A NEW OFFSET OPTIMIZATION METHOD FOR SIGNALIZED URBAN ROAD NETWORKS

Von der Fakultät für

Bauingenieurwesen und Geodäsie

der Universität Hannover

zur Erlangung des Grades eines Doktors der

Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

von

M.Sc. Essam Almasri

geboren am 25.02.1974, in Gaza-Palästina

2006

Referent: Univ.-Prof. Dr.-Ing. Bernhard Friedrich, Uni Hannover
Koreferent: Univ.-Prof. Dr.-Ing. Thomas Siefer, Uni Hannover
Tag der Promotion: 18. April 2006

DEDICATION

To my Parents and my Wife

ACKNOWLEDGMENTS

My gratitude goes first and foremost to my God who has given me life, strength, and good health throughout this study.

I would like to express my sincere and heartfelt gratitude to my supervisor Professor Bernhard Friedrich, for his invaluable support and guidance through my doctoral program. I am very fortunate to have been able to work with Prof. Friedrich. My association with him has taught me a great deal about traffic operations, and, more importantly, about the thinking process for approaching research problems in traffic engineering. He recommended me to the Fellowship Program of German Academic Exchange Service DAAD, whom I thank for providing me with financial support during my doctoral research.

I also wish to express my gratitude to my other advisory committee members: Prof. Thomas Siefer and Prof. Udo Nackenhorst. Both have been very kind and helpful.

The software for this work used the C++ GAlib (genetic algorithm library), written by Matthew Wall at the Massachusetts Institute of Technology. I want to extend my gratitude and thanks to him.

My deepest gratitude and appreciation go to my parents for their prayers. Finally, I wish to express my heartfelt thanks to my wife, Heba. Without her love, understanding and constant support, I would have already given up my doctoral research long time before.

Kurzfassung

In dieser Dissertation wird ein neues Verfahren für Versatzzeitoptimierung für signalisierte städtische Straßennetze beschrieben. Das Verfahren ist fähig, die Versatzzeiten sowohl für ungesättigte als auch für gesättigte Verkehrszustände zu optimieren und quasi-optimale Resultate zu finden. Die Strategie wurde prototypisch in ein C++ Programm umgesetzt, das aus drei Modulen besteht: (1) die Eingabeeinheit; (2) das Optimierungsmodul, das zwei Lösungsansätze auf der Grundlage genetischer Algorithmen (GA) nutzt; und (3) ein Verkehrsanalysemodul, das als die Zielfunktion für den GA-basierten Optimierer dient.

Das Analysemodul, das auf dem Cell Transmission Model basiert, modelliert die räumliche sowie die zeitliche Bildung von Warteschlangen und kann deren Aufbau, Ausbreitung und Rückgang voraussagen. Die hohe Berechnungsgeschwindigkeit des Modells erlaubt auch bei wiederholten Simulationsläufen die online-Anwendung. Es dient auch als Wirkungsmodell für die Zielfunktion des Optimierers und wird mit bekannten kontinuierlichen Warteschlangenmodellen und den mit dem Simulationsmodell AIMSUN erzielten Ergebnissen verglichen.

Das GA-basierte Optimierungsmodul besteht aus zwei heuristischen Lösungsansätzen, die die Parameter (Versatzzeiten) für das Analysemodul generieren und dann diese Parameter optimieren. Im ersten Ansatz, der hier paralleler genetischer Algorithmus (PGA) genannt wird, wird gleichzeitig über alle Versatzzeiten gesucht, indem im Ablauf Reproduktion – Verschneidung – Mutation das gesamte Chromosom verändert werden kann. Im zweiten Ansatz, der hier serieller genetischer Algorithmus (SGA) genannt wird, wird jeweils nur eine Gruppe der Versatzzeiten und damit nur ein Bereich des Chromosoms verändert, bis die beste Lösung gefunden ist. In den nächsten Schritten werden die Versatzzeiten der folgenden Gruppe der Knotenpunkte optimiert und es wird eine Methode für die Ermittlung der Suchreihenfolge entwickelt, da die Suchreihenfolge einen großen Einfluss auf die Optimierungsresultate hat.

AIMSUN wird als der unverzerrte Schätzer für den Vergleich des entwickelten Verfahrens mit dem bekanten Optimierungsverfahren TRANSYT-7F, der vollständigen Enumeration der Variablen und mit einem aus der Ingenieurpraxis bekanten „Papier- und Bleistiftverfahren“ Dominanzverfahren verwendet. Anhand von drei Anwendungsbeispielen wird das entwickelte Verfahren bewertet. Diese Anwendungsbeispiele sind ein Streckenzug mit drei signalisierten Knotenpunkten, ein kleines realistisches Straßennetz mit sechs signalisierten Knotenpunkten und ein verhältnismäßig großes Rasternetz mit zwölf signalisierten Knotenpunkten. Die Vergleichsergebnisse für die ersten zwei Beispiele zeigen, dass die Verfahren PGA und SGA die optimale Lösung entsprechend der vollständigen Enumeration finden können, wobei das SGA die CPU-Zeit erheblich verringert. Für diese zwei Beispiele sind die Verfahren PGA und SGA besser als das Dominanzverfahren und TRANSYT-7F. Für das verhältnismäßig große Rasternetz, kann nur das Verfahren SGA ein quasi-optimales Ergebnis finden. Dabei ist es TRANSYT-7F (10 % Unterschied in Bezug auf die Wartezeit nach AIMSUN) und dem Dominanzverfahren (8 % Unterschied in Bezug auf die Wartezeit nach AIMSUN) weiterhin überlegen.

ABSTRACT

A new offset optimization method for signalized urban road networks is described in this dissertation. The method is capable of optimizing offsets in both undersaturated and oversaturated conditions and of finding quasi optimal results. The method is converted prototypically in a C++ program, which consists of three modules: (1) the input module; (2) the optimization module consisting of a genetic algorithm (GA) based optimizer; and (3) a traffic analysis module that serves as the fitness function for the GA-based optimizer.

The analysis module is based on Cell Transmission Model; models spatial as well as temporal formation; and is able to predict build-up, propagation and dissipation of queues. Furthermore, its calculation time for repeated simulation runs of module permits online application. The analysis module also serves as the fitness function for the signal timing optimizer and is validated against well-known queuing models and AIMSUN, a state of the art microscopic traffic simulation.

The GA-based optimization module consists of two search algorithms that generate timing parameters (offsets) for the analysis module and then optimize these parameters. The first algorithm performs a simultaneous search over all offsets by the process of variation of reproduction – crossover – mutation of the entire chromosome. The second routine varies a group of offset values and therefore only a part of the chromosome until the best solution is found. In the next step, the offsets of the next group of intersections are optimized. A method is developed for the determination of the search order, since the order of treating the intersection and searching the offset has great influence on the optimization results.

AIMSUN is used as the unbiased estimator for comparing the developed method against well-know control software TRANSYT-7F, the full enumeration of the offset variables and a manual method based on engineering practice called dominance method. The comparison is carried out for three case studies to benchmark the performance capabilities of the developed method against existing ones. These case studies are a two-way street with 3 intersections; a small

realistic network with 6 intersections; and a relatively large grid network with 12 intersections. The results obtained from comparing the first two case studies show that both PGA and SGA can find the optimal solution as well as the full enumeration, whereas the SGA could reduce the CPU-time significantly. For these two case studies, PGA and SGA are superior to both the dominance method and TRANSYT-7F. However, for the relatively large network case study, only SGA including the developed search determination can find a quasi optimal result and is still superior to both TRANSYT-7F (10% relative difference of AIMSUN delay) and dominance method (8 % relative difference of AIMSUN delay).

Schlagworte

Versatzzeitoptimierung, genetischer Algorithmus, Cell Transmission Model

Key words

Offset optimization, genetic algorithm, Cell Transmission Model

Table of Content

1 Introduction..1

1.1 Problem Statement 1

1.2 Research Objectives...3

1.3 Research Methodology 3

2 Background on Traffic Signal Control...7

2.1 Basic Definitions of Signal Control..7

2.2 Elements of a Timing Plan 11

2.2.1 Signal Phasing and Phase Sequence...12

2.2.2 Cycle Length 13

2.2.3 Green Splits 15

2.2.4 Offset 16

2.3 Methods of Signal Control..17

2.3.1 Fixed-Time Control 17

2.3.2 Vehicle Actuated Control 18

2.3.3 Traffic Responsive Control...18

2.3.4 Adaptive Control 19

2.4 Well-Known Control Software 19

2.4.1 Offline Control Strategy...19

2.4.2 Online Control Strategy...22

3 Background on Optimization Techniques...29

3.1 What Does Optimization Mean? ... 29

3.2 Types of Optimization Problems ... 30

3.2.1 Linear Programming Problem ... 30

3.2.2 Quadratic Programming Problem ... 31

3.2.3 Quadratic Constrains Problem ... 31

3.2.4 Mixed-Integer Programming (MIP) Problem ... 32

3.2.5 A Smooth Non-linear Programming -Problem ... 32

3.2.6 Global Optimization (GO) Problem ... 32

3.2.7 Non-Smooth Optimization (NSP) ... 33

3.3 Simple Genetic Algorithm ... 33

3.3.1 The Procedure ... 34

3.3.2 Advantages and Disadvantages of GA ... 35

3.3.3 GA Terminology and Parameters ... 36

4 An Offset Optimization Method for Signalized Urban Road Networks ... 43

4.1 Traffic Analysis Module ... 43

4.2 Optimization Algorithms Module ... 45

4.3 Input Module ... 46

4.3.1 Global Data ... 46

4.3.2 Geometric Data ... 47

4.3.3 Control Data ... 48

4.3.4 Traffic Loading ... 49

4.3.5 Genetic Algorithm Settings ... 49

5 Traffic Analysis Module Based on Cell Transmission Model ... 51

5.1 The Cell Transmission Model ... 51

5.2 Network Modeling and Topologies ... 52

5.3 Traffic Flow Calculation ... 53

5.3.1 Traffic Flow on Normal Link ... 53

5.3.2 Traffic Flow on Diverge Links ... 54

5.3.3 Traffic Flow on Merge Links ... 55

5.4 Traffic Signal Control and Delay Estimation....................56
5.4.1 Traffic Signal Control 56
5.4.2 Delay Estimation....................56
5.5 Cell Representation and Boundary Conditions 57
5.6 Calculation Procedure....................60

6 Validation and Comparison of the Traffic Model....................63

6.1 Delay Validation at an Isolated Intersection....................63
6.2 Delay Validation at a Two-Intersection Street....................66
6.3 Delay Validation at a Small Network 68

7 GA-based Optimization Module 75

7.1 Parallel Genetic Algorithm....................75
7.1.1 The Procedure 75
7.1.2 Encoding and Decoding of the Offset Values 77
7.1.3 Other GA Characteristics....................77
7.2 Serial Genetic Algorithm 78
7.2.1 The Procedure 78
7.2.2 A Method for Search Order for SGA 79

8 Method Comparisons....................89

8.1 Comparison Methods....................89
8.1.1 Full Enumeration....................89
8.1.2 Dominance Method....................91
8.1.3 TRANSYT-7F....................91
8.2 Statistical Testing....................92
8.3 Case Studies....................95
8.3.1 Hypothetical Arterial....................95
8.3.2 A Small Realistic 6-Intersection Network 100
8.3.3 Grid Network with 12 Intersections 107

9 Conclusions and Recommendations....................117

9.1 Conclusions 117

9.2 Recommendations 120

References 121

List of Figures 133

List of Tables 137

List of Abbreviation 139

Annex A 143

Annex B 147

Annex C 157

Annex D 161

1 Introduction

1.1 Problem Statement

Ever since Garrett Morgan invented the first electronic traffic signal in the United States in 1914 [4], there has been little doubt of the benefit to society of an efficient traffic-signal control system. Over the years, millions of dollars have been invested worldwide in the pursuit of ever more intelligent systems, and innovation has occurred in two broad fields:

- With the development of technology in the areas of control, computers and communications, traffic signal control systems have gradually developed from simple electromechanical controls to sophisticated microprocessor-based systems.
- In the belief that increased responsiveness to traffic leads to improved traffic performance, [45], many researchers have contributed to the rapid development of online traffic signal control, which has evolved from the First and Second, through to the Third Generation Control - Systems actually envisioned more than 35 years ago in the original Urban Traffic Control System (UTCS) Project [71].

Online traffic signal control in urban road networks requires information on the present and future traffic state. This information needs to be as complete and precise as possible. For this reason advanced control methods involve respective traffic models in order to complement local measurements and to derive criteria on the traffic state that cannot be measured. As criterion to be optimized some models calculate either the queue length or the delay. It is therefore the traffic flow model precision and its short term forecast of the mentioned criteria which is a crucial prerequisite for any online control strategy.

Equilibrium queuing models or discrete models can be employed for optimisation purposes in online traffic control methods. Whereas equilibrium models can only be applied to macroscopic adaptation of split and cycle time at single intersections, discrete models also allow for microscopic adaptation and in particular for offset representation. Discrete models are classified into microscopic models or macroscopic ones.

In microscopic simulation models of traffic, such as AIMSUN, Paramics, or Vissim, each vehicle is separately identified and tracked and has its own characteristics, e.g. desired speed, reaction time, destination, vehicle type and so on and produce a lot of output than is rarely needed. Besides the large input and output data, a second major problem connected with microscopic models is the difficulty of optimization due to the expensive time cost for many repeat runs.

Macroscopic models deal with traffic flow as if it were a fluid. They can be roughly divided into two classes, models for undersaturated traffic, which assume that traffic flows more or less at the design speed and that delays are given by classical formulas of queuing theory; or models for oversaturated traffic, where queues cannot be cleared totally. TRANSYT is the most widely used macroscopic offline program of the first type – recent version TRANSYT-7F 10 has been proposed to be used online– for determining optimum signal timings in any network of roads for which the average traffic flows are known. Another famous program of the first type is SCOOT, which is essentially an online TRANSYT. When optimising the signal settings, the TRANSYT/SCOOT flow model does not take into consideration whether any links are blocked or not. It is assumed that vehicles effectively form a vertical queue at the stop-line, rather than a linear queue back along a link, and the model can then release traffic into the upstream link. As a consequence, TRANSYT/SCOOT does not correctly take account of the physical length of the queue and the green starting wave. There are several signal-control models developed especially for oversaturated traffic conditions such as D'Ans and Gazis [19] , Eddelbuttel and Cremer. [26], Abu-Lebdeh and Benekohal. [1] and Wey and Jayakrishnan [105]. According to Lo [68], none of them –including TRANSYT and SCOOT- is intended for the entire range of the fundamental diagram (see Figure 1-1) derived from a flow-fluid analogy.

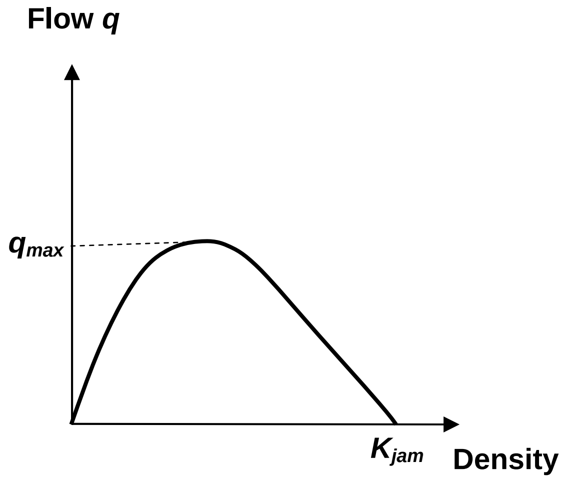


Figure 1-1: Fundamental diagram

Offset does not only need a special attention to its representation (it is represented only by discrete traffic model), but also to its optimization. When one has fixed phase sequences, cycle times and green times for n signalized intersections in a road network, then one can have a number of $ct^{(n-1)}$ combinations (ct is the duration of the cycle time) for the solution space of offsets optimization. For small networks, the optimal solution can be determined exactly with an already high computation time by complete enumeration. For larger networks and particularly for online use, this is no longer possible because of the required CPU time. In searching the literature for what has been achieved so far in terms of state-of-the-art traffic signal control, it has been reported that all adaptive control methods are unable to find the exact (or quasi) optimal variables due to the complexity of especially offset optimization and due to the method used in the optimization; e.g. TRANSYT used hill climbing as a search method.

1.2 Research Objectives

Given this context, the goal of this dissertation is to develop an offset optimization method for signalized urban road networks. This method has to be converted prototypically in a C++ program. It should be able to provide timely and near-optimal results and to coordinate signals in both undersaturated and oversaturated traffic conditions. In order to achieve this goal, the following set of objectives has to be achieved:

1. Develop a network-wide traffic analysis model that can accurately analyze traffic conditions in both undersaturated and oversaturated conditions. It must model the spatial as well as the temporal formation and be able to predict build-up, propagation and dissipation of queues. Furthermore, its calculation time for repeated simulation runs of module must permit online application.
2. Develop a signal timing optimizer for network. The search capabilities of the selected optimizer should be independent of the shape of the objective function being optimized; and
3. Demonstrate that the developed method performs as well as, if not better than, existing control strategies like TRANSYT-7F [51], dominance method [90]. The near-optimal results provided by the developed method should be comparable to the full enumeration of the control variables.

1.3 Research Methodology

The methodology represented in Figure 1-2 starts with a literature review of the different methods of traffic signal control and existing online and offline control software. The strengths and weaknesses of the existing software are obtained from the literature. Literature is also reviewed on the optimization techniques, especially on genetic algorithm (GA), since this technique is independent of the objective function being optimized.

The core of the work begins with the decision on which traffic model has the required characteristics. The cell transmission model (CTM) originally developed by Daganzo [16] and [17] is selected to be the basis of the simulator. Based on the CTM, a delay equation is first derived and then used as an objective function for the optimization. A C++ program is developed for the delay calculation.

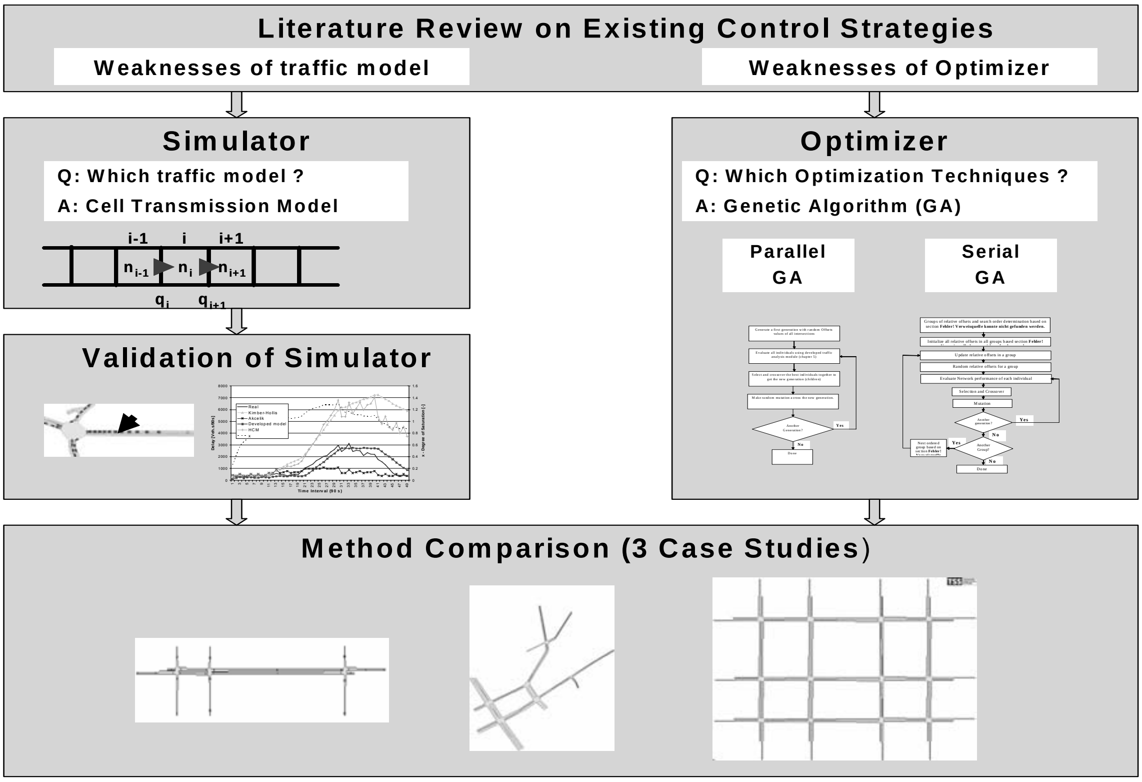


Figure 1-2: Methodology for achieving the research objectives

After developing the simulator, it should be validated. A virtual environment is prepared for that purpose using AIMSUN, which is used as a reference measurement. The validation is carried out first at a realistic isolated intersection (found in Hanover, Germany) with realistic data. At this intersection, the delay is estimated at one link using the developed estimator and compared against some well-known equilibrium queuing models using AIMSUN delay as the unbiased estimator. Secondly, the validation is conducted at a one-way street with two intersections to check delay estimation when varying the offsets at these intersections. Furthermore, the simulator is tested at a realistic network consisting of six intersections, which is a part of List district of Hanover, Germany. The validation at this network aims to see whether the simulator can correctly model turning lanes at intersections.

Genetic algorithm (GA) is selected from the optimization techniques as a basis for the optimizer, since it is independent of the shape of the objective function being optimized. Two search routines are developed based on GA. The first (called parallel genetic algorithm) optimizes simultaneously all offsets in the network using genetic operators. The second search routine (called serial genetic algorithm) optimizes a group of offsets until the best solution is found. In the next step, the offsets of the next group of intersections are optimized. As the order of treating the intersections has a great influence on the optimization results, a method is developed for the determination of the search order. This method is based on the flows at entry links and the turning proportions.

Finally, the developed method is compared against the widely used optimization software (TRANSYT-7F), the full enumeration of the control variables and the dominance method. These comparisons are conducted on three case studies to benchmark the performance capabilities of the developed method against existing methods. These case studies are a two-way street of 3 intersections, a small network with 6 intersections and a relatively large grid network with 12 intersections.

2 Background on Traffic Signal Control

The use of traffic signals is considered to be one of the most effective ways to control traffic at road intersections. Traffic signals allow vehicle movements to be controlled by allocating time intervals, during which separate traffic demands for each arm of the intersection can make use of the available road-space. They are frequently adopted as means of traffic control at busy urban junctions. The use of traffic signals to control traffic movements can bring about major reductions in congestion and improve road safety.

The traffic engineering profession has more than 35 years of experience with computer control of traffic signals and the development and testing of various types of control strategies. Therefore, it is the purpose of this chapter to review the literature on the strategies of signal control. This chapter starts with the basic definitions of terms used in the signal control. Then it discusses briefly the elements of a timing plan in addition to the methods used in the signal control. Later, online and offline methods of signal control strategies are reviewed. For most of the methods, the main interests, if applicable, are: (1) the capabilities of its traffic flow model and the applicability of this model in all traffic conditions (undersaturated and oversaturated conditions); (2) its ways to generate offset plans; and (3) the effectiveness of its optimization process.

2.1 Basic Definitions of Signal Control

Before presenting the main variables and methods of signal control, it is necessary to define the commonly used terms in the design of signal control according to Garber and Hoel [47], HCM [80] and RiLSA [94].

Controller

A controller is an electrical device located in a cabinet for controlling the operation of a traffic control signal that changes the colors indicated by the signal lamps according to a fixed or variable plan. It assigns the right-of-way to different movements at appropriate times. [47]

Flow Rate

The flow rate is an equivalent hourly rate at which vehicles pass or arrive at a given point or section during a selected time interval of less than one hour, usually 15 min. It is calculated by dividing the number of vehicles observed by the time (in hours). For example, a volume of 200 vehicles observed in a 15-min period means a rate of flow of 200 veh/0.25h or 800 veh/h. [80]

Saturation Flow Rate

The saturation flow rate is the flow rate in veh/h that the lane group could carry if it had the green indication continuously. The saturation flow rate depends on the ideal saturation flow rate, which is usually equal to 1800 veh/h of green time per lane. This ideal saturation is adjusted for the prevailing traffic conditions to obtain the saturation flow rate for the considered lane group. Examples of prevailing conditions include: the number of lanes in the lane group; the lane width, the percentages of heavy vehicles in the traffic; and the approach grade. [80]

Lane Group

A lane group is a set of lanes at an intersection approach, which serves one or more traffic movements receiving the same green phase. The following guidelines are used to establish the lane groups for an approach:[47] [80]

- Unless the intersection approach contains a shared left-turn and through lane, it is preferred to establish separate lane groups for an exclusive left-turn lane. For exclusive right-turn lanes, the same guidelines are also applied.
- When exclusive left-turn lane(s) and/or exclusive right-turn lane(s) are provided on an approach, all other lanes are generally established as a single lane group.

Figure 2-1 shows the typical lane groups used for analysis. Note that when two or more lanes have been established as a single lane group for analysis, all subsequent computation must consider these lanes as a single entity. In the Figure, the abbreviations LT, TH and RT mean left-turn, through and right turn respectively, and EXC means exclusive.

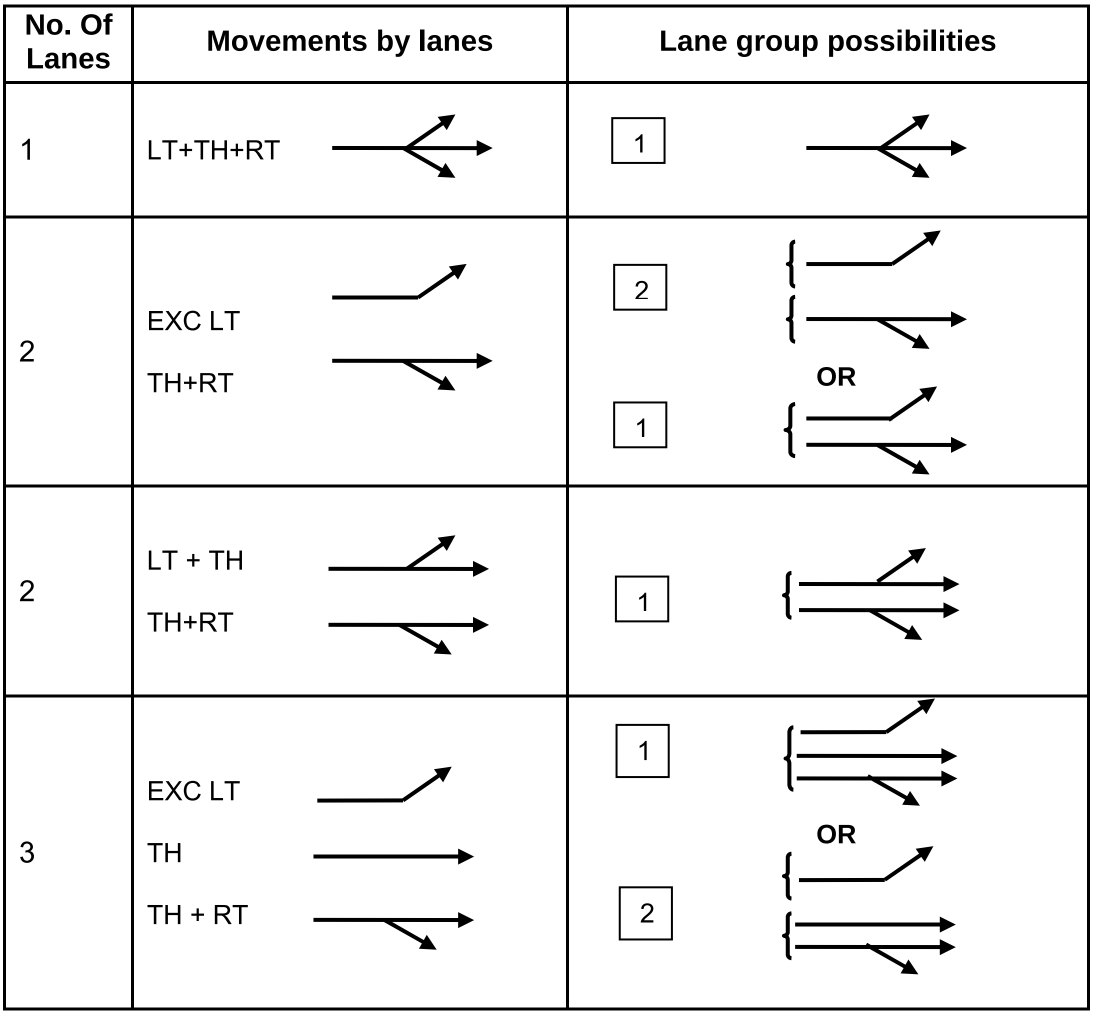


Figure 2-1: Typical lane group for analysis (source [80])

The critical Lane Group

The lane group that has the highest flow ratio for a given signal phase is known as the critical lane group. The flow ratio is the design (or actual) flow rate divided by the saturation flow rate. For example, Figure 2-2 shows three lane groups belonging to the same phase with their arrival flow rate. If the saturated flow rate is assumed to be 1600 veh/h for lane group 1 and 1700 veh/h for lane group 2 and 1600 veh/h for lane group 3, then the flow ratios for groups 1, 2, and 3 are 95/1600, 400/1700, 80/1600 respectively. Clearly lane group 2 is the critical lane group which has the largest flow ration 400/1700. [47]

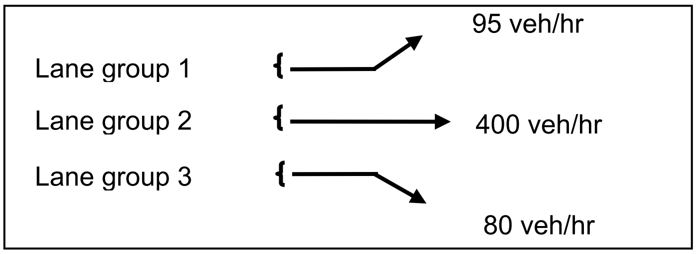


Figure 2-2: **Critical lane group determination**

Signal Phase

The signal phase is that portion of the signal time cycle allocated to any signal combination of lane groups receiving the right-of-way simultaneously during one or more intervals.

Interval

The discrete part of the signal cycle during which the signal indications do not change.

Amber before Red Interval

The amber before red interval is used after the green interval to warn drivers to the fact that the green light is about to change to red. [94]

Red and Amber Interval

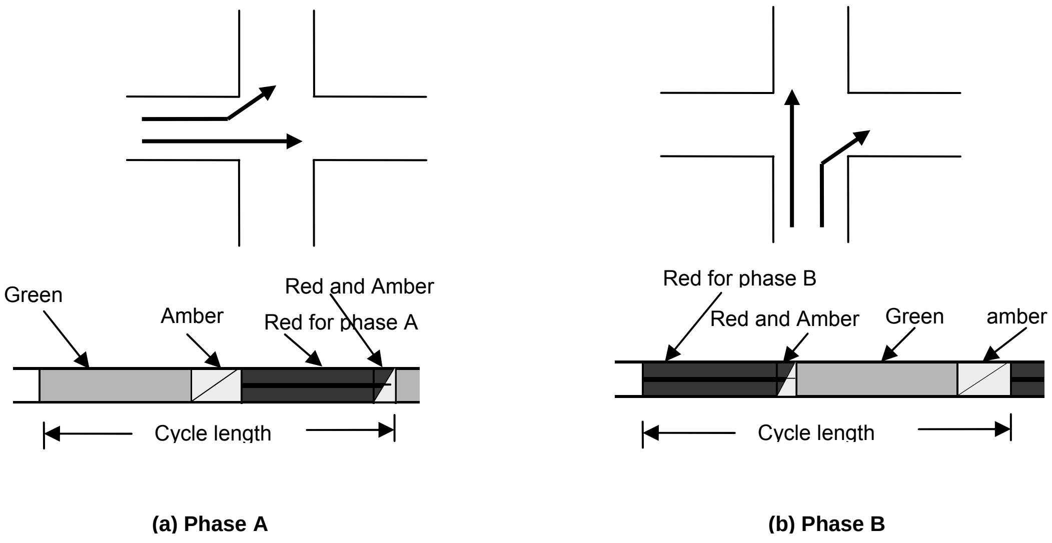
The red and amber interval (together) is used before green to prepare the road users for the immediately following green interval. [94]

Intergreen Time

The intergreen period is the interval between the end of green time of one signal group and the beginning of green time of the next coming (crossing or entering) signal group. [94]

Cycle Length

The cycle length is defined as the time in seconds required for one complete sequence of signal indication [47] or the sum of green phase and intergreen times [94]. Figure 2-3 is a graphical representation of a cycle. For a given phase, the cycle length is the time that passes from the start of green indication of phase A to the end of red indication of phase or the time that passes from the start of red indication of phase B to the end of amber indication of phase B.



(a) Phase A (b) Phase B

Figure 2-3: Two-phase Signal system at one-way intersection

Split

Split is a percentage of a cycle length assigned to each of the various phases in a single cycle.

Phase Sequence

Phase Sequence is a predetermined order in which the phases of a cycle occur.

Offset

The offset of an intersection is the difference between the beginning of green time of the first phase at the intersection, and a given system reference.

All-Red Interval

All-red interval is the interval at which a red color is indicated for all approaches of an intersection. Sometimes, it is used as an exclusive phase for pedestrian crossing or to allow vehicles and pedestrian to clear very large intersections before opposing approaches are given the green indication. [47]

2.2 Elements of a Timing Plan

To establish a signal control system for a network, it is necessary to develop a timing plan for all of the signals in the system. Such a timing plan consists of the following elements:

2.2.1 Signal Phasing and Phase Sequence

Non-conflicting and conflicting signal groups must be classified before phasing signals. Non-conflicting groups can be combined in a single phase, while conflicting ones must be separated into different phases, with the exception of turning movements when they are subject to the priority. [94]

When different movements are combined in a single lane, then they must simultaneously flow in a single phase, however, when they are separated with different lanes, then they may flow successively.

For the determination of the number of phases, two factors are considered: (1) the decision on which movement is to be protected by signalization, and (2) the requirements for coordination with neighboring intersections. If turning flows were not protected against opposing flows, the number of phases might be a minimum of two phases (see Figure 2-4). If all of them were protected, then the intersection requires at least 4 phases (see Figure 2-5).

To sequence the phases in an intersection, the following points must be taken into consideration:

- Certain directions have to be released one after the other, so that queuing vehicles do not cause blockage.
- Heavy pedestrian flows or cyclists have to be allowed to cross successive crossings rapidly.

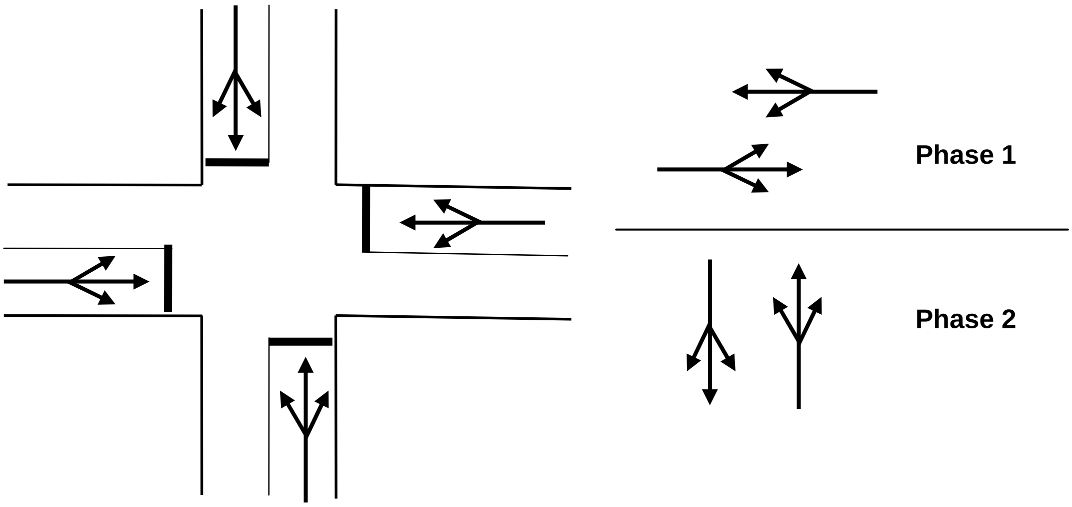
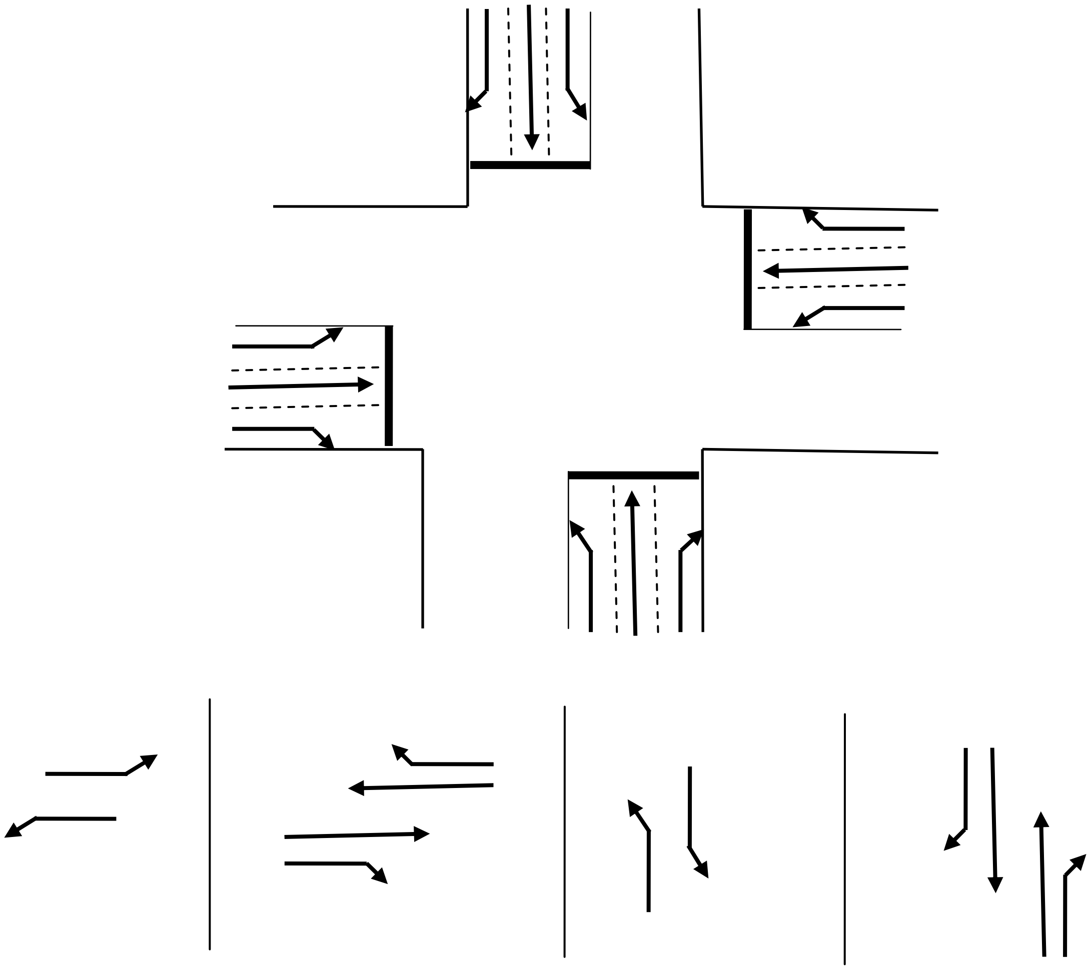


Figure 2-4: An example of two-phase control



Phase 1 Phase 2 Phase 3 Phase 4

Figure 2-5: An example of four-phase control

2.2.2 Cycle Length

For efficient signal control, the cycle length should be long enough to serve all of the critical groups, but not longer. If the cycle is too short, there will be so many phase changes that the time lost due to these changes will be high compared to the designated green time. However, if the cycle is too long, delays will be prolonged, as vehicles wait for their turn to discharge through the intersection. Figure 2-6 provides a graphical representation of this phenomenon. [47] [62]

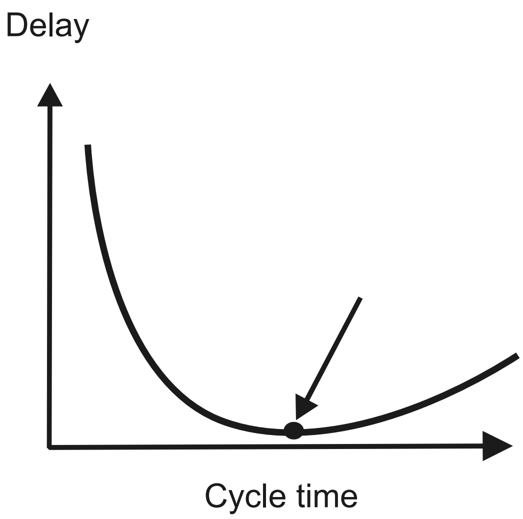


Figure 2-6: Cycle length versus delay (source: [62])

To solve this optimization problem, several methods have already been developed. Webster [104] provides an equation that minimizes intersection delay and gives the optimum cycle length as a function of the critical flow ratios and the Intergreen times as follows:

$$C_O = \frac{1.5 \cdot T_Z + 5}{1 - \sum_{i=1}^{P} q_i / s_i} \tag{2-1}$$

Where

C_0 = Optimum cycle length (sec)

P = Number of phases

q_i = The flow rate for the critical lane group in phase *i*.

s_i = The saturation flow rate for the critical lane group in phase *i*.

T_Z = The sum of Intergreen times for all phases, which is calculated by:

$$= \sum_{i=1}^{P} t_{Zi} \tag{2-2}$$

t_{zi} = Intergreen time for a phase change.

After calculating the optimal cycle time, one can increase it to the nearest multiple of 5 s. For example, if calculating a cycle time of 88.4, change it to 90s.

2.2.3 Green Splits

Once the total cycle length is calculated, the length of time that is allocated for green signal indications can be determined by subtracting the intergreen times from the total cycle length. [47] [62]

The critical lane groups are used to distribute the available green time among all of the phases. The flow ratio for a lane group is the actual flow rate divided by the saturation flow rate. The critical flow ratio, which is the one that is important for this calculation, is the flow ratio for the critical lane group.

Green time is assigned using a ratio equation. Each phase is allocated a part of the available green time depending on the ratio of its critical flow ratio to the sum of all the critical flow ratios. The part of the available green time that should be assigned to phase "i" can be determined using the following equation: [35]

$$g_i = \frac{(q_i / s_i)}{\sum_{i=1}^{P} q_i / s_i}(C_a) \qquad (2\text{-}3)$$

Where

g_i = The length of the green time for phase "i" (sec)

C_a = $(C - T_Z)$ the available green time for the cycle (sec)

It is necessary to provide a minimum green time that will allow the pedestrians to cross the intersection safely. The green time assigned to the traffic moving in the east-west direction should not to be less than the time needed for the pedestrians to cross the north-south approaches. In the same way, the green time assigned to the traffic moving in the north-south direction should not to be less than the time needed for the pedestrians to cross the east-west approaches. According to RiLSA [94], the minimum green time for vehicle traffic flows should not be less than 10s and for through traffic in the main direction 15 s is recommended. Alternatively, the minimum green time can be calculated based on the HCM expression as shown: [47]

$$g_{\min} = 3.2 + \frac{L}{S_P} + \left[0.824 \frac{N_{ped}}{W_E}\right] \quad for\ W_E > 3m \qquad (2\text{-}4)$$

$$g_{\min} = 3.2 + \frac{L}{S_P} + \left[0.27 N_{ped}\right] \quad for\ W_E < 3m \qquad (2\text{-}5)$$

Where

g_{min} = minimum green time.
L = crosswalk length in m.
S_P = average speed of pedestrians, usually taken as 1.22 m/sec.
3.2 = pedestrian start up time.
W_E = effective cross walk width in m.
N_{ped} = number of pedestrians crossing during an interval.

Once the minimum green time has been calculated for each phase, the length of the green interval for each phase is checked with these minimum values. If it is not greater than the length of the phase's minimum green interval, then the cycle length must be increased and green time should be added to that phase until the green interval is equal to or greater than the minimum.

2.2.4 Offset

Until now, the interactions that have been described of the above elements of a timing plan pertain to a single intersection. The relationship between signal timings at different intersections in terms of these variables has not as yet been described. The offset variable – which was previously defined as the difference between the beginning of green time for the first phase at an intersection, and a given system reference – serves this purpose.

The coordination starts with allocating a common cycle or some multiple of a common cycle for all intersections in a considered area. The green splits may vary at each intersection but the offset values for each intersection must be determined. The offset is usually established with reference to one master intersection in this system.

