planını kullanmanın koridorda bulunan trafik sıkışıklığını verimli bir şekilde azaltabildiği gösterilmiştir. Sezgisel algoritmaların (Benzetilmis Tavlama (Simulated Annealing), Tepe Tırmanan (Hill Climbing) ve Genetik (Genetic)) başarım ölçütlerinin değerlendirilmesine göre Genetik Algoritma tüm senaryolar için en iyi zamanlama planını tüm kavşaklar için sağladı. Bu çalışmada, Tepe Tırmanma Algoritmasının çalışma hızı Genetik Algoritmasından 28 kat daha fazla olduğu, fakat genel en iyiye ulaşmakta başarısız olduğu belirlenmiştir. Genetik Algoritmanın arama süreci, Benzetilmiş Tavlama ve Tepe Tırmanan Algoritmalarına göre daha uzundur, ancak Genetik Algoritmanın belirtilen zaman surecinde genel uygun değeri bulma performansı en iyidir.

TABLE OF CONTENTS

DEDICATION……iii

ACKNOWLEDGEMENTS……iv

ABSTRACT…… v

OZET……... vi

TABLE OF CONTENTS……... vii

LIST OF FIGURES…….. x

LIST OF TABLES…… xiii

LIST OF SYMBOLS……xiv

LIST OF ACRONYMS/ABBREVIATIONS……xvi

1. INTRODUCTION…… 1

1.1. General 1

1.2. Problem Description 2

1.3. The Aim and Objectives of the Thesis 2

1.4. Outline of the Thesis 3

2. LITERATURE REVIEW…… 4

3. THEORY…….. 7

3.1. Traffic Flow Theory 7

3.2. Car Following Theory 11

3.2.1. Intelligent Driver Model……... 12

3.3. Traffic Simulation 14

3.3.1. Developed Microscopic Simulator……... 16

3.3.1.1. Segments of the Simulator…….……17

3.3.1.2. Measure of Effectiveness……..…….18

3.4. Traffic Signals 19

3.5. Heuristic Algorithms 20

3.5.1. Definition of Chromosome……... 21

3.5.2. Solution (Search) Space……21

3.5.3. Simulated Annealing (SA)……22

3.5.4. Genetic Algorithm (GA)…….. 25

3.5.5. Hill Climbing (HC)…….. 28

4. METHODOLOGY……. 29

4.1. Definition of the Parameters..... 29

4.1.1. Parameters of Car Following Theory and Traffic Flow Model…… 29

4.1.2. Defining the Solution or Chromosome and the Solution……

(Search) Space…….…... 30

4.1.3. Geometry of Network and Characteristics of Traffic Signals…….. 32

4.2. Development of Microscopic Simulation Model 34

4.3. Heuristics 36

4.3.1. Genetic Algorithm (GA)..... 37

4.3.2. Simulated Annealing (SA)..... 40

4.3.3. Hill Climbing (HC)..... 42

4.4. Method of Comparison of the Heuristic Algorithms 46

4.5. Results of the Simulation 46

4.5.1. Genetic Algorithm (GA)..... 50

4.5.1.1. Mutation Probability (MP) 50

4.5.1.2. Number of Parent Chromosomes (NPC).....53

4.5.2. Simulated Annealing (SA)..... 56

4.5.2.1. Alpha Value & Initial Temperature.....56

4.5.2.2 . Number of Mutation Points (NMP).....61

4.5.3. Hill Climbing (HC)..... 62

4.6. Results of Comparison of the Performances of the Heuristic Algorithms 63

5. CONCLUSIONS..... 68

REFERENCES..... 72

APPENDIX: SMALL PORTION OF GENERATED CODES IN MATLAB..... 80

LIST OF FIGURES

Figure 3.1. Evolution of fundamental diagram (Wageningen-Kessels et al., 2014). ...8

Figure 3.2. Genealogy of traffic flow models (Wageningen-Kessels et al., 2014). ...10

Figure 3.3. Two vehicles in a car following process. ...11

Figure 3.4. Details of two following vehicles. ...14

Figure 3.5. Behaviour of vehicles when the street is full of vehicles. ...18

Figure 3.6. Visualization of the simulation. ...18

Figure 3.7. Cycle length and splits of all traffic signals. ...20

Figure 3.8. Shape and structure of chromosomes (string). ...21

Figure 3.9. Solution (Search) space. ...22

Figure 3.10. The basic algorithm of SA (Cerny, 1985). ...24

Figure 4.1. Illustration of chromosome. ...31

Figure 4.2. Flowchart of defining the solution space. ...32

Figure 4.3. Geometry of the network. ...33

Figure 4.4. Phases and ring-barrier diagram of all intersections. ...34

Figure 4.5. Details of the operator which prohibit vehicles to inter intersection. ...35

Figure 4.6. Calculating the measure of effectiveness by using the simulator. ...36

Figure 4.7. Genetic Algorithm's Crossover operation. ...38

Figure 4.8. Flow chart of GA's search process. ...39

Figure 4.9. Mutation process of a SA sample. 40

Figure 4.10. Flowchart of the SA. 41

Figure 4.11. HC's first part of search process. 44

Figure 4.12. HC's second part of search process. 44

Figure 4.13. Flowchart of HC's algorithm. 45

Figure 4.14. Measure of effectiveness (MOE) of all chromosomes. 48

Figure 4.15. Continues shape of MOE. 48

Figure 4.16. Sorted value of measures of effectiveness. 49

Figure 4.17. Best and worst traffic signal timing plans. 49

Figure 4.18. Results of the GA's process for one sample run. 50

Figure 4.19. Average minimum objective values. 51

Figure 4.20. Average number of generations. 52

Figure 4.21. Average CT of each generation. 53

Figure 4.22. Unsuccessful attempts. 53

Figure 4.23. Average minimum value of objective values of NPC. 54

Figure 4.24. Unsuccessful attempts. 54

Figure 4.25. Average number of generations of every PC value. 55

Figure 4.26. CT of the GA with different NPC values. 55

Figure 4.28. Average numbers of generations of SA. 57

Figure 4.28. Average number of generations for initial temperatures with……..
α equals 0.98. 58
Figure 4.29. A sample Search process of SA with initial temperature equals……..
100 and α equals 0.98.....58
Figure 4.30. Average number of generations with α equal to 0.75. 59
Figure 4.31. A sample Search process of SA with initial temperature equals……..
100 and α equals 0.75.....59
Figure 4.32. Average number of generations of different initial solutions. 60
Figure 4.33. Average number of generations of initial temperatures.....60
Figure 4.34. Unsuccessful attempts of SA with different initial temperatures. 61
Figure 4.35. Average number of generations of different NMP values. 62
Figure 4.36. Unsuccessful attempts of SA with different NMP values. 62
Figure 4.37. Sample illustration of four HC's search processes. 63
Figure 4.38. Objective values of 1000 generations of HC. 63
Figure 4.39. Minimum objective values of HC. 64
Figure 4.40. Minimum objective values of SA. 65
Figure 4.41. GA's minimum objective value of 2000 runs. 66

LIST OF TABLES

Table 3.1. Strengths of Simulation Models (May, 1990). 15

Table 4.1. Parameters of IDM. 30

Table 4.2. Permutations of 10, 20, 30, 40 and 50 seconds time durations. 31

Table 4.3. Descriptive statistics of solution space. 47

Table 4.4. Descriptive statistics of 2000 runs of HC. 64

Table 4.5. Descriptive statistics of 2000 runs of SA. 65

Table 4.6. Descriptive statistics of 2000 runs of GA. 66

Table 4.7. Descriptive statistics of 2000 runs of HC,SA and GA. 67

LIST OF SYMBOLS

a	Acceleration in each day traffic
α	A constant
b	Comfortable braking deceleration in each day traffic
dv/dt	Calculated acceleration rate of IDM
δ	Acceleration exponent, delta
Δt	Time interval
ΔJ	Difference between initial and neighbor solutions objective value
Δv	Speed difference or relative speed of two vehicles
h	Hour
k	Density
L	Vehicle length
m	Meter
ml	Mile
q	Flow
s	Bumper-to-bumper distance s to the leading vehicle
s0	Minimum bumper-to-bumper distance to the front vehicle
Si	Initial solution
Sj	Neighbor solution
t	Current time
T	Current temperature of SA
Th	Desired safety time headway when following other vehicles
Tf	Final temperature
Ti	Initial temperature
Tf	Final temperature
Si	Initial solution
Sj	Neighbor solution
v	Speed of vehicle
v0	Desired speed when driving on a free road
Vs	Space mean speed

Vt	Time mean speed
x	Position of one individual vehicle
X	A counter
Y	A counter

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial intelligence
CCTV	Close circuit television
CF	Car following
CPU	Central processing unit
CT	Computational time
EBT	East bound through
FD	Fundamental diagram
GA	Genetic Algorithm
GM	General Motors
IDM	Intelligent driver model
km	kilometer
MOE	Measure of effectiveness
MP	Mutation probability
NBT	North bound through
NDP	Network design problem
NMP	Number of mutation points
NBT	North bound through
NPC	Number of parent chromosomes
PC	Parent chromosome
rnd	Random number
SA	Simulated Annealing
SBT	South bound through
sec	Second
veh	Vehicle
WBT	West bound through

1. INTRODUCTION

1.1. General

Traffic congestion is one of the biggest challenges for citizens in urban areas and it is obvious that they occur every day. This phenomenon reduces life standards in the society and wastes lots of capital in countries. Congestion increases the air pollution in urban areas and affects the health of the people (Traffic signal timing manual, 2008). Therefore, by reducing the congestion in urban areas, life standards in these regions can increase.

Traffic signals play an important role in controlling the traffic flow in entire network. A timing plan is needed to determine the green times, red times and yellow times of traffic signals in arterial networks. Determining this timing plan is a difficult task because every traffic signal can affect the performance of entire network. Therefore, determining the best and suitable timing plan of traffic signals in the network is essential for reducing the congestion.

Nowadays, there are several kinds of traffic simulator programs and software which help authorities to simulate the traffic flow in arterial networks. These powerful tools can mimic the real time traffic flow in high detail in virtual world. Hence by using these soft wares, it is possible to determine the best timing plan of traffic signals.

There are three kinds of traffic flow models and each one has its own characteristics. Microscopic traffic flow models simulate single driver units and calculate every vehicle's speed, acceleration and position. The microscopic simulation is selected in this thesis because these traffic models can analyse the traffic models in high detail and determine the effects of each vehicles characteristics on the performance of entire network.

Genetic Algorithm (GA), Simulated Annealing (SA) and Hill Climbing (HC) are three prominent search algorithms. These heuristic algorithms are based on different principles and have different performances for determining suitable solutions from solution

space. The GA is an evolutionary algorithm, SA is based on the behaviour of materials and HC is based on simple principles of decision making.

Heuristics are useful tools for determining solutions from solution space. In this study, a solution space is created that is consisted of chromosomes, and every chromosome is representative of the red times and green times of traffic signals. By using heuristic algorithms, it is possible to determine the best traffic signal timing plan for intersections. The search performance of heuristic algorithms can vary in different optimization problems. Performance of heuristic algorithms should be evaluated and compared in terms of best heuristic method for determining the best combination of traffic lights timing plan.

1.2. Problem Description

Traffic signals control the traffic flow in arterial network. Every traffic signal needs a timing plan. Finding the suitable timing plan can reduce the congestion in entire network. Determining the suitable timing plan for an isolated intersection is not a challenging task but, opting suitable timing plan for several numbers of intersections along a corridor can be very cumbersome.

1.3. The Aim and Objectives of the Thesis

The main aim of this study is determining the best traffic signal timing plan for three intersections along a corridor by using selected heuristic algorithms with a microscopic traffic simulator. The heuristic with the best performance is selected among three selected ones after carrying out several tests.

To achieve this goal, the following objectives are determined:

Firstly, Intelligent Driver Model (IDM) selected as the base car following theory in developing a microscopic traffic model. Then, a microscopic traffic simulator is developed through a code written in MATLAB programming environment. After creation of solution space that contains all available traffic signal timing plans, the measure of effectiveness of microscopic simulator is defined. Selected three heuristic algorithms selected for the study’s purpose and each ones parameters and performance evaluated. Then, three heuristic algorithms are compared and the best is determined.

1.4. Outline of the Thesis

The remaining part of the thesis is organized in four Chapters. Chapter 2 demonstrates an overall review of the methods used for optimizing traffic signals. Chapter 3 states the basic theory of microscopic traffic flow simulator and heuristic algorithms. In Chapter 4, the proposed methodology for developing microscopic traffic simulator, the evaluation of heuristic algorithms and the results of several tests and simulations are reported. Finally, in Chapter 5, the conclusions are provided.

2. LITERATURE REVIEW

In arterial networks, traffic signals control traffic flow to reduce congestion, prevent accidents, improve safety and provide special scenarios such as minimizing delays, prioritizing public transport, and improving environmental pollutions (IHT, 1997). For several years the process of determining best traffic timing plan developed and continuously improved. Previous methods such as (Webster, 1958) only considered an isolated intersection. Later, pre-time strategies were developed that optimized several signalized intersections by using historical traffic data (e.g., TRANSYT: Robertson, 1969). Real time traffic flow data has also been used for optimization in some cities. These methods commonly named demand-responsive strategies (e.g., SCOOT: Hunt et al., 1981). The focus of this research is on pre-time traffic signal timing plan.

Genetic Algorithm (GA) and other heuristic algorithms have been used widely in the transportation field. For problems such as generating zoning (Balling et al., 2004), transit network design (Bielli et al., 2002 ; Chakroborty, 2003), traffic incident detection (Srinivasan et al., 2000), transit scheduling (Chakroborty et al., 1998), traffic parameter estimation (Sharma et al., 2004), dynamic traffic management (Lo et al., 2001) and (Abu-Lebdeh and Benekohal, 2003).

In signal timing design, GA has been used to optimize cycle time, green time, offsets, and stage sequences (Foy et al., 1992; Park et al., 2000; Park and Yun, 2005). Foy et al. (1992) determined that GA, by considering how traffic moves among multiple intersections, can determine a logical, near-optimal timing configuration. In another study, Park et al. (2000) founded that TRANSYT-7F tended to select longer cycle lengths than the GA to reduce delay. Park and Yun (2005) proposed a method based on CORSIM (microscopic traffic simulator) and a GA for stochastic optimization for coordinated actuated traffic signal systems. Their proposed method outperforms the existing timing plan and the Synchro-optimized traffic signal timing method. Lee (1998) performed a comparison of GA and Simulated Annealing (SA) with iterative and local search

algorithms and showed that different algorithms perform better for different network supply and demand scenarios.

Park and Schneeberger (2002) tried to optimize traffic signals time plan with GA using VISSIM microscopic simulator but, they did not take all possible time plans into consideration and simulation performed only for few time plans. In another work, He et al. (2011) proposed a useful heuristic algorithm for calculating the signal timing plan for isolated intersections. The proposed heuristic algorithm could reduce average bus delay in congested conditions by about 50%, especially with a high frequency of conflicting priority requests. In another study Lee et al. (2005) used GA for determining best traffic signal timing plan. They utilized a commercial microscopic traffic simulator namely, PARAMICS and performed lots of iterations to optimize traffic signal's schedule. The results show that the developed adaptive system consists of a GA, a database manager and an internal traffic simulator for fitness evaluation provided efficient demand-responsive signal control in all the testing scenarios. Taale and Van Zuylen (2003) obtained good results by applying a GA to optimize green times within a network design problem (NDP) on small artificial networks.

According to all above mentioned literature, it is obvious that GA has been used widely in transportation because GA has a good performance for determining suitable solutions. However, the performance of more simple algorithms like SA and HC for determining suitable solutions for traffic signal problem is not clear. Gentile et al. (2009) explained that determining an optimal solution on not very small and simple arterial network may take long time. To do this, a genetic-like algorithm applied, in which new solutions generation follows heuristic conceptions. Thompson and Bilbro (2000) compared the GA and SA for the problem of optimizing the topological design of a network and determined that the average GA solution cost less than the average SA's solution. Therefore the performance of GA was better than SA. Kenekayoro (2012) compared HC and SA's performance in Course timetabling problem. He realised that utilizing SA or HC is preferable because of ease of execution. Then, he concluded that poor performance of SA shows how problem specific heuristic algorithms are.

In another study, Ross and Corne (1995) compared GA, SA, and Stochastic HC on timetabling problem. Comparisons were done in terms of final solution quality, and number of distinct solutions obtained. Their main conclusions were that HC and SA has better performance when final solution's quality is concerned. When considering number of distinct solution numbers, large number of usefully distinct solutions was created by GA and this phenomenon indicated that GA has weak performance.

Unfortunately there is not enough literature about comparing GA with SA and HC, especially in transportation field. Therefore in this thesis, performances of these heuristic algorithms are compared to selecting the suitable heuristic for determining best traffic signal timing plan.

3. THEORY

`

This part of the thesis consists of five chapters. In the first chapter, traffic flow theory is described. Then, car following theories are explained. In the third chapter, importance and theory of traffic simulation is discussed. In the fourth and fifth chapters, traffic signals and heuristic algorithms are explained, respectively.

3.1. Traffic Flow Theory

Knowledge of fundamental traffic flow characteristics and related analytical techniques are necessary for planning, design and operation of transportation systems and traffic engineering. Traffic flow characteristics include time headways, flow, speed and density (May, 1990).

The elapsed time between the arrivals of two following vehicles is time headway. Flow (q) is the rate at which vehicles pass a fixed point of a road (vehicles per hour). Speed is defined as distance covered per unit time. Its unit is kilometers per hour (km/h) or miles per hour (ml/h). There are two kinds of speeds: time mean speed and space mean speed. Time mean speed (Vt) is the average speeds of vehicles passing a particular point. Space mean speed (Vs) is the average speeds over a length of roadway.

Density (k) is the longitudinal space occupied by individual vehicles in the traffic flow. Density is defined as vehicles per kilometer (veh/km) or vehicles per miles (veh/ml). In practice, density is measured from one aerial photo (May, 1990).

The Equation 3.1 defines the fundamental relationship between density, space mean speed and flow of traffic flow stream. According to this fundamental equation, flow is calculated by multiplying the density with space mean speed.

$$q = k.\mathrm{Vs} \tag{3.1}$$

By using the Equation 3.1, the fundamental diagram of density-flow can be derived.

Studying traffic flow is difficult because driver behaviours are not the same at arterial network and it is very difficult to predict the behaviour of drivers. In another hand, driver behaviours are in a consistent range and it is possible to simulate the driver behaviour by using mathematical formulas like fundamental diagram.

The Fundamental Diagram (FD) describes the relationship between density (k) and flow of vehicles (q) in a stream of vehicles. Greenshields (1935) presented the FD and relates the distance between two vehicles (spacing) to their speed. The evolution of FD is described in Figure 3.1. Greenshields purposed the first FD which has a parabolic shape. After that, Drake (1967) changed the shape of FD. In 1990, Smulders suggests a Parabolic-linear shape for FD and in final attempt; Deganzo modified the shape of FD as illustrated in Figure 3.1.

Wageningen-Kessels et al. (2014) described the Genealogy of traffic flow models in their study. The Genealogy is showed in Figure 3.2. It is obvious that there are three kinds of traffic flow models: Microscopic, Mesoscopic and Macroscopic models. It can be seen that there is two kinds of microscopic models: car following theories and cellular automata.

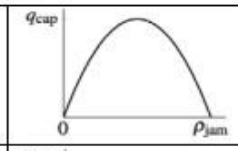
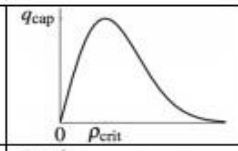
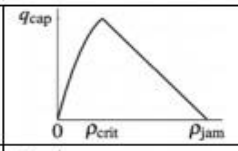
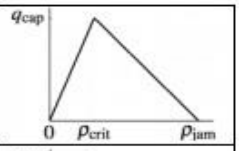
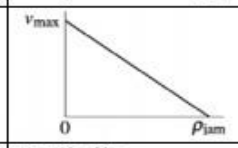
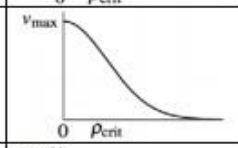
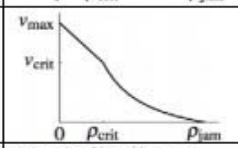
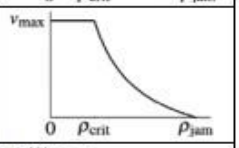
Density–flow				
Density–speed				
Shape	Parabolic	Bell	Parabolic–linear	Bilinear
Author	Greenshields	Drake	Smulders	Daganzo
Year	1934	1967	1990	1994

Figure 3.1. Evolution of fundamental diagram (Wageningen-Kessels et al., 2014).

Mesoscopic models maintain individual vehicle representation but with a more aggregate representation of traffic dynamics. Mesoscopic models describe vehicle behavior in aggregate terms such as in probability distributions (Wageningen-Kessels et al., 2014).

Macroscopic models use less detailed models and represent the traffic as a compressible fluid with the main properties flow, density and speed. Individual vehicles are not modeled in these kinds of models. The main advantage of macroscopic traffic models over microscopic models is the significantly lower computational costs due to lower complexity. These models describe traffic flow like a continuum flow and are often compared to, or derived in analogy with, continuum models for fluids (Wageningen-Kessels et al., 2014).

Microscopic traffic models are based on the assumption that drivers set their speed and acceleration rate according to the leading vehicle. According to Park and Schneeberger (2003), microscopic simulation models are widely used because simulation is safer, less expensive, and faster than field implementation and testing. Most microscopic models are car following models (Wageningen-Kessels et al., 2014). As mentioned before there is two kinds of microscopic models: car following models and cellular automate. Car following models are considered in this study.

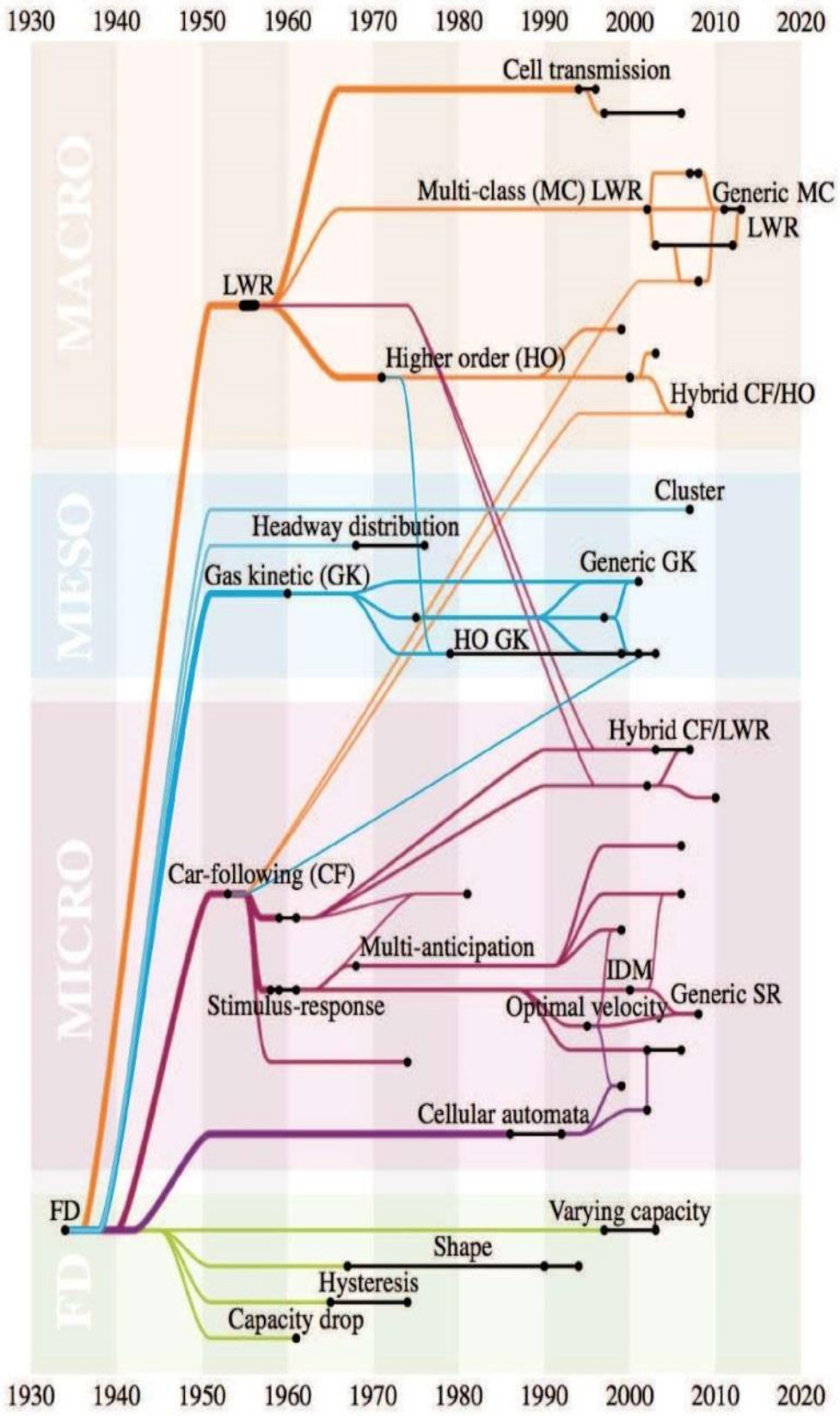


Figure 3.2. Genealogy of traffic flow models (Wageningen-Kessels et al., 2014).

