MODELING AND EVALUATION OF INTEGRATED DYNAMIC SIGNAL AND DYNAMIC SPEED CONTROL IN SIGNALIZED NETWORKS

By

Hui Chen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Civil Engineering-Doctor of Philosophy

2013

ABSTRACT

MODELING AND EVALUATION OF INTEGRATED DYNAMIC SIGNAL AND DYNAMIC SPEED CONTROL IN SIGNALIZED NETWORKS

By

Hui Chen

A new integrated Dynamic Speed and Dynamic Signal (DSDS) control algorithm for signalized networks is developed in this research. The algorithm is formulated as a dynamic optimization problem with the objective of maximizing the number of vehicles released by the network and minimizing the number of stops in the network. The control algorithm is optimized by Genetic Algorithms (GAs).

The developed DSDS algorithm is applied to signalized networks. The benefits of implementing a DSDS control algorithm on network efficiency are first evaluated through looking at key measures of effectiveness (MOEs). It is demonstrated that the algorithm is able to reduce queues over time, avoid gridlocks, and improve system performance. Vehicle speed profiles under DSDS control and dynamic-signal fixed-speed (DSFS) control are compared to evaluate the advantages of control with dynamic speed on minimizing speed noise and speed variation. DSDS control generates smoother flow profiles by reducing speed noise and speed variation. The comparison provides evidence that implementing DSDS control in signalized networks is an effective way to achieve safer and environmentally friendly signalized network operations. The operational and safety enhancement brought about by the implementation of DSDS varies depending on the levels of driver compliance. The microscopic simulation model

VISSIM is used to evaluate the impacts of different levels of driver compliance. Results show that speeding and slow driving each have negative impacts on the performance of DSDS control.

Parallel GAs (PGAs) is investigated and deployed in order to improve computational performance. Both a simple GA (SGA) and island PGAs are used to solve the DSDS control problem, a standard GA-difficult, and a standard GA-easy problem. For all problems, savings in computation resources were realized when PGA was used. The magnitude of improvements brought about by a PGA depended on the difficulty of the problem. An empirical approach is explored to configure Parallel Genetic Algorithms (PGAs) to optimize the DSDS control algorithm developed in this research. Two of the most important island PGA parameters are examined: the number of islands (subpopulations) and the migration rate. The results show 1) increasing the number of subpopulations does not always bring worthwhile savings in time, 2) increasing the number of subpopulations decreases the importance of migration rate, 3) there is an optimal migration rate associated with each number of subpopulations and it is problem-dependent, and 4) PGA configuration and performance with the standard benchmark functions can be used as benchmarks to configure the PGA for problems of unknown complexity, such as the DSDS control algorithm developed in this research. The results suggest that off-line processing may be necessary to ensure optimal performance of the PGA.

ACKNOWLEDGMENTS

I would like to sincerely and gratefully thank my advisor, Dr. Ghassan Abu-Lebdeh for his invaluable guidance and for his continuous encouragement and support throughout my graduate study at Michigan State University. Without his patience and encouragement, it would not have been possible to finish this work. It has been a very enlightening and fruitful experience to work with Professor Abu-Lebdeh.

I would like to thank my committee members, Dr. William Taylor, Dr. François Dion and Dr. Erik Goodman for their valuable time, expertise, constructive comments, and suggestions.

I am also grateful for the invaluable helps granted to me by the faculties and staffs of Department of Civil and Environmental Engineering, colleagues in the department and close friends at MSU during the course of my study and research.

At last, I should also thank my family for their support and continuous encouragement. I thank my parents, Tieying and Zhuanyun, for their faith in me. They guided me through various situations in my life, allowing me to reach this achievement. Most of all I would like to thank my husband, Zhu wang. His perpetual love and constant support greatly encouraged me throughout my study.

TABLE OF CONTENTS

Chapter 7

Chapter 8

Chapter 9

LIST OF TABLES

LIST OF FIGURES

Chapter 1. Introduction

In the last two decades, congestion and safety have been two important issues in traffic engineering. In the United States, travel demand continues to grow rapidly due to fast population growth and increase in number of trips per capita. The increase in roadway capacity has been much slower than the increase in travel demands because of various factors, such as cost, community resistance, and environmental and social equity concerns, just to name a few. Construction of new roadways to increase capacity cannot keep pace with population increase and the resulting increase in travel demand. This explains why congestion is occurring everyday in urban areas and is getting worse continually.

According to estimates by the Texas Transportation Institute, 437 urban areas in the United States experienced 4.2 billion vehicle-hours of delay in 2007 [TTI 2007]. Congestion costs more than $78 billion annually including more than 4.2 billion hours of delay and 2.9 billion gallons of excess fuel consumed. These numbers have not accounted for the extra time that travelers and freight shippers would plan for in regards to important trips because of the unreliability caused by congestion areas. While congestion continues to worsen, safety issues also remain at a critical level. There are about 42,000 people killed and 3.3 million seriously injured every year [National Highway Traffic Safety Administration 2004]. The total cost of motor crashes is $231 billion annually including medical expenditures and property damages. Given these statistics, efforts must be made to improve roadway safety, use available roadway capacity more efficiently, manage congestion with order, and improve reliability of traffic flow.

Congestion and safety issues are much more serious in metropolitan areas, where a large portion of the traffic facilities consist of local streets and signalized intersections. Improving the safety and operation of signal network systems has become one of the primary objectives of many intelligent transportation system (ITS) applications such as advanced transportation management system (ATMS), advanced traveler information system (ATIS), and vehicle infrastructure integrations (VII). However, most of the research devoted to improving operations in signalized networks has emphasized developing strategies that deal only with signal timing parameters such as cycles, green splits, and offsets. In most of the existing control strategies, speed limit has thus far only been used as a safety constraint.

Selection and management of vehicle speeds directly impact road safety and quality of operation. Thus far only fixed speed limits have been used to guide drivers on their speed selections in signalized network systems. A traffic network system with fixed speed limits has limitations. For instance, drivers are constantly faced with changing conditions as they travel along a road, e.g., changes in lane width, changes in horizontal and vertical alignments, the presence of sink-source accesses and on-street parking, pedestrian and cyclist activity, etc. These conditions all need to be taken into consideration when selecting a safe speed to travel. Even on the same section of a roadway, different speed limits may be appropriate due to the change in traffic demand, which could vary by time of the day, day of the week, and season of the year.

Actual vehicle speeds impact the effectiveness of signal control measures and subsequent quality of traffic operations. For example, a signal control algorithm based on fixed speed (either a speed limit or a fixed average speed value) is used to calculate the optimal offset and green split for a specific movement at a certain time. However if the approaching vehicles travel at a speed that is either slower or faster than the speed assumed in the algorithm (i.e. the fixed speed), then the calculated ñoptimalò offset even the green splits are no longer optimal.

A more advanced and flexible control algorithm is one that combines the control of signal timing and vehicle speed management. With the advances in wireless communications, computation, sensor technologies and their deployment in intelligent transportation system settings, it is possible to develop and implement such algorithms. Integrated Dynamic Signal and Dynamic Speed (DSDS) control will not only improve the operational efficiency of a signalized network but will also improve roadway safety, reduce harmful emissions, and reduce energy consumption. It would achieve these benefits through reducing speed variations within a traffic stream and improving the effectiveness of signal timing plans. The major task of this research is to develop a control algorithm for signalized networks by applying dynamic signal control and dynamic speed control techniques and then identify the operational, safety and environmental benefits of the developed algorithm. Genetic algorithm (GA) is deployed in this study to optimize the algorithms developed. While the validation of the developed models is performed by using VISSIM, a microscopic simulation program, the evaluation of the algorithm has been conducted on both macroscopic and microscopic levels.

1.1 Problem Statement

Thus far, most of the research on signalized network control algorithms has focused on adjusting signal timing parameters only. In fact, due to the complexity of the control process in a signalized network, controlling signals alone (without considering speed as another control parameter) may not always produce optimal or even nearly optimal control strategies for signalized networks. This is especially true when responding to special events such as highly variable directional and bidirectional flows and large scale failures of system components (links and/or intersections). Hence, it is necessary to develop new and more flexible control methods which can tackle extreme traffic events and be fit for normal conditions as well.

In this section, the relationship between signal timing and vehicle speed is described first. After that, difficulties and limitations in signal-only control algorithms are presented. The potential advantages of using dynamic speed control and dynamic signal control are discussed at the end.

1.1.1 Signal Coordination

Signal coordination is one of the most important methodologies for efficient operation of signalized networks. Coordination aims at providing progression of traffic movements between known origins and destinations. Currently, most of the traffic control algorithms for signal coordination assume constant traveling speed over time on the same section of a road. This could be the speed limit or other value determined based on engineering judgments for normal traffic and environmental conditions. The following sub-sections briefly describe the basics of signal coordination and point out the limitations of fixed-speed signal coordination algorithms.

One-way signal coordination: Signal coordination on one-way streets is relatively simple. The offset between two adjacent intersections needs to be set to an ideal value to utilize the green time efficiently. According to Roess et al. [Roess et al. 2004], the ideal offset is calculated by equation (1-1)

$$ideal\_offset = \frac{L}{S} - Q \cdot h + l, \tag{1-1}$$

where the parameters are defined as follows:

L = distance between signals, ft

S = average speed on the link between signals, ft/s

Q = number of vehicles queued per lane, veh

h = discharge headway of queued vehicles, s/veh

l = start-up lost time, s

Because traffic conditions change over time, it is obvious that using constant speed value (speed limit or average speed) over time in the calculation of ideal offset is problematic. In some cases, there are long queues waiting at downstream intersections, vehicles released from upstream intersections will not be able to accelerate to the pre-specified fixed speed value before reaching the tail of the downstream queue. In other cases, drivers may choose to drive at higher speeds than the fixed speed value used in the calculation of ideal offset if they are not aware of the speed that they are supposed to be driving at. Overestimate or underestimate of actual vehicle speed will have severe consequences on the effectiveness of signal coordination.

Signal coordination for two-way streets and networks: For a two-way street, it is assumed that the offset of one direction is defined as *offset*1 and the offset of the other is defined as *offset*2. According to Roess et al. [Roess et al. 2004], the offsets of these two directions must satisfy:

$$offset1 + offset2 = nC \,, \tag{1-2}$$

where n is an integer and C is the cycle length.

Based on equation (1-2), if the offset of one direction is specified, then the offset of the other direction is automatically set. If the offset is set to the ideal for one direction, it is not always possible to have an ideal offset in the other direction.

Since it is difficult to find ideal offsets for both directions of a two-way street, one may believe that a one-way street system is the solution. However, signal coordination for networks is much more complicated. Even in one-way signal network systems, there still exist network closure problems. Network closure refers to the fact that setting offsets for one direction on three links automatically determines all offsets between all four signals in any set of four signals along a traffic loop. For the grid network shown in Figure 1-1, if *offsets*1, *offset*2, and *offset*3 are specified, the fourth offset *offset*4 is automatically determined.

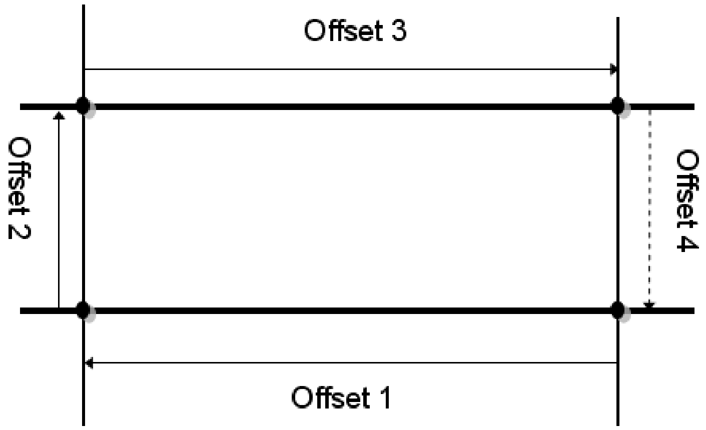


Figure 1-1 Network Closure Illustration

Offsets determined by the offsets of other directions or other links are called locked-in offsets. If the offsets (that are not locked-in offsets) are set to ideal values, locked-in offsets may not be suitable for the traffic they are serving. For instance, if *offsets*1, *offset*2, and *offset*3 in Figure 1-1 are set to ideal offset according to the traffic condition on the corresponding links, *offset*4 may not be ideal offset for the traffic it is serving. In a network with two-way streets, the offsets for secondary directions are also locked-in offsets, thus more of the offsets may not be suitable for the traffic they are serving.

1.1.2 Inefficient Use of Green Time

In order to use green time efficiently, not only do offsets need to be ideal, but traffic is also expected to pass the signal at or near saturation flow rate. But in reality, especially when traffic demand is lower than capacity, vehicles enter the system at lower than saturation flow rate. If only fixed speeds are used to guide drivers on the speed selection, vehicles may not be traveling in a platoon and at the same time maintain saturation headway. In this case, more green time will be needed to process the same number of vehicles traveling at longer time headways compared to the case where vehicles are traveling at shorter headways.

1.1.3 Integrating Speed Control with Signal Control

Non-ideal offsets and inefficient use of green time have a negative impact not only on operation of signalized networks, but also on safety and the environment. Speed control can play an important role in the control of signalized networks. The integration of speed control (i.e., changing the speed as traffic conditions and signal timing parameters change) and signal control may improve many aspects of traffic operations and safety along with significant benefits related to energy consumption and harmful emissions. However, no such algorithm exists today. The objective of this research is to develop, validate, and evaluate such an algorithm. In the integrated Dynamic Speed and Dynamic Signal (DSDS) control algorithms developed, signal timing parameters and speed parameters are optimized to achieve optimal system performance. It is important to note that the speed parameters to be optimized as part of the integrated algorithm are NOT traditional speed limit values which define maximum safe speed; they are the optimal speeds for the corresponding traffic condition and signal timing. They are the speeds at which drivers are expected to operate their vehicles under the DSDS control scheme. In this research, these speed parameters are defined as dynamic optimal speeds, which adapt to different traffic conditions and signal timing settings.

1.2 Research Objectives

The objective of this research is to develop an integrated dynamic signal control and dynamic speed control algorithms for signalized networks, and then evaluate the advantages of the developed control algorithm on various aspects of traffic operations, traffic safety, and the environment. The algorithms developed will consider the queue effect to ensure the ability to deal with oversaturated conditions as well as undersaturated conditions. The proposed research aims to fulfill the following more specific objectives:

1. Develop analytical dynamic signal and dynamic speed (DSDS) control algorithms for signalized networks
2. Develop a macroscopic simulation model to capture traffic flow conditions under DSDS control algorithms and calculate necessary measures of effectiveness (MOEs)
3. Optimize solutions using Genetic Algorithms and the macroscopic simulation model developed in 2 by using simple Genetic Algorithm (SGA) and parallel Genetic Algorithm (PGA)
4. Validate models using microscopic traffic simulation models
5. Identify and quantify the operational, safety, and environmental benefits of the algorithms by using microscopic traffic simulation models
6. Evaluate the impact of different levels of driver compliance on the effectiveness of DSDS control models
7. Improving efficiency of GA for online-implementation of DSDS algorithms

1.3 Research Tasks

The following tasks were executed to achieve the above objectives:

A. Review existing dynamic signal control literature on signalized networks

B. Review existing variable speed control literature

C. Review existing Genetic Algorithms-based techniques for solving signal control problems and other transportation related problems

D. Formulate dynamic signal and dynamic speed control algorithm

E. Develop C++ based DSDS algorithms and macroscopic traffic simulation model that will simulate traffic under DSDS control

F. Solve DSDS algorithms using both Simple Genetic Algorithm (SGA) and parallel Genetic Algorithm (PGA)

G. Validate DSDS algorithms using a microscopic traffic simulation. VISIM will be used

H. Evaluate DSDS control to examine and demonstrate benefits on aspects such as safety and the environment. The microscopic simulation software VISSIM will be used in this task.

I. Quantify the improvement of PGA over SGA on solving DSDS algorithms. Evaluate DSDS algorithms for on-line implementations.

J. Evaluate the impact of different numbers of subpopulations and migration rate on the efficiency of PGA

1.4 Dissertation Organization

This dissertation is organized in nine chapters. Chapter 1 introduces issues with signal-only controls for signalized networks and outlines research objectives, tasks, and thesis organizations. Chapter 2 provides the background and literature review for dynamic signal control algorithms, variable speed limit (VSL), and genetic algorithms (GA). Chapter 3 presents the development of the dynamic signal dynamic speed control (DSDS) models. Two different types of dynamic speed control strategies are described. The procedure and results of validation of throughput, delay, and number of stops models developed are presented in Chapter 4. Chapter 5 discusses how to use GA to optimize the DSDS control algorithm. It also demonstrates the application of DSDS control strategies to different sizes of signalized networks. Chapter 6 describes how DSDS control improves safety of signalized networks by minimizing speed noises and speed variations. In Chapter 7, the impact of different levels of driver compliance on the performance of DSDS models are analyzed. Chapter 8 presents the use of parallel genetic algorithms (PGA) to optimize the DSDS models and two other benchmark problems. An empirical approach of configuring PGAs is also described. Conclusions and recommendations for future subsequent research are summarized in Chapter 9.

Chapter 2. Literature Review

Traffic signals and speed limit signs are the two most common control devices in signal networks systems. So far numerous studies on signal controls have been conducted to optimize signal timing settings, and hence, improve traffic operations. Among these studies, speed and speed limits have only been considered as safety measures and are seldom considered as decision parameters for a signal system. Until now, only fixed speed limits are applied in real signalized networks.

The technologies of variable speed limits have only been developed and implemented on freeways in the United States and some European countries. With the help of advanced electronic and communication instruments, real-time dynamic speed information can be transmitted to drivers via overhead variable speed signs or in-vehicle display systems. In this research, dynamic optimal speed (i.e., the optimal or near-optimal speed at which drivers should drive at) is to be applied to signal network systems.

The main objective of this research is to develop an integrated dynamic signal and dynamic speed control algorithm for signalized networks. In the developed algorithm, the signal timing and speed parameters are adjusted intelligently according to prevailing traffic conditions. In such an algorithm, a larger number of traffic control parameters within dynamical systems are unavoidable. Genetic Algorithms, a powerful optimization approach, has been used to solve such complex and large scale optimization problems.

The following sections of this chapter review the knowledge and background materials related to variable speed limit/intelligent speed adaptation, dynamic signal, and Genetic Algorithms. In the first section, the applications of variable speed limit are summarized. After that, various types of dynamic signal control algorithms are presented. The last section describes the applications of GAs in transportation engineering.

2.1 Variable Speed Limit

Safety, throughput, delays, and emissions are the key measures of effectiveness in measuring the performance of any road segment. In general, all of these measures of effectiveness are greatly affected by the speed of vehicles, and thus varying speeds could have significant impact on the performance of a roadway section. The variable speed limit (VSL) system is a speed management system that alters speed guidance on a segment of roadway and posts a variable speed on Variable Speed limit signs or in-vehicle display devices. Such speed guidance can be advisory or mandatory. The superiority of using such a speed management system is obvious. The optimal operating speed of the same road may vary due to the constant change of traffic conditions, pavement, or construction and maintenance activities. Using a speed management system will improve safety and the capacity of the roadway segment in comparison to a fixed speed limit.

The variable speed limit system has been widely used for highway work zones and freeway management. Coleman [Coleman 1996] reported that the potential risk of rear-end collisions on freeways can be reduced if the traffic flow speeds are properly regulated with VSL. Pei-wei Lin et al. [Lin et al, 2004] developed VSL adjustment algorithms to fulfill the objectives of queue reduction and throughput maximization on highway work zones. Hegyi et al.[Hegyi et al., 2005] also presented a model predictive control approach to optimally coordinate variable speed limits for freeway traffic with the aim of suppressing shock waves. While different control algorithms are developed for different objectives, VSL systems are used more and more extensively in United State and some other European countries. Examples of the VSL system in Europe are discussed below [Robinson, 2000].

Since 1993, a variable speed limit system has been operational on the F6 toll way, south of Sydney in Australia. This system serves the objective of avoiding rear-end collisions during foggy days. It is an advisory system and the suggested driving speeds are displayed on variable speed limit signs, which are connected to loop detectors and visibility detectors. The determining factors which dictate a speed limit is based on the visibility distance and the speed of proceeding vehicles.

The experiments on the VSL system in Finland started in 1994, and it is aimed at improving road safety by influencing driversô behavior. The system operates on a rural highway. The recommended speed limits are given based on current weather information. Survey results indicated that 95 percent of drivers find the system more viable and satisfactory.

In France, a VSL system is active on a five mile urban section in Marseille. Speed limits are determined according to both prevailing speeds and weather conditions.

A VSL system was enacted in Germany in 1970 to achieve multiple objectives, including stabilizing flow under high demand, reducing crash probability, improving driver comfort, and reducing environmental impacts. The system is used on three rural autobahns. The variable speed is calculated based on collected traffic and environmental data and then entered into a computer-controlled algorithm. The German ministry of transportation has noted a reduction of crash rate by 20 to 30 percent and better driver compliances to variable speed limit signs at the same time.

The Netherlands has implemented VSL systems in both urban and rural settings. The system on urban highways is provided to promote safer driving behavior during foggy days. The posted speed is reduced with the reduction in visibility. The mean speeds are decreased by 5-6 miles per hour when fog is present. The enforceable system in the rural setting was installed in 1992 on a section of the A2 motorway between Amsterdam and Utrecht. The objective of this system is to suppress shockwaves and reduce crashes and congestion. Travel speed is determined by a system-controlled algorithm based on one-minute averages of speed and volume across all lanes. If an incident is detected, a speed reduction is applied. If the speeds are posted with a red circle, they are enforced by photo radar, if posted without the circle, they are advisory. The results of this system have been promising.

The United Kingdom used a VSL system on the M25 London Orbital to eliminate stop-and-go traffic conditions. The system is enforced by photo radar, and the speed limits are set based on vehicle volumes on the urban motorway. High driver compliance was reported and accidents declined by 10-15 percent. With the help of the VSL system, although travel speeds were lower, stop-and-go driving was greatly reduced. Meanwhile, adequate headways were maintained and driver compliance was improved.

In the United States, a number of states including Colorado, Michigan, Minnesota, Nevada, New Jersey, New Mexico, Oregon, and Washington have experimented with or deployed VSL systems [Robinson et al. 2002]. Probably the largest VSL system in the United States is in New Jersey [Wilkie 1997]. The VSL system on the New Jersey Turnpike uses approximately 120 signs. It uses weather sensing equipment and loop detectors for collecting input data. The VSL system is based on a logic that chooses the speed limit based on the average travel speed. The speed limits are displayed in increments of 5 mph. The speeds can be reduced to a minimum of 30 mph due to crashes, congestion, construction, ice, snow, fog, etc. The signs are used to provide motorists with information on unusual roadway conditions and are enforced by the state police.

As noted above, the application of variable speed limit has, so far, been limited to freeways. Use of a variable speed limit on freeways is successful; however, such applications have not been developed for signalized systems before. Research on control of signalized networks has been limited to optimizing signal timings. In this research, dynamic speed control algorithms have been successfully developed and applied to the control of signalized networks..

2.2 Dynamic Signal Control

In contrast to the pre-timed or fix-timed signal control, the dynamic signal control adjusts signal timing dynamically and intelligently according to prevailing traffic conditions. In this section, the development of a real time signal control is reviewed and then summarized. The dynamic signal has been combined with a dynamic speed control in this research.

SCOOT (Split, Cycle, Offset, Optimization, Techniques) is one of` the most famous adaptive signal control software. In this software, a real-time signal control procedure was successfully developed and applied [Bretherton et al. 1999, Hunt et al. 1991, Robertson and Bretherton 1991]. SCOOT is a model based system that is similar to off-line traffic network study tool TRANSYT [Chard and Lines 1987, FHWA 1984]. Robertsonôs recurrence relation is used to predict flow, delays and stops of traffic movement slowed by signal timings based on flow and occupancy data received from upstream detectors. Unlike TRANSYT however, SCOOT recalculates the prediction of flow, delay, and stops every few seconds to ensure rapid response to changing traffic conditions. Three optimizers are used in this SCOOT system. The cycle time optimizer computes an optimum cycle length for the critical intersection in the network. The split optimizer then assigns green splits for each intersection based on this cycle length and the offset optimizer calculates offsets. Cycle length, green splits, and offsets are then recalculated and implemented if required. This frequent update of traffic performance parameters and signal timings are made in order to achieve the concept of adaptive control [Hunt et al. 1991].

SCATS(Sydney Coordinated Adaptive Traffic System) is another well-known software [Lowrie 1982, Luk 1984]. It combines theoretical concepts of traffic movement at intersections and empirical considerations using a so called ñcritical intersectionò control philosophy. In this scheme, interrelated intersections are grouped with one intersection in every group identified as the critical intersection. Traffic demand at the critical intersection sets the cycle time for the group in a real-time manner. Control parameters are adjusted in every cycle by small amounts in accordance with traffic demand. Strategic decisions on the control parameters (i.e., using pre-defined library of split and offset plans) are made for the critical intersection and implemented for other intersections in the group. This control scheme uses detectors at stop lines for both central and local control decisions. Even though detectors are required at every intersection for full adaptive control operation, the concept of critical intersection provides a flexibility to operate adaptive controls with detectors installed only at the critical intersection.

OPAC is a real-time signal control algorithm based on the optimal switching of signal timings [Gartner 1982, 1983]. OPAC uses dynamic programming to identify the optimal signal timing sequence for a relatively long period (from 50 to 100 seconds). It was developed for use on a single isolated intersection. In this control algorithm, time is divided into stages, which are sub-divided further into an integer number of intervals. Each stage allows one to three signal changes. For any given switching sequence at a specified stage, a performance function expressing the total delay is calculated. The switching sequence that has minimum delay is chosen, and the first part of it is implemented (using the rolling horizon concept). With this rolling horizon concept, OPAC defines the optimal switching based on both actual vehicle arrivals and predicted arrivals. Prediction of arrivals makes the model very sensitive to error in the estimation of future arrivals.

Other adaptive signal control algorithms include LA-ATCS [Vincent and Young 1986], UTOPIA [Donati et al. 1984], PRODYN [Henry et al. 1983, Henry and Farges 1989], RHODES [Head 1997, Head and Mirchandani 2001].

The models described above are well designed for undersaturated conditions. Under oversaturated conditions, they cannot provide accurate control strategies because they do not explicitly describe the over-saturation phenomenon. Yagar and Dion presented results to show that

SCOOT performed well in moderate traffic conditions but had major deficiencies in oversaturated and highly fluctuating conditions [Yagar and Dion 1996]. The most recent research on SCOOT added a congestion term in its signal optimization function and a queue management strategy known as gating to improve its congestion and incident management facilities. The field experiment results show that the gating strategy would decrease delay by 26% [Bretherton et al. 1999]. Wolshon and Taylor also indicated in their study that SCATS was only effective at reducing delay during low volume periods [Wolshon and Taylor 1999].

At high traffic demand, non-optimal signal timing could cause queues to develop at intersections and therefore cause the intersection to operate under oversaturated conditions. Oversaturation is a condition that occurs when queues of vehicles fill entire blocks and interfere with the performance of adjacent upstream intersections [Pignataro et al. 1978]. Chaudhary [Chaudhary 1997] has shown that queue spill back could result in a significant reduction of capacity. The objective of signal control for oversaturated networks should be to prevent blockages and gridlocks caused by queue spillback and efficiently use available green time. The studies done by Abu-Lebdeh [Abu-Lebdeh 1999] and Park et al. [Park et al. 2000] provided a framework for developing dynamic signal coordination and queue management algorithms for oversaturated arterials. The work of Abu-Lebdeh [Abu-Lebdeh 1999] was based on dynamic queue management of a single arterial system. Park et al. [Park et al. 2000] developed a mesoscopic simulator, which was able to model oversaturated traffic conditions. Girianna et al. [Girianna and Benekohal 2002] extended Abu-Lebdehôs model to grid traffic networks with oversaturated intersections.

In all these studies, only signal timing parameters (green split, cycle and offset) are considered as control parameters. The first attempt of using speed control in signalized arterial systems was made by Abu-Lebdeh [Abu-Lebdeh 2002]. An algorithm was developed to dynamically control speed in conjunction with dynamic signals for control and management of a single urban arterial and the results showed that integrating the control of these two kinds of parameters can improve system performance including average travel speed, number of stops, and travel delay. Chen and Abu-Lebdeh [Chen and Abu-Lebdeh 2006, 2007] developed a framework for integrating dynamic speed control and dynamic signal control. That study proved the feasibility of integrating dynamic speed control and dynamic signal control and provided preliminary assessment of the operational improvements. However, not all potential benefits of using dynamic speed in signalized networks were explored.

In the DSDS control algorithm developed in this research, speed is also considered as a control parameter and not just a regulatory safety-based measure. The dynamic (variable/adaptive) speed and dynamic (adaptive/real-time) signal control are combined to realize a whole new approach to controlling or managing traffic in signalized networks.

2.3 Genetic Algorithms

Optimization based on Genetic Algorithms (GAs) has become very popular in recent years. GAs have been successfully applied in many fields such as traffic control, computer science, bioscience, statistics, and fluid mechanics, etc. These techniques are based on the mechanism of natural selection and natural genetics [Goldberg 1989] and are considered emerging optimization tools due to the simplicity, minimal problem restriction, parallelism, and global perspectives that they offer. GAs have also been successfully used in transportation engineering. Foy et al.[Foy et al. 1992] reported on using GAs to determine signal settings. Hadi and Wallace [Hadi and Wallace 1993] used GAs to obtain optimal signal coordination strategies. Memon and Bullen [Memon and Bullen 1996] compared GAs to the Quasi-Newton gradient method and chose GAs for real-time optimization of signals. Abu-Lebdeh [Abu-Lebdeh 1999, 2002], Girianna and Benekohal [Girianna and Benekohal 2002] used GAs to obtain a near-optimal solution for signal coordination and queue management along oversaturated arterials. Sadek et al. [Sadek et al. 1997] solved a dynamic route assignment problem using GAs. Duerr [Duerr 2000] used GAs to dynamically control right of ways for transit vehicles in mixed traffic arterials.

Most of the GA applications in transportation have only used Simple GA (SGA) due to its simplicity. SGA operates on a group of chromosomes (a population of individuals) by selecting the most influential individual through genetic forces. Parallel Genetic Algorithms (PGAs) distributes the task of a single-population GA to different processors. As those tasks are implicitly parallel, little time will be spent on communication, and thus, the algorithm runs much faster and finds better results. Probably the first attempt to map Genetic Algorithms to existing parallel computer architectures was made by Grefenstette [Grefenstette 1981]. PGAs have been applied in many search, optimization, and machine learning problems [Belding 1995, Grefenstette 1981, Schnecke and Vornberger 1996, Chen et al. 2004, 2005].

2.4 Summary and Conclusions

This chapter reviews the previous studies on the variable speed limit, dynamic signal control, and the application of Genetic Algorithms in traffic engineering. Research on the signalized network control is dominated by signal timing control algorithms with fixed speed limits. Studies on the variable speed limit are limited to applications on freeways and the results show that using a variable speed limit has great potential on regulating traffic flow, improving operations, safety, and driver compliances. With the advances in wireless communications, computation, sensor technologies and their deployment in intelligent transportation system settings, it is possible and necessary to develop and implement more advanced and integrated algorithms to control signal timing and speed in order to achieve better operations for signal networks. As discussed earlier, the speed parameters to be optimized in this study are optimal speeds that drivers are expected to drive at. This is different from the traditional speed limit concept. These optimized speed parameters are denoted as dynamic optimal speed.

Chapter 3. Dynamic Signal and Dynamic Speed Control

Dynamic Signal and Dynamic Speed control (DSDS) algorithm developed in this research integrates the dynamic signal and dynamic speed technologies. It provides optimal signal timing and optimal speed combinations for prevailing traffic conditions. Two kinds of DSDS algorithms are developed in this research: cycle-based and vehicle-based algorithms.

In the cycle-based algorithm, traffic departures, arrivals and queue status are evaluated at the end of each cycle and only one speed decision is made for vehicles leaving the intersection during the same cycle. The vehicle-based DSDS algorithm, on the other hand, will evaluate traffic departures, arrivals, and queue status at an interval shorter than cycle length, and specify multiple speed decisions in one cycle as needed. Additional information on cycle-based and vehicle-based algorithms is discussed below.