The arterial street signal coordination is based on the concepts of pulse flow – that groups of vehicles (platoons) leave a signal and travel in platoons to the next signal. Therefore, it is desirable to determine a time relationship between the beginning of green at one intersection and the beginning of green at the downstream intersection such that vehicles that arrive during the red interval may receive a green before the next platoon arrives. This allows the continuous (progressive) flow of traffic along an arterial street and helps to reduce stops and delays. [84]

The concept of arterial street coordination can be graphically represented in a time-distance diagram, as shown in Figure 2-7. The through-band is the space between a pair of parallel space lines that defines a progressive flow on a time-distance diagram. The slope of the through-band represents the progressive speed of traffic moving along the arterial. The width of the through-band in seconds indicates the period of time available for traffic to flow within the band. [84]

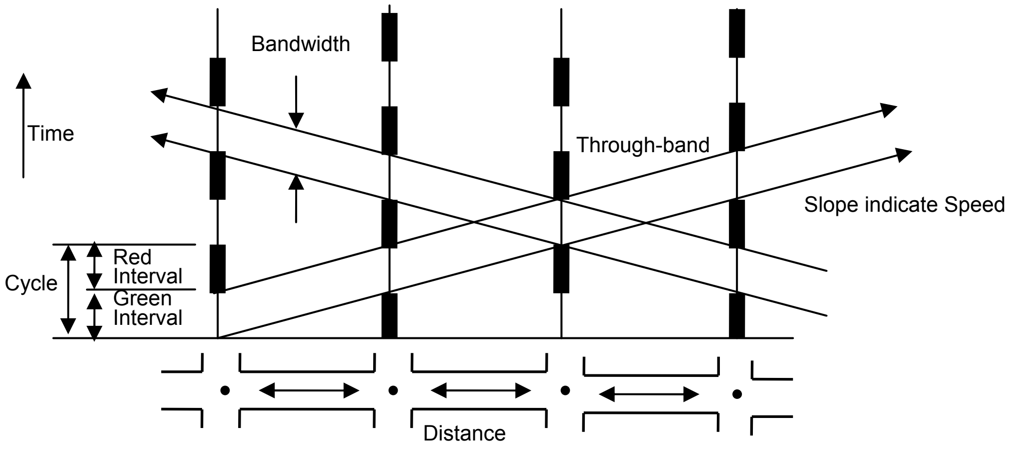


Figure 2-7: Time-distance diagram (source: [84])

The use of one-way or two-way operations is a major consideration in developing timing plans for an arterial street. If the street is one-way providing full progression can be obtained, unlike when it is two-way which is more difficult. The problem is much more complex when coordinating a network which is still a point of research.

2.3 Methods of Signal Control

Traffic signals may operate independently, or as a system. The scale of control can be grouped into 3 categories: [83]

- Individual Intersection Control– A single traffic signal operates in a pre-timed, actuated, or traffic responsive mode, without affecting the operation of other traffic signals.
- Arterial Control– Two or more traffic signals operate synchronously along an arterial street in a pre-timed progression, traffic responsive, or adaptive control mode.
- Network Control– Traffic Signals throughout an entire network of intersections are coordinated through a timing plan created offline, or an adaptive control strategy.

There are many different levels of traffic signal control, from the individual intersection with pre-timed control to the network-wide system with adaptive control. Here are short descriptions of the different methods of operation.

2.3.1 Fixed-Time Control

Fixed time control also known as time of day control (TOD) is the most widely used form of signal control. In fixed time control, signals are controlled using fixed timing plans. There are usually several coordination timing plans that are put into operation on a ‘time of day, day of week' schedule. The plans consist of splits, cycle lengths, and offsets for every signalized intersection. These plans are developed offline either using manual methods or using special software and then loaded onto the controller. These plans are fixed. Consequently, if a timing plan is not adjusted as traffic patterns change over the years, it becomes outdated. Fixed time control is limited because timing plans are developed to serve certain traffic conditions. If there is an unexpected condition, there may not be a timing plan to handle it. Furthermore, fixed-time signals have the major disadvantage that they cannot adjust themselves to handle fluctuation volumes. [42]

2.3.2 Vehicle Actuated Control

Vehicle-actuation control is used for the purpose of handling fluctuation volumes. When using such a control signal, vehicles that arrive at an intersection are recorded by detectors, which send this information to the controller. The controller then adjusts the phase lengths to correspond to the existing traffic conditions. Actuated signal control can be either semi-actuated or fully actuated. [47] [83]

A semi-actuated controller provides for traffic actuation of all phases except the main phase. A continuous green is kept on the major street except when a demand is registered by the minor street detector. When no vehicles are present on the minor street or a timing limit has been reached, the right of way always returns to the major street. Semi-actuated operation is best suited for locations with low volume minor street traffic. It may also be used to permit pedestrian crossings at major streets.

In the full actuated control, the purpose of the controller is to measure traffic flow on all approaches to an intersection and allocate the right of way in accordance with traffic demand. Full actuated control requires placement of detectors on all approaches to the intersection. The controller's ability to respond to traffic flow provides for maximum efficiency at individual locations. This type of control is appropriate for intersections where the demand proportions from each leg of the intersection are less predictable.

Although actuated control methods have the advantage of working well for fluctuating light traffic, they have the disadvantage of functioning like fixed time controllers with high traffic loads.

2.3.3 Traffic Responsive Control

With traffic responsive control, signals are also coordinated using fixed timing plans as in fixed-time control. However, traffic responsive control differs on how the plans are chosen for operation. Traffic responsive control chooses for operation an appropriate timing plan from a library of different plans based on different traffic conditions. There are several traffic detectors that are continuously collecting real time traffic measurements. When traffic reaches certain predefined limits, various timing plans are put into operation. With traffic responsive control the same problems are faced as with fixed time control in that timing plans become old, and that there may not be a coordination plan already developed to handle unexpected conditions. [42]

2.3.4 Adaptive Control

As opposed to the methods of signal control outlined above, which use historical data to create fixed-time optimized timing plans, traffic adaptive controls use real time data obtained from several detectors to perform constant optimizations on the signal timing plan for an arterial or network. The split, cycle length, and offset timing parameters are continuously and incrementally adjusted by the control system. As traffic changes, the splits, offsets, and cycle lengths are adjusted by small amounts, usually several seconds. The goal is to always have the traffic signals operating in a manner that optimizes traffic flow.

Adaptive signal control provides three main benefits: [42]

- Reducing stops and delay;
- Accommodating non-recurring congestion, incidents, events, or traffic demand growth over time; and
- Keeping up with the gradual changes in traffic over time.

2.4 Well-Known Control Software

Recent decades have seen both online and offline strategies developed for computerized signal control. A signal control plan generating strategy (online or offline) basically contains two major components: an optimizer and a traffic flow model. An optimizer is a search procedure that explores a solution space for optimality. On the other hand, a traffic flow model is used along the optimization process to assess performance indices (or optimization criteria) of concern. Next, several existing online and offline software of signal control plan generation are reviewed.

2.4.1 Offline Control Strategy

Traffic Network Study Tool (TRANSYT)

TRANSYT is one of the most widely used offline signal coordination programs, which was originally developed by the Transportation and Road Research Laboratory in England in 1968. TRANSYT is a macroscopic, deterministic simulation and optimization model. The link flows and link turning proportions are required as input for the simulation model and assumed to be constant for the entire simulation period. The traffic model represents traffic behavior in a highway network in which most junctions are controlled by traffic signals. [83]

The model predicts the value of a ‘performance index’ (PI) for the network for a given fixed-time plan and an average set of flows on each link. The performance index measures the overall cost of traffic congestion and is usually a combination of the total delay and the number of stops made by vehicles. The optimization process adjusts the signal timings and tests if the adjusted timings improve the performance index using the traffic model. By successive adoption of beneficial timings, an optimum is reached.

TRANSYT has the following known weaknesses and limitations that have been reported: [15]

- Its traffic model is not applicable for oversaturated traffic conditions.
- Its hill-climbing optimization algorithm does not guarantee that a global optimum will be achieved and consequently does not guarantee that the “best” signal control plan will be found either. This is because the signal control problem in general has a solution space for the PI, which consists of a number of local optima. It is computationally infeasible, when using the hill-climbing technique, to search through all local optima to find the best one.
- Its solution is mainly dependent on the quality of the starting solution, which is not always available. Again, this is due to the nature of the hill-climbing technique.

SOAP

SOAP provides a macroscopic analysis to develop signal control plans for isolated intersections; therefore the offset is not included in the optimization process. It develops cycle lengths and splits that minimize a performance index. Inputs include truck and bus composition traffic flows, turning proportions, saturation flow rate, and signal data. Outputs include delay, queues, degree of saturation, excess fuel consumption, left turn conflicts, and stops. [83]

MAXBAND

MAXBAND is a bandwidth-based optimization program that finds signal timing plans on arterials and triangular networks. MAXBAND produces cycle lengths, offsets, speeds, and phased sequences to achieve maximal progression bandwidths [83]. The weaknesses of the MAXBAND model have been reported as follows:

- It is not generally true that maximizing bandwidth minimizes such measures-of-effectiveness as delay. [15]

- It requires extensive computation time since it is based on the mixed-integer linear programming (MILP) formulation and employs branch-and-bound techniques for its solution, which are infeasible for realistic network problems. [11]
- Its generated progression schemes are of uniform width, which does not always hold. The effect of using average through-moving volume for allocating the total bandwidth is that the green band may be either wasted at intersections with lower than the average through-moving volume, or be deficient at intersections with higher than the average through-moving volume. This has been a major drawback of progression methods and, for varying traffic volumes, optimum results cannot be guaranteed. [93]

PASSER

The PASSER series of software, PASSER II (Messer et al. [74] [75]), PASSER III (Messer et al. [76]) and PASSER IV (Chaudhary and Messer [12]), optimizes signal timings for arterials, diamond interchanges and networks (including arterials), respectively. All three methods are based on a macroscopic traffic model and are not capable of optimizing signal timing in oversaturated conditions.

PASSER II

PASSER II is a macroscopic simulation model developed to optimize signal timing parameters to provide good progression on arterial streets. It uses a concept based on Brooks' bandwidth algorithm for providing progression along an arterial [7]. To achieve maximal bandwidth, interference with progression bands is minimized in the objective function. By optimizing cycle length, offset and phasing sequence, the program finds the maximum progression efficiency that can exist along an arterial street.

The weaknesses of the PASSER II model have been reported as follows:

- It is unable to cope with multi-arterial networks with closed loops. [11]
- The control strategy of this model ensures the enhancement of traffic flow on the main arterial street, while often generating poor performance on the cross streets. In networks consisting of several intersecting arterial streets, this deficit becomes more evident. A basic limitation of such a model is that the progression schemes generated are of uniform width. [93]
- Its optimizer does not guarantee a global optimum. [98]

PASSER III

PASSER III optimizes signal timings at conventional diamond interchanges. Delay-difference-of-offset methodology is used to calculate interior delays, and HCM methodology is applied to calculate external delays. [100]

PASSER IV

PASSER IV applies a mixed-integer linear programming model to optimize networks [13] and optimizes progression bandwidth efficiency. It is capable of optimizing both arterials and networks. Like other progression-based signal optimization models, PASSER IV suffers from the fact that it employs a simplistic traffic model. Consequently, its applicability is limited. Besides, it does not have the capability to estimate traffic performance measures such as delay.

SIGOP III

By using a macroscopic traffic flow model, SIGOP determines cycle length, splits, and offsets of signals in a grid network that minimize delay. SIGOP can handle up to 150 intersections. Outputs include time-space plots along selected arterials and link statistics. The program is limited to model four phases and does not model permissive and unprotected turns explicitly. [33]

2.4.2 Online Control Strategy

In this section, several existing online signal control strategies are reviewed in the classification of UTCS. They are categorized based on the definitions of the three generations of control methodology in UTCS.

2.4.2.1 First Generation Control Strategies

The First Generation Control (1-GC) uses a library of pre-stored timing plans, which are optimized offline and based on historical traffic data using popular signal optimization models such as TRANSYT. Essentially, the criteria to choose a plan that drives a signal control system are: [71]

- On the basis of time-of-day (TOD) and day-of-week (DOW);
- By direct operator selection; or
- By matching from an existing library a plan best suitable for recently measured traffic conditions (e.g., volumes and occupancies).

One example of signal control strategies that fits into this category is:

1-GC UTCS

1-GC UTCS [71], the corresponding method of signal control is named as the traffic responsive control (TRSP), which updates frequency once every 15 minutes. The results from a comprehensive study prove that TRSP generally matched or was superior to the performance of TOD and well-timed fixed-time control. [30]

1-GC UTCS equips itself with a transition routine to ensure a smooth transition from one signal control plan to another. Furthermore, it includes a critical intersection control (CIC) feature to enable adjustment of green splits at selected signals that saturate frequently.

The major critique of the 1-GC UTCS is the difficulty in maintaining an up-to-date library of signal control plans, which leads to it becoming irresponsive to real-time traffic conditions [42]. After failures of the 1-GC UTCS in the late 60ies and 70ies, the second generation had a breakthrough with the SCOOT method in the early 80ies. [58] [71]

2.4.2.2 Second Generation Control Strategies

The Second Generation Control (2-GC) is a real-time strategy that calculates and implements timing plans based on surveillance data and predicted changes. Three online signal control strategies that fit into this category are presented in this section, which are 2-GC UTCS, SCOOT and SCATS.

The 2-GC UTCS

The 2-GC UTCS program, or TANSTP (Traffic Adaptive Network Signal Timing Program) [31], contains an optimization algorithm (i.e. SIGOP), a traffic prediction model, sub-network configuration models, a CIC feature, and a signal transition model. In this program, the optimization process is repeated at 5-minute intervals. However, new signal control plans cannot be applied more often than once every 10 minutes to avoid transition disturbances [71]. The sub-network configuration models are designed to decompose the network dynamically into sub-networks based on prevailing traffic conditions so that the optimum signal timing plans are then calculated for individual sub-networks and the sub-networks are interfaced to ensure smooth traffic progression across the sub-network boundaries.

The performances of 2-GC UTCS were mixed [30]. Although it demonstrated some small improvements on arterial streets compared to well-timed fixed-time signal, it showed a degraded performance on a network-wide basis.

SCOOT

SCOOT was developed by the Transport Research Lab in the UK [41]. The basic principle of SCOOT is to measure traffic volumes continuously on all intersection approaches in the network and to change the signal timings to minimize a Performance Index (PI), which is a composite measure of queue length, delay and stops [72]. To prevent major disruptions in traffic flow, the changes in timing plans are made small enough, but to permit rapid response to changing traffic conditions they are made frequent enough. SCOOT consists of three basic principles [88].

- Online measurement of a Cyclic Flow Profile (CFP). A CFP is fundamental to SCOOT. It is also the foundation of the traffic flow model used in TRANSYT. Therefore, the first step of SCOOT is to measure CFP online.
- Online traffic flow model. SCOOT is essentially an online TRANSYT. Based on the generated CFP, SCOOT then projects platoon movement and dispersion at the downstream intersection. This helps it to model queue formation and queue discharge, which are needed in real-time by the signal optimizer. SCOOT assumes platoons traveling at a known cruising speed with some dispersion and queues discharging at a known and constant saturation flow rate.
- Incremental optimization. SCOOT makes signal control plan changes in a series of frequent, but small, increments because of the difficulty of predicting traffic situations in a relatively long period of time. In SCOOT, the timing plan elements can be increased or decreased to fit into the latest situation suggested by the measured CFP in the following manner: a few seconds before every phase changes, its split optimizer determines whether to advance or hold back the scheduled change by up to four seconds or to leave it unaltered. Then, once a cycle, the offset optimizer assesses whether the PI can be reduced by altering the offset of each intersection by up to four seconds earlier or later. Favorable split and offset alterations are implemented immediately. The cycle time for a group of intersections may, in a similar fashion, be incremented up or down by a few seconds every few minutes.

The benefits of SCOOT are reported to be comparable to well-timed fixed-time-signal control [41]. However, there are some disadvantages on SCOOT:

- It has no mechanism to impose turning restrictions based on traffic demand to improve network performance;
- SCOOT builds on a vertical queue model and thus can not consider the effect of downstream link congestion on the signal output. It operates fairly well as long as the network is not overcongested. However, it fails to model the effect of downstream congestion on the capacity of upstream intersections during queue spillback.

SCATS

SCATS has been developed by the Australian Roads and Traffic Authority of New South Wales since the early 1970’s [69]. SCATS, like SCOOT, changes cycle time, splits and offsets in response to real-time traffic measurements to minimize stops and delay. However, unlike SCOOT, it is not model-based. Instead it chooses from a library of timing plans and therefore relies extensively on available traffic measurements [72].

SCATS controls signals in groups with a critical intersection specified for each group. Cycle time and splits are calculated for each critical intersection, while the timing plans for other intersections are selected to handle the plans for the critical intersections. At the strategic level, control is carried out by a higher-level system, responsible for a group of up to about 10 signals. At the tactical level, control is handled by local controllers and optimization occurs at the intersection level, which is basically seeking self-optimization within the restrictions imposed by strategic level. Tactical control permits early termination of green phases when the demand is less than average and for phases to be omitted completely when there is no demand. All the extra green time is added to the main phase or can be used by subsequent phases.

Cycle time optimization takes place each cycle, while splits may vary each cycle based on the average degree of saturation on approaches over the last three cycles. Phases may terminate earlier or, when there is no demand, be omitted altogether. The optimum offset is calculated each cycle, but only implemented when at least three out of the previous five cycles have suggested a change to that offset.

Four modes of controls are offered by SCATS: [72]

- In the first or "normal mode" integrated traffic responsive control is provided;
- The second or "fall-back mode," implements time-of-day plans when computer or communication failure occurs;
- In the third mode or "isolated control mode," there is only local vehicle actuation with isolated control; and
- In the fourth mode, the normal signal display shows flashing red or flashing yellow on all approaches.

A survey done by Luk et al. [70] pointed out that SCATS showed no significant reduction in travel times compared against TRANSYT; however, a large reduction in the number of stops was proved.

2.4.2.3 Third Generation Control Strategies

The Third Generation Control (3-GC) is a fully traffic-responsive, online signal control strategy. It basically permits "cycle free" operations and allows the parameters of signal control plans to change continuously in response to real-time traffic measurements. Three online signal control strategies are presented in this section, which are 3-GC UTCS, OPAC, and PRODYN.

The 3-GC UTCS

Like 2-GC, the third Generation Control (3-GC) is a fully responsive real-time traffic control program, which computes control plans to minimize a network-wide objective function using predicted traffic conditions [83]. CYRANO policy or the Cycle-Free Responsive Algorithm for Network Optimization is implemented at undersaturated conditions. Along oversaturated paths, another policy is applied CIC/QMC, which consists of a CIC policy coupled with a QMC (Queue Management Control) policy. The 3-GC differs from the 2-GC model in that the period after which timing plans are adjusted is shortened by 3 to 5 minutes, and the cycle lengths are allowed to vary among the signalized intersections during the control period. [32] [71]

The following table presents a comparison between the three UTCS control strategies in terms of three points (1) how long the update interval is; (2) how the control plans are generated (online/offline); (3) how the traffic is predicted; and (4) weather the cycle length is varied or not. [83]

Table 2-1: Comparison of UTCS Control Strategies (Source: [46])

FEATURE	First Generation Control (1-GC)	Second Generation Control (2-GC)	Third Generation control (3-GC)
Update interval	15 min	5-10 min	3-5 min
Control plan generation	Off line optimization selection from a library by time of day, traffic responsive, or manual mode.	Online optimization	Online optimization
Traffic prediction	None	Historically based	Smoothed current values
Cycle length determination	Fixed within each intersection	Fixed within variable groups of intersections	Variable in time and space. Predetermined for control period.

OPAC

OPAC, which is an online computational strategy for demand-responsive traffic signal control, has been developed in the U.S. since the early 1980's by Parsons Brinkerhoff Farradyne Inc. and the University of Massachusetts at Lowell [44]. In OPAC, the network is divided into subnetworks, which are considered independently. It has two models: one for oversaturated networks and the other for undersaturated networks. [72]

The oversaturated model considers the saturation flows and maximizes the number of vehicles that can pass through an intersection. It also considers the critical links like those on the border of spillback. Except for the computation of cycle length, OPAC is not controlled by a central computer. Hence, it can run independently if communications to the central server fail. [72]

The undersaturated model, on the other hand, determines the signal timings using one of two methods: (1) Fixed-time plans are obtained offline, or (2) the plan is determined dynamically. The level of local to network control can be configured by the user. The local signal timings are based on detected data (15 seconds) and predicted data (60 seconds). These are implemented for a time-step (roll period) of 2-5 seconds.

Here are some critiques of OPAC that have been reported in literature:

- Its optimization process is limited because of the tremendous computational efforts involved in the optimal sequential constrained (OSCO) method, which is an exhaustive search of all possible combinations of valid switching times within the stage to determine the optimum set.
- Even though up to three phase changes are allowed for each stage, it allows at most two phase changes to further reduce the search space. [32]
- Even though its calculated results approach the theoretical optimum, it does not guarantee a global optimality due to the fact that its optimization process does not lend itself very well to dynamic programming. [91]

PRODYN

PRODYN is a French real-time signal control program. It has been developed by the Centre d’Etudes et de Recherches de Toulouse (CERT) over the last decade ([56], [3] and [57]). PRODYN is part of the European DRIVE (Dedicated Road Infrastructure for Vehicle Safety in Europe) program. [64]

PRODYN has been developed in two stages: two-level hierarchical control (PRODYN-H) and then decentralized control (PRODYN-D). The former provides the best result; however, its applicability is limited to about 10 intersections of network size due to the complex computations involved. The latter, on the other hand, overcomes those limitations.

The major reported critiques of PRODYN are; its limitation to two-phase operations; its restriction of allowing only four approaches to an intersection; and its employment of a simplistic traffic flow model. [28] [27]

3 Background on Optimization Techniques

Once it is recognized that travel demand and the number of vehicles are rapidly increasing, optimization can play an important role in urban network traffic management and transportation modeling. There are at least two reasons for continuous development in this field:

- The increasing importance of environmental effects such as safety, air pollution, economics, etc; and
- The limited availability of land and financial resources, especially in developing countries.

In addition, rapid development of digital computers, which rely on numerical approximation rather than on exact analytical solutions, stimulates the advances in today's predominating optimization techniques, which combined constitutes a new and powerful technology, opens new horizons and reveals new possibilities.

Signal coordination methodology usually involves the optimization of an objective function (traffic simulator) that calculates the performance index (PI) of the given system. This chapter begins with the definition of the optimization and then gives an overview of the different types of optimization problems. Since the considered objective function (the developed traffic simulation routine) is a non-smooth optimization problem, genetic algorithm (GA), which is a solution method for such a problem, is used as the basis of the developed optimizer in this research. Therefore, an introduction to genetic algorithms and the terminology used therein is also provided.

3.1 What Does Optimization Mean?

Both words “optimum” and “optimus” are Latin; the former means “the ultimate ideal” and the latter means “the best”. From this, optimizing refers to bringing whatever we are dealing with to its ultimate state.

Mathematically, the optimization problem could be defined as:

$$\textit{minimize/maximize } f(x) \qquad \textit{subject to } x \in \Omega \tag{3-1}$$

The function f that one wishes to minimize is a real or integer value function, and is called the objective function, fitness function or the cost function. The vector x is a vector of n independent variables, that is $x = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n$. The variables x_1, …, x_n are often referred to as control variables or decision variables. The set Ω is a subset of $\mathbb{R}^n$, called the constraint set or feasible set.

The above optimization problem can be considered as a decision problem that involves finding the best vector x of control variables over all possible vectors in Ω. The “best” vector means the one that is resulting in the smallest/largest value of the objective function. This vector is called the minimizer/maximizer of *f* over Ω.

3.2 Types of Optimization Problems

In an optimization problem, the solution methods or algorithms that can be used for optimization, how it is solved, and the certainty one can have that the solution is truly optimal are determined based on the mathematical relationships between the objective functions and constraints and the decision variables. The following types of optimization problems are arranged in order of increasing difficulty for the solution of method according to Frontline Systems Inc.[39].

3.2.1 Linear Programming Problem

In a linear Programming (LP) problem, both the objective and the constraint functions are a linear combination of decision variables. An example of a linear function consisting of four decision variables is:

$$\textit{minimize } f(x) = 20 \cdot x_1 - 14 \cdot x_2 + 0.5 \cdot x_3 + x_4$$

$$\textit{subject to } g(x) = x_1 + x_4 = 50 \tag{3-2}$$

Where f(x) is a linear objective function consisting of four decision variables, i.e. x_1, x_2, x_3, and x4 and g(x) is a linear constraint function forcing the summation of x_1, and x_4 to be 50.

The Simplex method is usually used to solve LP problems, and was originally developed by Dantzig in 1948. By using advanced methods from numerical linear algebra, the simplex method has been improved dramatically in the last decade. For that reason, it is possible to solve LP problems with up to hundreds of thousands of decision variables and constraints. [39]

In 1984, Karmarkar developed an alternative to the Simplex method, called the Interior Point or Newton-Barrier method. According to Frontline Systems Inc. [39], this method has been improved dramatically with advanced linear algebra methods in the last decade. Therefore, it is often in competition with the Simplex method, especially where very large problems are concerned.

3.2.2 Quadratic Programming Problem

In a quadratic Programming (QP) problem, the objective function is a quadratic combination of decision variables while the constraint functions are all a linear combination of decision variables. An example of a quadratic function consisting of two variables is:

$$\text{minimize } f(x) = 20 \cdot x_1^2 + 14 \cdot x_2$$

$$\text{subject to } g(x) = 2 \cdot x_1 - x_2 = 30 \qquad (3\text{-}3)$$

Where f(x) is a quadratic objective function consisting of two decision variables, i.e. x_1 and x_2 and g(x) is a linear constraint function of decision variables.

A QP problem can be solved by a smooth non-linear optimization method such as the Generalized Reduced Gradient (GRG) or Sequential Quadratic Programming (SQP) method, because it is a special case of a smooth non-linear problem. Nevertheless, an extension of the Simplex method or an extension of the Interior Point or Barrier method are considered to be faster and more reliable at solving a QP problem.

3.2.3 Quadratic Constrains Problem

In a quadratic constraint problem, both the objective and constraint functions are quadratic combinations of decision variables. An example of a quadratic constraint function consisting of two variables is:

$$\text{minimize } f(x) = 20 \cdot x_1^2 + 14 \cdot x_2$$

$$\text{subject to } g(x) = 2 \cdot x_1^2 - x_2 = 30 \qquad (3\text{-}4)$$

Where f(x) is a quadratic objective function consisting of two decision variables, i.e. x_1 and x_2 and g(x) is a quadratic constraint function consisting of decision variables. A quadratic constraint problem can be solved via specialized Interior Point methods.

3.2.4 Mixed-Integer Programming (MIP) Problem

In a mixed-integer programming (MIP) problem, the objective and constraint functions can be quadratic or linear combinations of decision variables. However, some of the decision variables are constrained to have only integer values (such as -1, 0, 1, 2, etc.) at the optimal solution. Here is an example:

$$\begin{aligned} &\text{minimize } f(x) = 20 \cdot x_1 + 14 \cdot x_2 \\ &\text{subject to } g(x) = 2 \cdot x_1 - x_2 = 30 \\ &\qquad x_1 \in \{0,1,2,3,4,5\} \end{aligned} \tag{3-5}$$

Branch and Bound is a "classic" method for solving this type of problem. Examples of alternative methods are Genetic and evolutionary algorithms. These methods randomly generate initial populations of candidate solutions belonging to the solution space of the problem that satisfies the integer constraints. Then, methods such as integer mutation and crossover are used to transform the initial solutions into a new candidate solution, which continues to satisfy the integer constraint but may have better fitness values. This procedure is repeated until a convergence criterion is satisfied. [39]

3.2.5 A Smooth Non-linear Programming -Problem

In a smooth non-linear programming (NLP) problem the objective or at least one of the constraints is a smooth non-linear function (those and their gradients are continuous) of the decision variables. The variables of non-linear functions may be divided or multiplied by other variables or raised to a power. The non-linear functions may also use transcendental functions sine, cosine, exp and log. Here is an example:

$$\text{minimize } f(x) = 20 \cdot x_1^3 + 14 \cdot x_1 \cdot x_2 + \log x_1 \tag{3-6}$$

The most widely used and effective methods for solving NLP problems are the Generalized Reduced Gradient (GRG), Sequential Quadratic Programming (SQP) methods and the Interior Point or Barrier methods. [39]

3.2.6 Global Optimization (GO) Problem

Global Optimization (GO) Problem is a smooth non-linear optimization problem where a global optimal solution is sought. Since it is believed that the time needed to solve a GO problem increases exponentially with the number of variables and constraints, GO problems are very difficult to solve. [39]

Multi-start methods are famous methods to search global optimal solutions with the help of smooth non-linear solver, which finds only local optimum solutions. These methods are based on the idea of automatically starting the non-linear solution from different random starting points. Next, the best of the different local optimal solutions reached is selected and considered as the proposed global optimal solution.

Other methods called Continuous Branch and Bound methods are also used to seek the global optimal solution. They rely on systematically subdividing the feasible solution space into small sub-regions and finding local optimal solutions in each sub-region. Then, the best of the different local optimal solutions reached is selected and considered as the proposed global optimal solution.

Although Tabu Search, Scatter Search and Genetic Algorithms are used to find "good" solutions to non-smooth optimization problems; they can also be applied to smooth non-linear problems to search and propose a global optimal solution. According to Frontline Systems Inc. [39], these methods are more effective at finding better solutions than "classic" smooth non-linear solvers alone, but they usually take much more computing time.

3.2.7 Non-Smooth Optimization (NSP)

The most difficult type of optimization problem to solve is a non-smooth problem (NSP). It is one in which some of the functions are non-smooth or even discontinuous. The relationships may include *if* functions like:

$$minimize\ f(x) = 20 \cdot x_1^3 + 14 \cdot x_1 \cdot x_2 + \log x_1$$

$$x_1 = \begin{cases} x_2 + 5 \text{ if } x_2 < 0 \\ x_2 - 1 \text{ if } x_2 > 0 \end{cases} \tag{3-7}$$

Genetic or Evolutionary Algorithms, Tabu Search and Scatter Search are used to find "good" solutions for this type of problem.

3.3 Simple Genetic Algorithm

Search algorithms are frequently used to find an optimal solution in a feasible solution space. The uninformed search methods, the depth first, and the breadth first algorithms are the simplest search methods [49]. Hill Climbing Method is an improved method of searching, starts with an initial estimate of the solution and then gradually improves the initial estimate until it reaches the optimum. It jumps onto the function and moves in the steepest (best) permissible direction [49]. This methodology certainly allows it to find the local optimum, but not necessarily the global one.

According to Obitko [82], Rechenberg introduced Evolutionary Computing in the 1960s in his work "Evolution strategies" (Evolutionsstrategie in original). Then, other researchers developed his idea. In 1975, Holland and his colleagues developed Genetic Algorithms (GAs) and published a book called “Adaptation in Natural and Artificial Systems".

Evolutionary algorithms are general-purpose search procedures based on the concept of natural selection and population genetics [92]. There are four popular variants of evolutionary algorithms: genetic algorithms (GA), evolutionary strategies (ES), evolutionary programming (EP), and genetic programming (GP). The best known evolutionary computation methods are genetic algorithms.

According to Goldberg [49], Genetic algorithms (GAs) are optimization techniques based on the concepts of natural selection and genetics. Genetic algorithms differ from traditional algorithms in that they (1) search from a population of points, not a single point; (2) work with a coding of the parameter set, not the parameters themselves; (3) use stochastic rules, not deterministic rules; and (4) use information of objective function, not derivations.

3.3.1 The Procedure

In the genetic algorithm approach, the variables are represented as genes of a chromosome. The simple genetic algorithm procedure can be summarized in the following steps: [49] [101]

1. Randomly generate an initial population of candidate individuals belonging to the solution space of the problem.
2. For every evolutionary step, known as a generation, the individuals in the current population are decoded and evaluated according to some predefined quality criterion, referred to as the fitness, or fitness function.
3. New individuals are selected according to their fitness values, in order to form a new population (the next generation). This is done through natural selection and the genetic operators- selection, crossover, and mutation.
4. Replace the old members of the population with the new ones.
5. If a fixed number of generations are up or convergence is reached, then stop and return the best chromosome; if not, go to step 2.

One iteration of this loop is referred to as a generation. The initial generation (generation 0) of this process operates on a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure, operate to improve the performance of the newly selected population. Natural selection guarantees that chromosomes with the best fitness will reproduce into future populations. Using the crossover operator, the GA combines genes from two parent chromosomes to form two new chromosomes (children) that have a high probability of having better fitness than their parents. Mutation allows the generation to jump outside a local minimum. A flow chart of this methodology is shown in Figure 3-1.

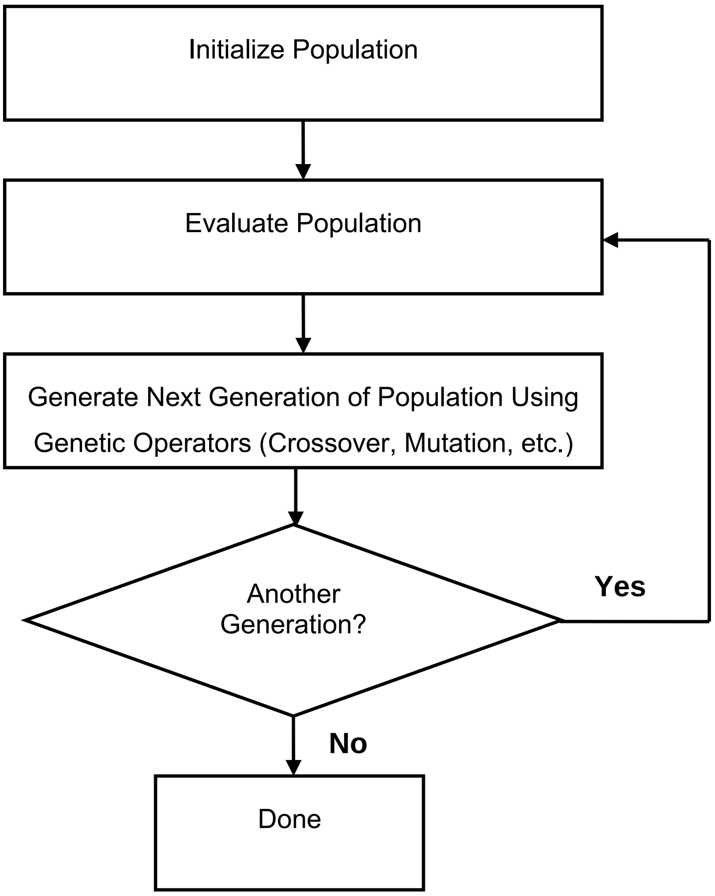

Initialize Population
Evaluate Population
Generate Next Generation of Population Using Genetic Operators (Crossover, Mutation, etc.)
Another Generation?
Yes
No
Done

Figure 3-1: Flowchart of simple genetic algorithm

3.3.2 Advantages and Disadvantages of GA

The main advantages of GA are: [49]

- GA requires no knowledge or gradient information about the solution space;
- Discontinuities on the solution space have little effect on overall optimization performance;
- It is resistant to becoming trapped in local optima;
- It performs very well for large-scale optimization problems; and
- Can be employed for a wide variety of optimization problems.

The main disadvantages of GA are: [49]

- GA has trouble finding the exact global optimum; and
- It requires large number of response (fitness) function evaluations.

3.3.3 GA Terminology and Parameters

A description of the commonly used GA terminology and parameters studied in this dissertation is described in this section according to Obitko [82] and Wall [102].

The simple genetic algorithm creates an initial population by cloning the individual or population passed when it is created. For each generation, the algorithm creates an entirely new population of individuals by selecting two individuals from the previous population then mating them to produce two new offspring for the new population. This process continues until the stopping criteria are met (determined by the terminator).

3.3.3.1 Encoding of a Chromosome

An important step in genetic algorithm is to “transform” the real problem into “biological terms”. This transformation is called chromosome encoding. Binary encoding is the most common and simplest one. In binary encoding every chromosome is represented by a binary string of bits, 0 or 1. Each bit in the string can represent some characteristics of the solution. For example:

Chromosome 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chromosome 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3-2: Encoding of a chromosome by binary string

3.3.3.2 Crossover

Crossover operation is carried out after encoding. Crossover operates on selected genes from parent chromosomes and creates new children. There are different techniques for producing crossover. The simplest one is single-point crossover, which is done by randomly choosing a point. All genes after that point in the chromosome are swapped between the two parent chromosomes. The resulting chromosomes are the children as illustrated in Figure 3-3.

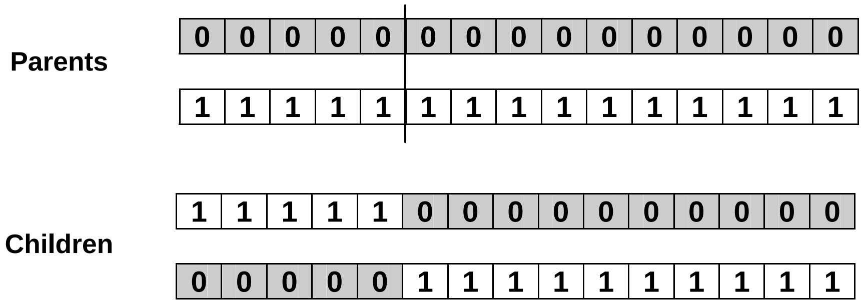


Figure 3-3: Single-point crossover

Another technique is two-point crossover, which is done by randomly choosing two points and all genes between the two points are swapped between the parent chromosomes, resulting in two child chromosomes as shown in Figure 3-4.

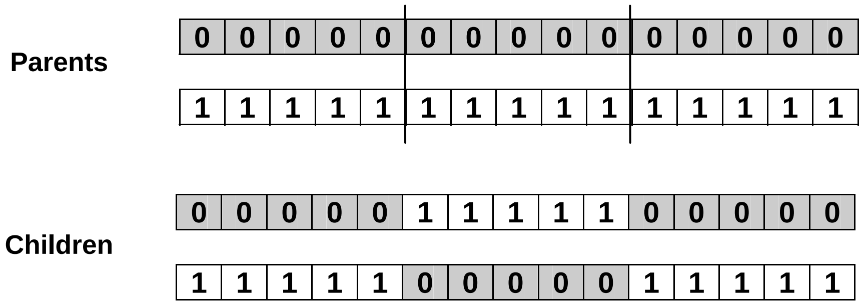
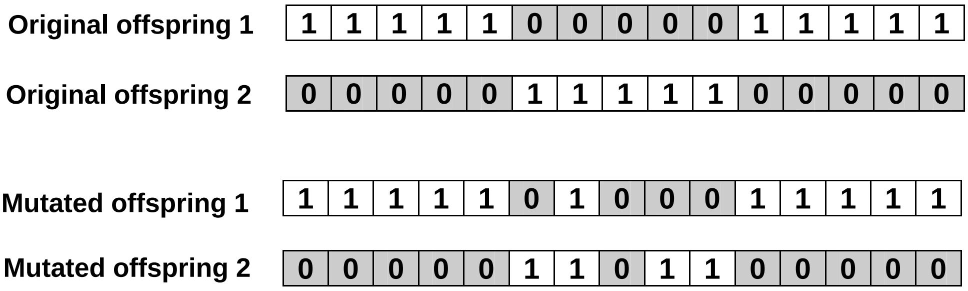


Figure 3-4: Two point crossover

3.3.3.3 Mutation

Mutation is the next step following crossover, and is done to prevent all solutions in the population becoming trapped in a local optimum of the solved problem. This is done by randomly altering the offspring produced in the crossover. For example, if binary encoding is used, a few bits are randomly swapped from 0 to 1 or from 1 to 0 as shown in Figure 3-5.



Original offspring 1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
Original offspring 2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
Mutated offspring 1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1
Mutated offspring 2	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0

Figure 3-5: An example of mutation

The probability with which a given chromosome switches its bits is called the mutation rate/probability. If only 1 bit is switched in e.g. offspring 1, then the mutation rate is 1/15 (0.067). A high mutation rate will essentially lead to a random search of the solution space.

3.3.3.4 Elitism

When performing crossover and mutation, there is a big possibility of losing the best individual. This problem is solved using a method called elitism. It first copies the best individual (or few best individuals) to the new population, while the rest of the population is created by crossover and mutation. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best solution.

3.3.3.5 Selection Scheme

The selection method determines how individuals are picked for mating. If one uses a selection method that chooses only the best individual, then the population will quickly converge to that individual. Therefore, the selector should be biased toward better individuals, but it should also pick some that aren't quite as good (but hopefully have some good new genetic material in them). Some of the more common selection methods include: roulette wheel selection, tournament selection and rank selection. [82]

In roulette wheel selection, parents are picked in accordance with their fitness. The more fitness the individuals/chromosomes have, the more chances to be picked they have. Figure 3-6 represents a roulette wheel where all the chromosomes in the population are placed. The size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is.

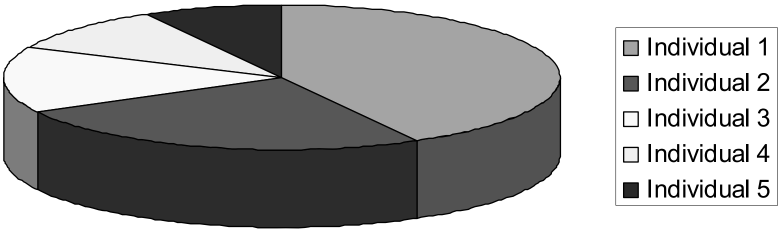


Figure 3-6: A roulette wheel (Source: [82])

When one throws a marble in the roulette wheel, the individual where it stops is picked. Obviously, the individual with bigger fitness value will be picked more times. This process can be described by the following steps:

1. Rank all individuals from the highest fitness to the lowest and calculate the probability of an individual being selected so that the summation of probabilities will be 1:

$$P_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{3-8}$$

Where:

P_i = The probability of individual being selected.

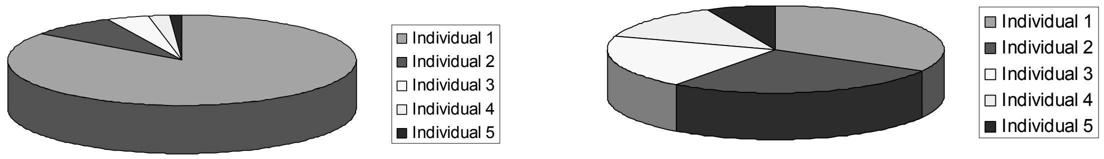
f_i = The fitness of individual *i*.

N = The number of individual in a population

2. Generate random number (r) from the interval (0, 1); and
3. Go through the ranked population and sum (S) the probabilities from 0. If the sum S is greater then r, stop and return the individual where you are.

If the probability of one individual (e.g. 0.85) is big in comparison with the summation of the probabilities of other individuals, these individuals will have very few chances of being chosen when applying roulette wheel selection. Rank selection is the method that solves this problem.[82]

In rank selection, the population is first ranked and then every individual is assigned a fitness value based on this ranking. The best individual is assigned fitness N (number of individual in population), the second best is assigned N-1 etc. and the worst will have fitness 1. Figure 3-7 shows how the situation changes after changing fitness to the numbers determined by the ranking.



Situation before ranking (graph of fitnesses) *Situation after ranking (graph of order numbers)*

Figure 3-7: Situation before and after ranking (Source: [82])

Now all the individuals have a chance of being selected. However this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

In tournament selection, a number “Tour” of individuals is chosen from the population based on roulette wheel selection and the best individual from this group is selected as a parent. This process is repeated as often as individuals must be chosen. The parameters for tournament selection are determined by the tournament size. It takes values ranging from 2 to the number of individuals in the population. If the tournament size is higher, weak individuals have a smaller chance of being selected. [49]

3.3.3.6 Fitness Scaling

Fitness scaling converts the raw scores that are returned by the objective function to values in a range that is suitable for the selection function. The selection function uses the scaled fitness values to select the parents of the next generation. The selection function assigns a higher probability of selection to individuals with higher scaled values.

The range of the scaled values affects the performance of the genetic algorithm. If the scaled values vary too widely, the individuals with the highest scaled values reproduce too rapidly, taking over the population gene pool too quickly, and preventing the genetic algorithm from searching other areas of the solution space. On the other hand, if the scaled values vary only a little, all individuals have approximately the same chance of reproduction and the search will progress very slowly. Some of the more common scaling methods include: linear scaling, sigma (σ) truncation and power law scaling. [49]

By linear scaling, the scaled fitness *f’* is calculated from the objective function value *f* using a linear equation of the form:

$$f' = af + b \tag{3-9}$$

Where a and b are calculated based upon the objective scores of the individuals in the population as described in. [49]

If the objective scores are negative, the sigma (σ) truncation method will be used. It scales based on the standard deviation (σ) and truncates arbitrarily at 0. The mapping from objective to fitness score for each individual is given by:

$$f' = f - (\bar{f} - c\sigma) \tag{3-10}$$

Where c is chosen between 1 and 3 and negative results are arbitrarily set to 0. $\bar{f}$ is the average of the objective scores of all individuals.

