A Traffic Signal Control Algorithm for Emergency Vehicles

by

©Md Asaduzzaman

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Memorial University of Newfoundland

March 2017

St. John's Newfoundland

Abstract

Signal preemption disrupts normal traffic signal to allow emergency vehicles to pass through the intersection more safely and quickly. In medical emergency situations, EVP (Emergency Vehicle Preemption) offers a faster response to the sufferer which improves the chance of survival. Despite this lifesaving advantage, conventional preemption has some problems which need more attention. Two important issues are increased delay of overall traffic due to preemption and absence of prioritization of conflicting preemption requests.

This thesis presents a traffic signal control algorithm that addresses the above. We have used TSP (Transit Signal Priority) techniques to improve the EVP system. TSP is a proven strategy to provide a better quality public transit operation in urban areas. Our proposed algorithm adjusts signal phases using TSP techniques to serve an emergency vehicle. We consider both single and multiple simultaneous emergency vehicle requests. TSP techniques help us to alleviate the impact on general traffic. For multiple emergency vehicle requests, a branch and bound algorithm is developed that prioritizes among conflicting requests. Experiments have been conducted using the VISSIM microscopic traffic simulator. Results show that the proposed traffic control algorithm reduces overall traffic delay by up to 8% compared to conventional EVP system.

Acknowledgements

I would like to express my sincere gratitude to my academic and research supervisor, Dr. Krishnamurthy Vidyasankar for his guidance toward this journey.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Emergency vehicle provides emergency response to save lives and properties. From minor accidents to major disasters, emergency vehicles (police cars, fire trucks, ambulances) rush to the scene to provide initial help and treatment. In some emergency situations, early response is really important and a couple of seconds can make a difference between life and death. For the patients having sudden cardiac arrest(SCA), chances of survival are reduced by 7-10% with every minute delay and after 8-10 minutes the chances are very slim [4]. Stroke is another kind of emergency situation where early response reduces the chance of death and disability. In fire accidents, the extent and heat increase rigorously with time and may go out of control after flashover (the point when all combustibles in the space have been heated to their ignition temperature and spontaneous combustion occurs). One third of deaths related to motor vehicle accidents can be prevented by ensuring quick medical help [5].

To ensure a faster response, traffic control systems provide preference for emergency vehicles over other vehicles. One such preferential traffic control system is EVP

(Emergency Vehicle Preemption), which is used to ensure quick and safe movement of emergency vehicles at signalized traffic intersections. Many types of EVP systems have been deployed in different cities. These systems usually use sensors to detect emergency vehicles and interrupt the regular traffic signal to offer green indication for Emergency vehicles.

Although EVP reduces emergency response time and emergency vehicle crashes dramatically, there are still some limitations to overcome. As EVP disrupts the regular traffic signal operation, it has a negative impact on general traffic particularly near the emergency facilities (i.e., hospitals, trauma centers, and fire/rescue and EMS stations) in big cities. Frequent use of preemption in peak hours can increase traffic congestion. Also, EVP system doesn't work effectively with multiple simultaneous emergency requests. When multiple emergency vehicles preempt in an intersection within a short period of time, average travel time for general traffic is increased significantly [6]. Conventionally, EVP systems serve on a first come, first served basis. Sometimes we need to prioritize between multiple emergency requests based on the type of emergency situations. For example, consider a situation where a police car and an ambulance which is rushing to respond to a sudden cardiac arrest patient are requesting for right of way at the same time at the same intersection. If both of these vehicles are seeking for conflicting approaches, only one car can get the right of way immediately. Clearly, here the ambulance needs to get higher preference to save the life of the patient even if the police car is the first who requested for preemption. This kind of situation may seem very rare but there are several recorded incidents when multiple emergency vehicles collided with each other [7] [8] [9]. Implementing a priority strategy among different types of emergency vehicles not only can save response time but it also can reduce the chance of emergency vehicle collision. There is a high probability of having multiple preemption requests at the same intersection in case of a major accident (e.g. train derailment, fire accident, bus crash) or natural disaster (e.g. earthquake) where many emergency vehicles rush to respond to the same scene at the same time.

1.2 Research Scope

Reducing traffic delay is a challenge for EVP implementation. Different research works use different techniques to confront its adverse effect on general traffic. In this thesis, we have addressed this issue using a new approach. We have used TSP (Transit Signal Priority) techniques (i.e. early green, green extension, phase rotation) which are used for transit buses. A two-step algorithm is developed to use the right TSP technique at the right time to reduce the traffic delay. This algorithm is then extended to accommodate multiple request cases where multiple emergency vehicles ask for conflicting approaches. Multiple emergency requests were addressed in some other research, but no research work which serves conflicting requests based on their priority needs has been found. Using our algorithm, we can adjust time-spans of traffic signal phases to assist high priority EVs while keeping the traffic delay at a minimum level.

1.3 Objectives

To overcome the limitations of EVP, we have addressed following objectives to achieve.

- Develop a traffic control algorithm to minimize the impact on general traffic.
- Implement priority based on emergency situations over competing emergency vehicles in real time.

1.4 Thesis Organization

The reminder of the thesis is organized as follows:

- Chapter 2 presents a background study of currently available technologies and related works.
- Chapter 3 presents our traffic control algorithm and evaluation of the algorithm under different traffic volumes.
- Chapter 4 presents future works and concludes with a summary of the thesis.

Chapter 2

Background Study

2.1 Traffic Signal Control

Traffic signal lights are being used to control traffic flows for nearly 150 years and are one of the most important components of modern day traffic systems. Traffic signal controlling techniques can be divided into three major categories: pre-timed, actuated and adaptive.

2.1.1 Pre-timed Traffic Signal Control

Many state-of-the-practice pre-timed traffic control systems are operated based on the traffic flows at different times of the day, and a signal timing plan is predetermined for each of them. A day can be divided into 3-5 segments. The basic idea is that the traffic flow pattern is different in different times of the day and is relatively consistent within the same segment. Pre-timed traffic signal control is a low cost solution for traffic problems at intersections. But this is not robust for current day fluctuating traffic conditions where traffic patterns at an intersection can change even for the same time of day and day of week [10].

2.1.2 Actuated Traffic Signal Control

Actuated traffic signal control uses detectors to locate vehicles at different road segments of an intersection. The most common method is to use inductive loop detectors which are generally installed 40 m upstream of the stop line. The traffic signal control system uses this real-time measurement to execute vehicle actuation logic [11]. Actuated traffic control systems can be installed to operate in two modes: fully actuated and semi-actuated.

Fully actuated traffic signal control requires detectors installed on all approaches to the intersection. All signal phases have a minimum and maximum preset values to serve green indication based on current demand. In semi-actuated mode, detectors are installed only on the side street approaches. The traffic signals on the main street approaches remain green until a vehicle is detected on the side street. After detecting a vehicle, the side street detectors place a service call to signal controller which then decides an appropriate time to terminate main street green indication and serve the side street. Minimum and maximum green time is maintained for the side street movements. Minimum green time ensures the vehicle gets enough time to pass the intersection safely. Maximum green time is used to make sure that the traffic on the main street approaches does not wait for a long time. After serving the last vehicle on the side street or reaching the maximum allowed green time, the main street signal returns to the green state.

2.1.3 Adaptive Traffic Signal Control

Adaptive traffic signal control adjusts traffic signal timings based on the current traffic demand and system capacity. The system gets input from detectors, uses an algorithm to predict the future traffic stream and optimizes signal timing plans based on this prediction [12]. It requires extensive detection data from surface street detectors and communication between central and local controllers [13]. The Sydney Coordinated Adaptive Traffic System (SCATS) and the Split Cycle Offset Optimization Technique (SCOOT) are two of the most widely used adaptive traffic signal control systems all over the world. When traffic demand is less, SCATS minimizes overall traffic stops by coordinating among adjacent intersections which reduces average travel time. During peak hour when traffic demand is high, it maximizes traffic throughput and controls traffic queue formation at intersections [14]. On the other hand, SCOOT is continuously fed with real-time traffic measurements and run incremental changes to signal timings to optimize traffic throughput [15]. Adaptive traffic signal control systems improve traffic flow and ensure faster responses to traffic conditions. But these systems are complex and expensive to install, and have a high maintenance cost.

2.1.4 Isolated And Coordinated Traffic Signal Control

Based on the number of intersections coordinating to implement a traffic signal control strategy, traffic signal control system can be divided into two types: isolated and coordinated signal control.