The cycle-based DSDS algorithm should be employed when networks are operating at near or over capacity, where vehicles enter the system at saturation flow rate. With proper signal timing and dynamic speed control strategies, vehicles will maintain constant and tight headway and only one speed decision is needed for vehicles leaving an intersection in the same cycle. The vehicle-based algorithm should be employed when traffic demand is lower than capacity, where vehicles enter the network at lower than saturation flow rate. Hence not all vehicles will travel in a platoon and saturation flow rate cannot be reached if the system is controlled by cycle-based DSDS algorithm. More green time will be needed to process the same number of vehicles traveling at longer time headway compared to where vehicles are traveling at shorter headways. The vehicle-based algorithm can specify a speed decision for each platoon or even each vehicle to help regulate traffic flow to a desired pattern according to the system control strategy. For example, the vehicle-based algorithm can reduce the headway between vehicles by setting a higher speed for the following vehicle than the vehicle ahead until required headway is achieved. Thus less green time is needed to process the same number of vehicles. As a result, system capacity can be used more efficiently and delays will be reduced.

Both algorithms are developed to optimize traffic operation in signalized networks through adjusting signal timing parameters and speed parameters according to prevailing traffic conditions. In the following sections, the potential benefits of DSDS models are conceptually demonstrated first. Then the development of a cycle-based DSDS algorithm and vehicle-based DSDS algorithm are described along with the formulations.

3.1 Potential Benefits of DSDS

The idea of using speed as a control variable (as opposed to being a constraint) that can be optimized dynamically and used in controlling signalized networks is explored in this research. Such speed is an operating speed, not a speed limit. This section conceptually demonstrates how to use dynamic operating speed to realize different control objectives including improved traffic flow and safety.

The objectives of this research are to examine the deficiencies in the signal-only approach in controlling signalized networks under different traffic conditions, conceptually explore possible remedies using dynamic speed algorithms, and then analyze the benefits of different control algorithms. This section presents the ideas and the supporting evidence using a theoretical approach as well as common assumptions on fundamental traffic properties.

The potential benefits of using dynamic speed control algorithms are discussed using the traffic network system shown in Figure 3-1. In such a system, only two intersections are under consideration. It is assumed that the signal timings are optimized by existing signal timing algorithms with a fixed speed limit. As discussed before, network closure conditions, unbalanced flow on different directions, and sub-optimal signal timing are inevitable, especially in larger urban traffic signalized networks because of the changes in traffic condition over time. The deficiency of control with signal-only control under different traffic conditions is discussed, and dynamic speed control remedies are then described based on different control objectives.

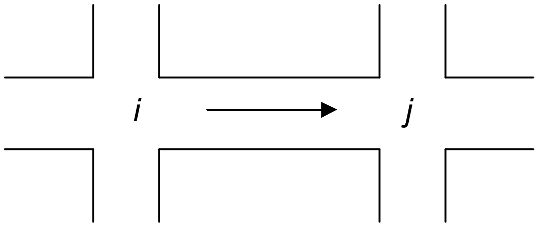


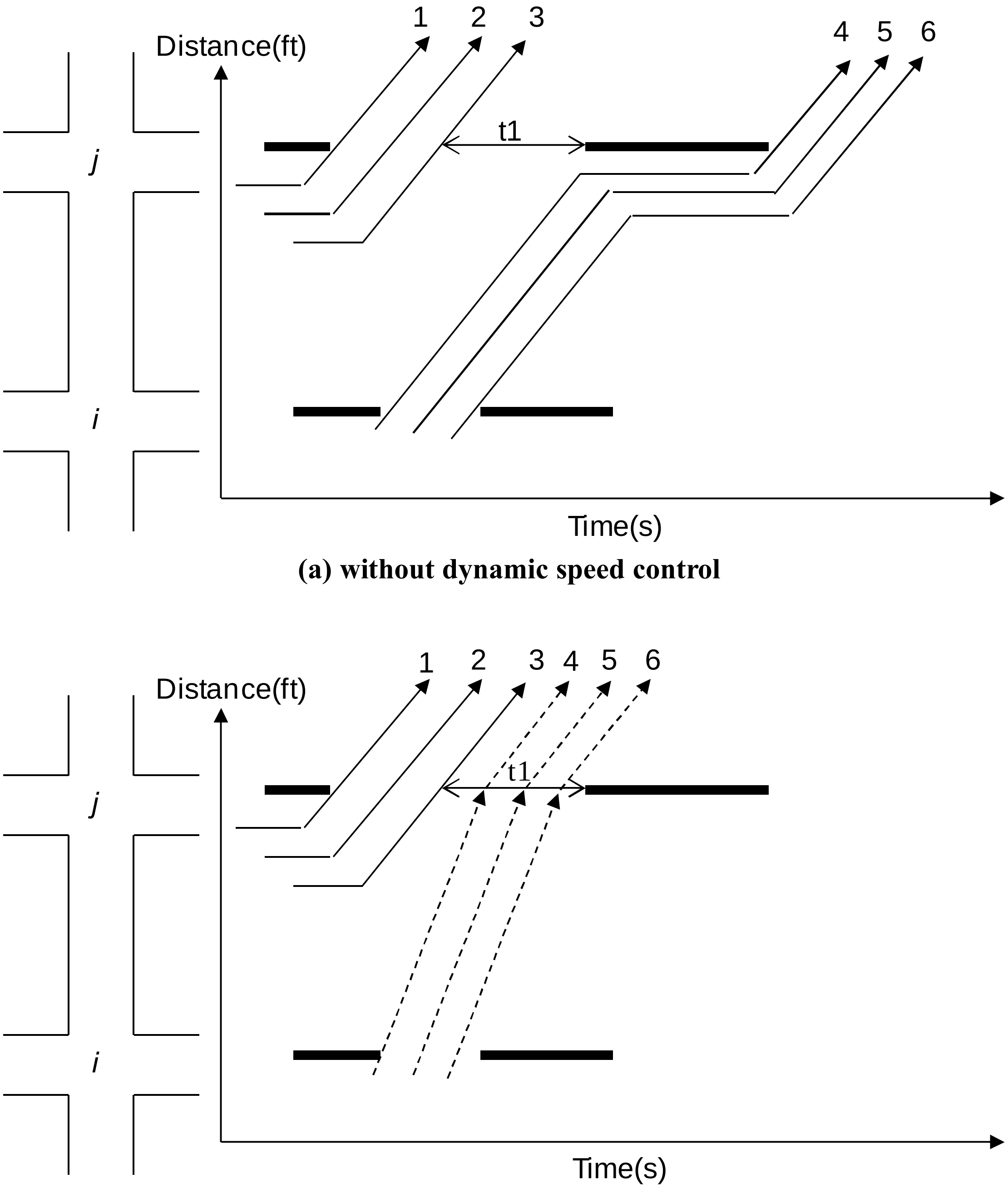
Figure 3-1 Two-intersection system

3.1.1 Better Signal Coordination

When fixed speed is used in the optimization of signal timing, non-ideal offsets are inevitable because of the change of traffic condition over time, network closures, roadway designs, and unbalanced flow with different directions. Having a sub-optimal signal coordination plan has negative impacts on the operation of signalized networks. Two examples are illustrated in Figure 3-2(a) and 3-3 (a).

In Figure 3-2(a), since the offset between intersections *i* and *j* is smaller than the ideal offset, the first vehicle (vehicle #4) released from intersection *i* during a typical cycle arrives at the downstream intersection *j* and is stopped by the red light. Under this scenario, the total wasted green time is *t1*. But if dynamic speed control is employed (in this case vehicles are guided to travel at higher speeds than the fixed speed limit), then such a wasted green could be avoided. The vehicle trajectories of the improved traffic flow due to use of DSDS are shown in Figure 3-2(b). In this case, under the same signal timing plan, the system with dynamic speed control produces higher system throughput, lower delay, less number of stops and shorter average travel time.

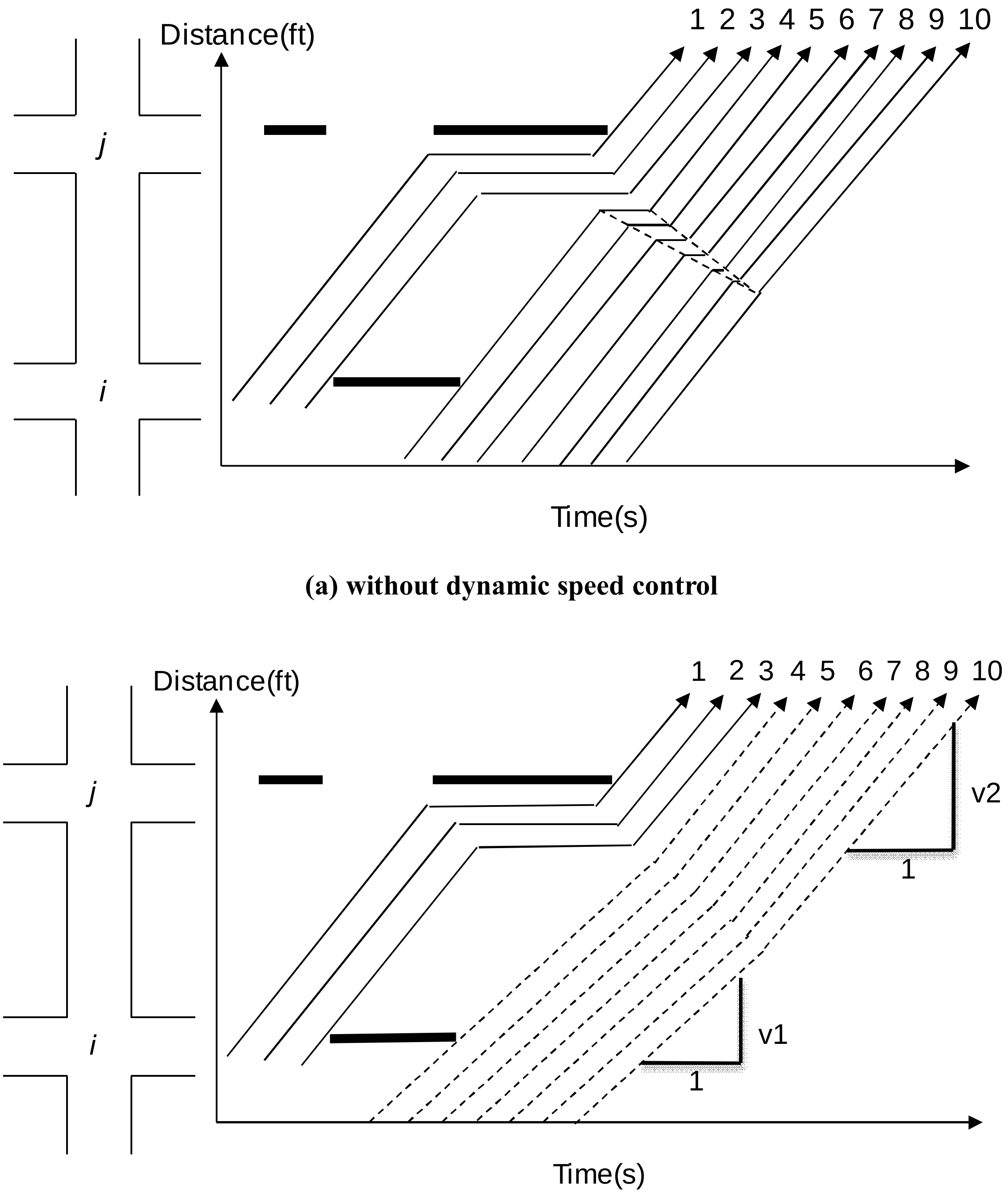
In the second case, shown in Figure 3-3(a), the offset between intersections *i* and *j* is larger than the ideal offset. Vehicle #4 was the first vehicle released from intersection *i* during a typical cycle and arrived at intersection *j* before the traffic clears there. The vehicle has to slow down and stop, and then accelerate to a desired speed. In this case, if the vehicles leaving intersection *i* are guided to travel at lower speeds, smoother traffic flow could be achieved while system throughput, delay, and travel time are kept the same. The smoother traffic flow achieved with dynamic speed control as shown in Figure 3-3(b) is more favorable from safety, fuel consumption, and environmental emission perspectives.



(a) without dynamic speed control

(b) with dynamic speed control

Figure 3-2 Vehicle trajectories when actual offset is smaller than ideal offset



(a) without dynamic speed control

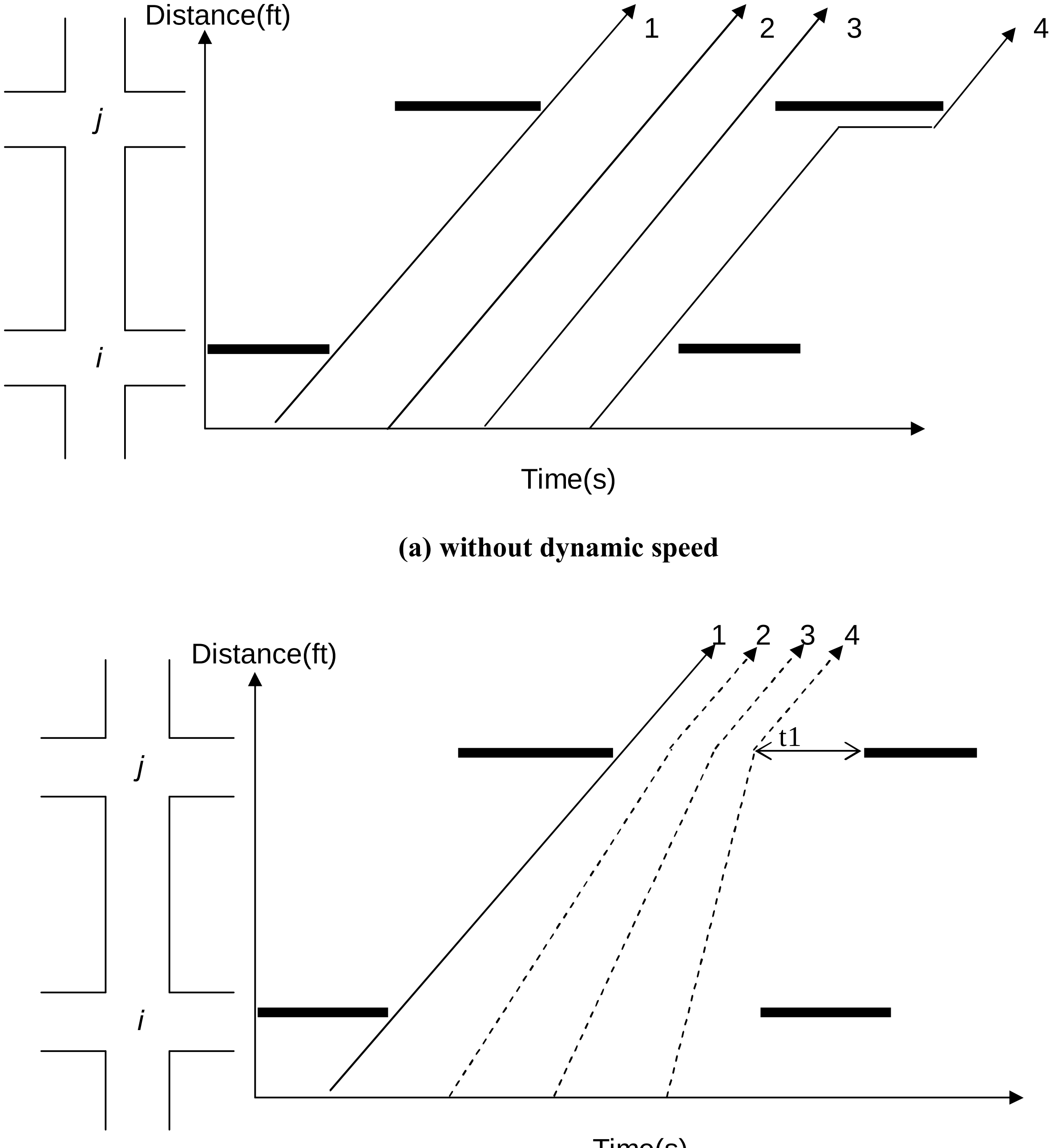
(b) with dynamic speed control

Figure 3-3 Vehicle trajectories when actual offset is bigger than ideal offset

3.1.2 Efficient Use of Green Time

If vehicles are traveling at uniform and saturation headways, green will only be wasted if the offset is varied from the ideal one as shown in Figure 3-2(a). But vehicles enter traffic networks at random and larger headways rather than uniform and tight headways as shown in Figure 3-4(a). This is especially true for lower demand situations. Under such conditions, even though the offset is set to ideal offset, green time can still be wasted between the vehicles if all vehicles are guided to travel at the same speed. However, a vehicle-based dynamic speed control can potentially prevent that.

When vehicles travel in the system with larger headways, more greens are required to process the same number of vehicles. In order to use green time more efficiently, dynamic speed could be used to guide the following vehicles to travel faster than the leading vehicle until tight headways are achieved. Figure 3-4 (b) shows the improved traffic flow after a vehicle-based dynamic speed control is implemented. Green saved in this case can be used to serve more vehicles if traffic demand is high. If traffic demand is low, saved green can be used for other conflicting movement such as left turning movement from an opposite direction or through and left turning traffic of a crossroad. This is especially valuable when the network is experiencing unbalanced flow in different directions, which is very common.



(a) without dynamic speed

(b) with dynamic speed

Figure 3-4 Vehicle trajectories when actual offset is bigger than ideal offset

3.1.3 Dynamic Speed Control Provides More Flexibility

Besides the potential benefits of operation, safety, and fuel consumption, dynamic speed control also adds flexibility to the control of signalized networks. In many cases, dynamic speed can supplement signal timing control and improve the effectiveness of signal timing control. Consider the case where signal timings for the signals along a long corridor have been optimized and coordinated for normal traffic conditions. If for any reason there is overflow and residual queues on one or more links, the offsets between the signals at the end of these links need to be changed according to prevailing (new) traffic conditions in order to maintain good signal coordination. Such changes may necessitate changing of signal timing for more or even all signals along the arterial. Changing in signal timings along the arterial might also impact the signal timing and signal coordination on crossroads. Such transitions between different signal timings and coordination plans may incur considerable delays [Pearson (Update 11/01/01)]. However, with DSDS in place, good coordination can be achieved by adjusting only the speeds on the impacted link or links. No changes to signal timings are needed elsewhere. Employing dynamic speed control, in some cases, is easier and more cost effective to use than traditional signal-only control algorithms.

3.2 Implementation of DSDS Algorithms

DSDS control in signalized networks is envisioned to be implemented in VII systems setups with functionalities emulating those found in closed-loop control systems. Vehicles and infrastructure components are envisioned to be able to communicate and exchange information to enhance both traffic operations and safety. As noted by the US Department of Transportation, the ñcoordinated deployments of communication technology in all vehicles by the automotive industry and on all major U.S. roadways by the transportation public sector are the fundamental building blocks of the VII conceptò [RITA [updated 2007]]. In a VII setting, the roadway would be divided into segments. Each segment has a ñhot postò that collects data from traffic along that particular segment using Dedicated Short Range Communications (DSRC). Vehicles can communicate with roadside units (RSU), or ñhot postsò and within themselves via the hot posts. Hot posts of different roadway segments can communicate with each other. Operationally, a vehicle would transmit anonymous on-board sensor data to an RSU every time it passes an RSU. A vehicle would store time samples of the data it collected between successive RSUs and then transmit these data samples as it passes the RSU. The anonymous data received at an RSU would be sent to an aggregation point from which it is then forwarded to authorized subscribers (e.g., Traffic Operations Centers, DOTs, etc.). Each aggregation point may receive data from several thousand RSUs. All data are organized and ordered by the geographic coordinates from the vehicles and would be available to authorized subscribers. At a higher level, each section of the road ñknowsò what all other roadway segments are experiencing, which means that vehicles on any given segment can ñknowò what is happening with traffic on downstream segments. A simplified architecture of data flow is shown in Figure 3-5 [DOT 2005]

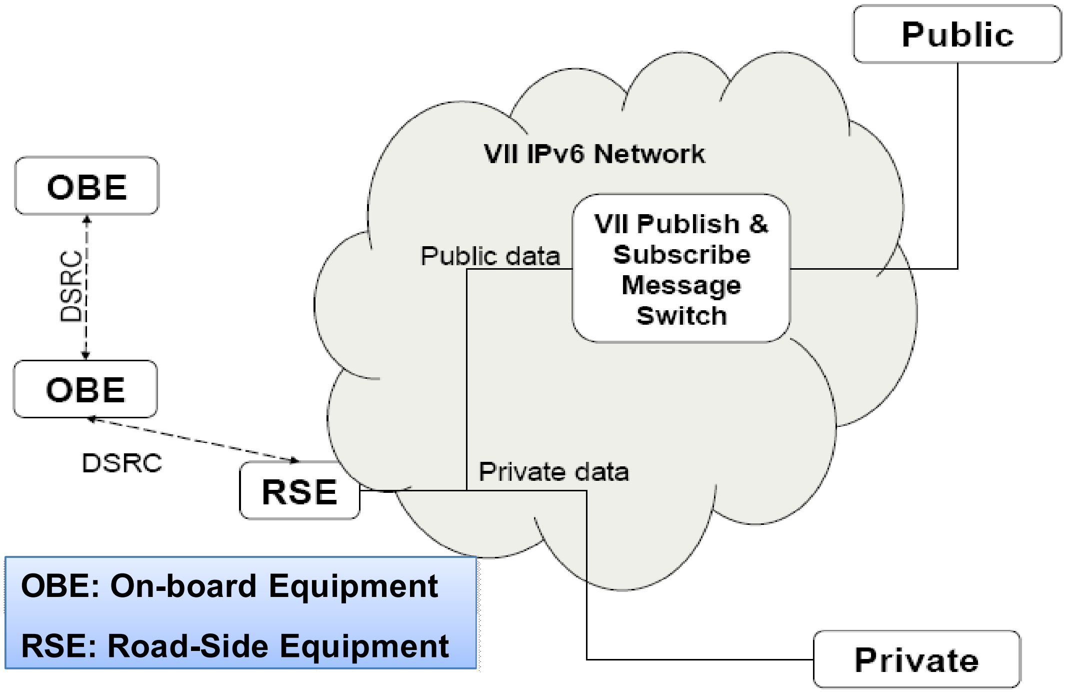


Figure 3-5 Simplified architecture data flow (source: Ref. [DOT 2005]). For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

In signalized networks, a VII based DSDS system would require a display medium, which could be either Variable Message Signs (VMS), or in vehicle display devices. The display media is directly connected via dedicated communication channels to a traffic management center where real time traffic data from the hot posts and traffic signals are collected, processed, and used in decision making. There, near-optimal speeds are determined. The VMS and/or the in-vehicle devices would be used to display a speed (depending on control objectives and level of information made available to motorists) when a vehicle enters a link. If VMSs are used, they would be installed overhead along with traffic signals. The VMSs can be used to provide lower information levels where a group of vehicles are instructed to follow a given speed. Such a setup would limit the flexibility of the control algorithm, but would require less detection, computation and communication resources. Departures, arrivals and queue status would be evaluated at the end of each cycle and only one speed decision is made for vehicles leaving the intersection during the same cycle. On the other hand, if in-vehicle devices are to be used to provide a higher level of information whereby speed information can be customized for every individual vehicle (or vehicles in a given platoon; in this case speeds of vehicles would be additionally constrained to meet given safety thresholds). With this higher level of information more flexible control can be provided, but in this case more detection, computation and communication resources will be required. The arriving and departing volumes and queue status would be evaluated at smaller time steps, all depending on the vehicle arrival rate.

3.3 Definitions and Variables

A sample network N(3,3) as shown in Figure 3-6 is used to demonstrate the development of the algorithm. N (*ny*,*nx*) denotes a grid traffic network of *ny* intersections on all north-south arterials and *nx* intersections on all east-west arterials. Such a network will have *ny***nx* signals and 2*[(*ny*+1)**nx*+(*nx*+1)**ny*] one-way links. Thick lines in the figure indicate independent arterials, and small arrows indicate optimized greens. Detailed discussions on independent links are in section 3.5.2., and on optimized greens are in Chapter 5.

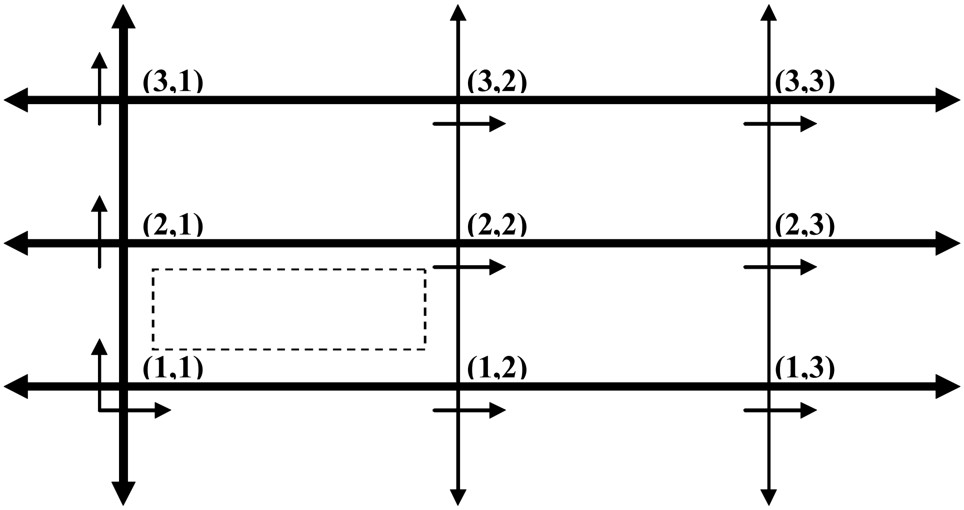


Figure 3-6 A sample grid network (Thick lines indicate independent arterials; small arrows indicate optimized greens)

The definition of the variables used in this chapter are summarized and listed as follows:

S_i: Saturation flow rate per lane of approach *i*, *veh/hr*;

LV: Average effective vehicle spacing when stopped, *ft/veh*;

acc: Acceleration, ft/sec^2 ;

$L_{((i,j),(l,m))}$: Length of the link between signal (*i*,*j*) and signal (*l*,*m*), *ft*;

l: Start-up lost time for each phase, *sec*;

h: Average time headway between two vehicle, *sec*;

vu: Speed of start-up shockwave when signal turns green, *ft/s*;

vt: Speed of stop shockwave when signal turns red, *ft/s*;

n: Number of cycles to be simulated and evaluated;

$off_{((i,\ j),(l,m))k}$: Offset between signal (*i*,*j*) and (*l*,*m*) during cycle *k*. In this study, an offset is negative when downstream signal turns green first.

When the following variables are used for one specific direction, an extra letter, which indicates the direction, will be put in front of the variable name. For example, *eg* (*i,j*)k refers to effective green time for the eastbound approach at signal (*i,j*) during cycle *k*.

$g_{(i,j)k}$: Effective green time for an approach at intersection (*i,j*) during cycle *k*, *sec*;

$gs_{(i,j)k}$: Starting time of green for an approach at intersection (*i,j*) during cycle *k*;

$ds_{(i,j)k}$: Dynamic Optimal speed on approach to intersection (*i,j*) during cycle *k*, *ft/sec*;

$dv_{(i,j)k}$: Discharged vehicles from an approach at intersection (*i,j*) during cycle *k*, number of vehicles;

$av_{(i,j)k}$: Vehicles received on an approach at intersection (*i,j*) during cycle *k*, number of vehicles;

$q_{(i,j)k}$: Length of queue on approach at intersection(*i,j*) at the beginning of cycle *k*, number of vehicles.

Other variables will be defined where they are first used.

DSDS algorithms are formulated as an optimization problem with the objective of maximizing throughput. A disutility term is used to account for the queues accumulated at the intersections and the stops caused by non-ideal offsets. To allow the algorithm to effectively deal with over-saturated and under-saturated traffic conditions, a set of constraints are developed to avoid unfavorable situations under oversaturated traffic conditions. Such unfavorable situations include wasted green and queue spillback.

3.4 Constraints for Oversaturated Condition

Control of oversaturated intersections requires two key constraints: 1) Use green time efficiently to maximize system throughput; 2) Prevent spillback of downstream queues into upstream intersections to avoid gridlock and subsequent capacity loss. Methods to satisfy these constraints are discussed in the following sub-sections.

3.4.1 Use Green Efficiently

Using green time efficiently can be attained through strict and intelligent control of offsets. An offset between two adjacent signals is set such that the first vehicle of an upstream traffic arrives at the downstream signal when the tail of the downstream queue clears the intersection. In order to differentiate from ideal offsets that are calculated based on fixed speed parameters, the offset that can achieve this objective in DSDS algorithm is defined as *compact offset* and is denoted by *c_off* in this dissertation. A compact offset can be computed based on link length *L*, queue length at downstream intersection *q*, and dynamic optimal speed *ds*. Given an upstream signal (*i,j*) and a downstream signal *(l,m),* if the upstream vehicle starts from a stopping position, the compact offset *c_off* can be calculated by

$$c\_off = \left(L + ds^2_{(i,j)k} /(2 \cdot acc) \right) / ds_{(i,j)k} - q_{(l,m)k} \cdot h \tag{3-1}$$

otherwise *c_off* should be calculated by

$$c\_off = L / ds_{(i,j)k} - q_{(l,m)k} \cdot h \tag{3-2}$$

With compact offsets, tight average time headways *h* can be maintained under DSDS control. Furthermore, it is assumed that vehicles can maintain this headway as they leave the intersection. Using compact offsets not only provides efficient use of green, but also reduces the number of stops.

3.4.2 Prevent Defacto Red

A defacto red condition develops when a signal is green but traffic can not move because of backed-up traffic on the receiving link. In order to prevent this from happening, green time at the upstream signal should be less than or equal to the sum of the effective green time of the downstream signal, the offset between the two signals and the time that takes for a stopping shockwave to propagate to the upstream intersection, as described by Abu-Lebdeh [Abu-Lebdeh 1999]. For two-signal system (*i*,*j*) and (*l*,*m*), this condition can be expressed by:

$$g(i,j)k \le g(l,m)k + \mathit{off}((i,j)(l,m))k + L(i,j),(l,m)/vt \tag{3-3}$$

where (i,j), $(l,m) \in \mathrm{N}(ny,nx)$, $k = 1, 2,$é n, and (i, j) is the immediate upstream signal to signal (*l*,*m*). Here function *g* is the available green, $\mathit{off}_{((i,j)(l,m))k}$ denotes the actual offset between intersection (*i*,*j*) and (*l*,*m*), $L_{(i,j),(l,m)}$ is length of the link between intersection (*i*,*j*) and (*l*,*m*), and *vt* is the speed of the stopping shockwave.

3.5 Other Constraints

Besides the constraints for oversaturated conditions described in the previous section, there are also some other constraints that need to be satisfied under any traffic conditions.

3.5.1 Queue Storage Capacity

Queue along any link should be less than the storage capacity of that link:

$$q(i,j)k < q_{\max}. \tag{3-4}$$

This constraint is also necessary under oversaturated conditions because the length of the queue has to be short enough such that it does not block the traffic movements of the upstream intersections. Depending on the specific queue management requirements, $q_{\max}$ is within the range shown in equation (3-5)

$$0 < q_{\max} \le \frac{L(i,j)(l,m)}{LV}, \tag{3-5}$$

where LV is the average vehicle spacing. When $q_{\max}$ equals to zero, it represents the extreme situation that no queue is allowed at the intersection. Note that this constraint could be adaptive (*i.e.*, time-dependent or traffic conditions-dependent).

3.5.2 Network Closure

For a signalized grid traffic network, the network closure constraints always exist, as discussed by Roess et al [Roess et al. 2004]. For the grid network shown in Figure 3-6, if the offsets of any three links of a loop are set for directional traffic flow, then the offset of the fourth link is automatically fixed. This constraint for a single loop can be extended and applied to the whole network: if the offsets of all east-west streets are independently specified, then the offsets of one north-south street ñlock inò all other north-south offsets. Similarly, if the offsets of all north-south streets are independently specified, then the offsets of one east-west arterial lock in all other east-west offsets. If the locked-in offsets are not suitable for prevailing traffic conditions, serious problems can develop especially for a congested network. Under congested conditions this may lead to spillbacks and gridlocks, which can be catastrophic, especially during emergencies.