Power law scaling maps objective scores to fitness scores using an exponential relationship defined as:

$$f' = f^k \tag{3-11}$$

Where k in some studies has been assumed to be 1.005.

3.3.3.7 Population Size

The size of the population often affects the accuracy of GA [50]. A population size of five to a population size of tens of thousands can be used, depending on the nature of the problem that one is trying to solve. In a solution space of N possible solutions, a population of N individuals can solve the problem in 1 generation; however, N is often far too big to do that. Research shows that after some limit it is not useful to use very large populations because it does not solve the problem faster than moderate sized populations. Multiple-runs need to be conducted for each kind of problem to select the optimal population size. [82]

3.3.3.8 Termination Criteria

There are two criteria to terminate GA – (1) Convergence, and (2) Number of Generations. Convergence can be defined as the ratio of the N^{th} previous best-of-generation score to the current best-of-generation score. One can also define the maximum number of generations after which the GA evolution should stop.

4 An Offset Optimization Method for Signalized Urban Road Networks

A new offset optimization method for signalized urban road networks was developed in this dissertation that is capable of optimizing offsets in all traffic conditions. The method is converted prototypically into a C++ program. Examples of pseudo-code and c++ code for this program are provided in Annex A. The program consists of three main modules: (1) the input module, (2) the analysis module and (3) the optimization module. An outline of the three modules is presented in Figure 4-1. A description of the input module and an overview of each of the analysis and optimization modules are presented in this chapter. A detailed description of the analysis and optimization modules is provided in chapter 5 and 7 respectively.

4.1 Traffic Analysis Module

The traffic analysis module includes a delay analysis routine which is a time-space discrete traffic model based on Cell Transmission Model (CTM). CTM was originally developed by Daganzo [16] and [17]. This analysis routine defines the fitness or objective function for an optimization routine.

The CTM model has the following characteristics:

- It captures the entire range of fundamental diagram.
- It models the spatial as well as the temporal formation and can predict build-up, propagation and dissipation of queues.
- The calculation time for repeated simulation runs of module permits online application.

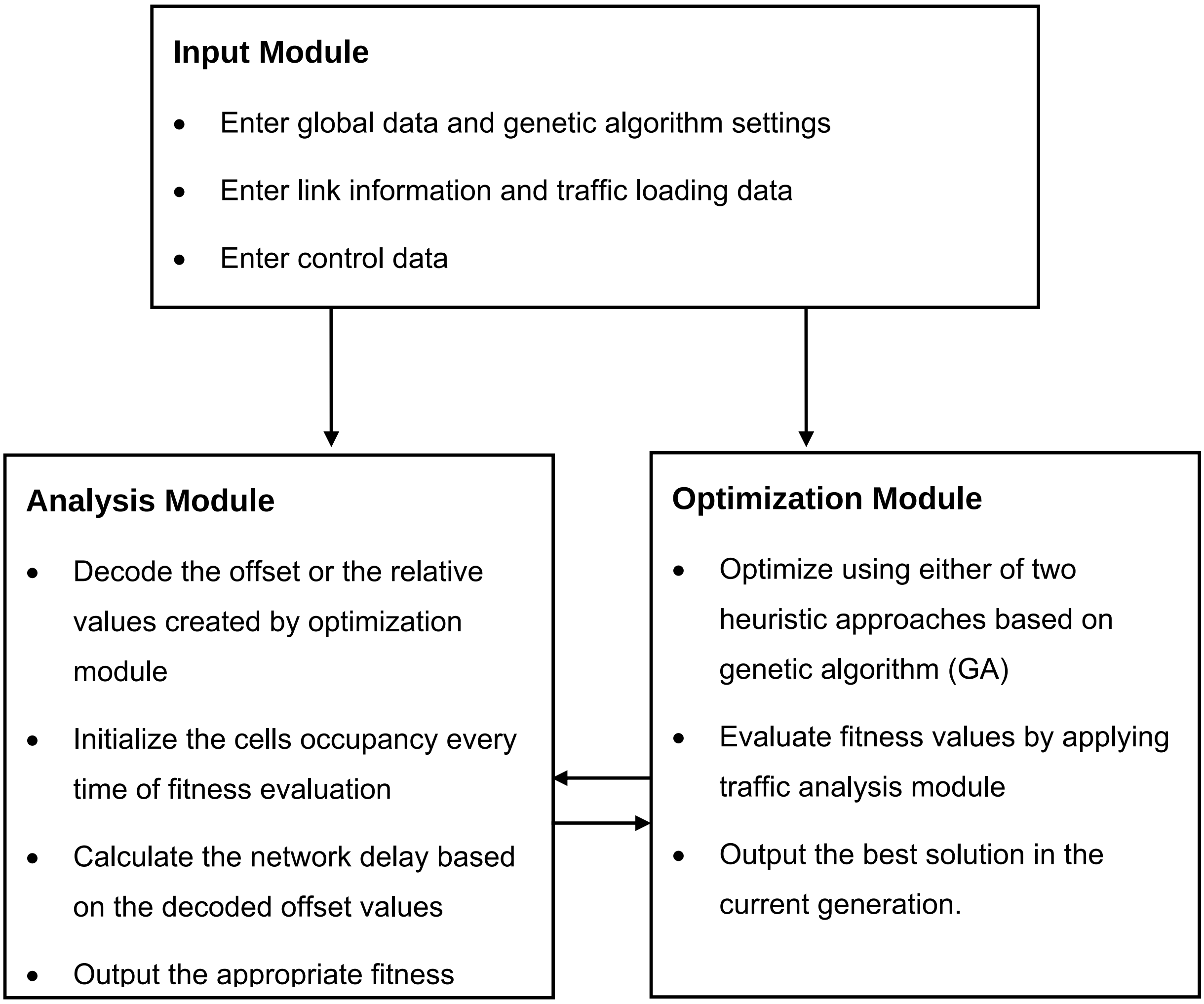

Input Module
• Enter global data and genetic algorithm settings
• Enter link information and traffic loading data
• Enter control data
Analysis Module
• Decode the offset or the relative values created by optimization module
• Initialize the cells occupancy every time of fitness evaluation
• Calculate the network delay based on the decoded offset values
• Output the appropriate fitness
Optimization Module
• Optimize using either of two heuristic approaches based on genetic algorithm (GA)
• Evaluate fitness values by applying traffic analysis module
• Output the best solution in the current generation.

Figure 4-1: An outline of the offset optimization method

The CTM employs a simplified version of fundamental diagram usually based on a trapezium form (see Figure 4-2) assuming that a free-flow speed v_f at low densities and a backward wave speed $-w$ for high densities are constant. In the CTM, the roads of the network are formed by the formation of successive cells (see Figure 4-3). These cells have lengths calculated by multiplying the free flow speed by the time step of analysis (1s). Delay calculation and a complete description of the model are provided in chapter 5.

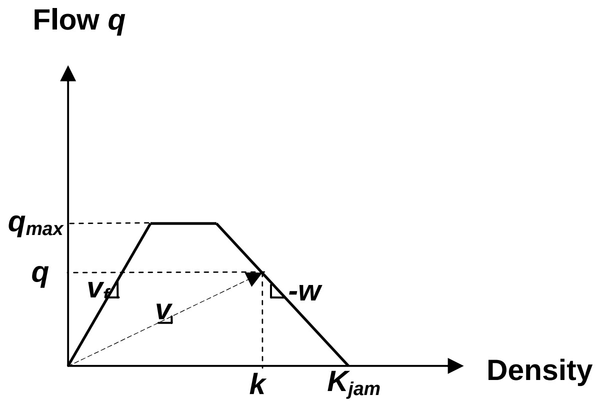
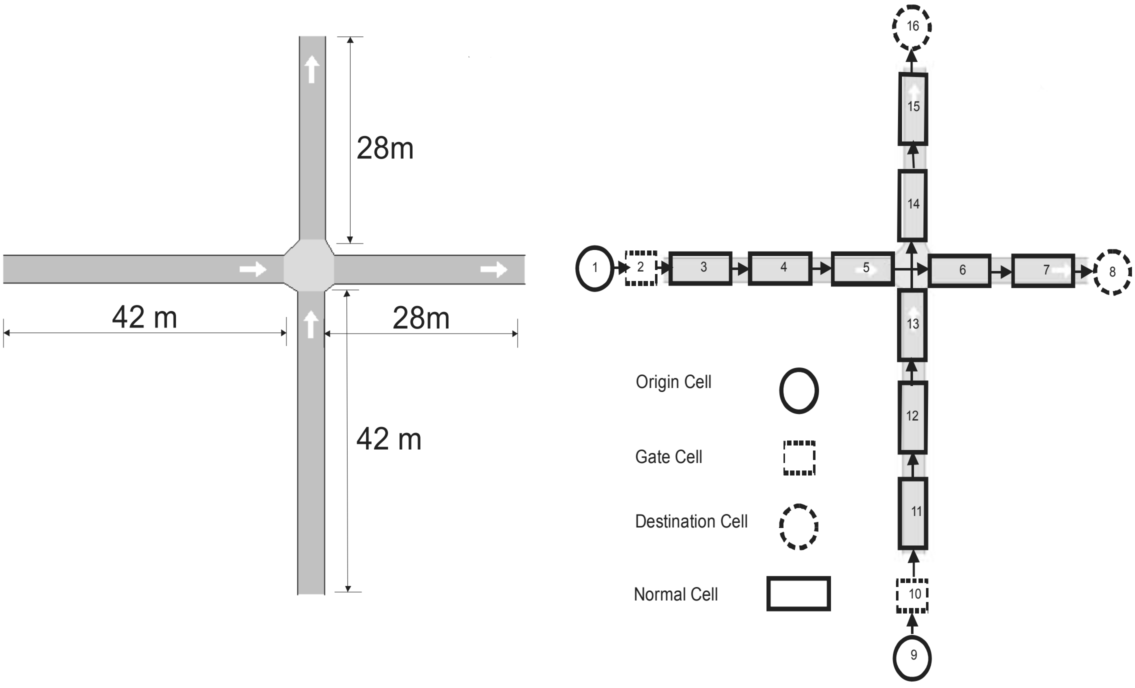


Figure 4-2: **The trapezoidal flow-density relationship used in the analysis module**



a) A simple intersection

b) Cell Representation

Figure 4-3: **A cell representation of a simple intersection**

4.2 Optimization Algorithms Module

The chosen objective for the optimization was the minimization of the total network delay, where delay is calculated using the analysis module. The proposed objective function is as follows:

$$f = \min \sum_{t} \sum_{j} d_j(t) \tag{4-1}$$

Where *f* is the sum of delays in all the cells of the network throughout the planning horizon. The objective is to select the offset values for all intersections such that *f* is minimized.

When one has fixed phase sequence, cycle times and green times for *n* signalized intersections in a road network, then one can have a number of $ct^{(n-1)}$ combinations (*ct* is the duration of the cycle time) for the solution space of offsets optimization. For small networks the optimal solution can be determined exactly – albeit with a high computation time – by complete enumeration. For larger networks and particularly for online use, this is no longer possible because of the required CPU time.

Since the objective function has an irregular shape in the solution space and therefore the classical search methods cannot be employed, two heuristic approaches based on genetic algorithm (GA) are developed in the optimization module.

The two approaches are different in their choice of search direction. In the first approach, referred to in this research as parallel genetic algorithm (PGA), a simultaneous search over all offsets (the entire chromosome) is performed using the reproduction-crossover-mutation process of variation. A convergence criterion or the computing time available ends the search. In the second approach, which is called serial genetic algorithm (SGA), a group of offsets and therefore only a part of the chromosome is varied until the best solution is found. In the next step, the offsets of the next group of intersections are optimized. In a serial search such as this, the order in which the intersections are treated and the offsets searched greatly influences the optimization results. A method has been developed for the determination of the search order. Background on genetic algorithm has been provided in chapter 3 while a complete description of the optimization module is presented in chapter 7.

4.3 Input Module

The input module of the program consists of a user interface through which input data can be entered. These input data consist of 5 groups, which are global, geometric, control, traffic loading and genetic algorithm data.

4.3.1 Global Data

The first input data consist of the following:

- The simulation time (s);
- The cycle time (s);

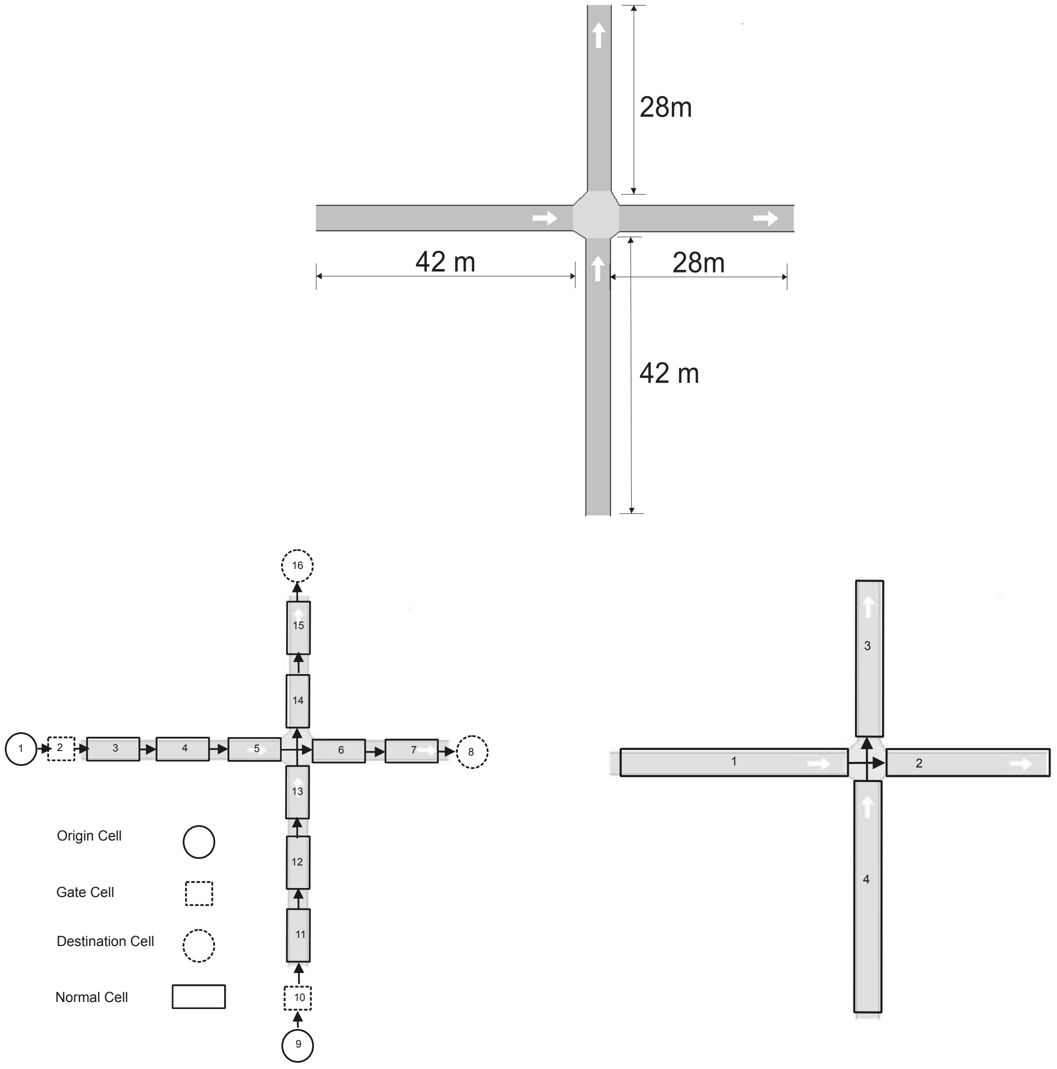
- The time step (s) of the simulation, which ranges between 0 to 5 s;
- The number of sections needed to build the network;
- W/V ratio, which is the ratio of back wave speed to the free flow speed;
- The assumed jam density (veh/km); and
- The saturation flow rate (veh/km).

4.3.2 Geometric Data

Although the geometric/space representation of the analysis model described in chapter 5 is based on dividing the network into cells, the input module uses the section instead of cells as a basis for network representation. This is for the purpose of input simplification as described in Figure 4-4; instead of having 16 elements when using cell representation, only 4 elements are required when using section representation. However the analysis model internally converts the section representation into cell representation.

The section input data consists of the following:

- The section ID;
- The initial occupancy proportion (set to 0 when the network is empty or a value less than 100% when the network is preloaded (%);
- The section length (m);
- The number of lanes in a section;
- The diverge ratio when the section is diverging;
- The ID of the upstream section (set to 0 when the considered section is input section);
- The ID of the second upstream section (set to 0 when no second upstream section exists);
- The ID of the downstream section (set to 0 when the considered section is output section);
- The ID of the second downstream section (set to 0 when no second downstream section exists);
- The free flow speed at the section (km/h);
- The Saturation flow adjustment factor;



a) Cell Representation

b) Section Representation

Figure 4-4: **Cell and section representation**

4.3.3 Control Data

The control data includes the following (See Figure 4-5):

- The number of signalized intersections;
- The intersection ID;
- The initial offset for each intersection;
- The number of phases for each intersection;
- The time assigned for each phase at each intersection;

- The number of signal groups in each intersection (e.g. in intersection 1, 3 signal groups);
- The signal group ID (e.g. 1, 2 or 3 in intersection 1);
- Which intersection this signal group belongs to;
- Which phase this signal group is assigned to (e.g. signal groups 1 and 2 are assigned to phase 1 in intersection 1);
- How many sections are assigned to this signal group and the ID's of these sections (e.g. to signal group 1 in intersection 1: 2 sections are assigned, which are sections 2 and 3)

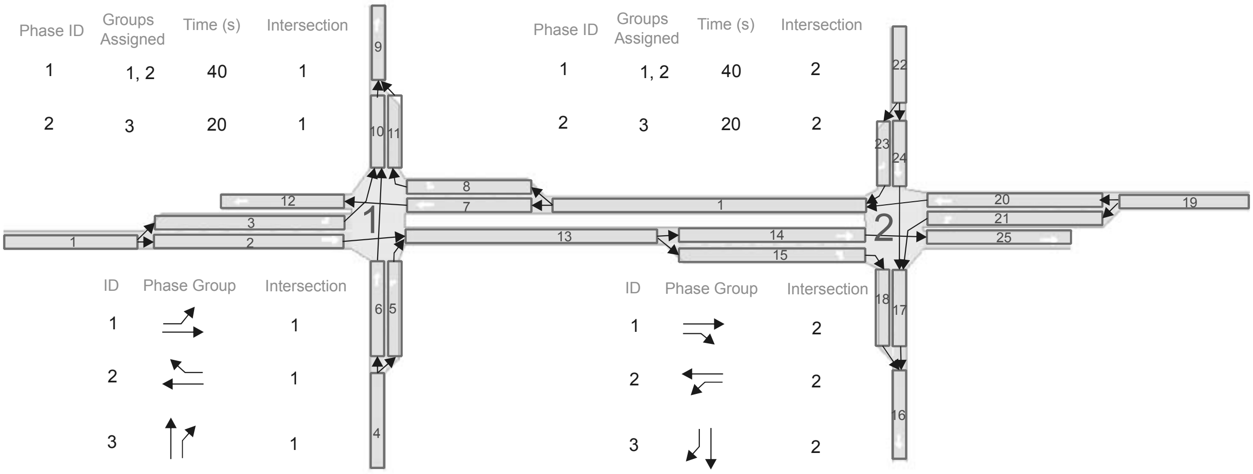


Figure 4-5: Description of control data

4.3.4 Traffic Loading

The traffic loading information includes the following:

- The number of loading sections (input sections);
- The volume adjustment factor;
- Which section is loaded; and
- The loading for each input section (veh/h).

4.3.5 Genetic Algorithm Settings

The genetic algorithm settings include the following:

- A query for using PGA or SGA ;
- A query for elitism;

- The number of generations;
- The population size;
- The mutation rate; and
- The crossover rate.

5 Traffic Analysis Module Based on Cell Transmission Model

This chapter describes the traffic analysis module, which is applied as the fitness function for the GA-based optimization algorithms. As stated in the previous chapter, a delay analysis routine is included in the analysis module which is a time-space discrete traffic model based on Cell Transmission Model (CTM). A detailed description of this analysis model is presented in this chapter.

5.1 The Cell Transmission Model

The approaches of macroscopic traffic modelling can be classified into the simple continuum model and the high order continuum model based on the assumption of the speed equation. First order models are based on the LWR model (or Lighthill and Whitham [67] model and Richards [86]) derived from flow-fluid analogy to represent traffic along directed road links in time and space by three variables: density *k,* flow *q*, and speed *v*. Mongoeot [78] gives an overview of the first-order macroscopic approach. Second-order models have been proposed to try to correct some of the deficiencies of the LWR model (see, for example, Payne [85]). Daganzo [18] pointed out that all existing models of traffic flow formulated as a quasi linear system of partial differential equations smooth out all discontinuities in density, with the exception of the LWR model, they all predict negative speeds under certain conditions.

A basic assumption of the LWR model is that vehicles are not created or lost along the road. This conservation law of the number of vehicles leads to the continuity equation:

$$\frac{\partial k}{\partial t}+\frac{\partial q}{\partial x}=0 \tag{5-1}$$

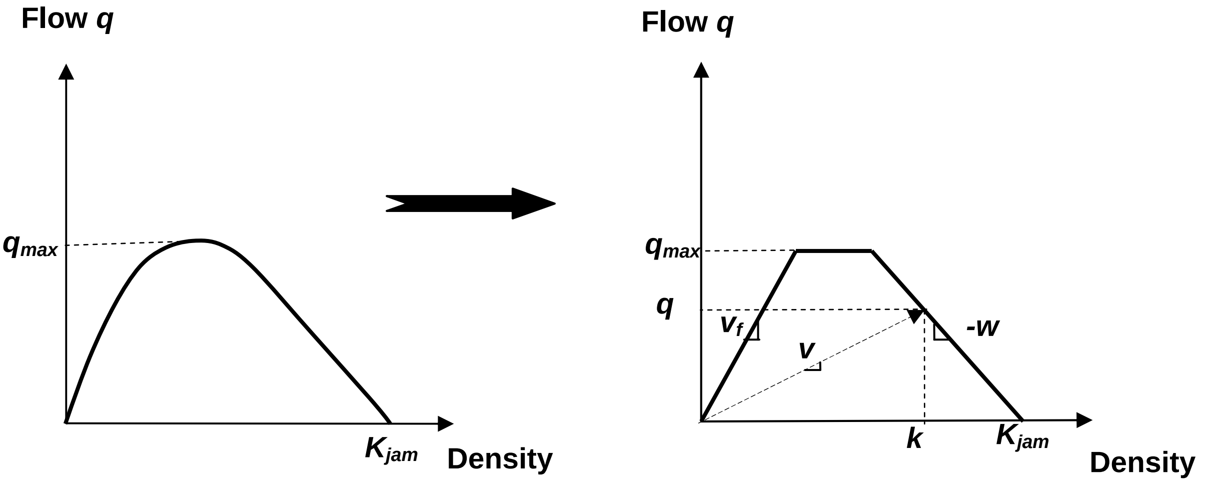
The second basic condition for the LWR is a function *f* related to q and k as follows:

$$q=f(k,x,t) \tag{5-2}$$

Where q is the traffic flow, k is the density, x is the space variable and t is the time variable. The relation *f* is a fundamental relationship in traffic flow theory. Given a set of well-posed initial conditions, one can determine q and k at any (x, t) by solving (5-1) and (5-2). LIGHTHILL and WHITHAM [67] and Newell [81] developed two different solution approaches to this model. Daganzo ([16] and [17]) further simplified the solution scheme by adopting the following relationship between q and k:

$$q = \min\left\{v_f k, Q, w(k_{jam} - k)\right\} \tag{5-3}$$

Where k_{jam}, Q, $_f$, and w denote, respectively, the jam density, inflow capacity (or maximum allowable inflow), free flow speed and the speed of the backward wave. Essentially, equation (5-3) approximates the flow density relationship (fundamental diagram) into trapezium form shown in Figure 5-1.



(a) Fundamental Diagram

(b) The Flow Density Relationship for the CTM

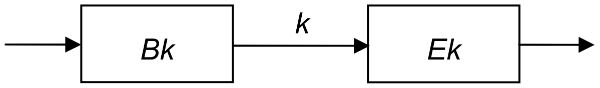
Figure 5-1: Fundamental diagram and the Flow Density Relationship for the CTM

5.2 Network Modeling and Topologies

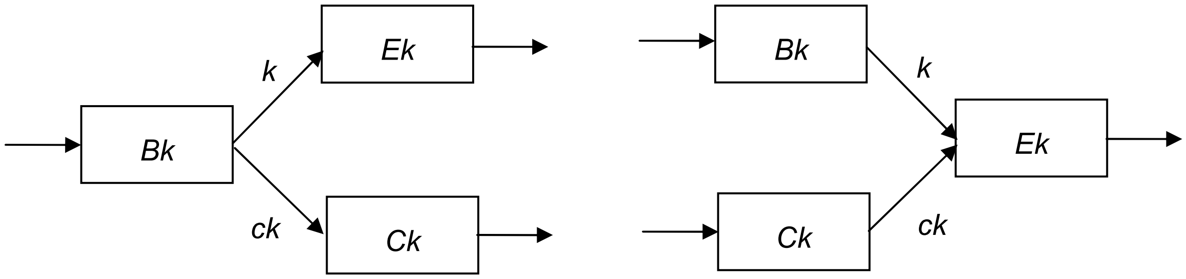
In the CTM, the roads of the network are formed by the formation of successive sections or cells. The CTM uses the common modeling of traffic networks by directed graphs with nodes and links. In this system, the cells are described as the nodes {I} and the possible vehicle transfers by a set of links {k}. To define the topology of the network, a beginning (upstream) and an ending (downstream) cell for each link k is specified. The prefixes B and E are added to the link label, so that the beginning and ending cells of link k become Bk and Ek respectively.

CTM limits the number of links connecting a cell to three. This results in the following three types of connections:

- The simple or normal connection shown in Figure 5-2 a, where one link enters a cell and one leaves it.
- The diverge connection shown in Figure 5-2 b, where one link enters a cell and two links diverge from it.
- The merge connection shown in Figure 5-2 c, where two links merge to a cell and one link leaves it.



a) normal



b) diverge

c) merge

Figure 5-2: Modeling of normal, diverge, and merge links

5.3 Traffic Flow Calculation

5.3.1 Traffic Flow on Normal Link

Dividing the road into homogeneous sections (or cells) and time into intervals of duration T such that the cell length L is equal to the distance traveled by free-flowing traffic in one time interval, the LWR results are approximated by a set of recursive equations:

$$n'_{Bk}(t+1) = n_{Bk}(t+1) - q_k(t)\, ;\, n'_{Ek}(t+1) = n_{Ek}(t+1) + q_k(t) \qquad (5\text{-}4)$$

$$q_k(t) = \min\{n_{Bk}(t), \min[Q_{Bk}(t), Q_{Ek}(t)], (w/v_f)(N_{Ek}(t) - n_{Ek}(t))\} \qquad (5\text{-}5)$$

The variables n(t), N(t), q(t), and Q(t) denote the actual number of vehicles (occupancy), the maximum number of vehicles (or holding capacity), the actual inflow, and the inflow capacity that can be presented in cell B or E at time t, respectively.

With equation (5-5), the entire range of traffic densities described in Figure 5-1b of the simplified fundamental diagram is considered in the calculation of traffic flow. Based on the value of traffic density, three cases of traffic conditions are considered:

- The first case occurs when the density is low (free flow condition), all vehicles can travel from cell Bk to cell Ek at time t.
- The second case occurs when the traffic demand becomes high resulting in increasing density, then the traffic flow is limited by the smaller value of the capacities $Q_{Bk}(t)$ or $Q_{Ek}(t)$.
- The third case occurs when the density of the downstream cell is high, then not all vehicles can travel from cell Bk to cell Ek. In this situation, the traffic flow is limited with the free space available ($N_{Ek}(t)-n_{Ek}(t)$) in the downstream cell Ek.

Equation (5-5) can be simplified by the following two equations:

$$S_{Bk}(t) = min\{Q_{Bk}(t), n_{Bk}(t)\} \text{ and}$$
$$R_{Ek}(t) = min\{Q_{Ek}(t), (w/v_f)(N_{Ek}(t) - n_{Ek}(t))\} \qquad (5\text{-}6)$$

$$q_k(t) = min\{S_{Bk}(t), R_{Ek}(t)\} \qquad (5\text{-}7)$$

where the actual flow on link *k* $q_k(t)$ is the minimum of sending $S_{Bk}(t)$ and receiving $S_{Bk}(t)$.

After the flows have been determined for each cell for a specified time step, the occupancies can be easily updated with the conservation equation (5-4) by taking the link flows away from the beginning cells and adding them to the ending cells. The occupancies, n`(t+1), are intermediate variables introduced; they can be eliminated during computer operation.

5.3.2 Traffic Flow on Diverge Links

In Figure 5-2 b, cell BK is the beginning of divergence into either cell EK or cell CK. The maximum number of vehicles $S_{BK}(t)$ that can be sent from cell BK and the maximum number of vehicles ($R_{EK}(t)$ and $R_{CK}(t)$) that can be received by cells EK and CK respectively are determined using equation (5-6).

The proportions of vehicles β_{Ek} and β_{Ck} leaving cell BK are assumed to be constant, where $\beta_{Ek}+\beta_{Ck}=1$. Since part of the flow leaving cell BK is diverging into cell CK and part into cell EK, all flow is impeded if either one of the diverging links is unable to accommodate its share of flow in order to maintain a first-in-first-out queuing regime. This means that vehicles unable to exit will prevent all those behind, regardless of destination, from moving. Considering these conditions, the flow that exits from BK can be determined by the following equation:

$$q_{Bk}(t) = \min\{S_{Bk}, R_{Ek}/\beta_{Ek}, R_{Ck}/\beta_{Ck}\} \tag{5-8}$$

The flows on both diverging links can be determined as follows:

$$q_k(t) = \beta_{Ek} q_{Bk} \text{ and } q_{Ck}(t) = \beta_{Ck} q_{Bk} \tag{5-9}$$

5.3.3 Traffic Flow on Merge Links

As shown in Figure 5-2c, cells B and C are the start of a merge to cell E. S_{BK} and S_{CK} are the outflows (Sending) that can be sent from cells B and C respectively. R_{Ek} is the inflow (Receiving) ability of cell E. Depending on whether the receiving ability of Ek is less or more than the sending ability of Bk and Ck, three cases are considered in order to build the equations for the flows on the two links:

Case 1: If the receiving ability R_{Ek} is more than the sending ability of the two cells S_{Bk} + S_{Ck}, the actual outflows are determined as follows:

$$\textit{for } R_{Ek} \geq S_{Bk} + S_{Ck}: \quad q_k(t) = S_{Bk} \textit{ und } q_{Ck}(t) = S_{Ck} \tag{5-10}$$

Case 2: If the receiving ability is less than the sending ability of the two cells, then the actual outflows are determined as follows:

$$\textit{for } S_{Bk} > R_{Ek} p_k \wedge S_{Ck} > R_{Ek} p_{Ck}: \quad q_k(t) = p_k R_{Ek} \textit{ and } q_{Ck}(t) = p_{Ck} R_{Ek} \tag{5-11}$$

The constant p_k is the proportion of vehicles that comes from BK and the remainder p_{ck} from Ck, where p_k+p_{ck}=1, and p_k/p_{ck} = q_k/q_{ck}. These constants are the characteristics of merge junctions that describe any priority and are determined based on the capacity proportions of the merge cells.

Case 3: If the sending ability of one of two cells is limited by the receiving ability, then the actual outflows are determined as follows:

$$\textit{for } S_{Bk} < R_{Ek} p_k \wedge S_{Ck} > R_{Ek} p_{Ck}: \quad q_k(t) = S_{Bk} \textit{ and } q_{Ck}(t) = R_{Ek} - S_{Bk} \tag{5-12}$$

$$\textit{for } S_{Bk} > R_{Ek} p_k \wedge S_{Ck} < R_{Ek} p_{Ck}: \quad q_k(t) = R_{Ek} - S_{Ck} \textit{ and } q_{Ck}(t) = S_{Ck} \tag{5-13}$$

For the cases 2 and 3, Daganzo [17] has proved that the flows on links k and ck can be determined by following simple equations:

$$\textit{for} \qquad R_{Ek} < S_{Bk} + S_{Ck}: \tag{5-14}$$

$$q_k(t) = mid\left\{S_{Bk}, R_{Ek} - S_{Ck,} p_k R_{Ek}\right\}$$

$$q_{Ck}(t) = mid\left\{S_{Ck}, R_{Ek} - S_{Bk,} p_{Ck} R_{Ek}\right\}$$

5.4 Traffic Signal Control and Delay Estimation

5.4.1 Traffic Signal Control

Until now the established characteristic quantities (Q, N, S, R, ß, p) were assumed as constant values, however, they also can be time variant. For example, a time variant capacity Q (t) can be used to model inflow controls. Traffic signal control can be modeled by the restriction of the capacity in the controlling cells of the road junction or by the restriction of a time variant p(t). In this case p takes the value 0 when the signal group is closed or 1 when the signal group is released. Then the traffic flow equation is:

$$\text{for } p_k(t) = 0: \qquad q_k(t) = 0 \tag{5-15}$$

$$\text{for } p_k(t) = 1: \qquad q_k(t) = \min\left\{S_{Bk}, R_{Ek}\right\} \tag{5-16}$$

With the discrete representation of time and place, one can quantify the effect of the variation of all control variables (cycle time, split, green time and offset).

5.4.2 Delay Estimation

In the flow density diagram shown in Figure 5-1 b, the slope of the line drawn from the origin represents the actual speed *$v_I(t)$* in cell I at time t which is equal to the outflow divided by the density as follows:

$$v_I(t) = \frac{q_k(t)}{k_I(t)} \tag{5-17}$$

where:

$$k_I(t) = \frac{n_I(t)}{L} \tag{5-18}$$

The relationship between the delay of one vehicle in cell I at time t and the actual speed of this vehicle is assumed to be linear as shown in Figure 5-3:

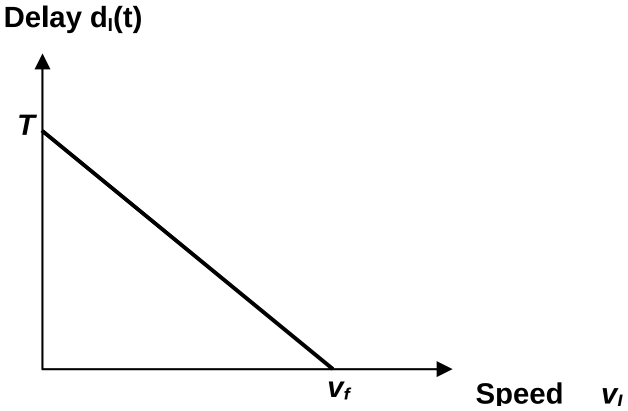


Figure 5-3: The delay-speed relationship

Based on this relationship, delay ranges from 0 when the actual speed is equal to v_f to the duration of T (time step) when the vehicles are not moving. This delay times the number of vehicles $n_I(t)$ determines the total delay of all vehicles in cell I at time t:

$$d_I(t) = \left[T - \left(\frac{T}{v_f} \times v_I(t)\right)\right] \times n_I(t) \tag{5-19}$$

where:

$$v_f = \frac{L}{T} \tag{5-20}$$

By substituting equations (5-17), (5-18), and (5-20) in equation (5-19) the following equation is obtained:

$$d_I(t) = T\left[n_I(t) - q_k(t) \times T\right] \tag{5-21}$$

Once delay has been determined at the cell level, the total delay D_{link} can easily be estimated at the link level by summing up all delays in all cells during the studied time period as follows:

$$D_{link} = \sum_t \sum_I^{link} d_I(t) \tag{5-22}$$

5.5 Cell Representation and Boundary Conditions

Consider an intersection of one-way streets without turning movements shown in Figure 5-4. This intersection consists of two input sections with a length of 42m and two exit sections with a length of 28m. When the free flow speed is 50 kph and the simulation step is 1s, then the length of each cell in the network is 13.89m ((50/3.6) × 1). Therefore, the input and exit sections should have 3 and 2 cells respectively (see Figure 5-5).

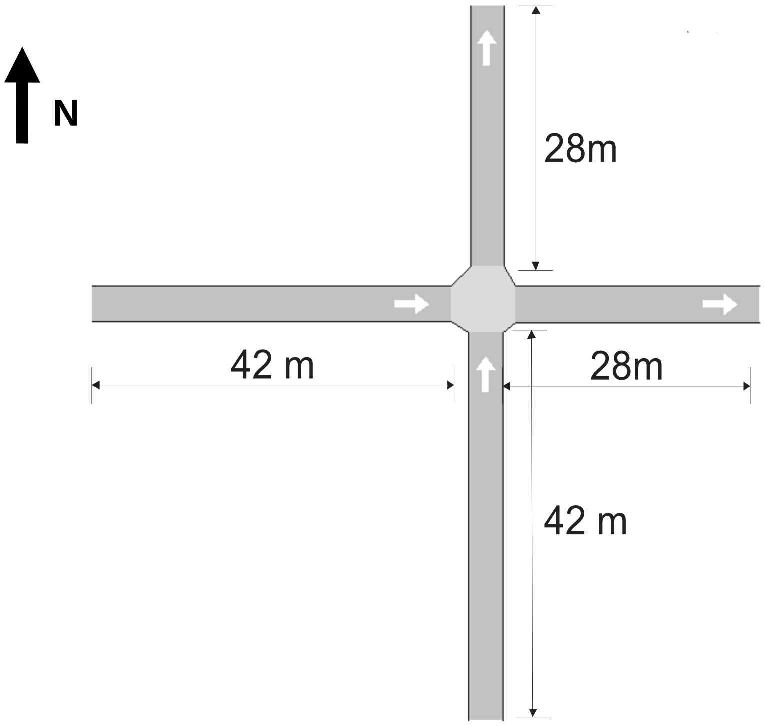


Figure 5-4: A simple example of one intersection without turn lanes

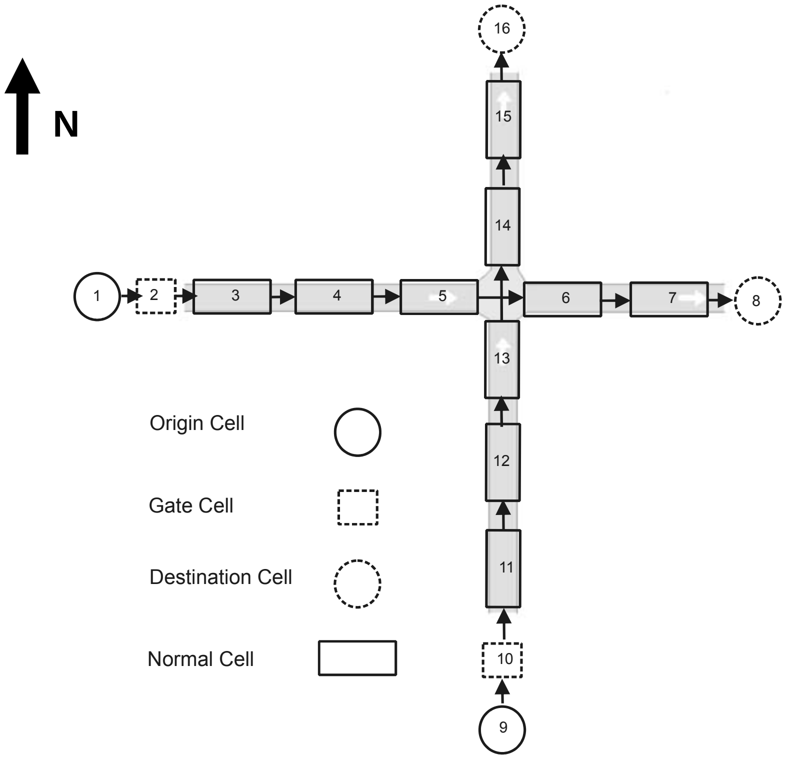


Figure 5-5: Cell representation of an intersection without turn lanes

To describe the cell representation when turning movements are allowed, consider an intersection shown in Figure 5-6. This intersection consists of two input sections with a length of 42m (having turn bays with a length of 28m) and two exit links with a length of 28m. When the free flow speed is 50 kph and the simulation step is 1s, then the length of each cell in the network is 13.89m ((50/3.6) × 1). Therefore, the input sections, turn bays and exit sections should have 3, 2 and 2 cells respectively.

Origin, destination, and gate cells are used to specify boundary conditions as follows:

- Origin cells (e.g. cells 1and 9 in Figure 5-5 and in Figure 5-7) must have an infinite number of vehicles ($n = \infty$) that discharge into empty "gate" cells.
- Gate cells (e.g. cells 2 and 10 in Figure 5-5 in Figure 5-7) must have an infinite size ($N = \infty$) and the inflow capacities of the cells $Q(t)$ are set equal to the desired section input flow for time interval t .
- Destination cells (e.g. cells 8 and 16 in Figure 5-5 in Figure 5-7), where traffic flows terminate and exit the network, should have infinite sizes ($N = \infty$).

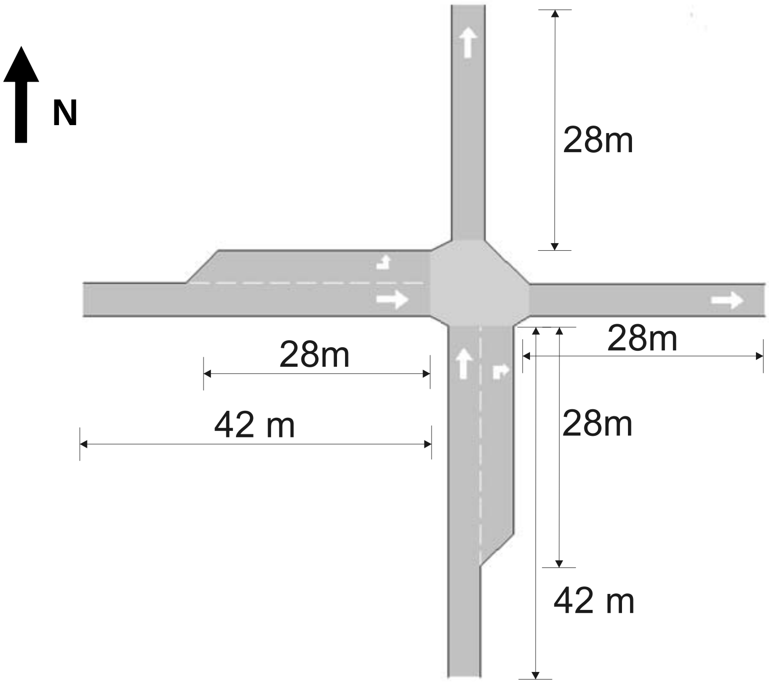


Figure 5-6: A simple example of one intersection with turn lanes

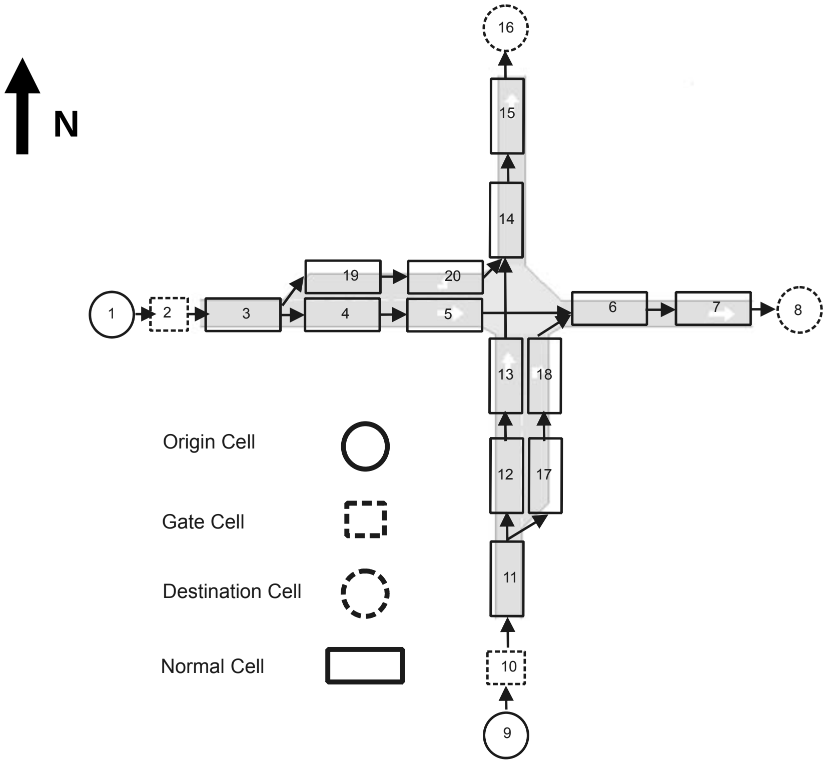


Figure 5-7: **Cell representation of an intersection with turn lanes**

5.6 Calculation Procedure

Prior to starting the calculation of the cell occupancies (n), the inflow (q) and the delays, a conversion from section representation to cell representation must be done. The calculation process of the developed simulator based on CTM is presented in a pseudo-code shown in Figure 5-8 and described in the following steps:

1. Initialize the cell occupancy (n) based on the input data of initial occupancy proportion (n/N) either with zeros when the network is empty or with definite values when the network is preloaded.
2. Calculate the flows q(t) for the first time step in all cells using the already described flow equations without any restriction of cell order.
3. Calculate the delays d(t) for the first time step in all cells using the already derived delay equations also without any restriction of cell order.
4. Calculate the cell occupancies n(t+1) for the next time steps in all cells.
5. Repeat steps 2-4 for each time step till the end of the time horizon.