3.2. Car Following Theory

May (1990) described that car following theory is a description of how one vehicle follows another vehicle in a traffic flow. Figure 3.3 illustrates two following vehicles. Reuschel and Pipes were the pioneers in the development of car following theory in early 1950s. Three fundamental efforts were committed in the late 1950s and continued to the mid-1960s: Kometani and Sasaki (1959) in Japan, Forbes (1958,1963 and 1968) at the institute for research and Michigan University and a group of researchers associated with General Motors (GM).

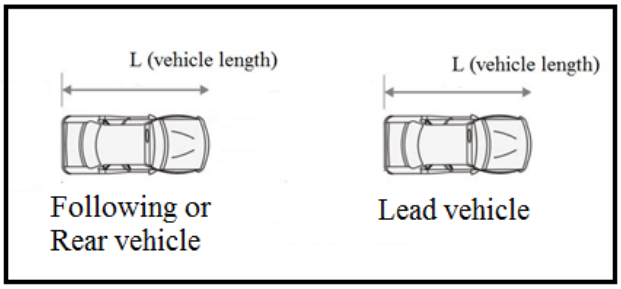


Figure 3.3. Two vehicles in a car following process.

In another effort, Forbes (1963) proposed a car following theory. He approaches car following by consideration of reaction time for drivers to perform brake. According to his work, time gap between two vehicles should always be equal or greater than reaction time. Therefore, the minimum time headway of two following vehicles should be equal to the reaction time of driver of following vehicle.

Another car following theory is the General Motors Theory (May 1990), scientists, engineers and analyser of this company, performed lots of field measurements and efforts and at last purposed five car following theories. The prominent one is the fifth GM's car following theory. All of these car following theories have the same principle, which explain that response is equal to function of sensitivity and stimuli. The response was represented by acceleration and a stimulus was represented by the relative velocity of lead and rear vehicles.

Gazis et al. (1961) determined a surprisingly relationship between the third General Motors car following theory and Greenberg's traffic flow model. After lots of field

measurements and mathematical proofs, they determined a direct relationship between the GM's third model and Greenberg's macroscopic model. These results were a big discovery at that time and were a strong bridge between microscopic and macroscopic modeling.

Wageningen-Kessels et al. (2014) defined that car following theory have six major branches: Safe-distance models, Stimulus–response models, more recent Stimulus–response models, Action point models, Multi-class and Multi-anticipation car following models and Cellular-automata models. According to this article, IDM is a Multi-class and multi-anticipation car following model.

3.2.1. Intelligent Driver Model

Intelligent Driver Model (IDM) is a time-continuous car following model. It was developed by Treiber, and Helbing (2000) to improve upon results provided with other "intelligent" driver models such as Gipps's Model. Treiber (2013) described that the time-continuous Intelligent Driver Model (IDM) is probably the simplest complete and accident-free microscopic model that can produce realistic of traffic flow. According to this book, by comparing the virtual detector data with the real data, they determined almost quantitative agreement of the flow-density, speed-density and speed-flow diagrams.

Sultani et al, (2010) explained that there are mainly three types of traffic on roads, i.e., free flow, synchronized and congested traffic and IDM is a novel and well established car following theory that describes behaviors of driver for all three types of traffic.

Treiber et al. (2000) in their article described that IDM is simple, has only a few intuitive parameters with realistic values, reproduces a realistic collective dynamics, and also leads to plausible microscopic acceleration and deceleration behavior of single drivers.

IDM is a microscopic traffic flow model, i.e., every vehicle act like a particle in the system. This model defines the speed and position of every vehicle in entire network. In this model, the decision of any driver to accelerate or decelerate depends only on his or her own speed, position and speed of the leading vehicle. The structure of the IDM can be described as follows:

- The influencing factors (model input). Are the own speed of the vehicle (V), the bumper-to-bumper distance to the leading vehicle (S), and the relative speed (speed difference, ΔV) of the two vehicles.
- The model output. Is the acceleration (dv/dt) chosen by the driver depending on the situation of driver.
- The model parameters. Describe the driving style, i.e., whether the simulated driver drives slow or fast, careful or care free.

The IDM's formulas are described as follows (Treiber, and Helbing, 2000):

$$\frac{dv}{dt} = a\left(1 - \left(\frac{V}{v0}\right)^{\delta} - \left(\frac{S*(V,\Delta V)}{S}\right)^{2}\right) \tag{3.2}$$

where

$$S*(V,\Delta V) = s0 + \max\left[0, \left(V * Th + \frac{V.\Delta V}{2\sqrt{ab}}\right)\right] \tag{3.3}$$

By using Equations (3.2) and (3.3), (Δt) seconds after current time, speed, acceleration and position of each vehicle on the road network are evaluated.

New speed: $v(t+\Delta t)=v(t) + (dv/dt)\ .\Delta t,$ (3.4)

New position: $x\ (t+\Delta t)=x(t)+v(t)\Delta t+1/2\ (dv/dt).\ (\Delta t)^2,$ (3.5)

New gap: $s\ (t+\Delta t) =x(t+\Delta t)- x(t+\Delta t)- L.$ (3.6)

In the above mentioned formulas, (t) is the current time, (Δt) is the interval time (in this study (Δt) is equal to one second), (x) is the position of one individual vehicle, (s) is the gap between two consecutive vehicles and (L) is the length of the leading vehicle.

Desired velocity (v) is the maximum speed of vehicle which tries to reach it. Headway (Th) is the distance or time between vehicles in a transportation system, it is measured as the distance from bumper of one vehicle to bumper of the following vehicle, expressed as time it will take for the following vehicle to cover that distance. Maximum acceleration rate (a) is the maximum acceleration rate which a vehicle can reach in the simulation.

Desired deceleration parameter (b) defines the deceleration which depends on the driver's reaction time and braking system of the vehicle. According to National Cooperative Highway Research Program (1997) a deceleration rate of 3.4 m/s2 is used to determine deceleration rate.

The acceleration exponent (δ) determines that how the acceleration decreases when getting close to the desired velocity. The most realistic and practical behavior is expected in between the two limiting cases of exponential acceleration (for δ=1) and constant acceleration (for δ=∞).

Minimum distance (s0) or jam distance is the minimum desired net distance between the rear bumpers of two following cars when they are in a jammed traffic. It can vary between 6 and 10 meters and depends on the behavior of the vehicle's driver. If the driver is aggressive, the (s0) will be shorter, if the driver is calm and obedient, the distance will be longer. Figure 3.4 shows details of two following cars.

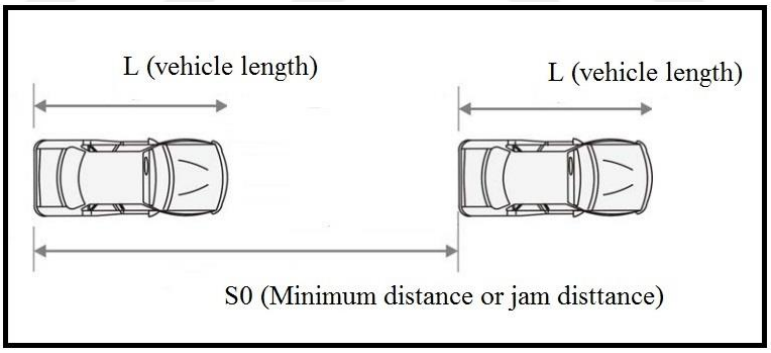


Figure 3.4. Details of two following vehicles.

3.3. Traffic Simulation

The definition of simulation is a numeric technique for conducting experiments on a digital computer, which may include stochastic characteristics, be microscopic or macroscopic in nature, and involve mathematical models that describe the behaviour of a transportation system over extended periods of real time (May, 1990). Traffic simulation is the mathematical simulation of traffic in virtual environment. Simulation models may also be classified according to the level of detail with which they represent the system to be studied: Microscopic, Mesoscopic and Macroscopic (FHWA 2015). Pursula (1999) pointed

out that the use of computer simulation started when D.L. Gerlough published his dissertation in 1955. From those times, computer simulation has become a widely used tool in transportation engineering with a variety of applications from scientific research to planning, training and demonstration.

According to May (1990), earliest computer simulation work on highway transportation occurred in the 1950s. Intersection simulation was undertaken by the road research laboratory in the United Kingdom in 1951. The first simulation work in the United States was published in 1953 and reported on intersection and freeway models developed at the University of California at Los Angeles. The development of simulation models grew rapidly during the 1960s and 1970s and bibliographies were published devoted exclusively to computer simulation models developed for the highway systems. Strengths of simulation models are described in the Table 3.2.

Table 3.1. Strengths of Simulation Models (May, 1990).

Simulation model strengths	
1	Other analytical approaches to the problem may not be appropriate
2	Can experiment off-line
3	Can experiment with new situations and scenarios
4	Can yield insight into which variable effect the system mostly
5	System can be studied in real time, compressed time or expanded time
6	Potentially unsafe experiments can be done on computers
7	Can replicate several simulations by changing variables and compare the results
8	The several: what if…what happen? Questions can be answered
9	Can handle interacting queuing processes
10	Can transfer unserved queued traffic from one time period to next
11	Demand can be varied over time and space
12	Unusual arrival and service patterns can be modeled

According to Traffic Signal Timing Manual (2008) microscopic simulation models can be an evaluation tool for signal timing plans. Most microscopic simulators do not have any built-in signal timing optimization algorithms and instead depend on other programs to

provide a fully specified signal timing plan. However, these kinds of models can evaluate the effects of interactions between intersections.

There are several numbers of Microscopic simulators. Some of prominent microscopic simulators are described as follows:

- TRANSIMS. A well-known microscopic simulator that uses cellular automata principles to simulate traffic conditions and analyze the interaction between individual vehicles, producing emergent behaviors such as traffic congestion. The TRANSIMS approach is based on the new paradigm of modeling individual travelers and their multi-modal transportation based on synthetic populations and their activities (Maciejewski, 2010).
- SUMO. This simulator developed by German Aerospace Center. SUMO is a free and open source microscopic simulator which is available since 2001. SUMO can simulate intermodal traffic systems including road vehicles, public transport and pedestrians. It uses an extension of Gipps safe distance car following model (Maciejewski, 2010).
- VISSIM. VISSIM is a commercial system for microscopic traffic simulation. For car following modeling purposes, it uses a driver behaviour model that takes into consideration an influence of driver's perception on velocity control(Maciejewski, 2010).

3.3.1. Developed Microscopic Simulator

A microscopic traffic simulator was developed and created by using a fourth-generation programming language, Matlab. The simulator can simulate the traffic flow in low traffic conditions and evaluates traffic data of every vehicle in the system. Intelligent Driver Model is utilised in the microscopic simulators structure because of simplicity and effectiveness of this method (Treiber and Kesting, 2013). After developing the simulator and simulating the traffic flow in the corridor, several traffic signal timing plans assigned to the traffic signals and different results evaluated.

3.3.1.1. Segments of the Simulator. The microscopic simulator consists of three main Segments as follows: parameter definition, acceleration calculation and simulation visualization.

In the first segment, the parameters of IDM, number of lanes, length and type of vehicles, position of lights and sensors, green time, red time and yellow time duration of each light, cycle lengths, link length of road network, position of intersections, position of stop lines for vehicles, vehicle arrival rate or traffic flow, initial speed and position of initial cars, duration of simulation are determined.

Because the hourly flow rate is under 800 vehicle per hour, according to May(1990) all the vehicles are traveled independently in the links, therefore time headways is considered as random time headways and headway distribution is a random time headway distribution. Therefore, the Poisson distribution is used to create vehicles in the road network.

The acceleration calculator of simulator evaluates position, speed and acceleration of each individual vehicle. Vehicles by obeying IDM's principles, try to keep the safe distance from leading vehicle and by taking the traffic signal rules into consideration, resume their move in the road network. They are forced to stop in the stop line of intersections if traffic lights indicate yellow and red and are allowed to enter the intersections when the light is green. Vehicles move through in the intersections.

In the road networks, the traffic moving through intersections may be blocked at the upstream signal by the queue spillback from the downstream signal. This queue reduces the capacity of movement at the upstream intersection significantly. Therefore, all three intersections have box junction which generally prohibiting entry to the intersection unless the exit is clear. According to Figure 3.5, vehicles beside intersection number one cannot enter in the road number one even if the traffic light in intersection one is green.

By using powerful functions of Matlab program, it is possible to create animations of traffic simulator in this environment. Each car depicted as a rectangle. Traffic signals are illustrated as circles in the intersections which the color of the circle indicates the lights condition. Traffic lights indicate red, green or yellow lights. Dashed lines indicate the links

of network and squares exhibits intersections stop line for vehicles. A sample of visualization is shown in Figure 3.6. Horizontal axe indicates the distance from major road in meters. Vertical axe is length of main road. Arrows indicate direction of vehicles move. Blue cars move from east to west, green ones move in opposite directions, purple cars move from south to north and red cars move from north to south.

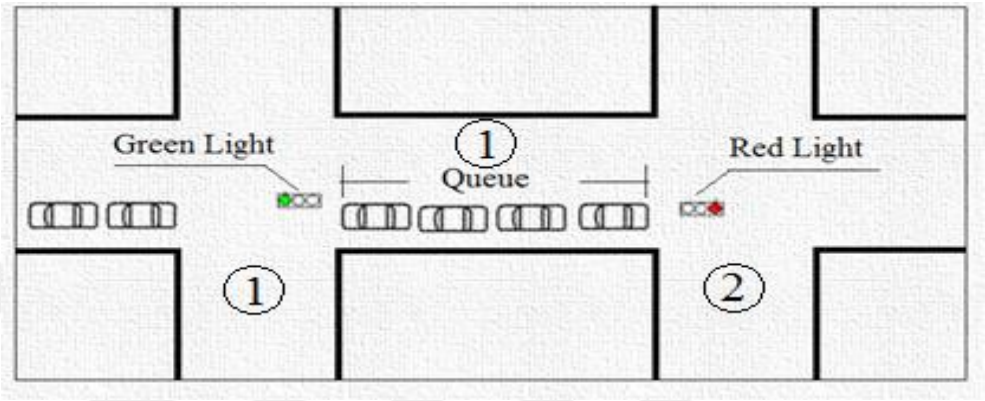


Figure 3.5. Behaviour of vehicles when the street is full of vehicles.

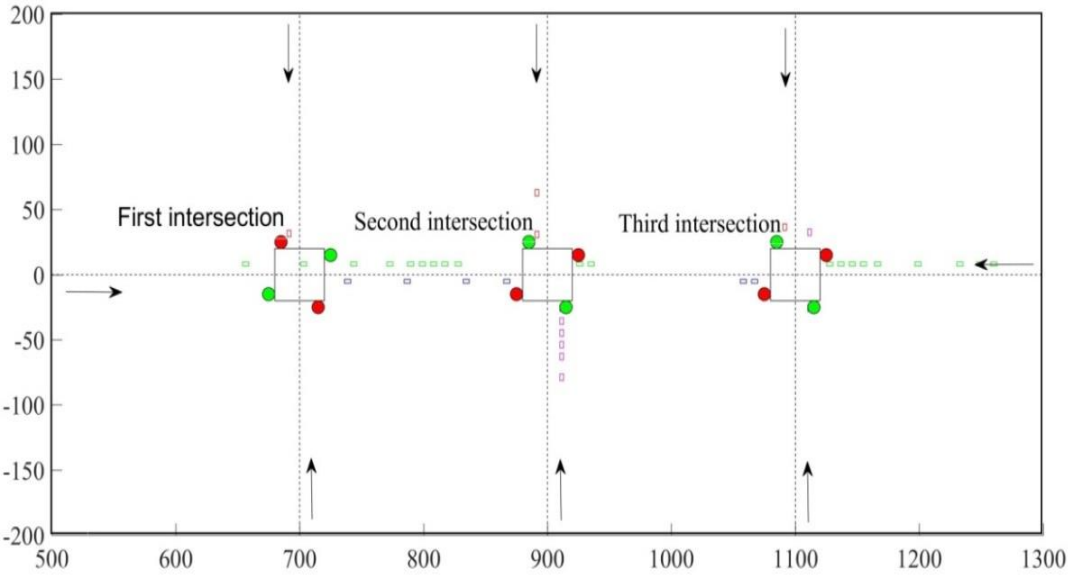


Figure 3.6. Visualization of the simulation.

3.3.1.2. Measure of Effectiveness. Measure of effectiveness (MOE) or objective value is a measurement which indicates the performance of the model or simulation in different situations. According to Traffic Signal Timing Manual (2008), measure of effectiveness is a quantitative parameter that shows the performance of a transportation facility or service.

The measure of effectiveness of this study is the duration of stopped vehicles (delay) during 10 minutes of simulation. This measure of effectiveness defines total delay of the vehicles in the system. Higher measure of effectiveness indicates congestion in the network and lower measure of effectiveness indicates that a suitable timing plan is assigned to all three traffic signals. Park and Schneeberger (2003) used a microscopic simulator to simulate a traffic Signal System. They selected two measures of effectiveness: Eastbound left-lane travel times and maximum queue length between the intersections. They determine that these measures of effectiveness were chosen because of their ease of collection from their simulator.

3.4. Traffic Signals

The Manual of Uniform Traffic Control Devices (MUTCD, 2003) describe that traffic signals are devices which control the traffic flow in entire network and guide the traffic flow into links of arterial network. According to MUTCD (2003) there are several kinds of traffic signals that control the flow of vehicles and pedestrians in the networks. Traffic signals are the most complicated form of traffic control devices.

The standard traffic signal indicates three colors: Green, Yellow and red. In the typical sequence of color phases, the green light allows vehicles to proceed in the given direction. The yellow (or amber) light warns that the signal is going to change to red and the red signal prohibits any vehicle from entering into intersection (Traffic signal timing manual 2008). In this study, in all directions of each intersection, drivers are not allowed into the intersections if the lights are yellow or red.

Cycle length is another parameter which defines the time required for a complete sequence of indications in a traffic signal. Within a cycle, splits are the sections of time allocated to each phase at an intersection. The measured split for a phase consists of its green time, yellow change, and red clearance times. These are calculated based on the intersection phasing and flow of vehicles. Splits can be expressed either in percentages of the cycle or in seconds.

Figure 3.7 is a time-space diagram that shows a simplification of the signal indications for the phases.

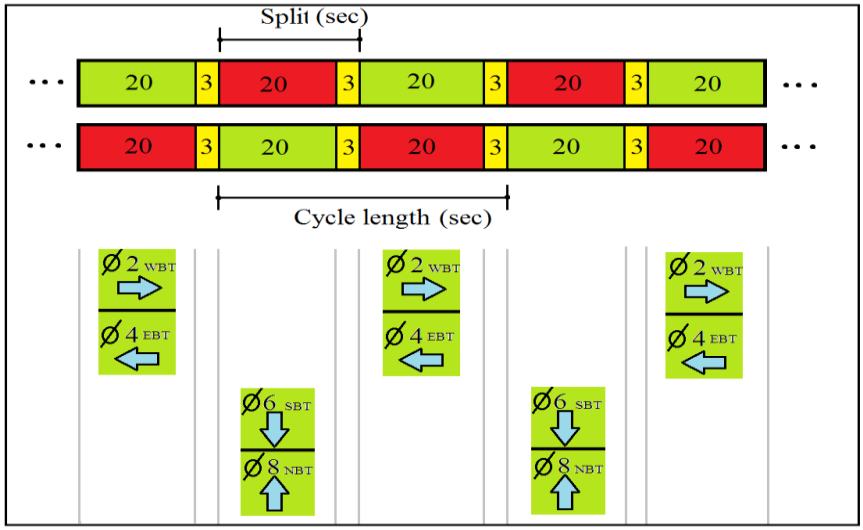


Figure 3.7. Cycle length and splits of all traffic signals.

Finding best traffic signal timing plan play an important role in reduction of congestion in entire network. According to (USDOT, 2001), an individual's travel distance approximately is 40 miles per day in 1990. At the same time, traffic signals at busy intersections in urban area might direct the movement of about 100,000 vehicles per day in a typical day. In fact more than ten percent of all intersections in California carry more than 60,000 Average Daily Traffic (ADT) for movements (HSIS, 1998). It is obvious that by determining the best timing plans for traffic signals of a network, congestion can be reduced.

Optimizing the traffic signals means determining the best signal timing plans which creates the smallest amount of delay in the network. Optimizing an isolated traffic signal's timing plan is not a time consuming operation but, determining best timing plan for several intersections is a challenging task.

3.5. Heuristic Algorithms

Glover (2003) expressed that Heuristics, in their original definition, are solution methods that coordinate an interaction between little improvements and overall strategies to avoid local optima and reach global optima in a solution space. Heuristic could refer to any device used for solving a problem, be it a program, a data structure, and etc. (Romanycia et al., 1985).

The mechanism and algorithms of heuristic algorithms are described in this section. In this study three heuristic algorithms selected as search algorithms of solution space. Then, performance of each heuristic is evaluated.

3.5.1. Definition of Chromosome

In the case of Genetic Algorithms (GA) and heuristic algorithms, numbers of strings are used. In GA's literature these strings are often referred to as chromosomes. Chromosome is a string that consists of elements. Each one of these elements is a gene. The design of the chromosome and its parameters is specific to the problem to be solved (Reevs, 2003). Figure 3.8 illustrates three samples of chromosomes. It can be seen that a string or chromosome consist of genes and these genes are the characteristics of chromosome. Gene can be a number, character, letter and etc.

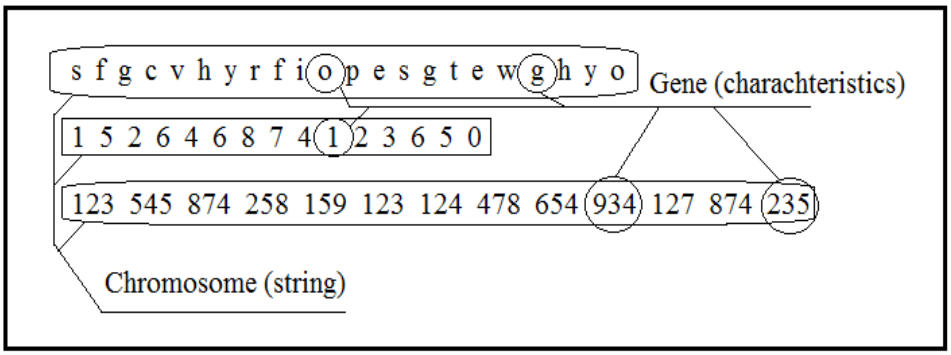


Figure 3.8. Shape and structure of chromosomes (string).

3.5.2. Solution (Search) Space

Solution space is a set of all possible solutions or chromosomes of an optimization problem that satisfies the problem's constraints, potentially including inequalities, equalities, and integer constraints. Solution spaces may be bounded or unbounded. Because the number of solutions in this problem is a constant number and green and red time durations have a maximum value, the solution space of this problem is bounded. Figure 3.9 illustrates structure of solution space. It resembles a sphere, consisted of chromosomes (solutions) of the optimization problem.