Isolated traffic signal control does not depend on other intersections and solely depends on traffic controlling strategy installed on this intersection. On the other hand, coordinated traffic control system synchronizes multiple intersections to allow a platoon of vehicles to move without stopping. A timing relationship is established among successive intersections such that traffic travelling towards a certain direction at a predetermined speed, can pass through successive intersections with green indications [1]. These coordinated traffic signals at successive intersections are called a green wave. Figure 2.1 illustrates a time-space diagram where a timing relationship (offset) is established among intersections to make a green wave. Green wave reduces total stops, delays and hence less fuel consumption and emissions.

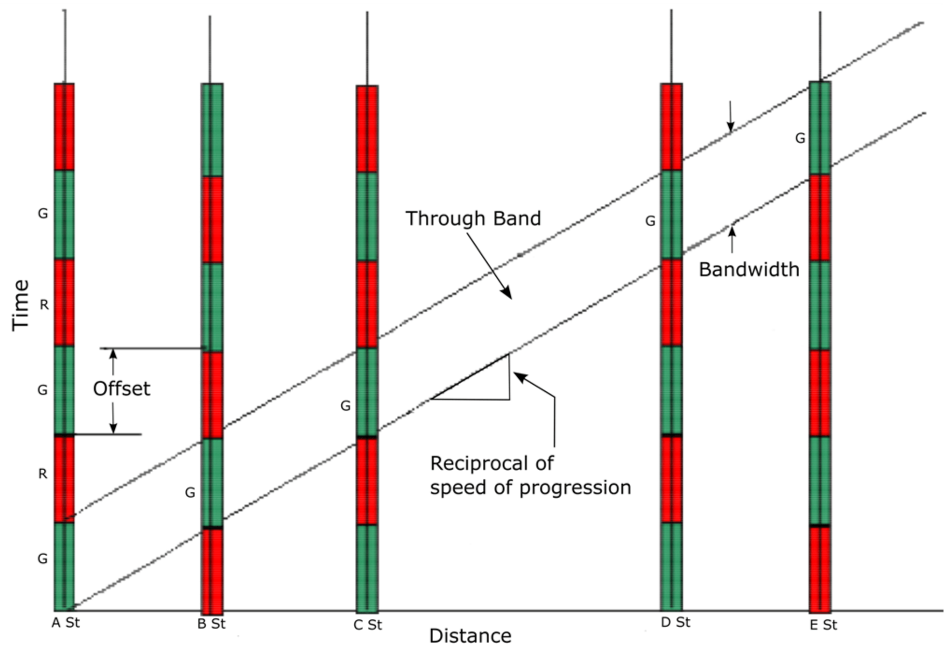


Figure 2.1: Traffic signal coordination on a time-space diagram [1]

2.2 Traffic Signal System

Traffic signal control depends on underlying infrastructure and available system components that work together to improve travel time and safety. A modern traffic signal system consists of following components.

Detectors: Detectors are used to track vehicles at various approaches to an intersection. Inductive loop is the most widely used detection technique which is embedded in the road's surface and senses metal objects from a running or a stopped vehicle by inducing currents. Video detection can provide us a wide-range detection data and relies on real-time image processing [16]. Infrared sensors, radar and magnetic detectors are some other types of vehicle detection techniques available.

Local Controller: Local controller is responsible for switching different signal indications electronically. It can take signalling decisions based on detector data or from the traffic control center.

Master Controller: Master controller is an optional component and can be used to establish a communication medium between local controllers and the traffic control center. It controls timing plans and monitors equipments for a group of intersections [2].

Traffic Control Center: Traffic control center stores controller data, monitors the traffic signal control system, and applies changes to current signal decision parameters. Adaptive and coordinated systems rely heavily on traffic control center operations.

2.3 Traffic Signal Terminology

A number of traffic signal terminologies which are important for signal designing are discussed below.

Cycle: Traffic signal cycle refers to a complete sequence of green indications for all approaches to the intersection. Cycle length is the total time needed to complete the sequence. Figure 2.2(a) shows a four-way intersection with eight approaches. A traffic signal cycle represents the sequence of green indications for all of these approaches.

Interval: Change and clearance are two types of intervals which are used to indicate changes of traffic signal states. Yellow signal indication represents the interval for the changing state from green to red. After a yellow indication, a red indication occurs. After that, the controller waits for a moment before it starts a green indication for the next approach. This extra red indication before starting an approach is called clearance interval. Clearance interval allows vehicles to clear the intersection before starting a green indication for a conflicting approach.

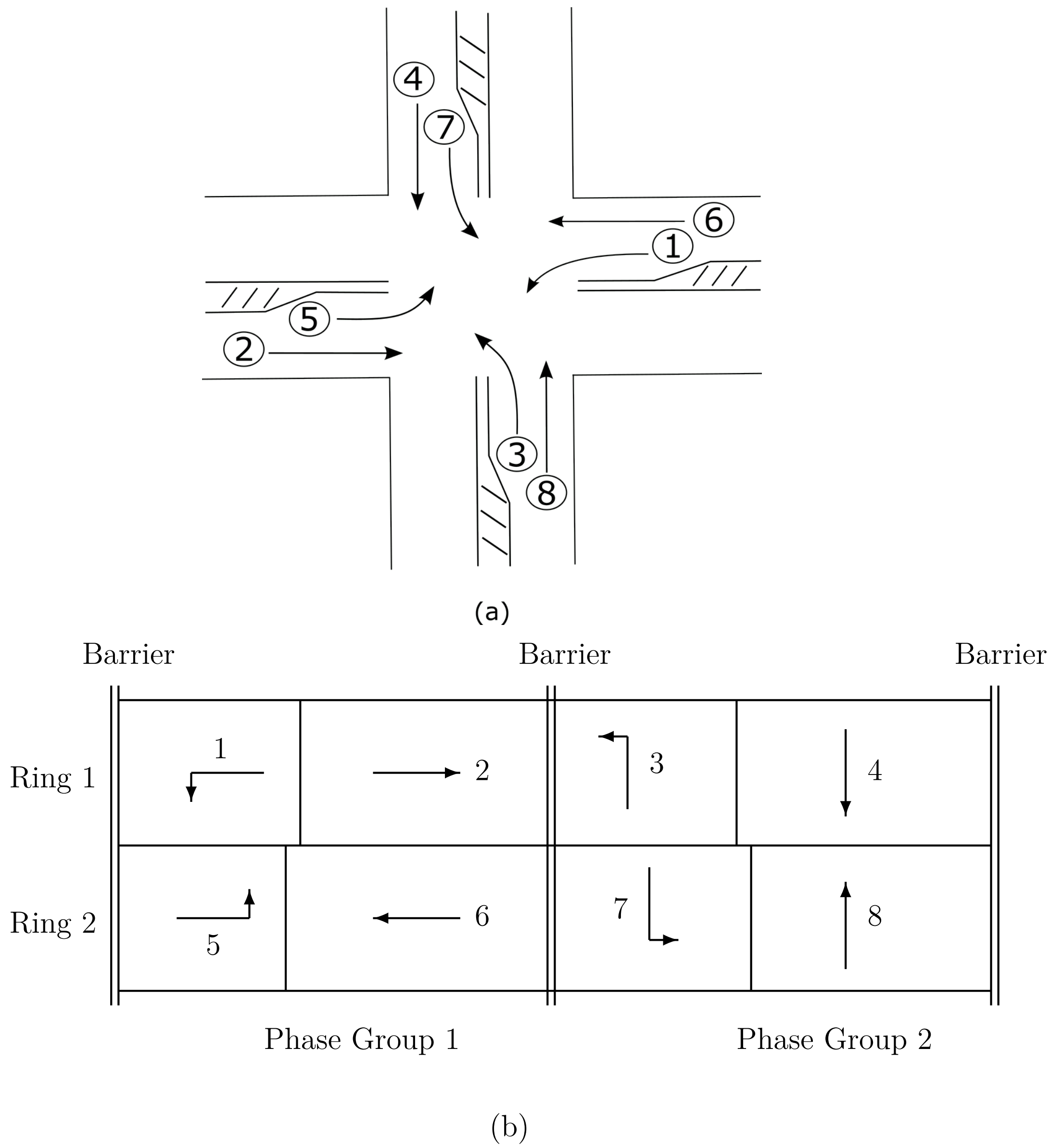


Figure 2.2: Phase, ring and barrier in a dual ring traffic controller

Phase: A phase is a part of the cycle allocated to a particular traffic movement to the intersection. It consists of green, change and clearance intervals for this movement. More than one phase can be served at a time, and these phases are called concurrent or non-conflicting phases. Concurrent phases may have different phase durations in a cycle. In figure 2.2(b), phase 1 and phase 5 are concurrent phases where phase 1 has a longer phase duration compared to phase 5. Two phases with conflicting movements that cannot be served together are called conflicting phases. Phase 1 and phase 2 are two conflicting phases.