6. Sum up the delays in all cells in all time steps to get the system/network delay.

```
// Initialization
Initialize $n_l$ for all cells with 0 or percentages of $N_l$
// Loop
For time t = 0 to the time horizon
        For cell l = 1 to number of cells
                Calculate flow q(t) at time t for all cells ( No order of calculation)
        End loop
        For cell l = 1 to number of cells
                Calculate delays d(t) for all cells ( No order of calculation)
        End loop
        For cell l = 1 to number of cells
                Calculate number of vehicles n(t+1) for all cells ( No order of calculation)
        End loop
End loop
```

Figure 5-8: Pseudo-code for the calculation steps of flows and number of vehicles

6 Validation and Comparison of the Traffic Model

The analysis model presented previously had to be validated before applying it as the fitness functions for the offset optimization model. This chapter presents the validation procedures applied to the delay estimation. A virtual environment was prepared for that purpose by means of microscopic simulation (AIMSUN) and was used as reference measurements of delay. The model validation was conducted at an isolated intersection, a one-way street with two intersections and at a realistic network with 6 intersections.

6.1 Delay Validation at an Isolated Intersection

In this section, the validation of the developed analysis model is carried out at an isolated intersection. For this purpose, a four-way intersection existing in the List District of Hanover city was modelled using AIMSUN (see Figure 6-1). The simulated intersection is fixed time controlled by a 90 second cycle and the green time of the analysed lane is 32 seconds. The validation is carried out by comparing the developed analysis model against three equilibrium queuing models, which are the Kimber-Hollis model [63], Akçelik Model [2] and HCM model [80]. The comparison was conducted against these models by simulating 6 different scenarios of traffic demand (see Table 6-1).

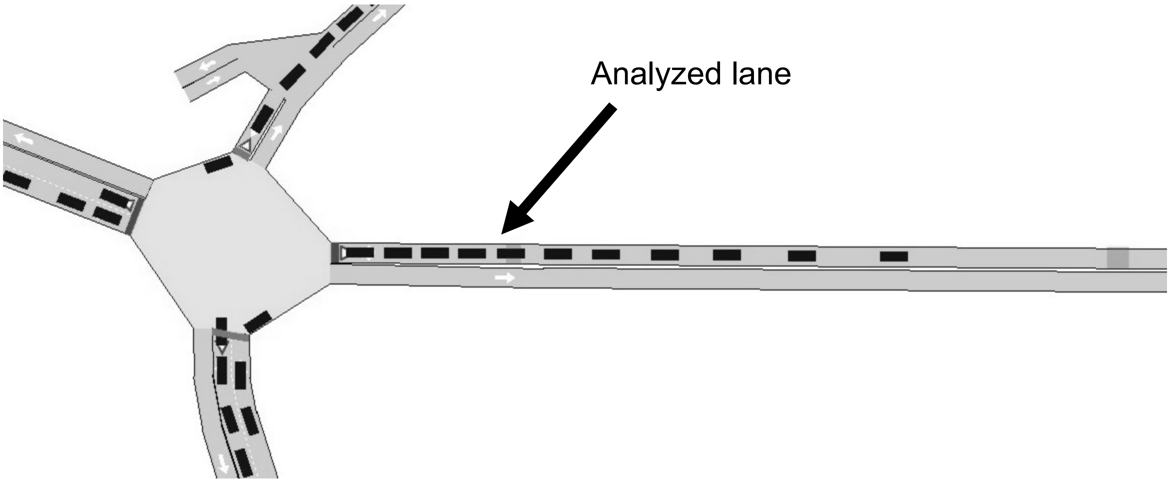


Figure 6-1: **A four-way intersection located in List District of Hanover City**

These scenarios were chosen so that they reflect undersaturated and oversaturated traffic conditions. The degree of saturation presented in the table is based on the varied demand (every 15 minutes) and the saturation flow rate.

Table 6-1: **Scenarios and corresponding degrees of saturation x for queue comparison**

Scenario	Degree of Saturation x(in periods of 15 min.)
1	0.50 / 0.80 / 1.00 / 1.20
2	0.50 / 1.20 / 0.50 / 1.20
3	0.50 / 0.80 / 1.00 / 0.80 / 0.50
4	0.80 / 1.00 / 1.20 / 1.00 / 0.80
5	1.00 / 1.20 / 1.20 / 1.20 / 1.00
6	0.05 / 0.10 / 0.20 / 0.35 / 0.50 / 0.65 / 0.80 / 0.90 / 1.00 / 1.10 / 1.25 / 1.40

The delay was calculated every 90 s (cycle time), therefore the flow was counted in defined intervals equal to the cycle time for all comparison methods except for the developed model which was done every 5 s.

To evaluate and compare the estimated queue lengths L_{est} based on the developed model and the well-known queuing models against the real ones L_{real} obtained from AIMSUN, different statistical indicators were used as follows:

the coefficient of determination

$$R^2 = 1 - \frac{\sum\left(L_{real} - L_{est}\right)^2}{\sum\left(L_{real} - \sum L_{real}/N\right)^2} \quad [-], \qquad (6\text{-}1)$$

the root mean square error:

$$RMSE = \sqrt{\frac{\sum\left(L_{real} - L_{est}\right)^2}{N}} \quad [veh] \quad and \qquad (6\text{-}2)$$

the relative root mean square error:

$$RRMSE = \frac{RMSE \cdot 100\%}{\left(\frac{\sum L_{real}}{N}\right)} \quad [\%]. \qquad (6\text{-}3)$$

Table 6-2 reports the values of the three indicators for the delay estimation models. From this table, it could be ascertained that in almost all scenarios the developed model gives the best results. In scenario 1, Kimber and Hollis (KH) and HCM produce **R^2** almost as well as the developed model, but when considering the other criteria it performs better in this scenario.

Table 6-2: Comparison between the estimated delay to the real one

	Model	**KH**			**developed Model**			**HCM**			**Akcelik**		
		R^2	**RMSE**	**RRMSE**	**R^2**	**RMSE**	**RRMSE**	**R^2**	**RMSE**	**RRMSE**	**R^2**	**RMSE**	**RRMSE**
Scenarios	**1**	0.92	81.5	128.7	0.96	9.9	24.7	0.98	63.7	67.6	0.59	19.6	25.5
	2	0.45	128.8	311.1	0.74	33.1	80.0	0.85	78.3	189.2	0.38	37.7	90.9
	3	0.38	60.3	218.1	0.58	9.7	34.9	0.55	18.9	68.3	0.46	6.9	25.1
	4	0.42	255.1	328.0	0.82	25.1	33.1	0.64	175.4	205.3	0.16	61.4	46.2
	5	0.91	302.6	157.1	0.90	62.8	32.7	0.86	99.0	41.7	0.01	135.0	62.4
	6	0.72	187.7	550.4	0.48	30.4	89.2	0.79	213.9	627.1	0.39	27.5	80.6

An example of graphical presentation for scenario 4 is presented in Figure 6-2 which shows the comparison of the cyclic vehicular delays (veh.s/90s) for different delay estimation models and the corresponding degree of saturation (x). Since AIMSUN generates the flow randomly, the degree of saturation calculated every 90s was a bit different from that already presented in Table 6-1.

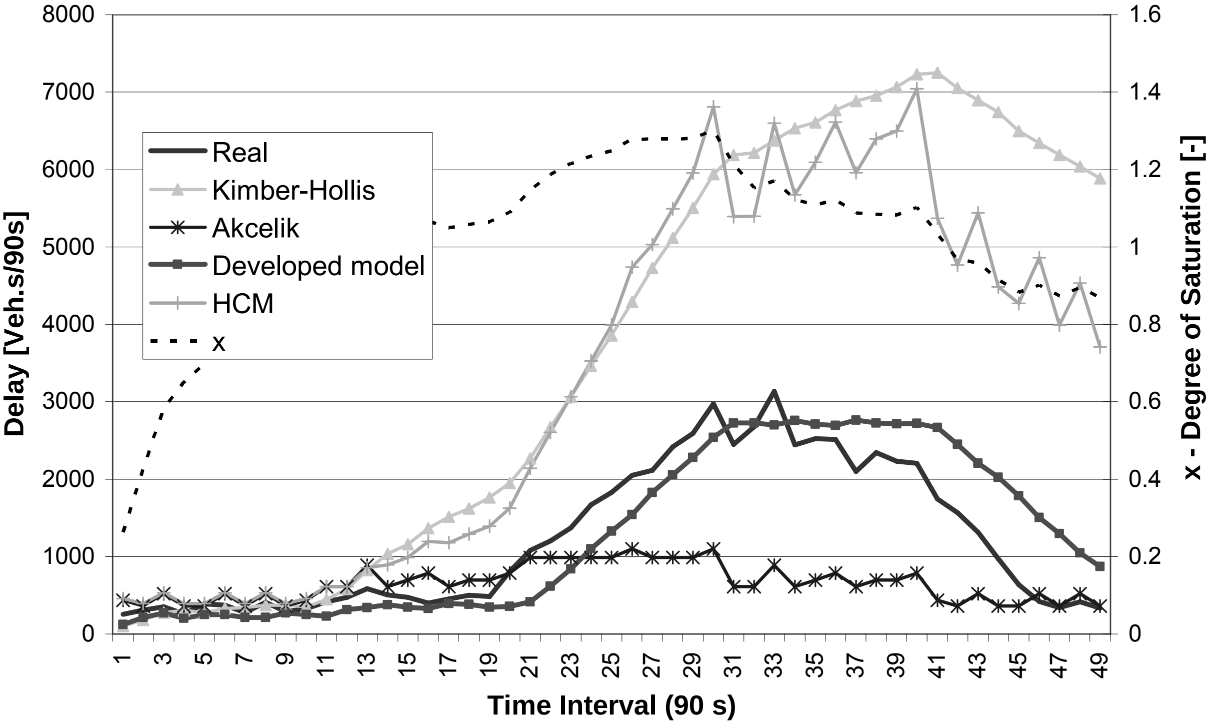


Figure 6-2: **An example (scenario 4) of the comparison of the total vehicular cyclic delay for the different queuing models.**

Figure 6-2 shows that, for estimating the total delay, Kimber-Hollis (KH) and HCM give good results for relatively low degrees of saturation, however when the degree of saturation increases these methods overestimate the reference value. This problem is not observed in the developed model. In contrast, Akcelik's method gives good results for relatively low degree of saturation, however when the degree of saturation increases the method underestimates the reference value. This problem is also not seen in the developed. Given this context, the developed analysis model is superior to the well know queuing model for delay estimation, and its results matched rather well with AIMSUN's results for both undersaturated and oversaturated conditions.

6.2 Delay Validation at a Two-Intersection Street

For the delay estimation at an isolated intersection any change of offset value at this intersection has no effect on the delay at links. However, when the delay is estimated at a street with more than one intersection, the offset values affect the delay significantly. Therefore, a one way street with two intersections was studied to validate the delay estimation of the developed model when changing the offsets.

Figure 6-3 shows the studied example with the considered link for analysis. The flow in the east direction is 1300 veh/h and in the north direction is 1600 veh/h at both intersections. Here are the assumed parameters:

- Intergreen time is 5 s;
- Cycle time is 120;
- Green time is 49s in east direction at each intersection;
- Green time in north direction at each intersection is 61s;
- Free flow speed and backward wave speed are 50 km/h;
- Jam density is 150 veh/km;
- Saturation flow is 1800 veh/h;
- Time interval is 1 second;
- Modeling horizon is 8 cycles (equivelent to 960 time intervals).

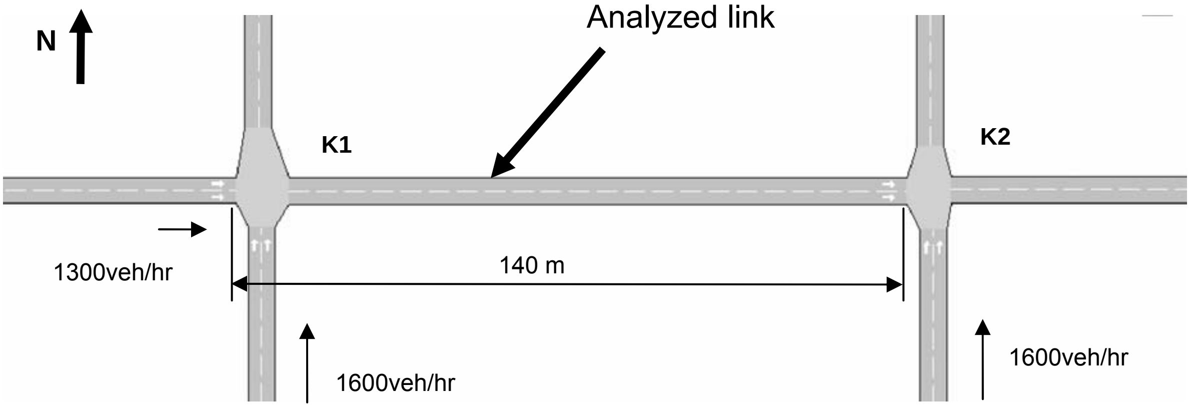


Figure 6-3: A one-way street with 2 signalized intersection

To study the effect of varying offset values on the delay estimation, the offset at intersection K2 was varied from 0 to the cycle time (120s) with a step of 10s while the offset at intersection K1 was fixed to 0. At each value of varying offset, the system was simulated with a total of 30 replications and the average of the delays of these replications was used for the comparison.

Figure 6-8 shows the relationship between offset and delay for the analyzed link as estimated by AIMSUN and the developed model. The Figure shows that the delay values of the developed traffic model are similar to AIMSUN. A high regression coefficient R^2 of 0.98 was obtained in this comparison.

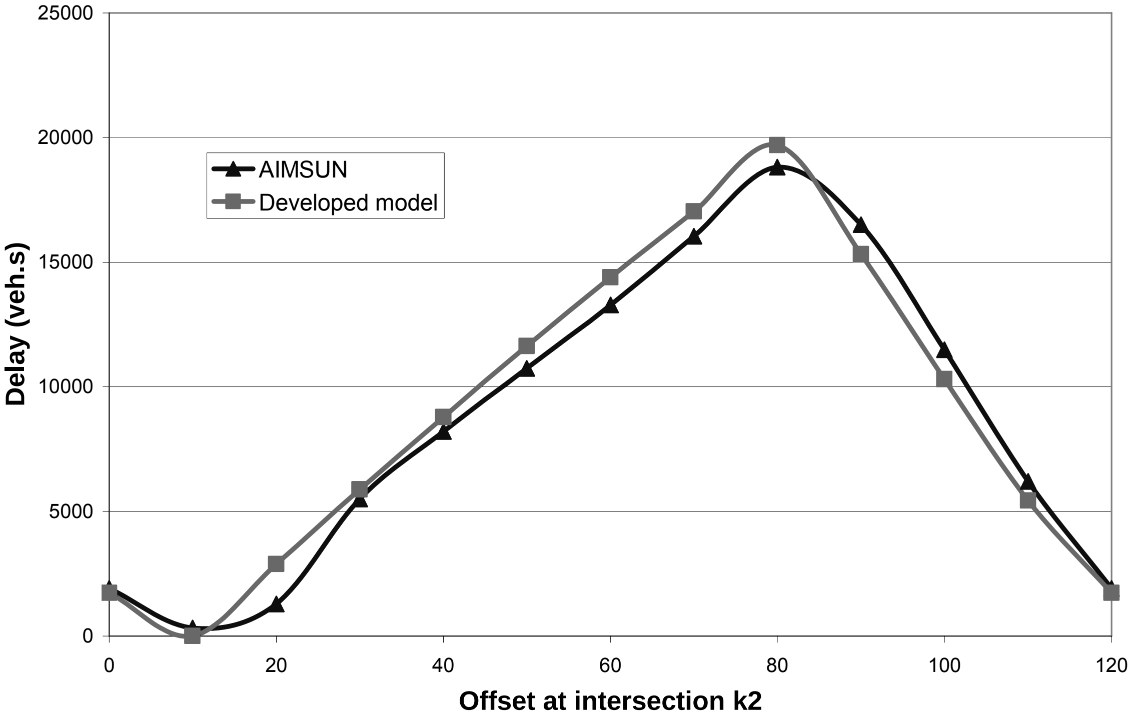


Figure 6-4: Delay comparison between the developed model and AIMSUN at one way street with 2 intersections

6.3 Delay Validation at a Small Network

Unlike the previous sections that tested only one link with no converge or diverge cells, in this section we test a realistic network which has turning movements and therefore does have diverge and converge cells. The network is a part of List District in Hanover, Germany.

This network consists of 6 intersections as shown in Figure 6-5 and the detailed layouts of these intersections are presented in Annex B. The existing timing plans for all intersections are also presented in Annex B.

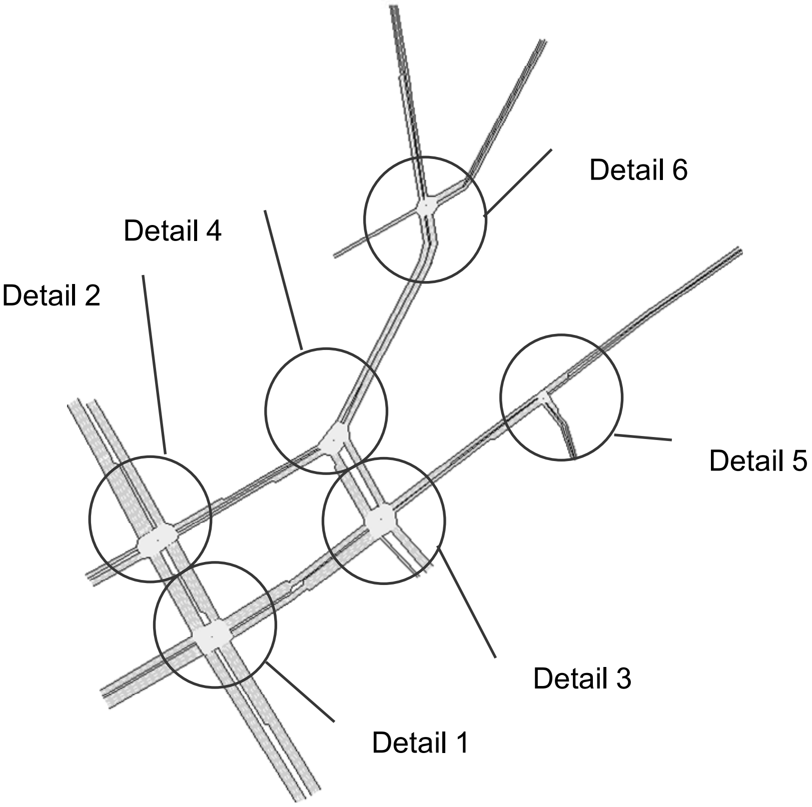


Figure 6-5: Site Location of a part of List District in Hanover city in Germany

In this network, we divide the links and turnings at intersection into cells of length equal to the product of the maximum speed (50 km/hr) by the time step of 1 s. In the cell representations, we find all types of cells (normal, merge, and diverge). Figure 6-6 shows the cell representation of the network.

For the purpose of validation, the estimated flows and delays at different locations of the network are compared against the ones obtained by AIMSUN. The following parameters were used:

- Horizon time: 900s;
- Free flow speed: 50 km/h (it differs at some turnings);
- Backward wave speed: 50 km/h;
- Jam density: 120 vehicles/km;
- Saturation flow: 1800;

- Time step: 1 second;
- Cycle time: 90 seconds; and
- Modeling horizon: 10 cycles (equivelent to 900 time intervals).

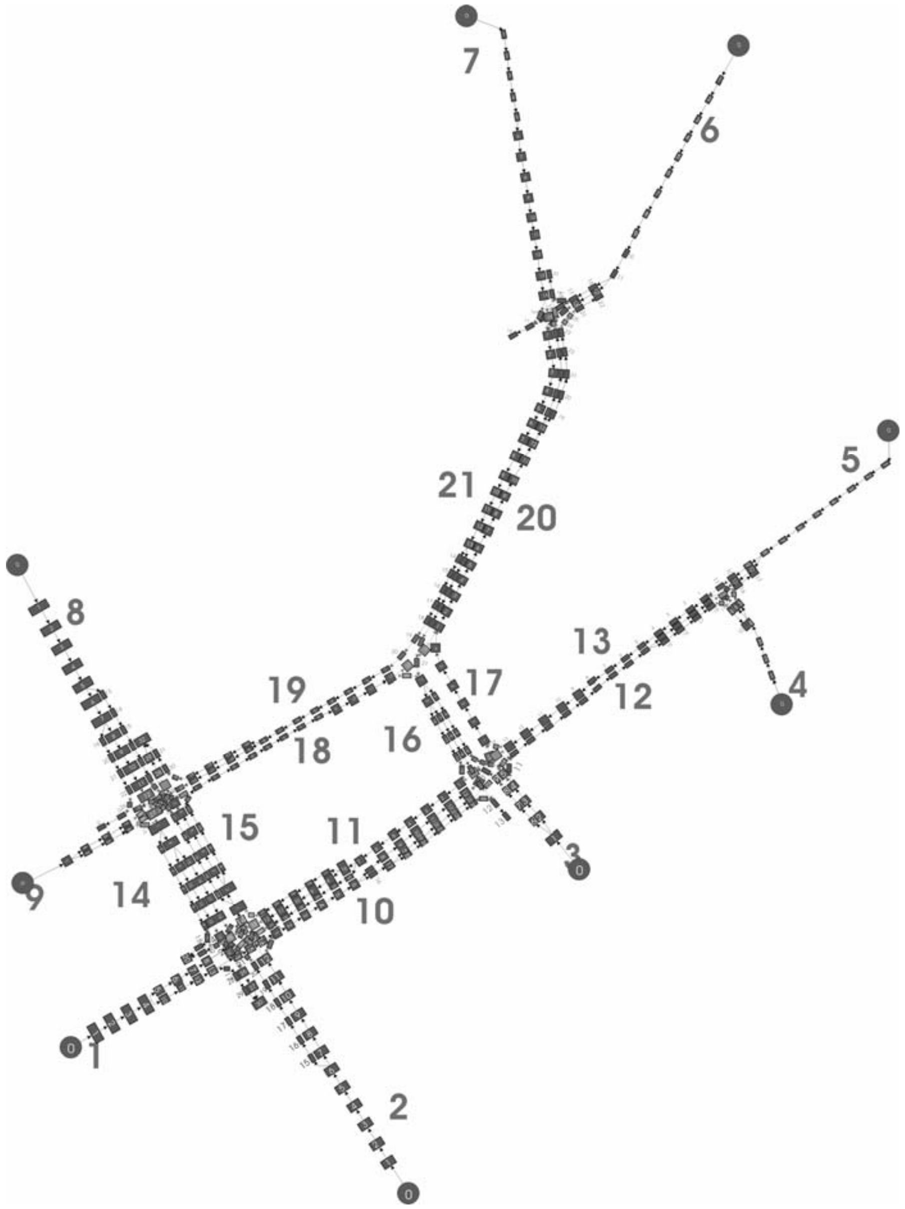


Figure 6-6: Cell representation of the 6-intersection network

Table 6-3 presents the entry flow data at the entry links (1-9) for the studied network in the studied period, and the turning proportions at all links. Note that the network was loaded before simulation in both the AIMSUN simulation and the developed analysis model; this means that the network was not empty.

Table 6-3: Link inflows and traffic splits

Link number	Inflow (veh/h)	Traffic Splits		
		Left	Through	Right
1	472	73%	27%	0%
2	656	0%	94%	6%
3	80	40%	60%	0%
4	4	100%	-	0%
5	256	0%	100%	-
6	248	42%	45%	13%
7	240	-	93%	7%
8	784	2%	98%	1%
9	52	31%	62%	8%
10	-	20%	65%	15%
11	-	23%	29%	47%
12	-	-	93%	7%
13	-	-	95%	5%
14	-	19%	34%	47%
15	-	1%	91%	8%
16	-	13%	31%	56%
17	-	-	-	100%
18	-	-	97%	3%
19	-	79%	21%	0%
20	-	5%	85%	10%
21	-	14%	86%	-

At each link-exit of the network, a detector was positioned in the simulated environment and the traffic flows within the modeling horizon were counted. At the same locations the flows were also counted by the developed model. The comparison of the flows presented in Figure 6-7 shows a good correspondence between the flows of the developed model and the ones of the simulated environment by AIMSUN.

Furthermore, the delay (s.veh) at each link was measured by AIMSUN and compared with the calculated delay by the model. A good R^2 of 0.97 could be obtained which proves the performance of the developed model. Figure 6-8 shows the comparison of delay between the developed model and AIMSUN and documents correspondence of both.

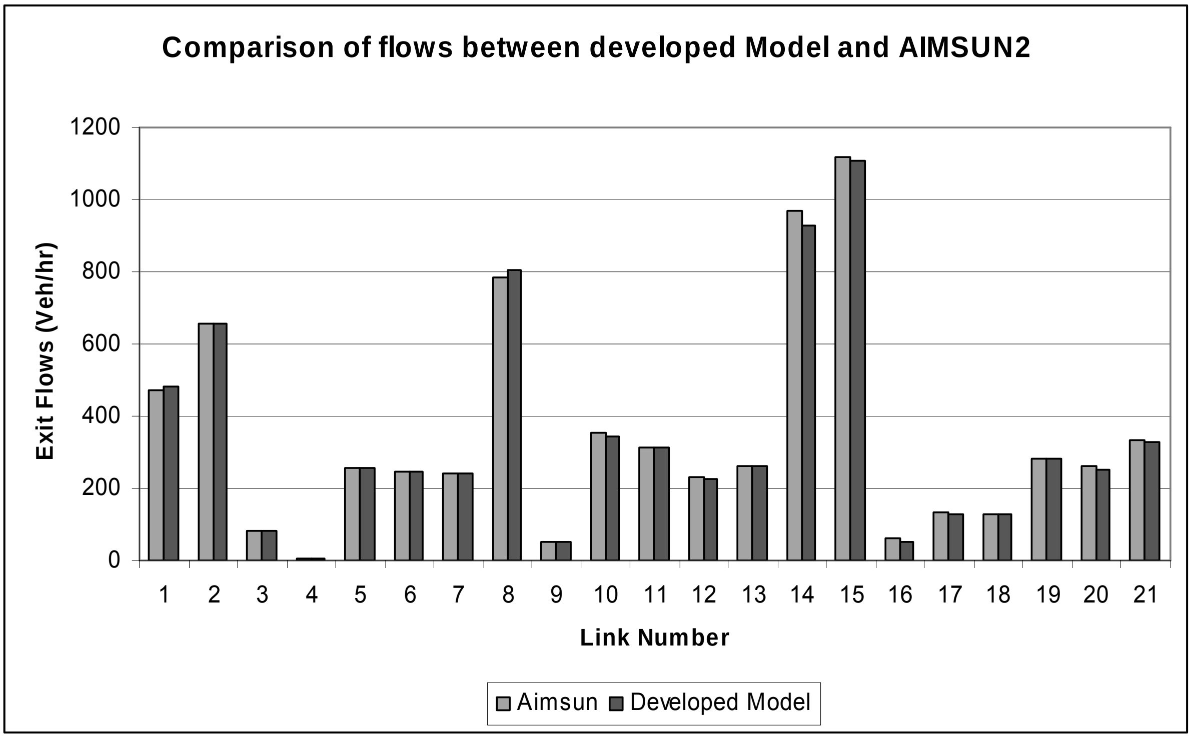


Figure 6-7: Comparison of flow rate of all links of the realistic network between the developed model and AIMSUN

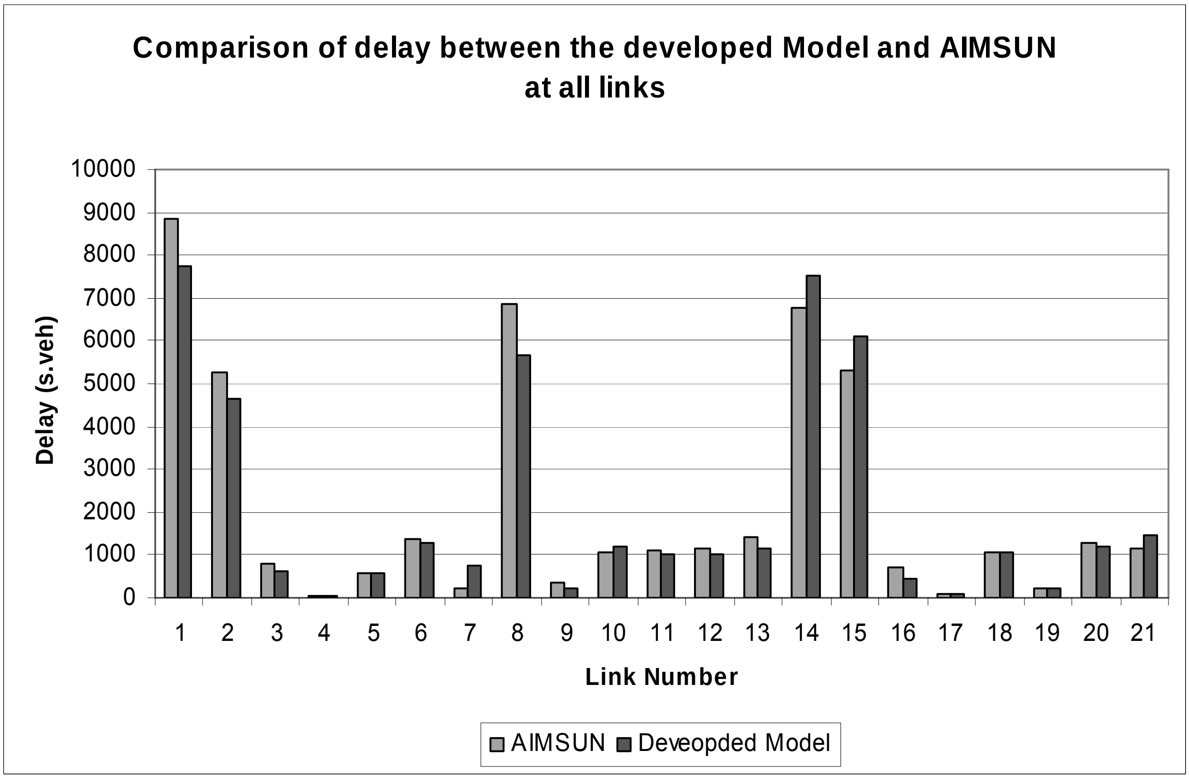


Figure 6-8: Comparison of delay all links of the network between the developed model and AIMSUN

To study the effect of varying offset values on the delay estimation, 5 trials of offset variations were studied as presented in Table 6-4. At each trial of offset variations, the network was simulated in AIMSUN with 30 replications and the average of the delays of these replications was used for the comparison.

Figure 6-9 shows the relationship between offset and delay for the tested network as estimated by AIMSUN and the developed model. The Figure shows that the delay values of the analysis model are similar to AIMSUN.

Table 6-4: Comparison of delay estimation at a small realistic network between the developed model and AIMSUN

Trial	Offset at intersection						Delay (veh.s)	
	K1	K2	K3	K4	K5	K6	Developed Model	AIMSUN
1	87	82	14	0	48	88	38476	38531
2	0	3	35	0	64	73	44849	45569
3	0	81	18	0	33	21	40248	39924
4	0	7	0	3	83	86	46452	44953
5	87	84	9	0	40	0	38976	38748

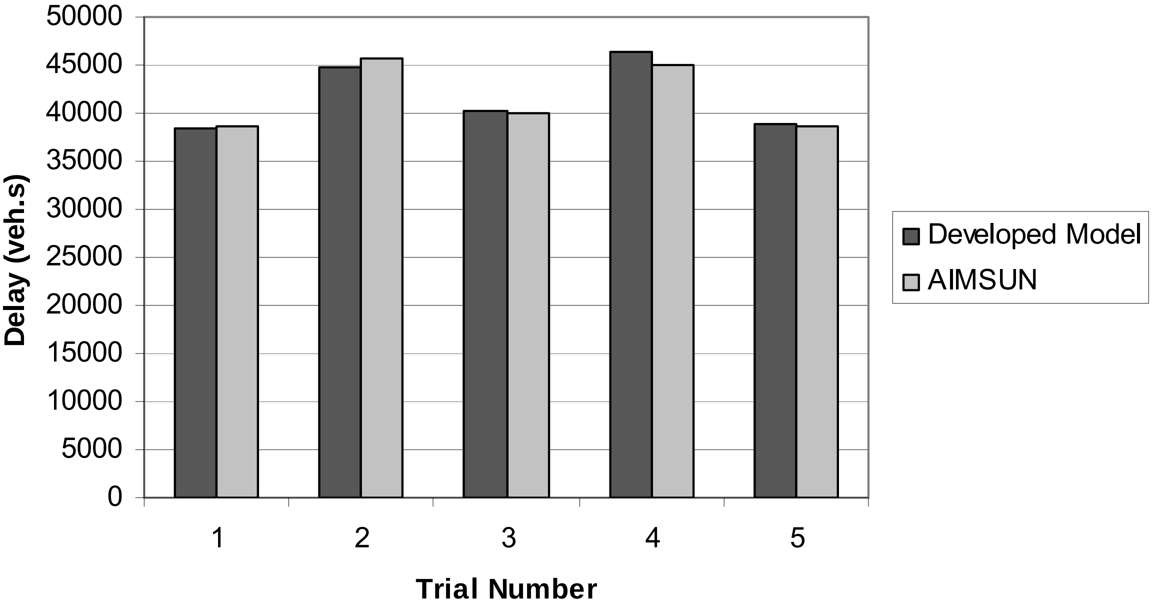


Figure 6-9: Comparison of delay estimation at a small realistic network between the developed model and AIMSUN for 5 trials of offset variations

7 GA-based Optimization Module

This chapter describes the optimization algorithm module. It includes two heuristic optimization approaches, which are based on Genetic Algorithm (GA) whose parameters and terminology were described in chapter 3. GA provides the capability of optimizing all the parameter in parallel, unlike the hill climbing method where each timing parameter has to be optimized one-after-another. All the studies conducted to date show that GA based optimization performs better than the hill climbing method. [36]

The two methods are different in the choice of the search direction. In the first approach, which is called parallel genetic algorithm (PGA), a simultaneous search over all offsets by the process of variation of reproduction – crossover – mutation of the entire chromosome is performed. A convergence criterion or the computing time available terminates the search.

In the second approach, which is called serial genetic algorithm (SGA), a group of offsets and therefore only a part of the chromosome is varied until the best solution is found. In the next step, the offsets of the next group of intersections are optimized. In a serial search such as this, the order in which intersections and offsets are respectively treated and searched greatly influences the optimization results. A method has been developed for the determination of the search order.

7.1 Parallel Genetic Algorithm

7.1.1 The Procedure

In the parallel genetic algorithm (PGA), a simultaneous search over offsets of all intersections in the network and therefore the entire chromosome is performed by the process of variation of reproduction – crossover – mutation. PGA procedure is shown in Figure 7-1.

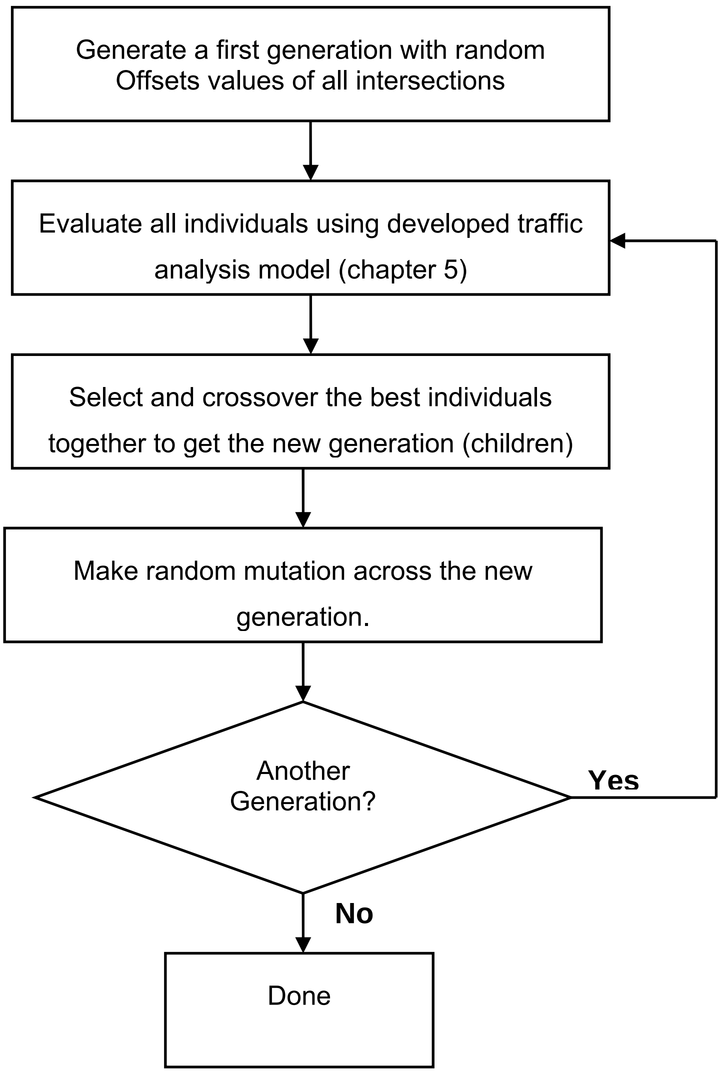

Generate a first generation with random Offsets values of all intersections
Evaluate all individuals using developed traffic analysis model (chapter 5)
Select and crossover the best individuals together to get the new generation (children)
Make random mutation across the new generation.
Another Generation?
Yes
No
Done

Figure 7-1: **A flowchart of the PGA**

As stated in the Background on Optimization Techniques (chapter 3), PGA consists of applying the following steps:

- Randomly generate a population of chromosomes encoding each solution (each solution represents a set of offsets for all intersections in the tested network); and
- While the termination condition is not met (in this research a maximum number of generations):
  - decode chromosomes into sets of offsets and then evaluate the fitness of all the individuals using the traffic analysis model (chapter 5);
  - put fittest chromosome into a mating pool;
  - produce a number of offspring by crossover and mutation; and then

- replace the weakest chromosomes with superior offspring.

7.1.2 Encoding and Decoding of the Offset Values

The PGA structure has to optimize offsets that minimize the objective function with consideration to maximum value of offset (cycle time). For the optimization, the offset values have to be encoded into a PGA structure and later decoded for conducting the analysis. Binary encoding, the simplest and most common of encoding, is used in this research. In binary encoding, every chromosome is represented by a binary string of bits, 0 or 1. Each bit in the string can represent some characteristics of the solution. Depending on the number of possible values for the offset, each decision variable (offset) is coded as an n-bit binary string. For example, when the cycle time is 90 s and the time step in the simulation model is 1s, then the offset has 90 possible values, therefore a 7-bit binary string is sufficient to represent one decision variable. In other words, when the number of decision variables are 5, then the length of the chromosome is 35 bits (7×5) and the chromosome structure is like [(1001010) (0101110) (1000010) (1011010) (1101101)].

7.1.3 Other GA Characteristics

The following GA parameters are used in this algorithm:

- Elitism in order to prevent a loss of the best found solution.
- Tournament selection, which is proved to be the best.
- Truncation scaling method is used as the purpose is to maximize the objective function that has a minus value (minimize the delay).
- The population size ranges between 50-200 depending on the number of variables.
- The maximum number of generations is determined by a combination of the simulation time horizon, the calculation time needed for one simulation, and the population size. For example, if the time horizon is 900 s, the calculation time needed for one simulation is 1 s and the population size is 100, then the maximum number of evaluations is equal to 900 (900s / 1s) leading to a maximum number of generations of 9 (900 / 100).
- A single-point crossover is used.
- The mutation rate ranges from 0.05 to 0.2 depending on the length of the chromosome.

7.2 Serial Genetic Algorithm

7.2.1 The Procedure

Rather than considering all the intersections of the network simultaneously, a serial genetic algorithm (SGA) is proposed, which considers the relative offsets of a group of intersections one at a time. These intersections are located on traffic direction r of vehicles moving from origin i to destination j. Instead of performing GA operations on the entire chromosome, only the part related to this group of intersections is considered. The remaining intersections are fixed by the optimal values found in the last iteration. However, the fitness of new chromosomes is determined by *evaluating all of the network intersections*, and not only the group of intersections under analysis. By disconnecting the simultaneous search into a sequence of searches, the length of the chromosome in the SGA is considerably shorter than that in the PGA. By shortening the chromosome, and thus the solution space, one can use a small population of chromosomes and find quasi-optimal offsets quicker than with the more conventional PGA. In the next step, the offsets of the next group of intersections are optimized. In a serial search such as this, the order in which the intersection and offsets are respectively treated and searched greatly influences the optimization results. A method has been developed for the determination of the groups of offsets and their search order, which is described in the next section.

The SGA procedure, shown in Figure 7-2, consists of the following steps:

1. Determine groups of relative offsets and their order of search based on section 7.2.2;
2. Initialize each relative offset in each group with a value equal to the time needed to travel freely from the first intersection to the second one and binary encode this value (the recommended values are believed to have a good fitness, which can speed up the solution);
3. Start the GA process with the first group determined in step 1 and randomly generate a population of chromosomes encoding each solution, which represents all relative offsets in this group of intersections in the tested network;
4. While termination condition is not met (number of generations):
    - decode chromosomes into sets of relative offsets and then evaluate the fitness of all the individuals using the developed traffic model;
    - put fittest chromosome into a mating pool;
    - produce a number of offspring by crossover and mutation; and then
    - replace weakest chromosomes with superior offspring.

5. Fix the best relative offsets of this group and repeat steps 3 and 4 with the order determined in step1 until the groups are finished.

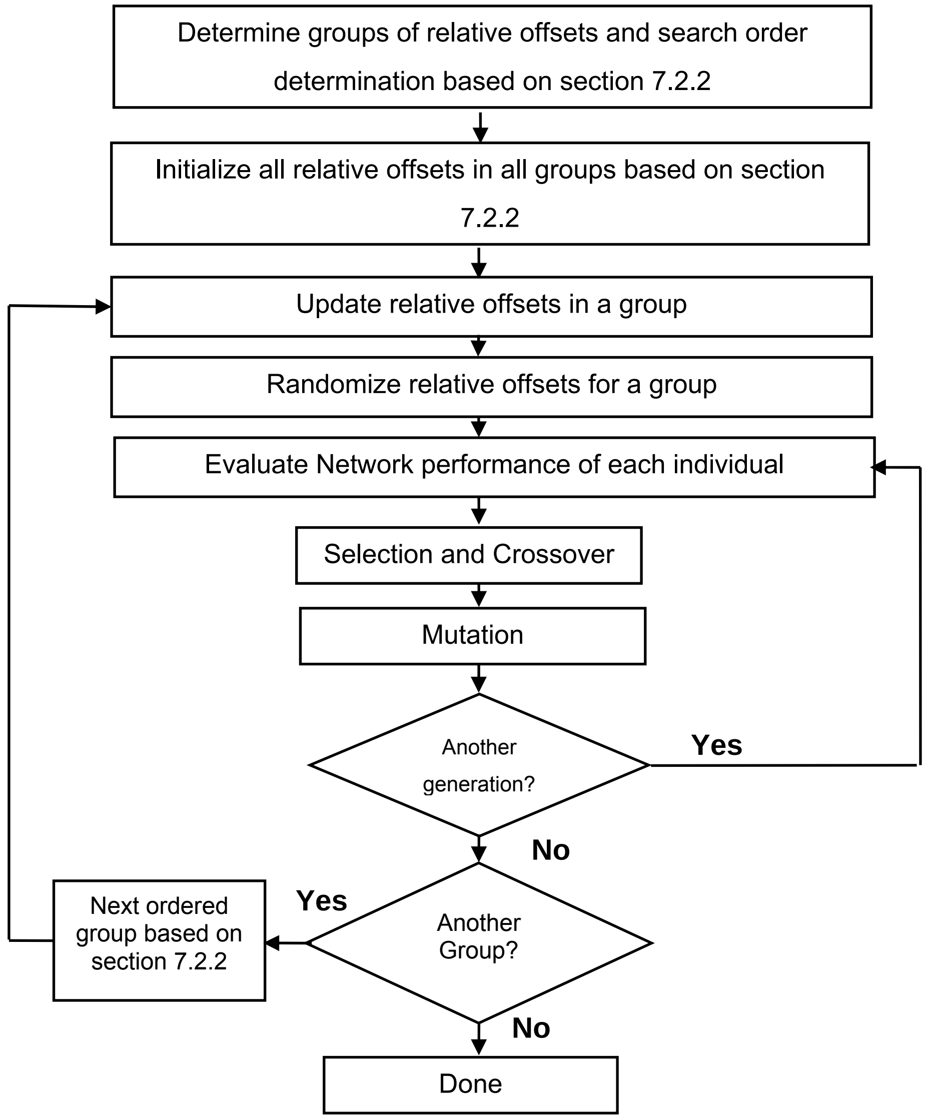

Determine groups of relative offsets and search order determination based on section 7.2.2
Initialize all relative offsets in all groups based on section 7.2.2
Update relative offsets in a group
Randomize relative offsets for a group
Evaluate Network performance of each individual
Selection and Crossover
Mutation
Another generation?
Yes
No
Another Group?
Yes
Next ordered group based on section 7.2.2
No
Done

Figure 7-2: A flowchart of the SGA

7.2.2 A Method for Search Order for SGA

The determination of the search order in a grid network is complex because of the high interdependencies in the network. A proposed solution for this problem is based on the entry flows plus the turning proportions at intersections. The following assumptions are the basis of the proposed method:

- Throughout the optimization period, the entry flows at entry links as well as the turning proportions at each intersection remain constant.