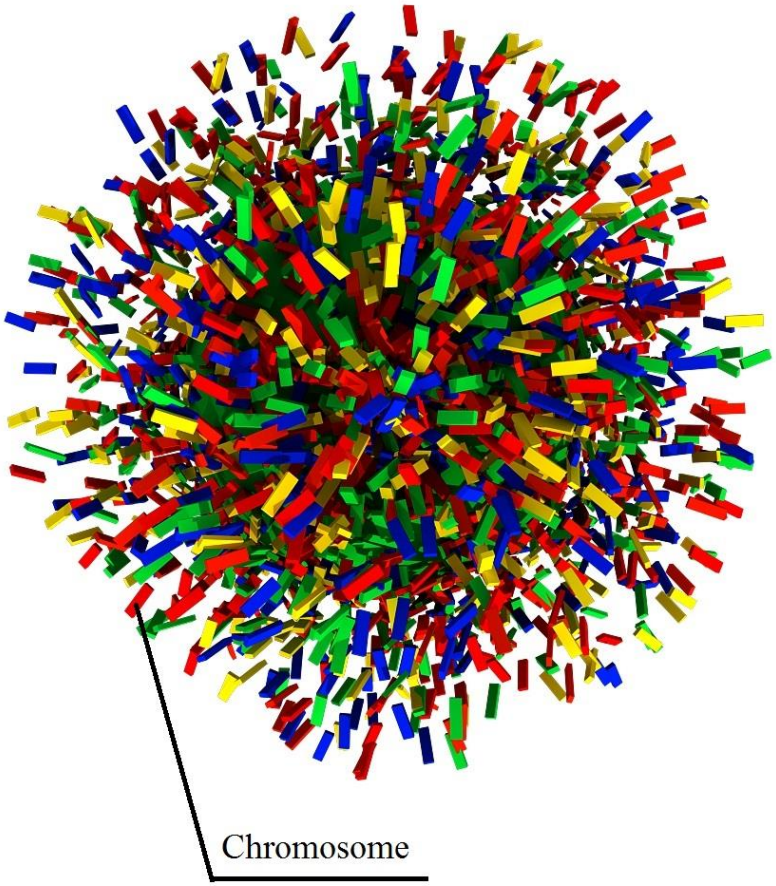


Figure 3.9. Solution (Search) space.

3.5.3. Simulated Annealing (SA)

SA is so named because of its analogy to the process of physical annealing with solids. In this method a solid is heated and then allowed to cool very slowly and thus, the solid is free of crystal defects (Henderson et al., 2003).

Henderson et al. (2003) described that SA is a local search algorithm that is able to escape from local optima. Its simple structure, convergence properties and its use of Hill Climbing moves to escape local optima, has made it a popular technique.

According to Kirkpatrick et al. (1983), SA is a heuristic to find global optima in the solution space. Kirkpatrick et al. (1983) described that SA's strength is that it avoids getting caught at local minima or maxima.

Burvall et al. (2015) suggested that SA generally determines better solutions than GA in small search spaces and that SA and GA are comparable on larger search spaces. Berstimas et al. (1993) mentioned that SA is a generally applicable and easy to implement probabilistic approximation algorithm that is able to produce good solutions for an optimization problem even if the structure of the problem is not understand well.

By taking all above mentioned descriptions and literature into consideration, it is obvious that SA is one of the efficient local search algorithms. Therefore, SA was selected as one of the heuristic algorithms to satisfy the desire of this thesis.

The SA's algorithm was defined by Cerny (1985). SA is a local search algorithm that is able to find good chromosomes (solutions) in solution space because; the SA is able to avoid local optima in solution space. The basic elements of the SA are as follows:

- Solution space.
- Objective function.
- For each solution, a set of neighbor solutions.
- A cooling schedule.
- An initial state of solutions.
- For each solution, a random number selected among zero and one.

There are several ways to create the neighbor solution. It can be created by doing crossover or mutation operations in different ways. The number of genes which mutation performed on them is named the Number of Mutation Points (NMP). The process of mutation is completely random, because randomness prevents bias selection in creating of the neighbor solution. SA needs a cooling schedule and an initial temperature. Usually, the initial temperature is started at 1.0 and decreases during the iterations by multiplying it by a constant called alpha (α). Typical α is between 0.8 and 0.99 (Aarts et.al, 2005). It is very important to select a suitable alpha value because it affects the performance of the SA. The initial state of solutions was selected randomly from solution space.

The basic algorithm of the SA is shown in Figure 3.10. Initial solution (Si) is a randomly chosen solution. Initial temperature (Ti) and final temperature (Tf) can be any number but, final temperature must be smaller than initial temperature. In third step of

algorithm, neighbor solution (Sj) is created by using mutation operator. By using the objective function, the measure of effectiveness of neighbor solution (J (Sj)) and initial solution (J (Si)) are calculated. Then difference between these two measures of effectiveness (ΔJ) evaluated. The problem is a minimization problem, therefor, if (ΔJ) is greater than 0, the neighbor solution must be equal to initial solution because, SA fails to find better solution. It is undeniable that SA can get caught in local minima. Therefor by using the acceptance probability's function, SA moves to a far neighbor solution despite of negative value of (ΔJ,). In forth step, the temperature decreases by using α. If termination criteria reach by the SA, the SA's algorithm stops to generate new solutions. Figure 3.15 illustrates the mechanism of SA for determining suitable solutions from solution space.

1	Initializing: Initial solution Si Initial and final temperature: Ti and Tf T=Tf
2	Temperatures cycle:
3	Generating Sj from Si ΔJ=J(Si)-J(Si) Si=Sj Else if acceptance probability> rnd(0,1) Si=Sj
4	Stop criterion: If the final temperature Tf is reached End Else Update T goto2

Figure 3.10. The basic algorithm of SA (Cerny, 1985).

This problem is a minimization problem and the aim of SA is determining solutions with smaller measure of effectiveness values. Negative sign of ΔJ means that neighbor solution's measure of effectiveness is less than initial's. Thus initial neighbor replaces with neighbor solution. In another hand, positive sign of ΔJ indicates that selected neighbor solution is not a better solution than the initial one. Therefore SA deletes neighbor solution and creates another neighbor solution. In this situation SA may get caught in local minima. Therefore, acceptance probability is used in the SA's algorithm. Acceptance probability is a number between 0 and 1 which allows the SA to accept unsatisfactory neighbor

solutions. This special mechanism helps SA to search the solution space in an efficient and random way. The acceptance probability is calculated by using Equation 3.7.

$$\text{Acceptance probability} = e^{(\frac{-\Delta J}{T})} \tag{3.7}$$

In Equation 3.7, (e) is the Euler's constant (2.71828…); (ΔJ) is difference of initial and neighbor chromosomes objective values. T is the current temperature of SA. Highest (T) allows SA to accept more number of unfavorable solutions. Alpha reduces (T) in iterations therefore, after some number of iterations, the SA loses energy and stays in local or global optima.

Initial temperature, alpha and NMP are three important variables of SA. In the beginning of algorithm's search process, the value of these variables should be defined for SA. In this study, several tests on the SA with different variable values performed.

3.5.4. Genetic Algorithm (GA)

One of the most useful and magnificent search algorithms is the GA. In most of literature, authors confirm the good performance of GA in solving most of the problems.

Turing (1950) proposed a "learning machine" which would parallel the principles of evolution. Turning purposed that humans may hope machines will eventually compete with men in all purely intellectual fields. After that work scientists start to create evolutionary algorithms like bee hive and ant colony algorithms which gain inspiration from nature.

Holland's book, Adaptation in Natural and Artificial Systems (1975) increased the popularity of GA. His work originated with studies of cellular automata, conducted by Holland and his students at the University of Michigan. Holland introduced a formalized framework for predicting the quality of the next generation, known as Holland's Schema Theorem.

Reeves (2003) explained that the term Genetic Algorithm, almost universally abbreviated nowadays to GA, was first used by John Holland. Reeves defined that each user's GA is unique because, many variations in population size, initialization methods,

fitness definition, selection, replacement strategies, in crossover and mutation are obviously possible.

Thompson and Bilbro (2002) proposed that it is more probable that SA may determine good solutions than GA because SA retains a single solution and GA retains a population of solutions. Therefore in some problems, SA has a better performance than GA.

Abuiziah and Shakarneh (2013) explained that GA is robust and has been proven theoretically and empirically to be able to efficiently search complex solution spaces. According to their article, a simple GA consists of three basic operators: reproduction, crossover and mutation. Because GA requires no derivative information and it is stochastic in nature, GA is capable almost to find the global optimum.

In another paper Mukhopadhyay et al. (2009) defined that Research in GA is progressing very fast and lots of scientist try to increase GA's performance. They mention that There is lots of applications of GA in current time, and approximately the usage of this heuristic is in each aspect of life.

Genetic Algorithm (GA) was first used by John Holland (1997). GA is an adaptive search algorithm based on the ideas of natural selection and genetics. In this kind of heuristic, a population of candidate solutions or parent chromosomes (PC) will evolve during the generations. In each generation, solution population will change by using mutation and cross over operators (Reevs, 2003).

The basic process for a GA is described as follows:

- Initialization. First of all some number of solutions must be created and named parent chromosomes (PC). These solutions are randomly generated and the number of solutions can be a few individuals or thousands of individuals. Each solution is named a chromosome. Each chromosome consists of number of sections which each section is a gene. In this study each chromosome consists of 6 genes. Each gene is a representative of time duration for intersections green or red light. GA's chromosome is similar to SA's chromosome.

.

- Evaluation. Each chromosome of the PC list evaluated and fitness value of each one calculated. The calculated measure of effectiveness describes that how well the selected chromosome, satisfies the problem's desire. The desire of problem can be faster process time, exact measure of effectiveness determining or both of them. It depends on the characteristics of the problem. In this study the measure of effectiveness, is a function which calculates vehicles duration of stop. By using this fitness function, delays of vehicles are calculated and overall performance of system in simulation is evaluated.
- Selection. During each successive generation, some number of chromosomes should be selected to generate new generation of chromosomes. Numbers of selection method are completely random; other types are based on the objective value of the chromosome. In the first selection method, chromosomes are selected randomly for cross over operation. In the second selecting method, chromosomes with higher objective value have bigger chance to be selected. The selecting operation varies from problem to problem and directly affects the performance of GA. There are a different selection methods but the basic idea is the same, make it more likely that chromosomes with higher measure of effectiveness preserve for the next generation. Chromosomes are selected from the solution space to be parent chromosomes (PC) to crossover. Random selection method has higher speed and measure of effectiveness based method can consume lot of time to select chromosomes. The aim of GA is to decrease the objective values of all PCs in each generation. Selection helps to achieve this goal by deleting the bad chromosomes and only preserving the best individuals in the PC list.
- Crossover. During crossover, some number of genes combines with another PC's genes. The goal is that by combining genes from two or more chromosomes, an even better chromosome which inherits the best genes from each of its parents created. Crossover selects genes from PC list and creates new chromosomes. Therefore a randomly crossover point created and genes from both selected parent chromosomes combined with each other. Crossover can be rather complicated and very depends on encoding of the chromosome.

- Mutation. There is a demand to add some randomness into GA's parent chromosomes. Mutation typically works by making very small changes at random to a chromosome's genes. After crossover operation, mutation takes place. Mutation prevents GA to get caught in local optima.
- Repeat. After creation of first generation, GA should start again until reaching a termination condition. Termination condition depends on the structure of problem and the choice of GA's user.

3.5.5. Hill Climbing (HC)

HC is a mathematical optimization technique which belongs to the family of local search. It starts from an arbitrary solution and by changing the characteristics of solution, tries to determine good solutions from solution space.

Torbaghan et al. (2013) mentioned that HC is suitable for determining local optima but, it is not guaranteed to find the global optima in solution space. He also proofed that the proposed algorithms can be used more efficiently than some other well-known methods in the literature. The results also demonstrated that improved SA is more efficient than the improved HC algorithm. According to Chaudhary et al. (2002) HC's methods guarantee optimal solution only when the solution space has one peak or valley. For multi-modal functions, the HC method may stop with a sub-optimal solution depending on how good the base structure of HC is. HC has a simple and un-complex structure. This heuristic can reach local or global optima very quickly (Selman et al., 2006). In most of search processes, HC get caught in local optima and this is the drawback of HC. There are different kinds of generalised HC algorithms which can determine more favorable solutions than basic HC. The simplicity of the HC makes it a popular, first option amongst heuristic algorithms. HC's basic algorithm works to improve a single solution, starting from a randomly selected starting point. From the starting position, the neighbours of solutions in solution space evaluated. If a solution with lower objective value is found, the search moves to that point. The method has been called 'Hill Climbing', because the process is linked to the climbing of hills on the surface of the fitness function (Herman et al., 2007).

4. METHODOLOGY

A microscopic traffic simulator is developed to mimic real traffic data of vehicles in an arterial network. Three different heuristic algorithms are explained and tested. By using the microscopic traffic simulator and these heuristic algorithms, the suitable traffic signal timing plan for the corridor defined and Performance of these heuristic algorithms for determining suitable solutions from solution (search) space evaluated.

The algorithms are described in this chapter. Firstly, value of parameters and characteristics of the network are defined in section 4.1. Then, developed simulator is described in section 4.2. Subsequently, heuristic methods that are evaluated in this study are explained in section 4.3. After that, methods of comparing three heuristic algorithms defined in section 4.4. Results are illustrated and discussed in section 4.5. Eventually, selected heuristic algorithms are compared in section 4.6.

4.1. Definition of the Parameters

In this section, all utilized parameters in the simulator and network are defined.

4.1.1. Parameters of Car Following Theory and Traffic Flow Model

During the simulation period, Traffic flow in the network is stable. The traffic flow from west to east, north to south and south to north are 500, 100 and 200 vehicles per hour respectively. Equation 4.1 and 4.2 are the formula of IDM.

$$\frac{dv}{dt} = a\left(1 - \left(\frac{V}{v0}\right)^{\delta} - \left(\frac{S*(V,\Delta V)}{S}\right)^{2}\right) \tag{4.1}$$

where

$$S*(V,\Delta V) = s0 + \max\left[0, \left(V * Th + \frac{V.\Delta V}{2\sqrt{ab}}\right)\right] \tag{4.2}$$

The parameters of IDM are defined as follow:

Throughout this study δ equals 4. Desired speed, V0, when driving on a free road set to 50 kilometer per hour. S0 and L are set to 1 and 5 meters respectively.

The acceleration rate‘s realistic value is between 1.5 and 3 m/s^2. By choosing a number between these two values, vehicle’s reactions are much more realistic. The acceleration rate depends on the engine power a vehicle. In this study all the vehicles have the same engine power, and their drivers assume to be a normal driver, not too aggressive and not too calm. After several tests acceleration rate set to 1.52 m/s^2.

According to National Cooperative Highway Research Program (1997) a deceleration rate of 3 m/s^2 is used to determine deceleration rate.

Safe time headway parameter recommendation in German driving schools is 1.8 sec. (Treiber et al, 2000). Realistic values vary between 2 sec. and 0.3 sec. and even below. Several simulations performed and it concluded that suitable safe time headway in the entire network is about 0.5 sec that by choosing this value, the simulation is more realistic. The selected values for parameters of IDM are described briefly in Table 4.1.

Table 4.1. Parameters of IDM.

Description	Parameter	Value
Desired speed when driving on a free road	V0	50km/hr
Desired safety time headway when following other vehicles	Th	0.5 second
Acceleration in each day traffic	a	1.52 m/s^2
Comfortable braking deceleration	b	3 m/s^2
Minimum bumper-to-bumper distance to the front vehicle	S0	1 meter
Acceleration exponent, delta	δ	4

4.1.2. Defining the Solution or Chromosome and the Solution (Search) Space

For evaluating and simulating traffic flow in the developed simulator, several timing plans for the traffic signals created. Each timing plan consists of three green and red time durations. For easing the calculations and simplicity, a solution or namely, a chromosome is developed. Each chromosome consists of six numbers. Each number is a gene. First

three genes of the chromosome, in order from left to right are representatives of green time durations of three intersections. Next three genes are representatives of three red time durations of traffic signals. The shape and structure of chromosome is illustrated in Figure 4.1.

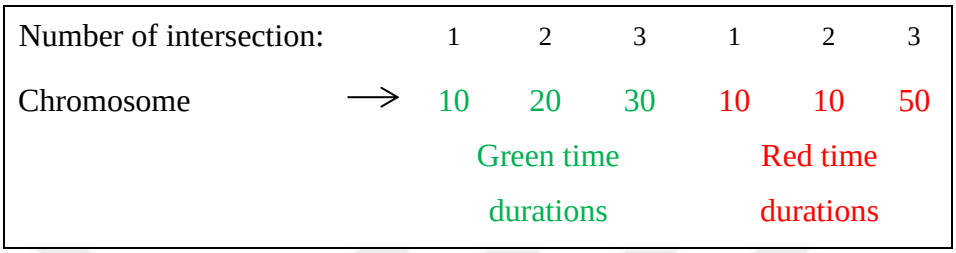


Figure 4.1. Illustration of chromosome.

According to Traffic Signal Timing Manual (2008), in this kind of road network, maximum green time and minimum green time of traffic signals are 50 and 10 seconds respectively. It is obvious that there are 40 seconds between 10 and 50 second. Thus, there can be 40*40*40*40*40*40 permutations of traffic signal timing plans.

By means of reducing the number of chromosomes and shrinking the size of solution space, the lights duration are selected among 10,20,30,40 or 50 seconds. This action shrinks the solution space and the number of solutions fall to 5^6=15625 chromosomes. Table 4.2 shows small sample of possible chromosomes and duration permutations.

Table 4.2. Permutations of 10, 20, 30, 40 and 50 seconds time durations.

Number of chromosome	Chromosome
1	10 10 10 10 10 10
2	10 10 10 10 10 20
3	10 10 10 10 10 30
…	…
15624	40 50 50 50 50 50
15625	50 50 50 50 50 50

Possible number for each gene can be selected among 10,20,30,40 and 50. Therefore, there can be 5*5*5*5*5*5 combination of numbers in the solution space which equals 15625. Figure 4.2 shows the process of creation of the solution space. It can be seen that by changing the value of genes, all possible chromosomes are created.

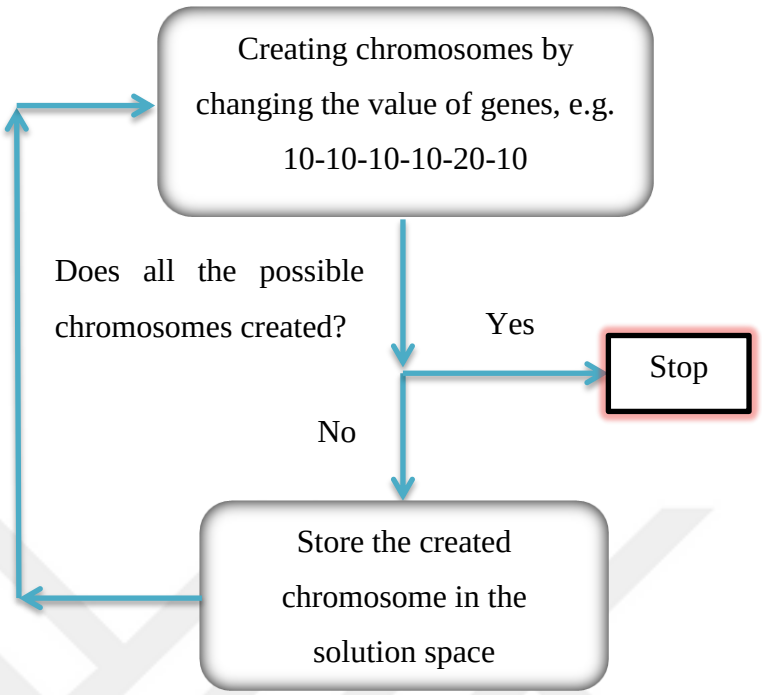


Figure 4.2. Flowchart of defining the solution space.

4.1.3. Geometry of Network and Characteristics of Traffic Signals

The main road which has greater traffic flow and connects all three intersections is a one lane collector road. Six perpendicular streets that cut the main road are minor roads. Minor roads are local roads and speed limit in the entire network is 50 km/hr. All the roads have one lane; there is not any parking facility for vehicles along the roads and no one of vehicles are allowed to stop or park during the simulation. At intersections, there are not any left or right turns. Therefore, there is not any turn phases in intersections and vehicles move through in the intersections. The slope of entire network equals zero. The stop line for vehicles in the intersections is 20 meters away from the center point of intersection. Lane width in the network is 3.6 meters. Figure 4.3 shows the entire network in detail.

The road network consists of three intersections, the distance between two nearby intersections is 200 meters. All three intersections are 4-way intersection involve a crossing over of two streets, the crossing streets are perpendicular to each other. The intersections are Signal-controlled intersections that depend on electric pre-timed traffic signals.

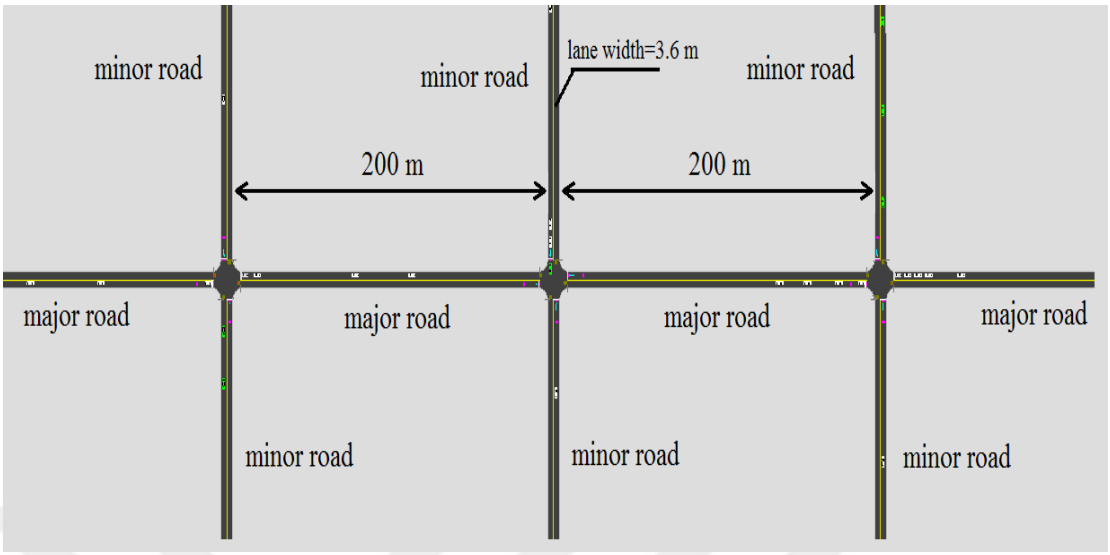


Figure 4.3. Geometry of the network.

During simulations, there is not any pedestrian or cyclist in the network.

Each intersection's timing plan consists of two phases. In the first phase east bound through (EBT) and west bound through (WBT) are allowed to move. In the phase number two, north bound through (NBT) and south bound through (SBT) are allowed to move.

As showed in Figure 4.4 It can be seen that there is two phases for each intersection. The yellow time duration of the entire network is 3 seconds and there is not any red clearance time in entire network. In this study, the minimum and maximum green times are 10 and 50 seconds. Therefore, greatest and smallest cycle lengths are 106 and 26 seconds respectively.

Barrier is a separation that separates the movements in rings to prevent accidents in intersection. Ring is sequence of traffic lights or phases (Traffic Signal Timing Manual, 2008). Figure 4.4 shows the phases and ring - barrier diagram of three intersections.

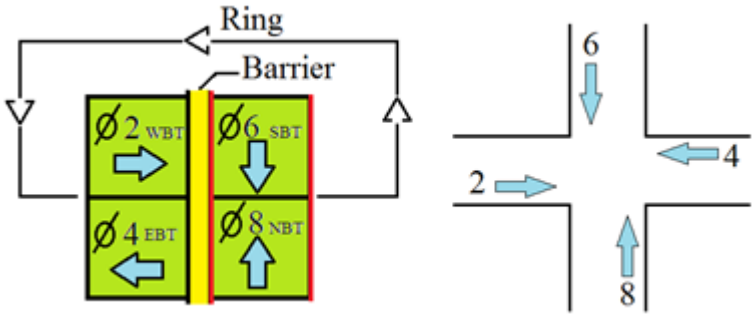


Figure 4.4. Phases and ring-barrier diagram of all intersections.

4.2. Development of Microscopic Simulation Model

The simulator of this thesis, utilize Intelligent Driver Model (IDM) as car following model to simulate the real traffic flow in entire network. In this section the developed simulator is described step by step as follows: 1- Definition of required parameters, 2- Calculation of vehicle arrivals during the Simulation, 3- Calculation of traffic signals timing plan for Each Signal, 4- Simulation of traffic flow, 5- Simulation visualization, 6- calculation of objective value.

In first step, by utilizing the selected chromosome, a matrix for every one of green, red and yellow lights created. Then by using these matrixes, timing plan of every light is calculated. When hourly flow of traffic is under 800 vehicles, accordingly, all the vehicles will travel independently in the links and time headways could be consider as random time headways and headway distribution will be random time headway distribution. Thus, the Poisson distribution is used in the structure of simulator (May, 1990).