In figure 2.2(a), phases 1, 2, 5, and 6 are same-road (eastbound-westbound) phases. Similarly, phases 3, 4, 7, and 8 are another same-road phases (northbound-southbound). A phase may or may not be in conflict with same-road phases, but this phase is always in conflict with other road phases. For example, phase 1 and phase 5 are not in conflict, but phase 1 conflicts with all phases from the other-road (phases 3, 4, 7, and 8).

Phase Number: NEMA (the National Electrical Manufacturers Association) maintains a standard numbering system for phases which can be used by all traffic controllers. For an eight phase, four-way intersection, odd numbers are assigned for left turn phases, and even numbers are for through phases. Figure 2.2(a) shows a standard four-way intersection where all the phase movements are numbered according to NEMA standard.

Dual-Ring Controller: A ring is a sequence of conflicting phases in a cycle which occur in a fixed order. Dual-ring controller contains two interlocked rings that can run concurrently. Figure 2.2(b) depicts a phase sequence diagram of a four-way intersection where two rings with all the phases are shown in a cycle. Here, ring 1 consists of phases 1,2,3 and 4 and ring 2 contains the rest of the phases. All phases in the same ring are in conflict with each other. So they are served sequentially, not concurrently. For example, eastbound-westbound phases (phases 1,2,5, and 6) are divided between two rings. Ring 1 contains phases 1 and 2 while ring 2 contains phases 5 and 6. Phase 1 and phase 2 are in conflict with each other if served together. So they are served one after another and stay in the same ring. Similarly, phase 5 and phase 6 are conflicting phases and they stay in ring 2. But these two phases are not in conflict with phase 1 and phase 2. So, phase 1 and phase 2 can be served concurrently with phase 5 and phase 6 [figure 2.2(b)].

Barrier: A barrier is a reference point where rings are interlocked in a dual-ring controller. At this point, rings must finish current phases and red is shown for all approaches to implement clearance interval. Barrier prevents conflicting other-road phases to run concurrently. Figure 2.2(b) shows barriers in the phase sequence diagram where a barrier separates eastbound-westbound phases (phase group 1) from northbound-southbound phases (phase group 2).

Phase Group: All the same-road phases are called a phase group. Phase groups are separated by a barrier in a dual ring controller.

2.4 Intelligent Transportation System

ITS (Intelligent Transportation System) refers to the application of information and communication technologies in transportation to save lives, time and the environment [17]. The ever increasing urban growth leads to the problem of traffic congestion which causes billions of hours wasted each year just in the US [18]. Safety and environment are other factors which need to be addressed to ensure better and more secure transportation system for our current and future world. Limited budget and land use constraints have resulted in a shift toward better management of existing infrastructure [17]. With the advent of the sensor and computer technologies, ITS offers leading-edge techniques to achieve safety, productivity, efficiency and mobility within the transportation sector.

ITS uses sensors and wireless networks to collect data and algorithms to process these data to take decisions in real-time [19]. Connected vehicle is an emerging ITS technology to connect all the neighbouring vehicles with each other and with the infrastructure. ITS offers services for all modes of transportation including the navigation system, air traffic control, water transportation, railroad and highway traffic. Some applications of ITS include Traffic Management Systems (TMS), Emergency Vehicle Preemption (EVP) and Transit Signal Priority (TSP) [20].

2.4.1 Connected Vehicle Technology

Each year, road traffic accidents cost us 1.25 million lives globally [21]. In 2015, the US lost 35,092 people due to crashes which is a 7.2-percent increase compared to 2014 [22]. To prevent roadside crashes, connected vehicle technology can play a significant role by increasing situation awareness. Safety features such as automated braking, forward collision warnings, traffic warnings, hazard warnings and blind spot detection can help the driver to avoid crashes [23]. Traffic congestion is another problem where extra travel time and fuel cost US cities 160 billion dollars in 2014 only [24]. Connected vehicle can provide efficiency by on-board driving assistance such as optimal speed advisory, optimal route guidance, and advanced warning which can enhance traffic flow and produce less congestion. By ensuring fuel efficiency, connected vehicle can help reduce pollution and total carbon emission.

A vehicle can get connected through vehicle-to-vehicle(V2V) and by vehicle-to-infrastructure(V2I) communications. In both V2V and V2I, safety-related applications (e.g. advanced warnings) generally use dedicated short-range communications technology(DSRC) which is fast, secure, and reliable. Other applications use different wireless technologies such as 3G or LTE [25].

2.4.2 Emergency Vehicle Preemption

Preemption refers to the transfer of regular traffic signal operation to a special traffic signal control. An EVP (Emergency Vehicle Preemption) system ensures that emergency vehicles get the right of way at and through a traffic intersection. Green indication is used on the desired approach by interrupting regular signal operation to provide the right of way to the emergency vehicle. Fire trucks, ambulances and police cars are the most common types of emergency vehicles which need support from preemption control. Figure 2.3 shows an example of EVP where the detector on the signal head detects the EV and on board controller at the intersection switches to preemption.

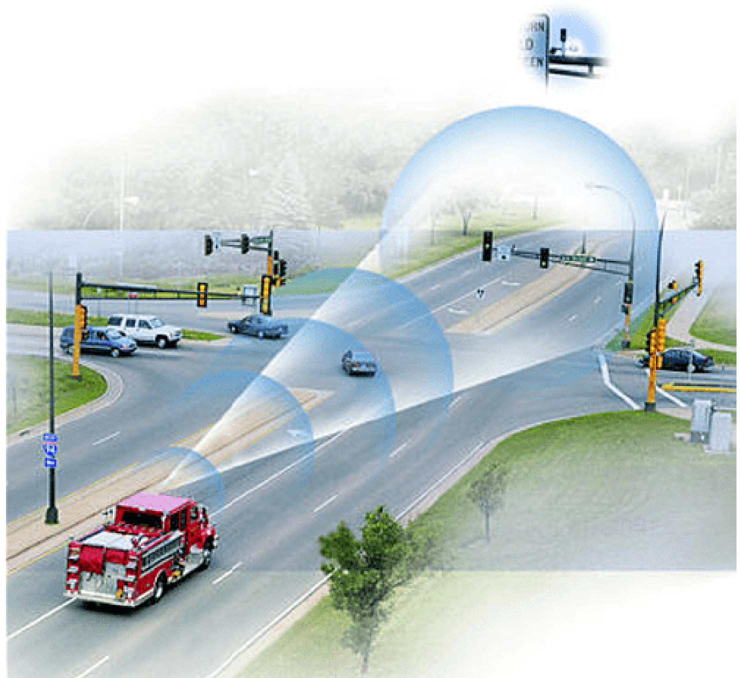


Figure 2.3: Emergency Vehicle Preemption at an intersection [2]

The concept of preemption control was first introduced in 1929 when the American Engineering Council published "Street Traffic Signs, Signals, and Markings" where they described the need for preemption control for emergency vehicles [26]. Technology for preemption control started to evolve in the late 1960s. In the early 1970s, 3M developed a preemption system using pulses of strobe lights received by detectors attached to the traffic signal head. In 1979, they introduced two different kinds of priority, preemption for higher priority (e.g. emergency vehicles) and TSP (Transit

Signal Priority) for lower priority requests (e.g. transit buses). Encryption was added in 1992 to get rid of hackers [27].

Various types of preemption systems including light-based, infrared-based, sound-based, and radio-based systems are installed in different cities [26]. Optical systems use light or infrared emitters which are different from emergency lights and usually mounted on the roof of the EV. A detector typically mounted on the signal head detects light or infrared from the EV and the traffic signal controller serves the request. Optical systems require a clear line-of-sight to function properly [28]. A Sound-based system does not require any special equipment for the EV. The EV siren works as the emitter and a directional microphone mounted on the signal pole detects the approaching EV [29]. Directional microphones can determine the direction of an incoming siren. Radio-based systems require the EV to have an emitter which can transmit digitally coded radio signal. The direction of preemption is encoded in the radio signal and an omnidirectional antenna installed at the intersection detects the radio transmission. Once a radio signal is detected, the direction of preemption is determined, and the traffic controller at the intersection processes the request [26]. Unlike optical systems, sound-based and radio-based systems do not require clear line-of-sight. Along with these systems, a combination of cellular and GPS technologies are available today which can pinpoint a vehicle's location and speed, and a centralized system can monitor and take decisions based on these data [27]. The benefits of preemption control include faster response to emergency needs and improved safety for emergency vehicles and general traffic.