- All intersections are controlled by signals.
- The optimization is carried out on relative offsets. Each one is between two movements at two successive intersections, e.g. between movements A and B (or A and C) at intersections 1 and 2 (see Figure 7-3).
- For a network that has a number of n signalized intersections, only n-1 relative offsets can be optimized. [43]

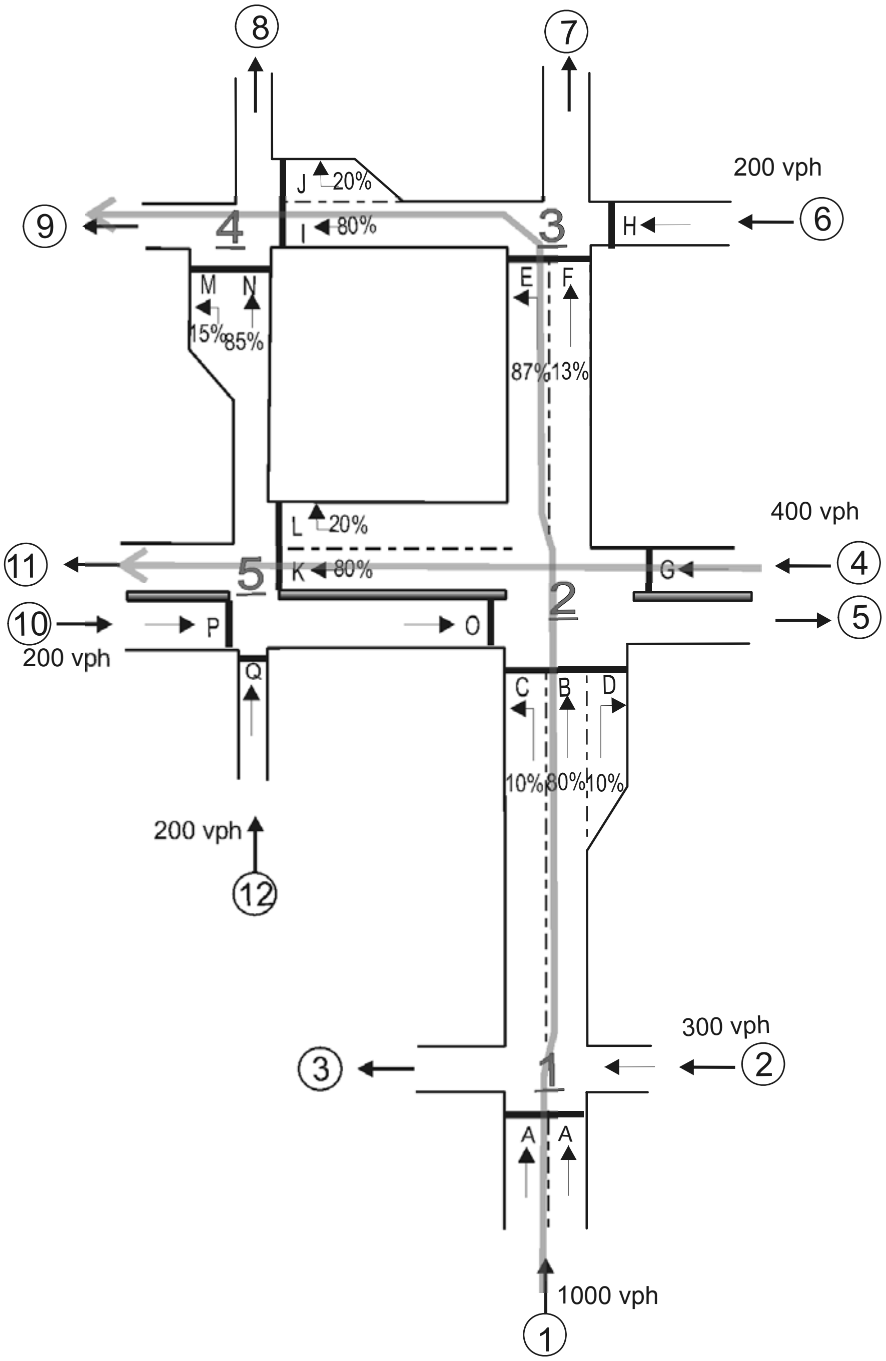


Figure 7-3: A network with 5 intersections for illustrating the search order method

The search order method consists of the following 6 steps:

Step1:

For each origin i to each destination j, calculate the traffic flow T_{ijr} moving via route r as follows:

$$T_{ijr} = F_i \prod_{k=1}^{n_{ijr}} \beta_{ijr}^{k} \tag{7-1}$$

Where:

F_i = The traffic flow generated from origin i.

n_{ijr} = The number of intersections located on route r when traveling from origin i to destination j.

β^{k}_{ijr} = The turning proportion at intersection k for a movement located on route r when traveling from origin i to destination j.

To clarify equation (7-1), consider a route marked in Figure 7-3 with a bold line when traveling from origin 1 to destination 9 via intersections 1,2,3 and 4. The number of intersections n_{ijr} located on this route is 4. The traffic flow F_i generated from origin 1 is 1000 veh/h. Four turning proportions are considered for this route:

- The turning proportion β^{1}_{ijr} at intersection 1 for movement A is 1;
- The turning proportion β^{2}_{ijr} at intersection 2 for movement B is 0.8;
- The turning proportion β^{3}_{ijr} at intersection 3 for movement E is 0.87; and
- The turning proportion β^{4}_{ijr} at intersection 4 for movement I is 0.8.

Based on these values, the traffic flow T_{ijr} moving for origin 1 to destination 9 via the route (r= [1, 2, 3, 4]) is:

$$\begin{aligned} T_{ijr} &= 1000(1 \times 0.8 \times 0.87 \times 0.8) \\ &= 557 \text{ vph} \end{aligned}$$

Step2:

Calculate the *total* traffic flow TF_{ijr} for the traffic moving for origin i to destination j via route r as follows:

$$TF_{ijr} = \sum_{k=1}^{n_{ijr}} q_{ijr}^{k} \tag{7-2}$$

$$q^{k}_{ijr} = \begin{cases} T_{ijr} & \forall k \in r \\ 0 & \forall k \notin r \end{cases} \quad (7\text{-}3)$$

Where:

q^{k}_{ijr} = The traffic flow at intersection k for a movement located on route r when traveling from origin i to destination j.

For applying equations (7-2) and (7-3), consider the same route in Figure 7-3. The traffic flow q^{k}_{ijr} at each intersections 1, 2, 3 and 4 is equal to T_{ijr} (=557 veh/h), therefore, the total traffic flow TF_{ijr} is equal to 2228 (=557+557+557+557). Table 7-1 presents the calculation of TF_{ijr}, when traveling from each origin i to each destination j via all possible routes.

Step3:

Rank all the values of TF_{ijr} from the highest to the lowest as presented in Table 7-1.

Table 7-1: Calculations of search order procedure for the network with 5 intersection

i	j	F_i	r	n_{ijr}	β^{k}_{ijr}				T_{ijr}	TF_{ijr}	**Rank**	N_{var}
1	5	1000	[1,2]	2	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.1$	_	_	100	200	9	
1	7	1000	[1,2,3]	3	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.8$	$\beta^{3}_{ijr} = 0.13$	_	104	312	6	
1	8	1000	[1,2,3,4]	4	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.8$	$\beta^{3}_{ijr} = 0.87$	$\beta^{4}_{ijr} = 0.2$	139	557	3	
1	8	1000	[1,2,5,4]	4	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.1$	$\beta^{5}_{ijr} = 0.2$	$\beta^{4}_{ijr} = 0.85$	17	68	10	
1	9	1000	[1,2,3,4]	4	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.8$	$\beta^{3}_{ijr} = 0.87$	$\beta^{4}_{ijr} = 0.8$	557	2227	1	3
1	9	1000	[1,2,5,4]	4	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.1$	$\beta^{5}_{ijr} = 0.2$	$\beta^{4}_{ijr} = 0.15$	3	12	13	
1	11	1000	[1,2,5]	3	$\beta^{1}_{ijr} = 1$	$\beta^{2}_{ijr} = 0.1$	$\beta^{5}_{ijr} = 0.8$	_	80	240	7	
4	8	400	[2,5,4]	3	$\beta^{2}_{ijr} = 1$	$\beta^{5}_{ijr} = 0.2$	$\beta^{4}_{ijr} = 0.85$	_	68	204	8	
4	9	400	[2,5,4]	3	$\beta^{2}_{ijr} = 1$	$\beta^{5}_{ijr} = 0.2$	$\beta^{4}_{ijr} = 0.15$	_	12	36	12	
4	11	400	[2,5]	2	$\beta^{2}_{ijr} = 1$	$\beta^{5}_{ijr} = 0.8$	_	_	320	640	2	1
12	8	200	[5,4]	2	$\beta^{5}_{ijr} = 1$	$\beta^{4}_{ijr} = 0.85$	_	_	170	340	5	
12	9	200	[5,4]	2	$\beta^{5}_{ijr} = 1$	$\beta^{4}_{ijr} = 0.15$	_	_	30	60	11	
10	5	200	[5,4]	2	$\beta^{5}_{ijr} = 1$	$\beta^{4}_{ijr} = 1$	_	_	200	400	4	

Step4:

Start from the highest TF_{ijr} to the lowest , calculate the number of variables of relative offsets as:

$$n_{\text{var}} = n_{ijr} - 1 \tag{7-4}$$

Step5:

Repeat step 4 until the total number of variables reaches the maximum allowed number of variables which is the total number of intersections in the network minus 1. When calculating the number of variables, don't allow any repetition of the variables. For example if the first and the second groups have the same relative offset between intersections 1 and 2, this must not be counted and optimized twice.

Since the maximum number of relative variables to be optimized in this network is 4, only 2 groups of decision variables could be found (see Table 7-1):

- The first group has 3 variables of relative offsets along the route r = [1,2,3,4] marked in Figure 7-3: (1) is between movements A at intersections 1 and B at intersections 2; (2) is between movements B at intersections 2 and movement E at intersection 3; and (3) is between movement E at intersection 3 and movement I at intersection 4.
- The second group has 1 variable along the route r = [2,5], which is between movement G at intersection 2 and movement K at intersection 5.

Step6:

Since the developed traffic analysis model considers only offsets in the evaluation, a relationship between the offsets and relative offsets must be derived. To help to derive this relationship, consider Figure 7-4, which shows part of the network already presented in Figure 7-3. In Figure 7-4, we coordinate between movement A at intersection 1 and movement B at intersection 2. Based on the time-space diagram drawn for the coordinated movements, the following equation is valid:

$$BGT_A^1 + Offset_1 = BGT_B^2 + Offset_2 - Rel.Offset_{AB} \tag{7-5}$$

Where:

$Offset_1$ = Offset of intersection 1.

$Offset_2$ = Offset of intersection 2.

BGT_A^1 = Difference between the beginnings of the green time of the phase assigned to movement A and of the green time of the first phase at intersection 1.

BGT_B^2 = Difference between the beginnings of the green time of the phase assigned to movement B and of the green time of the first phase at intersection 2.

$Rel.Offset_{AB}$ = Relative offset between movements A and B

When the offset at intersection 2 is sought and all other variables are known, equation (7-5) can be simply restated as follows:

$$Offset_2 = Offset_1 + BGT_A^1 + Rel.Offset_{AB} - BGT_B^2 \tag{7-6}$$

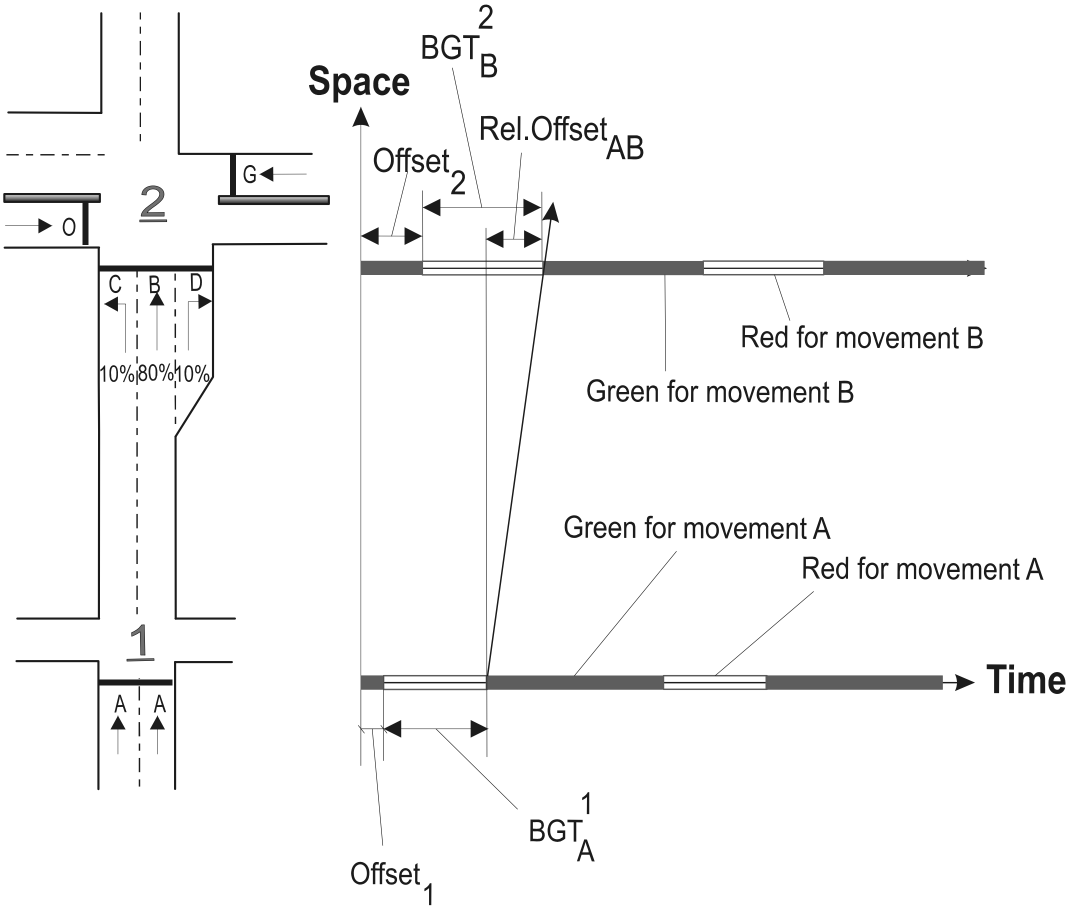


Figure 7-4: A time-space diagram for illustrating the relationship between the offsets and the relative offsets

To see the effect of the values of turning proportions on the search order method, the same network with different turning proportions shown in Figure 7-5 is studied again. The changes of the turning proportions are for movements E and F at intersection 2 and for the movements K and L at intersection 5.

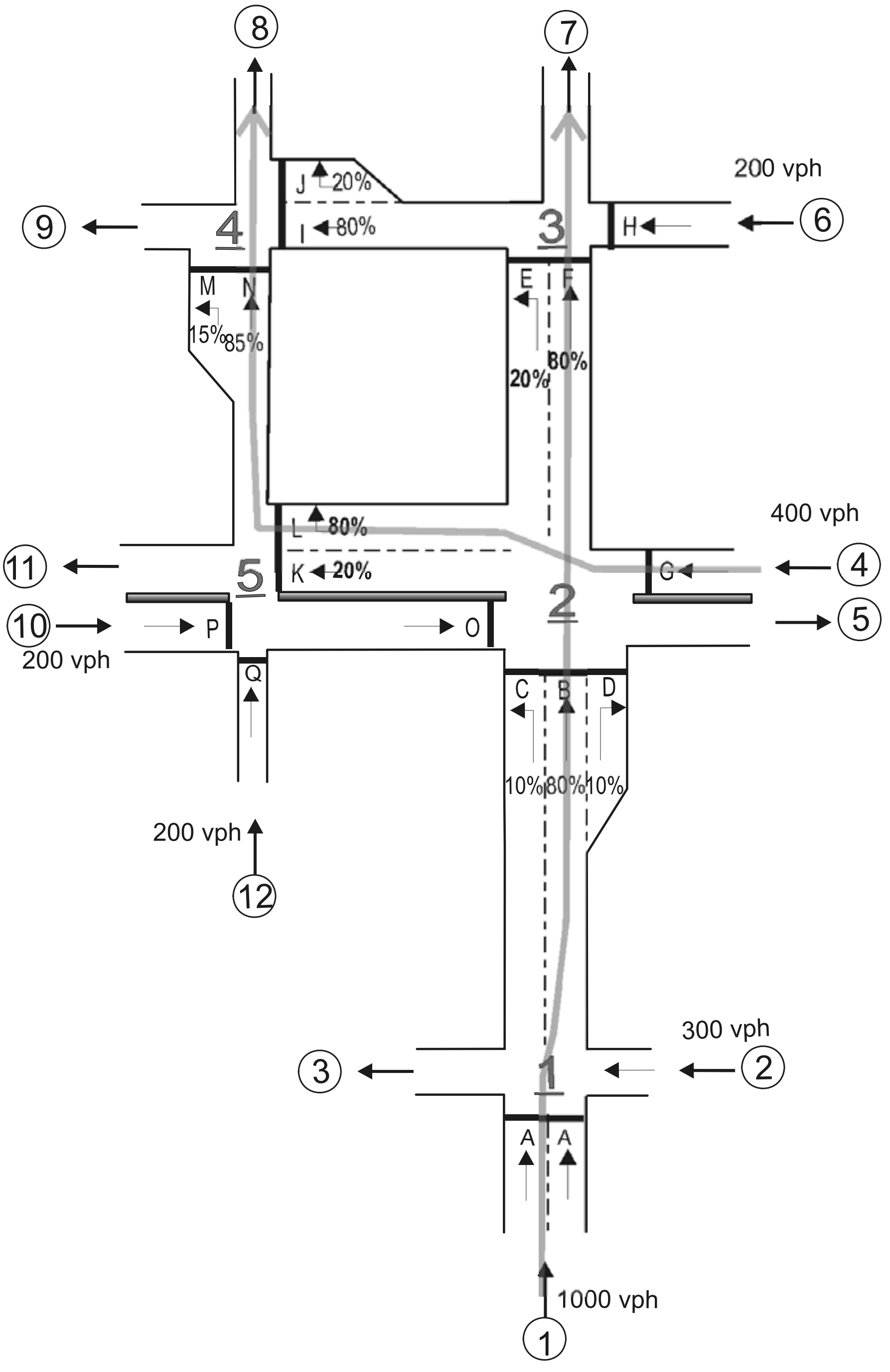


Figure 7-5: **A network with 5 intersections with different turning proportions**

Table 7-2 presents the calculations of search order method for the new turning proportions. The results show that there are two groups of relative offsets. The first group has a total traffic TF_{ijr} of

1920 veh/h along a route r (= [1, 2, 3]) as shown in Figure 7-5. This group has two variables of relative offsets:

- The first one is between movements A and B at intersections 1 and 2 respectively.
- The second one is between movements B and F at intersections 2 and 3 respectively.

For the second group, TF_{ijr} was 816 veh/h via route r (= [2, 5, 4]) as described in Figure 7-5. This group has also two variables of relative offsets:

- The first one is between movements G and L at intersections 2 and 5 respectively.
- The second one is between movements L and N at intersections 5 and 4 respectively.

Table 7-2: Calculations of search order procedure for the network that has 5 intersections with different turning proportions

i	j	F_i	r	n_{ijr}	β^k_{ijr}				T_{ijr}	TF_{ijr}	Rank	N_{var}
1	5	1000	[1,2]	2	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.1$	_	_	100	200	6	
1	7	1000	[1,2,3]	3	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.8$	$\beta^3_{ijr} = 0.8$	_	640	1920	1	2
1	8	1000	[1,2,3,4]	4	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.8$	$\beta^3_{ijr} = 0.2$	$\beta^4_{ijr} = 0.2$	32	128	9	
1	8	1000	[1,2,5,4]	4	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.1$	$\beta^5_{ijr} = 0.8$	$\beta^4_{ijr} = 0.85$	68	272	5	
1	9	1000	[1,2,3,4]	4	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.2$	$\beta^3_{ijr} = 0.2$	$\beta^4_{ijr} = 0.8$	32	128	10	
1	9	1000	[1,2,5,4]	4	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.1$	$\beta^5_{ijr} = 0.8$	$\beta^4_{ijr} = 0.15$	12	48	13	
1	11	1000	[1,2,5]	3	$\beta^1_{ijr} = 1$	$\beta^2_{ijr} = 0.1$	$\beta^5_{ijr} = 0.2$	_	20	60	11	
4	8	400	[2,5,4]	3	$\beta^2_{ijr} = 1$	$\beta^5_{ijr} = 0.8$	$\beta^4_{ijr} = 0.85$	_	272	816	2	2
4	9	400	[2,5,4]	3	$\beta^2_{ijr} = 1$	$\beta^5_{ijr} = 0.8$	$\beta^4_{ijr} = 0.15$	_	48	144	8	
4	11	400	[2,5]	2	$\beta^2_{ijr} = 1$	$\beta^5_{ijr} = 0.2$	_	_	80	160	7	
12	8	200	[5,4]	2	$\beta^5_{ijr} = 1$	$\beta^4_{ijr} = 0.85$	_	_	170	340	4	
12	9	200	[5,4]	2	$\beta^5_{ijr} = 1$	$\beta^4_{ijr} = 0.15$	_	_	30	60	12	
10	5	200	[5,4]	2	$\beta^5_{ijr} = 1$	$\beta^4_{ijr} = 1$	_	_	200	400	3	

Another network is also studied in this section, since it is relevant to chapter 8 of method comparison. This network shown in Figure 7-6 consists of 12 intersections. The entry traffic flow and the ID's for all origins are shown in the Figure. It is assumed in this network that turning proportions at intersections are fixed during the simulation period. These proportions are 0.3 for left/right turns; and 0.7 for through movement.

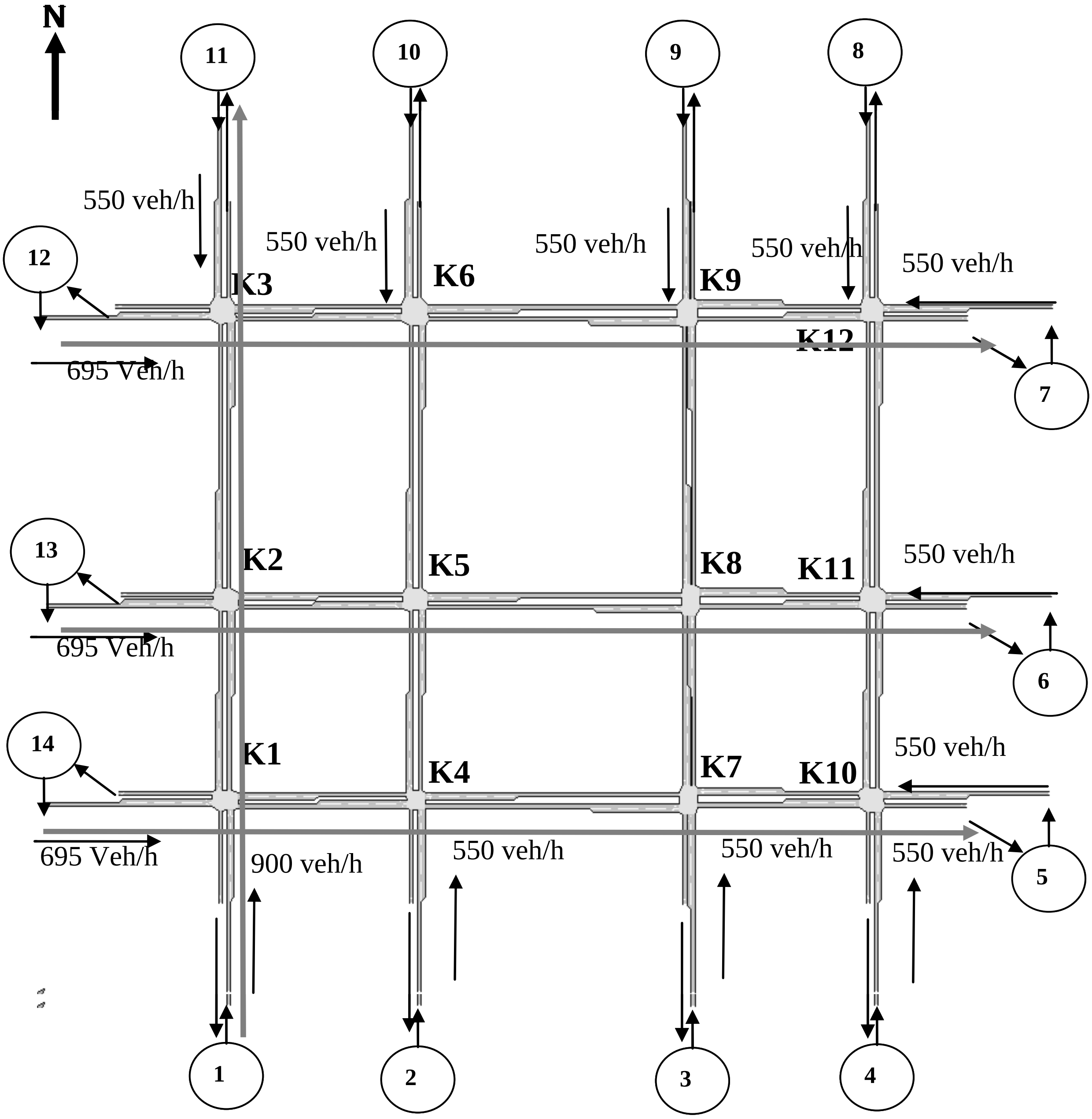


Figure 7-6: **A grid network for the illustration of search order**

Table 7-3 summarizes the calculations for this network. Since 11 variables of relative offsets are allowed for optimization, only four groups of relative offsets could be found. The routes for these groups were already drawn in Figure 7-6. These groups were chosen, because their total traffic flows TF_{ijr} were the highest as shown in Table 7-3. Table 7-4 presents them with their variables.

Table 7-3: A summary of calculations of search order procedure for the grid network

i	j	F_i	r	n_{ijr}	β^k_{ijr}						T_{ijr}	TF_{ijr}	Rank	N_{var}
1	3	900	[1,4,7]	3	0.30	0.70	0.30				57	170	16	
1	3	900	[1,4,5,8,7]	5	0.30	0.30	0.30	0.30	0.70		5	26	24	
1	5	900	[1,4,7,10]	4	0.30	0.70	0.70	0.70			93	370	11	
1	5	900	[1,2,5,8,7,10]	6	0.70	0.30	0.70	0.30	0.30	0.70	8	50	23	
1	6	900	[1,2,5,8,11]	5	0.70	0.30	0.70	0.70	0.70		65	324	12	
1	6	900	[1,4,5,8,11]	5	0.30	0.30	0.30	0.70	0.70		12	60	21	
1	7	900	[1,2,3,6,9,12]	6	0.70	0.70	0.30	0.70	0.70	0.70	45	272	14	
1	7	900	[1,4,7,10,11,12]	6	0.30	0.70	0.70	0.30	0.70	0.30	8	50	22	
1	8	900	[1,2,3,6,9,12]	6	0.70	0.70	0.30	0.70	0.70	0.30	19	117	18	
1	8	900	[1,4,7,10,11,12]	6	0.30	0.70	0.70	0.30	0.70	0.70	19	117	19	
1	10	900	[1,2,3,6]	4	0.70	0.70	0.30	0.30			40	159	20	
1	10	900	[1,2,5,6]	4	0.70	0.30	0.30	0.70			40	159	20	
1	11	900	[1,2,3]	3	0.70	0.70	0.70				309	926	1	2
2	10	550	[4,5,6]	3	0.70	0.70	0.70				189	566	8	
2	5	550	[4,7,10]	3	0.30	0.70	0.70				81	243	15	
3	9	550	[7,8,9]	3	0.70	0.70	0.70				189	566	6	
4	8	550	[10,11,12]	3	0.70	0.70	0.70				189	566	7	
5	14	550	[10,7,4,1]	4	0.70	0.70	0.70	0.70			132	528	9	
8	4	550	[12,11,10]	3	0.70	0.70	0.70				189	566	5	
14	5	695	[1,4,7,10]	4	0.70	0.70	0.70	0.70			167	667	2	3
14	3	695	[1,4,7]	3	0.70	0.70	0.30				102	306	10	
13	6	695	[2,5,8,11]	4	0.70	0.70	0.70	0.70			167	667	3	3
13	10	695	[1,4,7]	3	0.30	0.70	0.70				102	306	13	
12	7	695	[3,6,9,12]	4	0.70	0.70	0.70	0.70			167	667	4	3

Table 7-4: The 11 variables of relative offsets for the grid network

		From			To		
Variable	Group	Movement	Direction	intersection	Movement	Direction	intersection
1	1	TH	NB	K1	TH	NB	K2
2	1	TH	NB	K2	TH	NB	K3
3	2	TH	EB	K1	TH	EB	K4
4	2	TH	EB	K4	TH	EB	K7
5	2	TH	EB	K7	TH	EB	K10
6	3	TH	EB	K2	TH	EB	K5
7	3	TH	EB	K5	TH	EB	K8
8	3	TH	EB	K8	TH	EB	K11
9	4	TH	EB	K3	TH	EB	K6
10	4	TH	EB	K6	TH	EB	K9
11	4	TH	EB	K9	TH	EB	K12

8 Method Comparisons

The final step in the development of a method for the offset optimization is to compare the optimization output with existing methods. Comparisons of both PGA and SGA were conducted against TRANSYT-7F, dominance method and Full Enumeration Method using AIMSUN as the evaluator.

This chapter begins with a small introduction of these three comparison methods, followed by a description of the method of statistical testing. The results of the comparisons are presented for three case studies: a two-way street of 3 intersections; a small network with 6 intersections; and finally with a relatively large grid network with 12 intersections.

8.1 Comparison Methods

8.1.1 Full Enumeration

Full enumeration of offset variables using the developed traffic model is used as a reference for the optimal offsets in a network. It applies a simple nested loop algorithm, which is shown in Figure 8-1.

In any network that has **n** connected number of intersections, only **n - 1** number of offset variables can be assigned arbitrary values [43]. If a network consists of n intersections and the common cycle time is CT, then the solution space is CT^{n-1}. Given that one run of the developed simulator in this research requires t_{run} seconds; the calculation time needed for the full enumeration is $CT^{n-1} \times t_{run}$.

```
For variable 1=0 to CT (cycle time) do
    For variable 2=0 to CT (cycle time) do
        For variable 3=0 to CT (cycle time) do
                    ⋮
            For variable n-1=0 to CT (cycle time) do
                Calculate delay (CTM simulation)
                ……………..
            End loop n-1
                    ⋮
        End loop 3
    End loop 2
End loop 1
```

Figure 8-1: A Flowchart of full enumeration (nested-loop algorithm)

In the case of a network is small, the CPU-time in a normal PC needed for the full enumeration of offset variables is feasible in proper time. For example, as in the first case study in this chapter, in a network of 3 intersections, the common cycle time is 70 s and the time needed to calculate the delay based on the developed analysis module in this research is 400 ms (0.4 s); this results in the calculation time needed for the full enumeration being equal to 1960 s ($70^{3-1} \times 0.4$).

However, when a larger network is considered, this becomes unrealistic. For example, as in the second case study in this chapter, in a network of 6 intersections, the common cycle time is 90 s and the time needed to calculate the delay based on the developed analysis module in this dissertation is 700 ms (0.7 s); this results in a required calculation time of 4.133×10^9 s or 131 years ($90^{6-1} \times 0.7$) for full enumeration. For this reason, one can use a 10 s time step instead of 1 s for each offset variable, resulting in each offset being iterated 9 times. Then, the time needed for the full enumeration is equal to 11.48 hrs ($9^5 \times 0.7$s= 4.133×10^4s), which is possible. When more accurate results are required, another nested loop can be used with a smaller time step, say 5s, around the optimal values found in the previous nested loops. A third full enumeration can also be made later with a time step of 2 or 1 s around the new optimal values.

8.1.2 Dominance Method

The dominance method, which is called in German “Dominanzverfahren”, was developed by Schnabel [90] in 1981. This method is selected for comparison because it is a practical and well-known method in Germany. It is based on setting up a continuous green band for a traffic direction and coordinates the traffic direction with the heaviest traffic. The heaviest traffic direction is described as a main direction (HR) and determined by comparison of the traffic flows. The total traffic flow M_r of the continuous traffic of a direction r is:

$$M_r = \sum_{i=1}^{K} \sum_{J=1}^{Ni} M_{rij} \tag{8-1}$$

Where:

K = Number of intersections

N_i = Number of lanes available for the continuous traffic at intersection *i*.

M_{rij} = Traffic flow of lane *j* at intersection *i* in the traffic direction *r*.

When the traffic flows of both directions are equal, any direction can be considered as the main direction. The direction with less traffic flow is referred as the opposed direction (GR). It is called the dominance method because it is concerned with the dominant (or main) direction of the traffic. For detailed description of this method, see Schnabel [90].

8.1.3 TRANSYT-7F

TRANSYT (Traffic Network Study Tool) is an offline program for calculating optimum coordinated signal timings in a network of traffic signals. It has become the most widely used program of its type in the world. Since the first program was developed in 1967, a number of versions have been produced, all of which have two main elements: a traffic model and a signal optimizer.

In 1981, a version modified specifically for the United States was created and entitled as TRANSYT-7F (Release 1).The resale 10 of TRANSYT-7F which was created in 2004 is compared with the method developed in this dissertation. The reasons for selecting TRANSYT-7F for comparison are:

- Its traffic model has recently been extended to overcome oversaturated traffic conditions;
- GA has recently been added to hill climbing for optimization purposes; and
- The manual of the last version (TRANSYT-7F10) states that it can be used for online purposes.

The model predicts the value of a ‘performance index’ (PI) for the network for a given fixed-time plan and an average set of flows on each link. The performance index measures the overall cost of traffic ‘congestion’ and is usually a combination of the total delay and the number of stops made by vehicles. For detailed description of TRANSYT-7F, see Hale [51].

8.2 Statistical Testing

In order to compare the various coordination methods, the optimal offsets obtained from all the the strategies were fed into AIMSUN. Since AIMSUN is a stochastic microscopic simulation model and does not give a unique solution to a problem, different replications/runs of each optimal solution were conducted. The AIMSUN user manual [99] provides a method, based on Law et al. [66], to calculate the number of replications required in order to obtain a value within the limits of k% (e.g. 5%) of the mean of a α% (e.g. 95%) level of confidence. The number of recommended replication based on this method is between 20 and 30. In this research, 30 replications are used in all case studies.

After conducting the 30 replications for each case study, all results of system delays obtained from AIMSUN for all replications were gathered for appropriate statistical testing for the purpose of comparison. This statistical testing is ANOVA followed with Scheffè Post Hoc Comparisons. ANOVA (Analysis of variance) is used to test hypotheses about differences between two or more means. In ANOVA tests, when the null hypothesis is rejected, this rejection indicates that at least one pair of means is unequal. However, this test does not identify which pairs of means are significantly different. This problem is solved by a method called Scheffè Post Hoc Comparisons [95].

According to Gerstman [48], the following hypothesis testing was applied for ANOVA test:

$$H_0(\text{Null Hypothesis}):\ \mu_1=\mu_2=\ldots=\mu_\text{k} \tag{8-2}$$

$$H_a(\text{Alternative Hypothesis}):\ H_0 \text{ is false (at least one experiment mean differs)} \tag{8-3}$$

Where μ_i represents the experiment mean of group *i.*

However, the analysis of variance is used to compare group means in terms of two components: variance between groups (s_B^2) and variance within groups (s_W^2). Therefore, the above hypotheses are restated as:

$$H_0(\text{Null Hypothesis}):\ s_B^2 \le s_W^2 \cdot F_{table} \tag{8-4}$$

$$H_a(\text{Alternative Hypothesis}):\ S_B^2 > S_W^2 \cdot F_{table} \tag{8-5}$$

Where F_{table} is a factor obtained from a table, which is based on the group sizes and the level of confidence α.

The ratio of variance between groups (s_B^2) and variance within groups (s_W^2) is called the ANOVA *F* statistic as follows:

$$F_{stat} = \frac{s_B^2}{s_W^2} \tag{8-6}$$

Therefore, the test hypotheses are restated as:

$$H_0(\text{Null Hypothesis}): F_{stat} \le F_{table} \tag{8-7}$$

$$H_a(\text{Alternative Hypothesis}):\ F_{stat} > F_{table} \tag{8-8}$$

The variance between groups (s_B^2) is calculated as follows:

$$s_B^2 = \frac{SS_B}{df_B} \tag{8-9}$$

The SS_B (sum of squares between) is calculated as follows:

$$SS_B = \sum_{i=1}^{K} n_i(\overline{x}_i - \overline{x})^2 \tag{8-10}$$

Where n_i represents the size of group *i*, $\overline{x}_i$ represents the mean of group *i*, $\overline{x}$ represents the grand mean and *k* represents the number of groups.

The df_B (degree of freedom between groups) is calculated as:

$$df_B = k - 1 \tag{8-11}$$

For calculating variance within groups (s_W^2), the following equation is used:

$$s_W^2 = \frac{SS_W}{df_W} \tag{8-12}$$

Where the sum of square within SS_B is calculated as:

$$SS_W = \sum_{i=1}^{K} (n_i - 1) S_i^2 \tag{8-13}$$

and the degree of freedom is calculate as:

$$df_w = N - k \tag{8-14}$$

Where S_i represents the variance of group *i and N* represents the size of all groups.

After conducting the ANOVA test, the Scheffè Post Hoc Comparisons follows, of which procedure is described according to Sytsma [95] as follows:

1. When the null hypothesis is rejected calculate the number of pairs required to be compared, which is:

$$n_{pairs} = \frac{k(k-1)}{2} \tag{8-15}$$

   Where k is the number of groups needed to be compared.

2. Calculate all possible mean differences for each pair to be compared:

$$\overline{x}_a - \overline{x}_b \tag{8-16}$$

3. Calculate confidence interval around each mean difference as follows:

$$\overline{x}_a - \overline{x}_b \pm \sqrt{(k-1) F_{table} S_w^2} \sqrt{\frac{1}{n_a} + \frac{1}{n_b}} \tag{8-17}$$

   Where n_a and n_b are the respective sample sizes for groups *a* and *b*.

4. If a confidence interval is either totally positive or totally negative, this indicates that groups *a* and *b* are significantly different, otherwise no difference exists.

Excel software already provides ANOVA testing, which is used throughout all case studies. For the Scheffè Post Hoc Comparisons, Sytsma [95] provides an Excel template worksheet that conducts the comparisons, which is also used in this research.

8.3 Case Studies

8.3.1 Hypothetical Arterial

In this case study, we compare the offset plan optimized by the developed method against the full enumeration method, a randomly bad plan, the dominance method and the TRANSYT-7F method.

8.3.1.1 Arterial Description

Figure 8-2 shows the studied example of a two-way arterial represented in AIMSUN. The flow in the main direction is 650 veh/h and 585 veh/h in the opposite direction, while the flows in the crossing direction are shown in the Figure. The detailed layout, the phasing and the turning proportions of intersections k1, k2, and k3 are presented graphically in Annex B. Here are the assumed parameters:

- Intergreen time is 5 s;
- Cycle time is 70;
- Main Direction is EB;
- Green time is 20s in both directions at each intersection;
- Required green time for crossing traffic at each intersection is 15s;
- Free flow speed and backward wave speed are 50 km/h;
- Jam density is 120 veh/km;
- Saturation flow is 1800 veh/h;
- Time interval is 1 s;
- Modeling horizon is13 cycles (equivelent to 910 time intervals).

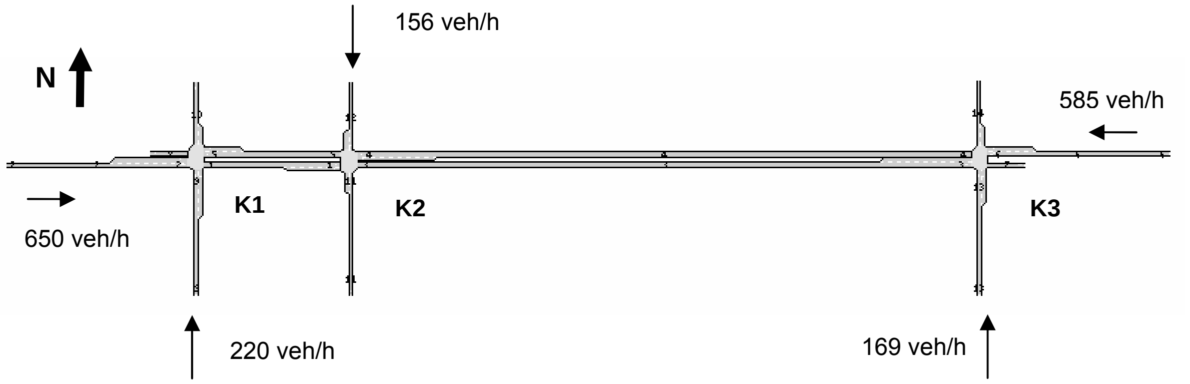


Figure 8-2: A two-way arterial with 3 signalized intersection (AIMSUN Representation)

Figure 8-3 shows the CTM representation of the hypothetical two-way arterial with the required cells (180 cells) and links. For the simulation of this network using the developed simulator with a time horizon of 910 s, one needs 400 ms on a PC (Athlon-1 GHz).

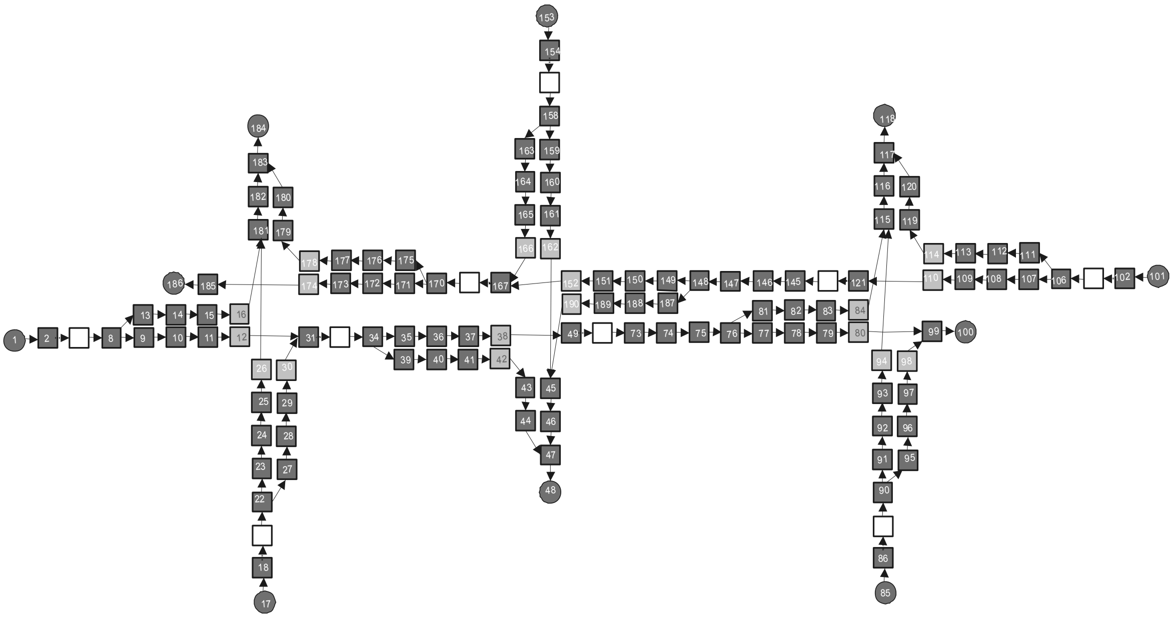


Figure 8-3: Cell representation of the arterial case study

8.3.1.2 Full Enumeration Results

With 3 intersections and a common cycle time of 70 s, the solution space has 4900 (70^{3-1}) possible solutions in this case study. Because the time needed to calculate the delay on this arterial based on the developed analysis module in this dissertation is 400 ms, the calculation time needed is approximately 1960 s (4900 × 0.4 s), which is considered to be feasible.