By using Poisson distribution, arrival times of all vehicles are generated. After that, by taking IDM's formulas into consideration, acceleration of every vehicle in the network evaluated. Then the speed and position of each vehicle for every second calculated and stored in database. Special algorithms forced vehicles to stop before red and yellow lights and allow vehicles to enter intersections during green lights. Each vehicle tries to reach the maximum speed limit and keep following the vehicle ahead with safe distance. Another operator prohibits the vehicles to proceed in the intersection if vehicle's exit from it is not clear. Figure 4.5 illustrates this mechanism of simulator in detail. It can be seen that despite of green light of traffic signal, when the link between two intersections is full of vehicles,

vehicles before first intersection in left side did not proceed into intersection. Then, position of all vehicles is stored in the database for further operations.

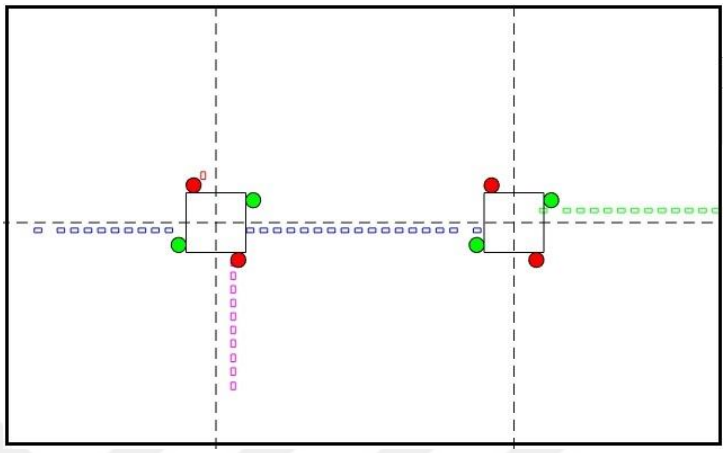


Figure 4.5. Details of the operator which prohibit vehicles to inter intersection.

Later, information from database and Matlab's powerful visualization tools are utilized to depict the simulation. These tools depict the duration of the simulation for the operator to see the performance of simulator and correct errors of the simulation.

Finally, the information from database provides objective value of the selected chromosome. The operator searches the speed's matrix and counts the elements of matrix that are equal to zero. By summing up all the zero speeds, objective value of selected chromosome calculated. Figure 4.6 depicts the flowchart of evaluation of a chromosome. By using the MATLAB code and objective function, Measure of effectiveness of each chromosome evaluated.

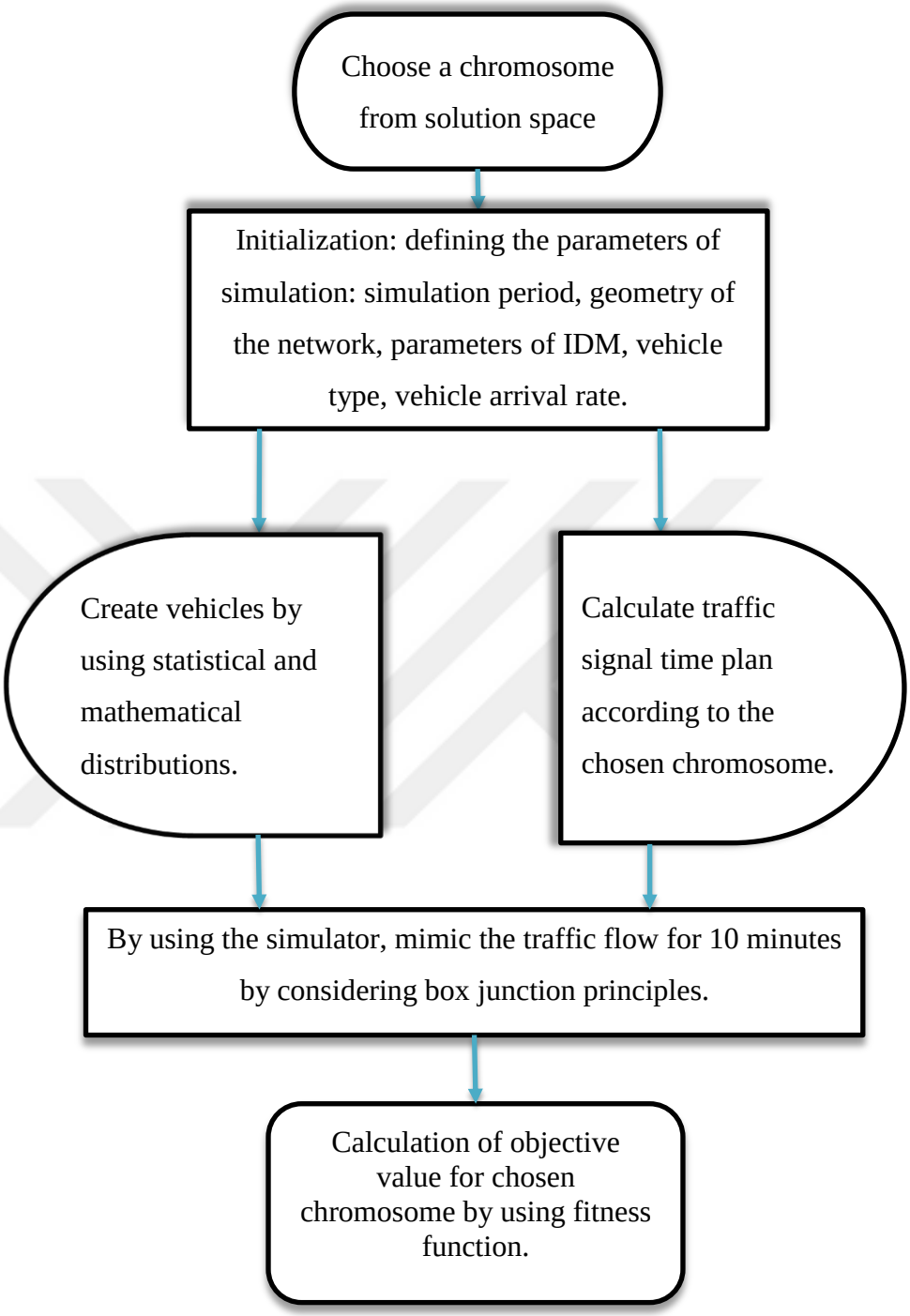


Figure 4.6. Calculating the measure of effectiveness by using the simulator.

4.3. Heuristics

Heuristic methods are used for determining the best timing plan among the solution space. They search the solution space to determine the global minima. Every heuristic has different characteristics which affects its performance. Therefore, in this study, the performances of heuristic algorithms are compared. Three different heuristic algorithms

are selected to search the solution space. Hill Climbing(HC) , because its algorithm is very simple and fast, Simulated Annealing (SA) because it is more complicated than HC and can search the solution space in different way and finally the Genetic Algorithm(GA) are chosen. The GA's structure is more complex than HC and SA. The GA needs more parameters than SA and HC like mutation probability and crossover operator parameters and it searches the entire solution space in much more complex way. Input for the heuristic algorithms is a randomly selected signal timing plan and the output of heuristic algorithms is an improved signal timing plan.

4.3.1. Genetic Algorithm (GA)

In this study GA, prefers to keep suitable chromosomes and eliminate weak chromosomes from parent chromosome (PC) list. This speciality helps GA to determine worthy solutions from solution space but, this complex structure increases the computation time of GA. The searching process of GA is described as follows: In the first step, some numbers of chromosomes are selected from solution space randomly. Then by using the simulator, the objective value of each PC is calculated. Later, by using elitism principle, two chromosomes with highest objective value are eliminated from the PC list. Elitism helps to preserve the good genes and chromosomes for the next generations and increases the overall performance of GA. After that two pair of parent chromosomes selected. The selection method is completely random in each generation and the chance of selection for each parent is equal. This action prevents bias choice. Subsequently, a random cross over point is defined and the crossover operator is combined two parent chromosome genes. This new generation's chromosomes, inherit genes from two different parent chromosomes. The crossover operation is depicted in Figure 4.7. For adding randomness to GA, and preventing GA from falling in local minima, mutation operation performed. In this step, the Mutation probability (MP) defined for GA. MP is a real number between 0 and 1. Then several random, real numbers that are between 0 and 1 are generated. Number of these random numbers is equal to number of genes and each one of these numbers is assigned to a gene. Then every gene's corresponded random value is compared with MP. If the gene's random value was smaller than MP, the gene is replaced with a random number which is selected among 10,20,30,40 and 50. After all, new generation's chromosomes created and objective value of each chromosome calculated. If termination criteria are

reached by GA, GA is stopped to generate new chromosomes, otherwise all the above mentioned steps are repeated until the termination criteria is reached. Figure 4.8 shows the process of GA of this study.

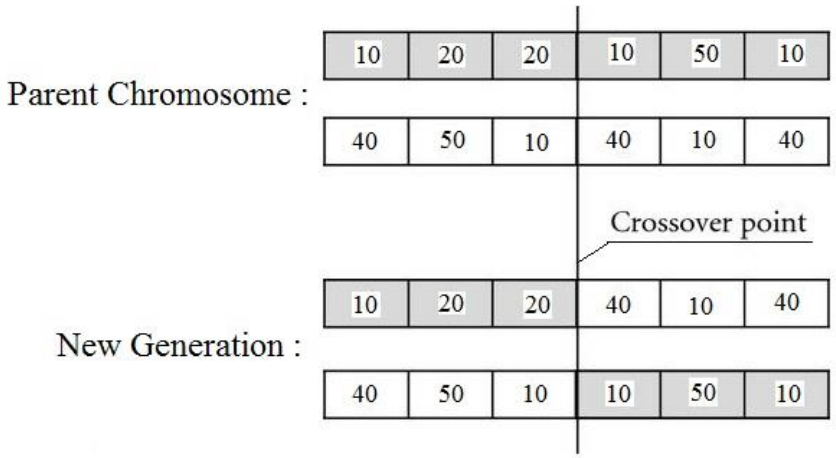


Figure 4.7. Genetic Algorithm's Crossover operation.

The Mutation Probability (MP) and Number of Parent Chromosomes (NPC) can affect the performance of GA for determining suitable solutions from solution space. Therefore, several tests performed for determining the size of effects.

For determining the best value of MP, different values assigned to MP and for each assigned value, 1000 runs performed. In each run, following parameters evaluated:

- Minimum objective value. This parameter is the minimum objective value that GA determines after reaching termination criteria.
- Number of generation. Is the number of generation that GA reaches termination criteria.
- Computation time. Is the consumed time by computer for evaluating a generation.
- Unsuccessful attempts. Unsuccessful attempts mean the number of runs that GA fails to reach desired objective value and reaches the generation number 1000.

After 1000 runs, the average value of minimum objective value, number of generations and computation time calculated. By evaluating these average values and number of unsuccessful attempt, the best value of MP selected and set For GA.

For determining the effects of NPC on the GA, the same tests executed for NPC and the suitable value of NPC selected and set.

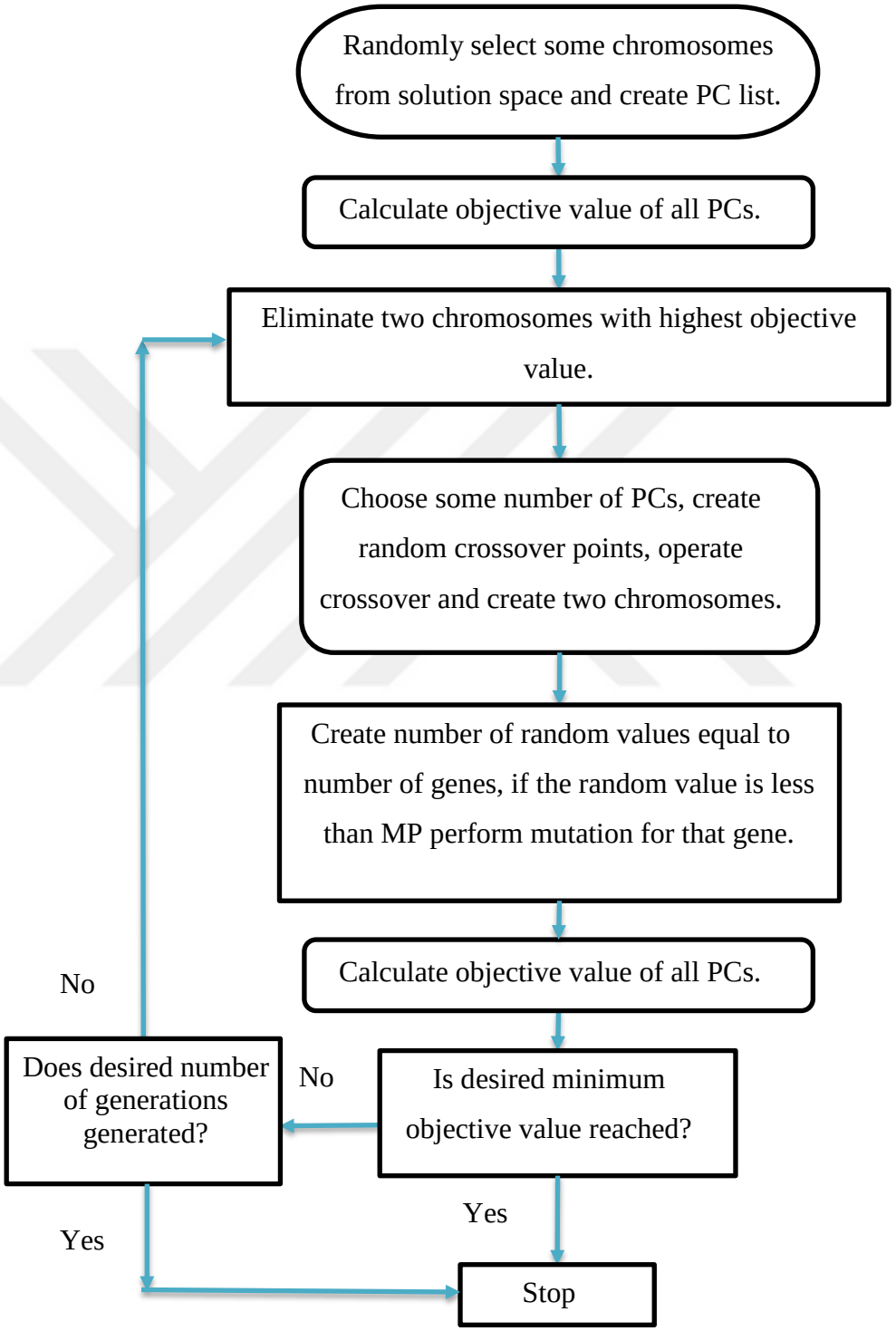


Figure 4.8. Flow chart of GA's search process.

4.3.2. Simulated Annealing (SA)

The SA's searching process is simpler than GA's. In several steps, SA tries to find and minimize objective value. After number of generations, if termination criteria are reached, the algorithm stops to generate new generations, otherwise SA resume generating new generations. In primary iterations, SA has high temperature and SA searches the solution space in a semi-random way. After several numbers of iterations, the temperature decrease and the search area's size reduce. In this condition, SA falls to local or global optima. For determining the effects of initial temperature, number of mutation points (NMP) and alpha value (α) on the performance of SA, several tests performed. After that, the results of different values of these variables compared to each other and best values for variables have been set for SA.

Mutation process of SA is described as follows: one, two or three numbers of initial solution's genes are selected randomly. Then a random number among 10, 20, 30, 40 and 50 is selected and replaced by the selected gene(s). All the process of mutation operation is described in Figure 4.9. This sample mutation shows that for generating the neighbor solution, one, two or three genes of solution (chromosome) are replaced by a random generated number.

Initial solution						:	10	10	20	30	10	20
Selected gene number						=	4					
Second random number						=	50					
10	10	20	~~30~~	10	20	:	10	10	20	50	10	20
1	2	3	4	5	6							
Neighbor solution						:	10	10	20	50	10	20

Figure 4.9. Mutation process of a SA sample.

The search process of SA is illustrated in Figure 4.10. It can be seen that in the first step, a random initial solution is created. Then by performing mutation, neighbor solution is generated. After that, by using the simulator, objective value off both initial and neighbor solutions are calculated. In iterations, the temperature of SA is reduced by alpha value and if the termination criteria are not reached, the heuristic resumes its search process.

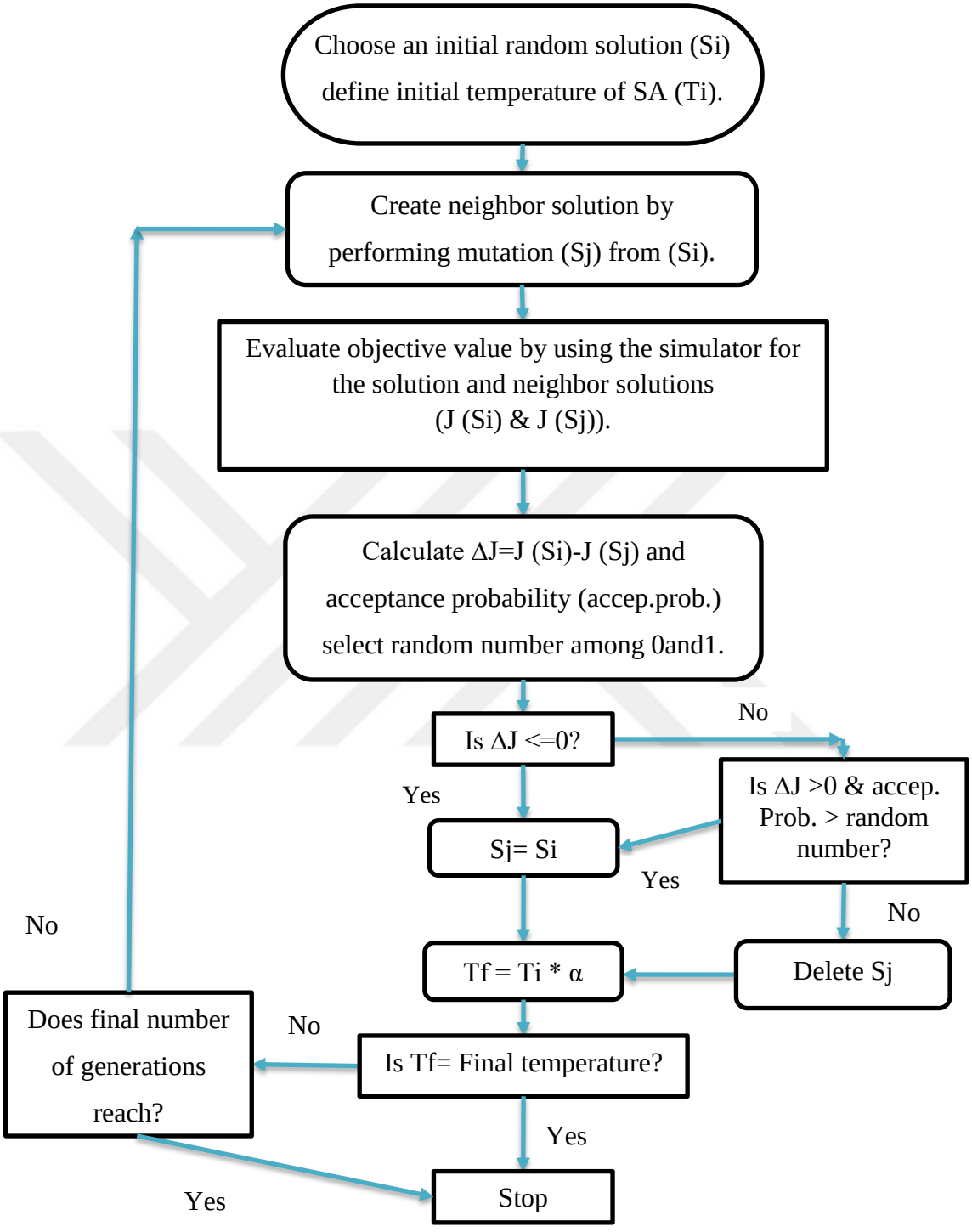


Figure 4.10. Flowchart of the SA.

The Simulated Annealing has three important variables: Alpha, initial temperature and Number of Mutation Points (NMP). Each one of these variables affects the SA's search performance in different way.

The alpha value decreases the temperature of SA in iterations; therefore the initial temperature and alpha value have a direct relationship. Thus several tests performed for all available combinations of alpha and initial temperature values. In each test, 2000 runs with different values of initial temperature and alpha performed. Then, the following parameters evaluated:

- Number of generations. Similar to GA, this parameter is the number of generation which SA reaches the termination criteria and stops to create new generations.
- Unsuccessful attempts. Identical to GA's parameter, this value defines the number of attempts which SA reaches generation number 2000.

After 2000 runs, the average value of number of generations calculated. By evaluating these average values and number of unsuccessful attempts, the best value of alpha and initial temperature opt for GA.

For determining the effects of NMP on SA's performance, the same tests performed for NMP. After comparing different values of NMP, the suitable value selected for NMP.

4.3.3. Hill Climbing (HC)

The HC does not have any complex structure or variable. This simplicity reduces Computational time (CT) of HC but, most of the time HC get caught in local optima. Therefore, a re-run mechanism helps HC to restart searching process until reaching suitable solutions.

The search process of HC of this study is described as follows:

In the first part of search process, a candidate solution is randomly selected from solution space. After that by assigning 10,20,30,40 and 50 to the first gene, five neighbor solutions created. Later by using the simulator, objective value of all five solutions evaluated. Then the solution, with lowest objective value is selected and preserved for next steps. After 6 steps, HC stops to create new neighbor solutions and preserves the improved solution. Figure 4.13 illustrates the first part of HC's search process.

The arrows in Figure 4.11 describe movement of the selected solution from one step to another. The objective value of each chromosome is placed next to it. Sometimes HC is not able to find better solution.

In the second part of search process, instead of five neighbors, three neighbors are created for the candidate solution. This act reduces the computation time of HC. Selection and replacing methods are similar to the first part. The second part of process continues until step six and after all the local or global optima is reached (Figure 4.12).

The only difference between the first and second part is the number of neighbor solutions. Three neighbor solutions are created. If the gene equals 10, the neighbor solution's genes will be 20 and 30. If the gene equals 50 the neighbor solution's genes will be 30 and 40. Otherwise if the gene equals G, the neighbor solutions genes will be G+10 and G-10.

Figure 4.13 illustrates the flowchart of HC's algorithm. It is obvious that HC did not have any complex variable or search mechanism. This characteristics, increases the speed of HC for determining suitable solutions from solution space. In another hand HC did not have any mechanism to avoid local optima or minima. Therefore lots of restarts should perform with HC to find suitable solutions in search space.

The HC of this study proposed a new search method. The structure of HC of this study is described as follow:

- Initialization. One initial solution selected from Solution space randomly.
- Evaluation. By using the Matlab code, the objective value of candidate solution calculated.
- Selection. Selection method is not a random process. Neighbor solutions with higher objective values are selected.
- Repeat. After 12 generations HC stops to work. The numbers of restarts is defined by the user.

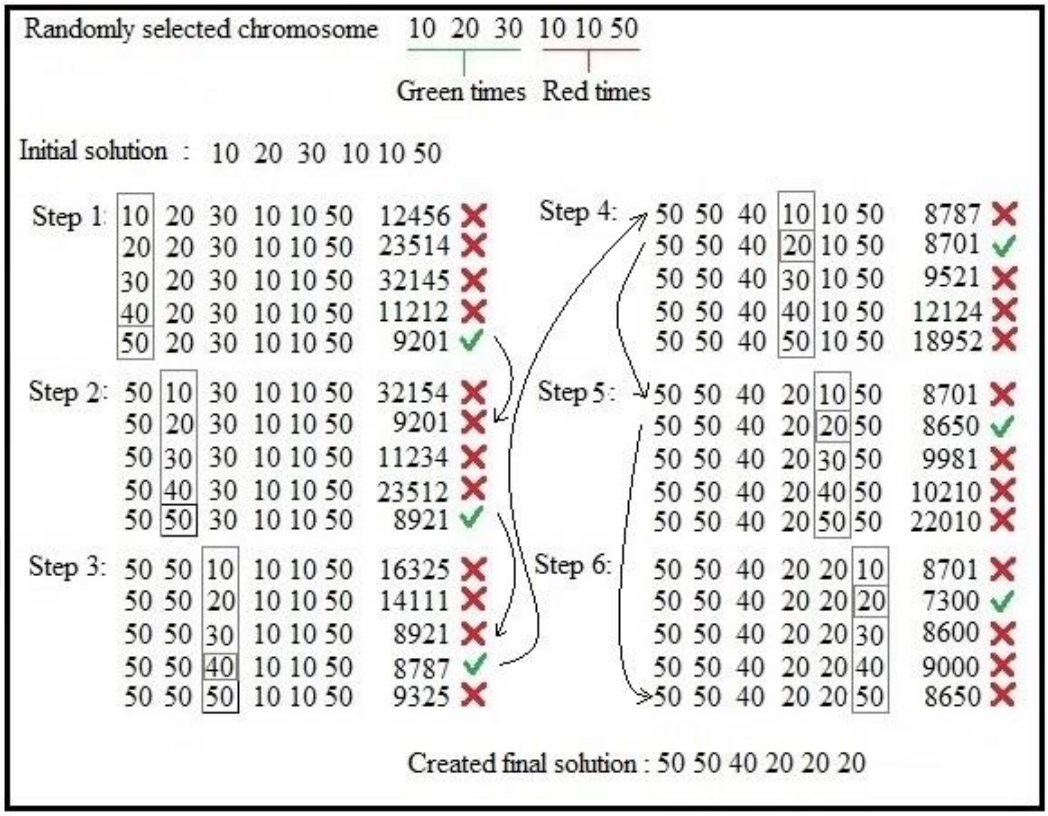


Figure 4.11. HC's first part of search process.