2.4.3 Transit Signal Priority

TSP (Transit Signal Priority) is an operational strategy that helps to reduce transit vehicle delay at signalized intersections. In TSP, transit vehicles (e.g. buses) communicate with traffic signals to get priority over general vehicles. Traffic signals at intersections respond to TSP calls by altering signal operations. Green extension and early green are two basic types of TSP strategies to adjust signal timings.

Although TSP and EVP (Emergency Vehicle Preemption) use similar equipment to alter traffic signal operations, there are differences between these two strategies. TSP *modifies* the current signal operation to accommodate priority requests, while EVP *interrupts* the current signal operation and provides a green indication on the desired approach for emergency vehicles. With TSP, priority request may not be granted immediately to maintain normal traffic operation.

Transit signal priority provides better schedule adherence, reliability and reduced travel time for transit vehicles. For instance, Los Angeles reduces 25 percent of bus travel time by using TSP [30]. By improving the quality of public transport, TSP encourages the usage of the public transportation system which leads to less traffic congestion and hence less fuel consumption and less carbon emission.

2.5 Other Relevant Topics

Apart from TSP technologies (e.g., green extension, early green, etc.), there are some other important factors that we rely on priority control implementation. Arrival time prediction, queue discharge time, and maximum and minimum green time are three important issues related to priority control research. There has been a good number of research works done on these topics.

Arrival time prediction: Predicting arrival time at the intersection is an important component in signal priority control. The conventional approach uses a detection technique near the intersection and predicts the arrival time to apply priority control accordingly. Recent advancements in connected vehicle and GPS technologies allow us to detect the vehicle early and predict the arrival time using real-time traffic data and vehicle location information. Early detection and accurate arrival time prediction are important issues to select optimal priority strategies [31]. Tan et al. [32] developed an arrival time prediction algorithm using a historical model and an adaptive model which adjusts itself based on real-time data. In this approach, detector data is replaced by real-time GPS vehicle location information and they focused on minimizing the prediction error to be less than $\pm 5s$. Liu et al. [33] proposed a virtual probe model to estimate arterial travel time at congested conditions. They used a queue length estimation method and traffic signal data to estimate arterial travel time more accurately in congested scenarios. Fagan and Meier [34] presented a technique which increases arrival time estimation accuracy by considering driver travel behavior and traffic flow pattern. Predicting arrival time for emergency vehicle needs consideration of some unique characteristics that EVs possess. EVs can break general traffic rules and get higher priority over ordinary vehicles. Budge and Ingolfsson [35] analyzed how emergency vehicle travel time varies with distance. Westgate et al. [36] presented a Bayesian model to estimate the distribution of ambulance travel times using error-prone GPS data. Zhang et al. [37] developed a utility model to evaluate the travel time performance of emergency vehicles. Wang et al. [38] proposed a travel time estimation model for an emergency vehicle under preemption control.

Queue discharge time: An uninterrupted movement of the emergency vehicle requires the traffic queue at the intersection to be discharged before the EV arrives. To know when to start releasing the queue, we need to know the queue length and the required time needed to discharge. Liu et al. [39] presented a real-time queue length estimation model by applying Lighthill–Whitham–Richards (LWR) shockwave theory with the traffic signal data. This approach works reasonably well even under congested condition with long queues. Based on this model, they developed a method for queue discharge time calculation [33]. Zhan et al. [40] utilized the M5P tree algorithm for predicting lane clearance time particularly for any traffic incident which considers the time of the day, number of blocked lanes, and incident information.

Maximum green time and minimum green time are another two important factors to achieve traffic safety. Minimum green time ensures adequate pedestrian crossing time and allows a vehicle to pass the intersection safely. On the other hand, maximum green time limits the time that a phase can be extended to serve a priority request. If we extend the green time beyond a certain limit, motorists can get confused which is a concern for traffic safety. It also has a significant negative impact on high traffic volume scenarios [41]. Zhang and Wang [42] presented a stochastic model to optimize maximum and minimum green times dynamically by using queue lengths and traffic arrival data.

2.6 Literature Review

EVP (Emergency Vehicle Preemption) has a great potential to reduce emergency response time. Study shows that EVP reduces emergency vehicle travel time by 14 to 23 percent in the US [26]. Another study was conducted in Kanazawa city in Japan, where they observed 54 seconds of reduction in average response time which leads to a significant improvement in survival rate for out-of-hospital cardiac arrest patient [43]. Kamalanathsharma and Hancock [44] proposed a congestion-based preemption control where travel time can be reduced as much as 31 percent.

Despite the extraordinary effectiveness, EVP has some problems which need to be addressed. One of the significant problems is the adverse effect on general traffic. Teng et al. [45] evaluate this problem in Las Vegas city focusing on traffic speed variation during preemption operations. A significant impact on cross street vehicles was identified during peak hour. Multiple preemption operations within a short period also have a contribution to increased traffic delay. Yun et al. [46] [47] analyzed the impact on traffic signal coordination using various preemption strategies. They used Hardware-In-the-Loop Simulation (HILS) with VISSIM to simulate traffic controllers under different traffic demands. They found that a single emergency preemption event can cause a significant impact on traffic signal coordination. They suggested that the transition method from preemption to the normal signal process should get some careful attention to recover signal coordination effectively.

Several researchers have worked to overcome these problems. Qin and Khan [48] developed two new control strategies that enable the signal transition from normal operation to EVP and from EVP to normal operation. The main goal of their research was to ensure a safe passage for the EV at its operating speed and to decrease the impact on general traffic. Kang et al. [49] presented a signal coordination approach where green wave is achieved along the emergency vehicle's route by changing coordination setting (offset value). Noori et al. [50] proposed a connected vehicle approach where traffic signals are preempted in time to clear queue at the intersection in a traffic congested area. Gedawy [51] proposed a D* lite path planning algorithm to select routes based on real-time traffic updates and a preemption strategy for high and low level congestion situation to minimize traffic disruption.

Safety is another major concern for operating emergency services. An emergency vehicle crash poses extra trouble to those who are already in misery and significantly reduces the capacity of local emergency service operations. Several studies have been conducted to evaluate the cause and prevention of emergency vehicle crashes [52] [53]. Violation of driving rules [54] and confusion at intersections [55] are some of the main reasons for these kinds of crashes. To mitigate these problems, an emergency vehicle alert system was developed to increase driver awareness [56]. Buchenscheit et al. [57]

proposed an emergency vehicle warning system based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication where drivers can get detailed information about position, speed and route of the approaching emergency vehicle earlier than the current system. Unibaso et al. [58] presented a preemption algorithm where they used information from cooperative awareness messages (CAM). The CAM protocol is a part of the ITS vehicular communications architecture and CAM messages contain safety relevant status information such as vehicle position, speed, acceleration or heading [59]. Huang et al. [60] used timed Petri nets (TPNs) to model an EVP system to ensure safety by avoiding urgent spectacles at intersections.

Effectively accommodating multiple preemption requests is another research area which needs proper attention. Very few references have been found that address this issue. Head et al. [61] presented a mixed integer nonlinear (MINP) mathematical model for multiple priority requests where total priority delay is minimized. However, this model has some problems to resolve. The MINP model uses a solver which takes relatively long time to solve a real-time problem. Another problem is the deterministic arrival time, which is not always realistic for emergency vehicles. He et al. [62] extended this idea for multi-modal traffic signal control while considering signal coordination at the same time. Multi-modal traffic control system includes TSP (Transit Signal Priority) and EVP (Emergency Vehicle Preemption) together where it considers TSP as a conditional priority and EVP as an unconditional priority. A mixed integer linear program (MILP) was formulated and total priority delay was minimized by using an optimization solver called CPLEX. They proposed another heuristic algorithm [63] for multiple priority requests at isolated intersections where the multiple priority problem was transformed into a network cut problem. This algorithm is solver-free and easy to implement in an embedded environment. All of these research works were focused on minimizing the total priority delay and do not guarantee serving the highest priority vehicle first. Other algorithms [64] [65] have addressed the multiple priority problem for transit buses with the advent of TSP technologies.