By using a dynamic speed controller in conjunction with the signal controller, a solution is provided to this problem: the dynamic speed controller is responsible for activating and controlling the optimal speed selection, and the signal controller would control the phasing and timing of the signals.

In this research, streets are divided into two groups for a grid traffic network: independent streets and dependent streets. Independent streets are those whose offsets are set to compact offset independently, while the dependent streets are those that have the ñlocked-inò offsets. For instance, as in the N(3,3) network shown in Figure 3-6, all east-west arterials are independent arterials and arterial A(1,1), (3,1) is the only independent north-south arterial.

In the grid network shown in Figure 3-6, the loop consisting of intersections (1,1), (1,2), (2,2) and (2,1) is used as an example to illustrate how to calculate the lock-in offset. If the offset between intersection (1,1) and (1,2), offset between (2,1) and (2,2), and offset between (1,1) and (2,1) are specified independently, the offset between signals (1,2) and (2,2) can be calculated by the following equation:

$$off((2,2)(1,2))k = \left(NB\_gs(1,1)k + off(1,1)(2,1)k + NB\_g(2,1)k + off(2,1)(2,2)k + EB\_g(2,2)k\right) - \left(NB\_gs(1,1)k + NB\_g(1,1)k + off(1,1)(1,2)k + EB\_g(1,2)k\right) \quad (3\text{-}6)$$

where *gs* denotes the start of green. The two letters before the variables are to indicate direction. For example, *NB_gs* denotes the start of green for the northbound (NB) direction. The terms in the first bracket on the right hand side of this equation calculate the time the northbound green at signal (2,2) starts. The terms in the second bracket calculate the time the northbound green at signal (1,2) starts.

3.5.3 Range of Control Parameters

Control parameters are the green splits, offsets, and dynamic optimal speed. Offsets on independent streets are set to compact offset, and offsets on dependent links are locked in by offsets for independent streets. Green splits and dynamic optimal speed should be within a reasonable range. Equations (3-7) and (3-8) present the formulations of the allowable effective green time and speed selection.

$$g_{\min} \le g(i,j)k \le g_{\max} \quad (3\text{-}7)$$

$$ds_{\min} \le ds(i,j)k \le ds_{\max} \quad (3\text{-}8)$$

3.6 Traffic Modeling-Macroscopic Simulation Model

In order to evaluate the effectiveness of any control strategy, traffic flow and queue states need to be properly modeled. Macroscopic models are formulated to evaluate flows and queues at signalized intersections. A macroscopic model is sufficient for providing aggregated MOEs for network level optimization and evaluations.

3.6.1 Cycle-Based Algorithm

In the cycle-based algorithm, speed decision is calculated when a signal turns green, while flow and queues are evaluated at the end of each cycle. For a given cycle, the traffic volume departing from any approach of signal (*i,j*) depends on the effective green time available, saturation headway, queue at the beginning of the cycle, arrival flow from the upstream intersection *i*, and the offset between the two signals. This is shown in the following equation,

$$dv_{(i,j)k} = \begin{cases} \min\left(q_{(i,j)k} + av_{(i,j)k}, \dfrac{g_{(i,j)k}}{h} \right) \quad if \;\; c\_off_{(i,j)(l,m)k} \geq off_{(i,j)(l,m)k} \\ \min\left(q_{(i,j)k} + av_{(i,j)k}, \dfrac{g_{(i,j)k} - \left(c\_off_{(i,j)(l,m)k} - off_{(i,j)(l,m)k} \right)}{h} \right); \\ \qquad\qquad if \;\; c\_off_{(i,j)(l,m)k} < off_{(i,j)(l,m)k} \end{cases} \tag{3-9}$$

where $(i, j),\ (l, m) \in N(ny, nx)\ and\ \ k = 1, 2, ...n.$

The queue status for cycle k+1 is determined by

$$q_{(i,j)k+1} = q_{(i,j)k} + av_{(i,j)k} - dv_{(i,j)k} \qquad (i, j) \in N(ny, nx), k = 1,2,...n\,. \tag{3-10}$$

Assuming that there is no sink and source between the two intersections, the arrival volume can be calculated by

$$av_{(l,m)k} = dv_{(i,j)k} \quad for\ \ (i, j), (l, m) \in N(ny, nx), k = 1,2,...n\,, \tag{3-11}$$

where (*i, j*) is the immediate upstream signal to (*l ,m*).

3.6.2 Vehicle-Based Algorithm

In the vehicle-based algorithm, speed decision is calculated when a vehicle leaves the intersection. Compared to the cycle-based algorithm, flow and queue status need to be evaluated and updated with a smaller time step æt. The value of æt is depending on the prevailing traffic demand and control strategies. The departing volume for any given time interval from any approach of intersection (*i,j*) can be calculated by

$$dv_{(i,j)t} = \begin{cases} 0 & \textit{when signal is red} \\ \min\left(q(i,j)_{t+\Delta t}\ , \dfrac{\Delta t}{h}, \left(L - LV \cdot q(l,m)_{t+\Delta t}\right)/\,LV \right) & \textit{when sigal is green} \end{cases} , \quad (3\text{-}12)$$

where $(i, j),(l,m) \in N(ny,nx)$ and (*i,j*) is the immediate upstream signal of (*l,m*). The departing volume for any given time interval should be less then the number of vehicles that can be accommodated on the receiving link to prevent spillback.

The queue status at time $t + \Delta t$ is determined by

$$q_{(i,j)t+\Delta t} = q_{(i,j)t} + av_{(i,j)t} - dv_{(i,j)t} \qquad (i,j) \in N(ny,nx) \qquad (3\text{-}13)$$

where it is assumed that there is no sink and source between the two intersections. And the arrival volume is calculated by

$$av_{(l,m)t} = dv_{(i,j)t} \quad \textit{for} \ \ (i,j),(l,m) \in N(ny,nx)\,, \qquad (3\text{-}14)$$

where (*i, j*) is the immediate upstream signal to (*l ,m*).

3.7 Objective Function

A fitness function is used to evaluate the effectiveness of a set of decision variables in satisfying the control objective. The fitness function is calculated by:

$$T(c) = throughput - \lambda \cdot stop\,, \tag{3-15}$$

where *c* denotes a set of decision control parameters. The objective function is to maximize the value of fitness function, as shown in equation (3-16)

$$\max(T(c)) = \max(throughput - \lambda \cdot stop)\,. \tag{3-16}$$

The first term ñ*throughput*ò in this equation is the throughput of a network and can be calculated by adding up the number of vehicles released at all exit intersections of the network. The second term ñ$-\lambda \cdot stop$ò represents a disutility function which penalizes the occurrence of both queues and stops caused by non compact offsets. Note that extra stops will only be caused by a non-compact offset that is larger than compact offset. The number of stops can be calculated by the following procedures.

As shown in Figure 3-7, *t*1 is the difference between the actual offset and compact offsets:

$$t1 = off - c\_off\,. \tag{3-17}$$

*t*2 is the time that the stopping shockwave will travel before it meets the starting up shockwave and can be calculated by the following equation:

$$t2 = \frac{t1 * vu}{(vu - vt)}, \tag{3-18}$$

where vu is the speed of the start-up shockwave and vt is the speed of the stopping shockwave.

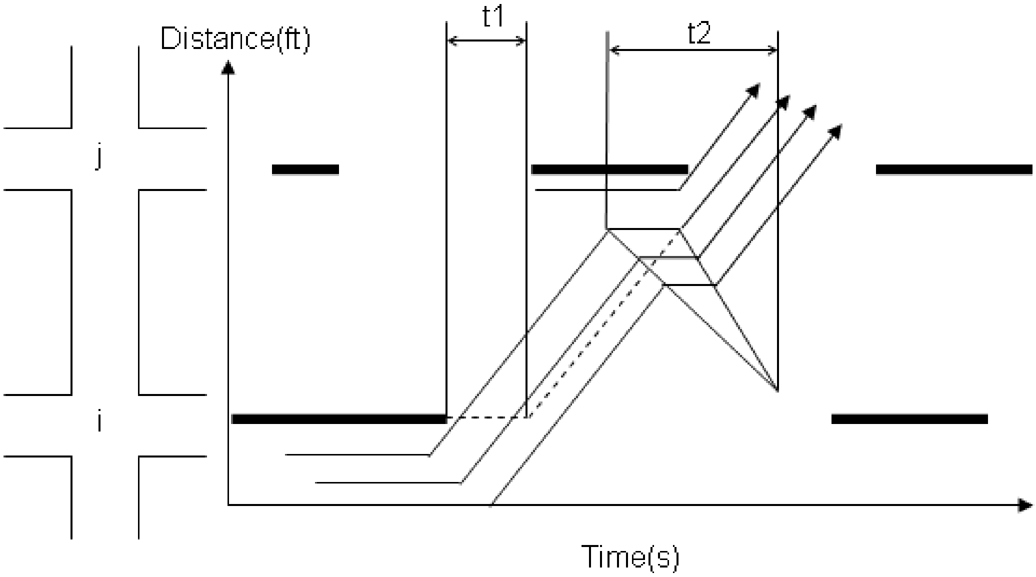


Figure 3-7 Calculation of stops caused by non compact offset

The number of stops caused by the non-compact offset can be calculated by

$$stop(i,j)k = t2/h. \tag{3-19}$$

Therefore the number of stops for one approach at intersection (*i,j*) during cycle *k* can be calculated by

$$stop(i,j)k = q(i,j+1)k + \left(off(i,j)(i,j+1)k - c\_off(i,j)(i,j+1)k\right) \cdot vu/(vu - vt) \cdot h \tag{3-20}$$

The number of stops needs to be calculated for all approaches and for all intersections. The number of stops for the system can be calculated by

$$stop = \sum_{k=1}^{n} \sum_{i=1}^{ny} \sum_{j=1}^{nx} \left(estop(i,j)_k + wstop(i,j)_k + nstop(i,j)_k + sstop(i,j)_k \right), \qquad (3\text{-}21)$$

where n is the number of cycles that are evaluated. The letter before the variable ñstopò indicates the direction. For example, $estop_{(i,j)k}$ denotes the number of stops that occurred on the EB link of intersection (i,j) during cycle k.

3.8 Delay Model

Except the throughput and number of stops, travel time delay is another important measure of effectiveness (MOE) used in this research. Travel time delay is defined as the difference between ideal travel time and actual travel time. Ideal travel time is defined as the travel time when vehicles are continuously traveling at desired speeds. The two-intersection system in Figure 3-8 is used to explain the calculation of travel time delay. Vehicle 1 is the first vehicle in the queue for cycle 1, vehicle 2 is the first vehicle that leaves the upstream intersection during cycle 1. In this case, the actual offset between the two intersections during cycle 1 is smaller than the compact offset. Vehicle 2 leaves the upstream intersection at $gs_i(1)$, where $gs_i(1)$ is the time when signal at intersection i turns green during cycle 1. It arrives at the downstream intersection at $gs_i(1) + c\_off(1)$ (see the definition of c_off in section 3.3.1), and the time it leaves downstream intersection j is also $gs_i(1) + c\_off(1)$. Vehicle 4 is the first vehicle in the queue for cycle 2. It leaves the upstream intersection at $gs_i(1) + (dv_j(1) - q_j(1)) * h$, where h is the average time headway, and it leaves the downstream intersection at $gs_j(2)$. Vehicle 6 is the first vehicle that leaves the upstream intersection during cycle 2 and it serves as an example for the case when the actual offset is larger than the compact offset. Vehicle 6 leaves upstream intersection i at $gs_i(2)$, and the time it leaves the downstream intersection j is $gs_j(2)+q_j(2)*h$. It is obvious that travel time for vehicle 2 is t2= $c\_off_j(k)$, travel time for vehicle 4 is t3= $gs_j(2)-(gs_i(1)+(dv_j(1)-q_j(1))*h)$ and travel time for vehicle 6 is t2=$off_j(2)+q_j(2)*h$. Based on the examples discussed, for the first vehicle leaving the upstream intersection during cycle k, the travel time on link i->j can be calculated by

$$t_{i->j}(k)=\begin{cases} off_j(k)+q_j(k)*h,\ if\ off_j(k)\geq c\_off_j(k) \\ \\ c\_off_j(k),\ if\ off_j(k)<c\_off_j(k) \end{cases} \tag{3-22}$$

where $off_j(k)$ is the actual offset between intersection i and j during for cycle k, $c\_off_j(k)$ is the compact offset between intersection *i* and *j* for cycle k, $q_j(k)$ is the initial queue at intersection *j* for cycle k.

Except for cycle 1, the travel time of the first vehicle in the queue during cycle *k* can be calculated by

$$qt_{i->j}(k)=gs_j(k)-(gs_i(k-1)+(dv_j(k-1)-q_j(k-1))*h) \tag{3-23}$$

The total travel time of vehicles in initial queue of cycle 1 can be calculated by

$$qT_{i->j}(1)=\frac{q_j(1)(q_j(1)-1)}{2}*h \quad . \tag{3-24}$$

Because DSDS model will regulate traffic to maintain known and tight time headway to use green as efficiently as possible, the following vehicles that leave link *i->j* during the same cycle will have similar travel times as the first vehicle. Except for cycle 1, all of the vehicles in the queue at the beginning of a cycle will have the same travel time too. The total travel time of all vehicles on link *i->j* during cycle k can be calculated by

$$T_{i->j}(k)=\begin{cases} qT_{i->j}(1)+t_{i->j}(k)*(dv_j(k)-q_j(k)); & k=1 \\ qt_{i->j}(k)*q_j(k)+t_{i->j}(k)*(dv_j(k)-q_j(k)); & k>1 \end{cases} \tag{3-25}$$

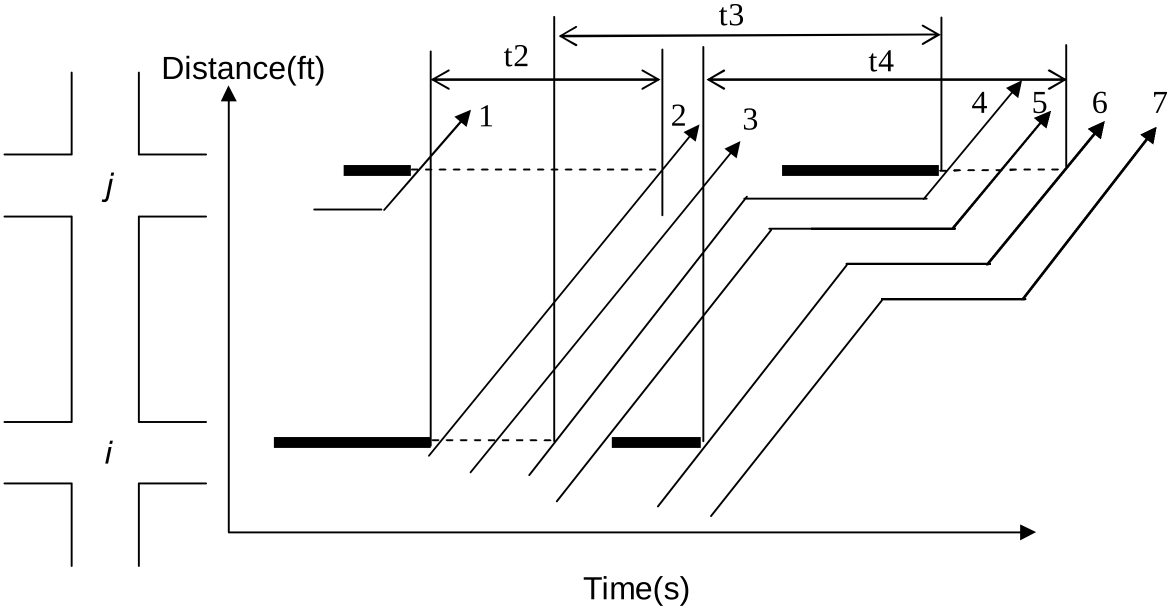


Figure 3-8 Calculation of travel time delay

The total distance traveled by all vehicles on link *i->j* during cycle k can be calculated by

$$D_{i->j}(k)=\begin{cases} \frac{q_j(k)(q_j(k)-1)}{2}*SD+L_{i->j}*(dv_j(k)-q_j(k)); & k=1 \\ L_{i->j}*dv_j(k); \quad k\neq 1 \end{cases} \tag{3-26}$$

where $L_{i->j}$ is the link length of link *i->j* and SD is the average distance headway for the vehicles in the queue.

The ideal travel time for cycle k can be calculated by

$$T\_i_{i->j}(k) = D_{i->j}(k) / ds_i(k) \tag{3-27}$$

where $ds_i(k)$ is the desired speed on link *i->j* during cycle *k* (please refer to the variable list in section 3.3).

Travel time delay on link *i->j* can then be calculated by

$$d_{i->j} = \sum_{k=1}^{n} (T_{i->j}(k) - T\_i_{i->j}(k)) \tag{3-28}$$

If there are *n* intersections in the network, the total travel time delay of the network can be calculated by equation (3-29)

$$d = \sum_{i=2}^{n} d_{i-1->i} \tag{3-29}$$

3.9 Summary and Conclusion

Potential benefits and implementation of DSDS models are presented in this chapter. Two types of DSDS models are developed for different traffic conditions. The cycle-based model is for high demand conditions where vehicles are traveling in a platoon, and the vehicle-based model is for lower traffic demand. A set of constraints were developed and they must be fulfilled during the optimization process. The objective function of DSDS models is formulated to maximize the throughput and decrease the number of stops caused by queue and non-compact offset. Macroscopic models are developed to describe the traffic operations under DSDS control and calculate MOEs. And the models for calculating MOEs are validated by microscopic simulation in the next chapter.

Chapter 4. Validation of Throughput, Stop, and Delay Models

The models developed for calculating throughput, number of stops, and travel time delay are either used in the objective function or used to evaluate the effectiveness of the dynamic speed and dynamic signal (DSDS) models developed. Hence, it is very important to make sure that these models are realistic and able to describe the operational performance of traffic flow in the signalized networks. The accuracy of these models can ensure that a near-optimal solution be achieved by an optimization process and provide accurate evaluation of the effectiveness of DSDS models developed.

Ideally, the best way to validate the models is to use a real signal system. However, due to limited resources, the throughput, number of stops, and travel time delay models are validated by VISSIM, a microscopic traffic simulation software. VISSIM does not optimize signal timings but is able to simulate traffic flows under given signal timings and speed control. In this chapter, VISSIM is introduced first. Next validation procedures of developed models are presented. After that, the validation results are given. At the end, the summary of validation results and concluding remarks are given.

4.1 VISSIM

VISSIM is a discrete, stochastic, time step-based microscopic traffic flow simulation model that has become increasingly popular. VISSIM is unique in the way that the roadway network geometry is coded. Links and connectors are used to build any network of functional roadway classification [PTV 2005]. In comparison to the link-node based structure, this approach enables the user to control vehicle paths as well as the interaction of vehicles within intersections (nodes). With VISSIM, it is possible to model any kind of intersection or arterial/intersection networks with a precision down to one millimeter.

In terms of operations, VISSIM is very flexible. It has the capability of modeling complicated intelligent transportation system control strategies such as ramp metering, transit signal priority, dynamic lane control signals, etc. VISSIM also provides a flexible platform with many user-definable features and an Application Programming Interface (API) which allows users to implement customized control logics. This is especially useful in modeling the DSDS control algorithms that are developed in this research.

Several studies have investigated the performance of VISSIM compared to other popular traffic microscopic simulation packages. Moen et al. [Moen et al. 2000], Bloomberg and Dale [Bloomberg and Dale 2000], and Tian et al. [Tian et al. 2002] had investigated the performance of VISSIM and compared it to CORSIM, a popular traffic microscopic simulator developed by the Federal Highway Administration (FHWA) and studied it extensively over the past 30 years. VISSIM compared favorably, especially in terms of analysis of complicated multi-model transportation systems and systems with current or future ITE strategies. Other advantages of VISSIM include [Leyva et al. 2004] :

- Driver behavior parameters are adjustable to provide flexibility in calibration and validation.
- Superior 3D graphics with viewing from any position and angle.

- Ability to ñpopulateò non-transportation features such as buildings, trees and people for high-quality graphic output.
- Dynamic traffic assignment ï Ability to use O-D trip tables.
- Can use GIS layers and/or ortho photos to help define inputs and reference animation output.

VISSIM has disadvantages of microscopic models as well. It can be very data intensive and time consuming to construct a scenario, especially from scratch. In the next section, the procedures of validation are presented by using VISSIM.

4.2 Validation Procedures

In this section, two major steps of the validation procedures using VISSIM are described. These steps are model calibration and experiment setup.

4.2.1 Model Calibration

A basic two signal system as shown in Figure 3-1 is built in VISSIM and used for validating MOE models. This basic network has 2 signalized intersections and a one-lane link. Length of the link is 800 ft. All signals are assumed to have two phases. Turning movements are ignored.

Once the VISSIM network is constructed, multiple simulation runs are made to calibrate the parameters used in the macroscopic level throughput, number of stops, and delay models developed in chapter 3. The purpose of calibration is to ensure that the developed models can appropriately describe the same traffic conditions as in the VISSIM network. The calibrated parameters include average time headway, the speed of start-up shockwave, stopping shockwave, acceleration rate, and average stopping distance headway.

4.2.2 Experiment Setup

In this research, a total of 10 scenarios are tested with VISSIM. Each of these scenarios has different initial queues, traffic demand, signal timings and desired speed. All scenarios were simulated for 10 cycles. For each given scenario, simulation was performed with 5 different random seeds. The throughput, number of stops, and delay values from VISSIM simulation were recorded and the average values were compared to the results calculated from the developed macroscopic models, which were described in Chapter 3. The simulation results are presented and discussed in the next section.

4.3 Validation Results

The simulation results of VISSIM, including the throughput, number of stops, and delay, are shown in Table 4-1, Table 4-2, and Table 4-3, along with the results calculated by the developed MOE models. Statistical tests were conducted to determine the significance of the differences between the MOEs obtained by VISSIM simulation and those calculated by developed models. The lower part of the table shows the result of the paired t-test for the above MOEs. The paired T-test considers the null hypothesis (NH) against the alternative hypothesis (AH). For throughput, NH claims that there is no significant difference between the throughput calculated from the developed DSDS model and those from VISSIM simulation.

Table 4-1 Paired t-test for throughput

No.	VISSIM	DSDS	Difference	Difference in percentage (%)
1	215	211	4	1.86
2	229	225	4	1.75
3	193	190	3	1.55
4	265	272	-7	-2.6
5	234	242	-8	-3.4
6	247	250	-3	-1.2
7	170	156	14	8.24
8	256	251	5	1.95
9	158	155	3	1.9
10	176	174	2	1.14
Paired t-test t-value:0.8406 critical t-value two-tail: 2.2622 P(T<=t)two tail:0.4223				

With the degree of freedom of 10-1 = 9, for the 5% level of significance, the critical T value was 2.2622. For the throughput, the T-value was 0.8406, which is less than the critical T value. Therefore the difference between the throughput calculated from the developed DSDS model and those from VISSIM simulation was not significant at the 5% level, and NH is accepted. The P-value is 0.4223, which was high and confirms that the NH is accepted. In other words, the throughput calculated from the model was not significantly different than those that resulted from VISSIM.

Similarly, T = 1.9640 for the number of stops, and T = 0.1962 for delay. Both were less than the critical T-values at the 5% level of significance, i.e., 2.2622. The number of stops and delay calculated from the models were statistically equal to those of VISSIM simulation.

Table 4-2 Paired t-test for number of stops

No.	VISSIM	DSDS	Difference	Difference in percentage (%)
1	0.03	0.03	0	0
2	0.27	0.26	0.01	3.7
3	0.02	0.02	0	0
4	0.57	0.55	0.02	3.51
5	0.36	0.36	0	0
6	0.07	0.07	0	0
7	0.01	0	0.01	0
8	1.08	1.06	0.02	1.85
9	0.04	0.05	-0.01	-25
10	0.6	0.59	0.01	1.67
Paired t-test t-value:1.9640 critical t-value two-tail: 2.2622 P(T<=t)two tail:0.0811				

Table 4-3 Paired t-test for delay

No.	VISSIM	DSDS	Difference	Difference in percentage (%)
1	4.5	4.7	-0.2	-4.44
2	37	34.9	2.1	5.68
3	0.2	0.1	0.1	50
4	17.7	17.7	0	0
5	18.4	18.9	-0.5	-2.72
6	19.3	21	-1.7	-8.81
7	0.5	0.4	0.1	20
8	24.5	22.8	1.7	6.94
9	14.6	15.6	-1	-6.85
10	19.6	19.5	0.1	0.51
Paired t-test t-value:0.1962 critical t-value two-tail: 2.2622 P(T<=t)two tail:0.8488				

4.4 Conclusions

Throughput, number of stops, and delay models developed in chapter 3 are validated by using the microscopic simulation program VISSIM. Ten different scenarios are tested and five random seeds were used for all scenarios. The MOE values from VISSIM simulation were compared to those calculated from the developed macroscopic models. The T-test was utilized to compare the three MOEs. The results have shown that the differences of throughput, number of stops, travel time and delay were not statistically significant. This confirmed that the developed models for MOE calculation are valid and can be used to accurately evaluate traffic flow under the DSDS control model developed in this research.

Chapter 5. Using GA to Optimize DSDS Control

This chapter presents the procedures of using Genetic Algorithms (GA) to optimize a dynamic signal and dynamic speed (DSDS) control. For the integrated DSDS algorithms developed in this research, near optimal signal timings and dynamic optimal speeds need to be determined for extended control period. The DSDS control is formulated as a large combinatorial optimization problem. In this study, GAs are used to solve the problem due to their power, speed, and effectiveness in solving optimization problems. GAs are searching techniques based on the mechanism of natural selection and natural genetics [Goldberg 1989]. The process of optimization by GA is realized by evolving a population of candidate solutions by repeatedly applying genetic operators such as selection, crossover, and mutations. GA is regarded as a well established optimization tool due to its simplicity and capability for solving large combinatorial problems and parallelism.

The first step of optimization of the DSDS control is to code all decision control parameters into chromosomes. The coding of decision control parameters and the calculation of all other control parameters is described in the first section. Constraint handling and penalties are discussed in the second section, followed by discussion of fitness calculation. The last section describes and analyzes traffic operation as two sizes of traffic networks under near-optimal DSDS control solutions.

5.1 Control Parameters

Due to the presence of offset constraints and network closure constraints, not all control parameters can be set independently. Therefore, not all control parameters need to be optimized by GA. Control parameters that are directly optimized by GA are termed decision control parameters. In order to apply GAs in this research, the first step is to identify the decision control parameters and code them into a chromosome. The coding refers to the process of representing each decision control parameter by a string of binary codes and then connecting all binary strings into a long binary string called a chromosome. The detailed discussion of the cycle-based and vehicle-based algorithms is given in the following sub-sections.

5.1.1 Cycle-Based Algorithm

In the cycle-based algorithm, green splits are shown by arrows in Figure 5-1 and dynamic optimal speeds for all independent streets are decision control parameters.

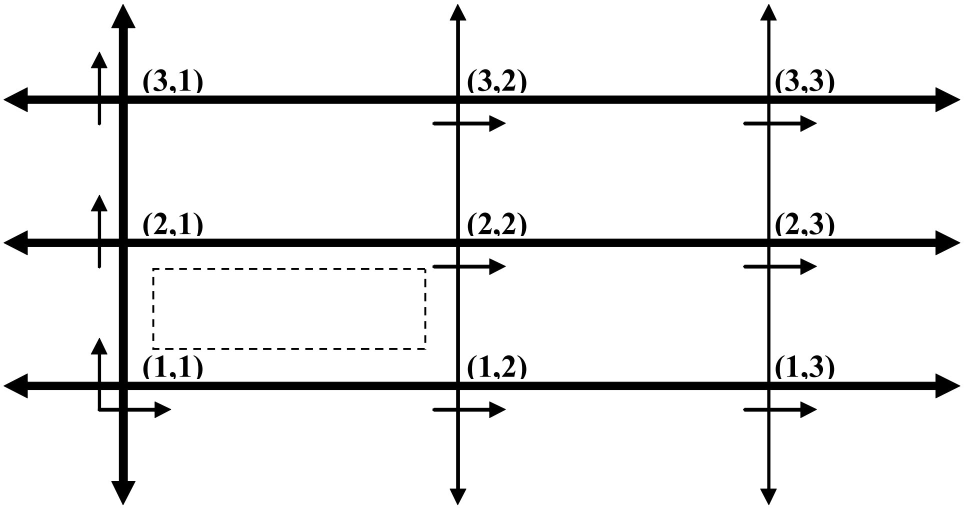


Figure 5-1 A sample grid network (Thick lines indicate independent arterials; small arrows indicate optimized greens)

These decision control parameters are coded into the chromosome and the rest of the control parameters are calculated from corresponding decision control parameters. In order to use the green time efficiently, offsets on all independent links are set to compact offsets. Given that the arterial greens are decision control parameters and are coded into the chromosome, the green time serving the crossroad traffic (*gc*) can therefore be calculated as follows:

$$gc_{(l,m)k} = cycle_{(i,j)k} + off_{((l,m),(i,j))k} + off_{((l,m),(i,j))k+1} - g_{(l,m)k}, \qquad (5\text{-}1)$$

where $(i,j),(l,m) \in N(nx,ny)$, k = 1,2,é n, and $(i,\ j)$ is the immediate upstream signal to (l,m). This equation is a variation of the equation used by Girianna and Benekohal [Girianna and Benekohal 2002].

Computation of the crossroad green splits starts from the independent North-South arterial. In the sample network shown in Figure 3-6, signal (1,1) is used as a reference signal with a known cycle length. Because the cycle length of reference signal (1,1) is already known, the crossroad green times for signals on this North-South arterial that serve eastbound traffic can be calculated by equation (5-1). Subsequently, all signals on this North-South arterial, whose signal lengths are already known, become reference signals for the corresponding East-West arterials. Using this information, the crossroad green time for other signals on East-West arterials that serve northbound traffic can be calculated by equation (5-1).

Since dynamic optimal speed on all independent streets are specified by the GA and the offsets on these streets are set to compact offsets, the offsets of the dependent streets can be calculated using the offsets of independent links. An example is shown by equation (3-6). Dynamic optimal speeds on the dependent streets improve the utilization of green time and/or decrease the number of stops by making locked in offsets on dependent streets as close to the compact offsets as possible. In the cycle-based algorithm, the dynamic optimal speeds for the dependent streets are computed as follows: 1) Set the calculated offset equal to the right hand side of equation (3-1) or (3-2); 2) The solution from step 1) is then compared to the speed constraint shown in equation (3-8). If the solution is larger than the maximum speed constraint, then the desired speed is set to the maximum speed.

5.1.2 Vehicle-Based Algorithm

In the vehicle-based algorithm, the chromosome represents a set of green times of all streets for the entire control period. Note that in the vehicle-based algorithm there are no dependent and independent street designations and speeds are calculated subject to the start of optimized greens at adjacent intersections. Speeds are calculated when vehicles leave the intersection. In order to use green time efficiently, speeds are determined based on the following rules:

1) For the first vehicle that leaves the intersection after the signal turns green, the speed selections are set so that the vehicle reaches the downstream intersection when the downstream queue clears. For example, for signals (*i, j*) and (*l, m*), the speed selection can be calculated by following equation:

$$ds(i,j)_t = \frac{L(i,j)(l,m)}{q(l,m)_t \cdot h}. \tag{5-2}$$

where (*i,j*), (*l,m*) $\in$ *N*(*ny, nx*), *k*=1,2,é *n*, and (*i, j*) is the immediate upstream signal to (*l,m*).

2) For the following vehicles, speed selections are set so that they maintain saturation headway or other headway as specified in the control algorithm. This value may be time or traffic conditions-dependent. These calculated speeds are also compared to the speed constraints. If the solution is larger than the maximum value, then the dynamic optimal speed is set to the maximum speed.

In both cycle-based and vehicle-based DSDS algorithms, not all control parameters are directly optimized by the GA. Some of them are calculated from the variables that are optimized directly by the GA. However, since the GA optimization is based on network-wide measures of effectiveness (MOEs) such as system throughput and number of stops at all intersections, all control parameters are therefore near optimal from a network-wide perspective.