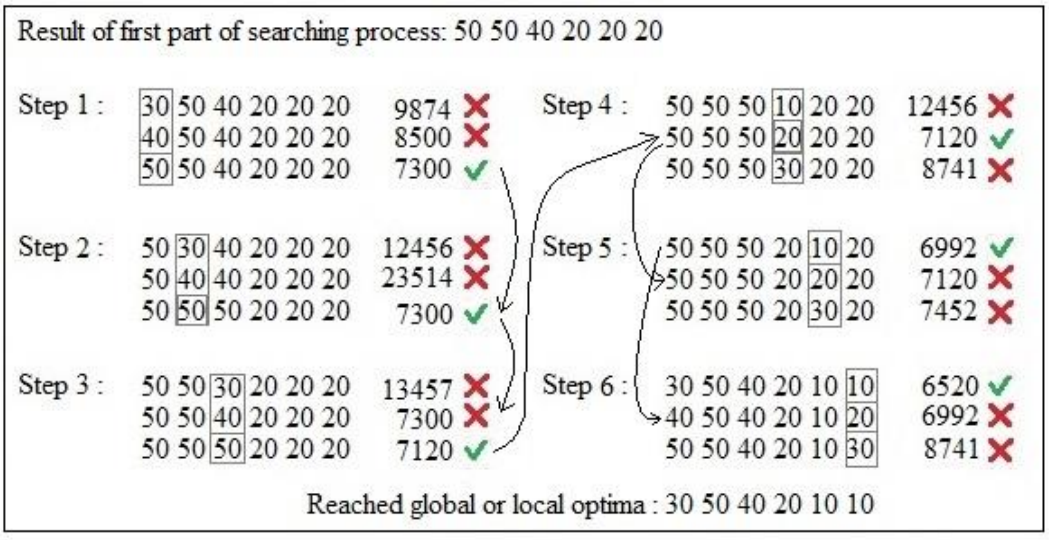


Figure 4.12. HC's second part of search process.

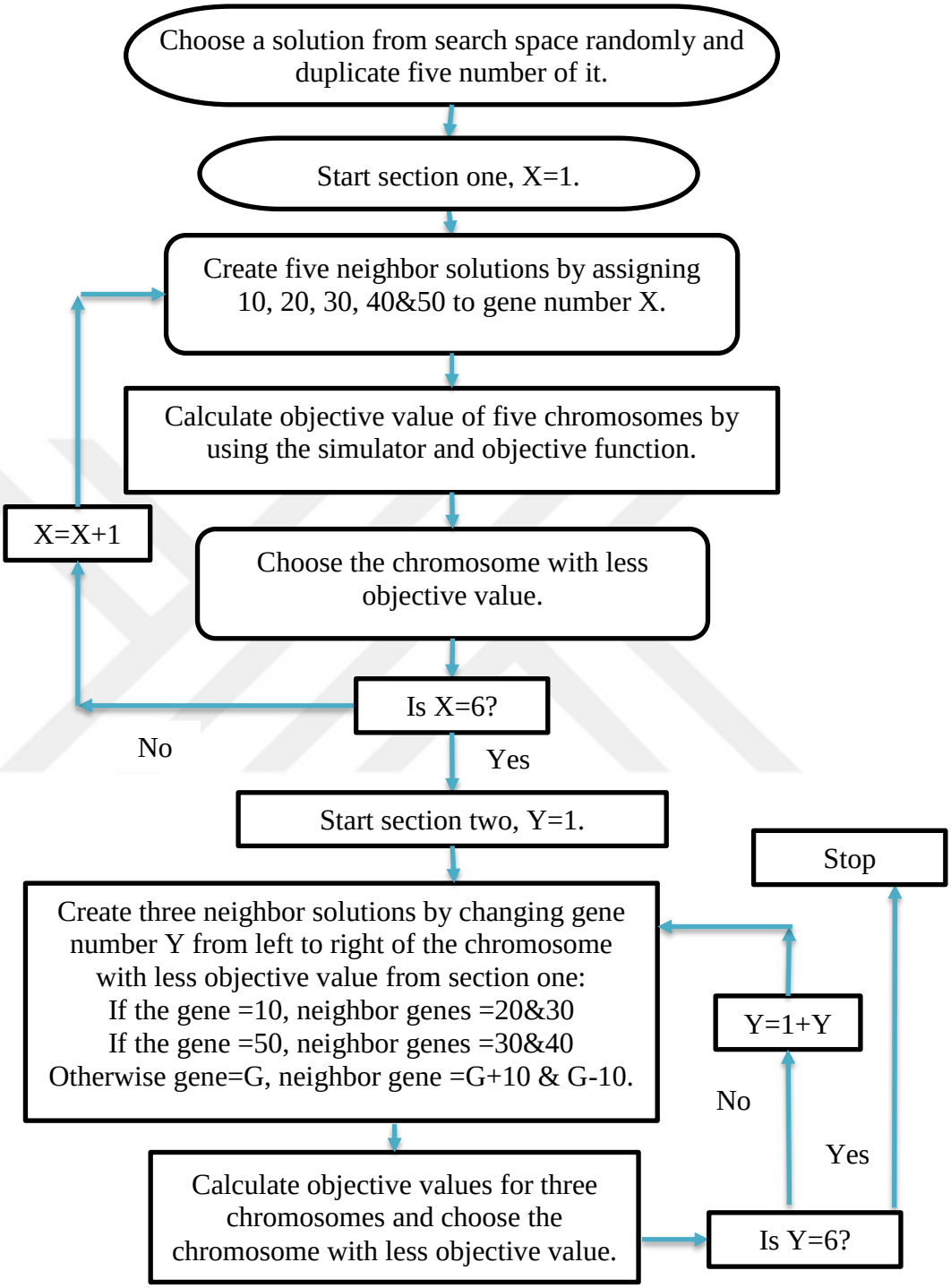


Figure 4.13. Flowchart of HC's algorithm.

4.4. Method of Comparison of the Heuristic Algorithms

The performance of the heuristic algorithms varies depending on the characteristics of optimization problems. Every heuristic in this study has different advantages and disadvantages. Therefore by comparing these search algorithms, the best heuristic is selected.

In this study, for distinguish the suitable heuristic, a special test executed. Firstly, by using each heuristic, 2000 runs performed. Then the following parameters evaluated:

- Minimum objective value. Is the minimum obtained objective value which is evaluated by the heuristic. Lower value of this parameter means that the Heuristic have better performance for determining good solutions from solution space.
- Computational Time (CT). The required time for generating and evaluating a generation or run. This parameter indicates the speed of heuristic for determining local or global optima.
- Successful attempts. Mean the number of runs that heuristic determines desired objective value.
- Unsuccessful attempts. This parameter indicates the performance of heuristic in different way. In some problems, determining the exact solution is the first priority. Therefore lower value of unsuccessful attempt is a good indicator for comparing the performance of heuristic algorithms.

After performing 2000 runs, the average and standard deviation of mentioned parameters evaluated and suitable heuristic is selected.

4.5. Results of the Simulation

Measure of effectiveness of all possible traffic signal timing plans evaluated and depicted in Figure 4.14 and 4.15. These figures illustrate measure of effectiveness of all chromosomes. It can be seen that the maximum value of the measure of effectiveness is 39201 seconds and the minimum value is 8180 seconds. It is obvious that there are lots of

local optima in the solution space and difference between maximum and minimum objective value is great.

Figure 4.14 shows scatter shape of measures of effectiveness. Figure 4.16 illustrates sorted measures of effectiveness and the difference between the measures of effectiveness of all chromosomes in detail. According to Figure 4.16, it can be seen that there is great difference between minimum and maximum value. Therefore, it is difficult for heuristic algorithms to determine the global minima.

Figure 4.17 illustrates the worst and best timing plans simulations. The above part of the figure shows the worst timing plan's effect on the network. In this condition there are long queues in streets and congestion is visualized. The below part of the figure indicates the best timing plan's performance. It is obvious that there are not long queues in the network in this situation.

According to Table 4.3 the mean and median of all measures of effectiveness are equal to 21838 and 21321 respectively. There is not significant difference between mean and median of measures of effectiveness. Global maxima is 39201 seconds and global minima is 8180 seconds.

Table 4.3. Descriptive statistics of solution space.

Description	Value
Size Or Number Of Chromosomes	15625
Global Maxima	39201 sec
Global Maxima's Chromosome	50 50 10 10 10 50
Global Minima	8180 sec
Global Minima's Chromosome	40 50 50 20 20 20
Mean	21838 sec
Median	21321 sec
Mode	17154 sec

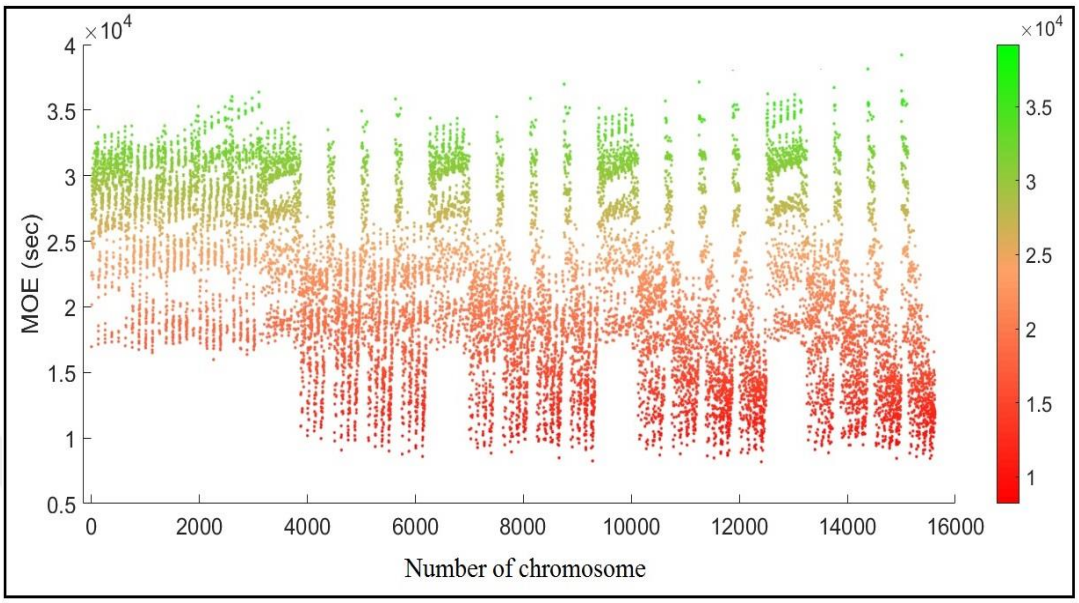


Figure 4.14. Measure of effectiveness (MOE) of all chromosomes.

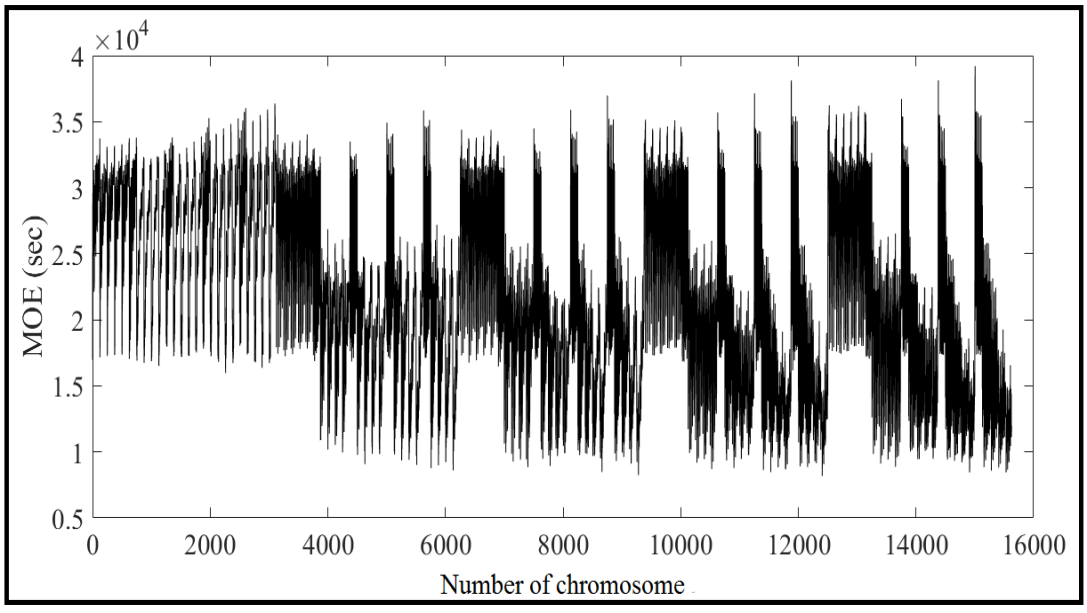


Figure 4.15. Continues shape of MOE.

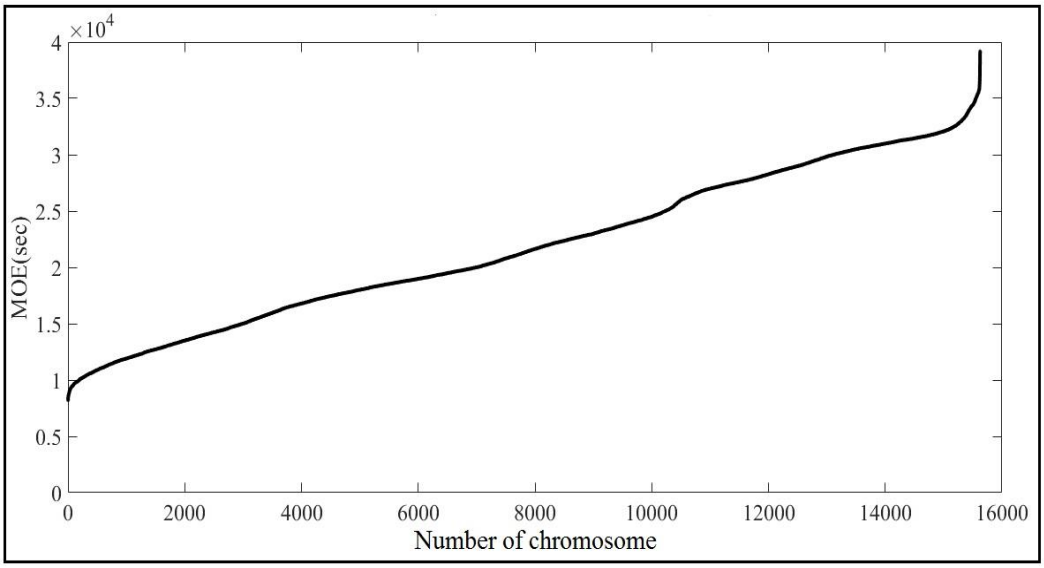


Figure 4.16. Sorted value of measures of effectiveness.

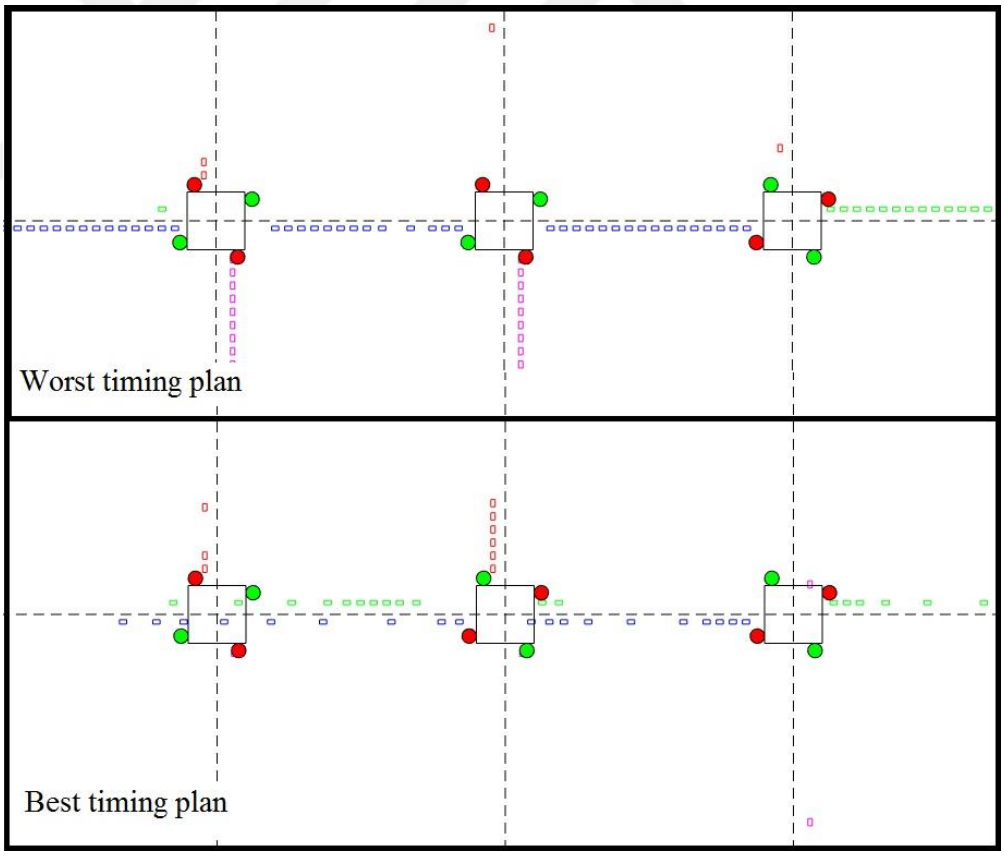


Figure 4.17. Best and worst traffic signal timing plans.

4.5.1. Genetic Algorithm (GA)

GA has two important variables: Mutation Probability (MP) and Number of Parent Chromosomes (NPC). Effects of these two variables on the performance of GA are analyzed and explained in the following section.

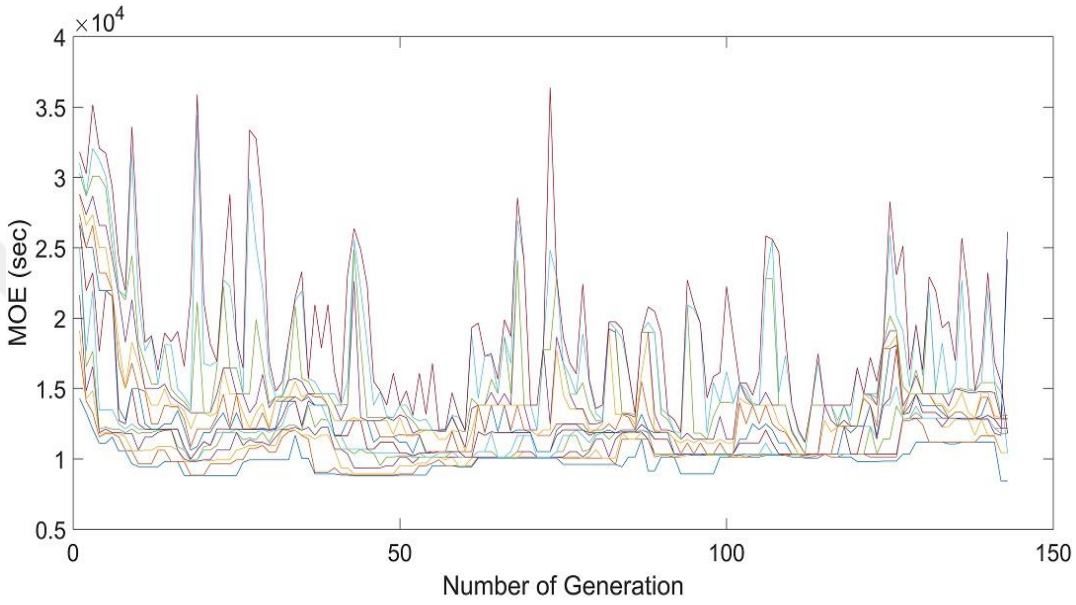


Figure 4.18. Results of the GA's process for one sample run.

Figure 4.18 shows a sample of generation process of GA. As it can be seen from Fig. 4.18, after about 140 generations the GA reaches the termination conditions and stops. In this particular run, GA has 10 parent chromosomes and each line represent the objective value of each parent chromosome during generations.

Termination conditions for the GA are: reaching 1000 generations or reaching an objective value equal or less than 8500 seconds. After reaching one of termination criteria the GA stops to generate new generation.

4.5.1.1. Mutation Probability (MP). MP is a real number between 0 and 1. Greater MP means that GA tries to find solutions from solution space in a random way. When MP has an unfavorable value, GA stays in local optima and is not able to reach and determine the global optima; therefore it is very important to find suitable MP value for the GA.

In this study, 6 numbers of MPs are chosen. The chosen MPs are: 0.2, 0.15, 0.1, 0.05, 0.025 and 0.01. For each MP, 1000 runs performed and the average objective value of each run of 1000 were calculated. Figure 4.19 illustrates the average values of the minimum objective values.

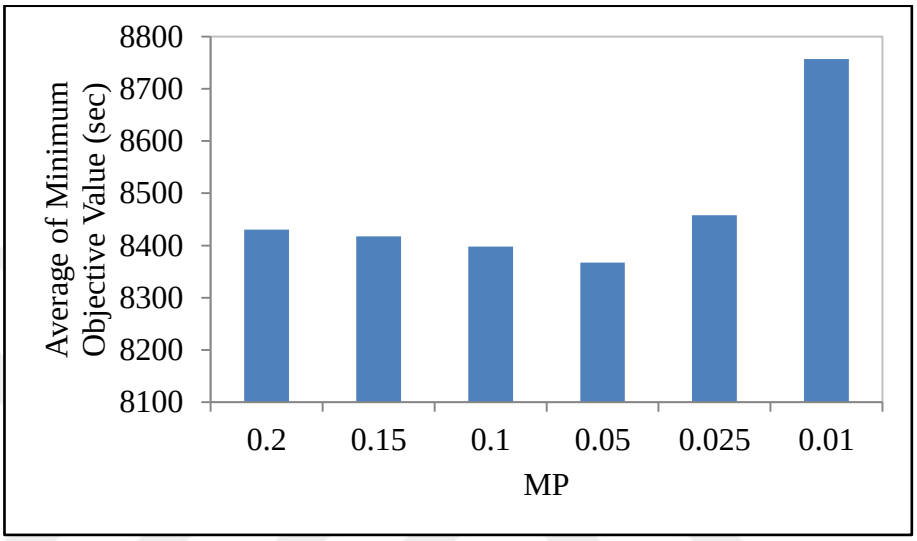


Figure 4.19. Average minimum objective values.

According to Figure 4.19, MP equal to 0.01 has the highest average value and 0.05 has the lowest average value. It is obvious that when MP is greater than 0.05, there are not any significant changes in the average of minimum value, therefore the value of MP temporarily set to 0.05.

Another important parameter for choosing the best MP is the number of generation when GA reaches the termination condition. The average value of number of generations were counted and displayed in Figure 4.20. It is obvious that when MP equals 0.05 and 0.01, GA determines lower and higher number of average generations respectively. Choosing MP greater than 0.05 is not logical because when MP is greater than 0.05, average number of generations is insensitive.