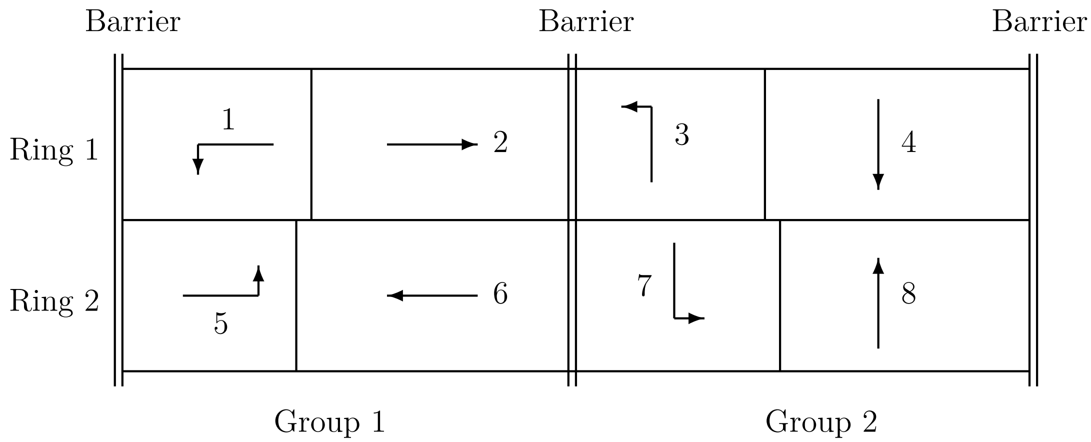
Chapter 3

Our Algorithm

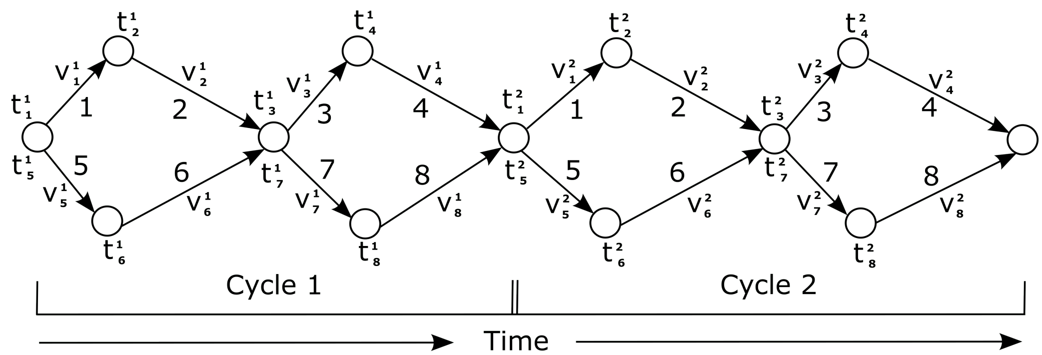
Based on the characteristics of an emergency vehicle request, we have used transit signal priority strategies to serve the desired phase. In our approach, preemption is used only when other TSP strategies don't work to allow the requested phase at the arrival time. This strategy will make sure that the emergency vehicle will get the requested phase and at the same time, the adverse effect on the general traffic will be minimized.

First, we design a traffic control algorithm for single emergency request. The algorithm is extended for multiple emergency requests afterwards. For multiple requests scenario, our main goal is to prioritize and allow requested phases to those who have higher priority needs.

A *precedence graph* presents a visual way for depicting traffic schedule and emergency requests [61]. We have used this graph to analyze our algorithm on different scenarios. A traditional eight phase dual-ring controller is modeled in a precedence graph in figure 3.1. Each edge in a precedence graph represents phase duration while node represents phase transition. Other controller components (ring, barrier, cycle) are also visible in this representation.



(a)



(b)

Figure 3.1: Precedence graph representation of a 8 phase, dual-ring controller

Here t_i^x represents the starting time of phase i in xth cycle. v_i^x presents the duration of phase i in cycle x where v_i^x comprised of green time g_i^x, yellow time y_i and clearance interval r_i. Here, we do not use cycle number for yellow and clearance interval, as yellow and clearance interval do not change from one cycle to another at an intersection. In this representation, timing starts from the beginning of a cycle with t=0 and can be written as:

$$t_1^1 = 0$$

$$t_5^1 = 0$$

$$t_2^x = t_1^x + v_1^x$$

$$t_6^x = t_5^x + v_5^x$$

$$t_3^x = t_2^x + v_2^x = t_6^x + v_6^x = t_7^x$$

$$t_4^x = t_3^x + v_3^x$$

$$t_8^x = t_7^x + v_7^x$$

$$t_1^{x+1} = t_4^x + v_4^x = t_8^x + v_8^x = t_5^{x+1}$$

An emergency request denoted as R_i^j represents a request which is active and seeks a green indication for phase i. The superscript j denotes an identification number to distinguish among active requests. Estimated arrival time at the intersection and requested phase are two basic properties of an emergency request. For our algorithm, we need to add one more property - priority weight denoted as W^j. The main purpose of this property is to differentiate and prioritize among emergency vehicles in case of conflicts. The value of this property is to be determined by emergency responder based on the emergency scenario.

There is a high level of uncertainty associated with estimating arrival times due to road condition, traffic congestion, time of the day, weather condition, etc. To ensure robust optimization [66], we consider estimated arrival times as a span of time rather than a point in time. In figure 3.2, we have used precedence graph to depict an emergency request in a traffic signal schedule. Request R_4^1 has id=1 in the active request queue and seeking for phase 4 on its arrival. The arrival time has a lower bound, $\underline{R}_4^1 = 87$s and an upper bound, $\overline{R}_4^1 = 94$s which means the emergency vehicle has much higher probability of arrival during this period. For this example, we do not need to alter our signal schedule here for R_4^1 as the schedule is already serving the requested phase (phase 4) on emergency vehicle's arrival time.

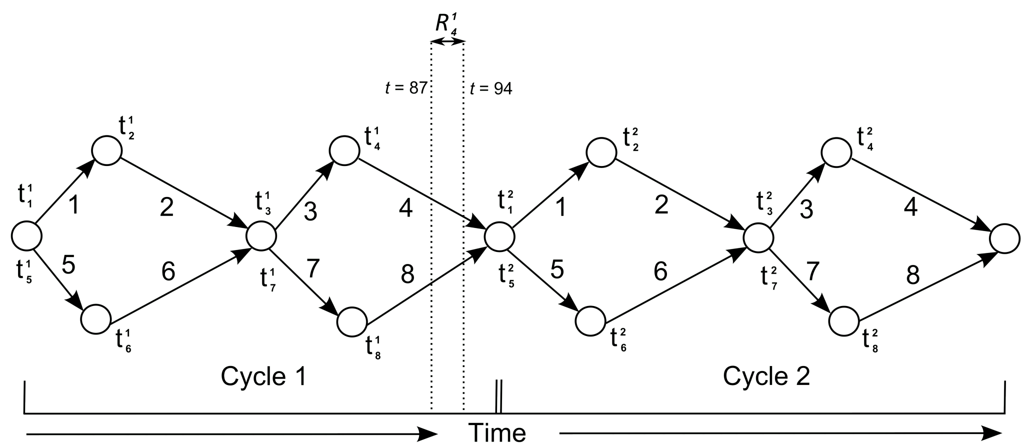


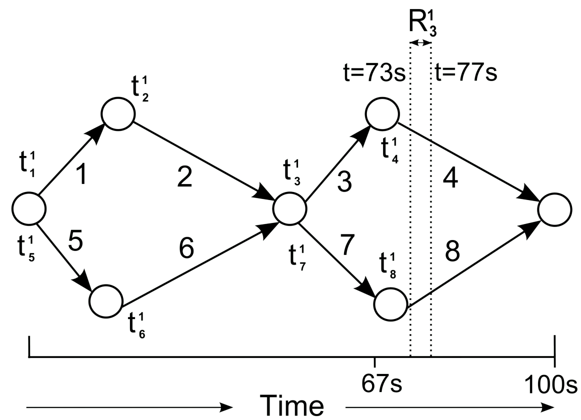
Figure 3.2: Precedence graph representation of an emergency request

3.1 Traffic Signal Priority Strategies

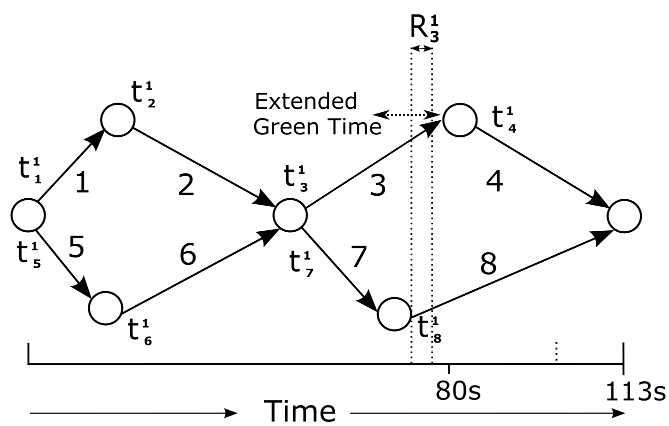
Transit Signal Priority (TSP) and Emergency Vehicle Preemption (EVP) are two different kinds of priority strategies aimed at implementing priority control in different situations. Each of these two strategies has different characteristics and limitations. To improve the signal preemption system for emergency vehicles, we have used TSP techniques (green extension, early green, and phase rotation) along with preemption to develop our signal control algorithm. These techniques are described below.