5.2 Constraint Handling: Penalty for Blockages and Spillbacks

In order to get feasible solutions, all constraints mentioned in the previous sections need to be satisfied. Some of those constraints can be satisfied simply by manipulating the binary code in the GA. Such constraints include control parameter constraints. For instance, if the green time constraint is g_{max}=60 and g_{min}=30, the total number of possible green time values must be 60-30=30. This number can be set equal to 2^l and l equal to $\log_2(30)$. As such five bits of binary string should be used for each green variable. Some constraints, e. g., offsets constraint, have been built into the formulating of the control algorithms. Other constraints, however, need to be checked and evaluated during the search process. These constraints are referred to as implicit constraints. A

GA can convert a constrained optimization problem into an unconstrained problem using penalties for the implicit constraints violations [Dasgupta 1997, Goldberg 1989]. For instance, in this research, the queue and defacto red constraints are implicit constraints. These two constraints are checked for each pair of signals and if the queue constraints are violated, a queue penalty is applied against the fitness function. The queue penalties are formulated as

$$P\_q(i,j)(l,m)k = \mu_1 \cdot \left(q(l,m)k - q\max(l,m)\right). \tag{5-3}$$

For the defacto red constraint violation, the penalty is formulated as

$$P\_r(i,j)(l,m)k = \mu_2 \cdot \left(g(i,j)k - g(l,m)k + off(i,j)(l,m)k + L(i,j)(l,m)/vt\right)/h, \tag{5-4}$$

where μ_1 and μ_2 are penalty coefficients and the values are decided based on control strategies.

5.3 Fitness Value Calculation

Fitness values need to be returned to GA solver and are used to evaluate the goodness of solutions. In this research, the fitness value is calculated by using the objective function value minus the queue and defecto red penalties of all intersection pairs. Because fitness evaluation function cannot return a negative value to the GA solver, the GA fitness function is formulated as:

$$fitness = T(c) - \sum P\_q - \sum P\_r\,. \tag{5-5}$$

where $T(c)$ is the fitness before the penalty shown in equation (3-15). The GA package *Gallops* [Goodman 1996] is used in this research to optimize the DSDS controls. Results are presented in the next section.

5.4 Experiment Setup

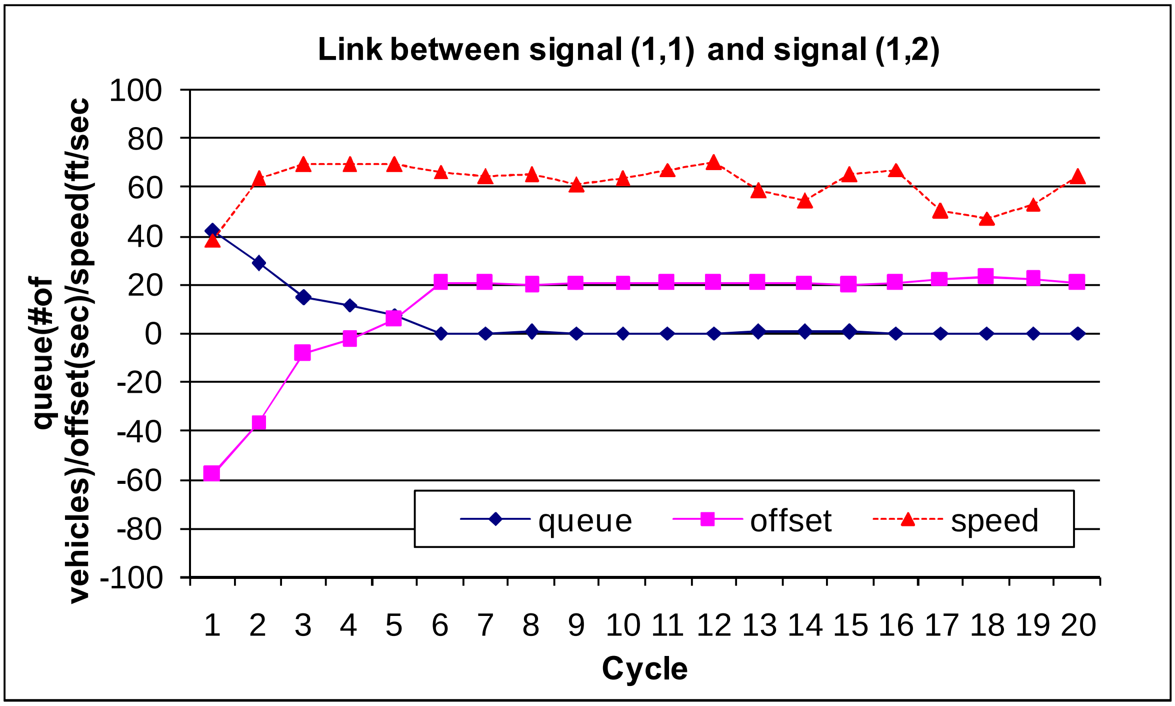
To assess the feasibility and demonstrate the advantages of a control using dynamic speed with dynamic signals, both a combined dynamic-signal dynamic-speed (DSDS) control and a dynamic-signal fixed-speed (DSFS) control are applied to two different sizes of over-saturated grid networks, N(2, 2) and N(5, 4). Link lengths were 800 ft. For the major flow directions (eastbound and northbound), constant and higher than capacity demand is assumed at the entrance intersections. Also vehicles were assumed to enter the system at a saturation flow rate. Initial queues were randomly chosen between 41 and 46 veh/lane (maximum is 53 veh/lane). For the secondary directions, lower demands were assumed and vehicles entered the system at a lower than saturation flow rate. Initial queues for secondary directions were set randomly between 5 and 10 vehicles. Minimum and maximum green times were set between 30 and 90 seconds, respectively. Speeds range between 15 mile/hr and 50 miles/hr for the combined control and 40 mile/hr for the fixed speed control. Twenty cycles were simulated for the N(2,2) network and 40 cycles for the N(5,4). The GA solver developed by Goodman [Goodman 1996] was employed with a population of 1000 and 1000 generations. All signals were assumed to have two phases and turning movements were ignored.

For the independent links, the offsets were set to compact offsets. For the dependent links, dynamic speeds were adjusted such that locked-in offsets were as close to compact offsets as possible. The GA optimizes the DSDS control parameters to maximize throughput and number of stops on the major directions. Constraints were applied to queues on both major and secondary direction links to prevent spillbacks. Once control solutions were obtained, traffic operations under near optimal control solutions were simulated by using the macroscopic models developed in chapter 3. Results are discussed in the next section.

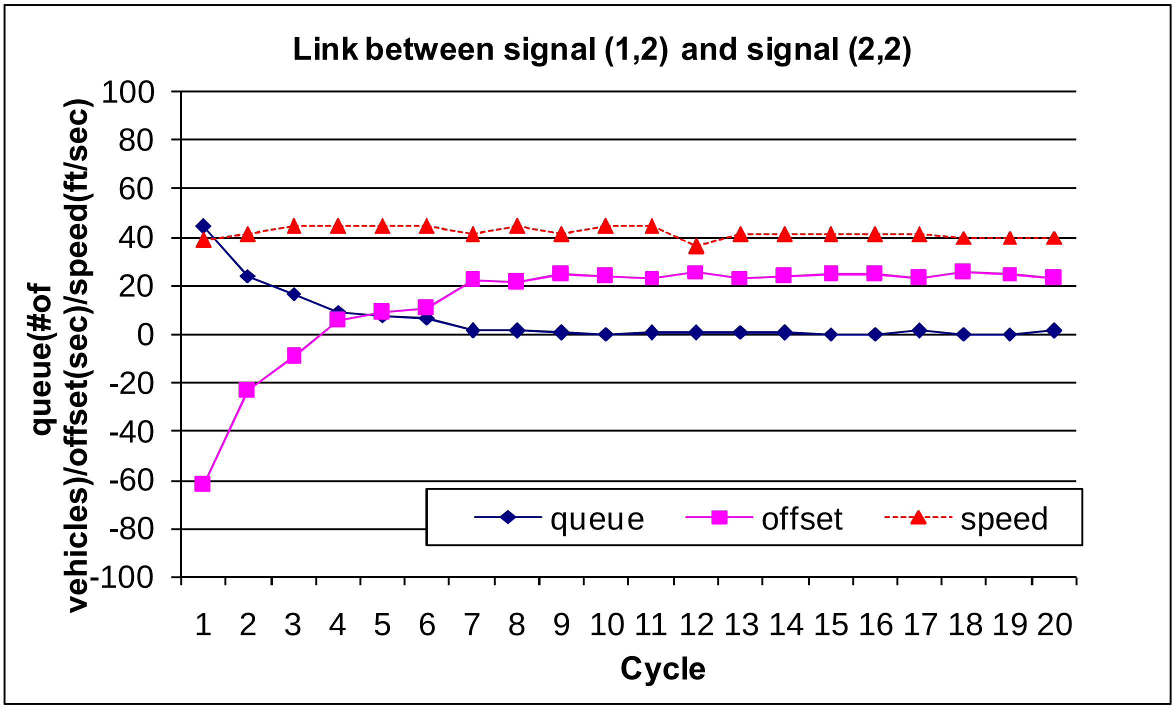
5.5 Results and Discussions

5.5.1 Interrelationship between Queues, Offsets, and Speeds

Figures 5-2 and 5-3 for the small and large network, respectively, illustrate the interrelationships between queues, offsets and speed on a typical dependent and independent link under DSDS control.

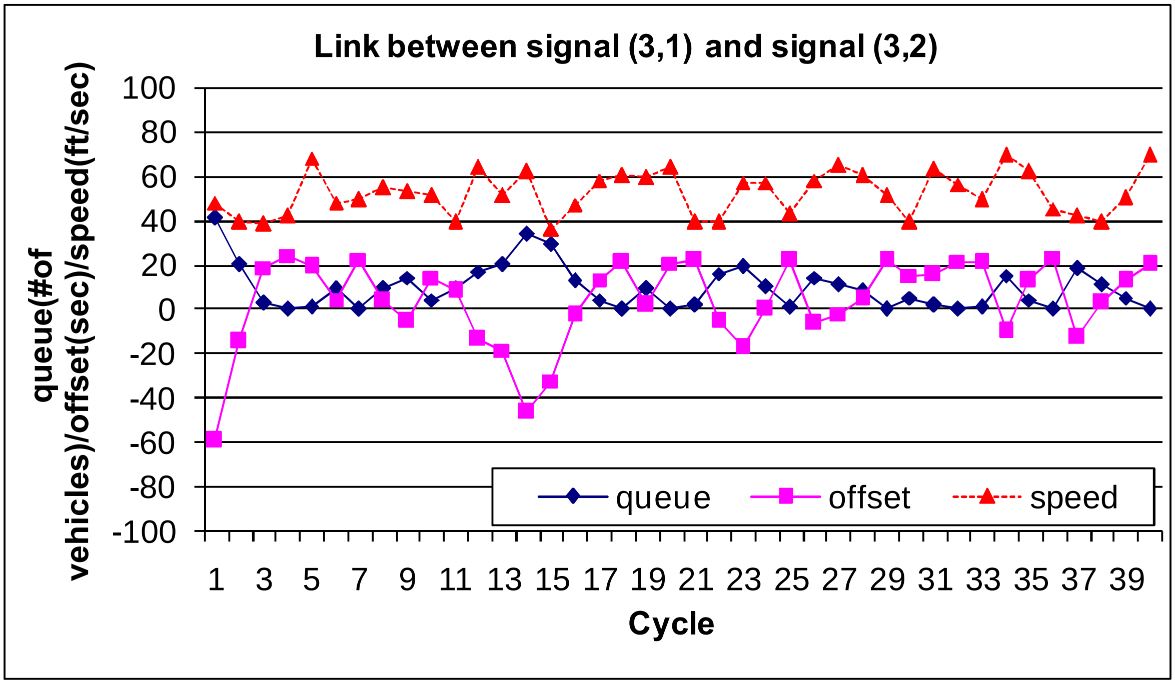


(a) a typical independent link

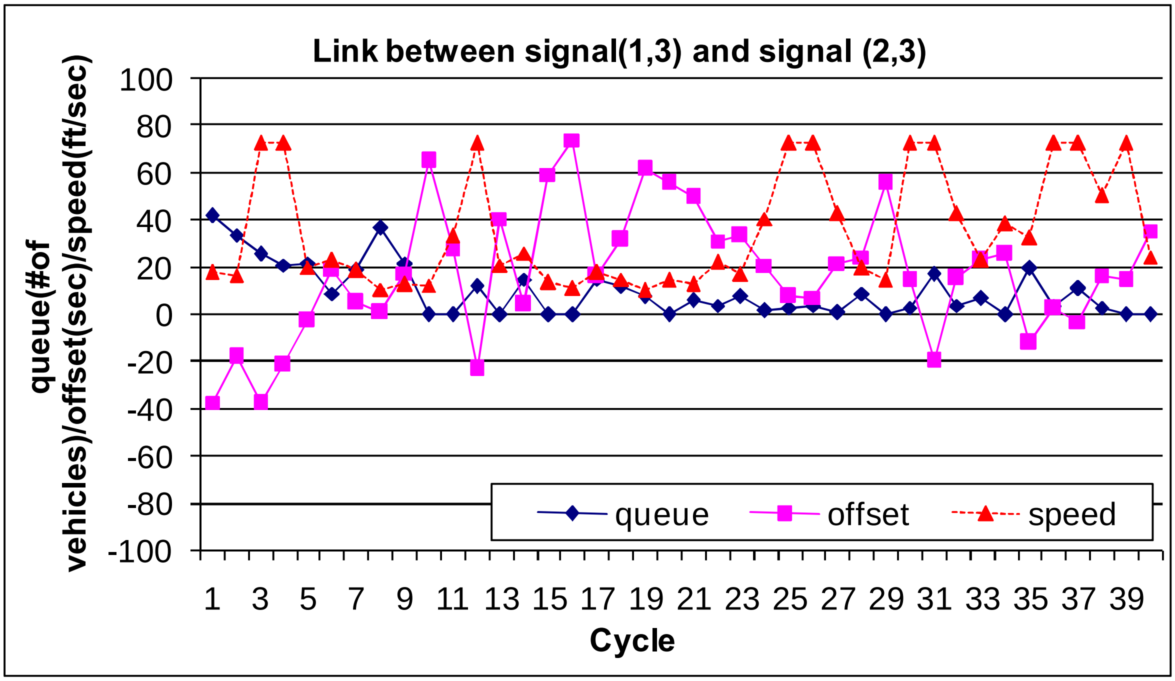


(b) a typical dependent link

Figure 5-2 Interrelationship between queue, offset, speed in N(2,2)



(a) a typical independent link



(b) a typical dependent link

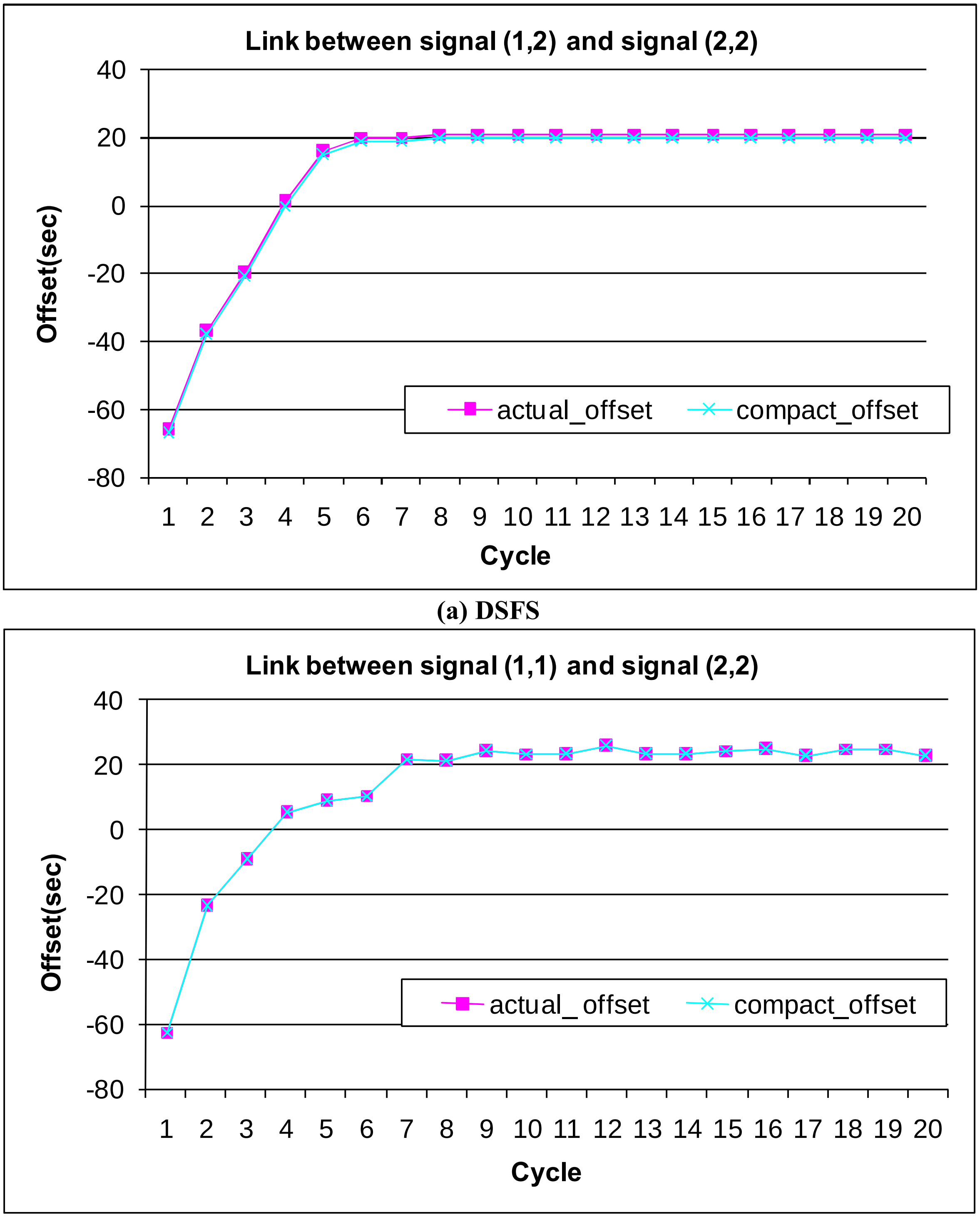
Figure 5-3 Interrelationship between queue, offset, speed in N(5,4)

In general, it is clear that these three parameters are related to each other and this is true for both dependent and independent links. The queues for these and all other links as well as all cycles did not exceed the maximum queue length limit of 53 vehicles. Furthermore, queue lengths decreased with time on all links. This demonstrates that the algorithm has the ability to dissipate queues and prevent spillback. Offsets at the earlier stages were set to negative values which is a desirable pattern to enable queue dissipation. After the queues decreased enough, or dissipated completely, positive offsets were used for normal progression. Speed variation was not significant in the small network, and this is expected as the dynamic signals alone can handle a simple oversaturated network. For the larger network, the operations were more complex and the dynamic speed had a more significant role to play, hence the wider variation. While the speed variation was smaller on the independent links, the association between offsets and queue lengths was stronger and that remained true over time. For the dependent links, the larger speed variation was associated with a not-as-strong-association between offset and queue length. The explanation of these trends is as follows: higher variation in speed on the dependent links is due to the fact that speed is adjusted up or down to lessen the negative impact of the unfavorable ñlocked-inò offsets on these links. For the independent links, offsets are optimal and hence there is no need for as much intervention via speed change. Notably, when speeds on the independent links vary, it is due to the less-than-favorable offsets on the dependent links. In other words, speeds and offsets on the independent links could change in response to conditions on the dependent links if the offsets and speeds on the dependent links, on their own, could not cope with the unfavorable traffic conditions. These results demonstrate how queues, speeds, and offsets on both dependent and independent links are working together in concert to manage evolving traffic conditions and bring about favorable traffic operations.

5.5.2 Impact of Dynamic Speed on Offsets

Figures 5-4 and 5-5 show the difference between actual (selected) offsets and the compact offsets on typical dependent links in the small and large networks, respectively. Parts (a) of these two graphs show the difference under dynamic-signal fixed-speed limit (DSFS) control, while parts (b) show the difference under combined dynamic-signal dynamic-speed (DSDS) control.

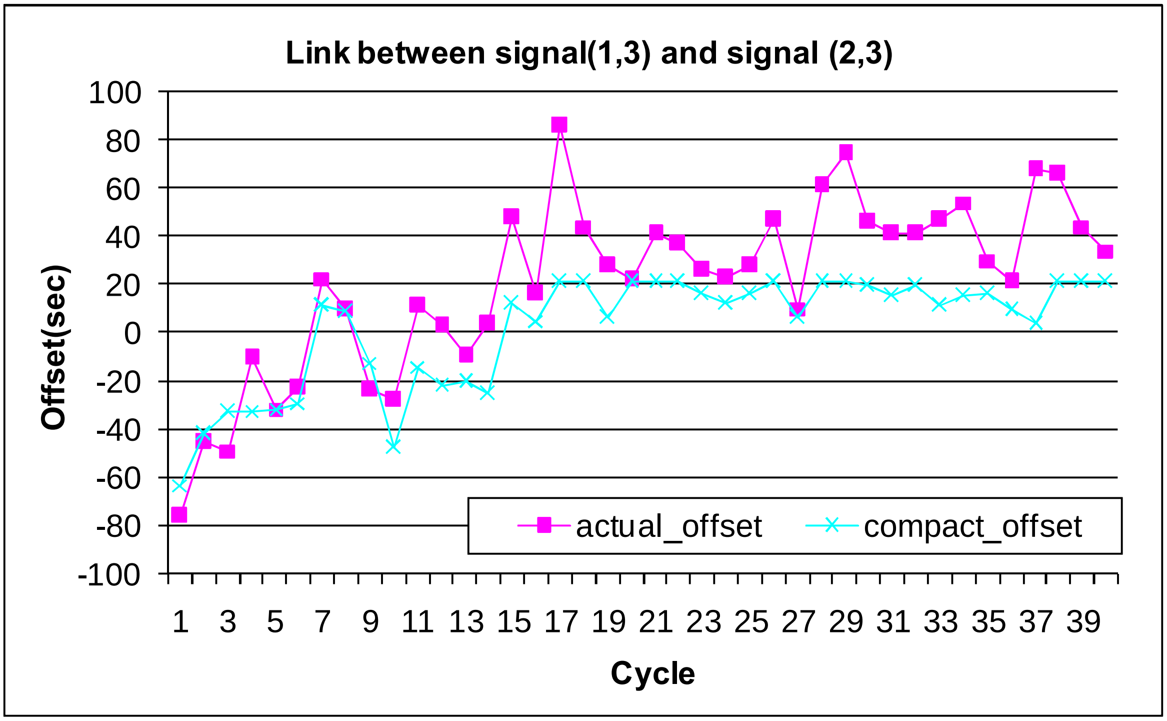
It is clear that for both size networks the offset values were much closer to the compact offset value under DSDS control than under DSFS control, although the difference was very small in the small network. The advantage of control with dynamic speed was much clearer in the larger network, thus suggesting that control with dynamic speed should create more difference in larger systems. The closeness of the actual to the compact offsets was realized not only by adjusting the speed on the dependent links, but also by optimally adjusting speeds on the independent links. The logic here is that when adjusting speeds on the dependent links is not enough to realize a compact offset on these links, speed on the independent links can be adjusted to change the offset on the independent links and therefore change the offset on the dependent links indirectly. Therefore, although the offsets between signals along the independent arterials were set to be compact under both DSFS and DSDS controls, the offsets set by the combined signal-speed control (DSDS) were more efficient from a system-wide perspective.



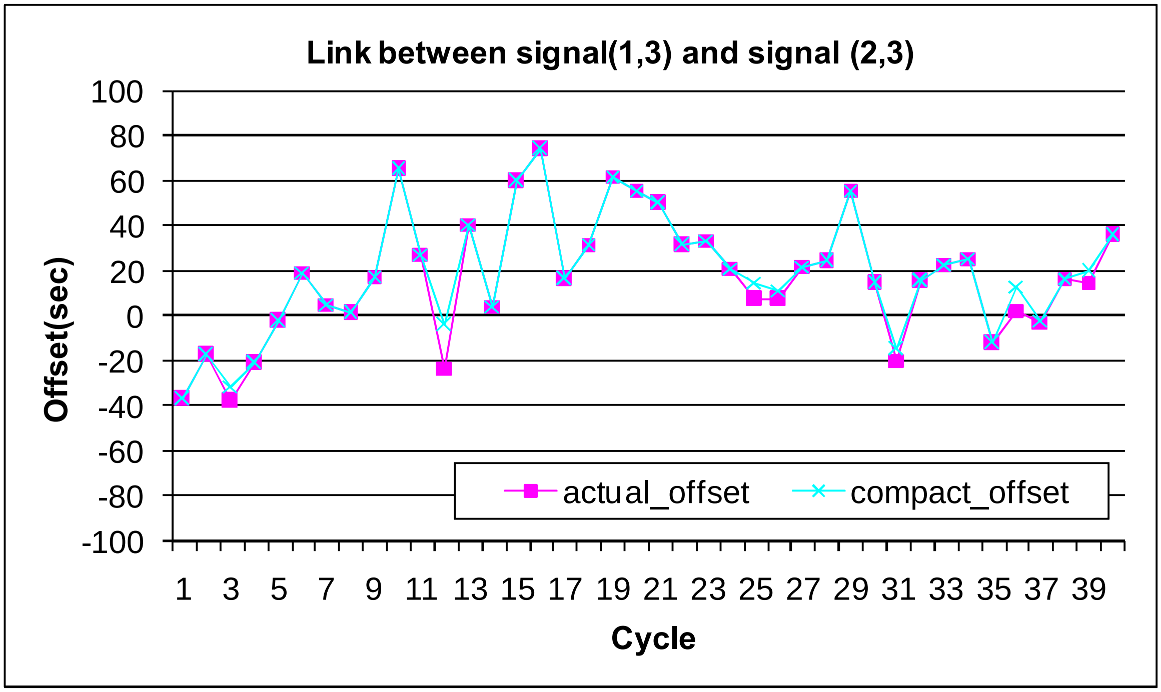
(a) DSFS

(b) DSDS

Figure 5-4 Actual and compact offsets for a typical dependent link in N(2,2)



(a) DSFS

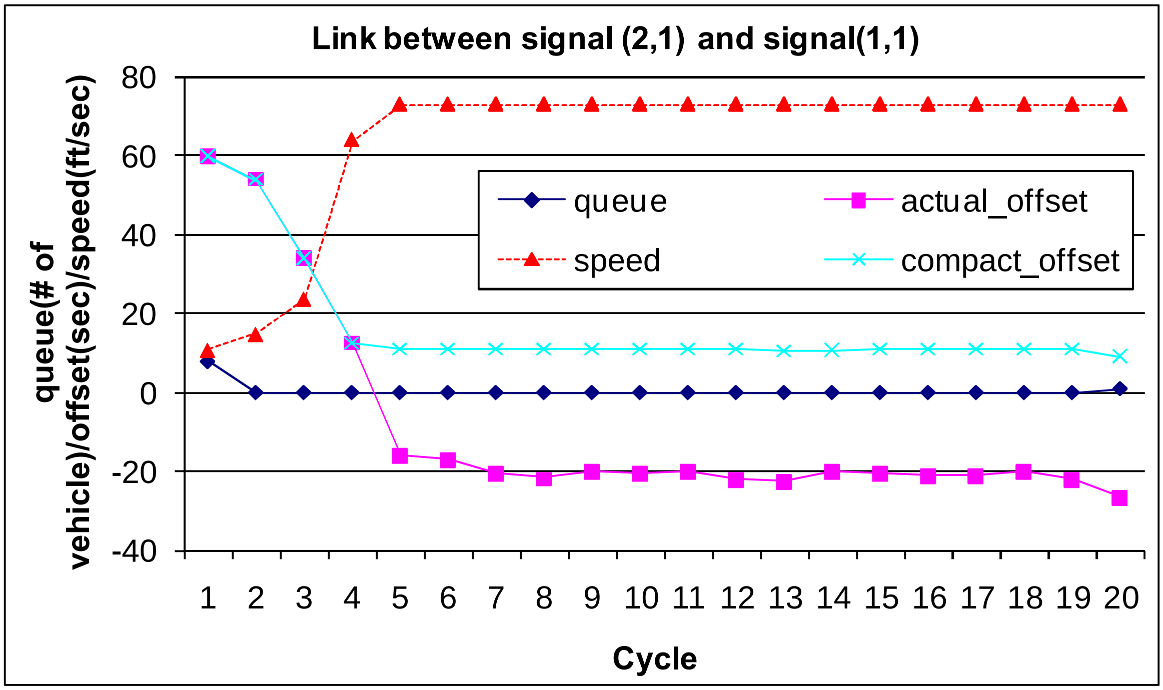


(b) DSDS

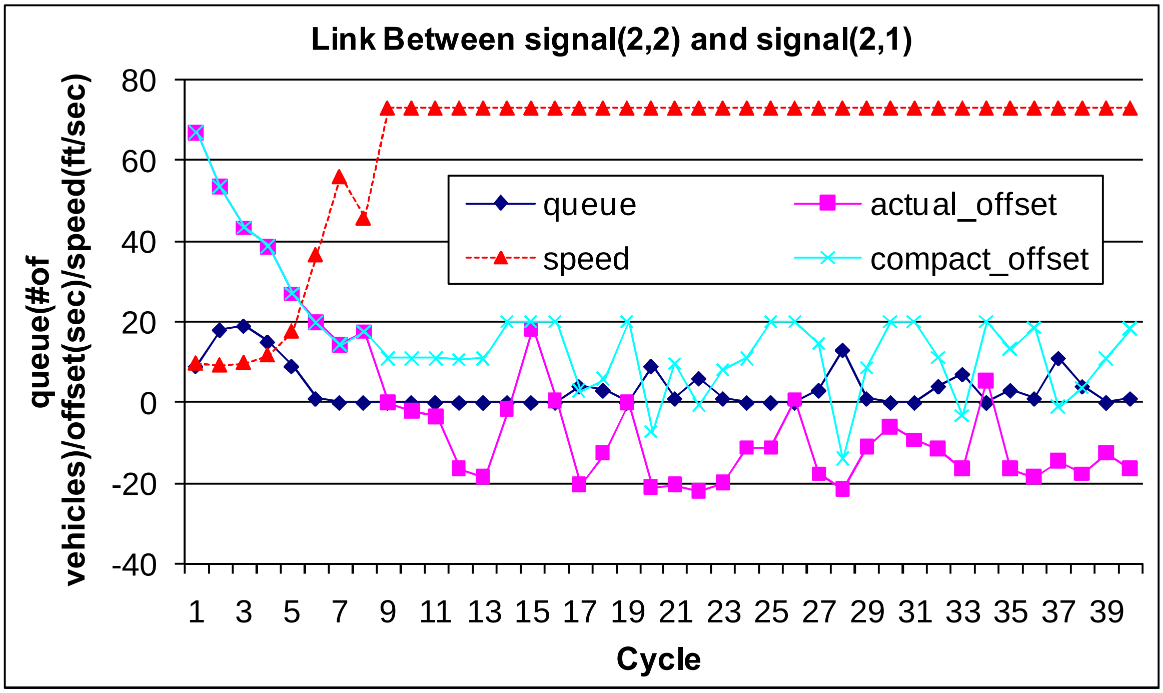
Figure 5-5 Actual and compact offsets: typical dependent link in N(5,4).

5.5.3 Operation on Secondary Direction

The previous discussion focused on operations with the major directions. Traffic in the secondary low-demand directions operated only in the shadow of the major direction traffic in the following ways: it shared the same green time as the major directions, offsets were the locked-in offsets, and speeds were adjusted based on traffic conditions on these links and the locked-in signal timing parameters. Queue constraints were applied to the secondary directions to prevent spillbacks. In effect, optimizing operations in the major directions determined the values of all of the control parameters. Because of the secondary directionsô lower demand, this control arrangement was sufficient to ensure smooth operations. Figures 5-6(a) and 5-6(b) are for operations on typical secondary links in the small and large networks, respectively. The figures demonstrate the changes in queue, offset, compact offset, and speed over time.