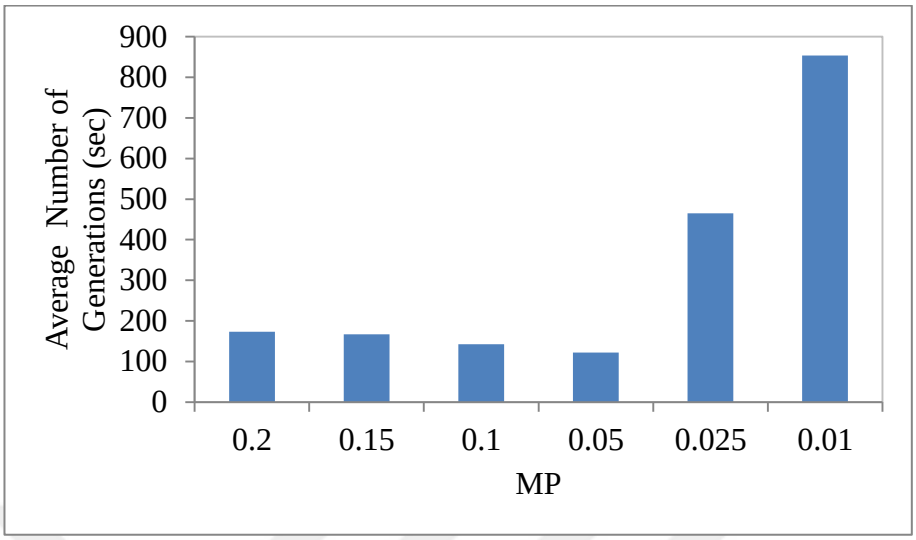


Figure 4.20. Average number of generations.

Computational time (CT) of each generation is another important parameter. Because of GA's structure, changing the MP value can affect the speed of GA. Choosing different values of MP can increase or decrease the speed of GA as a result of increasing or decreasing the number of calculations. Therefore the average CT of each generation were counted and displayed in Figure 4.21.

There is no doubt that when the MP equals 0.05, average CT of generations has the lowest value, accordingly MP equals 0.05 has the best performance among all MP values.

Figure 4.22 exhibits the number of unsuccessful attempts of GA for every MP. Unsuccessful attempt occur when the GA is not able to determine desired objective values and it reaches the final generation number. This means that GA fails to find suitable solutions from solution space. According to Figure 4.23, It can be seen that when the MP equals 0.01 or 0.025, GA fails to find suitable solutions whereas, when it is equal or greater than 0.05, the number of fails drop dramatically and approximately is equal to zero.

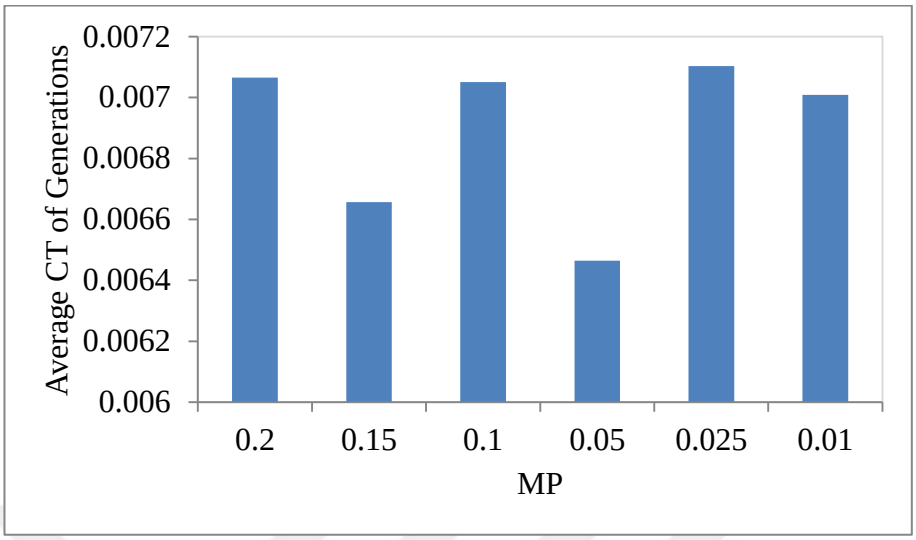


Figure 4.21. Average CT of each generation.

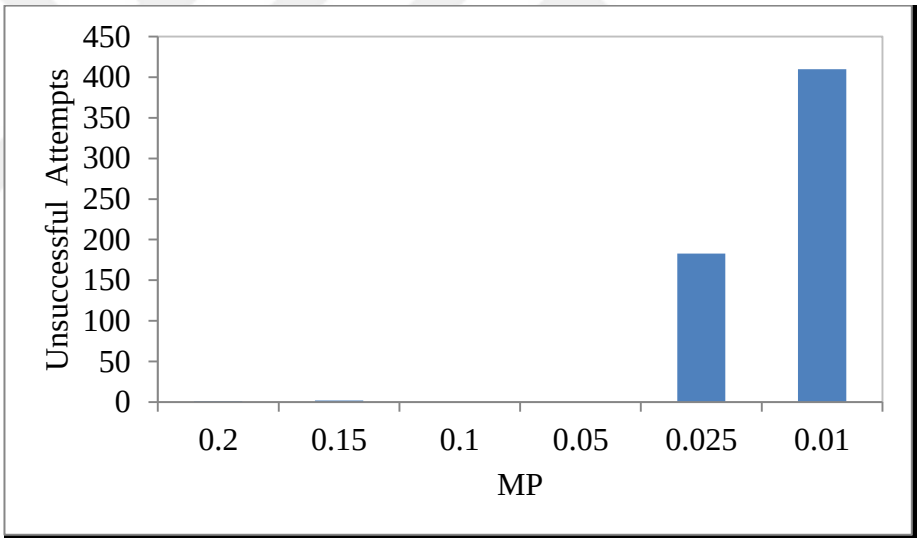


Figure 4.22. Unsuccessful attempts.

According to all above mentioned discussions, 0.05 is selected as MP value.

4.5.1.2. Number of Parent Chromosomes (NPC). The effects of this parameter on the performance of GA are described in this part. The chosen numbers of PC in this study are 20,18,16,14,12,10,8,6 and 4. There is not any difference between the termination criteria of PC and MP. Maximum number of generation of GA is 1000.

Average minimum objective value for every NPC is calculated and described in Figure 4.23. It can be seen that when the number of Pc is 4 or 6, Average value is

approximately 8500 but, when the number of PC is equal or greater than 8, the average value is about 8350 seconds. PC equals 8 has the less average minimum value. Because this problem is a minimization problem, selecting PC equals 8, 10, 12 or 14 seems to be logical.

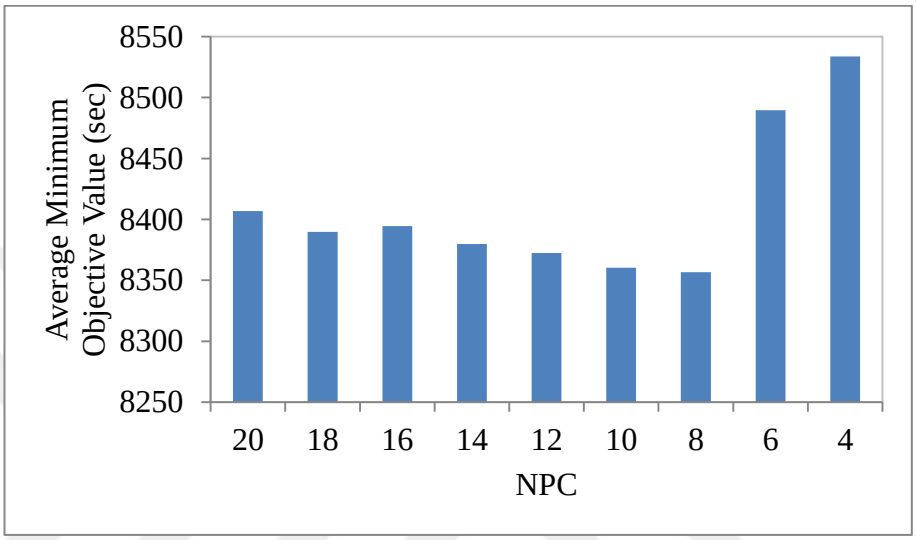


Figure 4.23. Average minimum value of objective values of NPC.

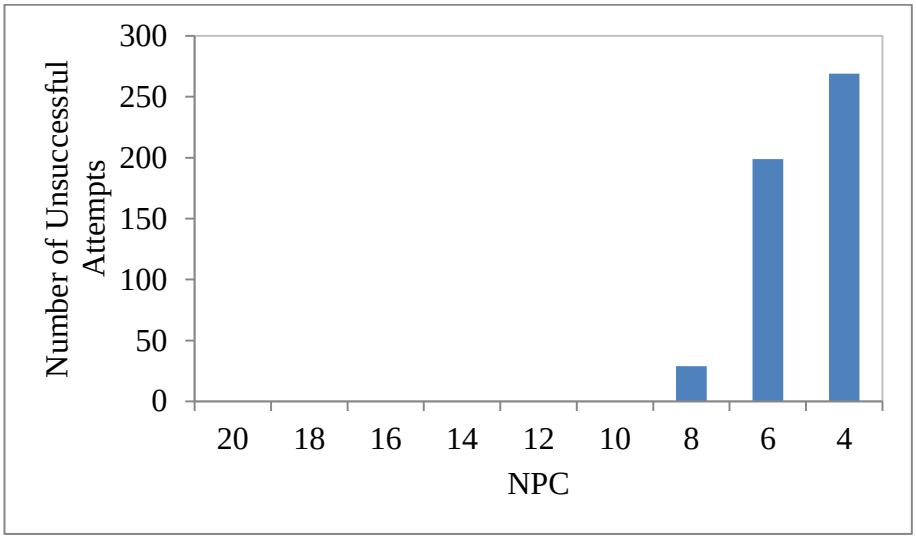


Figure 4.24. Unsuccessful attempts.

Figure 4.24 indicates that selecting NPC equal or greater than 10 seem to be rational because in this condition, the number of unsuccessful attempts equals 0 and GA never fail to find global optima.

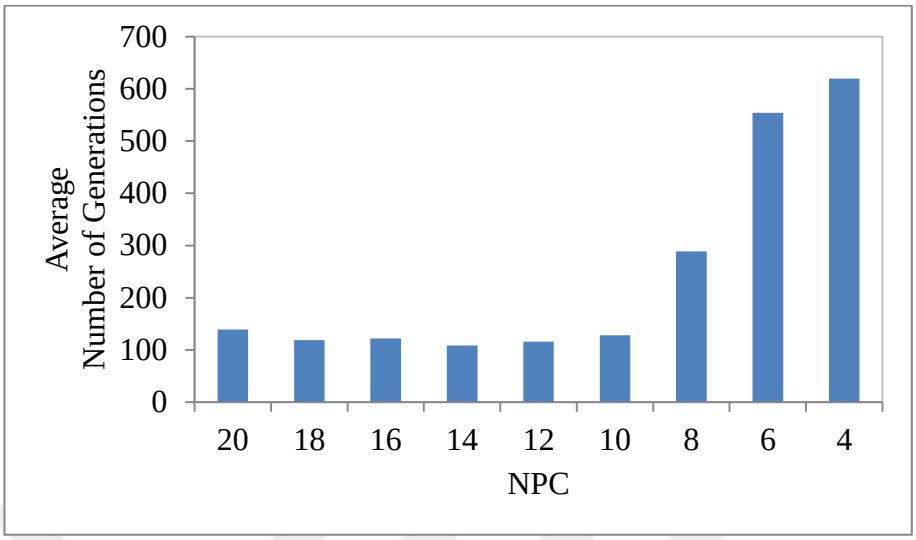


Figure 4.25. Average number of generations of every PC value.

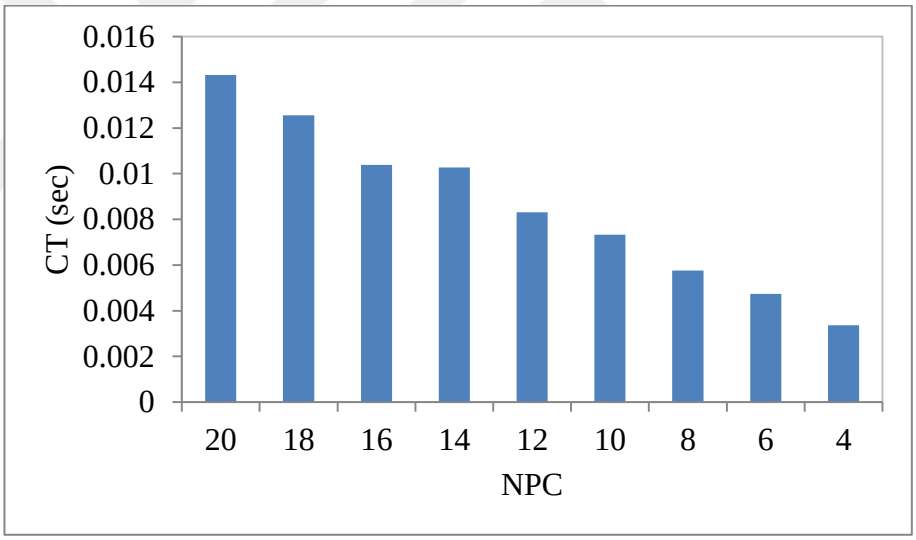


Figure 4.26. CT of the GA with different NPC values.

According to Figure 4.26, NPC equals 4 consumes the lowest computational time and PC equals to 20 has the highest CT. By increasing the NPC, the CT increases too. It seems that selecting the NPC equals 4 is the best choice because in this condition the GA works faster, but according to Figures 4.24 and 4.25, the NPC should be greater than 10. Therefore by taking all above mentioned discussions into consideration, 10 selected as NPC value.

4.5.2. Simulated Annealing (SA)

In this study, the SA has three important variables which can affect its performance. Alpha value, number of Mutation points (NMP) and initial. The effects of each parameter on the performance of SA are discussed in next sub-sections.

For testing the effects of changing the parameters on SA, 1000 runs performed by the SA for every variable, one by one. There are two termination criteria: reaching objective value equal or less than 9000 and reaching maximum generation number which is 2000. If SA reaches one of the above mentioned criteria, it stops to generate new generation.

4.5.2.1. Alpha Value & Initial Temperature. Alpha value (α) is a constant which reduces the temperature of SA in each generation. Alpha is a real number between 0 and1. Higher α means that temperature of SA reduces faster and after some number of iterations, SA does not have enough energy to search the solution space and escape from local optima. Lower α means that SA searches the solution space in a random way and most of solutions are selected randomly from the solution space. Therefor it is important to find the suitable α for the SA. Another important parameter is the initial temperature which defines the temperature of SA in the beginning of searching process. The alpha reduces the temperature of SA in generations. Therefore, different combinations of alpha and initial temperature can affect the performance of SA in different ways. Hence, effects of every combination of alpha value and initial temperature evaluated in this sub-section.

In this study Selected alpha values, for performing tests on SA are 0.75, 0.8, 0.85, 0.9, 0.95 and 0.98. There is not any constraint for choosing initial temperature. In the first step, for choosing the suitable alpha value , selected initial temperatures are 100,75,50,25,15,10,8,6,4,2 and 1. Then, in second step 500, 450,400,350,300,250,200,150,100 and 50 are selected as initial temperatures to determine the effects of high temperatures on the performance of SA. Average numbers of generations are illustrated in Figure 4.28. Higher value of alpha , leads to better performance in the SA for determining suitable solutions from solution space. According to Figure 4.27, different alpha values affect the performance of SA in different ways. When the alpha value equals 0.75, all the initial temperatures have the same performance

and average generation value is about 330 generations. In another hand, when the alpha equals 0.98, smallest average numbers of generations that is 260, evaluated.

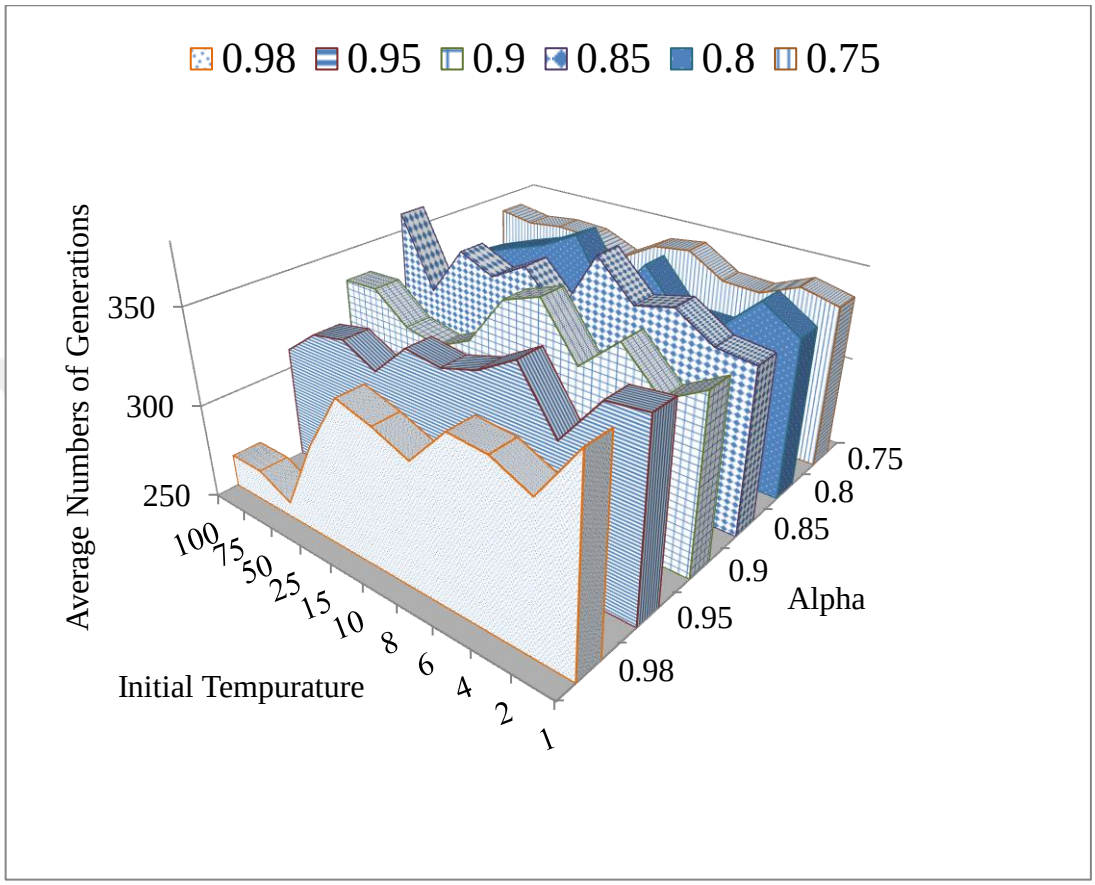


Figure 4.27. Average numbers of generations of SA.

According to Figure 4.28 when the alpha value equals 0.98, the average number of generations is decline by increasing the initial temperature. This phenomenon shows that when the alpha value equals 0.98, the SA works properly and can find suitable solutions from solution space.

Figure 4.29 shows the sample search process of SA in one run. In this particular run, the initial temperature and alpha are 100 and 0.98 respectively. It can be seen that until generation number 100, the SA searches the solution space in a semi-random way to find good solutions from solution space, after about 100 generations the temperature has been fall and SA stayed in a local optima.

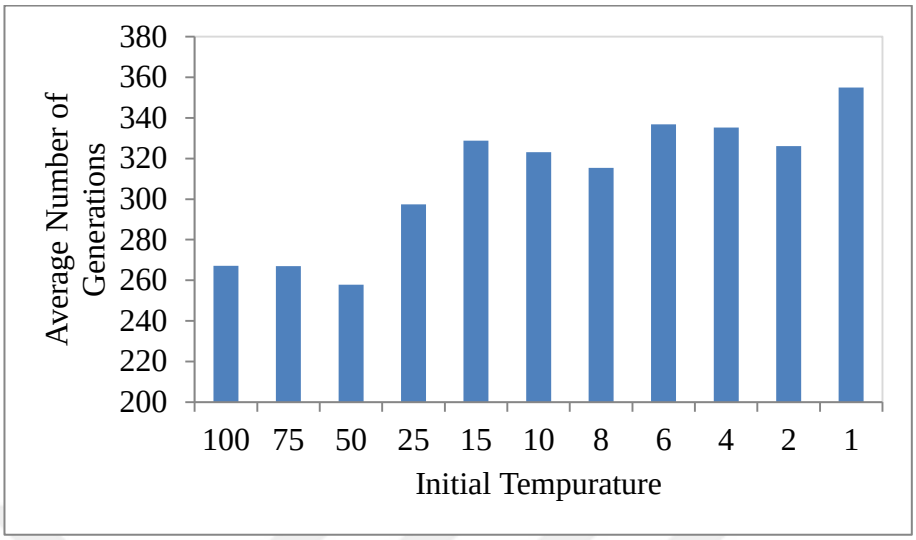


Figure 4.28. Average number of generations for initial temperatures with α equals 0.98.

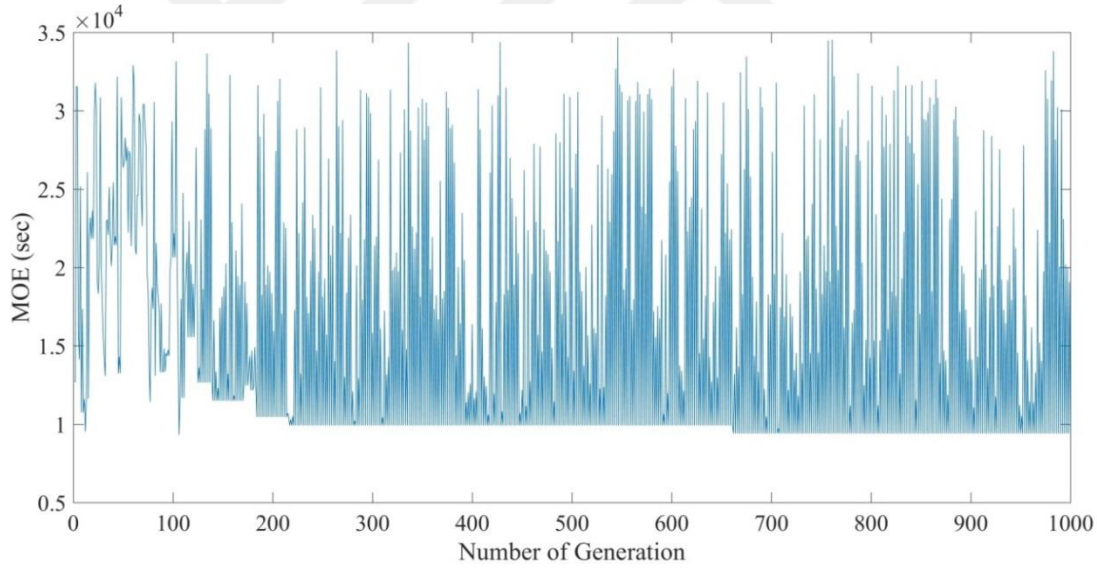


Figure 4.29. A sample Search process of SA with initial temperature equals 100 and α equals 0.98.

Besides, Figure 4.30 shows that when the alpha value equals 0.75, average number of generations is insensitive to initial temperature. Thus, performance of the SA is approximately the same in all conditions because lower alpha value decreases the temperature of SA very quickly and prevents SA from searching the solution space in an effective and semi-random way. Figure 4.31 exhibits that when α equals 0.75, SA cannot move in a semi-random way and get caught in local optima.

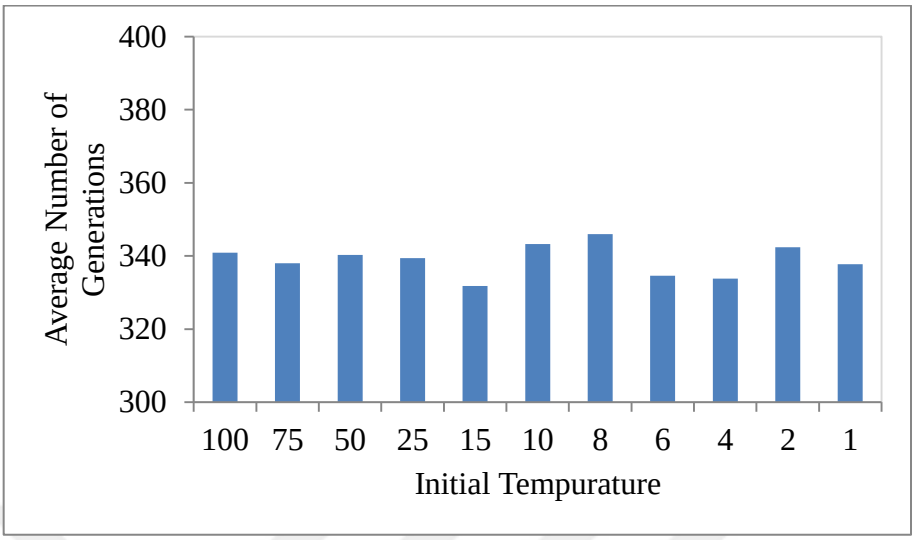


Figure 4.30. Average number of generations with α equal to 0.75.