Figure 8-4 plots part of the full enumeration results, where the horizontal axis presents the offset at intersection K3 and the vertical axis presents the delays of the developed simulator. The offset in the Figure ranges from 0 to the cycle time (70 s) given that the offset of K1 and K2 are kept to be 0 and 1 respectively. The optimal offsets found from these results were 0, 1 and 37 s for intersections K1, K2 and K3 respectively.

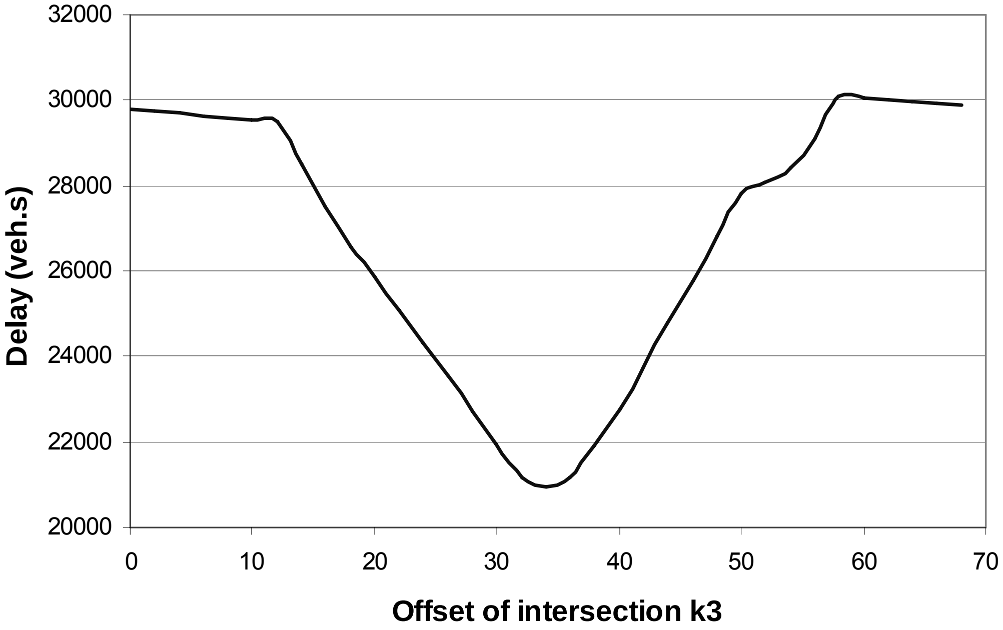


Figure 8-4: **Delays at varied offsets of intersection K3 given that offset at K2 is 0**

These optimal offsets were fed into AIMSUN and the simulation was conducted with 30 replications.

8.3.1.3 Dominance Method Results

An Excel worksheet was prepared for the calculation of offsets based on the dominance method. Since the flow in direction K1-K2-K3 was higher than the opposite direction, it is considered as the main direction. Based on this method, the optimal offsets were 0, 8, 40 s for intersections K1, K2, K3 respectively.

8.3.1.4 TRANSYT-7F Results

For the optimization with TRANSYT-7F, the step-wise multi-cycle simulation was used as a simulation method while the genetic algorithm was used as an optimization method. Furthermore, the disutility index was chosen as an objective function. TRANSYT-7F output data for the arterial case study are attached in Annex C. After 200 s of CPU time on a PC (Athlon- 1 GHz), the optimal offsets were 0, 9, 44 s for intersections K1, K2, K3 respectively.

8.3.1.5 Developed PGA and SGA Results

For the optimization with the PGA, 25 generations were chosen with a population of 10 chromosomes, while with the SGA, 8 generations were chosen for each sequence with a population of 5 chromosomes. With both PGA and SGA the same optimal offset were found, whereas the SGA could reduce the CPU-time from 100 s to 20 s. The optimal offsets were 0, 1, 37 s for intersections K1, K2, K3 respectively.

8.3.1.6 Statistical Comparison

Figure 8-5 presents the histogram of the system delay obtained from AIMSUN, when using the optimal offsets obtained from the full enumeration method or the developed optimization method for this case study. For the other methods, the histograms are provided in Annex D.

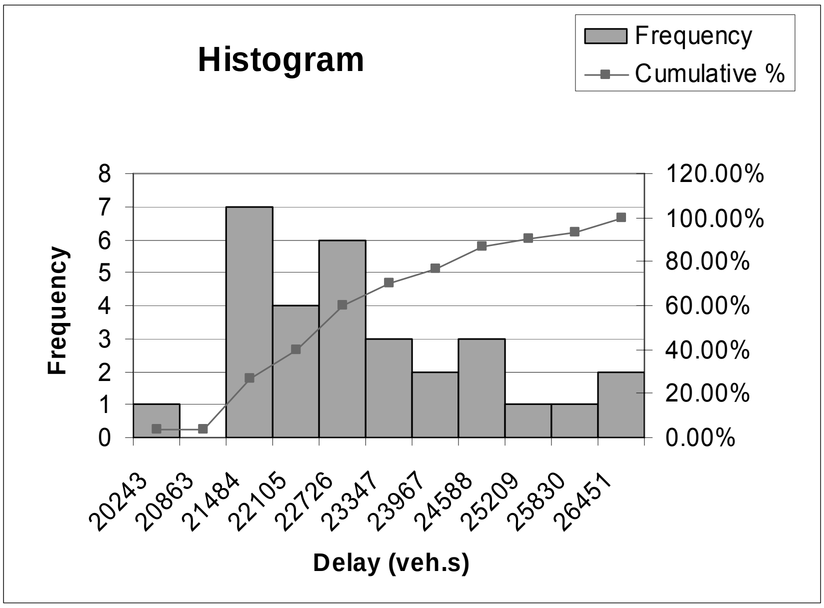


Figure 8-5: **Histogram of the system delay obtained from AIMSUN using the optimal offsets based on Full Enumeration Method for the arterial case study**

The application of the ANOVA test rejected the null hypothesis, implying that at least one coordination method gives significantly different results from at least one other method in this case study. This is shown in the ANOVA table (Table 8-1) generated by Excel; the F_{stat} (3.59) was larger than F_{table}(3.10). Hence, a Scheffè Post Hoc Comparison was conducted next. The summary results of this comparison are provided in Table 8-2.

The delay comparison of each of coordination method is provided in Figure 8-6. From the Figure and the statistical analysis, the following conclusions were made:

- SGA is significantly better than dominance method.
- No significant difference was found between SGA and TRANSYT-7F.

Table 8-1: ANOVA test results for arterial case study

Sammary

Groups	*Count*	*Sum*	*Average*	*Variance*
SGA/Full Enumeration	30	689564.4281	22985.48094	2531908.571
TRANSYT-7F	30	711673.4506	23722.44835	3440685.439
Dominance	30	726495.8232	24216.52744	3643830.244

ANOVA

Source of Variation	*SS*	*df*	S^2	F_{stat}	F_{table}
Between Groups (B)	23027106.16	2	11513553.08	3.591840202	3.101296
Within Groups (W)	278876303.4	87	3205474.751		
Total	301903409.5	89			

Table 8-2: Scheffè Post Hoc Comparison results for arterial case study

Scheffe Confidence Intervals & Significance Indicator:

	SGA/Full Enumeration			**TRANSYT-7F**			**Dominance**		
Group	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**
SGA/Full Enumeration									
TRANSYT-7F	1888.3	-414.33							
Dominance	2382.3	79.75	SIGNIF	1645.4	-657.22				

LL- lower limit of confidence interval around mean difference
UL- Upper limit of confidence interval around mean difference
SIGNIF- There is significance difference between the two methods

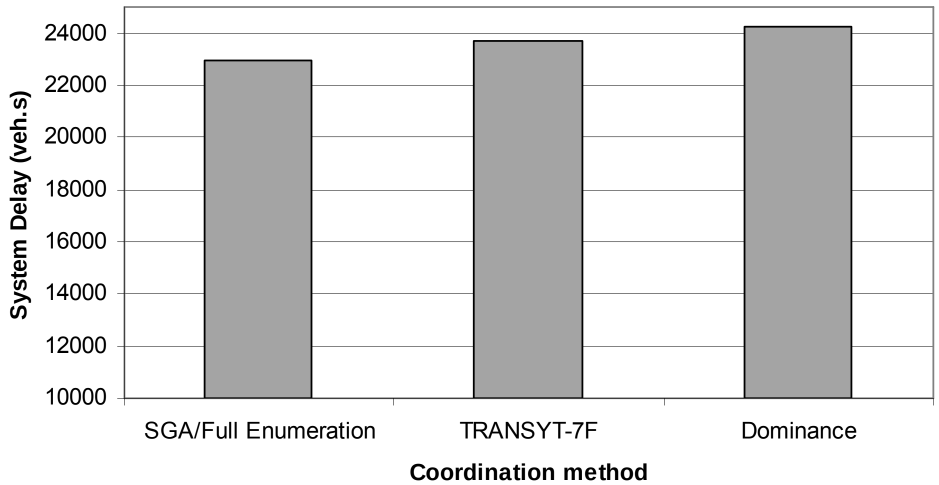


Figure 8-6: Delay comparison of coordination methods for the arterial case study

Table 8-3 presents the optimal offsets, the relative changes of the AIMSUN delay and the CPU time. It can be clearly seen from these results that there is a significant potential for optimization of 35% between the worst and the best solution. With both PGA and SGA the best solution could be found, whereas the SGA could reduce the CPU-time from 100 s to 20 s. With the dominance method and TRANSYT-7F one can obtain reasonable results, however, the absolute optimum could not be found. Therefore, the developed SGA is superior to both TRANSYT-7F and dominance method.

Table 8-3: Optimal offsets of all coordination methods for arterial case study

Method	**Offsets at intersections**			**Mean of 30 replication AIMSUN delay (veh.s)**	**Relative changes %**	**CPU time**
	1	**2**	**3**			
Full enumeration	0	1	37	22985	0%	15 min
Random bad coordination	0	58	46	31000	35%	15 min
Dominance method	0	8	40	24216	5.4%	1 s
PGA	0	1	37	22985	0%	100s
SGA	0	1	37	22985	0%	30s
TRANSYT-7F	0	9	44	23722	3.2%	200 s

8.3.2 A Small Realistic 6-Intersection Network

In this case study, we compare the offset plan optimized by the developed method against the existing time plan, the full enumeration method and a randomly bad plan. Because some of the intersections have more than 7 phases, which is the maximum number of phases that TRANSYT-7F software can model, it would not normally be possible to include the TRANSYT-7F method in this comparison. However, by combining similar phases, the overall number of phases was reduced, allowing TRANSYT-7F to model the outcome. Dominance method can also be included if we divide the network into different arterials.

8.3.2.1 Network Description

The network is a part of List District in Hanover, Germany. This network consists of six intersections already shown in Figure 6-5. The six signalized intersections of the network have a common cycle time of 90s. For the simulation of this network using the developed simulator with a time horizon of 900s, one needs 700ms on a PC (Athlon- 1 GHz).

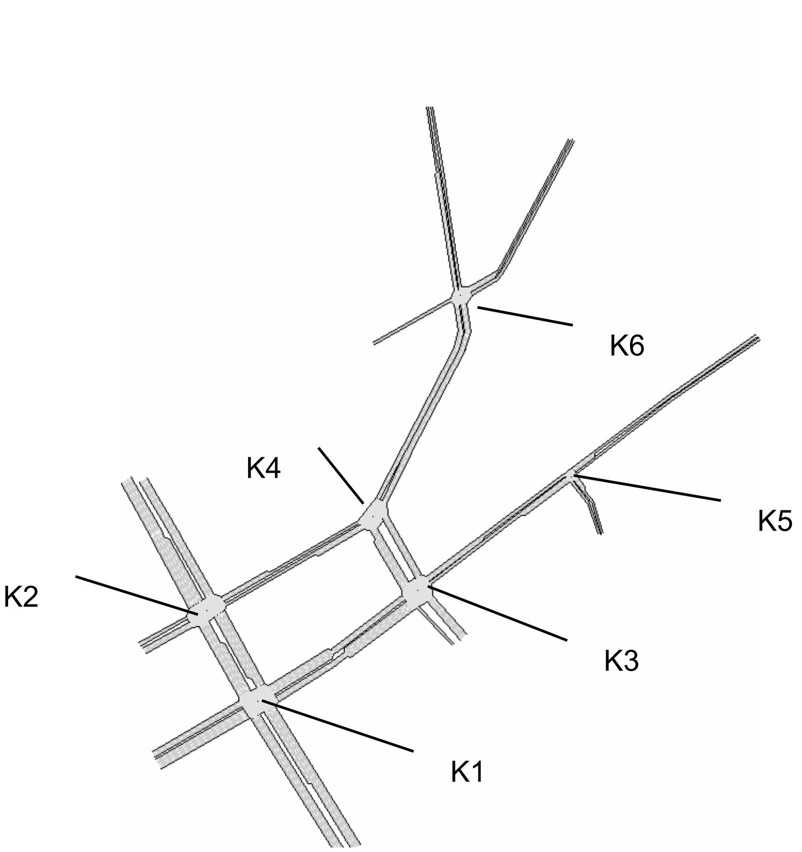


Figure 8-7: **Case study of the realistic network which is a part of List District in Hanover**

The detailed layouts of all 6 intersections are provided in Annex B. The existing timing plans for all intersections and the input flows and turning proportions are also presented. The layout of the network is provided again in this chapter in order to follow easily the comparsion results in the following sections.

8.3.2.2 Full Enumeration Results

In this network, five offsets are iterated, since it has 6 intersections. The offsets of intersections 1, 2, 3, 5, and 6 are chosen to be the control variables, and the offset of intersection 4 is set to be always at 0.

In the full enumeration algorithm, each offset variable has possible solutions ranging from 0 to cycle length (90s). With 5 optimizing variables and 90 possible solutions of each variable, the calculation time needed for the full enumeration on a PC (Athlon- 1 GHz) is expected to be 131 years (i.e. $90^5 \times 0.7$ s = 4.133×10^9 s).

Because of this unrealistic calculation time, a time step of 10 s is used for each offset which results in each offset being iterated 9 (90/10) times. Using this time step, the calculation time needed for the full enumeration is dramatically reduced to 11.48 hrs (i.e. 95 × 0.7 s = 4.133 ×

104 s), which is considered to be feasible. The optimal offsets found from this full enumeration were 0, 80, 10, 0, 40, 0 s for intersections 1, 2, 3, 4, 5, and 6 respectively and the delay found to be (41699.2 veh.s) at these offset values.

Table 8-4: Range of variation of the second full enumeration for realistic network case study

		Range of offsets			
intersection	**Old optimum offset**	**From**	**To**	**Time step**	**Iteration**
1	0	80	10	5	5
2	80	70	90	5	5
3	10	85	25	5	7
4	0	0	0	5	1
5	40	30	50	5	5
6	0	80	20	5	7

Starting from the obtained optimal values by the first full enumeration, another full enumeration with 5s time step is made around these values. Table 8-4 presents the proposed range of variation of each offset which brings about 6125 possible solutions that need 4287.5s of calculation time. The optimal offsets found from this full enumeration were 0, 85, 15, 0, 50, 0 s for intersections 1, 2, 3, 4, 5, and 6 respectively and the delay found to be (41389.6 veh.s) at these offset values which is already less the one (41699.2 veh.s) found by the first full enumeration.

A third full enumeration is made with a 2s time step based on the obtained optimal values by second full enumeration. Table 8-5 presents the proposed range of variation of each offset which results in 15125 possible solutions that need 10587s of calculation time. The optimal offsets found from this full enumeration were 87, 82, 14, 0, 48, 88 s for intersections 1, 2, 3, 4, 5, and 6 respectively and the delay found to be (40695.9 veh.s) at these offset values which is already less the one (41389.6 veh.s) found by the second full enumeration.

Table 8-5: Range of variation of the third full enumeration for realistic network case study

		Range of offsets			
intersection	Old optimum offset	From	To	Time step	Iteration
1	0	85	95	2	5
2	80	80	90	2	5
3	10	86	18	2	11
4	0	0	0	2	1
5	40	30	52	2	11
6	0	86	6	2	5

Figure 8-8 shows part of the full enumeration results when fixing the offsets of intersections 1, 2, 3, 4 and 6 to 0 s and varying the offset of intersection 5.

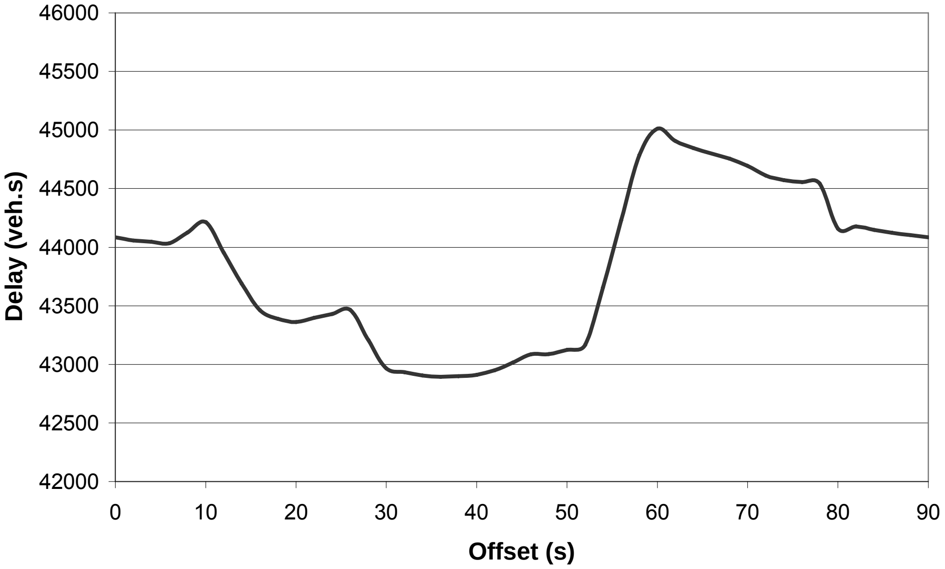


Figure 8-8: Part of full enumeration results for realistic network case study (part 1)

8.3.2.3 Dominance Method Results

Based on the traffic demand and the turning proportions already presented in Table 6-3, the heaviest traffic directions are determined to be from intersection K6-K4-K2-K1-K3-K5 as represented in Figure 8-9 with bold arrows. The optimal offsets found from this method were 0, 3, 35, 0, 64, 73s for intersections k1, k2, k3, k4, k5 and k6 respectively.

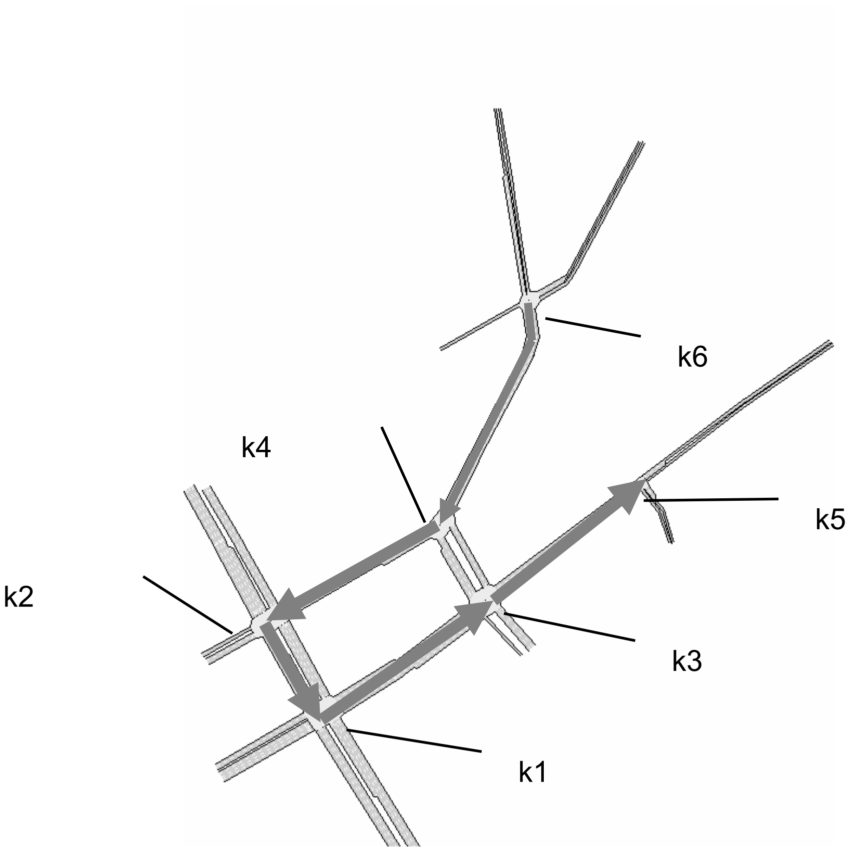


Figure 8-9: Description of the coordination direction for the dominance method

8.3.2.4 TRANSYT-7F Results

For the optimization with TRANSYT-7F, the step-wise multi-cycle simulation was used as a simulation method while the genetic algorithm was used as an optimization method. The disutility index was chosen as an objective function. TRANSYT-7F output data for this case study are attached in Annex C. After 30 minutes of CPU time on a PC (Athlon- 1 GHz), the optimal offsets were 83, 61, 11, 36, 46 and 89 s for intersections k1, k2, k3, k4, k5 and k6 respectively.

8.3.2.5 Developed PGA and SGA Results

For the PGA-based optimization 20 generations with a population of 50 chromosomes were chosen using a 1s time step. Since each run of the developed traffic analysis model takes 0.7 s, the computing time to run 20 generations with 50 individuals each generation amounts to approximately 700 s; this allows for online application. The mutation rate is assumed to be 0.05 and the crossover rate is 0.7. Each decision variable (offset) is coded as a 7-bit binary string, allowing for a total of 128 (2^7) distinctive values. For offset optimization, offset values in this case study are not higher than the cycle time which is equal to 90 time intervals (1s each); therefore, a seven-bit binary string is sufficient for the simulation with 1s intervals. In this case study using the PGA with 1s interval, we optimize 5 offsets and fix the value of offset at intersection 4 to 0.

For the optimization with the SGA, 10 generations were chosen for each sequence with a population of 5 chromosomes. It was easy to determine the best order of sequence optimization in this case study because of the low interdependencies within the network, which is 6, 4, 2, 1, 3, and 5 respectively. Each decision variable is coded as a seven-bit binary string, allowing for a total of 127 (2^7) distinctive values. Therefore the length of chromosome in each sequence will be 7 bits and the chromosome structure is like [(1001010)].

Table 8-6 presents the optimal offsets found by the PGA and SGA. The results show that the offsets are relatively consistent with each other and the delays are relatively equal.

Table 8-6: Optimal offsets results based on SGA methods for the realistic network

Method	**Intersections**					
	1	**2**	**3**	**4**	**5**	**6**
PGA	0	81	18	0	33	21
SGA	87	84	9	0	40	0

8.3.2.6 Statistical Comparison

Histograms of the AIMSUN system delay using the offsets based on the different comparison methods for the small network case study are provided in Annex D. The application of the ANOVA test also rejected the null hypothesis in this case study. This is shown in the ANOVA table (Table 8-7) generated by Excel; the F_{stat} (21.27) was larger than F_{table}(2.27). Hence, a Scheffè Post Hoc Comparison was conducted next. The summary results of this comparison are provided in Table 8-8.

Table 8-7: ANOVA test results for small network case study

Sammary

Groups	*Count*	*Sum*	*Average*	*Variance*
Enumeration	30	1155923	38530.8	14811441.4
SGA	30	1162437	38747.9	14734040.1
PGA	30	1197707	39923.6	14896142.0
Transyt	30	1329931	44331.0	15978913.6
dominance	30	1367063	45568.8	14077523.6
Existing	30	1348585	44952.8	16575854.6

ANOVA

Source of Variation	*SS*	*df*	S^2	F_{stat}	F_{table}
Between Groups (B)	1614355338	5	322871068	21.2709248	2.26606171
Within Groups (W)	2641143542	174	15178985.9		
Total	4255498880	179			

Table 8-8: Scheffè Post Hoc Comparison results for small network case study

Scheffe Confidence Intervals & Significance Indicator:

	Full Enumeration			SGA			PGA			Transyt			dominance		
Group	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**
Full Enumeration															
SGA	3603.21	-3168.9													
PGA	4778.88	-1993.3		4561.74	-2210.4										
Transyt	9186.34	2414.19	SIGNIF	8969.21	2197.06	SIGNIF	7793.54	1021.39	SIGNIF						
dominance	10424.1	3651.92	SIGNIF	10206.9	3434.79	SIGNIF	9031.28	2259.12	SIGNIF	4623.81	-2148.3				
Existing	9808.14	3035.99	SIGNIF	9591.01	2818.86	SIGNIF	8415.34	1643.19	SIGNIF	4007.88	-2764.3		2770.14	-4002	

LL- lower limit of confidence interval around mean difference
UL- Upper limit of confidence interval around mean difference
SIGNIF- There is significance difference between the two methods

The delay comparison of each coordination method is provided in Figure 8-10. From the Figure and the statistical analysis, the following conclusions were made:

- SGA, PGA and full enumeration are significantly better than both the TRANSYT-7F and dominance methods and also better than the existing offset plan.
- No significant difference was found between the full enumeration and both SGA and PGA.
- No significant difference was found between SGA and PGA.
- No significant difference was found between the existing offset plan and both the TRANSYT-7F and dominance methods.

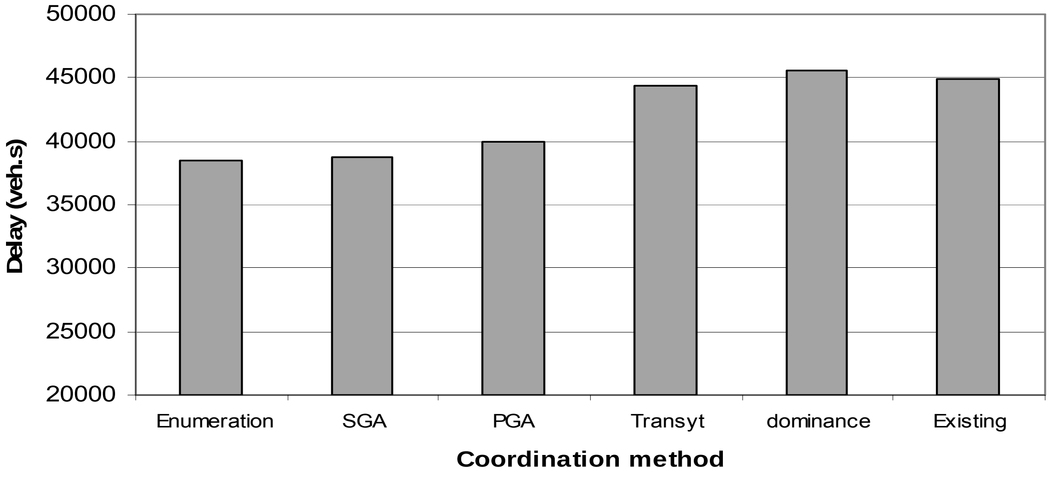


Figure 8-10: Delay comparison of coordination methods for small network case study

Table 8-9 presents the optimal offsets, the relative changes of the AIMSUN delay and the CPU time. The results show that for the real network, there is a potential for improvement of about 17% less delay than the existing coordination plan. With both the PGA and the SGA, a solution close to the absolute optimum could be found. They proved to be superior to both the TRANSYT-7F and the dominance methods.

Table 8-9: Optimal offsets results based on different methods for the realistic network

Method	Offset at intersection						Mean of 30 replication AIMSUN delay (veh.s)	Relative changes %	CPU time
	1	2	3	4	5	6			
Enumeration	87	82	14	0	48	88	38530	0.0%	24 hr
SGA	87	84	9	0	40	0	38747	0.6%	175 s
PGA	0	81	18	0	33	21	39923	3.6%	700 s
TRANSYT-7F	83	61	11	36	46	89	44331	15.1%	1800 s
Dominance	0	3	35	0	64	73	45568	18.3%	-
Existing	0	7	0	3	83	86	44952	16.7%	-

8.3.3 Grid Network with 12 Intersections

In this case study, we compare the plan optimized by the developed method for offset optimization against TRANSYT-7F and dominance method. Full enumeration is impossible even if one has a 10s time step; the calculation time needed for iterating 11 offset variables is estimated to be 18.69 years.

8.3.3.1 Grid Network Description

Figure 8-11 shows the layout of the 12-intersection grid network represented in AIMSUN. The input flows shown in the Figure range from 500 veh/h to 900 veh/h. The detailed layout of intersections, the phasing, and green times assigned for each phase are provided in Annex B.

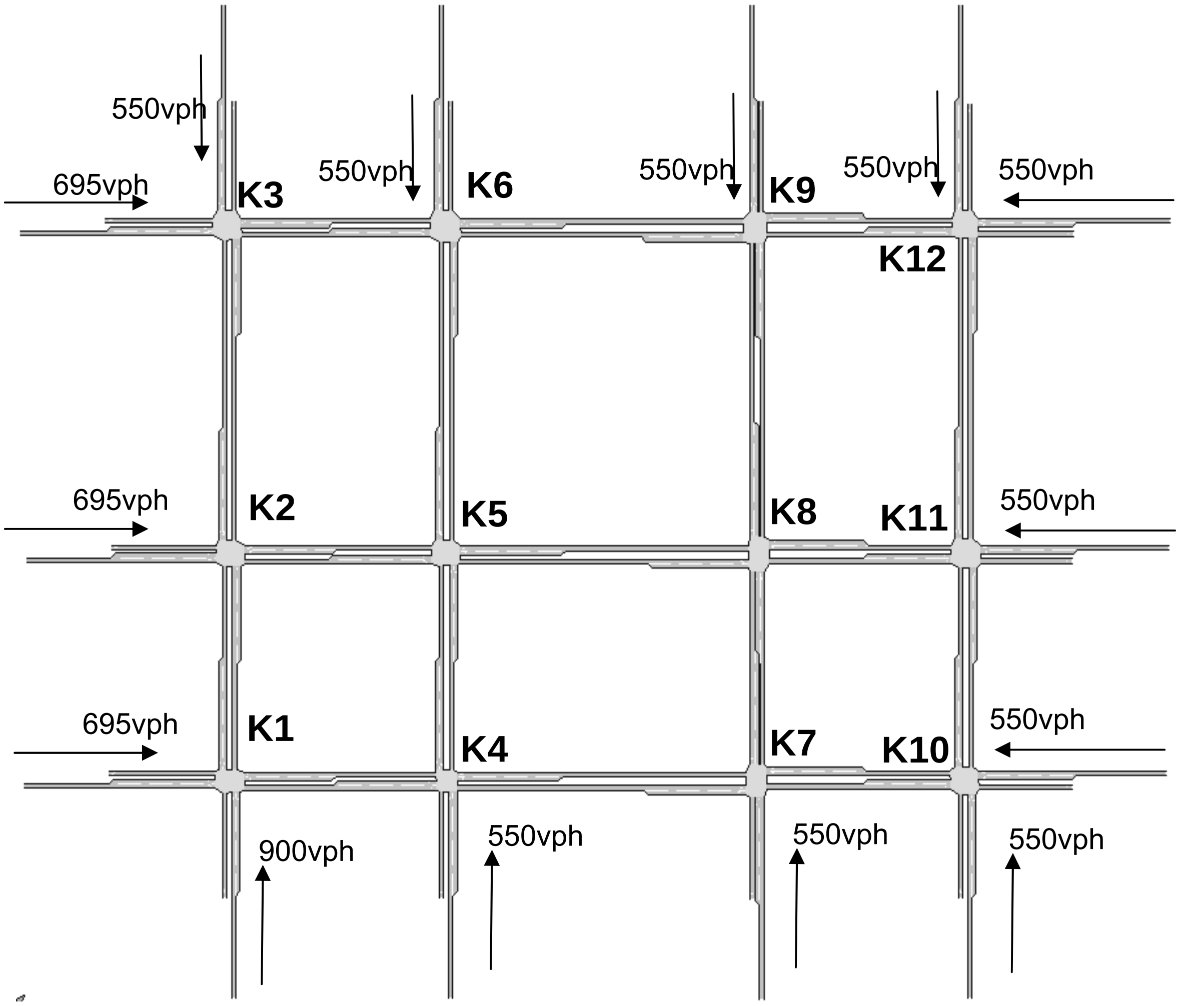


Figure 8-11: layout of the 12-intersection grid network (AIMSUN Representation)

Here are other assumed parameters:

- Intergreen time is 5 s;
- Cycle time is 90s;
- Each intersection has three phases ;
- Free flow speed and backward wave speed are 50 km/h;
- Jam density is 120 veh/km;

- Saturation flow is 1800;
- Time steps are 1,2.5, and 5 s;
- Modeling horizon is 10 cycles (equivelent to 900 time intervals).

8.3.3.2 Developed PGA and SGA Results

If the duration of the time step is increased, e.g. from 1s to 5s, then the calculation time (CPU time) of the developed simulator for this grid network is exponentially reduced (from 4000ms to 220ms on a PC Athlon 1Ghz). This is due to the reduction in the following two dimensions: (1) the time dimension, where the simulation time is reduced from 900 time steps (time step =1s) to 180 intervals (time step =5s) and (2) the space dimension, where the number of cells is reduced approximately from 920 to 214 cells. Three time steps (1, 2.5 and 5s) are studied in this case study.

Developed PGA

For the PGA-based optimization with 5s interval, 20 generations were chosen with a population of 200 chromosomes. Since each run of the developed simulator takes 0.22 s, the computing time to run 20 generations with 200 individuals amounts to approximately 900 s for each generation, which allows for online use. For the purpose of investigating what happens when the number of generations is increased, we extend the number of generations to 50 (2244 s). The mutation rate is assumed to be 0.1 and the crossover rate is 0.7. Each decision variable (offset) is coded as a five-bit binary string, allowing for a total of 32 (2^5) distinctive values. For offset optimization, offset values are not higher than the cycle time which is equal to 18 time intervals (5s each); therefore, five-bit binary strings are sufficient for the simulation with 5s intervals. In this case study, using the PGA method with 5s interval we optimize 11 offsets and fix the value of offset at the 12th intersection to 0. Then the chromosome length will be 55 bits (11×5) and the chromosome structure is like $[(10010)(01010)\ldots^{11}(11101)]$.

When optimizing the same network with 2.5s intervals, each run takes 925 ms on a PC Athlon 1Ghz. Since we are constrained with a 900s calculation time to permit online application, 5 generations with a population of 200 chromosomes were chosen. However, we ran the program for 2.5 hours to investigate the convergence and see the applicability for offline use. The mutation rate is also assumed to be 0.1 and the crossover rate is 0.7. Each decision variable (offset) is coded as a six-bit binary string, allowing for a total of 64 (2^6) distinctive values. For offset optimization with the 2.5s interval, offset values are not higher than the cycle time which is equal to 36 time intervals (2.5s each); therefore, six-bit binary strings are sufficient for the simulation with 2.5s intervals. In this case study using the PGA method with a 2.5s interval, the chromosome length will be 66 bits (11×6) and the chromosome structure is like [(100110)(011010)…^{11}(111101)].

With 1 s interval, each run takes 4000 ms on a PC Athlon 1Ghz. A number of 15 generations with a population of 200 chromosomes were chosen that needs 2.5 hours of CPU time. The mutation rate is also assumed to be 0.1 and the crossover rate is 0.7. Each decision variable (offset) is coded as a seven-bit binary string, allowing for a total of 127 (2^7) distinctive values. Therefore, the chromosome length will be 77 bits (11×7) and the chromosome structure is like [(1001010)(0110010)…^{11}(1110101)].

Developed SGA Including Search Order

For the optimization with the developed SGA, a total of 11 relative offsets are needed to be optimized. Four groups of relative offsets were found in this case study based on the developed search order. These are:

- The first group has two variables: (1) between intersections K1 and K2; and (2) between intersections K2 and K3.
- The second group has three variables: (1) between intersections K1 and K4; (2) between intersections K4 and K7; and (3) between intersections K7 and K10.
- The third group has three variables: (1) between intersections K2 and K5; (2) between intersections K5 and K8; and (3) between intersections K8 and K11.
- The fourth group has three variables: (1) between intersections K3 and K6; (2) between intersections K6 and K9; and (3) between intersections K9 and K12.

For the optimization using a 5s time interval, each relative offset is initialized with the time needed to travel between the two considered intersections under free conditions. The first group of variables used 6 generations with a population of 32 chromosomes. Each decision variable is coded as a five-bit binary string, allowing for a total of 32 (2^5) distinctive values. Therefore, the chromosome length for this group will be 10 bits (2×5) and the chromosome structure is like [(10010) (01010)]. The other three groups used 6 generations with a population. In these groups, decision variables are also coded as a five-bit binary string, allowing for a total of 32 (2^5) distinctive values. Therefore, the length chromosome for these groups will be 15 bits (3×5) and the chromosome structure is like [(10010) (01010) (01010)]. Based on the previous assumptions, the total number of runs was 1056 (1 group ×32 individuals × 6 generations + 3 groups × 48 individuals × 6 generations). When each run takes 220 ms then the calculation time will be 232 s.

The assumptions stated above are used for the optimization with 2.5 s, with the following exceptions:

- Each decision variable is coded as a six-bit binary string, allowing for a total of 64 (2^6) distinctive values, this results in having a chromosome length of 12 bits for the first group of optimization and 18 bits for the other three variables.
- When each run takes 925 ms, the calculation time will be 1056 times 925 ms equal to 977s.

For the optimization with 1 s the above assumptions apply, with the following exceptions:

- Each decision variable is coded as a seven-bit binary string, allowing for a total of 127 (2^7) distinctive values, this results in having a chromosome length of 14 bits for the first group of optimization and 21 bits for the other three variables.
- The number of generation was doubled. When each run takes 4000 ms, the calculation time will be 1056 × 2 × 4000 ms equal to 8448s.

Developed SGA Excluding Search Order

To investigate the importance of the developed search order, we optimize the same grid network using the SGA with 5s interval ignoring the search order. Instead of that we used the following 4 groups:

- The first group included intersections K1-K2-K3;
- The second group included intersections K4-K5-K6;
- The third group included intersections K7-K8-K9;
- The fourth group included intersections K10-K11-K12;

Other assumptions in terms of the previously mentioned chromosome structure remained the same.

The Results

Figure 8-12 shows a comparison between the methods cited, and plots the delay (veh.s) versus the CPU run time of a PC Athlon 1Ghz. The results show that the quasi absolute optimum could be found after 230s using the SGA with a 5s interval and after 900s using the SGA with a 5s interval; while it was not possible using PGA even after 2200s. Therefore, the SGA with the help of the proposed search order determination is applicable, and not only could find the quasi absolute optimum but could also shorten the computation time which allows for online optimization. However, when the developed search order is not used, the SGA failed to find this quasi absolute optimum.

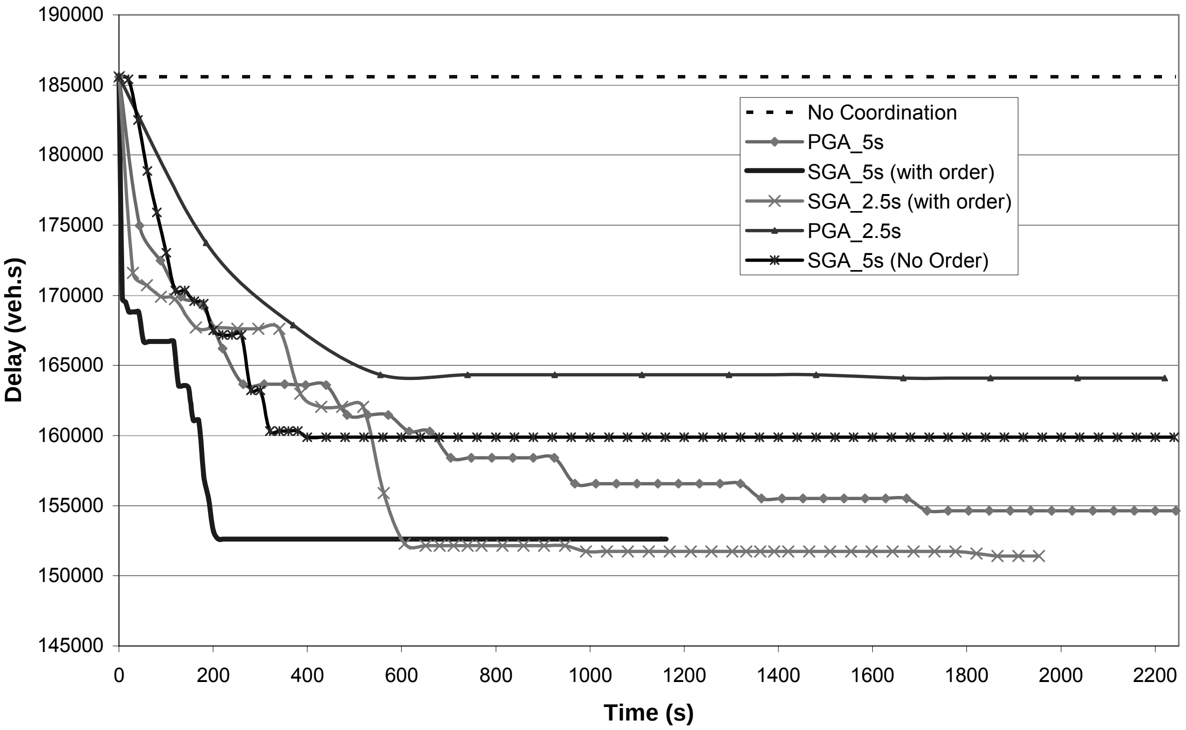


Figure 8-12: Delay versus CPU run time of a pc Athlon 1 GHz using different methods for the grid network case study

The SGA and PGA were also applied with a 1 s time step to suit for offline application. Figure 8-14 presents the comparison between a 1s and a 2.5 time steps for both SGA and PGA. Based on this Figure, the following conclusions could be made:

- It was not possible to reach the quasi absolute optimum using PGA for 1s after 9000s. Therefore, PGA with 1s interval is not applicable for offline use.
- PGA with a 2.5 s step time also could not reach the quasi absolute optimum, however it was better than the 1s step time.
- SGA with 1s step time could not reach the quasi absolute optimum, however it was similar to PGA with 2.5s and better than PGA with 1s.