(a) typical secondary direction link, N(2,2)



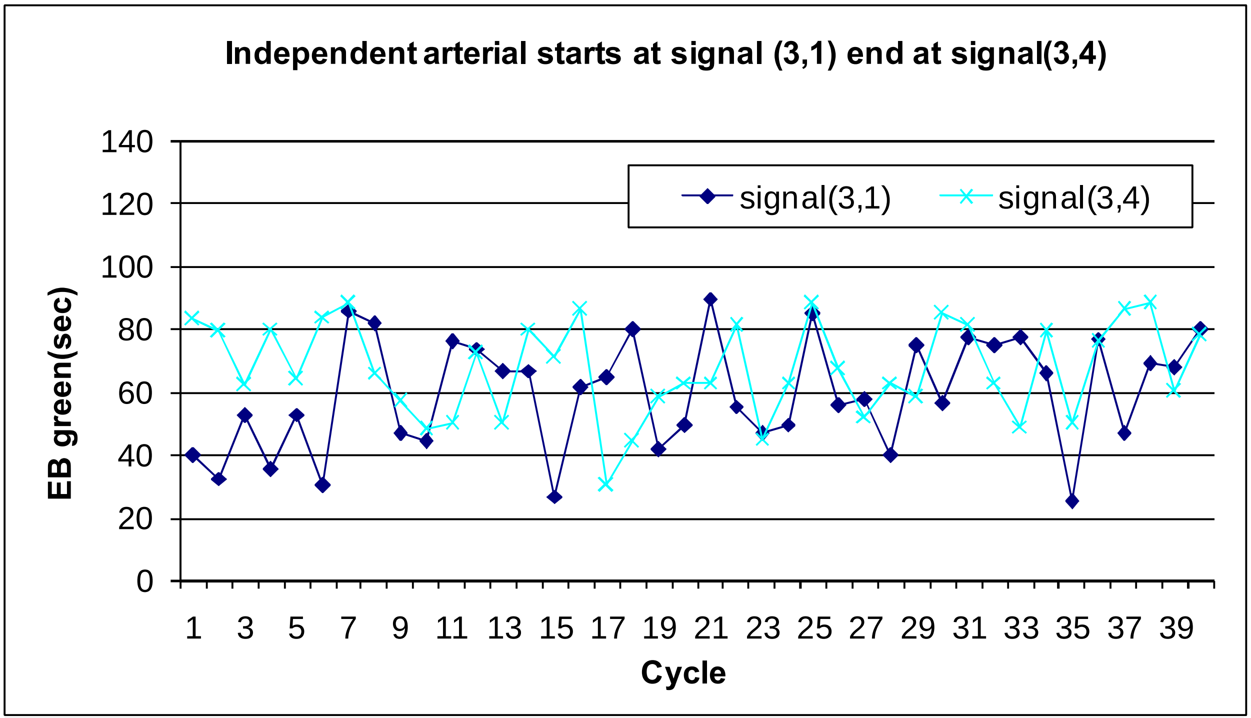
(b) typical secondary direction link, N(5,4)

Figure 5-6 Operation on typical secondary direction links

Since MOEs for the secondary direction were not directly optimized, offsets on secondary directions are not all compact offsets. Wasted green can be seen on these secondary approaches. For both networks, there is a point beyond which the compact and actual offsets diverge. This is the point where the actual offset could no longer be kept at the compact offset because further speed adjustments were not possible as the speeds hit the maximum allowable value. This happened earlier in the smaller network (cycle 5) than the larger network (cycle 9). At the beginning of the control period for the secondary link in the larger network, while the major direction was discharging its own queue, the queue on the secondary direction increased a little but was cleared shortly after that. This is because sufficient green is given by the DSDS control based on demand on the major direction; this green was more than enough for the queue on the secondary direction. If turning movements are present, this wasted green can be used to process turning movements on major directions. Speeds on the secondary directions were adjusted to minimize the impact of the undesired offsets. At the later stage of the control period, speed hits the maximum set by the speed constraint, and further improvement was not possible by only adjusting speed on the secondary direction. Consider the case of major direction dependent links (NB links shown in Figure 5-2 (b) and Figure 5-3 (b)). Further improvement can be expected by adjusting signal timing and speed parameters on major directions.

5.5.4 Green Time Allocation

Figure 5-7 illustrates the variation of green time of the first and last signals on one independent and one dependent arterial over the study period in the larger network. In both cases, the green time increased in the downstream direction at the early stage. This is desirable for oversaturated traffic conditions as it helps clear the queues. After the queues were cleared, the lengths of cycles on both arterials started to get closer to each other, although the differences on the dependent arterial were larger. It is well known that a common cycle length is necessary for normal progression. In other words, the control on both independent and dependent arterials first managed traffic in a way suitable to clear the queues and once traffic conditions became more suited for normal forward progression, the control adapted to those conditions. It is remarkable that the algorithm performed in this manner over an extended period of time of 40 cycles without causing any spillbacks or grid locks although it started with a network full of traffic.



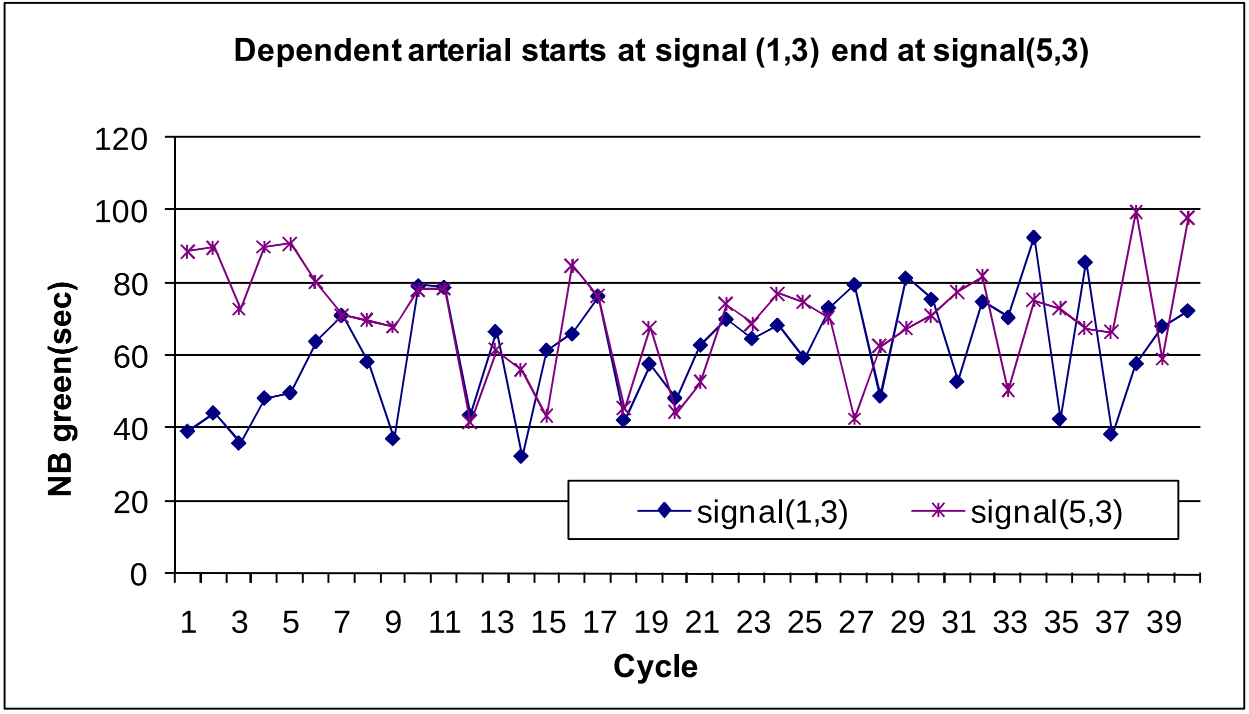


Figure 5-7 Green time on independent arterial $A_{(3,1),(3,4)}$ **and dependent arterial** $A_{(1,3),(5,3)}$

5.5.5 Network-wide MOEs

As shown in previous sections, the developed DSDS control algorithm worked well even in over-saturated traffic conditions. It realized one of the most important objectives for the control of over-saturated signal networks which is to effectively prevent queue spillback and gridlock. It also provides a lot more flexibility compared to fixed speed signal control systems.

System-wide measures of effectiveness including throughput, number of stops, average travel speed, and average delay were compared between DSFS and DSDS control algorithms. The simulation results indicate that with comparable throughput, DSDS control yields 13.1 vehicle-hours per hour less delay and 635 fewer stops per hour in the larger network N(5,4). For the small network, there were 2.7 vehicle-hours/hr less delay and 64 fewer stops per hour. These reductions in delay and stops will amount to significant values over time. Higher average travel speeds were observed in both networks under DSDS control. The improvement in a larger network is more significant. This is expected because there are more dependent links in the larger networks and it is these links that are the ones that DSDS reflects direct benefits.

5.6 Conclusions

DSDS control for N(2,2) and N(5,4) were solved using simple GA in this chapter. The cycle-based DSDS algorithms were applied to major directions with high demand and vehicle-based algorithms were applied to secondary directions with lower demand. The experiments were carried out on simplified hypothetical networks. The traffic operation in traffic networks under DSDS control were analyzed to evaluate the feasibility and effectiveness of developed DSDS controls. System wide MOEs were also compared to the same MOEs in traffic networks under DSFS control. The results demonstrate that dynamic speed control and dynamic signal control can be effectively integrated to control signalized networks. It provides an alternative solution and more flexible approach to control signalized networks.

Based on the simulation results, the DSDS control provides near-optimal signal settings and speed selections that were responsive to traffic conditions. Not only spillbacks and gridlocks were prevented over extended control times, queue lengths were reduced over time as well. Green time saved in secondary direction will turn into more improvement of operations such as throughput and delay when turning movements are considered. As the networkôs geometry and traffic operations grow in complexity, DSDS will likely have a larger impact.

The developed DSDS control also coordinates signal control and speed selection on all links over extended control time and obtains network-wide robust performance even under unfavorable traffic conditions.

Chapter 6. Minimizing Speed Variation and Speed Noise

In this chapter, the ability of the dynamic signal and dynamic speed (DSDS) model to minimize speed variation and speed noise is analyzed by using microscopic simulator VISSIM. First, an introduction of speed noise and speed variation is provided. Then the speed variation and speed noise are discussed for signalized traffic systems. After that, the setup of simulation with VISSIM is described and results are analyzed. The conclusions are summarized at the end.

6.1 Introduction

Vehicle speed noise and speed variation between vehicles have direct impact on road safety, fuel consumption, and emissions. Vehicle speed noise refers to the deviation of vehicle speed from its desired travel speed and vehicle variation refers to the speed difference between vehicles in a platoon. Thorton and Lyles [Thorton and Lyles 1996] had concluded that a major factor leading to an accident is not speed itself, but the variation of speed. Oh et al. [Oh et al. 2005] also proved that speed variation is the most distinguishing parameter between normal traffic conditions and traffic conditions just prior to accidents occurring.

In previous studies about optimal speed profiles [Schwarzkopf and Leipnik 1977], it has been proven that fuel consumption is approximately minimized by operating at a constant speed. The magnitude of fuel consumption increases as a result of any deviation from the constant speed [Chang et al. 2005]. Hence, in order to operate safer and more fuel conservative signalized roadway systems, efforts should be made to develop means to minimize speed noises and speed variations.

One possible approach is to use the dynamic speed approach to intelligently adjust the optimal speed for drivers according to changing road, weather, and traffic conditions. The DSDS control developed in this research can effectively combine dynamic speed control with dynamic signal control to build a more flexible control system. In this chapter, the advantages and capabilities of the developed DSDS control are explored in reducing speed noises and speed variations by analyzing vehicle speed profiles obtained from microscopic simulations.

6.2 Speed Noise and Speed Variation in Signalized Networks

In signalized networks, vehicle speeds are impacted by traffic signal operations, speed limit, and traffic flow conditions. For example, Figure 6-1 shows a typical vehicle speed profile along with a signalized arterial with fixed speed control. The profile during time period t1 is very typical when a downstream queue is encountered by the subjected vehicle. The vehicle slowed down first. After vehicles in the downstream queue accelerated to the desired speed and the queue dissipated, the vehicle accelerated back to its desired speed. The profile during time period t2 shows the situation when the vehicle encounters a red signal. The vehicle slowed down, stopped for a while, and then accelerated back to the desired speed.

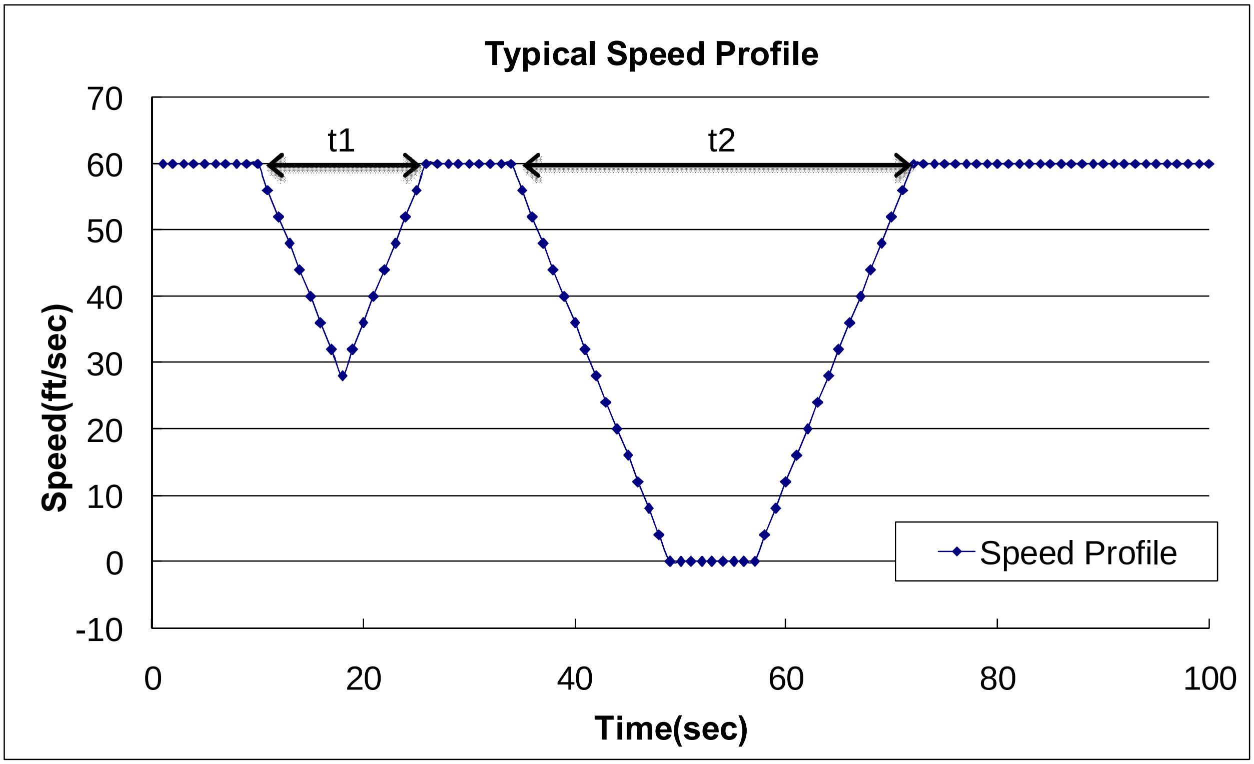


Figure 6-1: Typical Speed profile of a vehicle in a signalized network.

Under both conditions reflected in Figure 6-1, the vehicle speed deviated from its desired speed and thus speed noises occurred and, by implication, fuel consumption also increased. However these two conditions have different impacts on the following vehicles: the first condition will cause the following vehicle to decelerate and then accelerate while the second condition not only causes the following vehicle to decelerate and then accelerate, but also causes a more pronounced stopping and start-up shockwaves to travel upstream into the traffic platoon. The longer t1 and t2 are, the higher the number of vehicles that are impacted. Also, speed variation is higher during the acceleration and deceleration period because more interaction exists between vehicles. As such, accidents are more likely to occur during this time period.

Using dynamic speed control with dynamic signal control in signalized networks can help minimize undesirable vehicle speed noises and speed variations by guiding drivers to travel with optimal traveling speed, especially in over-saturated conditions where stop-and-go driving patterns happen more frequently.

6.3 Experiment Setup

Microscopic simulation software VISSIM [PTV, 2005] is used to obtain the speed profiles for vehicles under both dynamic-signal dynamic-speed (DSDS) control and a dynamic-signal fixed-speed (DSFS) control. First, DSDS control and DSFS control are applied to an over-saturated grid network N(5, 4). In the simulation, 40 cycles were simulated. Link lengths were 800 ft. For the major flow directions (eastbound and northbound), constant and higher than capacity demand is assumed at the entrance intersections, and initial queues were randomly chosen between 41 and 46 veh/lane (maximum is 53 veh/lane). For secondary directions, lower demands were assumed and initial queues were set randomly between 5 and 10 vehicles. Minimum and maximum green times were set to 30 and 90 seconds, respectively. Speeds ranged between 15 mile/hr and 50 miles/hr for the combined DSDS control and 40 mile/hr for the fixed speed control. The genetic algorithm (GA) solver developed by Goodman [Goodman 1996] was employed with a population of 1000 and 1000 generations to ensure a near-optimal solution will be reached. All signals in the network had two phases and turning movements were ignored.

Once near optimal control solutions for DSDS and DSFS were obtained, signal timing and optimal speed information were built into a VISSIM representation with the same physical features. Speed profiles of vehicles were obtained and compared to analyze the advantages of dynamic speed on speed noises and speed variations. Simulation results are discussed in the next section.

6.4 Results And Discussion

6.4.1 Speed Profile of Individual Vehicles

Figure 6-2 shows the speed profile of the first vehicle arriving at a typical link in a typical control cycle under both control scenarios.

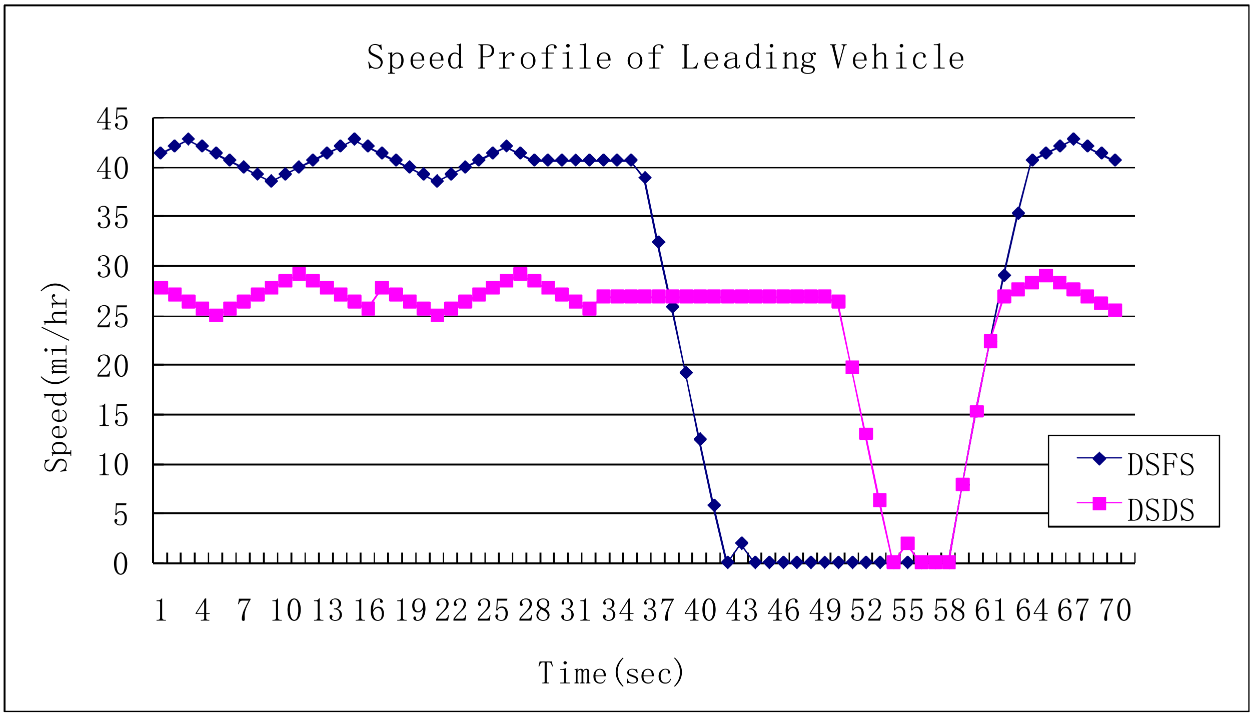


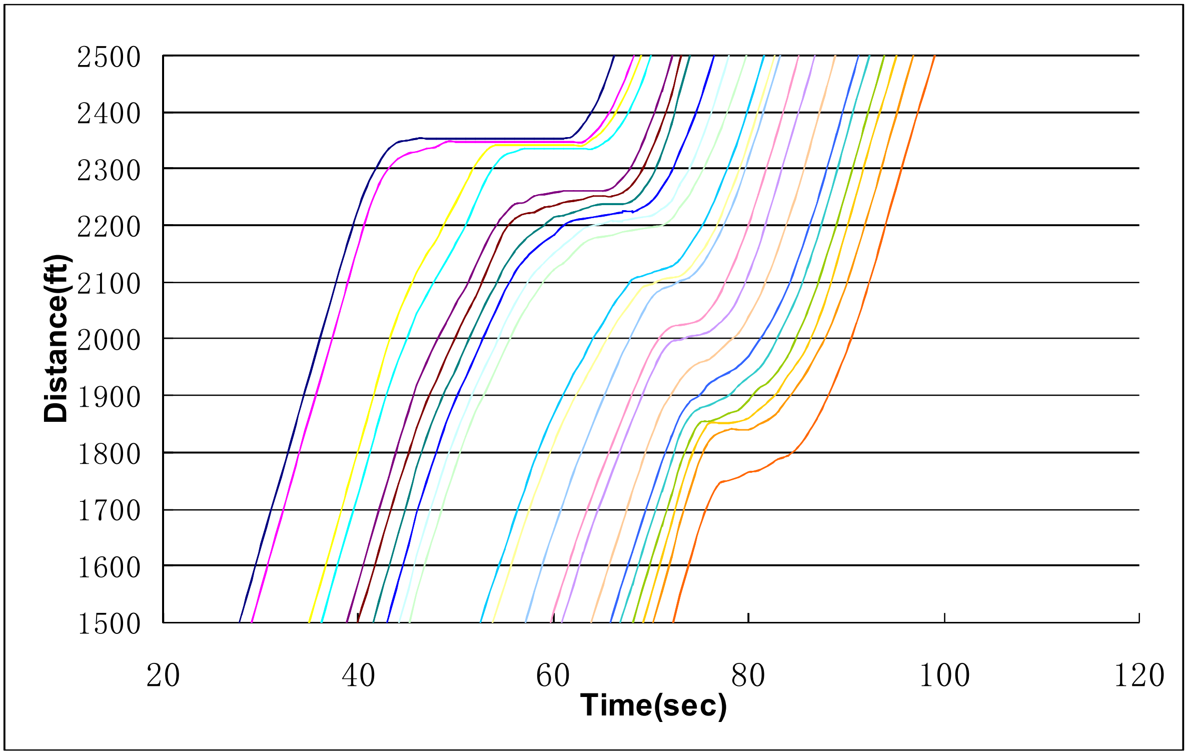
Figure 6-2: Speed profile of leading vehicle under DSDS and DSFS control

Under DSDS control, vehicles are given guidance to have lower desired speed compared to the DSFS control. The effect was that although both control schemes lead to almost the same average speed, there was less time when the actual speed deviates from the desired speed. Because of the minimum and maximum speed constraints, and the randomness inherent in driver behavior which is explicitly accounted for in VISSIM but not accounted for in the macroscopic simulation models that are used in obtaining the near-optimal solutions, speed noises were not completely eliminated. However, the duration of time when actual speed deviated from the desired speed was significantly shorter under the DSDS control.

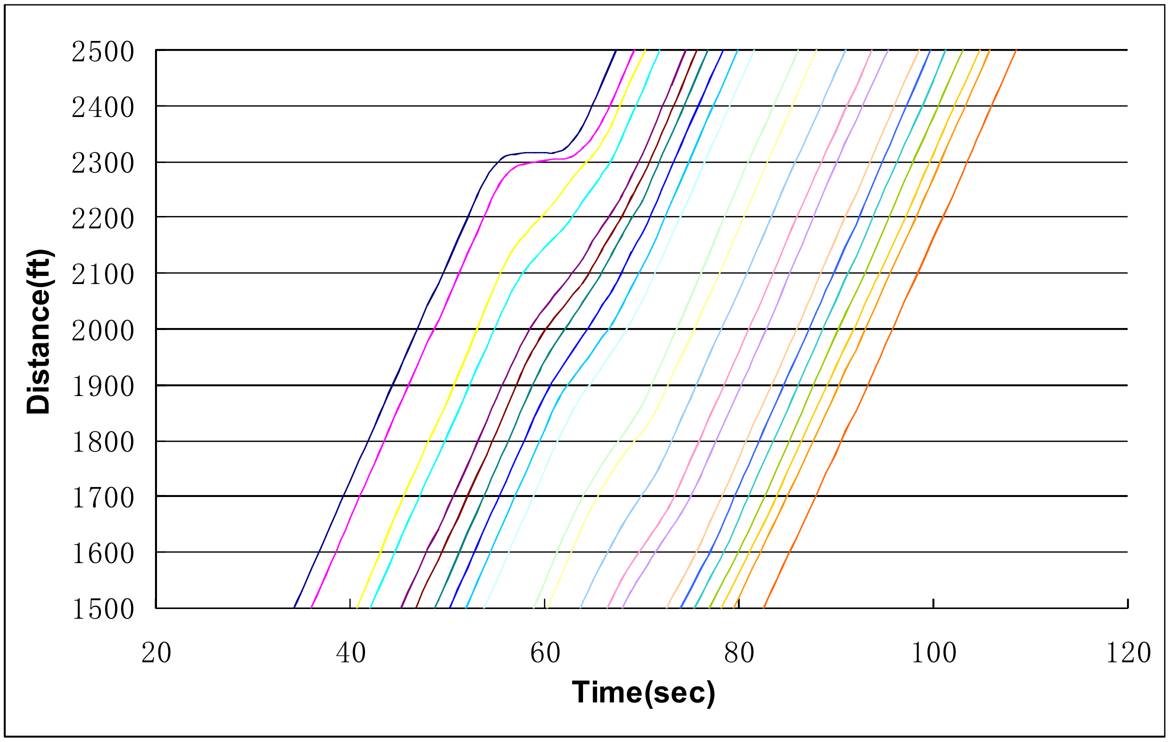
6.4.2 Speed Trajectory of Vehicles in a Platoon

Figure 6-3 shows the speed profiles of a platoon of vehicles arriving at a link during one typical cycle on a typical link. Part (a) is the speed profile of vehicles in a network under DSFS control and part (b) is for DSDS control.

As noted in the previous section, the duration of speed deviation from the desired speed is shorter under the DSDS control. Based on Figure 6-3, it is clear that vehicles impacted by the stopping shockwave were fewer, and that the resulting speed noises for the following vehicles were less. It was also found that speed variation was smaller in a traffic stream under DSDS control. This is due to shorter acceleration and deceleration periods and less vehicles involved. All of these conditions imply that traffic flow under DSDS control is safer and more fuel conservative.



(a) DSFS control



(b) DSDS control

Figure 6-3: Vehicle Trajectories

6.5 Conclusions

This Chapter evaluated the potential capability of DSDS control on reducing speed noises and speed variation in signalized networks. The vehicle speed profiles obtained from a microscopic simulation environment were analyzed and compared to the results acquired from DSFS control. The results have shown that DSDS control is able to achieve smoother traffic flow by reducing speed noises and speed variations. Such results indicate that dynamic speed control is an effective and necessary addition to the signal control system for signalized networks. DSDS control provides a more flexible and effective alternative to design safer and more fuel conservative signalized network systems.

Chapter 7. Impacts of Driver Compliance on Effectiveness of DSDS Control

It is a major traffic safety concern in all roadway environments that drivers do not comply with traffic control devices (speed limits, traffic signals, traffic signs, etc.). Driversô compliance to speed limits is of particular interest in signalized networks, in which slightly excessive speeds could have a profound impact not only on safety, but also on the effectiveness of signal timing control strategies. In this chapter, the impacts of driversô compliance on the effectiveness of dynamic speed and dynamic signal (DSDS) control developed in this research were analyzed and evaluated.

7.1 Introduction

According to Tignor and Warren, the compliance with speed limits is poor in general. On average, 7 out of 10 motorists exceed the posted speed in urban areas. Compliance ranges from 3 to 99 percent. Compliance tends to be worse on low-speed roads, but better on roads with prima facie limits or where the speed limit is based on more recent engineering studies. Here ñbetterò does not mean good compliance. It is found that less than 10 percent of the sites had more than 50 percent of obedience with the posted speed [Tignor and Warren 1991].

Both public education and enforcement have been used in order to achieve better driver compliance and consequently improve safety and efficiency at intersections. However, some of the improved enforcement methods, such as using of speed-cameras and increased police surveillance, have been shown to have a very limited effect [Comte et al. 1997]. According to the study conducted by the Federal Highway Administration [FHWA, 1997], the majority of drivers will not comply with a speed limit that they perceive as unreasonable for prevailing conditions. On the other hand, properly established speed limits foster voluntary compliance and separate the occasional high-risk driver from the vast majority of drivers [Tignor and Warren 1991]. Therefore, the best way to promote driversô compliance is to treat it at the source, i.e. at the vehicle or road-user level. The DSDS control algorithms developed in this research provide drivers with optimal speed for prevailing traffic conditions and signal timings, hence it is reasonable to expect that better driver compliance will be achieved under such optimal speed guidance.

Even though the dynamic optimal speed is the optimal speed for prevailing traffic conditions and signal timings, some drivers may still drive at higher or lower speeds. When driver compliance is lower than 100%, the DSDS algorithms are not likely to have the same safety and operational enhancing effects as described in previous chapters. The analysis in this chapter provides information to help understand how those unwanted driver behaviors will impact the net safety and efficiency gain of the DSDS algorithms.

7.2 Experiment Setup

In order to estimate the impact of driver compliance on the safety and efficiency for the developed DSDS control algorithms, microscopic simulation was conducted and results were studied using the VISSIM package. The larger traffic network N(5,4), which has the same physical features as the one used in Chapter 6, was built in VISSIM. Near optimal DSDS control solutions for this larger grid network N(5,4) were achieved by using the DSDS algorithm presented in Chapter 3 and the GA solver developed by Goodman[Goodman 1996]. Signal timing and optimal speed parameters were then built into the VISSIM network. Throughput, number of stops, and delay were used as measures of effectiveness (MOEs) to measure the performance of the DSDS control at different levels of driver compliance. The effect of speeding and slow driving was evaluated because they both have negative impacts on the performance of DSDS. Different levels of driver compliance(10% speeding, 25% speeding, 50% speeding, 75% speeding, 90% speeding, 100% speeding, 10% slow driving, 25% slow driving, 50% slow driving, 75% slow driving, 90% slow driving, and 100% slow driving) were simulated in 12 cases to analyze their impacts. Speeding drivers were assumed to have a desired speed at 10miles/hr higher than optimal speed and slow driving drivers a desired speed at 10miles/hr lower than the optimal speed. For example, 10% speeding represents the case when 10% of the drivers choose a desired speed at 10mile/hr higher than optimal speed and the other 90% choose to follow optimal speed guidance. One hundred percent speeding represents the situation when all drivers ignore the dynamic optimal speed guidance and choose to drive at 10miles/hr higher than the optimal speed. Each case was simulated with 5 different random seeds. The MOEs at different levels of driver compliances were recorded and the results were compared to the case in which all drivers follow the dynamic optimal speed guidance. Based on all of these simulation results, the impact of driver compliance on the performance of DSDS control algorithms was assessed. The results and discussions are presented in the next section.

7.3 Results and Discussion

Figure 7-1 shows the change in throughput of a typical link in a N(5,4) network when the level of driver compliance changes. Based on the results, speeding had very little negative impact on throughput. On the other hand, the throughput decreased when the percentage of slow driving drivers increased. However, the throughput reducing effect was smaller when the percentage of slow driving drivers increased, such effects almost diminished when the percentage of slow driving drivers reached somewhere between 50% and 75% .