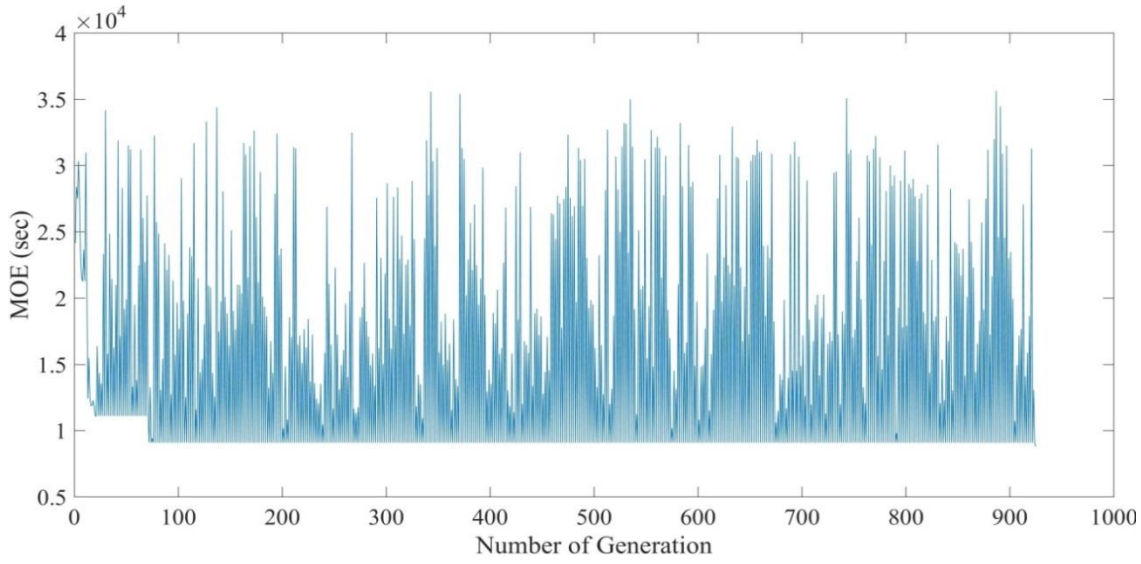


Figure 4.31. A sample Search process of SA with initial temperature equals 100 and α equals 0.75.

According to all above mentioned discussions, 0.98 is selected as alpha value. After selecting 0.98 as α value, the effects of different initial temperatures analysed and described in this section. In this part of tests, 500,450,400,350,300,250,200,150,100,50,10,5and 1 are selected as initial temperatures to determine the effects of higher temperatures on the performance of SA. Figure 4.32 shows the average number of generations for different initial temperatures.

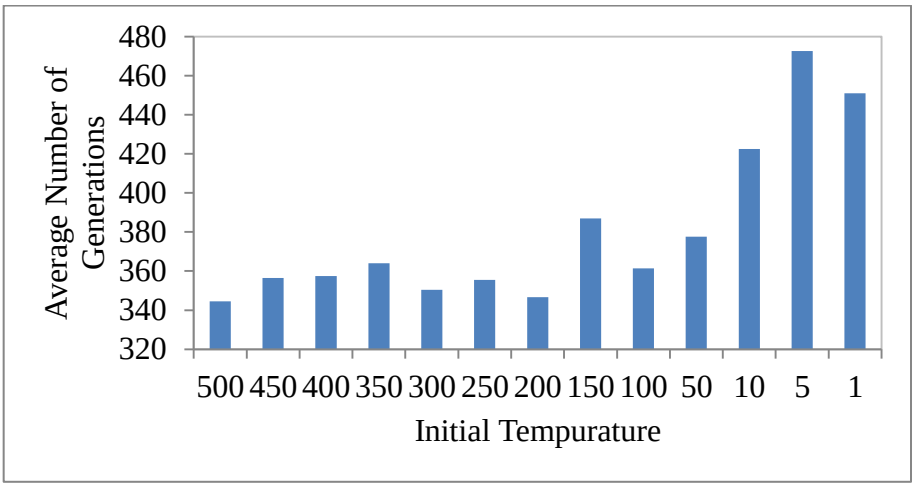


Figure 4.32. Average number of generations of different initial solutions.

According to Figure 4.33, when the initial temperature is equal or greater than 100, SA is insensitive to initial temperature. Choosing Initial temperature smaller or equal to 50 is out of question because the average values increase dramatically by choosing initial temperature smaller than 50. The best performance of SA occurs when the initial temperature is equal to 200 or 300. Therefore another test performed and results described in Figure 4.34. In this test the initial temperatures selected as 250,200,150,100 and 50. The alpha value set to 0.98. Also, the termination criteria changed and if the SA evaluated objective value equal or less than 9300 or reached generation number 2000, it stopped to work.

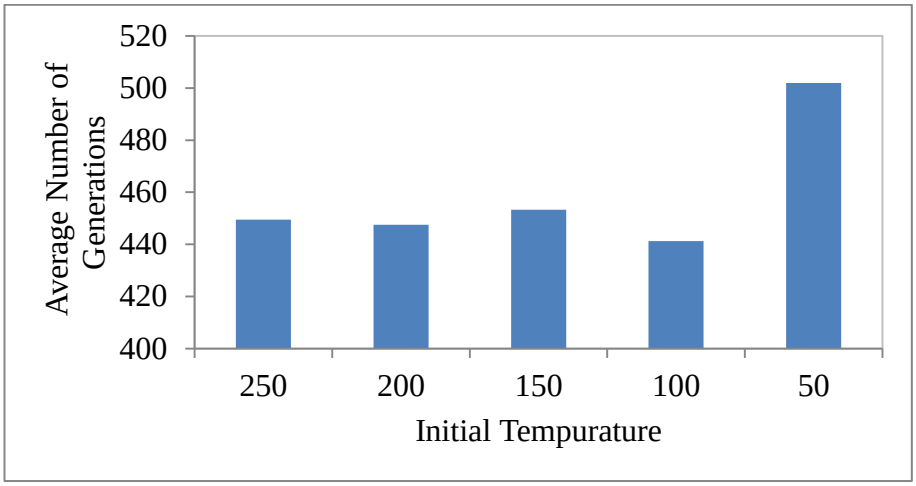


Figure 4.33. Average number of generations of initial temperatures.

According to Figure 4.33 the Average number of generations is the same for initial temperatures equal to 250,200,150 and 100 degrees and 50 degree has the highest amount of average number of generations. Therefore selecting 50 as initial temperature is not logical. Figure 4.34 exhibits the number of unsuccessful attempts of SA with different initial temperatures. It can be seen that when the initial temperature equals 100, the unsuccessful attempts has the less value and SA has the highest performance. Initial temperature equals 50 has the worst effect on the performance of SA. Therefore by taking all above mentioned reasons, 100 selected as the initial temperature for the SA.

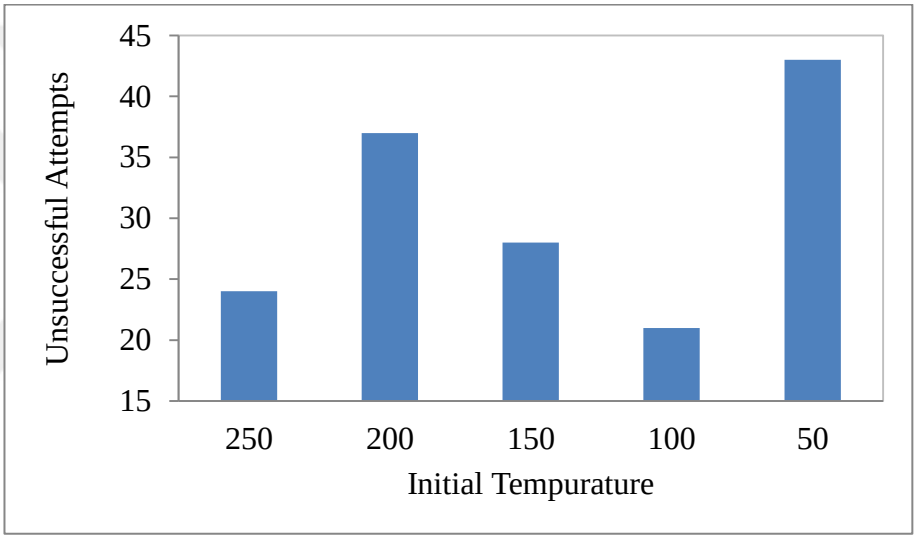


Figure 4.34. Unsuccessful attempts of SA with different initial temperatures.

4.5.2.2. Number of Mutation Points (NMP). Another important parameter of SA is the NMP. NMP defines the number of genes which is replaced by a random number in every generation. 1, 2, 3 and 4 are selected as NMP values. Figure 4.35 illustrates that NMP equals 2 has the smallest value of average generation. Therefore 2 selected as NMP for SA.

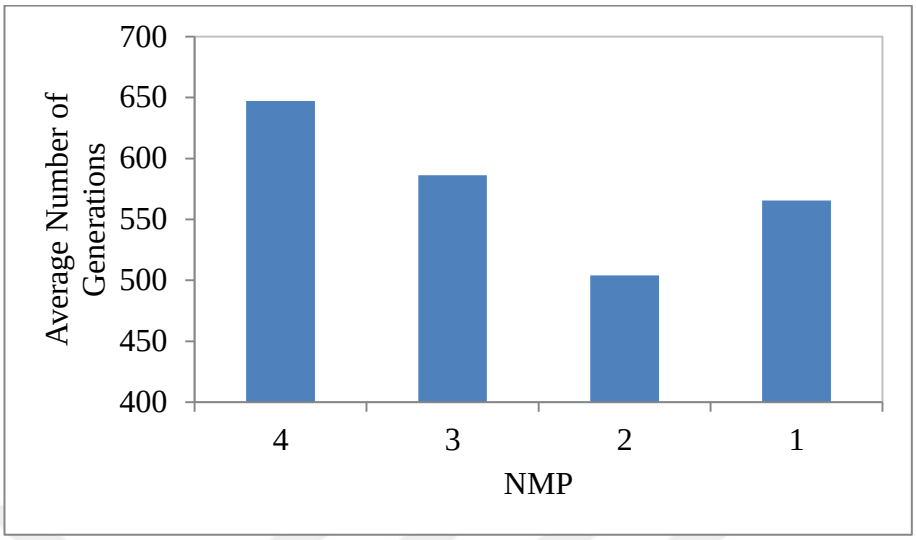


Figure 4.35. Average number of generations of different NMP values.

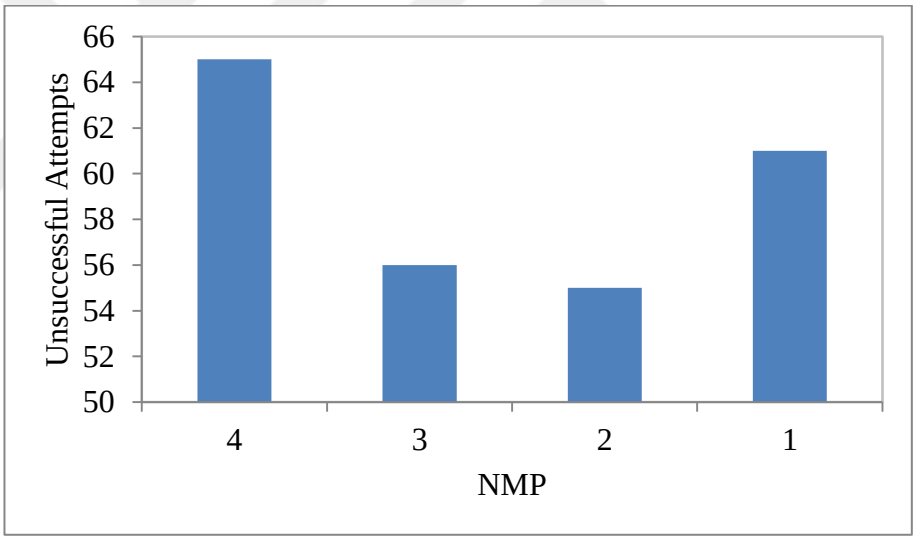


Figure 4.36. Unsuccessful attempts of SA with different NMP values.

4.5.3. Hill Climbing (HC)

Figure 4.37 illustrates samples of four search processes of HC. Every run has 12 steps. As it can be seen the HC get caught in local optima in the second attempt. In the Fourth attempt, the objective value falls from 31179 to 8180. It is obvious that in iterations the HC tries to reduce the objective value. In some attempts it gets caught in local optima. Figure 4.38 shows the objective value of 1000 Generations. It is obvious that most of reached objective values are between 10000 and 8000 seconds. The average objective

value and standard deviation of this data set are equal to 9190 and 1371 respectively. It shows that HC has good performance to find suitable solutions in this study.

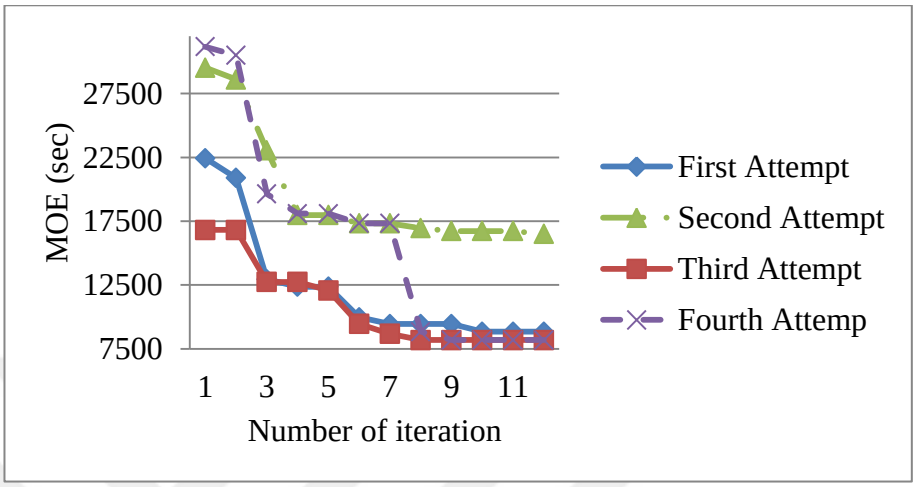


Figure 4.37. Sample illustration of four HC's search processes.

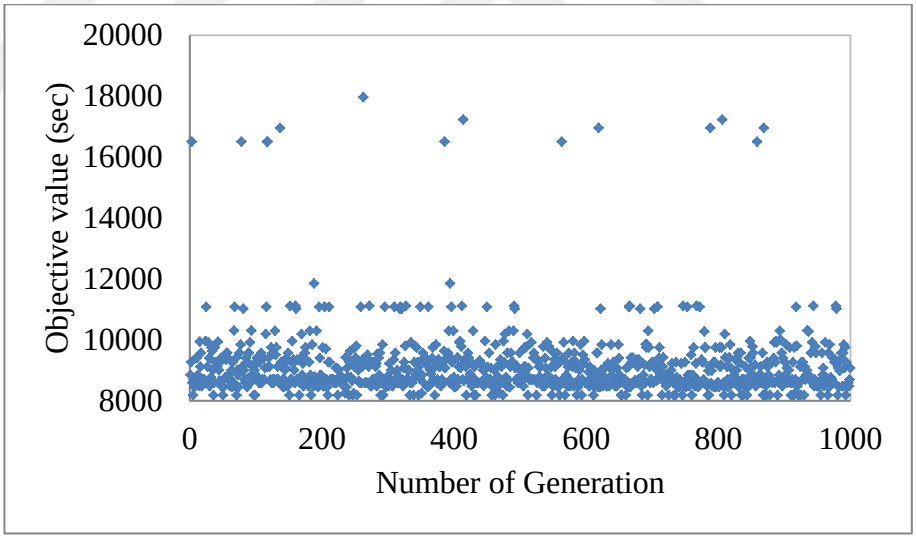


Figure 4.38. Objective values of 1000 generations of HC.

4.6. Results of Comparison of the Performances of the Heuristic Algorithms

Comparison has been done between three different heuristic algorithms and the results illustrated in this section. Each heuristic restart 2000 times. The first termination criterion was reaching objective value equal or less than 8500, the seconds criterion was reaching 12th, 2000th and 2000th generations for HC, SA and GA respectively. After performing 2000 runs for each heuristic, average minimum objective value, average CT of

runs, number of successful attempts and unsuccessful attempts evaluated and described in figures below. Firstly the performance of HC evaluated. Figure 4.39 illustrates the minimum objective values. It is obvious that most of objective values are about 9000 seconds. Small numbers of objective values are above 12000 seconds and these indicate good performance of HC. Table 4.4 shows the descriptive statistics of HC.

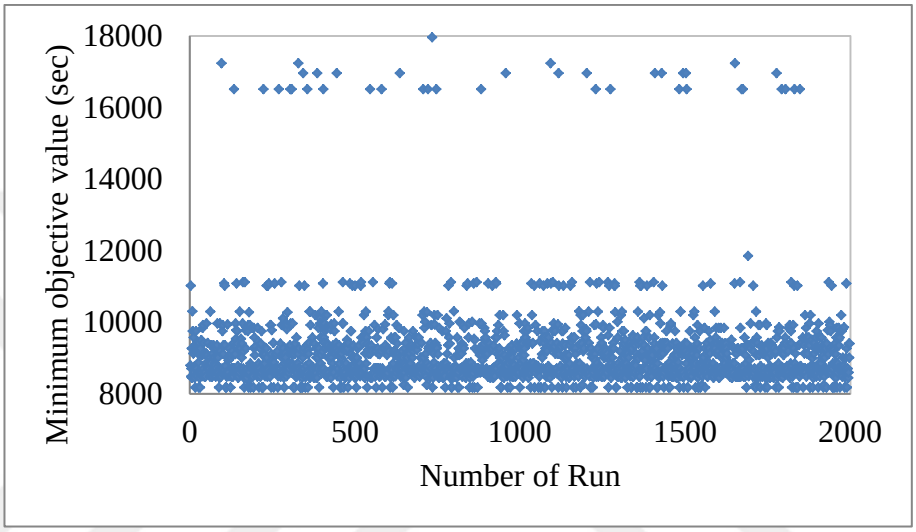


Figure 4.39. Minimum objective values of HC.

Table 4.4. Descriptive statistics of 2000 runs of HC.

Description	Value
Average minimum objective value	9167 sec
Standard deviation of minimum objective value	1265
Average CT of runs	0.055 sec
Standard deviation of CT of runs	0.007
Successful attempts	388
Successful attempts ratio	388/2000=0.19
Unsuccessful attempts	1612
Unsuccessful attempts ratio	0.8
Number of exact solution determining	129

Secondly the performance of SA is examined. The number of runs equals 2000. Similar to the HC, different parameters are evaluated and described in Figure 4.40 and Table 4.5.

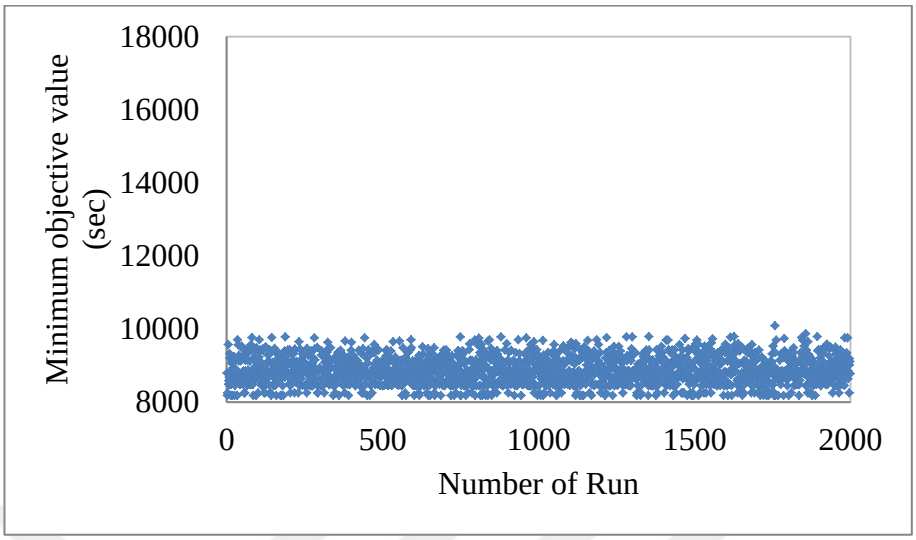


Figure 4.40. Minimum objective values of SA.

According to Figure 4.40, minimum objective values of the SA are about 8980 seconds and these values indicate good performance of SA in determining suitable solutions from solution space. Table 4.5 shows that SA's Computational time is higher than HC but, the average minimum objective value of SA is smaller than HC's average minimum objective value.

Table 4.5. Descriptive statistics of 2000 runs of SA.

Description	value
Average minimum objective value	8855 sec
Standard deviation of minimum objective value	370
Average CT of runs	0.76 sec
Standard deviation of CT of runs	0.24
Successful attempts	440
Successful attempts ratio	440/2000=0.22
Unsuccessful attempts	1560
Unsuccessful attempts ratio	0.78
Number of exact solution determining	79

Thirdly the performance of GA evaluated. The number of runs equals 2000. Termination criteria were similar to SA. The descriptive statistics of the GA are described in the Table 4.6.and Figure 4.41.

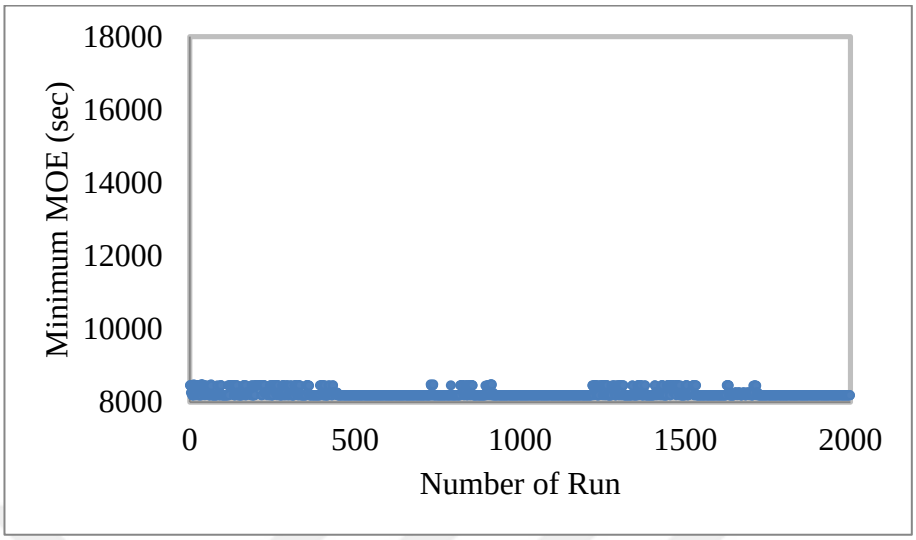


Figure 4.41. GA's minimum objective value of 2000 runs.

According to Figure 4.41, GA can find suitable solutions from solution space. Table 4.6 demonstrates that successful attempts ratio of GA is 100. This means that GA never fail to determine the global optima and have an outstanding performance in these kinds of solution spaces, but the computational time of GA is higher than other heuristic algorithms.

Table 4.6. Descriptive statistics of 2000 runs of GA.

Description	value
Average minimum objective value	8231 sec
Standard deviation of minimum objective value	98
Average CT of runs	1.5 sec
Standard deviation of CT of runs	1.3
Successful attempts	2000
Successful attempts ratio	100
Unsuccessful attempts	0
Number of exact solution determining	1482

As reported in Table 4.7, the HC's standard deviation and minimum objective values are higher than other heuristic algorithms. The GA's successful attempts equal 2000. Therefor GA reached the global optima in every run and HC is the faster heuristic among evaluated heuristics.

Table 4.7. Descriptive statistics of 2000 runs of HC,SA and GA.