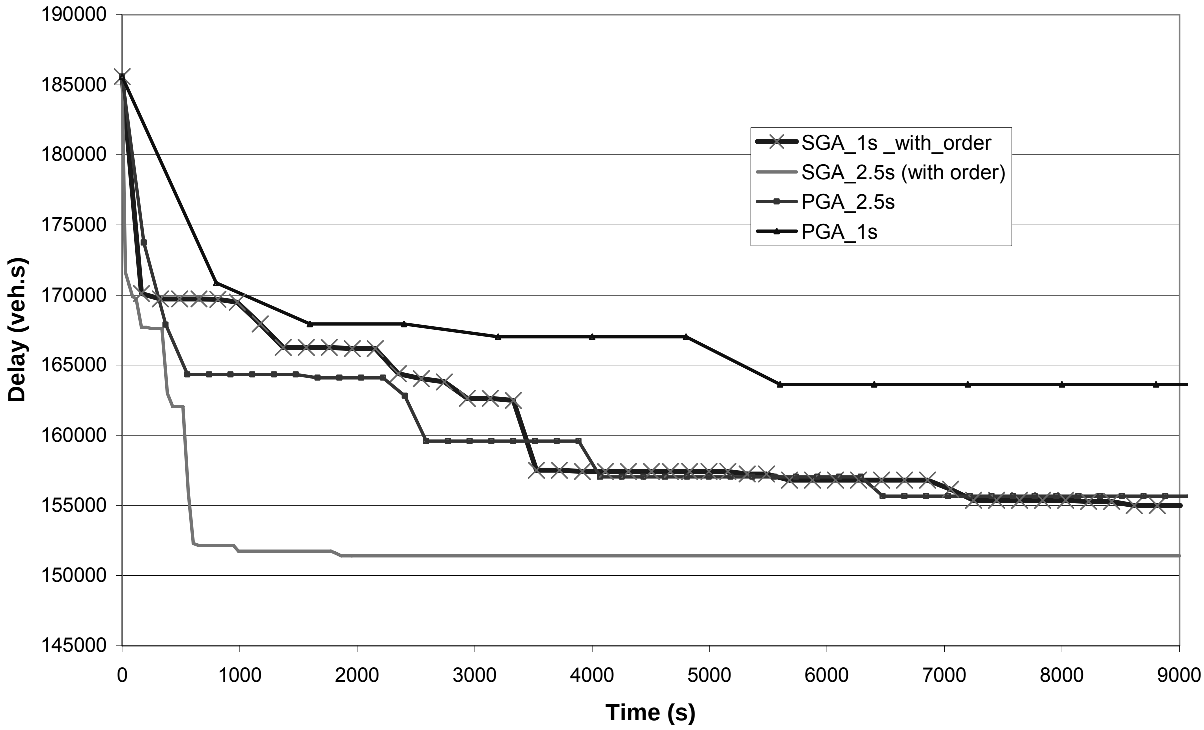


Figure 8-13: **Delay versus CPU run time of a PC Athlon 1 GHz using 1 and 2.5s step times for the grid network case study**

Table 8-10 presents the optimal offsets found by the SGA in both 5s and 2.5s intervals. The results show that the offsets are consistent with each other and that AIMSUN delays are relatively equal.

Table 8-10: **Optimal offsets results based on SGA methods for the 12 intersection case study**

Method	**Intersections**												**AIMSUN Delay**
	1	2	3	4	5	6	7	8	9	10	11	12	
SGA (5s)	0	5	40	70	70	15	80	80	30	20	20	65	158233
SGA (2.5s)	0	5	38	65	63	15	78	75	28	18	20	58	157394

8.3.3.3 Dominance Method Results

In this case study, four arterials were selected for the coordination using the dominance method. These four arterials were chosen according to the grouping of intersections already used in SGA. The optimal offsets for the first arterial (k1-k2-k3) were 0, 10 and 25s respectively. For the second arterial (k1-k4-k7-k10), the optimal offsets were 0, 10, 25 and 35s respectively. For the third arterial (k2-k5-k8-k11), the optimal offsets were 10, 20, 35 and 45s respectively. Finally for the fourth arterial (k3-k6-k9-k12), the optimal offsets were 25, 35, 50 and 60 s respectively.

8.3.3.4 TRANSYT-7F Results

The step-wise multi-cycle simulation was used as a simulation method while the genetic algorithm was used as an optimization method. The disutility index was chosen as an objective function. TRANSYT-7F output data for the arterial case study are attached in Annex C. The optimal offsets found are presented in Table 8-11. The running time was 1.5 hours on a PC (Athlon- 1 GHz)

Table 8-11: Optimal offsets results based on TRANSYT-7F methods for the 12 intersection case study

Method	**Intersections**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
TRANSYT-7F	0	4	14	46	47	57	10	62	70	44	11	14

8.3.3.5 Statistical Comparison

In this case study we statistically compare the SGA method with a 2.5s step time, to both the dominance TRANSYT-7F and dominance methods. The application of the ANOVA test rejected the null hypothesis. This is shown in the ANOVA table (Table 8-12) generated by Excel; the F_{stat} (148) was larger than F_{table}(3.1). Hence, a Scheffè Post Hoc Comparison was conducted next. The summary results of this comparison are provided in Table 8-13.

Table 8-12: ANOVA test results for grid network case study

Sammary

Groups	*Count*	*Sum*	*Average*	*Variance*
SGA	30	4721848.36	157394.9452	13091874.92
Dominance	30	5106086.13	170202.8709	15686855.75
TRANSYT-7F	30	5186361.57	172878.7191	12827919.92

ANOVA

Source of Variation	*SS*	*df*	S^2	F_{stat}	F_{table}
Between Groups (B)	4109503759	2	2054751879	148.1555364	3.101295757
Within Groups (W)	1206592867	87	13868883.53		
Total	5316096626	89			

Table 8-13: Scheffè Post Hoc Comparison results for grid network case study

Scheffe Confidence Intervals & Significance Indicator:

	SGA			**Dominance**			**TRANSYT-7F**		
Group	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**	**LL**	**UL**	**Sign**
SGA									
Dominance	15202.68	10413.17	SIGNIF						
TRANSYT-7F	17878.53	13089.02	SIGNIF	5070.607	281.0898	SIGNIF			

LL- lower limit of confidence interval around mean difference
UL- Upper limit of confidence interval around mean difference
SIGNIF- There is significance difference between the two methods

The delay comparison of each coordination method is provided in Figure 8-14. From the Figure and the statistical analysis, it is clear SGA is better than both the TRANSYT-7F and dominance methods.

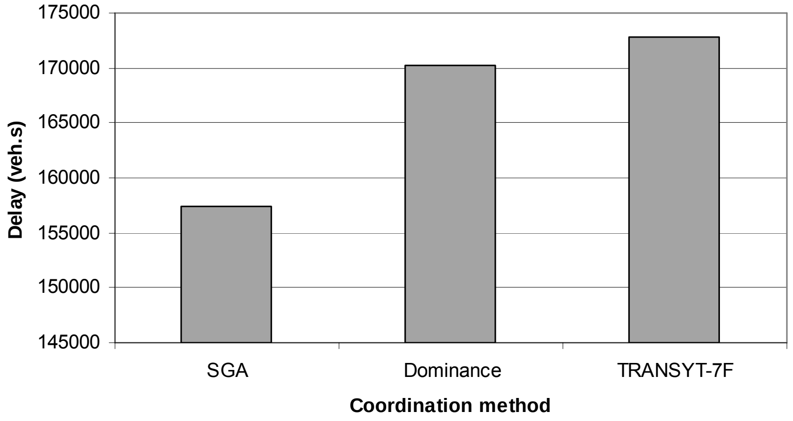


Figure 8-14: Delay comparison of coordination methods for grid network case study

Table 8-14 summarizes the results of optimization of the offsets based on different methods presented in this section. It can clearly be seen from the results that the developed SGA is superior to TRANSYT-7F with a saving of 10% delay, and to the dominance method with a saving of 8% delay.

Table 8-14: Summary of results of 12-intersection grid network case study

Method	Mean of 30 replication AIMSUN delay (veh.s)	Relative changes %
SGA (2.5s)	157394	0.0%
Dominance	170202	8.1%
TRANSYT-7F	172878	9.8%

9 Conclusions and Recommendations

The research embodied in this dissertation reviewed the available on- and offline technologies for optimizing traffic signals and their deficiencies, and developed an effective method for offset optimization for signalized urban road networks that can be applied under all traffic conditions and produces quasi optimal results. The research was incorporated into a C++ program, which consists of three modules: (1) the input module; (2) the optimization module consisting of a genetic algorithm-based optimizer; and (3) a traffic analysis module that serves as the fitness function for the GA-based optimizer.

Essentially, the developed program provides a convenient framework for running the traffic and optimization models embedded in it. This research studied the traffic models and the system model consisting of the traffic and optimization models. The traffic model was studied by making comparisons with well known queuing models and with an unbiased estimator (AIMSUN) at link level, and with only AIMSUN at network level. Finally, the performance of the system model consisting of the GA-based optimizer with the traffic model as its fitness functions was studied by making comparisons against existing widely used software, i.e. TRANSYT-7F version 10, in addition to the dominance method and the full enumeration of the offset variable. These comparisons were conducted for a two-way street consisting of 3 intersections, then for a small network of 6 intersections and finally for a relatively large grid-network of 12 intersections. All the comparisons were conducted by feeding the optimal solution from each method into AIMSUN and then comparing the system delay obtained from AIMSUN.

9.1 Conclusions

A new offset optimization method for urban road networks was envisioned in this dissertation. The various objectives that were satisfied to achieve this goal are described in this section.

Traffic Model

Delay calculation model based on cell transmission model CTM, which serves as fitness function to the GA-based optimizer was developed in the traffic Analysis Module. The reason of using CTM as the basic for the developed traffic model was its ability to consider all traffic conditions and to discreteize the space as well as the time, which are necessary for effective traffic representation and offset optimization. Cyclic link delays obtained in the developed model were validated against the well-known queuing models such as Kimber and Hollis [63], HCM 2000 [80] and Akcelik models[2] using AIMSUN as a reference model. The comparison was conducted for different scenarios of traffic demand in all traffic conditions. The results showed that the developed delay estimation is superior to the three aforementioned models.

For testing delay calculation when varying offsets, a one-way street with two intersections was studied. The offset at the first intersection was fixed to 0 and the offset at the other intersection was varied between 0 and 120 s (cycle time). A high regression coefficient R^2 of 0.98 was obtained when comparing the delay calculated by the developed simulator and the delay obtained from AIMSUN.

The developed delay calculation model comparison was also conducted on a realistic 6-intersection-network, which is part of the List District in Hanover, Germany. The comparison of the flows obtained at all link-exits proved a good agreement ($R^2 = 0.98$) between the developed model and AIMSUN, while the comparison of the delays obtained at all links of the network matched rather well ($R^2 = 0.97$) with AIMSUN results. In this network, the offsets at the intersections were also varied. The results showed similarity between the delays of the simulator and AIMSUN for different offset values.

Optimization Model

Optimization model, which consists of two search algorithms, was developed to generate timing parameters (offsets) for the analysis module and then to optimize these parameters. The search algorithm applied in this research was a GA-based model, as opposed to the traditional hill-climbing methodology employed by TRANSTY-7F. The first algorithm, which was called in this research parallel genetic algorithm PGA, performs a simultaneous search over all offsets by the process of variation of reproduction – crossover – mutation of the entire chromosome. Due to the requirement of online optimization and the large solution space when using PGA, the second algorithm, which was called in this research serial genetic algorithm SGA, aimed to shorten the optimization time by varying a group of offset values and therefore only a part of the chromosome until the best solution was found. In the next step, the offsets of the next group of intersections were optimized. Since the order of treating the intersection and searching the offset has a great influence on the optimization results, a method was developed for the determination of the search order.

Method Comparison

The developed offset optimization method was compared against TRANSYT-7F, the full enumeration of the developed traffic simulator and the dominance method using AIMSUN as a neutral model. TRANSYT-7F was selected for comparison because of the following: (1) It is one of the most widely used signal timing programs; (2) its traffic model recently has been extended to accommodate oversaturated traffic conditions; (3) GA recently has been added to the hill climbing for optimization purposes; and (4) it is said in the manual of the last version (TRANSYT-7F10) that it can be used for online purposes. Dominance method, which is German bandwidth coordination method, was selected for comparison because it is a practical and well-known method in Germany. Full enumeration of the variables was used as a reference for the global optimum. The comparisons were conducted on three case studies to benchmark the performance capabilities of the developed method against existing ones. These case studies were a two-way street with 3 intersections, a small network with 6 intersections and a relatively large grid network with 12 intersections.

In the case study of the two-way street with 3 intersections, the results of comparisons showed that both PGA and SGA could find the optimal solution as well as the full enumeration, whereas the SGA could reduce the CPU-time from 100 s to 20 s. The developed PGA and SGA were superior to the dominance method and comparable to TRANSYT-7F

For the realistic 6-intersection-network, the results showed that there was a potential for improvement of about 17% less delay than the existing coordination plan. With both the PGA and the SGA, a solution close to the absolute optimum could be found. They proved to be superior to both the TRANSYT-7F and the dominance methods (more than 15% less delay).

In the relatively large grid network with 12 intersections, full enumeration was impossible even where a 10s time step was used; the calculation time needed for iterating 11 offset variables was estimated to be 18.69 years. When the duration of the time step was increased, e.g. from 1s to 5s, the calculation time (CPU time) of the simulation of this grid network was exponentially reduced (from 4000ms to 220ms on a pc Athlon 1Ghz). This was due to the reduction in these two dimensions: (1) the time dimension, where the simulation time was reduced from 900 time intervals (interval =1s) to 180 intervals (interval =5s) and (2) the space dimension, where the number of cells was reduced approximately from 920 to 214 cells. Based on the developed PGA and SGA, five scenarios of simulation with the developed simulator were tested as follows:

- Optimization based on PGA using 5s time step when simulating using the developed simulator;
- Optimization based on PGA using 2.5s time step when simulating using the developed simulator;
- Optimization based on SGA including the developed search order determination using 5s time step when simulating using the developed simulator;
- Optimization based on SGA including the developed search order determination using 2.5s time step when simulating using the developed simulator;
- Optimization based on SGA excluding the developed search order determination using 5s time step when simulating using the developed simulator;

The results showed that the quasi absolute optimum could be found after 230s using the SGA with a 5s time step and after 900s using a 2.5s time step, which was not possible using PGA with both 5s and 2.5s time intervals even after 2200s. Therefore, the SGA with the help of the proposed determination is applicable and not only could find the quasi absolute optimum but also could shorten the computation time, allowing for online optimization. However, when the developed search order is not used, the SGA failed to find this quasi absolute optimum. Furthermore, the comparison against TRANSYT-7F showed that the developed SGA including search order determination is superior to TRANSYT-7F (10% relative difference of AIMSUN delay) and to the dominance method (8% relative difference of AIMSUN delay).

In conclusion, the developed method for offset optimization in this research was found to provide superior results to TRANSYT-7F and to the dominance method and comparable results with the full enumeration of control variables. It could be shown clearly that delay savings are obtainable, whereby the calculation speed on normal PCs permits an online use for relatively large networks.

9.2 Recommendations

Several recommendations have emerged from this research as follows:

- The method was converted prototypically into an object oriented C++ program, the use of objects allows easy extensibility or integration with the BALANCE control method. [37]
- Parallel implementation of genetic algorithm optimization should be incorporated into the optimization. This will reduce the computational time for the CTM-based simulator, and could lead to optimize larger network or to extend the developed method to optimize all of the four elements of timing plan. .

References

[1] Abu-Lebdeh, G., R. Benekohal.
Development of a traffic control and queue management procedure for oversaturated arterials
Transportation ResearchBoar d Annual Meeting, Paper #970707, Washington, DC, 1997

[2] Akcelik, R.,
Time-Dependent Expressions for Delay, Stop Rate and Queue Length and Traffic Signals,
Australian Road Research Board, Internal Report, AIR 367-1, 1980.

[3] Barriere, J. F, Farges, J.-L. & J.-J., Henry,
Decentralization vs. Hierarchy in Optimal Traffic Control,
Proceedings of the 5th IFAC/IFIP/IFORS Symposium on Control in Transportation Systems, Vienna, Austria, July, 1986, pp. 209-214.

[4] Bellis, M.
Gas Mask and Traffic Signal,
http://inventors.about.com/library/inventors/blgas_mask.htm, Accessed in September 26, 2005.

[5] Bernhard, J. and T., Riedel,
Erkennung von Stau mit kurzen Schleifendetektoren.
In: Tagungsband der Heureka 1999 Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln.

[6] Brilon, W. and N., Wu,
Delays at Fixed-Time Traffic signals under Time-Dependent Traffic conditions,
In Traffic Engineering & Control, December 1990, pp. 623-631

[7] Brooks, W.D.
Vehicular Traffic Control, Designing Arterial Progressions,
IBM, Watkins Glen, New York, pp. 27.

[8] BUISSON C., LEBACQUE, J.P., LESORT J.B. and H., MONGEOT,
The STRADA model for dynamic assignment,
ITS '96, Orland, FL 1996.

[9] Busch, F.,
MOTION – a new approach to urban network control,
Traffic Technology International, 1996.

[10] Chang, J., Lieberman, E. B. & E. S. Prassa,
Queue estimation algorithm for real-time control policy using detector data,
Paper prepared for presentation at the Transportation Research Board's 79th Annual Meeting in Washington, DC, January 2000 and Publication in the Transportation Research Record Series.

[11] Chaudhary, N. A., Pinnoi, A. & C. J., Messer,
Proposed Enhancements to MAXBAND 86 Program,
Transportation Research Record 1324, Transportation Research Board, National Research Council, Washington, D.C., 1991, pp. 98-104.

[12] Chaudhary, N.A. and C.J. Messer,
PASSER IV: A Program for Optimizing Signal Timing in Grid Networks,
Transportation Research Record 1421, TRB, National Research Council, Washington, D.C., 1993, pp. 82-93.

[13] Chaudhary, N.A. and C.J. Messer,
PASSER IV-96, Version 2.1, User/Reference Manual, Research Report 1477-1, Texas Transportation Institute, Texas A&M University System, College Station, Texas, 1996.

[14] Chong, E.K.P.and S.H., Zak,
An introduction to Optimization,
A Wiley-Interscience Publication, JOHN WILEY & SONS, ING, 2001.

[15] Cohen, S. L.,
Concurrent Use of MAXBAND and TRANSYT Signal Timing Programs for Arterial Signal Optimization,
Transportation Research Record 906, Transportation Research Board, National Research Council, Washington, D.C., pp. 81-84, 1983.

[16] Daganzo, C.
The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory.
Transportation research B 28 (4), 269-287, 1994.

[17] Daganzo, C.
The cell transmission model, Part II: Network traffic.
Transportation Research B 29 (2), 79-93, 1995.

[18] Daganzo, C.
Requiem for second-order fluid approximation of traffic flow,
Transportation Research B 29 (4), 277-286, 1995b.

[19] D’Ans, G. C., D. C. Gazis. 1976.
Optimal control of oversaturated store-and-forward transportation networks
Transportation Science 10, 1–19

[20] Dasgupta, D. and Michalewicz, Z.
Evolutionary Algorithms in Engineering Applications,
Springer-Verlag, New York, 1997.

[21] Davis, L.
The Handbook of Genetic Algorithms,
Von Nostrand Reinhold, New York, 1991.

[22] Dell'Olmo, P; and Mirchandani, P. B.
REALBAND: An approach for real-time co-ordination of traffic flows on networks. Transportation Research Record 1494, National Research Council, Washington, D.C., 1995

[23] Dell'Olmo, P; and Mirchandani, P. B.
A model for real-time traffic co-ordination using simulation based optimization.
Advanced Methods in Transportation Analysis. (ed. L. Bianco and P. Toth) Springer-Verlag, Germany. p.525-546, 1996

[24] Diakaki, C.
Integrated Control of Traffic Flow in Corridor Networks,
PhD Thesis, Technical University of Crete, Department of Production Engineering and Management, Chania, 1999.

[25] Donati, F.; Mauro, V., Roncolini, G. and Vallauri, M..
A Hierarchical Decentralized Traffic Light Control System,
The First Realization: 'Progetto Torino'. In: IFAC 9th World Congress, Vol II, 11G/A-1, 1984.

[26] Eddelbuttel, J., M. Cremer..
A new algorithm for optimal signal control in congested networks
Advanced Transportation 28(3), 1994
275–297.

[27] Farges, J.-L., Khoudour, L. & Lesort, J.-B.
PRODYN: On Site Evaluation,
Proceedings of the 3rd IEE Conference on Road Traffic Control, London,
England, IEE Conference Publication No. 320, pp. 62-66, 1990.

[28] Farges, J.-L., Kamdem, I. & Lesort, J.-B
Realization and Test of a Prototype for Real Time Urban Traffic Control,
Proceedings of the DRIVE Conference on Advanced Telematics in Road Transport, Brussels, February, Vol. 1, pp. 527-542, 1991.

[29] Feldman, O. & Meher, M.
A cell transmission model applied to the optimization of traffic signals.
Universities' Transport Study Group, 34th Annual Conference, V.2, p.17.1-17.14.

[30] FHWA.
Evaluation of UTCS Control Strategies, Executive Summary and Technical Report
Publication No. FHWA-RD-76-149 and -150, U.S. Department of Transportation, Federal Highway Administration, Washington, D.C., July, 1976.

[31] FHWA
Urban Traffic Control System Traffic Adaptive Network Signal Timing Program, Vol. I, II, and III,
Publication No. FHWA-RD-76-125, -126, and -127, U.S. Department of Transportation, Federal Highway Administration, Washington, D.C., August, 1976.

[32] FHWA,
Third Generation Control Software: Urban Traffic Control System (UTCS)
Software Support Project, Vol. I-VI, Publication No. FHWA-RD-76- 154/159, U.S. Department of Transportation, Federal Highway Administration, Washington, D.C., May, 1976.

[33] FHWA.
SIGOP-III User's Manual
Publication No. FHWA-IP-82-019, U.S. Department of Transportation, Federal Highway Administration, Washington,
D.C., July, 1983.

[34] FHWA.
Evaluation of the Optimized Policies for Adaptive Control Strategy,
Publication No. FHWA-RD-89-135, U.S. Department of Transportation, Federal Highway Administration, Washington, D.C., May, 1989.

[35] Forschungsgesellschaft für Straßen- und Verkehrswesen
Handbuch für die Bemessung von Straßenverkehrsanlagen (HBS)
Köln 2001

[36] Foy, M. D., Benekohal, R. F, and Goldberg, D. E.
Signal Timing Determination Using Genetic Algorithms.
In Transportation Research Record 1365, TRB, National Research Council, Washington, D.C., pp. 108-115, 1993

[37] Friedrich, B. and Keller, H..
BALANCE - A Method for Integrated Traffic Adaptive and Vehicle Actuated Signal Control.
In: Proc. of the 7th IFAC Symposium, August 24-26, 1994, Tianjin China.

[38] Friedrich, B. 2000.
Models for Adaptive Urban Traffic Network Control.
In: Proc. of the 8th Meeting of the Euro Working Group Transportation, 11 – 14 September 2000, Rome.

[39] Frontline Systems Inc.
Optimization Problem Types - Overview,
http://www.solver.com/probtype.htm. Accessed Juli 12, 2004.

[40] Henry, J. J., Farges, J. L. and Tufal, J..
The PRODYN real time traffic algorithm.
In: Proc. of the IFAC Symposium, Baden-Baden, 1983.

[41] Hunt, P. B., Robertson, D. I., Bretherton, R. D. and Winton, R. I.
SCOOT - a traffic responsive method of coordinating signals.
TRRL Laboratory Report 1014, 1981.

[42] Garbacz, R. M.
Adaptive Signal Control: What to Expect
ITS Cooperative Deployment Network, 2003

[43] Gartner, N.H.
Constraining Relations among Offsets in Synchronized Signal Networks,
Transportation Science. 6; 88–93, 1972.

[44] Gartner, N.H.
OPAC: A demand responsive strategy for traffic signal control.
Transportation Research Record 906, National Research Council, Washington, D.C., 1983.

[45] Gartner, N. H.
Demand-Responsive Traffic Signal Control Research,
Transportation Research 19A, pp. 369-373, 1985.

[46] GARTNER,N.H.; STAMATIADIS, Ch; and TARNOFF, P.J.
Development of Advanced Traffic Signal Control Strategies for Intelligent Transportation Systems: A Multi-Level Design,
Transportation Research Record 1494: Traffic Operations, TRB, 98-105, 1995

[47] Garber, N.J. and Hoel, L.A.
Traffic and Highway Engineering.
The Wadsworth Group, USA, 2002.

[48] Gerstman, B. Burt
Analysis of Variance (ANOVA)
http://www2.sjsu.edu/faculty/gerstman/StatPrimer/anova-a.pdf, Accessed December, 30, 2005

[49] Goldberg, D.
Algorithms in Search, Optimization and Machine Learning,
Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.

[50] GOTSHALL, S. and RYLANDER, B.
Optimal Population Size and the Genetic Algorithm
School of Engineering, University of Portland, Portland, USA.

[51] Hale, D.K.
Traffic Network Study Tool, TRANSYT-7F, United States version,
Mc Trans Center, University of Florida, January 2005

[52] Head, K.L.
An event-based short-term traffic flow prediction model.
Transportation Research Record 1510, National Research Council, Washington, D.C. p.45-52, 1995.

[53] Heitkötter, J.and David Beasley, D.
The Hitch Hikers Guide to Evolutionary Computation,
http://www.cs.bham.ac.uk/Mirrors/ftp.de.uu.net/EC/clife/www/. Accessed May, 8, 2005.

[54] HELBING D.
Traffic and Related Self-Driven Many-Particle Systems,
Reviews of Modern Physics, in print, 2001.

[55] Henninger, T.
Eine kombinierte Methode zur Schätzung von Kantenbelastungen, Abbiege-quoten und Störungen in Stadtstraßennetzen,
In: Tagungsband der Heureka, Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln, 1999.

[56] Henry, J.-J., Farges, J.-L. & Tuffal, J.
The PRODYN Real Time Traffic Algorithm,
Proceedings of the 4th IFAC/IFIP/IFORS Symposium on Control in Transportation Systems, Baden-Baden, Germany, April, 1983, pp. 305-310.

[57] Henry, J.-J. & Farges, J.-L.
PRODYN,
Proceedings of the 6th IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation, Paris, France, September, 1989, pp. 253-255.

[58] HOLROYD, J.; HILLIER, J.A.
The Glasgow Experiment: PLIDENT and After
RRL Report 384, 1971

[59] Homburger, W.H.; Hall, J.W.; Loutzenheiser, R.C.; Reilly, W.R.
Fundamentals of traffic engineering,
Institute of Transportation studies, University of Callifornia, Berleley, 1996.

[60] Hounsell, N. B. and McDonald, M.
Urban network traffic control,
Proc Instn Mech Engrs Vol 215 Part I, p 325-334, 2001.

[61] Kell, J.H. and Fullerton, I.J.
Manual of traffic signal design,
Institute of Transportation Engineers, Washington, 1991.

[62] Hunter-Zaworski, K.
Signal Timing Design,
http://www.webs1.uidaho.edu/niatt_labmanual/Chapters/SignalTimingDesign/Introduction/index.htm, Accessed May, 8, 2005.

[63] Kimber, R. M. and Hollis, E. M.:
Traffic queues and delays at road junctions.
TRRL Laboratory Report 909, 1979.

[64] Khoudour, L., Lesort, J.-B. & Farges, J.-L.
PRODYN: Three Years of Trials in the ZELT Experimental Zone,
Recherche Transports Sécurité (English Issue), N 6, pp. 89-98, (1991).

[65] KÜHNE R.
Verkehrsflußmodelle. Zur Theorie des Straßenverkehrs.
Forschungsgesellschaft für Straßen- und Verkehrswesen, Karlsruhe, 1993, 44-61.

[66] Law, Averill M. and Kelton W. David,
Simulation Modeling and Analysis.
McGraw-Hill International Editions. Second Edition. 1991.

[67] LIGHTHILL, M.J. and WHITHAM, J.B.
On kinematic waves. I. Flow movement in long rivers. II. A theory of traffic flow on long crowded road.
Proceedings of Royal Society,A229, 281-345, 1955.

[68] LO, H.K.
A novel traffic signal control formulation,
Transportation Research A 33, 433-448, 1999.

[69] Lowrie, P. R.
The Sydney Co-ordinated Adaptive Traffic System: Principles, Methodology, Algorithms, Proceedings of the International Conference on Road Traffic Signalling, London, England, March-April, 1982, IEE Conference Publication No. 207, pp. 67-70.

[70] Luk, J. Y. K., Sims, A. G. & Lowrie, P. R.
SCATS: Application and Field Comparison with a TRANSYT Optimised Fixed Time System, Proceedings of the International Conference on Road Traffic Signalling, London, England, March-April, 1982, IEE Conference Publication No. 207, pp. 71-74.

[71] MacGowan, J. & Fullerton, I. J.
Development and Testing of Advanced Control Strategies,
In the Urban Traffic Control System (Three Articles), Public Roads 43(2, 3 & 4), 1979-1980.

[72] Martin, P.; Perrin, J.; Chilukuri, B.; Jhaveri, C., and Feng, Y.
Adaptive Signal Control II,
Department of Civil and Environmental Engineering University of Utah Traffic Lab, January 2003.

[73] McNeil, D. R.
A Solution to the Fixed-Cycle Traffic,
Light Problem for Compound Poisson Arrivals. J. Appl. Prob. 5, pp. 624-635, 1968.

[74] Messer, C.J., H.E. Haenel, and E.A. Koeppe.
A Report on the User’s Manual for Progression Analysis and Signal System Evaluation Routine – PASSER II,
TTI Research Report 165-14, College Station, Texas, 1974.

[75] Messer, C.J., R.H. Whitson, C.L. Dudek, and E.J. Romano.
A Variable-Sequence Multiphase Progression Optimization Program,
In Transportation Research Record 445, TRB, National Research Council, Washington, D.C., 1973, pp. 24-33.

[76] Messer, C.J., D.B. Fambro, and S.H. Richards.
Optimization of Pretimed Signalized Diamond Interchanges,
In Transportation Research Record 644, TRB, National Research Council, Washington, D.C., 1977, pp. 78-84.

[77] Miller, A. J.
Australian Road Capacity Guide - Provisional Introduction and Signalized Intersections, Australian Road Research Board Bulletin No.4, (Superseded by ARRB report ARR No. 123, 1981).

[78] Mongoeot, H.
Modelling of traffic flow dynamics in incident conditions using the first-order macroscopic approach,
Traffic Engineering and Control 38 (11), 584-592, 1997.

[79] Mück, J..
Estimation Methods for the State of Traffic at Traffic Signals Using Detectors Near the Stop-line,
In Traffic Engineering & Control, December 2002.

[80] National Research Council
Highway Capacity Manual,
Transportation Research Board, Washington, D.C, 2000.

[81] Newell, G.F.
A simplified theory of kinematic waves,
Research Report, University of California, Berkeley, UCB-ITS-RR-91-12, 1991.

[82] Obitko, M.
Genetic Algorithms,
http://cs.felk.cvut.cz/~xobitko/ga/main.html, accessed December 25, 2003.

[83] Pearson, R.
Traffic Signal Control,
http://www.calccit.org/itsdecision/serv_and_tech/Traffic_signal_control/trafficsig_summary.html, accessed December 22, 2003

[84] Pline, J.L.
Traffic Engineering Handbook,
Institute of Transportation Engineers, Washington, 1992.

[85] Payne, H.J.
Modelling of freeway traffic and control,
Mathematics of Public Systems, Simulation Council 1 (1), 51-61, 1971.

[86] RICHARDS, P.I.
Shockwaves on the highway,
Operations Research B 22, 81-101, 1956.

[87] Robertson, D.I.
Transyt: a traffic network study tool,
RRL Report LR 253, Road Resarch Laboratory Crowthorne, Berkshire 1969.

[88] Robertson, D. I
Research on the TRANSYT and SCOOT Methods of Signal Coordination,
ITE Journal 56(1), January, 1986, pp. 36-40.

[89] Robertson, G. D.
Handling congestion with SCOOT,
In: Traffic Engineering & Control, April, pp.228-230, 1987.

[90] Schnabel, W.
Verkehrstechnische Berechnung von lichtsignalgesteuerten Straßennetzen,
Forschungsbericht, Zentrales Forschungsinstitut des Verkehrswesen der DDR, Berlin, 1981.

[91] Sen, S. & Head, K. L.
Controlled Optimization of Phases at an Intersection,
Transportation Science 31, pp. 5-17., 1997

[92] Sipper, M.
A Brief Introduction to Genetic Programming,
http://www.cs.bgu.ac.il/~sipper/ga.html http, accessed June 10, 2005

[93] Stamatiadis, C. & Gartner, N. H.
MULTIBAND-95: A Program for Variable Bandwidth Progression Optimization of Multiarterial Traffic Networks,
Paper Presented at the Transportation Research Board 75th Annual Meeting, Washington, D.C., January, 1996.

[94] Steering Committee for Traffic Control and Traffic Safety
Guidelines for Traffic Signals (RiLSA),
Road and Transportation Research Association, Köln, 2003.

[95] Sytsma, S.
Performing Scheffè Post Hoc Comparisons
http://www.sytsma.com/phad530/scheffeposthoc.html, accessed December 30, 2005

[96] The Institution of Highways & Transportation
Transport in the Urban Environment.
The Institution of Highways & Transportation, London, 1997.

[97] Tomicic, B.
Lecture Note on Optimization Techniques,
International Institute for Infrastructural, Hydraulic and Environmental Engineering, Delft, The Netherlands, 1999.

[98] Tsay, H.-S. & Lin, L.-T.
New Algorithm for Solving the Maximum Progression Bandwidth,
Transportation Research Record 1194, Transportation Research Board, National Research Council, Washington, D.C., pp. 15-30, 1988.

[99] TSS-Transport Simulation Systems
AIMSUN version 4.2 user manual.
TSS-Transport Simulation Systems

[100] Venglar, S, P. Koonce, and T. Urbanik II.
PASSER III-98 Application and User's Guide,
Texas Transportation Institute, Texas A&M University System, College Station, Texas, 1998.

[101] Vose, M. D.
The Simple Genetic Algorithm,
A Bradford Book, Massachusetts, London, England, 1999.

[102] Wall, M.
GAlib: A C++ Library of Genetic Algorithm Components, Version 2.4,
Documentation Revision B, August 1996. http://lancet.mit.edu/ga/. Accessed, December, 25, 2003.

[103] Wall, M.
Introduction to Genetic Algorithms,
http://lancet.mit.edu/~mbwall/presentations/IntroToGAs/. Accessed December 25, 2003.

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

[105] Wey, W., R. Jayakrishnan..
A network traffic signal optimization formulation with embedded platoon dispersion simulation
Transportation Research Board Annual Meeting Paper #971337, Washington, DC., 1997

List of Figures

Figure 1-1: Fundamental diagram 2
Figure 1-2: Methodology for achieving the research objectives 4
Figure 2-1: Typical lane group for analysis (source [80]) 9
Figure 2-2: Critical lane group determination 10
Figure 2-3: Two-phase Signal system at one-way intersection 11
Figure 2-4: An example of two-phase control 12
Figure 2-5: An example of four-phase control 13
Figure 2-6: Cycle length versus delay (source: [62]) 14
Figure 2-7: Time-distance diagram (source: [84]) 17
Figure 3-1: Flowchart of simple genetic algorithm 35
Figure 3-2: Encoding of a chromosome by binary string 36
Figure 3-3: Single-point crossover 37
Figure 3-4: Two point crossover 37
Figure 3-5: An example of mutation 38
Figure 3-6: A roulette wheel (Source: [82]) 39
Figure 3-7: Situation before and after ranking (Source: [82]) 40
Figure 4-1: An outline of the offset optimization method 44
Figure 4-2: The trapezoidal flow-density relationship used in the analysis module 45
Figure 4-3: A cell representation of a simple intersection 45
Figure 4-4: Cell and section representation 48
Figure 4-5: Description of control data 49
Figure 5-1: Fundamental diagram and the Flow Density Relationship for the CTM 52
Figure 5-2: Modeling of normal, diverge, and merge links 53
Figure 5-3: The delay-speed relationship 57
Figure 5-4: A simple example of one intersection without turn lanes 58
Figure 5-5: Cell representation of an intersection without turn lanes 58

Figure 5-6: A simple example of one intersection with turn lanes....................................59
Figure 5-7: Cell representation of an intersection with turn lanes....................................60
Figure 5-8: Pseudo-code for the calculation steps of flows and number of vehicles 61
Figure 6-1: A four-way intersection located in List District of Hanover City 64
Figure 6-2: An example (scenario 4) of the comparison of the total vehicular cyclic delay for the different queuing models..66
Figure 6-3: A one-way street with 2 signalized intersection...67
Figure 6-4: Delay comparison between the developed model and AIMSUN at one way street with 2 intersections 68
Figure 6-5: Site Location of a part of List District in Hanover city in Germany....................69
Figure 6-6: Cell representation of the 6-intersection network 70
Figure 6-7: Comparison of flow rate of all links of the realistic network between the developed model and AIMSUN...72
Figure 6-8: Comparison of delay all links of the network between the developed model and AIMSUN 72
Figure 6-9: Comparison of delay estimation at a small realistic network between the developed model and AIMSUN for 5 trials of offset variations.........................73
Figure 7-1: A flowchart of the PGA 76
Figure 7-2: A flowchart of the SGA 79
Figure 7-3: A network with 5 intersections for illustrating the search order method 80
Figure 7-4: A time-space diagram for illustrating the relationship between the offsets and the relative offsets...84
Figure 7-5: A network with 5 intersections with different turning proportions......................85
Figure 7-6: A grid network for the illustration of search order 87
Figure 8-1: A Flowchart of full enumeration (nested-loop algorithm)..................................90
Figure 8-2: A two-way arterial with 3 signalized intersection (AIMSUN Representation) 95
Figure 8-3: Cell representation of the arterial case study 96
Figure 8-4: Delays at varied offsets of intersection k3 given that offset at k2 is 0 97
Figure 8-5: Histogram of the system delay obtained from AIMSUN using the optimal offsets based on Full Enumeration Method for the arterial case study........................98
Figure 8-6: Delay comparison of coordination methods for the arterial case study 99
Figure 8-7: Case study of the realistic network which is a part of List District in Hanover.101
Figure 8-8: Part of full enumeration results for realistic network case study (part 1).........103
Figure 8-9: Description of the coordination direction for the dominance method..............104
Figure 8-10: Delay comparison of coordination methods for small network case study 107
Figure 8-11: layout of the 12-intersection grid network (AIMSUN Representation)............108

Figure 8-12: Delay versus CPU run time of a pc Athlon 1 GHz using different methods for the grid network case study..112
Figure 8-13: Delay versus CPU run time of a PC Athlon 1 GHz using 1 and 2.5s step times for the grid network case study 113
Figure 8-14: Delay comparison of coordination methods for grid network case study.........115

List of Tables

Table 2-1: Comparison of UTCS Control Strategies (Source: [46]) 26
Table 6-1: Scenarios and corresponding degrees of saturation x for queue comparison..64
Table 6-2: Comparison between the estimated delay to the real one................................65
Table 6-3: Link inflows and traffic splits 71
Table 6-4: Comparison of delay estimation at a small realistic network between the developed model and AIMSUN 73
Table 7-1: calculations of search order procedure for the network with 5 intersection 82
Table 7-2: calculations of search order procedure for the network that has 5 intersections with different turning proportions 86
Table 7-3: A summary of calculations of search order procedure for the grid network......88
Table 7-4: The 11 variables of relative offsets for the grid network 88
Table 8-1: ANOVA test results for arterial case study...99
Table 8-2: Scheffè Post Hoc Comparison results for arterial case study...........................99
Table 8-3: Optimal offsets of all coordination methods for arterial case study.................100
Table 8-4: Range of variation of the second full enumeration for realistic network case study...102
Table 8-5: Range of variation of the third full enumeration for realistic network case study 103
Table 8-6: Optimal offsets results based on SGA methods for the realistic network 105
Table 8-7: ANOVA test results for small network case study...106
Table 8-8: Scheffè Post Hoc Comparison results for small network case study..............106
Table 8-9: Optimal offsets results based on different methods for the realistic network..107
Table 8-10: Optimal offsets results based on SGA methods for the 12 intersection case study...113
Table 8-11: Optimal offsets results based on TRANSYT-7F methods for the 12 intersection case study 114

Table 8-12: ANOVA test results for grid network case study 115
Table 8-13: Scheffè Post Hoc Comparison results for grid network case study 115
Table 8-14: Summary of results of 12-intersection grid network case study 116

List of Abbreviation

1-GC	First Generation Control
2-GC	Second Generation Control
3-GC	Third Generation Control
AIMSUN 2	The Integrated Traffic Environment, version 2 (Translated from Spanish)
CERT	Centre d'Etudes et de Recherches de Toulouse
CFP	Cyclic Flow Profile
CIC	Critical Intersection Control
COP	Controlled Optimization of Phases
CPU	Central Processing Unit
CYRANO	Cycle-Free Responsive Algorithm for Network Optimization
CTM	Cell Transmission Model
DI	Disutility Index
DOW	Day-of-Week
DRIVE	Dedicated Road Infrastructure for Vehicle Safety in Europe
EP	evolutionary programming
ES	evolutionary strategies
EXC LT	Exclusive left turn movement
FHWA	Federal Highway Administration
GA	Genetic Algorithm

GAs	Genetic Algorithms
GO	Global Optimization
GP	genetic programming
GRG	Generalized Reduced Gradient
HCM	Highway Capacity Manual
KH	Kimber and Hollis method
LP	linear Programming
LT	Left turn movement
LWR	The Lighthill and Whitham model (1955) and Richards (1956)
MAXBAND	Maximal Bandwidth Traffic Signal Setting Optimization Program
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
MULTIBAND	An Extension of MAXBAND
NLP	smooth non-linear programming
NSP	non-smooth problem
OD	Origin Destination
OPAC	Optimized Policies for Adaptive Control
PGA	Parallel Genetic Algorithm
PARAMICS	Parallel Microscopic Simulation
PASSER	Progression Analysis and Signal System Evaluation Routine
PI	Performance Index
PRODYN	A French Real-time Traffic Algorithm
QMC	Queue Management Control
QP	quadratic Programming
RHODES	Real-time, Hierarchical, Optimized, Distributed and Effective System
RT	Right Turn movement
SCATS	Sydney Coordinated Adaptive Traffic System

SCOOT	Split, Cycle and Offset Optimization Technique
SQP	Sequential Quadratic Programming
SGA	Serial/Sequential Genetic Algorithm
SIGOP	Traffic Signal Optimization Model
SOAP	Signal Operations Analysis Package
TANSTP	Traffic Adaptive Network Signal Timing Program
TH	Through Turn movement
TOD	Time-of-Day
TRANSYT-7F	Traffic Network Study Tool, United States Version
TRRL	Transport and Road Research Laboratory
TRSP	Traffic Responsive
UTCS	Urban Traffic Control System
UTOPIA	Urban Traffic Optimization by Integrated Automation
VISSIM	Traffic in Towns - Simulation (Translated from German)Urban Traffic Control System (UTCS)
veh/h	vehicle per hour

Annex A

Examples of c++ code for developed program

In this Annex, part of the c++ code of the PGA optimizer is presented. The major elements of this part are:

- Binary phenotype creation
- Objective function creation using the created phenotype
- Genome creation using the created phenotype and the objective function.
- GA creation using the created genome and running it

```
// Create a phenotype
  GABin2DecPhenotype map_Off_only_PGA;
  int Nbits_offset_only_PGA=num_bits(f12.CT);
  float max_val_offset_only_PGA=Max_value_n_bits(Nbits_offset_only_PGA);

  for(int ii=0; ii<f12.Nu_Intersec-1; ii++)
  {
        map_Off_only_PGA.add(Nbits_offset_only_PGA, 0, max_val_offset_only_PGA);
  }

// Create the template genome using the phenotype map we just made.
  GABin2DecGenome                              genome_Off_only_PGA(map_Off_only_PGA,
objective_Off_only_PGA);
```

Figure A1: an example of c++ code

```
 //  create the GA using the genome and run it.
 // scaling so that we can handle negative objective scores.

          GASimpleGA ga(genome_Off_only_PGA);
          GAPowerLawScaling scaling;
          ga.scaling(scaling);
          ga.populationSize(popsize);
          ga.nGenerations(ngen);
          ga.pMutation(pmut);
          ga.pCrossover(pcross);
          ga.scoreFilename("bog.dat");
          ga.scoreFrequency(1);
          ga.flushFrequency(1);
          ga.selectScores(GAStatistics::AllScores);
          unsigned int CountGen=0;
          ga.initialize();
          outfile.open("All_indivsual__test_1.txt", (STD_IOS_OUT | STD_IOS_TRUNC));
          while(!ga.done())
          {
                              for(int ii=0; ii<ga.population().size(); ii++)
                              {
                                        genome_Off_only_PGA = ga.population().individual(ii);
                                        outfile << CountGen;
                                        for( int kk=0; kk<f12.Nu_Intersec-1; kk++)
                                        {
                                                  outfile                    <<                   "\t"
<<genome_Off_only_PGA.phenotype(kk);
                                        }
                                        outfile << "\t" <<-genome_Off_only_PGA.score() << "\n";
                              }
                    }
                    else
                    {
                              for( ii=0; ii<5; ii++)
                              {
                                        genome_Off_only_PGA = ga.population().individual(ii);
                                        outfile << CountGen;
                                        for( int kk=0; kk<f12.Nu_Intersec-1; kk++)
                                        {
                                                  outfile                    <<                   "\t"
<<genome_Off_only_PGA.phenotype(kk);
                                        }
                                        outfile << "\t" <<-genome_Off_only_PGA.score() << "\n";
                              }
                    }
                    ga.step();
                    CountGen++;
          }
 outfile.close();
```

Figure A1: **Continued**

```
//Obkective function

Float objective_Off_only_PGA(GAGenome & c)
{
	GABin2DecGenome & genome_Off_only_PGA = (GABin2DecGenome &)c;


	float Result;

	f12.Initialization();


	for(int ii=0; ii<f12.Nu_Intersec-1; ii++)
	{
		f12.Offset[ii+2]=genome_Off_only_PGA.phenotype(ii);
	}

	f12.Base_CotrolCode();

	f12.Offset_CotrolCode();

	Result= -f12.DelayCalc_Improv1();

	return Result;
}
```