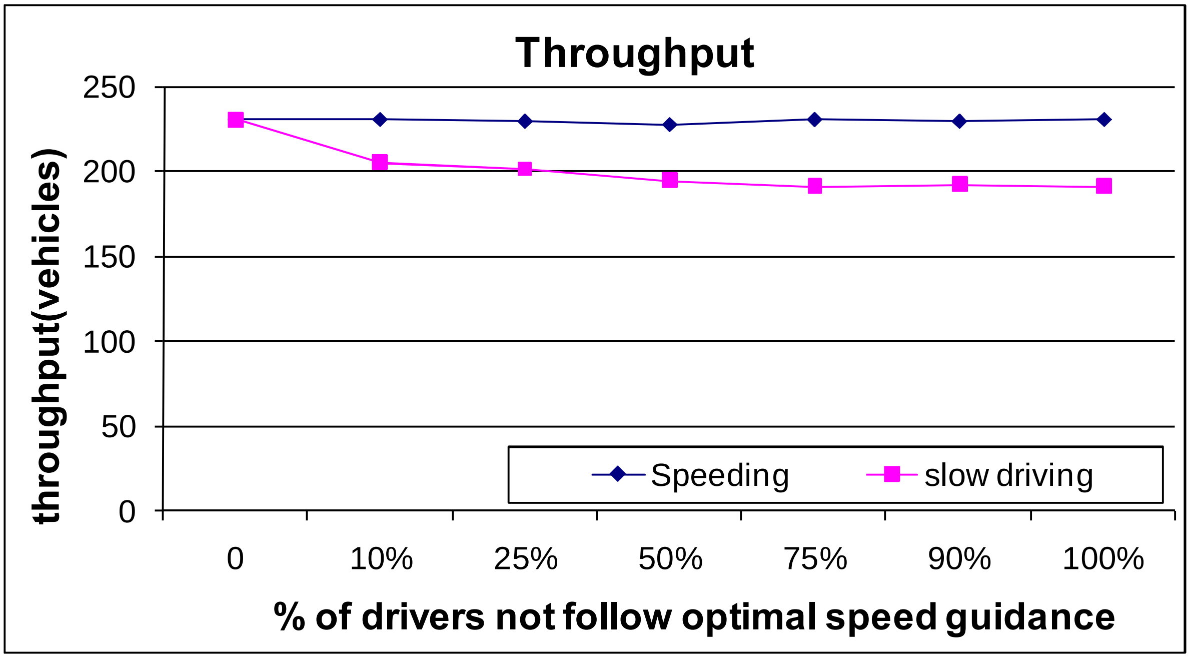


Figure 7-1 Throughput at different levels of driver compliance

Figure 7-2 shows the delay at different levels of driver compliance changes. The changes in delay had very similar trends as the change of throughput. Compared to the impact of slow driving, speeding had little negative impact on delay. Delay increased when percentage of slow driving drivers increased. Moreover, the increasing effect was smaller when the percentage of slow driving drivers increased. After the percentage of slow driving drivers reached around 50%, this change in delay was not significant while percentages of slow driving continued to increase.

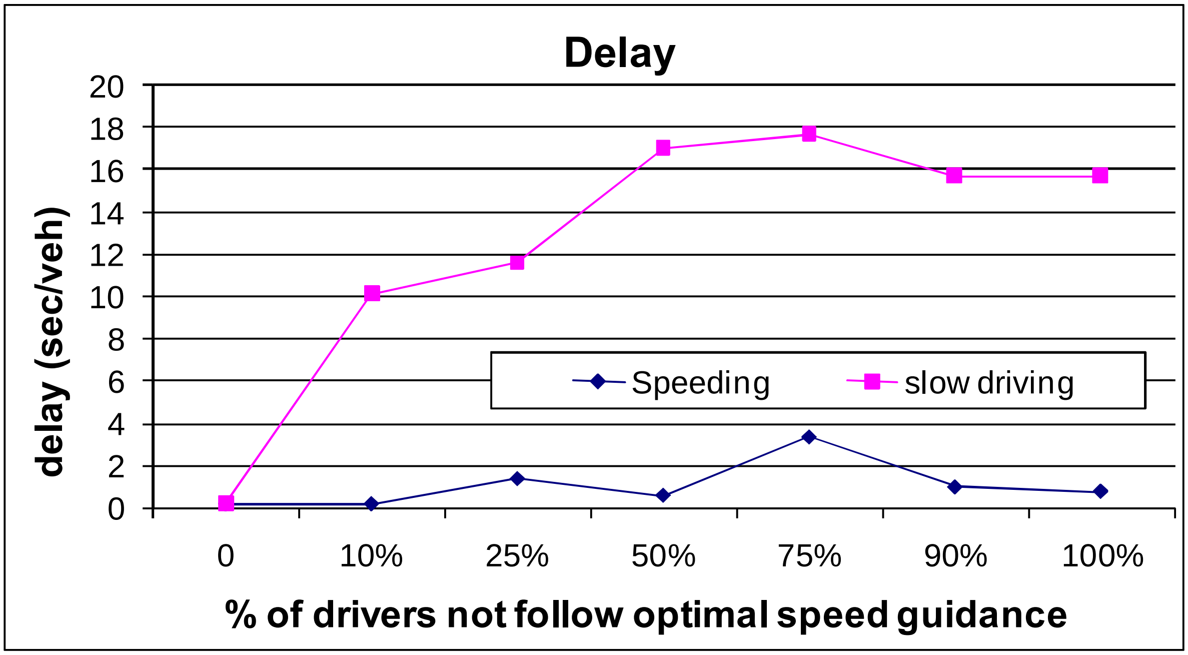


Figure 7-2 Delay at different levels of driver compliance

Figure 7-3 shows the number of stops per vehicle at different levels of driver compliance. When the percentage of slow driving drivers increased, the number of stops increased initially and became more stable after the percentage of slow driving drivers reached 50%. For the speeding cases, the number of stops did not increase initially when the percentage of speeding drivers was below 10%. After that, the number of stops increased in accordance with the increase in percentage of speeding drivers.

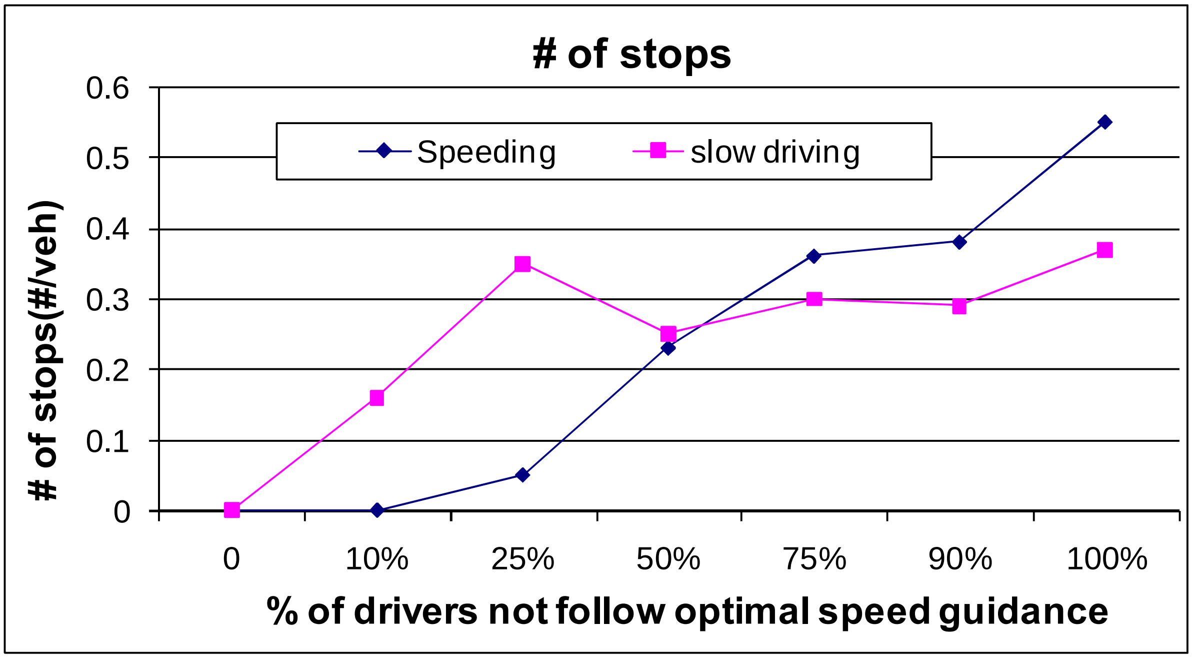


Figure 7-3 Number of stops at different levels of driver compliance

In signalized networks under DSDS control, vehicles are traveling in a platoon. Except the leading vehicle for each platoon, other vehicles are traveling in a car following mode (i.e. not in free flow). When a driver chooses to travel at a slower speed, all of the following vehicles have to slow down. Thus, even a very small percentage of slow driving vehicles could slow down a larger percentage of vehicles. Based on the results shown above, the greatest effect of slow driving occurs when the percentage of slow drivers reached between 50% and 70%. Thereafter, the performance of DSDS control will not drop dramatically while more drivers choose to drive more slowly. On the other hand, a driver can travel faster only if the vehicles in front also travel faster. Therefore, the impact of speeding only becomes obvious after there is a sufficient number of speeding drivers to influence the speed of traffic flow at many points in the network. As shown in Figure 7-3, the number of stops will start to increase only after more than 10% of drivers chose to drive faster. In a signalized traffic network under DSDS control, speeding will not increase waste green because speeding vehicles arrive at the downstream signal before it is supposed to. Speeding had very little negative impact on throughput and delay as shown in Figure 7-1 and 7-2.

7.4 Conclusions

This chapter explored the possible impact of driver compliance on the performance of the DSDS control algorithm developed. Various tests were conducted under different speed conditions. Of all the cases tested, half of them were used to evaluate the impact of slow driving and the other tests were used to evaluate speeding. Based on the simulation results, both slow driving and speeding had negative impacts on the performance of the DSDS control developed. Slow driving decreased the system throughput, increased delay and the number of stops. Speeding increased the number of stops but had very little impact on throughput and delay.

When a fixed speed limit is applied, the speed will not always be equal to dynamic optimal speed. Speed limits can be higher or lower than the dynamic optimal speed. The 100% speeding case can also be regarded as a situation where the speed limit is set higher than the dynamic optimal speed while the 100% slow driving case can be regarded as the case where a speed limit is set lower than dynamic optimal speed. Based on the simulation results, it can be concluded that employing dynamic speed control in a signalized network will improve efficiency and safety.

Chapter 8. Parallel Genetic Algorithms for Real-time Implementation of DSDS Control

The developed DSDS algorithms are formulated as dynamic optimization problems and solved by a simple genetic algorithm (SGA) in Chapter 5. The performance of SGA in terms of computation time diminishes as the size of the network increases. Fast optimization techniques are required for the DSDS control to be applicable in a real-time system. The objective of this chapter is to explore the potential benefit of the Parallel Genetic Algorithm (PGA) on a developed DSDS control algorithm.

GAs use a population, instead of individual points in their search. This attribute makes them amenable to parallelization. Parallel Genetic algorithms (PGAs) can accelerate computations as well as the optimization process. They are also less likely to get stuck at a local optimal solution than serial GAs. Hence PGAs are not only faster algorithms, but more often lead to superior numerical performance [Gordon and Whitley 1993].

Optimal performance of PGAs requires that all parameters be configured properly. Configuring a PGA to ensure efficacy of its near-optimal solutions is not a straightforward endeavor. This is, in part, due to GAsô analytic opacity and in part due to the multitude of PGA parameters that must be selected and the interaction therein. PGAs hold much promise for online traffic control optimization but more needs to be learned on how to configure them for best performance.

In this chapter, island parallel GA, which is one type of PGA, was used to solve the DSDS control algorithms developed in this research. Two traffic networks with different sizes were tested to evaluate the performance of the PGA in regards to the change of size of the problem. In order to verify the universality of the benefits PGAs have for traffic control problems, a PGA has also been applied to two benchmark functions: Onemax function which is a GA-easy function and Bi-polar function which is a GA- difficult (deceptive) function. These two benchmark functions are also used to develop an empirical approach to configure PGAs. Such an approach has been tested on a typical DSDS control problem. The GA solver developed by Goodman [Goodman 1996] was used in all experiments in the study.

The remaining of this chapter is organized as follows. The background of GA and PGA are introduced first. Next, the experiments and results of a PGA on traffic problems and two benchmark problems are presented. After that, an empirical approach to configure PGAs is described, followed by the experiments and results on PGA parameters. Conclusions are given in the last section.

8.1 Background

SGAs are powerful but can be inefficient, especially for hard and large-scale combinatorial problems. This inefficiency can stem from one or more of the following three issues. The first has to do with configuring the GA. This refers to how to determine what type of operators are needed, and what parameter values are best. The second part has to do with candidate solutions representation and/or stratification within the search space. The third issue concerns the GAsô sequential computation process.

For non-GA specialists, configuring a GA can be an involved task. Typically, good knowledge of the structure and properties of the problem (objective function and constraints) is necessary to select appropriate operators and parameter values. Good knowledge of efficient ways to represent the problem is also necessary. At the same time, problem representation is more of an art than a science. Efficient representation can have decisive impact on the performance of the algorithm-both quality of solution and time to convergence.

With sequential computation, the evaluations are very time-consuming; hence performing the many evaluations inherent in evolutionary search is sometimes impractical. This issue becomes particularly acute when the optimization solutions are needed in real time, for example, in online traffic control. The problems attributed to sequential computation can be addressed by using parallel GAs (PGAs). PGAs are not only an extension of the traditional GA sequential model, but they represent a new class of algorithms in that they may alter the behavior of the search. Thus not only do PGAs run faster; they also require smaller number of evaluations of the target function to solve difficult problems to near-optimality.

There are various potential ñconfigurationsò for PGAs. It is necessary that the chosen configurations be suitable for the application at hand. In the following subsections, knowledge on techniques for improving GA efficiency, PGAs, and Island PGA configurations will be provided.

8.1.1 Techniques for Improving GA Efficiency

Six courses of action commonly employed to make the application of GAs efficient, rapid, and productive are: 1) selection of appropriate operators and parameter values, 2) appropriate problem-specific representations of candidate solutions, 3) faster/better evaluation of solutions (or individuals), 4) structuring of individuals into subpopulations or various other classes that are treated separately with respect to application of various operators, etc., 5) division of workload among multiple loosely-coupled processors (as in a cluster or network, for example), and 6) hybridizing GAs with other non-evolutionary search methods. It is intuitive that in order to obtain the most from the GA, the first three areas need to be ñdoneò right. The fifth action can be done regardless of the structure of the GA employed or any of the other actions. The sixth area is less fundamental to the functioning of the GA per se; it is more of a supplement, but for the right type of problem, it can be valuable. A GA user will almost always get results even with a basic problem representation and primitive selection of parameter values. Those results, however, can easily be inferior, and the user might not know it. There is nothing about the GAôs mechanics that will either assure optimal search or ñwarnò the user of bad search results.

Although this paper is primarily concerned with points 3 and 5, a brief discussion of the other points is presented below.

Problem Representation: GAs can be used with either binary or real coding, and with many forms, depending on the nature of the problem to be solved. A simple method of improving GA performance is sometimes to change the genetic representation. Many researchers published results showing that binary coding worked better for their applications, while other researchers reported different results [Deb 2001]. Particularly for combinatorial optimization problems, the topic of appropriate representations is the focus of a great deal of current research [Rothlauf 2001].

There are certain advantages to each representation scheme that can be exploited for certain applications. Further discussion of this topic is beyond the scope of this paper.

Appropriate Operators and Parameter Values: One approach to improving GA search performance is to simply test different values for mutation rates, population size, number of generations, etc. Selection of different values for those parameters has become easier since the work of Goldberg and his students [Goldberg 1989, Sastry and Goldberg 2001,2002] helps to provide some guidance regarding those parameters, given some characterization of the difficulty of the problem. The Schema Theorem provides additional guidance [Goldberg 1989, Sastry and Goldberg 2001]. In many cases, however, the use of those rules and guidelines requires specialized knowledge of specific properties of the optimization problem, which may not be easy for typical non-evolutionary computation researchers. If changing the configuration parameters has no effect on the search performance, then a more fundamental problem may be the cause.

Hybrid or Mimetic Algorithms (Hybridizing GAs with Non-Evolutionary Search Methods): Many researchers have found that it is beneficial to augment the GA with additional search operators. Two types are commonly used ï a generic local search (such as using simulated annealing, non-linear sequential quadratic programming, neural nets, etc.) or problem-specific heuristic search rules. In either case, it is possible to subject each individual generated by the GA to local search, or to apply a local search only when particular ñtriggersò occur. In either case, it is possible to replace the starting GA-generated genotype by the representation of the newly-created solution (ñLamarckianò approach) or merely to assign the locally-optimized fitness to the original GA-generated genotype (ñBaldwinianò approach) [Rothlauf 2002].

Faster Evaluation of Solutions: Even when the GA is optimally configured and the problem is optimally represented, the user should ensure that the best solution is obtained as soon as possible. This is a particularly critical point for real-time online optimization such as real-time traffic controls. For these, the system operator is usually interested in the GAôs best solution so far.

In the domain of transportation, evaluation of a candidate solution often involves simulation of one or more ñscenariosò through some amount of simulated time. Anything that can be done to reduce the time to evaluate each solution can contribute enormously to reducing total time to identify good solutions. For example, rapid rejection and assignment of poor fitness (objective function values) to solutions that perform poorly in an initial time period can reduce the average evaluation time dramatically.

8.1.2 Parallel Genetic Algorithms

First, it is necessary to distinguish between Parallel GAs and parallel ñimplementationsò of a simple GA. By Parallel GA, we shall mean one in which the structuring of the population(s) is into some set of demes or subgroups which are treated separately, whether these groups are very large or as small as single individuals. A Parallel GA can be implemented on a single processor or can be implemented across multiple, loosely connected processors, but in either case, the generation of new individuals is affected by the structuring of the population into multiple groups. This is a separate issue from whether or not the GA is implemented across multiple processors (e.g., a network of loosely-coupled computers). Either a simple GA or a Parallel GA can be implemented on multiple processors, with no difference in the actual course of the search (unless asynchronous operation is allowed ï see Cantu-Paz [Cantu-Paz 2000] for more discussion). Below, we will discuss Parallel GAôs and also consider the possibilities of implementing them on multiple processors, further speeding their search time, but without fundamentally affecting their search trajectory. But a GA which uses multiple processors in parallel to evaluate the individuals in a single GA population (below, the ñGlobal Parallel GAò) will not be considered to be a parallel GA in the strictest sense.

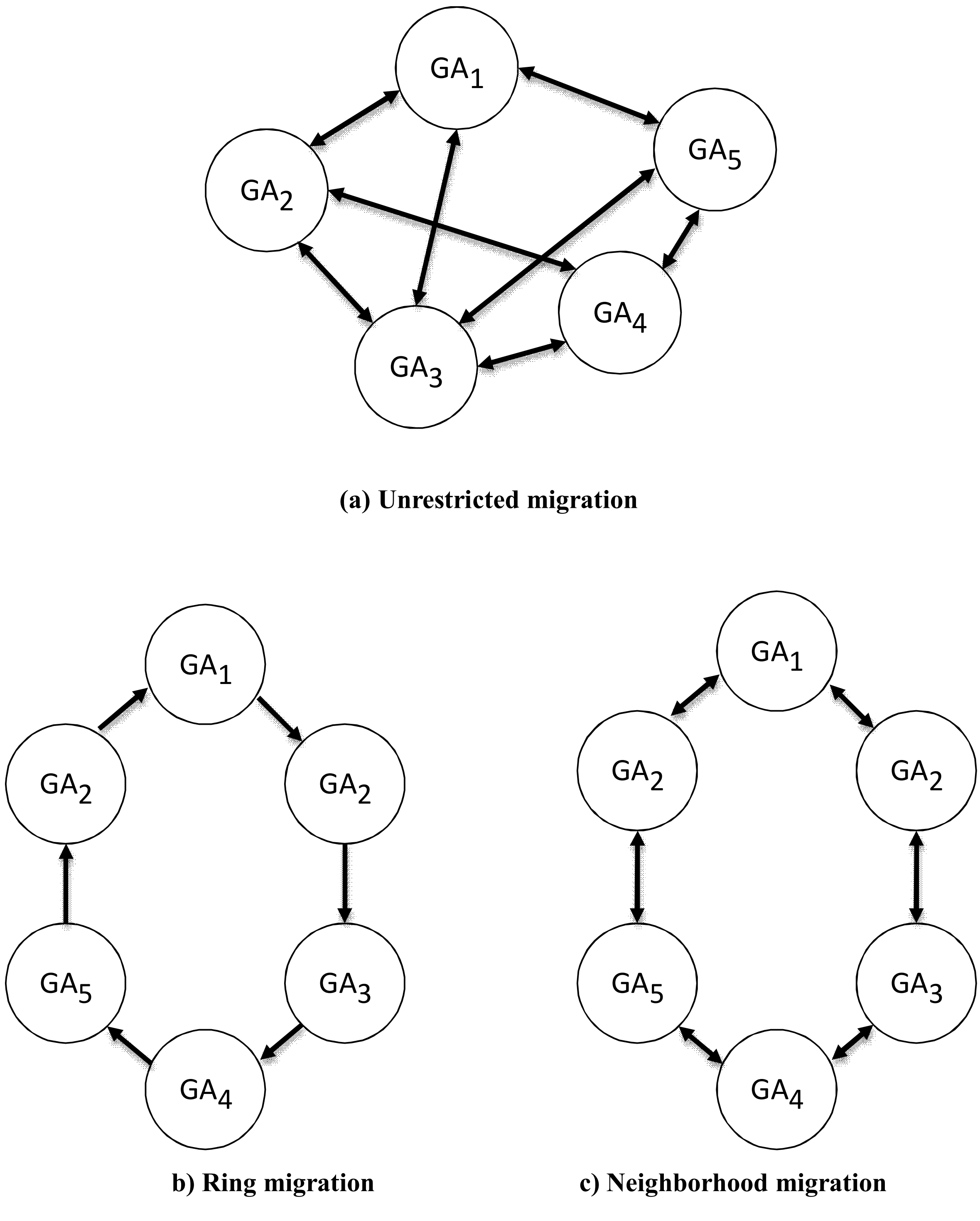
“Global Parallel” GAs: Parallel hardware can be utilized with a simple GA in the master/slave architecture. Here, an overall node called the master initializes and contains the entire population and performs the selection operation and any needed rescaling of fitness. A number n of ñslaveò nodes perform at least the function evaluations, in a parallel fashion, hence improving the speed of execution of the GA. If individuals are passed in a group to each slave, the slaves may also perform recombination and mutation in parallel, also relieving the master of some work. In this type of implementation the algorithm has to balance serial-parallel tasks to minimize bottlenecks hence the issue of a synchronous/asynchronous operation is an important consideration. However, unless the imbalance among processors is large, either mode of operation can produce similar results. The distributed evaluation GA is appealing when an objective function evaluation is expensive, but the same advantages of parallel execution can be gained by parallel GAs using parallel hardware as well.

Increasing the number of slaves, n, obviously will increase the efficiency of the GA; however, the communication requirements also increase. Therefore there is a point beyond which adding more slave nodes becomes counter-productive; however, for problems with long evaluation times for each individual, this limit is very large. Cantu-Paz and Goldberg derived a relationship that determines the optimal number of nodes [Cantu-Paz and Goldberg, 1999].

Parallel GAs: In contrast to the ñGlobal Parallel GAò described above, Parallel Genetic Algorithms (PGAs) are based on GAs, but instead of considering a single fully-mixed (panmictic) population (i.e., any individual can be crossed over with any other individual to produce offspring), parallel GAs treat individuals as being divided into groups, or as spatially distributed in some non-homogeneous fashion.

There are two types of Parallel GAs: 1) Island PGAs, and 2) Diffusion PGAs. The main differences between those types are in the population structure and method of selecting individuals for reproduction. The following subsections briefly describe these two types of PGA.

Island Parallel Genetic Algorithms: In Migration or Island or Coarse-Grained PGAs, the population is divided into small clusters, each of which is treated as a separate breeding unit under the control of a conventional GA. As noted in Figure 8-1.a, the Island PGA does not operate globally on a single population. Occasionally, individuals from the various subpopulations (islands) are permitted to migrate to other islands, where they may subsequently mate with other members of that island. There are many different implementations (topologies) of the island PGA scheme ï some examples are shown in Figure 8-1.



(a) Unrestricted migration

b) Ring migration

c) Neighborhood migration

FIGURE 8-1: Different topologies of Migration PGA

A few additional key parameters need to be defined when using an island PGA. The interval between migrations and the number of individuals to migrate are the most notable. Additionally, one has to decide which individuals are going to migrate. The topology is another decision that is determined by the target subpopulation. Traditionally, the topology is a ring and one individualð the best, or a randomly selected oneð migrates at each migration step, which takes place at predefined cyclic points in time. Many other migration schemes have also been successfully employed. Typical pseudo-code for an island PGA follows:

```
-- For Each node (GAi)
WHILE not finished
  SEQ
     é  Selection
é  Reproduction
é  Evaluation
PAR
é  send emigrants
é  receive immigrants
```

A notable advantage of island PGAs is a reduction in takeover time by superior individuals. A classical problem, premature convergence, affects simple GAs (SGAs) more strongly than island PGAs: when a superior individual is found, a simple GA will tend to begin converging toward that individual, skewing all future searches. In contrast, until that individual or its descendants propagate through all the subpopulations of an island PGA, search in the remaining islands goes on unaffected by that individual. It is possible that several superior individuals will exist in an island PGA at any time and migration allows for their recombination, but delays the convergence of the entire population to any of their genotypes.

The tradeoff is that in order to avoid an excessive number of function evaluations, each subpopulationǿs size must be smaller than the total population of the simple GA. Thus the choice of number of subpopulations and subpopulation sizes is a subtle and problem-dependent issue, which introduces some additional complication for the user of this GA. However, the advantages of island parallel GAs have been demonstrated many times [Cantu-Paz 2000].

Hierarchical Genetic Algorithms (HGAs) are another category of island PGA. They use a hierarchical topology for the layout of the subpopulations, as noted in Figure 8-2 [Lin 1994; Eby 1999]. In fact, if the GA subpopulations are also allowed to be heterogeneous (i.e., represent the problem or calculate fitness differently), this hierarchical topology makes it possible to have different layers performing different tasks. In one such implementation, the top layer concentrates on refinement while the bottom layer mostly performs exploration (see Figure 8-2). This type of implementation addresses the dilemma of having to choose between complex modeling that requires a long time to compute a fitness function or coarser but faster models. Hierarchical GAs with multiple models provide a way out of this problem through using distinct models for each level of the hierarchy.

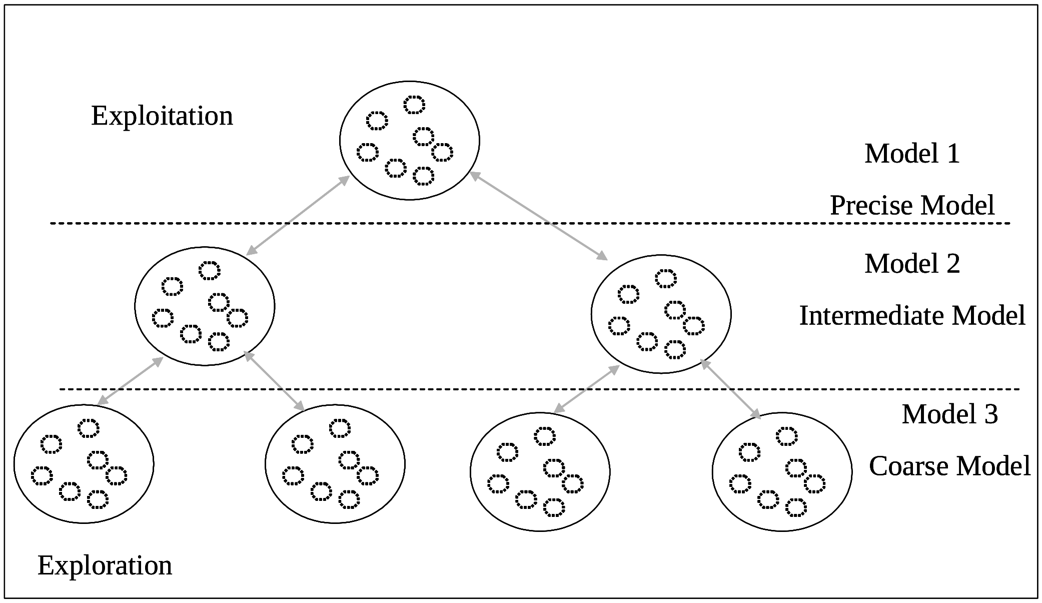


Figure 8-2: 3-level Hierarchical GA

A recent form of PGA uses the Hierarchical Fair Competition principle (ñHFCò) to structure a set of subpopulations [Hu et al. 2005]. Subpopulations are stratified according to fitness brackets, and whenever a newly-created individual has fitness higher than the range of the bracket in which it originates, it is moved out to a subpopulation with a fitness bracket corresponding to the individual. This allows for rapid exploitation of high-fitness individuals through recombination with others of like fitness, but keeps high-fitness individuals from taking over subpopulations composed mostly of low-fitness individuals, thereby restricting future exploration.

Any of these forms of island PGA can be implemented almost trivially across multiple computers or processors: each processor is assigned one or more islands, and communicates migrants to/from other subpopulations (via a buffer or synchronously) at appropriate intervals.

Diffusion (Cellular) Parallel GA: A diffusion PGA (also know as a cellular or massively parallel GA) is similar to the island PGA but overcomes the discontinuities generated by the island

PGA. Here the diffusion PGA represents the population as a single spatially distributed population with individuals being assigned a location within some 2- or 3-dimension space. Mating is allowed between individuals in the same or neighboring cells. Genetic operations take place in parallel (conceptually, at least) for every node of the population, and every individual interacts only with those in its neighborhood. The replacement policy typically destroys the considered individual by overwriting it with the newly computed string. A typical structure of a diffusion PGA is shown in Figure 8-3.

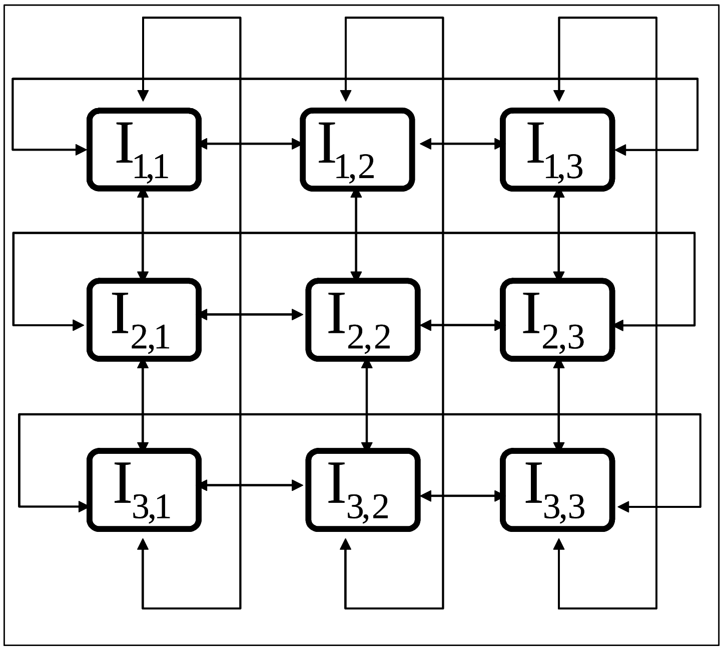


Figure 8-3: Diffusion or Cellular PGA

In a diffusion PGA, the population is a single continuous structure, but each individual is assigned a spatial location. Mating is only allowed within a small local neighborhood. For example, in Figure 8-3, I (2,2) can only mate with I (1,2), I (2,1), I (2,3), I (3,2). This isolation-by-distance property allows a high diversity, and the selection pressure is also weaker due to the local selection operator. The appearance of new species of solution in the grid and the refinement of existing solutions are both permitted and desired. In that respect, a diffusion PGA allows a well-developed balance between exploitation and exploration.