Description	Heuristics		
	HC	SA	GA
Average minimum objective value	9167 sec	8855 sec	8231 sec
Standard deviation of minimum objective value	1265	370	98
Average CT of runs	0.055 sec	0.76 sec	1.5 sec
Standard deviation of CT of runs	0.007	0.24	1.3
Successful attempts	388	440	2000
Successful attempts ratio	0.19	0.22	1
Unsuccessful attempts	1612	1560	0
Unsuccessful attempts ratio	0.71	0.78	0
Number of exact solution determining	129	79	1482

5. CONCLUSIONS

In this study, a microscopic traffic simulator is developed that is based on a car following theory Intelligent Driver Model (IDM) selected as car following theory for this study. Then, a solution space is created and performance of three different heuristic for determining the best traffic signal timing plan is evaluated. Conclusions are summarized as follows:

i. According to Table 4.3, the best and worst measure of effectiveness equals 8180 and 39201 seconds, respectively. Thus, there is great difference between the best and worst traffic signal timing plan's Measure of Effectiveness. According to Traffic Signal Manual (2008), good traffic signal timing plan contributes to the safe and efficient movement transportation system.

ii. Using microscopic simulators can be favorable because microscopic simulators can evaluate and analyse traffic flow in much more detail. Traffic Signal Manual (2008) describes that microscopic simulation models are an advanced evaluation tool of a proposed signal timing plan because these models can directly evaluate the effects of interactions between intersections or the effects of oversaturated conditions

iii. Among the tested heuristic algorithms, Genetic Algorithm provided the best traffic signal timing plan for proposed corridor. According to Table 4.7, the Genetic Algorithm's successful attempts ration equals 1. Thompson and Bilbro (2000) also concluded that the average GA solution cost less than the average SA's solution for the problem of optimizing the topological design of a network. Moreover, Lee (1998) demonstrated that GA performs better than SA in most of demand scenarios. Besides, Taale and Van Zuylen (2003) reported promising results from applying a GA to optimize green times within a NDP context. Teklu et al. (2007) determined that GA-based method has given promising results in determining optimum signal timings with stable flows.

iv. Simulated Annealing (SA) algorithm shows a moderate performance for determining the best traffic signal timing plan. This heuristic fails to determine the exact solution

in most of its attempts. SA's unsuccessful attempts rate is about 78 %. Hence, SA is not a reliable heuristic to find the exact solution of the specific problem modeled and tested in this study. Lee (1998) used GA and SA for study the combined control and traffic assignment problem. He determined that GA performs better than SA in most of demand scenarios. In another study, Gopalakrishnan and Sooda (2010) utilized Simulated Annealing and Genetic Algorithm for solving the path selection problem from a known sender to the receiver and their results show that, by using Genetic Algorithm, the probability of shortest path convergence is higher than Simulated Annealing.

v. According to Table 4-7, Hill Climbing's unsuccessful attempts rate is 71%. Therefore, HC has slightly better performance than SA in detecting the best solutions. Similar to SA, it is not a reliable heuristic for determining the best solutions from solution space. Patrick (2012) defined that Hill Climbing always gets caught in a local optima because downward or upward moves are not allowed.

vi. HC is the fastest heuristic among tested algorithms. The results show that the HC's algorithm works 28 and 14 times faster than GA's and SA's algorithms respectively. Hence, HC can be used for determining local optima in a short time.

vii. GA's algorithm works slowly when it is compared to HC. According to Table 4-8, the GA's computational time equals 1.5 seconds that is 28 times greater than the HC's computational time.

viii. SA algorithm can determine better results than HC because the Average minimum objective value of SA equals 8855 that is 500 seconds less than HC's Average minimum objective value.

ix. GA has better performance when the mutation probability (MP) is equal to 0.05. MP equals 0.01 have the worst effect on the performance of the GA.

x. The Number of parent chromosomes (NPC), has a great effect on the performance of the GA. According to Figure 4.24, when the NPC equals 4 and 10 the number of unsuccessful attempts of 1000 runs, equals 269 and 0 respectively. This result shows that NPC can affect the performance of GA greatly.

xi. According to Figure 4.20, Figure 4.22 and Figure 4.19, average number of generations, number of unsuccessful attempts and minimum value of duration of stopped vehicles are indicators of GA's performance. According to mentioned figures, the performance of GA is insensitive to number of chromosomes greater than 10. In another hand, number of parent chromosomes smaller than 10 decreases the performance of the GA. Therefore setting the number of parent chromosomes to 10, increases the performance of the GA. Roeva at al., (2013) performed an investigation of the influence of the population size on the Genetic Algorithm (GA). Their results illustrated the same results. Their conclusions defined that the optimal population size of their problem was 100 chromosomes and increasing the number of parent chromosomes did not improve performance of GA and only increase the computational time.

xii. As illustrated in Figure 4.35 and Figure 4.36 Number of mutation points (NMP) affects the performance of SA. SA has better performance when NMP equals 2 than when NMP equals 1, 3 or 4.

xiii. According to Figure 4.27 Alpha value affects the performance of SA. The results show that by increasing the value of alpha from 0.75to 0.98 , the SA can find better traffic signal timing plans with less stopped vehicle duration. This action increases the performance of SA about 20%. (Aarts et.al, 2005) defined that typical α is between 0.8 and 0.99 and selected alpha for this study is equals 0.98.

xiv. Figure 4.33 and 4.34exhibit that initial temperature has direct effect on the performance of the SA. Initial temperature equals 100 has the smaller average number of generations and unsuccessful attempts rate. Figures show that performance of SA is insensitive to initial temperatures greater than 100. In another hand initial temperature smaller than 100 decreases the performance of the SA for determining suitable traffic signal time plans. Therefore the initial temperature of the SA set to 100 for this particular problem.

xv. The results of this study show that GA finds the exact solution from solution space but, GA requires more computational time. Thus, in the future, using Multi thread

programming and execution model which enables the computer to use all existed CPU cores is recommended.

REFERENCES

Aarts, E., Korst, J., Michiels, W., Search Methodologies, Springer, US, 2005.

Abuiziah, I., Shakarneh, N," A Review of Genetic Algorithm Optimization: Operations and Applications to Water Pipeline Systems", International Science Index, Vol. 7, No: 12, pp.1782-1788, 2013.

Abulebdeh, G., Benekohal, R. F., "Design and Evaluation of Dynamic Traffic Management Strategies for Congested Conditions", Transportation Research Part A, Vol.37, Issue. 2, pp. 109–27, 2003.

Balling,R..J., "Pareto Sets in Decision-Based Design", Journal of Engineering Valuation and Cost Analysis, Vol.3, pp.189–98, 2000.

Bertsimas,D., Tsitsiklis,.J., "Simulated Annealing", Journal of The Institute of Mathematical Statistics, Vol. 8, No. 1 ,pp. 10-15,1993.

Bielli, M., Caramia, M., Carotenuto, P., "Genetic Algorithms in Bus Network Optimization", Transportation Research Part C, Vol.10, Issue.1, pp. 19–34, 2002.

Burghout, W., Koutsopoulos, H.N., Andreasson, I.,"A Discrete-Event Mesoscopic Traffic Simulation Model for Hybrid Traffic Simulation", Intelligent Transportation Systems Conference, ITSC '06, IEEE, Vol. 1102, No. 1107, pp., 17-20, Sept. 2006.

Burval, B., Olegard, J., " A Comparison of a Genetic Algorithm and Simulated Annealing Applied to a Traffic Light Control Problem-A Traffic Intersection Optimization Problem", Level1, School of Computer Science, Royal Institute of Technology, Sweden 2015.

Černý, V. "Thermodynamical Approach to the Traveling Salesman Problem an Efficient Simulation Algorithm", Journal of Optimization Theory and Applications, Vol.45, No. l, 1985.

Chakroborty, P. (2003), “Genetic Algorithms for Optimal Urban Transit Network Design”, Computer-Aided Civil and Infrastructure Engineering, Vol.18(3), pp.184–200, 2003.

Chakroborty, P., Deb, K., Srinivas, B. (1998), “Network-Wide Optimal Scheduling of Transit Systems Using Genetic Algorithms”, Computer-Aided Civil and Infrastructure Engineering, Vol.13 (5), pp. 363–76, 1998.

Charypar, D., Nagel, K., “Generating Complete All-Day Activity Plans with Genetic Algorithms”, Transportation, Vol.32 (4), pp.369–97, 2005.

Chaudhary, N.A, Kovvali, V.G., Mahabubul Alam, S.M.,n, Guidelines for Selecting Signal Timing Software, Texas Transportation Institute the Texas A&M University System College Station- FHWA, September 2002.

Daganzo, C.F., “The Cell Transmission Model: a Dynamic Representation of Highway Traffic Consistent with The Hydrodynamic Theory”, Transportation Research Part B, Vol. 28, Issue.4, pp. 269–287, August 1994.

Drake,J.S., Schofer, J.L., May, A.D., “a Statistical Analysis of Speed–Density Hypotheses”, 45th Annual Meeting of Transportation Research Board, Vol. 154, pp.112–117, Washington DC, 1967.

Federal Highway Administration (FHWA), 2013, “A Sampling of Emissions Analysis Techniques for Transportation Control Measures- Traffic Simulation Models”, https://fhwa.dot.gov/environment/air_quality/conformity/research/transportation_control_measures, Accessed at November 2015.

Federal Highway Administration (FHWA), Highway Safety Information System (HSIS), California database, FHWA, USA 1994-1998.

Federal Highway Administration (FHWA), Manual of Uniform Traffic Control Devices (MUTCD), FHWA, Washington, DC: USDOT, 2003.

Federal highway administration (FHWA), Traffic Signal Timing Manual, FHWA-HOP-08-024, 2008.

Forbes,T.W.,"Human Factor Consideration in Traffic Flow Theory", highway research board, Issue. 15, pp. 60-66, 1963.

Forbes, T.W., Simpson, M.E., "Driver and Vehicle Response in Freeway Deceleration Waves", transportation science, Vol.2, No.1, pp. 77-104, 1958.

Forbes,T.W., Zagorski,H.J., Holshouser,E.L., Deterline,W.A., "Measurement of Driver Reactions to Tunnel Conditions" , highway research board, Vol.37, pp. 345-357, , 1958.

Foy, M., Benekohal, R. F. & Goldberg, D. E.," Signal Timing Determination Using Genetic Algorithms", Transportation Research Record, Vol.1365, pp.108–15, 1992.

Gazis,D.C., Herman,R., Rothery,R.W., "Non-linear Follow-the-Leader Models of Traffic Flow ", Operational Research, Vol.9 , No.4, pp. 545-567,1961.

Gentile G., Tiddi D., "Synchronization of Traffic Signals Through a Heuristic-Modified Genetic Algorithm with GLTM", XIII Meeting of the Euro Working Group on Transportation, Padua, Italy, pp.1-9, 2009.

Greenshields, B.D., "A Study of Traffic Capacity", 14th Annual Meeting of the Highway Research Board, pp. 448–477 1935.

Greenshields, B.D., "The Photographic Method of Studying Traffic Behavior", 13th Annual Meeting of the Highway Research Board, pp. 382–399, 1934.

Glover, F., Handbook of Metaheuristic algorithms, Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2003.

Harman,M., McMinn,P., “A Theoretical & Empirical Analysis of Evolutionary Testing and Hill Climbing for Structural Test Data Generation” , International Symposium on Software Testing and Analysis, ISSTA 07, New York, pp. 73-83 , 2007.

He, Q., Head, K.L., Ding, J.,” Heuristic Algorithm for Priority Traffic Signal Control”, Transportation Research Record: Journal of the Transportation Research Board, No. 2259, pp. 1–7, D.C., 2011.

Henderson, D., Jacobson, S., H., Johnson, A.W., Handbook of Metaheuristic algorithms, Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2003.

Holland, J., H., Adaptation in Natural and Artificial Systems, MIT press, MA, April 1992.

Hunt, P. B., Robertson, D. I., Bretherton, R. D., Winton, R. I., SCOOT—A traffic Responsive Method of Coordinating Signals, Transport and Road Research Laboratory -TRRL, Berkshire , England ,1981.

Institution of Highways & Transportation (IHT), Transport in the Urban Environment. Institution of Highways & Transportation, London, 1997.

Kirkpatrick1, S., Gelatt , C. D. Jr., Vecchi, M. P., “Optimization by Simulated Annealing”, Science, Vol. 220, No. 4598 pp. 671-680, 13 May 1983.

Kometani,E., T. Suzaki,T., “On the Stability of TrafficFlow”, Japan Operations Research, Japan Vol.2, pp. 60-79, 1958.

Lee, C. (1998), Combined Traffic Signal Control and Traffic Assignment: Algorithms, Implementation and Numerical Results, Ph.D. dissertation, the University of Texas at Austin, Texas, 1998.

Lee, J., Abdulhaia, B., Shalabya, A., Chunga, E.H.," Real-Time Optimization for Adaptive Traffic Signal Control Using Genetic Algorithms", Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, Vol. 9, Issue 3, pp. 111-122, 2005.

Lo, H. K., Chang, E. & Chan, Y. C., "Dynamic Network Traffic Control", Transportation Research Part A, Vol. 35(8), pp.721–44, 2001.

Maciejewski,M., "A Comparison of Microscopic Traffic Flow Simulation Systems for an Urban Area", Transport Problems : an International Scientific Journal, Vol.4, Issue 4, pp. 28-38 , 2010.

May, A. D., Traffic Flow Fundamentals, Prentice Hall, NJ, 1990.

Mukhopadhyay, D.M., Balitanas, M.O., Farkhod, A., Jeon, S.H., Bhattacharyya, D.," Genetic Algorithm: A Tutorial Review", International Journal of Grid and Distributed Computing, Vol.2, No.3, Sep. 2009.

Nair, G.T.R., Sooda, K.., "Comparison of Genetic Algorithm and Simulated Annealing Technique for Optimal Path Selection In Network Routing", NCVN-09 9thand 10thoctober 2009 KCG College of Technology Cornell University, 22 Jan 2010.

Park, B., Messer, C. J., Urbanik, T., "Enhanced Genetic Algorithm for signal timing optimization of oversaturated intersections", Transportation Research Record, Vol.1727, pp.32–41, 2000.

Park, B.B., Schneeberger, J.D., Evaluation of Traffic Signal Timing Optimization Methods Using A Stochastic and Microscopic Simulation Program, Virginia Transportation Research Council-Virginia Department of Transportation, 2002.

Park, B., Schneeberger, J. D., " Microscopic Simulation Model Calibration and Validation: A Case Study of VISSIM for a Coordinated Actuated Signal System", Transportation Research Record: Journal of the Transportation Research Board, Vol. 1856, pp. 185-192, Washington, D.C., 2003.

Park, B.,Yun, I., Stochastic Optimization Method for Coordinated Actuated Signal Systems, , Center for Transportation Studies, University of Virginia, USA , 2005.

Kenekayoro, P., “Comparison of Simulated Annealing and Hill Climbing in the Course Timetabling Problem”, African Journal of Mathematics and Computer Science Research, Vol. 5(11), pp. 176-178, Sep.2012.

Pipes, L.A.,” an Operational Analysis of Traffic Dynamics”, Journal of Applied Physics, Vol. 24, No.3, pp. 274-287, 1953.

Pursula, M., “Simulation of Traffic Systems - An Overview”, Journal of Geographic Information and Decision Analysis, Vol.3, No.1, pp. 1-8, 1999.

Reeves, Handbook of Metaheuristic algorithms, Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2003.

Robertson, D. I., “TRANSYT Method for Area Traffic Control”, Traffic Engineering and Control, Vol.10, pp.276–81, 1969.

Roeva, O., Fidanova, S., Paprzycki, M., "Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling", in Computer Science and Information Systems (FedCSIS), 2013 Federated Conference , pp.371-376, 8-11 Sept. 2013.

Romanycia, M.H.J., Pelletier, F.J.,” What is a heuristic? ”, Computational Intelligence, Vol. 1, Issue 1, pp 47–58, Jan. 1985.

Ross, P., Corne, D.,” Comparing Genetic Algorithms, Simulated Annealing, and Stochastic Hillclimbing on Timetabling Problems”, Evolutionary Computing AISB Workshop, Sheffield, UK, April 3-4, 1995.

Selman, B., Gomes, C.P., Encyclopedia of Cognitive Science, Wiley, Published Online: 15 JAN 2006.

Sharma, S., Lingras, P. & Zhong, M., “Effect of Missing Values Estimations on Traffic Parameters”, Transportation Planning and Technology, Vol.27 (2), pp.119–44, 2004.

Smulders, S., “Control of Freeway Traffic Flow by Variable Speed Signs”, Transportation Research Part B: Methodological, Vol. 24(2), pp 111–132, 1990.

Srinivasan, D., Cheu, R. L., Poh, Y. P., Ng, A. K. C., “Development of an Intelligent Technique for Traffic Network Incident Detection”, Engineering Applications of Artificial Intelligence, Vol.13 (3), pp.311–22, 2000.

Sultani, W., Cho, Y., ”Abnormal Traffic Detection using Intelligent Driver Model”, International Conference on Pattern Recognition (ICPR 2010),IEEE,Istanbul, pp.23-26 , August 2010.

Taale, H., Van Zuylen, H. J., “The Effects of Anticipatory Traffic Control for Several Small Networks” 82nd Transportation Research Board Annual Meeting, Washington, DC, January 12–16, 2003.

Teklu, F., Sumalee, A., Watling, D., “A Genetic Algorithm Approach for Optimizing Traffic Control Signals Considering Routing”, Computer-Aided Civil and Infrastructure Engineering, Vol.22, pp.31–43, 2007.

Thompson, D.R., Bilbro, G.L., “Comparison of a Genetic Algorithm with a Simulated Annealing Algorithm for the Design of an ATM Network”, Communications Letters, IEEE, Vol. 4, Issue 8 , pp.267 – 269 , 2000.

Torbagan, M.K., Kazemi, S.M., Zhiani, R., Hamed, F.” Improved Hill Climbing and Simulated Annealing Algorithms for Size Optimization of Trusses”, International Science Index, Vol. 7, No: 2, 2013.

Treiber, M., Hennecke, A., Helbing, D.,” Congested Traffic States in Empirical Observations and Microscopic Simulations”, PHYSICAL REVIEW E, Vol. 62, No.2, pp 1-17, 2000.

Treiber, M., Kesting, A., Traffic Flow Dynamics, Springer Berlin Heidelberg, Prentice Hall, NY, 2013.

Turing, A.,” Computing Machinery and Intelligence”, Mind, Vol.236, pp. 433-460, 1950.

USDOT, Summary of Travel Trends-2001 National Household Travel Survey, U.S. Department of Transportation, Federal Highway Administration, Washington, D.C. 2004.

Wageningen-Kessels, F.V., Lint, H.V., Vuik, K., Hoogendoorn, S., " Genealogy of Traffic Flow Models ", EURO Journal on Transportation and Logistics, Vol. 4, Issue 4, pp 445-473, Dec.2015.

Vishnu Raja, P., Murali Bhaskaran, V.," Improving the Performance of Genetic Algorithm by Reducing the Population Size", International Journal of Emerging Technology and Advanced Engineering, Vol. 3, Issue 8, Aug. 2013.

Webster, F. V., Traffic Signal Settings, Road Research Technical Paper No. 39, HMSO, London, 1958.

APPENDIX: SMALL PORTION OF GENERATED CODES IN MATLAB

```
clear all;
close all;
clc;


  simulationmatrix = xlsread( 'F:\desctop\ce\master thesis\exel\1-50-
6.xlsx');
  objectivevaluee=simulationmatrix(:,7);
  simulationmatrix(:,7)=[];
  ff=2000;

difference=zeros(1,ff+1);
Acceptance_probabilities=zeros(1,ff+1);
Random_number_0_or_1=zeros(1,ff+1);
t=cputime;
%temp=[500 250 100 50 25 19 16 13 10 8 6 4 2 1];
%% pre calculations
karch=1;
BUMBEROFmp=4;
for  toto=1:1000
    tic
numberoflights=3;
dT=1;
lightsequence=zeros(1,numberoflights*3);
temperature(1,1) =100; temperature(1,2) = 100;
% define minimum tempurature for simulated annealing
temperature_min =
0.00000000000000000000000000000000000000000000000000000000000000000000
000001;
alpha = 0.98;
e=2.718281;
fprintf('toto');
% start the simulated annealing process

  %% start of simulated

%% pre calculations

Acceptance_probabilities(1,1)=0;

    Tikan=0;

for f=1:ff
    fprintf(' %d', toto);
if f==1
lightsequence(1,1:numberoflights)=(randperm(5,numberoflights)*10)/dT;
lightsequence(1,numberoflights+1:numberoflights*2)=(randperm(5,numberofli
ghts)*10)/dT;
lightsequence(1,numberoflights*2+1:numberoflights*3)=(randperm(1,numberof
lights-2)*10)/dT;
end
```

```
      % define the neighbor solution of initial random solution
if f==2
lightsequence(2,:)=lightsequence(1,:);
random_number=randperm(6,2);
lightsequence(2,random_number(1,1))=lightsequence(1,random_number(1,2));
lightsequence(2,random_number(1,2))=lightsequence(1,random_number(1,1));
%lightsequence(2,random_number(1,3))=lightsequence(1,random_number(1,4));
%lightsequence(2,random_number(1,4))=lightsequence(1,random_number(1,3));
end
  %% start of simulated
 if Tikan==0;
 for tr=1:1
    [~,xxxx(f,:)]=ismember(lightsequence(f,1:6),simulationmatrix,'rows');
    totalobjectivevalue(f,1)=objectivevaluee(xxxx(f,:));
end
  end
if f>=2
  % find the diffrence between first and second solutions objective

difference(1,f)=totalobjectivevalue(f,1)-totalobjectivevalue(f-1,1);
if difference(1,f) <= 0
Acceptance_probabilities(1,f)=0;
elseif difference(1,f) > 0
Acceptance_probabilities(1,f)=e.^((-difference(1,f))/temperature(1,f));
Random_number_0_or_1(1,f-1)=round(rand);
end

if difference(1,f) <= 0
lightsequence(f+1,:)=lightsequence(f,:);

tarzan=randi([1 5],1,BUMBEROFmp)*10;
tarzan2=randperm(6,BUMBEROFmp);
lightsequence(f+1,tarzan2(1))=tarzan(1);
lightsequence(f+1,tarzan2(2))=tarzan(2);
lightsequence(f+1,tarzan2(3))=tarzan(3);
lightsequence(f+1,tarzan2(4))=tarzan(4);

Tikan=0;
elseif difference(1,f) > 0 && Acceptance_probabilities(1,f) >
Random_number_0_or_1(1,f-1)
lightsequence(f+1,:)=lightsequence(f,:);

tarzan=randi([1 5],1,BUMBEROFmp)*10;
tarzan2=randperm(6,BUMBEROFmp);
lightsequence(f+1,tarzan2(1))=tarzan(1);
lightsequence(f+1,tarzan2(2))=tarzan(2);
lightsequence(f+1,tarzan2(3))=tarzan(3);
lightsequence(f+1,tarzan2(4))=tarzan(4);

Tikan=0;
elseif difference(1,f) > 0
lightsequence(f+1,:)=lightsequence(f-1,:);
totalobjectivevalue(f+1,1)=totalobjectivevalue(f-1,1);
tarzan=randi([1 5],1,BUMBEROFmp)*10;
tarzan2=randperm(6,BUMBEROFmp);
lightsequence(f+1,tarzan2(1))=tarzan(1);
lightsequence(f+1,tarzan2(2))=tarzan(2);
lightsequence(f+1,tarzan2(3))=tarzan(3);
```

```
lightsequence(f+1,tarzan2(4))=tarzan(4);

Tikan=1;
end
end

 % decrease the tempurature
temperature(:,f+1) = temperature(:,f)*alpha;
%if mod(f,100)==0
 % temperature(:,f+1)=3;
%end
% if tempurature reach the minimum value ,the loop will
% be terminated
%if temperature(:,f+1)<temperature_min
 % break
if totalobjectivevalue(f,1)<=9200
    break
end
end

toc
ee(toto,1) = toc;
toko(toto,1)=min(totalobjectivevalue(:,1));
sizee(toto,1)=size(totalobjectivevalue,1);
ratee(toto,1)=ee(toto,1)/toko(toto,1);
%plot(totalobjectivevalue)
totalobjectivevalue=[];
xxxx=[];
lightsequence=[];
end

xlRange = 'E';
xlswrite('F:\desctop\sa2 results',ee(:,1),karch,xlRange)
xlRange = 'C';
xlswrite('F:\desctop\sa2 results',toko(:,1),karch,xlRange)
xlRange = 'D';
xlswrite('F:\desctop\sa2 results',sizee(:,1),karch,xlRange)
xlRange = 'F';
xlswrite('F:\desctop\sa2 results',ratee(:,1),karch,xlRange)
fprintf('\n');
%subplot(1,2,1)
%plot(temperature)
%title('temperature')
%subplot(1,2,2)
temperature=[];
Acceptance_probabilities=[];
difference=[];
ee=[];
toko=[];
sizee=[];
ratee=[];
Random_number_0_or_1=[];


plot(totalobjectivevalue)
title('totalobjectivevalue')
%% create sequence matrix for checking the feasibility of solutionst=10;
pas=1/10;
```