Figure A1: **Continued**

Annex B

Detailed Description of the Arterial Case Study

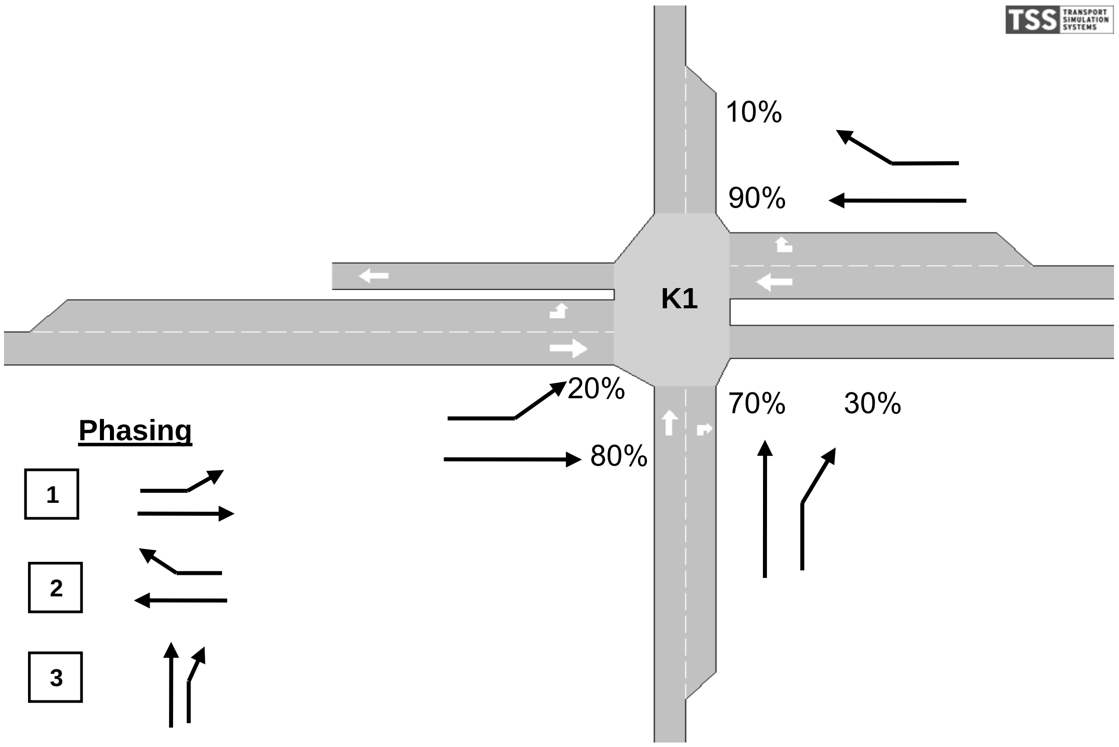


Figure B1: A detailed layout of intersection k1 (AIMSUN Representation)

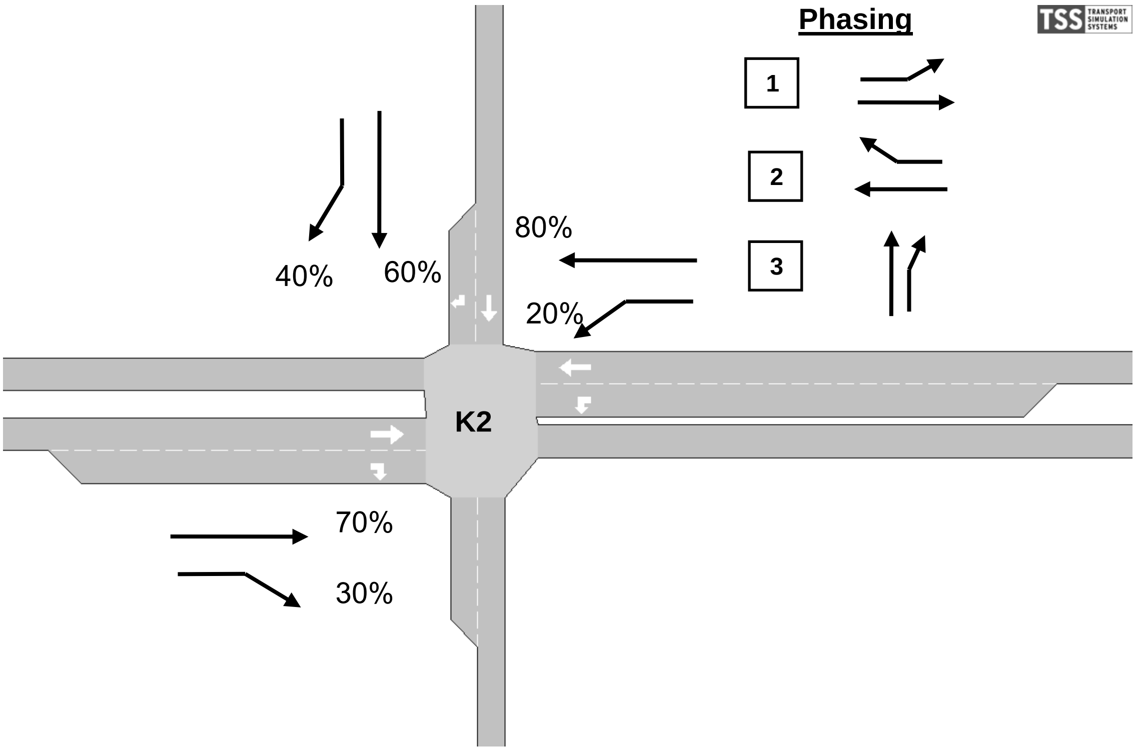


Figure B2: A detailed layout of intersection k2 (AIMSUN Representation)

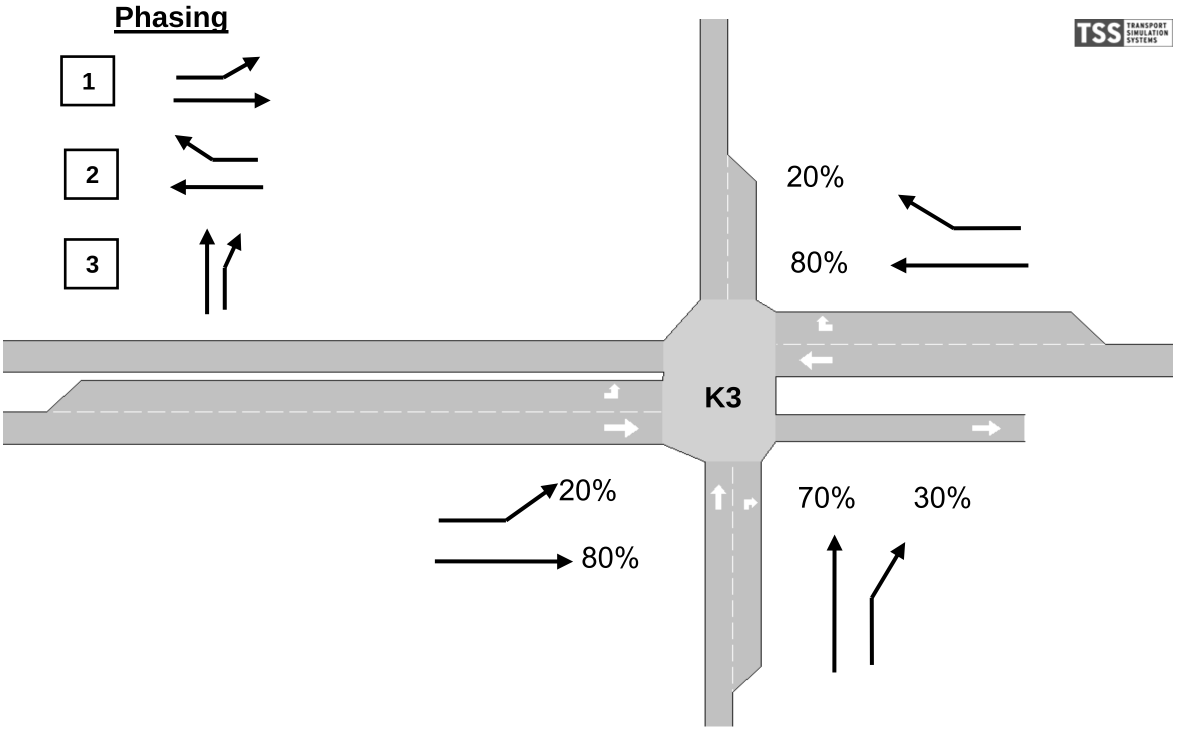
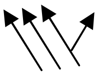
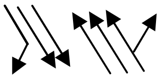
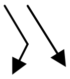
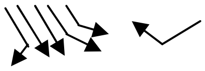
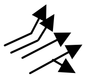
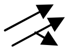
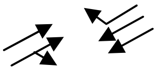
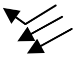
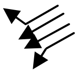
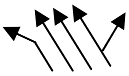


Figure B3: A detailed layout of intersection k3 (AIMSUN Representation)

Detailed Description of the Realistic Network

Table B1:: Existing signal plan for intersection 1

Phase	Lane Groups	Time Allocation (seconds)
1		9
2		7
3		5
4		10
5	All Red	7
6		10
7		11
8		5
9		4
10		10
11	All Red	5
12		7

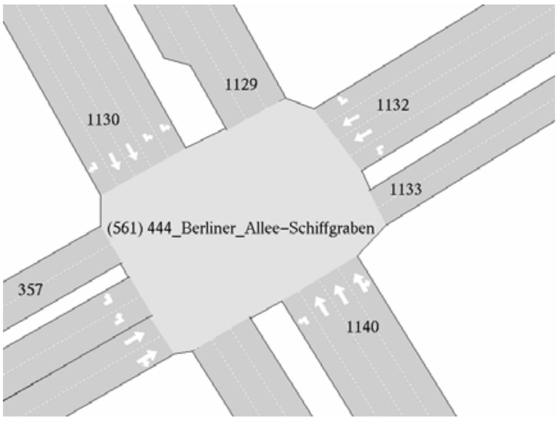
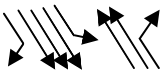
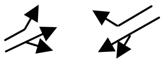
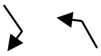


Figure B4: Detail 1: Layout of Berliner Allee-Schiffgraben intersection

Table B2: Existing signal plan for intersection 2

Phase	Lane Groups	Time Allocation (seconds)
1		28
2	All Red	7
3		33
4	All Red	6
5		9
6	All Red	7

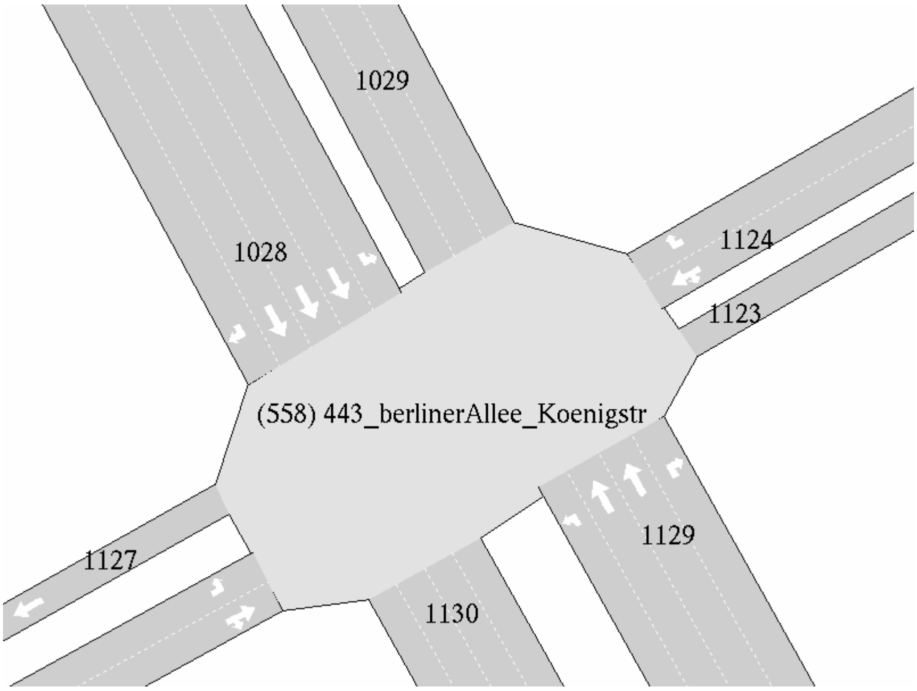
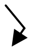
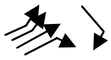
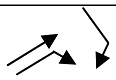
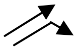
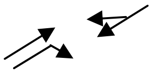
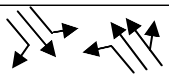


Figure B5: **Detail 2: Layout of Berliner Allee-Koenigstr intersection**

Table B3: **Existing signal plan for intersection 3**

Phase	Lane Groups	Time Allocation (seconds)
1	All Red	9
2		7
3		9
4		6
5		4
6		29
7	All Red	9
8		17

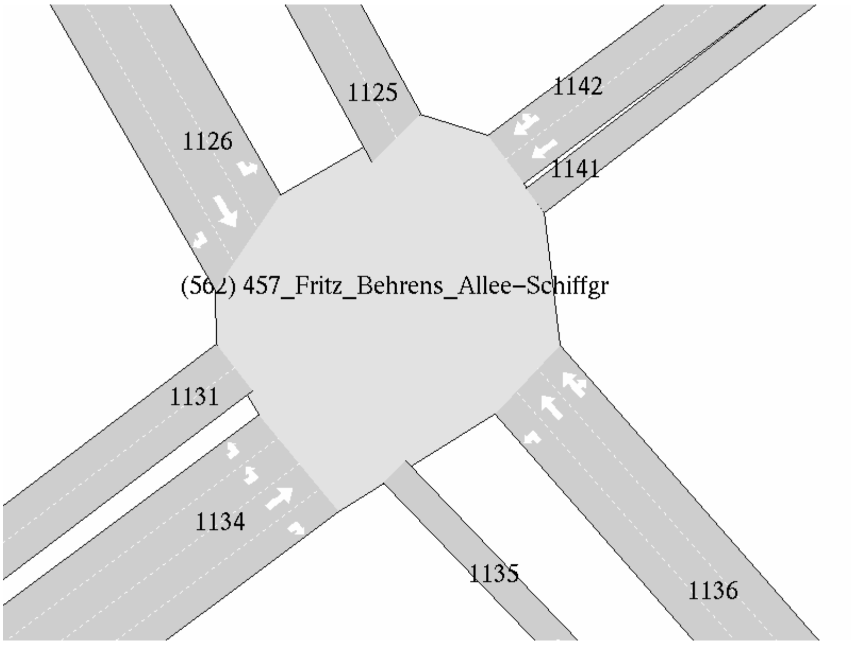


Figure B6: **Detail 3: Layout of Berliner Fritz-Behres-Allee-Schiffgr intersection**

Table B4: **Existing signal plan for intersection 4**

Phase	Lane Groups	Time Allocation (seconds)
1	↖↖	25
2	↖↖ ↙↙	29
3	↖↖ ↙	3
4	↙	4
5	↗↘ ↙	18
6	All Red	11

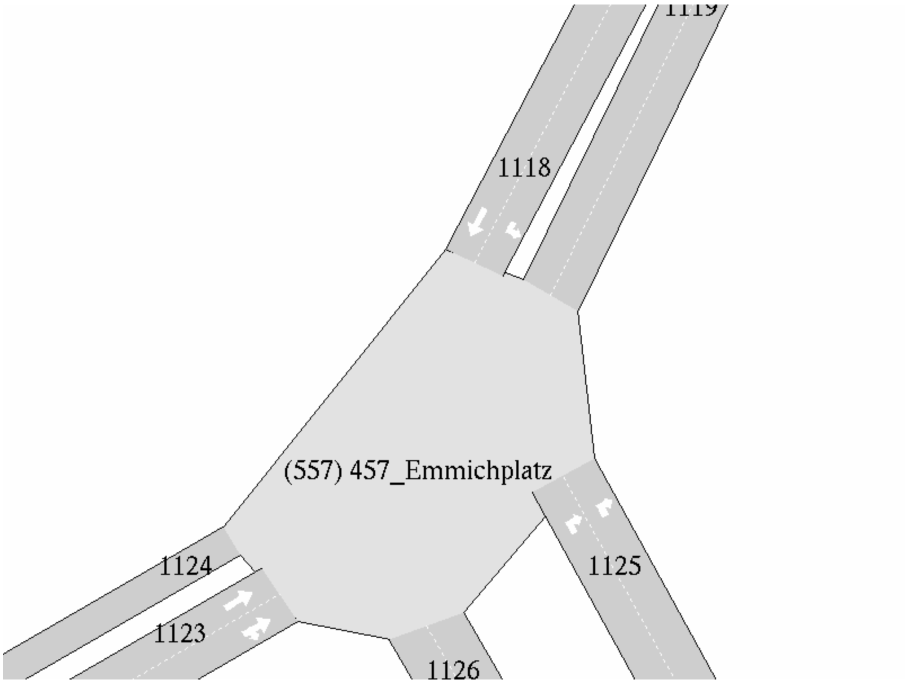
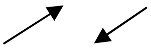
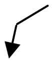
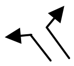
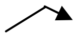
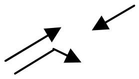


Figure B7: Detail 4: Layout of Emmichplatz intersection

Table B5: Existing signal plan for intersection 5

Phase	Lane Groups	Time Allocation (seconds)
1		43
2	All Red	5
3		5
4	All Red	5
5		18
6		4
7		10

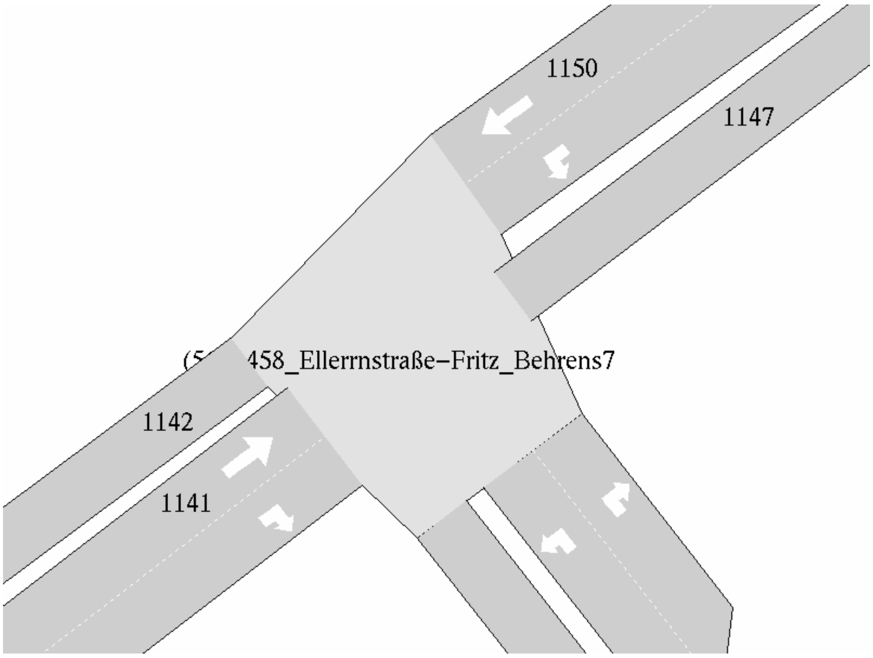
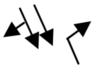
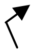
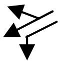
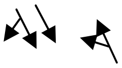


Figure B8: Detail 5: Layout of Ellerrnstrasse-Fritz_Behrens intersection

Table B6: Existing signal plan for intersection 6

Phase	Lane Groups	Time Allocation (seconds)
1		16
2		4
3	All Red	4
4		30
5	All Red	8
6		28

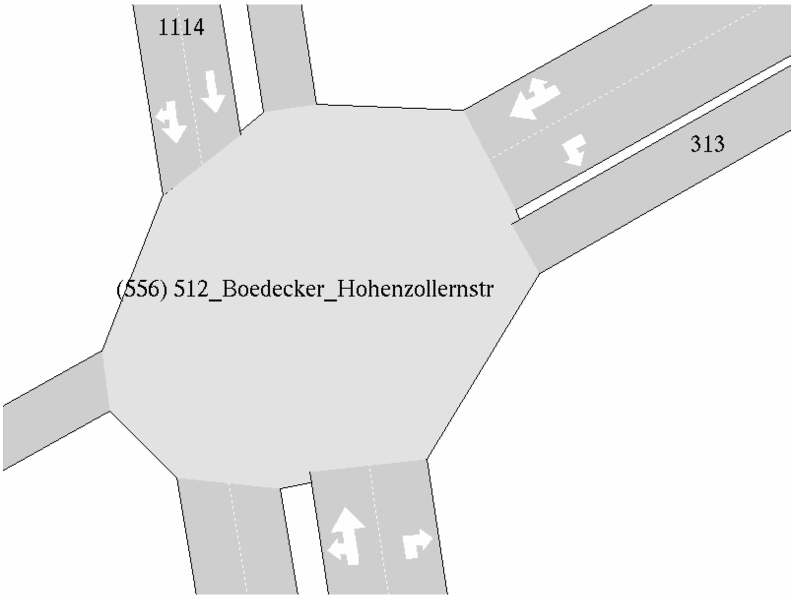


Figure B9: Detail 6: Layout of Boedecker_Hohenzollernstr. intersection

Detailed Description of the intersections of the grid network

Phasing

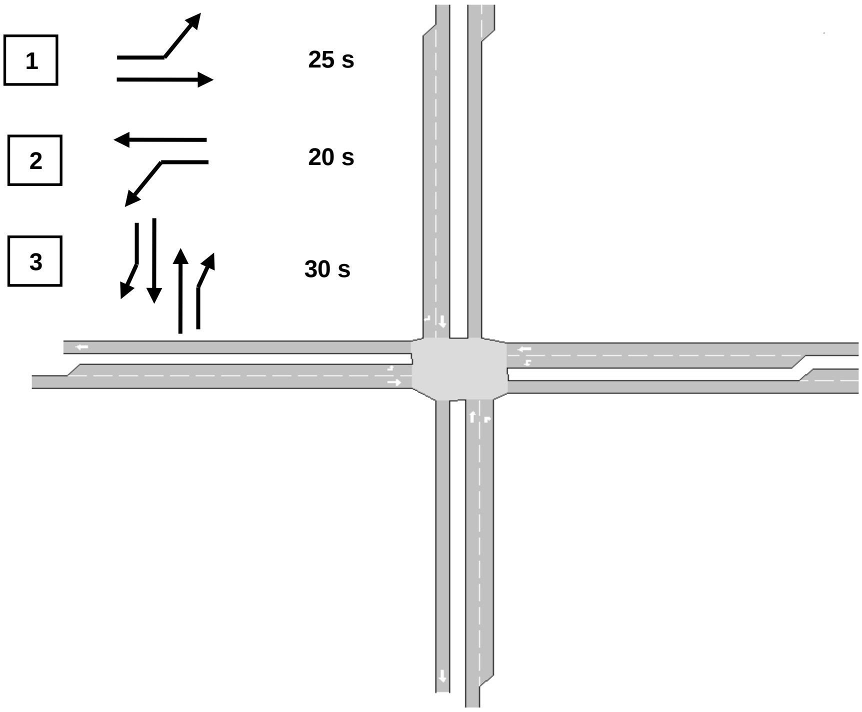


Figure B10: Detailed layout of intersections K1, K2, K3, K4, K5, K6, K10, K11, K12 (AIMSUN Representation)

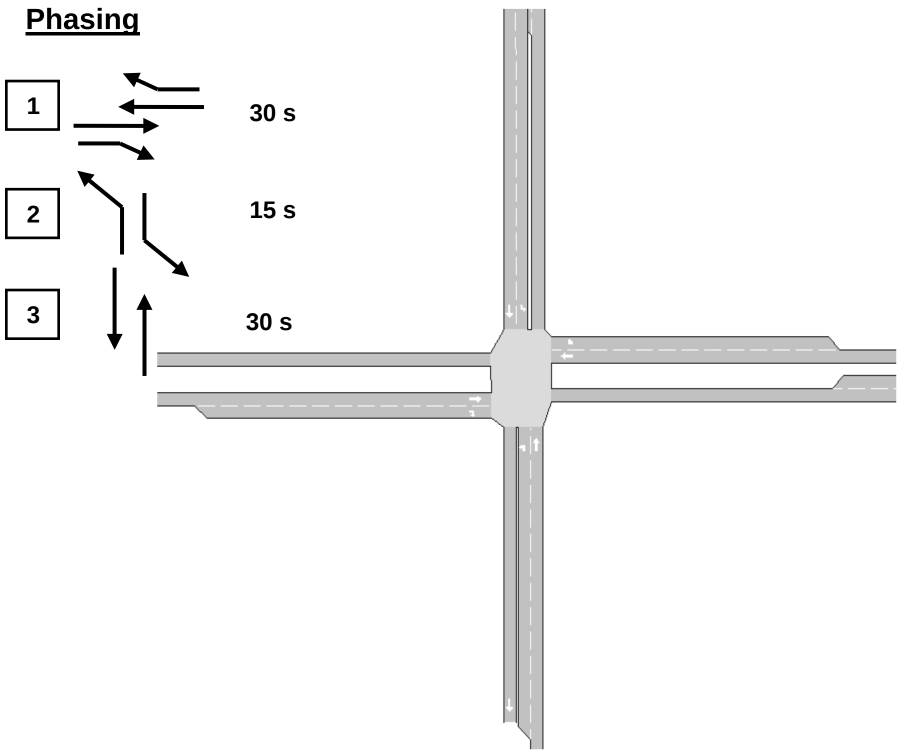


Figure B11: Detailed layout of intersections K7, K8, and K9 (AIMSUN Representation)

Annex C

TRANSYT-7F Output for the Arterial Case Study

```
TRANSYT-7F Release 10.2 -- Genetic Algorithm Optimization

File C:\Programme\Transyt-7f\T3int_1coma3fact_dem.tin
Simulation Engine: TRANSYT-7F

Genetic Algorithm Input Parameters
----------------------------------
Crossover Probability (%):                 70
Mutation Probability (%):                  5
Convergence Threshold (%):                 0,01
Maximum Number of Allowable Generations:   20
Actual Number of Optimization Generations: 20
Population Size:                           10
Random Number Seed:                        7781
Elitist Method:                            True
Objective Function:                        Throughput/DI

Optimization Results  (Offsets only)
--------------------

Node  Cycle      Offset   NS Phasing EW Phasing
    Init Final Init Final Init Final Init Final
   1  70   70     0    0   ---   ---  ---   ---
   2  70   70     0    9   ---   ---  ---   ---
   3  70   70     0   44   ---   ---  ---   ---

              Control    Total     Fuel     Travel  Performance
               Delay     Stops  Consumption  Time      Index
             (sec/veh)    (%)    (lit/hr) (veh-hr/hr)

Initial        34.7        94       224        48      53.29
Final          29.1        84       201        43      63.05

The final solution originated in generation #11
To obtain detailed output, process the punch file:
T3int_1coma3fact_dem.pun
```

TRANSYT-7F Output for the Realistic network Case Study

```
TRANSYT-7F Release 10.2 -- Genetic Algorithm Optimization

File C:\My PhD work\Essam\My work\T7F10\6int_realistic.tin
Simulation Engine: TRANSYT-7F

Genetic Algorithm Input Parameters
----------------------------------
Crossover Probability (%):                30
Mutation Probability (%):                 1
Convergence Threshold (%):                0,01
Maximum Number of Allowable Generations:  50
Actual Number of Optimization Generations: 50
Population Size:                          50
Random Number Seed:                       7781
Elitist Method:                           True
Objective Function:                       Throughput/DI

Optimization Results  (Offsets only)
--------------------

Node  Cycle      Offset   NS Phasing EW Phasing
    Init Final Init Final Init Final Init Final
   1  90   90     0    0   ---   ---  ---   ---
   2  90   90     0   61   ---   ---  ---   ---
   3  90   90     0   11   ---   ---  ---   ---
   4  90   90     5   36   ---   ---  ---   ---
   5  90   90     5   46   ---   ---  ---   ---
   6  90   90     5   89   ---   ---  ---   ---

              Control    Total     Fuel     Travel  Performance
               Delay     Stops  Consumption  Time      Index
             (sec/veh)    (%)    (lit/hr) (veh-hr/hr)

Initial        58.0        88       550       135      47.87
Final          50.2        91       515       122      55.08

The final solution originated in generation #27
To obtain detailed output, process the punch file:
6int_realistic.pun
```

TRANSYT-7F Output for Grid Network Case study Case Study

```
TRANSYT-7F Release 10.2 -- Genetic Algorithm Optimization

File C:\My PhD work\Essam\My work\T7F10\12int_other_phasing.tin
Simulation Engine: TRANSYT-7F

Genetic Algorithm Input Parameters
----------------------------------
Crossover Probability (%):                30
Mutation Probability (%):                 1
Convergence Threshold (%):                0,01
Maximum Number of Allowable Generations:  30
Actual Number of Optimization Generations: 30
Population Size:                          50
Random Number Seed:                       7781
Elitist Method:                           True
Objective Function:                       Disutility Index
```

```
Optimization Results  (Offsets only)
--------------------

Node  Cycle      Offset   NS Phasing EW Phasing
    Init Final Init Final Init Final Init Final
   1  90   90     0    0   ---   ---  ---   ---
   2  90   90     0    4   ---   ---  ---   ---
   3  90   90     0   14   ---   ---  ---   ---
   4  90   90     0   46   ---   ---  ---   ---
   5  90   90     0   47   ---   ---  ---   ---
   6  90   90     0   57   ---   ---  ---   ---
   7  90   90     0   10   ---   ---  ---   ---
   8  90   90     0   62   ---   ---  ---   ---
   9  90   90     0   70   ---   ---  ---   ---
  10  90   90     0   44   ---   ---  ---   ---
  11  90   90     0   11   ---   ---  ---   ---
  12  90   90     0   14   ---   ---  ---   ---

              Control    Total     Fuel     Travel  Performance
               Delay     Stops  Consumption  Time      Index
             (sec/veh)    (%)    (lit/hr) (veh-hr/hr)

Initial        41.3        88      1867       430      335.7
Final          36.6        92      1768       386      292.1

The final solution originated in generation #30
To obtain detailed output, process the punch file:
12int_other_phasing.pun
```

Annex D

Histograms for the arterial case study

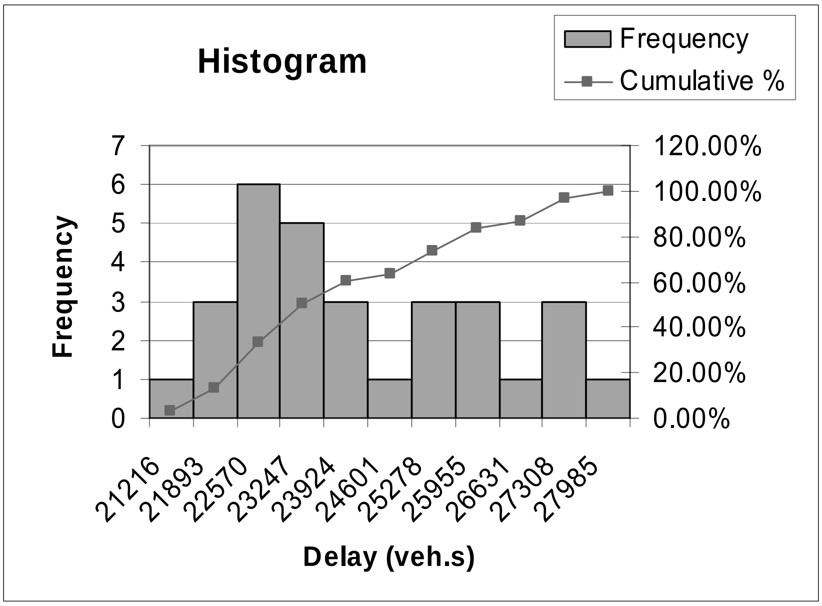


Figure D1: Histogram of the system delay obtained from AIMSUN using the optimal offsets based on the dominance method for the arterial case study

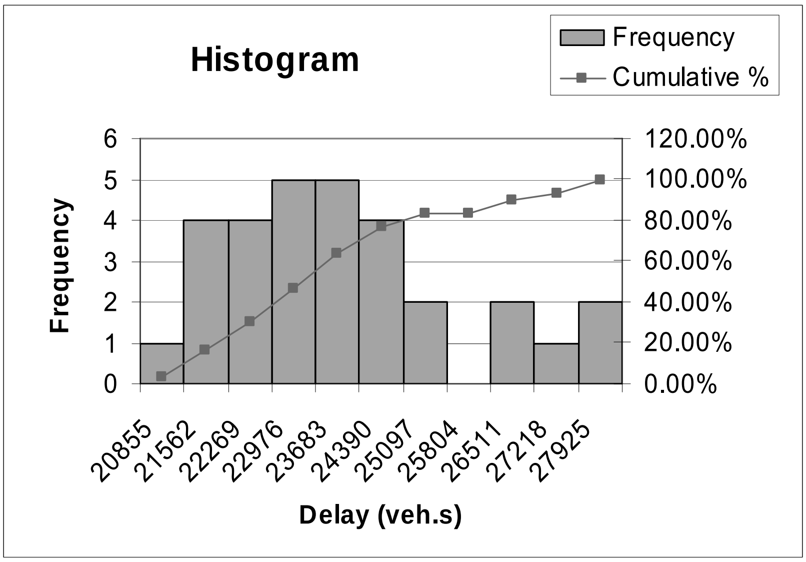


Figure D2: Histogram of the system delay obtained from AIMSUN using the optimal offsets based on TRANSYT-7F for the arterial case study

Histograms for the small network case study

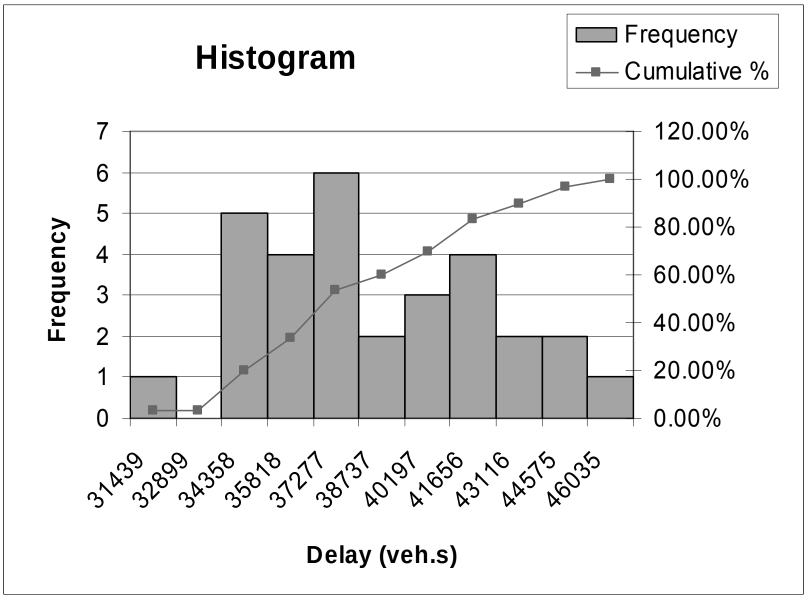


Figure D3: Histogram of the AIMSUN system delay using the optimal offsets based on Full Enumeration Method for small network case study

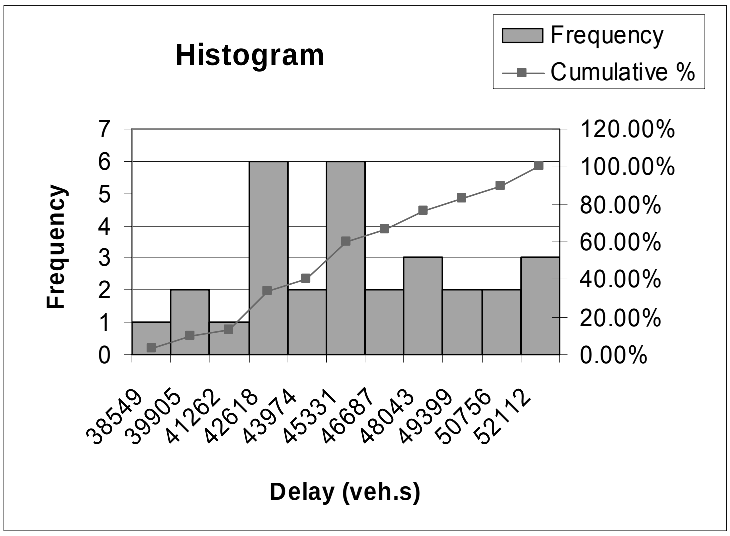


Figure D4: Histogram of the AIMSUN system delay using the optimal offsets based on dominance method for small network case study

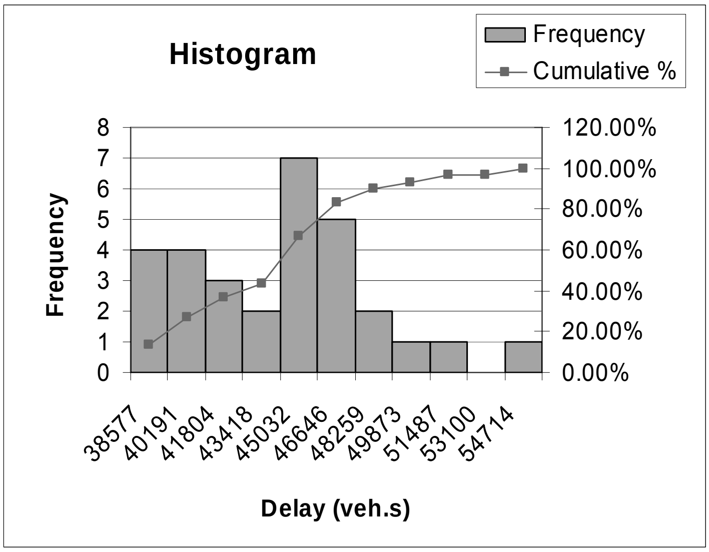


Figure D5: Histogram of the AIMSUN system delay using the optimal offsets based on TANSYT-7F for small network case study

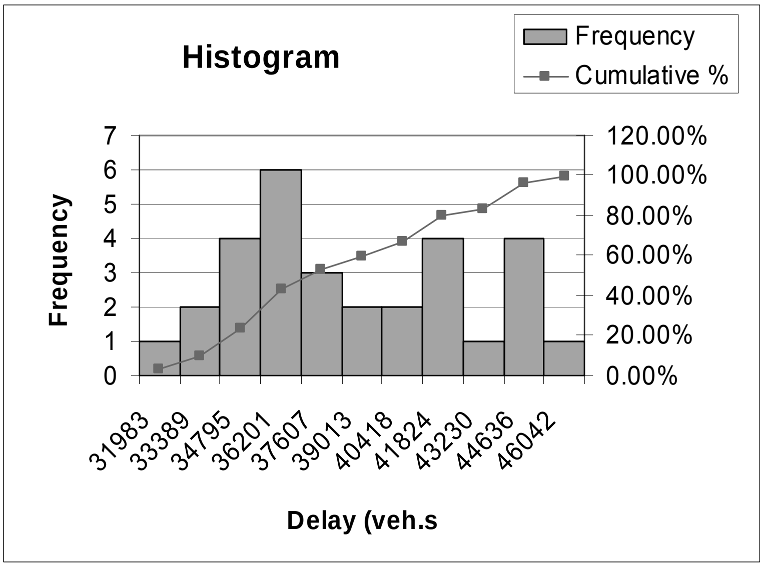


Figure D6: Histogram of the AIMSUN system delay using the optimal offsets based on SGA Method for small network case study

Histograms for the grid network

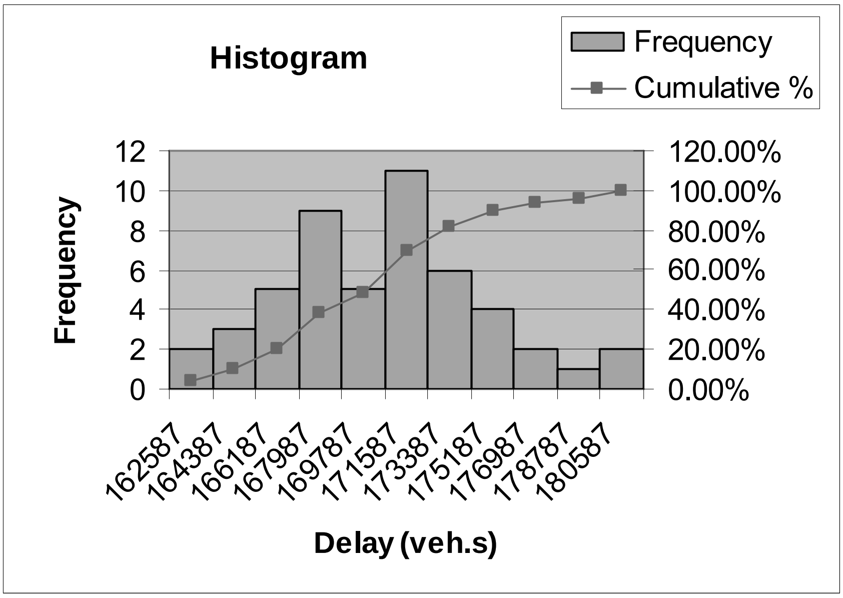


Figure D7: Histogram of the AIMSUN system delay using the optimal offsets based on dominance method for grid network case study

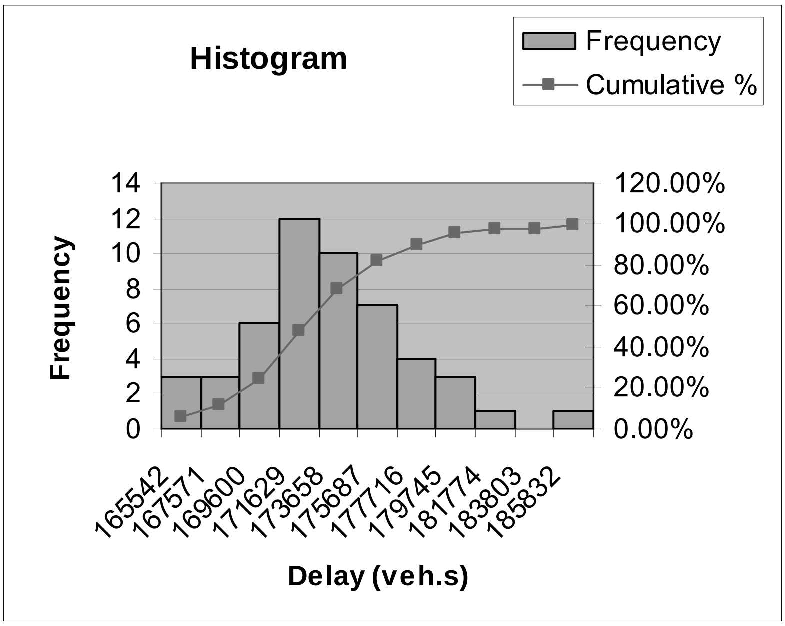


Figure D8: Histogram of the AIMSUN system delay using the optimal offsets based on TRANSYT-7F for grid network case study

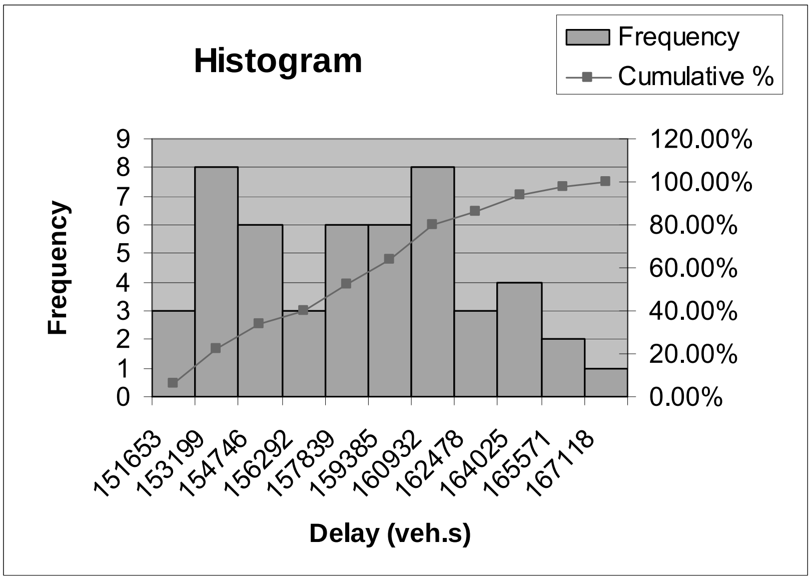


Figure D9: Histogram of the AIMSUN system delay using the optimal offsets based on SGA Method for grid network case study