Diffusion PGAs have often been implemented on multiprocessors due to the close similarities between the model and the physical arrangement of CPUs. Another possible approach is to simulate the Diffusion PGA in a network of workstations. However, the same architecture can be simulated on a single processor, at the cost of longer run times. A pseudo code for Cellular PGA is as follows:

```
-- Each Node (Ii, j)
WHILE not finished
SEQuential
é   Evaluate
PARallel
    é   Send self to neighbors
      é   Receive neighbors
é   Select mate
é   Reproduce
```

Cantu-Paz [Cantu-Paz 2000] discusses cellular PGAs in some detail, and provides more information on their advantages and disadvantages relative to island PGAs. In the limit, a large enough set of subpopulations and a small enough neighborhood for migration in an island PGA can yield an equivalence comparable to a cellular PGA. The distinction is largely conceptual, rather than a firm boundary.

8.1.3 Island PGA Parameters

Island PGAs are the most widely used PGAs. In the Island PGA, the population is divided into a number of subpopulations and each of these relatively large subpopulations evolves separately on different processors. Each subpopulation is free to converge toward different optima. The migration operator is employed to mix good features that emerge locally in the different subpopulations. There are five parameters when using an Island PGA.

1. Number of subpopulations
2. Migration topology is the migration route for the migrant solutions.
3. Migration interval defines how often migration occurs.
4. Migration rate defines the number of migrants that will move from the giving subpopulation to the receiving subpopulation
5. Migrant selection and replacement policy defines which individuals in the giving subpopulation migrate and which individual in the receiving subpopulation are replaced

For Island PGAs, the time to reach an optimal or near-optimal solution, consists of the computation time plus the communication time. Intuitively, there are an optimal number of subpopulations beyond which additional subpopulations will not significantly decrease the optimization time. Meanwhile, the higher the migration rate is, the longer the time used to communicate between subpopulations is. Setting these two parameters to optimal values is important.

Therefore, the problem PGA users face is two-fold: first they need to know how complex their problem is, and then they must configure the PGA accordingly. In all cases users set parameters by intuition or using some ad hoc rules. This is not sufficient if the PGA performance is to be optimal, especially for on-line optimization as in real-time traffic control. The experiments in this chapter explore ways to help resolve these two questions and to suggest some guidance on how to set two of all PGA parameters, number of subpopulations and migration rate.

8.2 Experiments Setup

8.2.1 Two Benchmark Problems

The difficult problem uses the deceptive function described by Goldberg et al. [Goldberg 1989]. The difficult problem is a 30-bit function constructed by concatenating five copies of the 6-bit bipolar function shown in Figure 8-4. The overall 30-bit function has over 5 million local optima and only 32 global optima. ñDifficultyò used here refers to a high likelihood of the GA getting stuck at local optima.

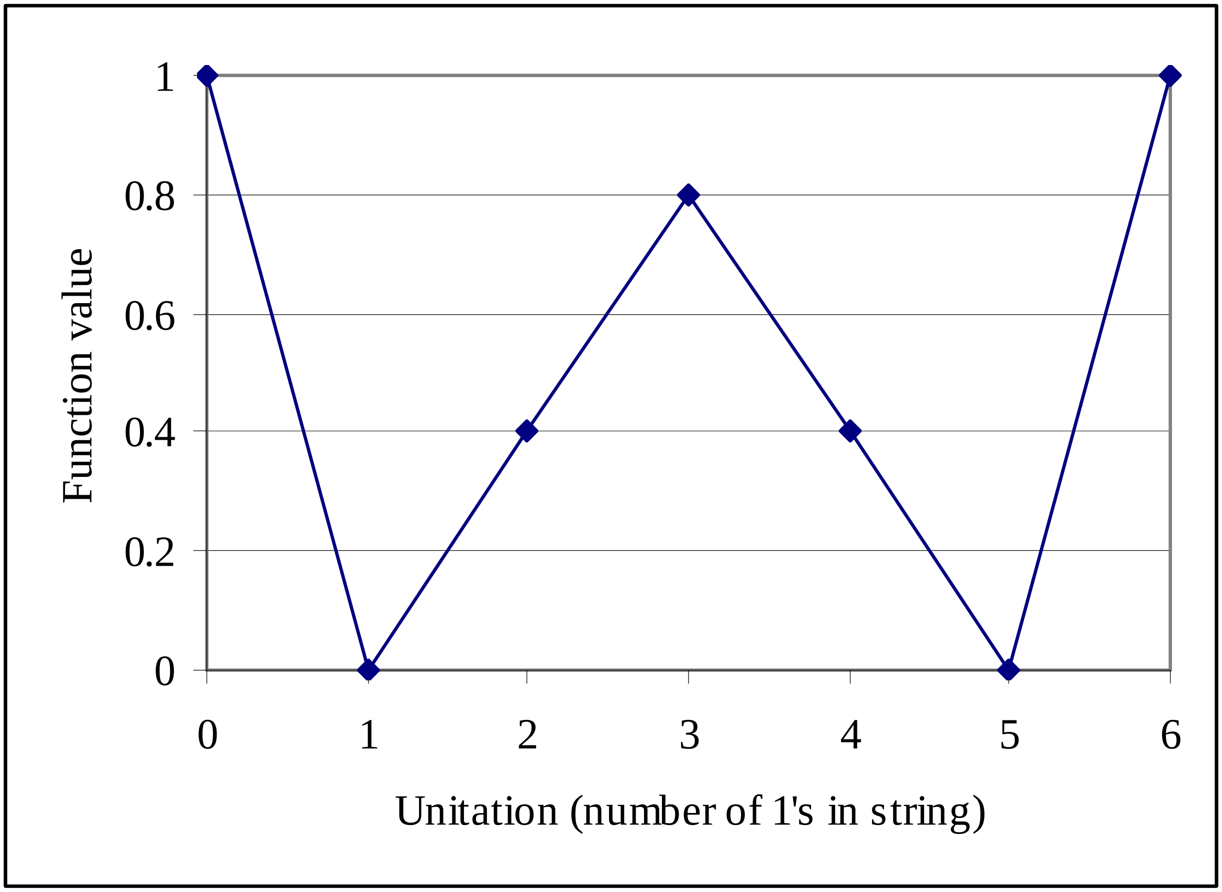


Figure 8-4: A 6-bit folded-trap functions

The standard easy problem is that of maximizing a 30-bit one-max function as follows:

$$\text{Maximize} \quad \sum_{i=1}^{n} x_i \qquad x_i \in \{0,1\}$$

The one-max function counts the number of bits set to 1 in the string and uses that value as the fitness of the individual. This is a simple function with one global optimum. It occurs when all string digits are 1s. For the 30-bit string, the maximum value of the function is 30. ñEasyò as used here refers to a low likelihood of the GA getting stuck at local optima.

8.2.2 Experiments on PGA efficiency

Three sets of experiments were conducted: 1) Experiments to compare SGA to PGA on the traffic control problem, 2) Experiments to compare different PGAs on the traffic problem, and 3) Experiments to compare SGA to PGA on standard easy and deceptive problems. In all experiments, the migration rate was set to once every 5th generations with two migrants: the ñbestò and a ñrandomlyò selected individual. The two migrants would replace the closest two individuals from a randomly selected ñgroupò of individuals from within the recipient subpopulation. A form of neighborhood migration was used. For traffic problems, the comparison was based on the computation resources needed to reach the same or closely comparable near-optimal solution. For the difficult and easy problems, the target was global optima. Different sizes of the traffic control problem were tested to see how the performance of the PGA would change with the size of the problem. Number of functions evaluations (FE) was used as a measure of computation resources. The ratios of FEs correspond approximately to the ratios of CPU time. It is noted that for both SGAs and PGAs there are many operators to choose from and parameter values to set. In the experiments of this paper, no attempt was made to select the optimal combination of operators and/or parameter values. However, in few of the experiments, different population sizes were used to evaluate their impact on the GA performance. That will be clearly noted when describing the respective experiments. Multiple independent runs were conducted in all experiments.

8.2.3 Experiments on PGA parameters

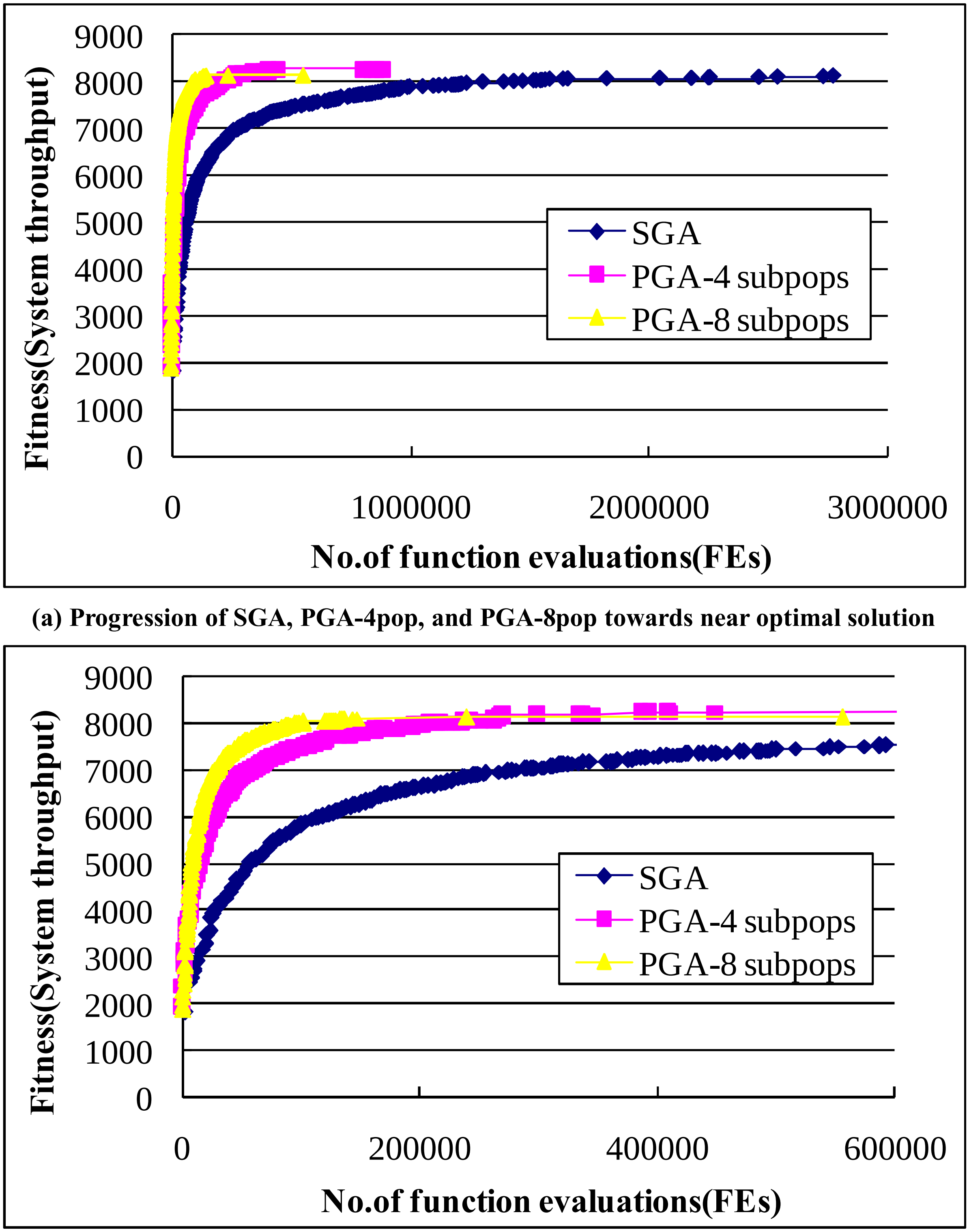
The approach used to configure a PGA consists of 4 steps: 1) use offline experiments and study the behavior of two extreme (one GA-easy and one GA-difficult) benchmark problems in relation to the subject PGA parameters, 2) correlate the optimal choice of parameters to the level of difficulty (deception) of the benchmark problem, 3) establish the level of difficulty of the problem at hand (in our case it is the traffic control problem) by comparing its performance to that of the known-complexity benchmark problems, and 4) use parameters as dictated by the benchmark problems.

Three sets of experiments were conducted: 1) compare PGAs with different numbers of subpopulations and different migration rates on the Onemax function, 2) compare PGAs with different numbers of subpopulations and different migration rates on the Bipolar function, and 3) compare PGAs with different numbers of subpopulations and different migration rates on the traffic control problem. In this section, the migration rate was larger than what was used in experiments on PGA efficiency, therefore communication time cannot be ignored. Optimization time (time to reach the target), as opposed to the number of function evaluations needed, is the basis for comparison. The target was the near-optimal solution (around 90% of the optimal fitness) for the traffic problems, and the global optima for the difficult and easy problems. Neighborhood migration was used. Migrants were randomly chosen from subpopulations as well as the individuals that were replaced in the receiving subpopulation. Multiple independent runs were conducted in all experiments to ensure statistical significance of differences between different PGA parameter combinations.

8.3 Results and Discussion

8.3.1 Results on PGA Efficiency

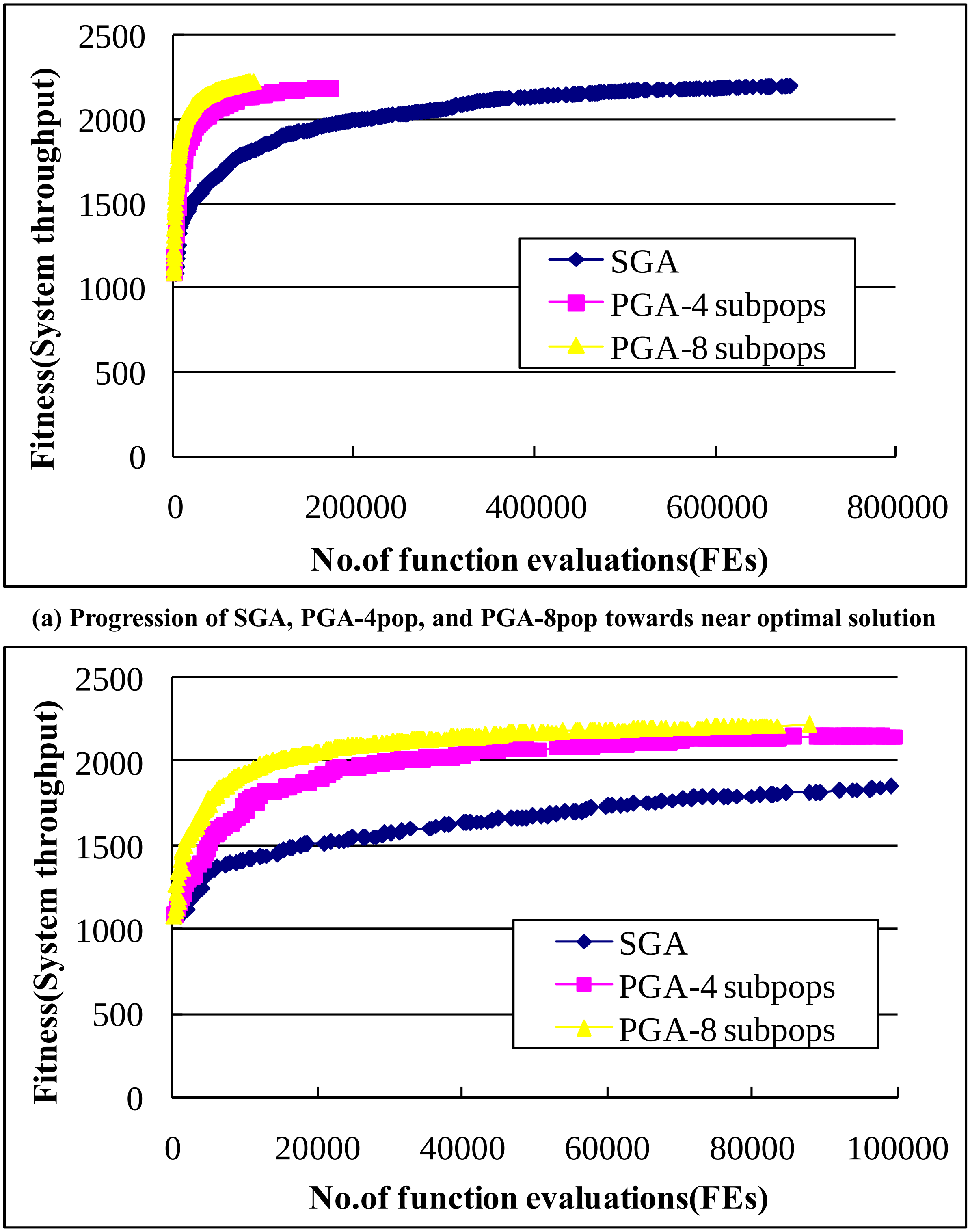
Results for Traffic problem: The experiments were carried out on two traffic systems. A small system N(2,2) composed of 4 intersections for twenty cycles, and a large system N(5,4) with twenty intersections for forty cycles. For each of the two systems, both SGA and PGA were allowed the same computation resources to optimize control to maximize system output. The focus was exclusively on the computational side of the problem and the comparison of the performance of the SGA to that of PGA. The computation resources needed to reach the near-optimal solutions were examined. A near-optimal solution is one that maximizes throughput and satisfies all constraints.



(a) Progression of SGA, PGA-4pop, and PGA-8pop towards near optimal solution

(b) Early part of the Progression shown in (a) above

Figure 8-5 Performance of SGA, PGA-4 and PGA-8 for the larger system



(a) Progression of SGA, PGA-4pop, and PGA-8pop towards near optimal solution

(b) Early part of the Progression shown in (a) above

Figure 8-6 Performance of SGA, PGA-4 and PGA-8 for the smaller system

Figure 8-5 shows the performance of the SGA, PGA-4 and PGA-8 in a typical run. Part ñaò of the figure shows the entire run (until the SGA was able to reach the same fitness as PGA-4 and PGA-8). Part ñbò of the figure shows only the early part of the same run. It is very clear that the PGAs outperformed the SGA by a significant margin. Figure 8-6 shows the equivalent results for the small system. Similar trends were observed as in the small system. Of course, if enough time is given to the SGA it will reach a comparable solution to the PGA. However the point of using a PGA is to obtain high quality solutions in shorter time, as would be necessary for on-line traffic control optimization.

Table 8-1 Number (and %) of FE needed to reach shown fitness--Large system

Fitness	**SGA**	**PGA 4 subpop**	**PGA 8 subpop**
4000	28562(100%)	7254(25%)	5497(19%)
5000	57218(100%)	14770(26%)	8727(15%)
6000	117521(100%)	27201(23%)	15015(12%)
7000	278059(100%)	59219(21%)	28047(10%)
8000	1432277(100%)	215075(15%)	95605(6%)

Table 8-2 Number (and %) of FE needed to reach shown fitness--Smaller system

Fitness	**SGA**	**PGA 4 subpop**	**PGA 8 subpop**
1000	1119(100%)	684(61%)	359(32%)
1200	3212(100%)	1914(59%)	494(15%)
1500	18467(100%)	5418(29%)	2051(11%)
2000	205473(100%)	31464(16%)	13501(8%)
2200	682496(100%)	173249(25%)	69194(10%)

Table 8-1 shows a numeric comparison of the performances of the SGA, PGA-4 and PGA-8. The numbers of FE needed to reach the shown fitness were noted for each of the three GAs. The number of function evaluations needed for the SGA was used as a benchmark and was taken as 100%. Compared to the SGA, the PGA-4 needed only between 15 and 26% of the number of FE to reach the same fitness, while the PGA-8 needed only between 6 and 19% of the number of FE. Table 8-2 shows the equivalent results for the small system.

The choice of four and eight subpopulations in the PGA is only one among an infinite number of choices. What influence would the number of subpopulations have on the performance of the PGA? This question is answered in part in the results shown in tables 1 and 2. It is shown that doubling the number of subpopulations, while keeping the total in all subpopulations the same, cuts the number of FE needed by at least one half. The results will show that depending on the complexity of the function being optimized, the relationship between number of subpopulations and FE may not be a linear one.

Results for Bipolar function: The result of PGA implementation on the difficult (bipolar) problem is noted in Figure 8-7.

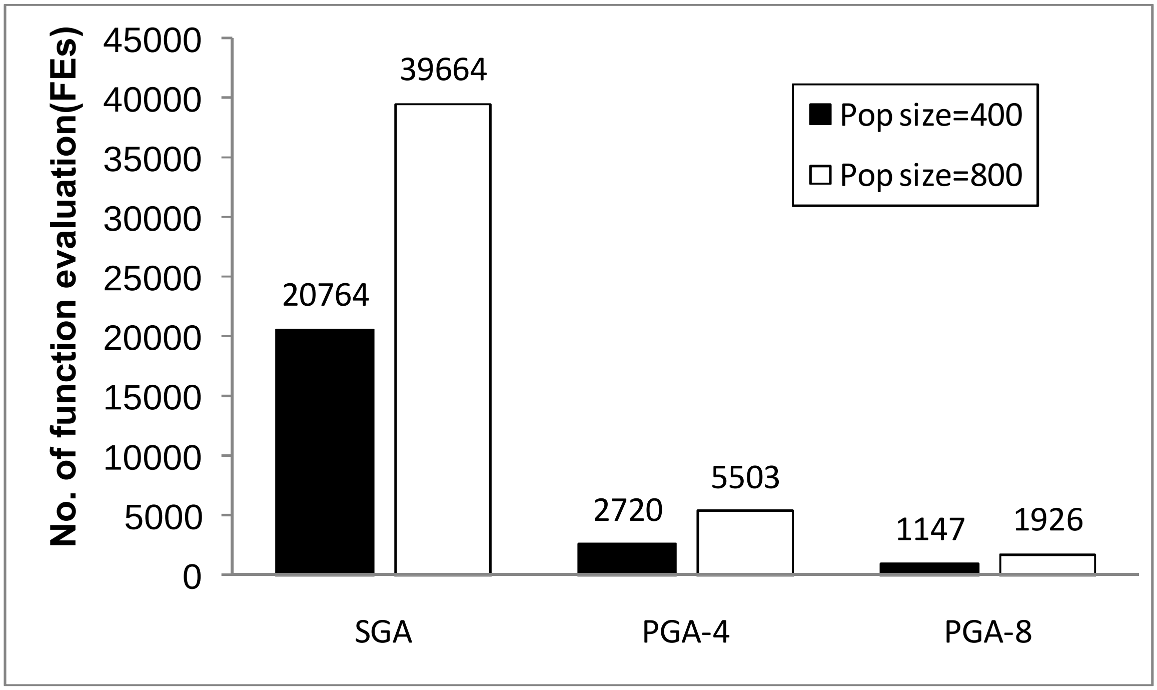


Figure 8-7: Results of PGA and SGA on Bipolar function

An SGA, a PGA-4 and PGA-8 each with two population sizes (400 and 800) was used to find the first of the 32 global optima of this function. The figure shows the average number of FEs to reach the global optima, which is the criterion used to compare the three GAs. The PGA-4 used less than one seventh (1/7) of the FEs that the SGA used to find the global optima. Doubling the population size, however, reduced the effectiveness of the PGA. This is not surprising since there is an optimal population size given the size and complexity of the problem being optimized. In this work no attempt was made to determine the optimal population size and hence it is not necessarily a ñbadò sign that a smaller population size-PGA outperformed one with a larger population size. The PGA-8 outperformed the PGA-4 by more than two folds for both population sizes. And within the PGA-8 the lower population size outperformed the higher one although the difference is not as marked as in the case of the PGA-4.

Results for Onemax function: Figure 8-8 summarizes the results of the experiments on the easy (one-max) problem. The PGA-4 used around 25% of the FEs the SGA used to reach the global optima. Doubling the population size reduced the effectiveness of the PGA. But this is neither good nor bad; it reaffirms the notion that optimal choice of different parameter values (as the population size) is necessary regardless of what type of GA is being used. Too large of a population is not good since it wastes computation resources and too small of a population causes genetic drift and leads to premature convergence. The PGA-8 outperformed the PGA-4 by more than two folds (as was the case with the bipolar problem).

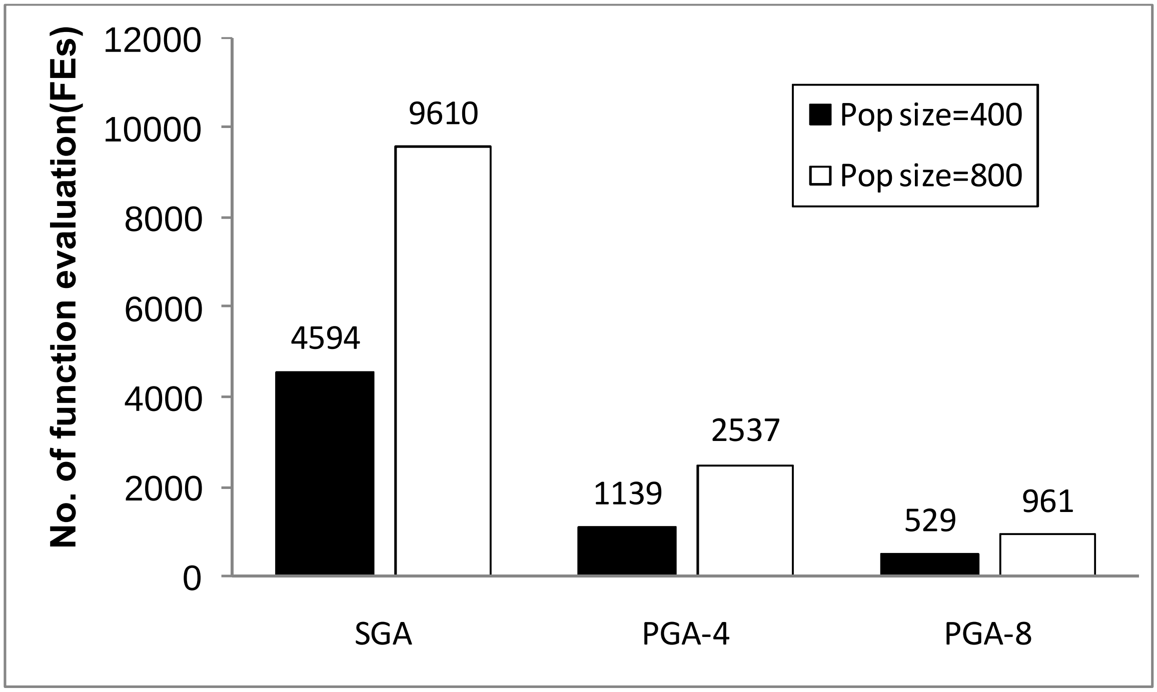
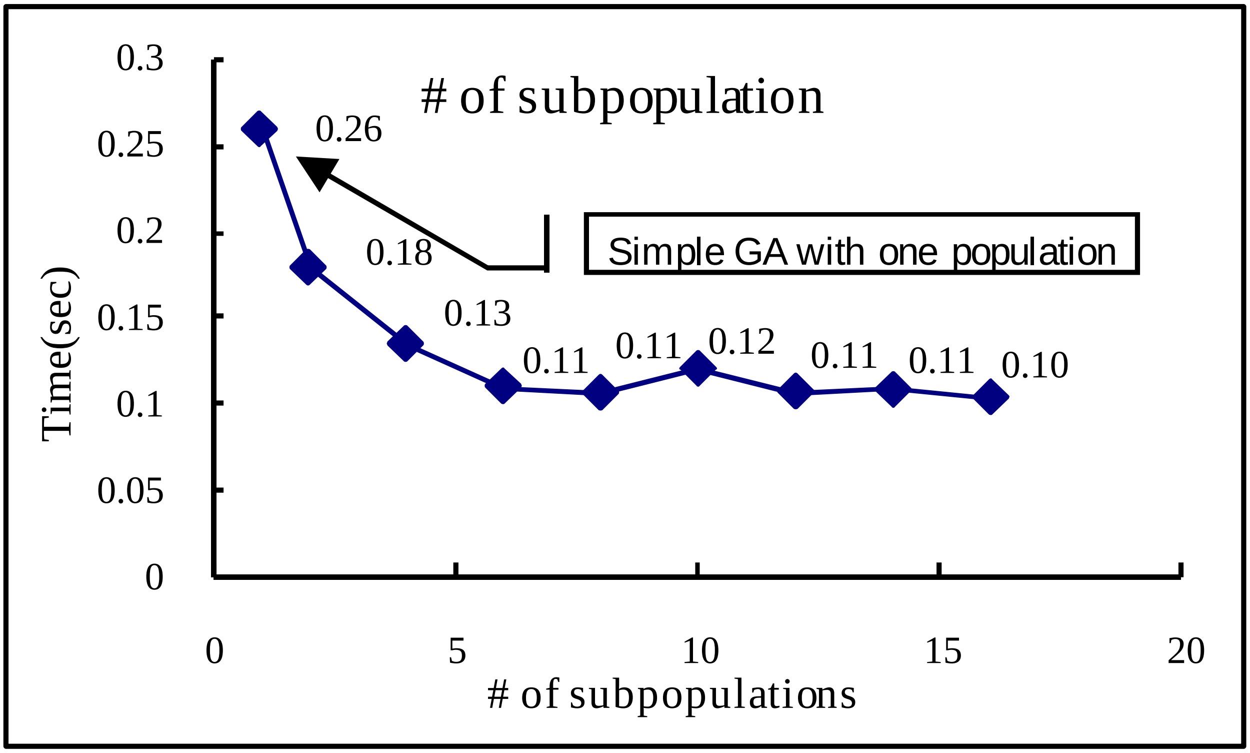


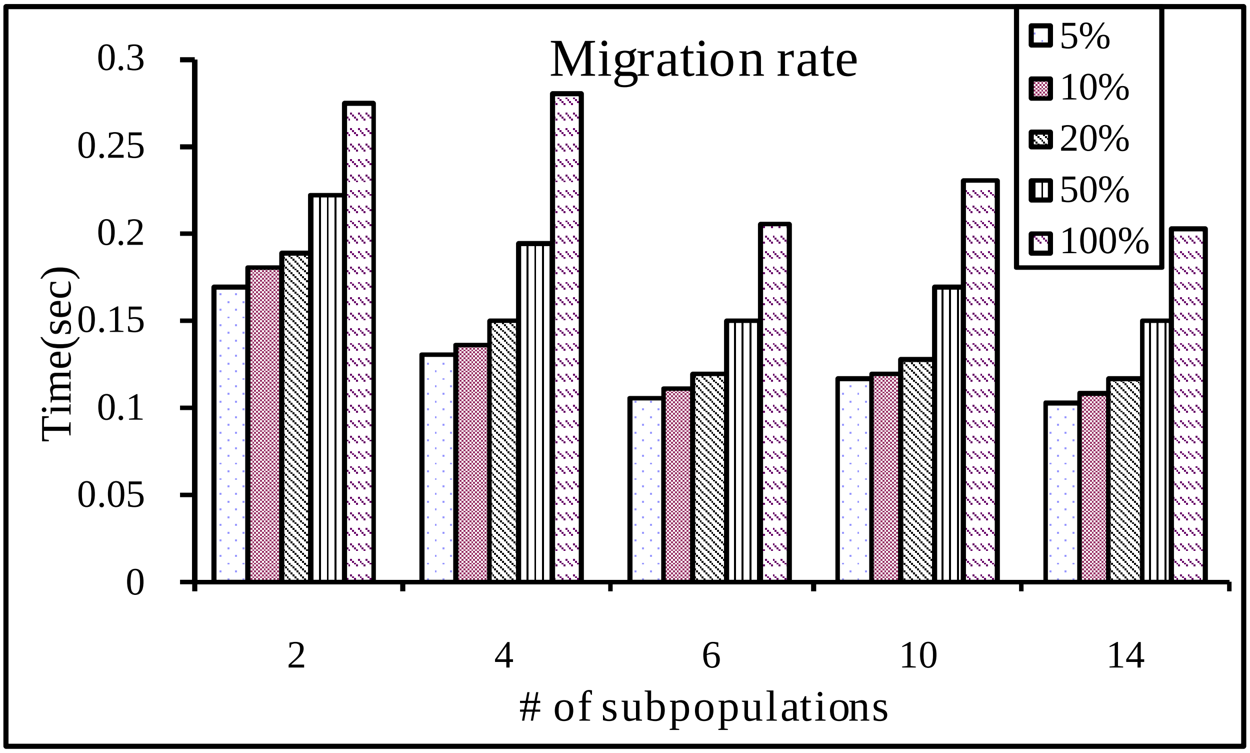
Figure 8-8: Results of PGA and SGA on Onemax function

8.3.2 Results on PGA Parameters

Results for Onemax Function: Performance of this function was evaluated with subpopulations of 2,4,6,8,10,12,14 and 16 and a 10% migration rate. Another set of evaluations was conducted with subpopulations of 2, 4,6,10 and 14 with migration rates of 5% 10%, 20%, 50% and 100%. The 100% migration rate is an extreme case where all individuals are allowed to migrate. This case serves as a benchmark. The results for Onemax function are shown in Figure 8-9.