3.1.1 Green Extension

Green Extension strategy extends the green time to serve the priority phase request. This strategy is highly effective when the emergency vehicle arrives just after the requested phase.



(a) Signal schedule with a priority request



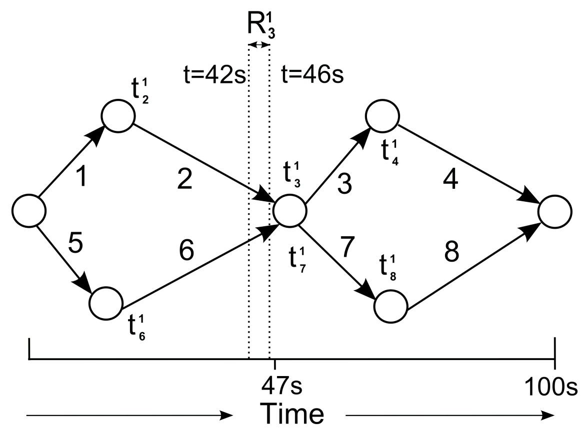
(b) After using Green Extension strategy

Figure 3.3: Green Extension for a priority request

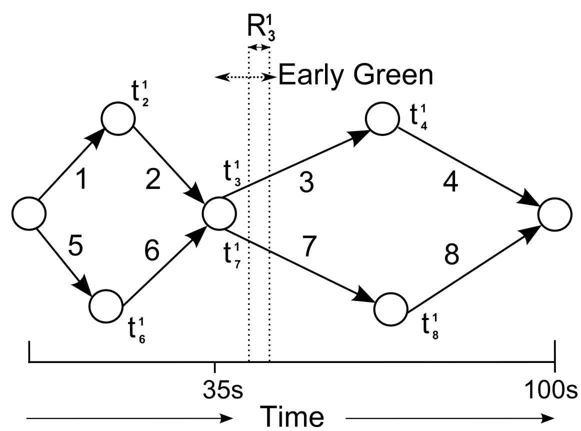
In figure 3.3(a), we have a priority request starting at 73s for phase 3 while phase 4 should be in service. Phase 3 ends at 67s here, but the request could easily be served extending phase 3 for a few more seconds. The traffic signal schedule is shown in figure 3.3(b) after using green extension strategy.

3.1.2 Early Green

Early green truncates the preceding phases to serve a green indication for the requesting phase. This strategy is very effective when the emergency vehicle arrives a few seconds before the requested phase.



(a) Signal schedule with a priority request

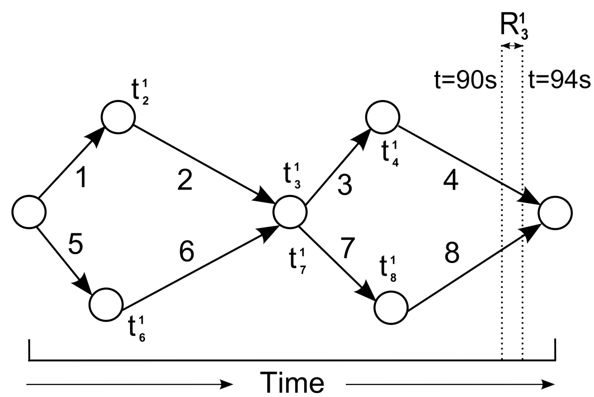


(b) After using Early Green strategy

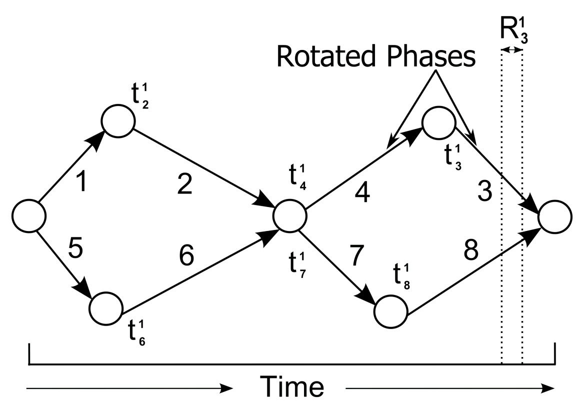
Figure 3.4: Early Green strategy for a priority request

An example is shown in figure 3.4 where phase 3 starts 12s early to serve the priority request. As a result, green time for phase 2 and phase 6 are shortened to accommodate phases 3 and 7. With early green strategy, green time may need to start a few seconds earlier than the emergency vehicle arrival time. This extra green time is used here for queue discharge which is needed to clear traffic congestion before the priority vehicle arrives at the intersection.

3.1.3 Phase Rotation



(a) Signal Schedule with a priority request

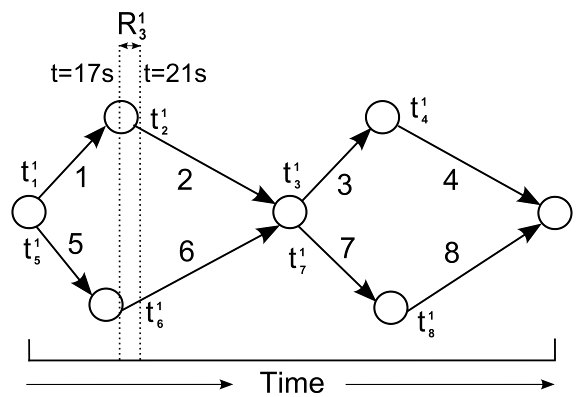


(b) After using Phase Rotation strategy

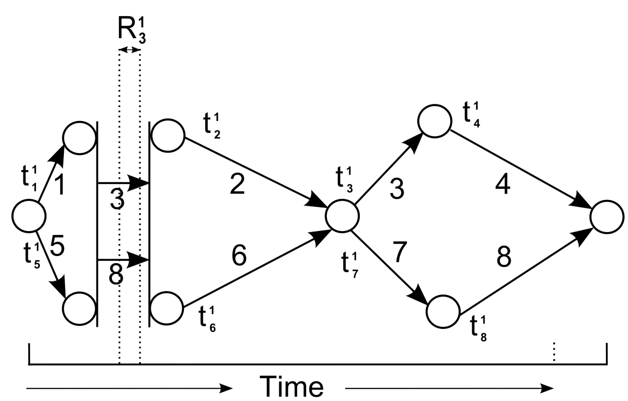
Figure 3.5: Phase Rotation strategy for a priority request

Phase rotation strategy alters the order of two conflicting phases in the same phase group to serve the priority request. In figure 3.5(a) we have a request for left-turn phase (phase 3) while the through phase (phase 4) should be in service. Phase 3 and phase 4 are two conflicting phases in the same ring in the same phase group. Figure 3.5(b) shows the traffic signal schedule after using phase rotation to serve the priority vehicle.

3.1.4 Preemption



(a) Signal schedule with a priority request



(b) Signal schedule after Preemption

Figure 3.6: Preemption for emergency vehicle

In figure 3.6 an example of EVP (Emergency Vehicle Preemption) is shown where phase 3 is requested just a few seconds after phase 1. None of the priority strategies discussed above can work here to serve the request. We need to use preemption which halts the current traffic signal schedule and serves phase 3 during the arrival of the emergency vehicle. Same statements are true if the EV requires any phase from the other phase group (phases 3,4,7,8) in this situation.

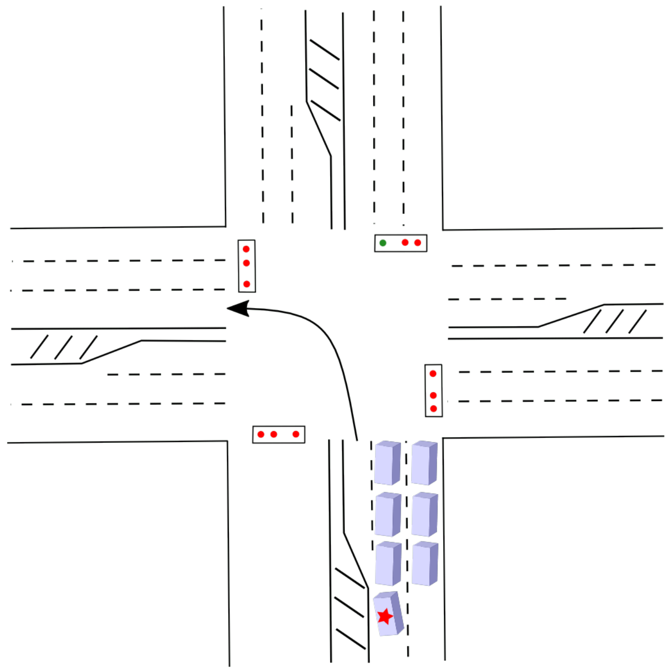


Figure 3.7: Problem with preemption phase only green indication

Figure 3.7 shows an example where the left turn lane is blocked due to through phase traffic when only phase 3 (left turn phase) is served for preemption. To clear the traffic, phase 8 (through phase) is served in figure 3.6 instead of phase 7 with phase 3.