(a) Number of subpopulation

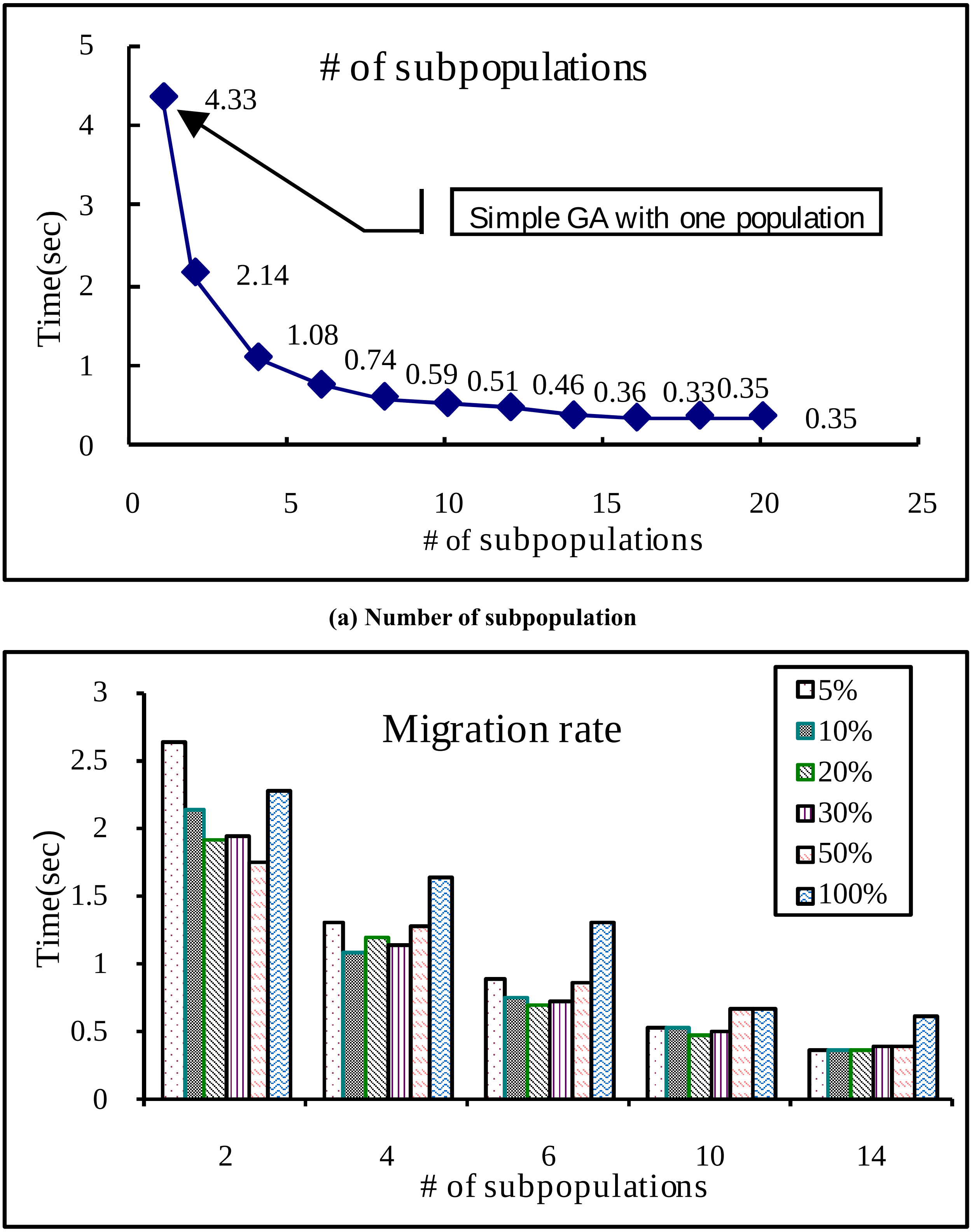


(b) Migration rate

Figure 8-9: PGAs on Onemax function

From figure 8-9(a), it is obvious that 6 subpopulations was the best number of subpopulation. A t-test was performed to verify that adding more subpopulation will not statistically decrease the optimization time. From figure 8-9(b), it is clear that for all subpopulation counts, lower migration rates help reach the optima quicker. When the number of subpopulation increases, the differences between migration ratesð especially lower ones--for a given number of subpopulations become less marked. There also appears to be a decline in the impact of migration rate as the number of subpopulations increases.

Results for Bipolar Function: Performance of this function was evaluated with subpopulations of 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 with a 10% migration rate. Another set of evaluations was conducted with subpopulations of 2, 4, 6, 10, and 14 with migration rates of 5%, 10%, 20%, 50%, and 100%. The results for the bipolar function are shown in figure 8-10.



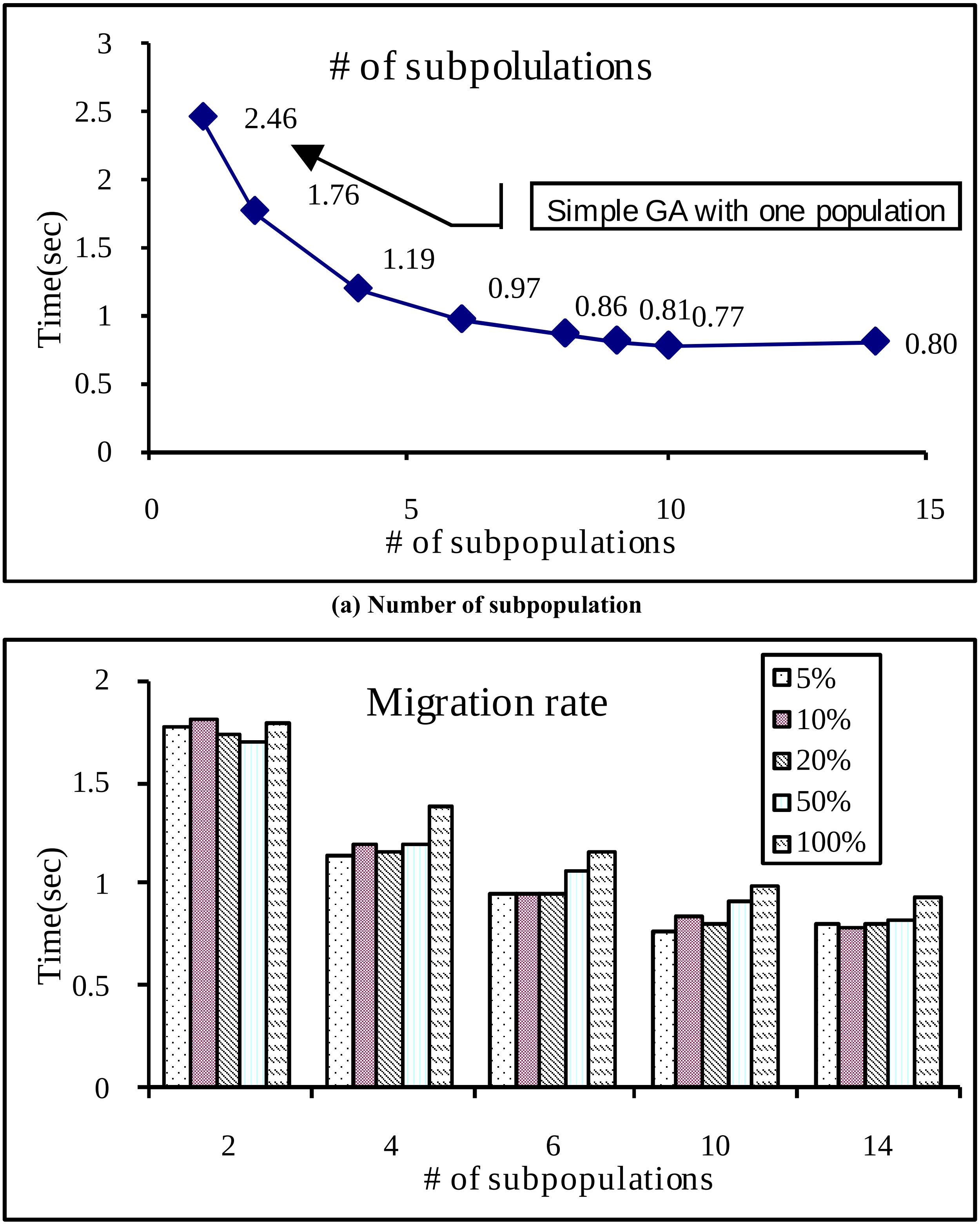
(a) Number of subpopulation

(b) Migration rate

Figure 8-10: PGAs on Bipolar function

Figure 8-10(a) shows that the optimal number of subpopulations was 14. A t-test was performed to verify that adding more subpopulations will not decrease the optimization time. Figure 8-10(b) shows that the best combination for the number of subpopulations and migration rate was 14 subpopulations and 20% migration rate. It is also noted that a 50% migration rate was the best for the PGA with 2 subpopulations. This implies that increasing migration rate to a reasonable level could help mitigate the impact of shortage of processors on difficult problems. As the number of subpopulations increases, the differences due to migration rates decreases and becomes almost nonexistent at the optimal number of subpopulations. The impact of migration rates, regardless of values, declines as the number of subpopulations increases. This implies that the migration rate is less important when more processors are available.

Results for traffic problem: Performance of this function was evaluated with subpopulations of 2, 4, 6, 8, 9, 10, and 14 with a 10% migration rate. In addition, subpopulations of 2, 4, 6, 10, and 14 were evaluated with migration rates of 5%, 10%, 20%, 50%, and 100%. Results for the traffic optimization function are shown in figure 8-11.



(a) Number of subpopulation

(b) Migration rate

Figure 8-11: PGAs on traffic problem

Figure 8-11(b) shows that the best combination of number of subpopulations and migration rate was 10 and 5%, respectively. As the number of subpopulations increases the role of migration rate becomes less significant although very high rates become counterproductive. The best migration rate for PGA with 2 subpopulations was 50% on this problem. This may imply that increasing the migration to a reasonable level could help mitigate the impact of shortage of processors on large but not difficult problems. It is also noted that with the increase in number of subpopulations, the difference between PGAs with different migration rate decreased, although this decrease is less significant than in the bipolar function.

8.4 Conclusions and Recommendations

There are different ways to improve performance of GAs including use of Parallel GAs (PGAs). This chapter presents an overview of different techniques to improve performance of GAs, with particular emphasis on parallel GAs. This chapter also presents results from applications of a simple GA (SGA) and migration PGAs on a traffic control problem, a standard GA-difficult, and standard GA-easy problem. For all problems, savings in computation resources were realized when PGAs were used. The advantage of PGAs is more pronounced for complex and difficult (deceptive) problems. On the difficult problem tested in this research, a PGA with four subpopulations was 7 times more efficient, and a PGA with 8 subpopulations was over 18 times more efficient. For smaller and less complex problems, the impact of parallelism is less dramatic when the computation resources are limited (such as limited time to obtain a solution). Use of

Parallel GAs does not reduce the importance of seeking efficient problem-specific operators, parameter values, and an efficient representation, but does magnify the effectiveness of such choices and increase the range of options available.

The number of subpopulations and migration rate are two critical parameters for proper configuration of PGAs. There are general guidelines on how to optimally select those two parameters. Experiments were carried out on benchmark GA-difficult and GA-easy problems and a traffic control problem. The benchmark problems are of known structure and complexity and helped establish benchmarks or reference points that can be used as a guide in configuring PGAs for problems of unknown difficulty. Time to reach an optimal or near optimal solution was used to compare and assess performances. The time needed by a single-population (serial) GA is 100% and was used as a reference.

Results show that more subpopulations reduce the time to reach a solution and there is a limit beyond which more subpopulations may become counter productive. This limit is dependent on problem difficulty. Larger number of subpopulations should be used for more difficult or larger problems. Migration rate has more impact on difficult or larger problems, although difficulty has more impact than size of the problem when choosing migration rates. The importance of migration rate will decrease when more subpopulations are used. With limited computational resources, migration rate is an option to improve the performance of the PGA for difficult and large problems. For easier and smaller problems, a minimum migration rate should be used.

Difficulty of a problem, and hence, its time requirement, can be judged from the time difference between a one-population GA (serial GA) and a 2-subpopulation PGA. The difference in times for the difficult and easy problems was obvious: 30% and 50%, respectively. This is a useful, quick way to judge the difficulty of a problem. It can be used to decide on the optimal number of subpopulations and migration rate. For the traffic control problem, the difference was 29% thus indicating that it is not a difficult problem although it is combinatorial. These observations were confirmed when the time difference between one-population GA and a 14-subpopulation PGA were compared. For the easy, difficult, and traffic problem, the differences were 57.7%, 91.7%, and 67.5%, respectively. Besides difficulty, size of the problem also impacts time requirements but more experiments are still needed to ascertain the exact relationship between problem size and number of subpopulations.

Based on the results reported here, the following recommendations may be made: for easy and difficult problems of sizes similar to the ones used in this research, use 6 and 14 or fewer subpopulations, respectively. For larger size problems, use a larger number of subpopulations than specified here. However, the relationship between the increase in problem size and that of the number of subpopulations is not clear at this point. More subpopulations may always be used but that will not lead to a significant savings in time, and sometimes can even be counterproductive. For difficult and large size problems, higher migration rates should be explored if computational resources (processors) are limited. Lower migration rates should be used for easy problems. Although some of these recommendations are not very specific, a ñcookbookò approach to configuring a PGA may not be feasible or even necessary given the nature and multitude of factors that impact the performance of PGAs.

Chapter 9. Conclusions and Future Work

9.1 Conclusions

In order to control the traffic flow more effectively and more flexibly, the integrated dynamic speed and dynamic signal (DSDS) control algorithms have been developed in this research. The developed DSDS control algorithms are optimized by exploring genetic algorithms (GAs) and they have been used to optimize traffic operations in signalized networks.

Two types of DSDS algorithms are developed: cycle-based and vehicle-based. Signal timing parameters and speed parameters are adjusted according to prevailing traffic conditions in both of these algorithms. Compared to traditional traffic controls, the potential benefits of using DSDS controls includes: better signal coordination, higher efficiency of green time, improved safety, reduction of fuel consumption, and more flexibility. Models are developed to evaluate the traffic flow in the signal network under DSDS control. The developed models have been validated by the microscopic software VISSIM. By comparing the results from VISSIM and the developed macroscopic DSDS control algorithm, it is concluded that the developed macroscopic models are valid and can accurately evaluate traffic flow.

Genetic algorithms, a fast search and optimization technique based on the mechanism of natural selection and natural genetics, have been used to optimize the DSDS control. Both cycle-based DSDS algorithms and vehicle-based algorithms are implemented on simplified hypothetical networks. The traffic operations under the DSDS control are compared to those under the DSFS control. The results have shown that the developed DSDS control can be effectively integrated to control signalized networks. It provides an alternative solution and a more flexible approach to control signalized networks.

The potential capability of the DSDS control on reducing speed noises and speed variation for signalized networks has also been demonstrated in this study. The results have shown that the DSDS control is able to achieve smoother traffic flow by reducing speed noises and speed variations. It indicates that dynamic speed control is an effective and necessary addition to a signal control system for signalized networks. The DSDS control provides a more flexible and effective alternative in designing safer and more fuel conservative signalized network systems.

The impacts of driver compliance on the performance of the DSDS control algorithm have been studied. Various tests have been conducted under different speed conditions. Both slow driving and speeding have been evaluated. Based on the simulation results, speeding and slow driving have negative impacts on the effectiveness of the DSDS algorithm. It can also be concluded that employing a dynamic speed control in a signalized network will improve efficiency and safety.

There are different ways to improve the performance of GAs including use of Parallel GAs (PGAs). Parallel Genetic Algorithms (PGAs) save computation resources by distributing the working load to different processors. Based on the results of the testing problems, it is found that the savings are more significant for complex or larger optimization problems. The empirical method described in this research provides guidance on how to configure two of the PGA parameters, number of subpopulations and migration rate.

9.2 Future Work

With the positive conclusions in this research, further extension of this research should be considered. The research presented in this thesis can be further extended into the following areas:

(1) The DSDS algorithm in this research was developed based on simplified hypothetical networks with only through movements. In order for the model to be practical for field implementations, the algorithm needs to be extended to cover major turning movements, stochastic effects of traffic flows, and demand-responsive plans. As the networkôs geometry and traffic operations grow in complexity, the DSDS will likely have a larger effect.

(2) This research analyzed operations of signalized networks under DSDS control and oversaturated traffic demands in major directions and lower demands in secondary directions, where the objectives are to avoid queue-spillbacks and grid locks to improve throughput and minimize delay. The DSDS control algorithm can also be tested in different scenarios such as incident management, special event traffic diverting management, and integrated corridor controls. Different objectives for different scenarios can be realized by developing different objective functions and fitness functions.

a) It is also valuable to explore potential applications of integrating the DSDS algorithm with other ITS technologies such as dynamic route guidance to create an even more flexible and comprehensive ITS system.

b) From the computational side, more effective parallel GA strategies can be further investigated. For example, one can explore what is the most efficient way for communication among processors.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abu-Lebdeh, G. (1999). Development of Dynamic traffic signal control procedures for oversaturated arterials and Genetic Algorithms Solutions. Ph.D. dissertation. University of Illinois at Urbana-Champaign

Abu-Lebdeh, G. (2002). Integrated adaptive-signal Dynamic-speed control of signalized arterials. Journal of transportation engineering. Volume 128, No.5, p447-451

Belding, T. C.(1995). The distributed Genetic Algorithm Revisited. In Eshelman L.J.(ed.): Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, San Francisco, CA 114-121

Bloomberg, L. and Dale, J. (2000). Comparison of VISSIM and CORSIM Traffic Simulation Models on a Congested Network. *Transportation Research Record1727,* TRB, National Research Council, Washington D.C. pp. 52-60.

Bretherton, D, Wood, K., &Baker, K.(1999). Congestion and incident management using the SCOOT UTC system. 6th World Congress on Intelligent Transport Systems, London, UK

Cantu-Paz, E.(2000). Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, 2000

Cantú-Paz, E. and Goldberg, D.E.(1999). Parallel genetic algorithms with distributed panmictic populations, University of Illinois Genetic Algorithms Laboratory (ILLiGAL), Report # 99006, 1999

Chang, D. and Morlok, E. 2005. Vehicle Speed Profiles to minimize work and fuel consumption. Journal of Transportation Engineering, Vol. 131, No. 3, pp. 173-182.

Chard B.M. and Lines C.J. (1987), "TRANSYT--The Latest Developments," Traffic Engineering and Control, July/August, pp. 387-390.

Chen, H., Abu-Lebdeh, G., & Najjar, Y. (2004): Parallel Genetic Algorithms for On-Line Traffic Control Problems. Artificial Neural Networks in Engineering (ANNIE), Vol. 14, pp 653-658

Chen, Hui; Abu-Lebdeh, G. & Goodman, E (2004): Improving Performance of Genetic Algorithms for Transportation Systems: The Case of Parallel Genetic Algorithms. Proceedings of the 83rd Transportation Research Board Meeting, DC, January 2004

Chen, H., Abu-Lebdeh, G., & Najjar, Y. (2005): Configuring Parallel Genetic Algorithms for On-Line Traffic Control. In Artificial Neural Networks in Engineering (ANNIE), Vol. 15, pp 253-258

Chen, H. and Abu-Lebdeh,G.(2006) ñDevelopment of a Framework for an Integrated Dynamic Signal-Dynamic Speed Traffic Management Algorithm for Signalized Networks.ò Compendium of papers CD-ROM of 85th Transporation Research Board annual meeting. Washington DC. Jan,2006.

Chen, H. & Abu-Lebdeh (2006): "Assessment of Capacity and Flow Improvements for Combined Dynamic Signals and Dynamic Speed in Signalized Networks". Submitted to IEEE Transactions on Systems, Man, and Cybernetics

Chen, H., Abu-Lebdeh, G. (2007) Minimizing Speed Variation and Speed Noise Using Dynamic Speed Control in Signalized Networks: Proceeding of the 2008 ANNIE Conference
Coleman, J.A., et al.: FHWA Study Tour for Speed Management and Enforcement Technology. FHWA-PL-96-006. Federal Highway Administration, U.S. Department of Transportation, Feb, 1996

Committee for Guidance on Setting and Enforcing Speed Limits (1998). Managing speed, Transportation Research Board, Special Report, 254

Comte S L, Várhelyi A, and Santos J (1997) The effects of ATT and non-ATT systems and treatments on driver speed behaviour. R3.1.1. MASTER, VTT Espoo, Finland.

Corben, B; Lenne, M; regan, M; and Triggs, T.(2001). Technology to enhance speed limit compliance. Proceedings of 2001 Australasian Road Safety Research, Policing and Education Conference

Dasgupta, D., Michalewich, Z. (1997). Evolutionary algorithms in engineering applications, Springer-Verlag, Berlin.

Deb, Kalyanmoy (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, John & Sons, 2001

Donati, f., et al.(1984). A Hierarchical decentralized traffic light control system, the first realization, IFAC 9th World congress, Hungry, Budapest, Vol. 11, 11G/A-1

Duerr, P.A.(2000). Dynamic Right-of-Way for Transit Vehicles: Integrated Modeling Approach for Optimizing Signal Control on Mixed Traffic Arterials. In Transportation Research Record 1731, TRB, National Research Council, Washington, DC, 2000, 31-37.

Eby, D., Averill, R. C., Goodman,E., and Punch, W. (1999). Optimal Design of Flywheels Using an Injection Island Genetic Algorithm. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing,* 13, p. 389-402, 1999.

Federal Highway Administration (FHWA) (1984), "TRANSYT-7F Users Manual," University of Florida.

FHWA (1997). Effects of Raising and Lowering Speed Limits. Publication No.
FHWA-RD-97-084 January 1997

Florida to implement variable speed limit(2003). Urban Transportation Monitor. V. 17, NO. 11

Foy, M.D.; Benekohal, R.F.; & Goldberg, D.E. (1992): Siganl Timing Determination Using Genetic Algorithms. TRR1365, TRB, Washington, D.C., pp 108-115

Fwa, T.F.; Chan, W.T;; & Tan, C.Y.(1994): Road-Maintenance Planning Using Genetic Algorithms. II: Analysis. Journal of Transportation Egineering, vol. 120, No. 5, Sept./Oct., pp 693-709.

Gartner, N.H.(1982). Demand responsive decentralized urban traffic control part I: single-intersection policies, U.S. department of transportation.

Gartner, N.H.(1983). Demand responsive decentralized urban traffic control part II: Network extension, U.S. department of transportation.

Girianna, M. and Benekohal, R.F.(2002). Dynamic signal coordination for networks with oversaturated intersections. Transportation research record 1811,P122-130

Goldberg, D.E.: Sizing Populations for Serial and Parallel Genetic Algorithms.ò Proceedings of the 3rd Intǒl Conference on Genetic Algorithms, 1989, pp.70-79

Goldberg, D.E. (1989). Genetic Algorithm in search, optimization and machine learning, Addison Wesley Longman, Inc.

Goldberg, D.E. and Sastry, K(2001). A Practical Schema Theorem for Genetic Algorithms Design and Tuning. Presented at the Genetic and Evolutionary Computation Conference (GECCO), July 2001, San Francisco.

Gondra, Iker; Samadzadeh, Mansur H. (2003). A Coarse grain parallel Genetic Algorithm for finding Ramsey numbers. Proceedings of the 2003 ACM symposium on applied computing.

Goodman, E. (1996). An Introduction to GALOPPS, the "Genetic Algorithm Optimized for Portability and Parallelism," Release 3.2, (software and Userôs Guide), Technical Report #960801, MSU Genetic Algorithm Research and Applications Group and Case Center for Computer-Aided Engineering and Manufacturing, Michigan State University, 80pp.

Grefenstette, J.J. (1981). Parallel adaptative algorithms for function optimization. Technical Report CS-81-19, Vanderbilt University, Nashville, T.N.

Hadi, M.A. and Wallace, C.E. (1993). Hybrid Genetic Algorithm To Optimize Signal Phasing and Timing. TRR #1421, TRB, Washington, D.C., pp104-112

Hegyi, A.; Bart De Schutter; Hellendoorn, J.(2005). Optimal coordination of variable speed limits to suppress shock waves Intelligent Transportation Systems, IEEE Transactions, Volume 6, Issue 1. pp102 - 112

Head, K.L., Mirchandani, P.B., and Shelby, S. (1997). Estimation and prediction in a real-time traffic adaptive signal control logic: the RHODES prototype, Informs meeting, San Diego

Head, K.L., and Mirchandani, P.B. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. Transportation Research. Part C: Emerging Technologies, Vol. 9 No. 6, p. 415-432

Hegyi, A., De Schutter, B and Hellendoorn, J. (2003). Optimal coordination of variable speed limits to suppress shock waves.; Transportation research record, 2003.

Henry, J.J., et al.(1983). The PRODYN real time traffic algorithm, 4th IFAC-IFIP-IFORS conference on control in transportation systems, Germany, Baden-Baden 307

Henry, J.J., and Farges, J.L.(1989). PRODYN, Proceeding 6th IFAC-IFIP-IFORS Symposium Transportation, Pergamon, Oxford, pp. 505-507

Hu, J., Goodman, E. and Seo, K.(2003). Continuous Hierarchical Fair Competition Model for Sustainable Innovation in Genetic Programming. In Genetic Programming Theory and Practice, Kluwer, 2003, pp. 81-98. May, 2003

Hunt, P.B., et al. (1991). TRRL Report 1014

Leyva, Ralph et al. (2004).A REPORT ON THE USE OF TRAFFIC SIMULATION MODELS IN THE SAN DIEGO REGION. ITE-california border section

Li, Z., et al, (2001). Realization of semiconductor device synthesis with the parallel Genetic

Algorithm. Proceedings of Asia and South Pacific Design automation conference, Yokohama, Japan :45-49

Lin, S.C. , Goodman,E. , and Punch,W. (1994) Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach. *IEEE Conf. on Parallel and Distrib. Processing*, Nov., 1994.
Lin, P-W, Kang, K-P and Chang, G-L(2004). Exploring the effectiveness of variable speed limit controls on highway work-zone operations. Intelligent Transportation System. 8:1-14 2004

Lowrie, P.R. (1982). SCATS principles, methodology, algorithm, IEE conference on road traffic signaling, IEE publication 207, England, London, 67-70

Luk, J. (1984). Two traffic-responsive area traffic control methods: SCATS and SCOOT, Traffic engineering and control, pp 14-20

Memon, G.Q. and Bullen, A.G.R. (1996): Multivariate Optimization Strategies for Real-Time Traffic Control Signals. Transportation Research Record No. 1554, p. 36-42

Moen, B., Fitts, J., Carter, D., and Ouyang, Y. (2000). A comparison of the VISSIM Model to Other Widely Used Traffic Simulation and Analysis Programs. Presented at the *Institute of Transportation Engineers*

National Highway Traffic Safety Administration(2004). Budget in brief, FY2004
Oh, J., Oh, C., Ritchie, S. and Chang, M. (2005). Real-time Estimation of Accident Likelyhood for Safety Enhancement. Journal of Transportation Engineering. Volume 131, NO. 5, pp. 358-363

Park, B., Messer, CJ and Urbanik, T, II(2000). Enhanced Genetic Algorithm for signal timing optimization of oversaturated intersections. Transportation research record 1727, P122-130.

Paterson, D.(2003). Melbourneȇs Variable Speed Limit System. Proceedings of the 21st ARRB and 11th REAAA conference. Transport. Our highway to a sustainable Future

Pearson, Rebecca [internet].[Update 11/01/01]. Traffic Signal Control. Available from: http://www.calccit.org/itsdecision/serv_and_tech/Traffic_signal_control/trafficsig_report.html
Pignataro, L.J., McShane, W.R., Crowley, K.W., Lee, B., &Casey, T.W.(1978). Traffic control in Oversaturated Street Networks, NCHRP Report 194, TRB, National Research Council, Washington, D.C.

PTV America, Inc.(2005). VISSIM 4.10 User Manual. Karlsruhe, Germany

Research and Innovative Technology Administration (RITA) [Internet]. [updated 2007 Nov 28]. VII concept of operations. Available from: http://www.its.dot.gov/vii/vii_concept.htm

Robertson, D.I., and Bretherton, D. (1991). Optimizing networks of traffic signals in real-time: The SCOOT Method, IEEE Transaction Vehicular Technology, Vol. 40(1), pp. 11-15

Robinson, M. (2000). Examples of Variable Speed Limit Applications. Speed Management Workshop, 79th Annual Meeting of the Transportation Research Board, January 2000, Washington, D.C.

Robinson, M., McGowen, P. , Habets A., and C. Strong (2002). Safety Applications of ITS in Rural Areas. United States Department of Transportation, ITS Joint Program Office, Washington, D.C.

Roess, R. P., Prassas, E. S. and Mcshane, W. R.(2004). Traffic Engineering (Third edition), Chapter 24, p695-697

Rothlauf, F.(2001) The Influence Of Binary Representations Of Integers On The Performance Of Selectorecombinative Genetic Algorithms. GECCO 2002, 2002: 695

Sadek, A.W.; Smith, B.L.; and Demetsky, M.J.(1997). Dynamic Traffic Assignment: A Genetic Algorithm Approach. In Transportation Research Record 1558, Transportation Research Board, Washington, D.C., 1997, 95-103.

Sastry, K.M.(2001). EVALUATION-RELAXATION SCHEMES FOR GENETIC AND EVOLUTIONARY ALGORITHMS. MS THESIS, Department of General Engineering, University of Illinois at Urbana-Champaign, 2001

Sastry, K., Goldberg, D. E.: Genetic Algorithms, Efficiency Enhancement, and Deciding Well with Differing Fitness Bias Values. University of Illinois Genetic Algorithms Laboratory (ILLiGAL), Report # 2002003, 2002

Sastry, K., Goldberg, D. E.: How Well Does a Single-Point Crossover Mixing Building Blocks with Tight Linkage? University of Illinois Genetic Algorithms Laboratory (ILLiGAL), Report # 2002013, 2002

Schnecke, Volker and Vornberger, Oliver(1996). An Adaptive Parallel Genetic Algorithm for VLSI-Layout Optimization.4th Int. Conf. on Parallel Problem Solving from Nature (PPSN IV), September 22-27, 1996, Berlin, Germany

Schwarzkopf, A. B. and Leipnik, R. B. 1977. Control of highway vehicles for minimum fuel comsumption over varying terrain. Transportation Research, 11(4), pp. 279-286.

Texas Transportation Institute (2007). 2007 Urban Mobility Report.

Thorton, M and Lyles, R. W. (1996). Freeway speed zones: Safety and compliance issues. Transportation Research Record 1635, pp 18-25.

Tian, Z. Z., Urbanik, T., Engelbrecht, R., and Balke, K. (2002). Variations in Capacity and Delay Estimates from Microscopic Traffic Simulation Models. *Transportation Research Record 1802,* TRB, National Research Council, Washington D.C. pp. 23-31.

Tignor, C. Samuel and Warren, Davey(1991). Driver Speed Behavior on U.S. Streets and Highways. ITE Car and Driver, May 1991

US Department of Transportation (ITS Joint Program Office)(2005). VEHICLE INFRASTRUCTURE INTEGRATION(VII)-VII Architecture and Functional Requirements, Version 1.1. July 20th, 2005

Vincent, R.A., and Young, C.P. (1986). Self-Optimizing Traffic Signal Control Using Microprocessors - the TRRL 'MOVA' Strategy for Isolated Intersections. Traffic Engineering and Control 27, no. 7-8. Jul-Aug. p.385-387.

Wilkie J. K.(1997). Using Variable speed limit signs to mitigate speed differentials upstream of reduced flow conditions. Compendium of graduate students papers: advanced surface transportation systems, 1997, pp M1-M37

Wolshon, B., Taylor, W.C. (1999). Analysis of intersection delay under real-time adaptive signal control, transportation research part C7, pp 53-72

Yagar, S., Dion, F. (1996). Distributed approach to real time control of complex signalized networks, Transportation research record 1554, pp 1-8