Traditionally in traffic signal schedule, a phase group starts with left only phases followed by through phases, especially for main road traffic. But in preemption, same road phases (left turn only phase and through phase) are served together to clear the road for the emergency vehicle. In our approach, we have used this technique along with the traffic signal priority techniques. For example, green extension strategy discussed above is modified in figure 3.8 where both the same road phases (phase 3 and phase 8) are served concurrently for the emergency vehicle.

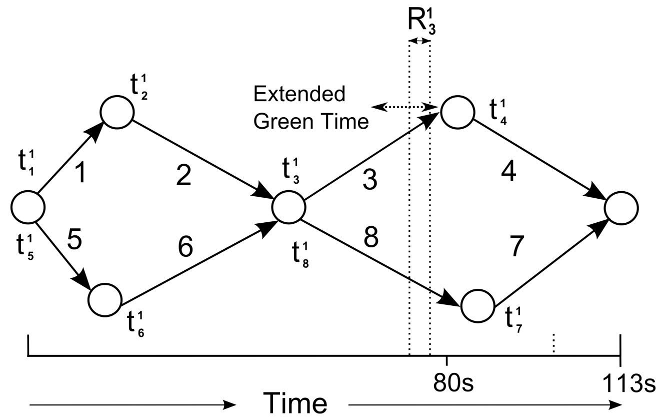


Figure 3.8: Green Extension strategy after phase adjustment

The maximum green time extension requirement can be relaxed for an emergency vehicle. In transit signal priority, a transit vehicle possesses moderate priority over other vehicles and a preset value for maximum green time ensures minimum disruption for other approaches. On the other hand, an emergency vehicle possesses the highest priority and signal preemption halts the regular traffic to allow the emergency vehicle to pass through the intersection within a minimum amount of time. Moreover, transit signal priority is much more frequent event than signal preemption. In our approach, we have used maximum green time for emergency vehicle, which is different from maximum green time for transit buses. The yellow and all-red vehicle clearance need to be applied for transitions for all strategies above.

3.2 Our Approach

3.2.1 Signal Control Algorithm For An Emergency Vehicle (Single-Request Algorithm)

A traffic signal schedule for a traditional dual ring controller consists of continuous cycles each having 2 phase groups and 8 phases.

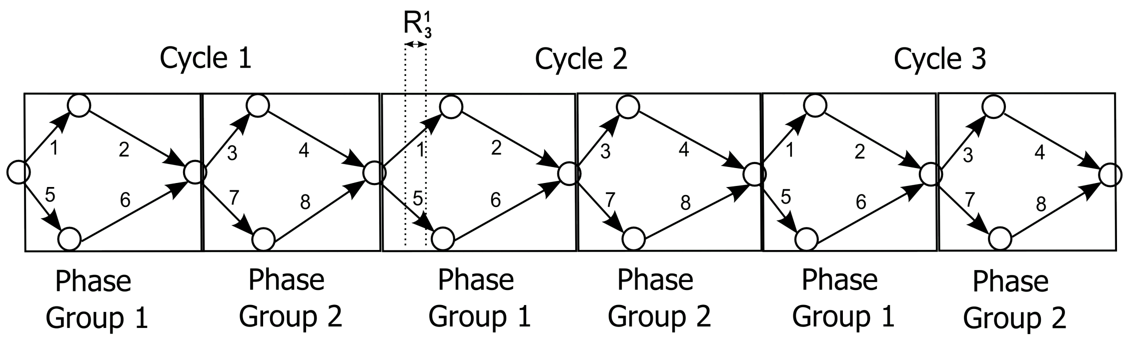


Figure 3.9: Traffic signal schedule

Figure 3.9 depicts an example of traffic signal schedule for three upcoming cycles. In this example, emergency request R_3^1 is seeking for phase 3 at its arrival on cycle 2. Clearly, current schedule can not serve the desired phase as different phases are active at that period. We need to build a traffic signal phasing algorithm using traffic signal priority strategies to serve this type of requests.

Step 1 . Accommodate emergency vehicle arrival time into the same phase group.

If the emergency vehicle does not get the required phase group at its estimated arrival time with the current schedule, we need to make some adjustments using a green extension or an early green strategy to ensure the required phase group. In our previous example in figure 3.9, the emergency vehicle arrives at phase group 1 while seeking for phase 3 which is in phase group 2. To get the emergency vehicle phase 3, we need to ensure that phase group 2 is in service at its estimated arrival time. We can do this by either extending phase group 2 of the first cycle (green extension) or by shortening cycle 1 and phase group 1 of cycle 2 so that phase group 2 of cycle 2 starts before the vehicle arrives (early green). To decide which option to take, we introduce a decision variable λ which represents the comparison between these two options. Other variables that are used to take the decision are described in the following list.

Lower bound of estimated arrival time, $\underline{R}_i^j$

Upper bound of estimated arrival time, $\overline{R}_i^j$

Required Phase Group, $R_i^j(PG) = m$

Start time of a Phase Group, $\underline{PG}$

End time of a Phase Group, $\overline{PG}$

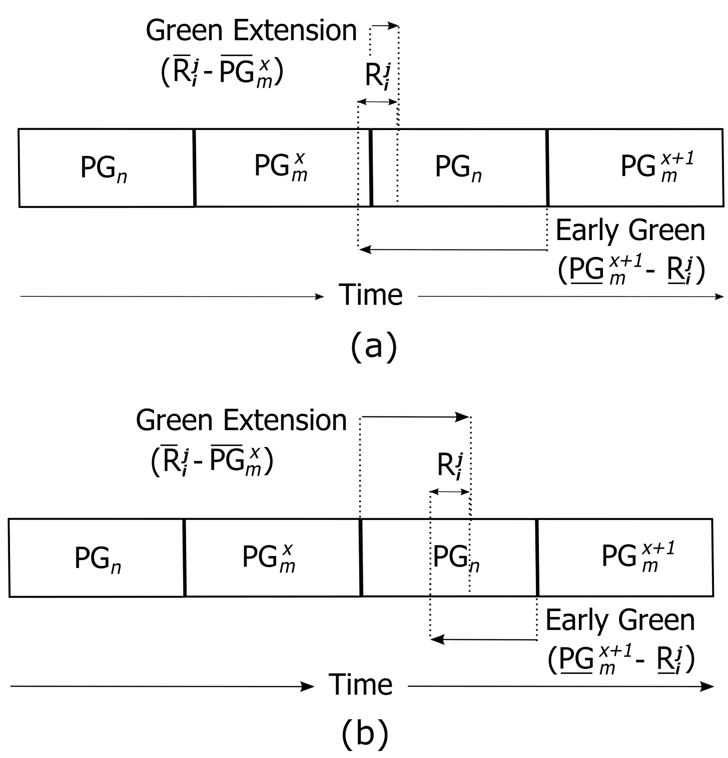


Figure 3.10: Green Extension vs Early Green for phase group

Figure 3.10 shows a generalized version of traffic signal schedule to visualize phase groups and emergency requests. The estimated arrival time of an emergency request may overlap between two adjacent phase groups [figure 3.10(a)], or it can stay inside a phase group [figure 3.10(b)]. The variables m, n represents phase group numbers where $m \neq n$ and $m, n \in \{1,2\}$. Here, x indicates the candidate cycle number for green extension. If the lower bound of an estimated arrival time starts at the same phase group which the emergency vehicle requires [figure 3.10(a)], then that phase group is the candidate phase group for green extension. And if the lower bound starts at the opposite phase group, then last occurrence of the required phase group is the candidate for green extension [figure 3.10(b)]. In these examples in figure 3.10, the emergency vehicle requires the phase group m (i.e. PG_m) while its estimated arrival time ends at the opposite phase group (PG_n). Here the previous phase group (PG^x_m) needs to get extended up to the upper bound of the estimated arrival time ($\overline{R}{}^j_i$) for green extension. Clearly, phase group m of the next cycle (PG^{x+1}_m) is the candidate for early green strategy.

$$\lambda = \frac{\overline{R}{}^j_i - \overline{PG}{}^x_m}{\underline{PG}{}^{x+1}_m - \underline{R}{}^j_i} \quad [\text{when } \overline{R}{}^j_i > \overline{PG}{}^x_m \text{ and } \underline{PG}{}^{x+1}_m > \underline{R}{}^j_i] \tag{3.1}$$

Here λ represents the comparison between a green extension ($\overline{R}{}^j_i - \overline{PG}{}^x_m$) and an early green ($\underline{PG}{}^{x+1}_m - \underline{R}{}^j_i$) strategy. If λ has a large value, it means the required time for an early green is less than the required time for a green extension. In this case, it is better to choose the early green strategy to ensure minimum disruption to other phases. And if λ has a smaller value, the required time for a green extension is less and green extension is better than the early green strategy. We can use a variable C to decide which option to take.

if $\lambda > C$ **then**
    apply Early Green on phase group
else
    apply Green Extension on phase group
end if

Green extension and early green have some distinct features. Traffic engineers may prefer green extension strategy as the phase gets enough time to clear the traffic congestion at the intersection before the emergency vehicle arrives. Early green and green extension do not introduce delay to the general traffic in the same way. And average vehicle delay is one of the most important factors while determining the best option. Some times early green creates much larger delay than green extension approach (figure 3.27). But we cannot extend the phase unnecessarily long which can disrupt the traffic from other phases. Also there is uncertainty involved with the estimated arrival time and this estimation can change over time. If the emergency vehicle arrives a little earlier than the estimated arrival time, we do not need any more change in the traffic schedule if green extension was selected. Because the phase has enough time before the emergency vehicle arrives and if the emergency vehicle arrives a little early, the arrival time is still inside the extended phase. On the other hand, if the emergency vehicle arrives later than the estimated arrival time and green extension was our choice, the required phase needs to be extended more unnecessarily. In this case, early green could be a better solution. Traffic engineers can determine the value of C considering the delay factor and emergency vehicle travel history data.

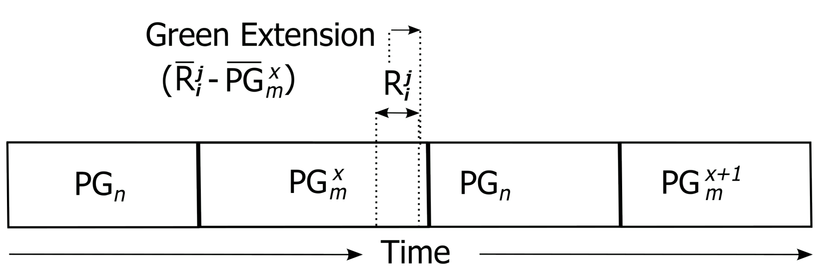


Figure 3.11: Applying Green Extension strategy for phase group

After *step 1*, the emergency vehicle gets the required phase group by either green extension or early green. Figure 3.11 shows an example of traffic schedule after *step 1*, where the required phase group (PG_m^x) is extended to accommodate emergency vehicle arrival time (R_i^j).

While applying early green strategy, there are different ways to truncate green signal duration from the phases. We can truncate equal time duration from each of the phase groups (i.e. all current and upcoming phase groups before the required phase), or we can truncate proportionally to their phase durations. The signal schedule from figure 3.9 needs to truncate green signal from three phase groups (both phase groups from cycle 1 and phase group 1 from cycle 2) for early green implementation. For example, if we need to start the required phase 30s early, we can shorten these three intermediate phase groups by 10s each. Another way is to truncate each of the phase groups proportional to their length (e.g. time duration). For example, if we have two intermediate phase groups where one phase group is 20% longer than the other, the phase group with longer time duration is reduced by 20% more. These phase groups can be truncated based on current traffic demand as well. With this approach, traffic controller truncates more green time with those phases which have lower traffic demand. Similar statements are true for a green extension as well. We can stretch the intermediate phases by adding an equal amount of time for each of the phases, or by adding time proportional to their duration, or we can add different time duration based on current traffic demand.

Throughout the implementation process, we always need to maintain minimum green time and maximum green time for emergency vehicle. In case of early green, we need to shorten previous phases in order to get some extra time for the required phase. But in order to do so, we should not compromise minimum green time as this may have a bad effect on the current traffic. If the phase group where the emergency vehicle arrives is currently in service, we do not have these two choices together at the same time to get the required phase group. In this case, we only have green extension if the emergency vehicle requires current phase group, and early green if it needs the next phase group. If we find neither early green nor green extension work for getting the required phase group, we need to use preemption to make sure the emergency vehicle gets the required phase. In case of preemption, the algorithm does not need to proceed to step 2.

Step 2. Arrange the phases inside the phase group to serve the emergency vehicle.

From *step 1*, the emergency vehicle gets the required *phase group* at its arrival time. In *step 2*, traffic signal priority strategies are used to ensure the required *phase* for the emergency vehicle.

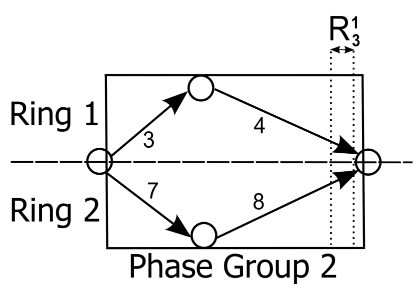


Figure 3.12: Phase group after step 1

An example is shown in figure 3.12 where the emergency vehicle has the required phase group, but not the required phase. In this example, the emergency vehicle is seeking for phase 3 while it arrives at phase 4. Phase rotation could be a good solution for this example. But in other cases, green extension and early green strategies could get us better results.

From figure 3.12, a request can be seen inside a phase group, and a phase group consists of two different rings. A ring from a phase group consists of two phases and these two phases can interchange with each other (phase rotation) unless any of them

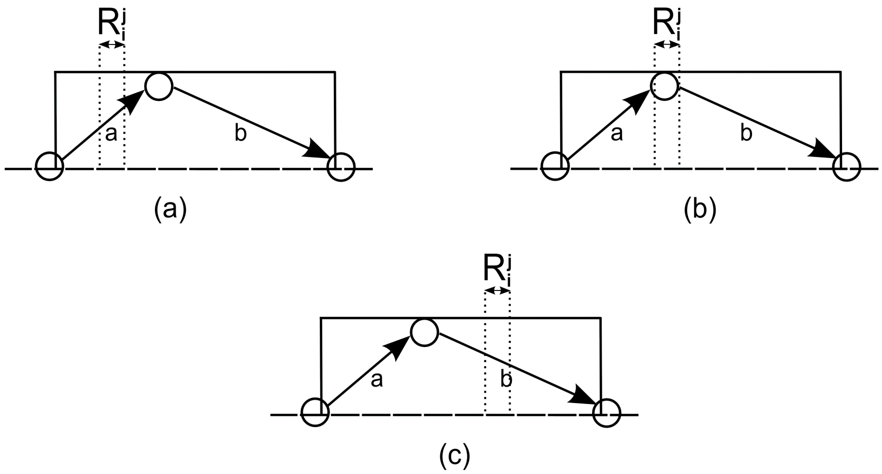


Figure 3.13: Different positioning of a request inside a ring

is currently in service (figure 3.14). The emergency request R_i^j seeks for phase i which can either be from ring 1 ($i <= 4$) or from ring 2 ($i > 4$). Figure 3.13 shows three different cases where a request can be seen inside a ring from a phase group. The emergency vehicle can arrive when phase a [figure 3.13(a)], or phase b [figure 3.13(c)] is in service, or it can arrive during the transition of these two phases [figure 3.13(b)]. From *step 1*, if the emergency vehicle already has the required phase at its estimated arrival time, we don't need to proceed to *step 2* and can omit this step. Otherwise, we need to find a way to assign the required phase at the estimated arrival time.

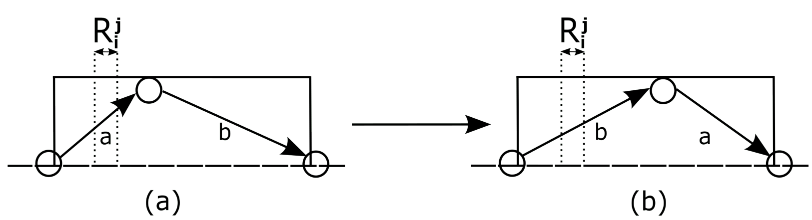


Figure 3.14: Phase Rotation inside a ring

Figure 3.14 shows phase rotation strategy inside a ring which can solve many requests especially if the emergency vehicle arrives at the beginning, or at the end of the phase. In this example, if the emergency vehicle requires the first phase of the phase group (if $i = a$), then we don't need to do anything else as the emergency vehicle gets the required phase at its arrival time [figure 3.14(a)]. On the other hand, if it seeks for the second phase (if $i = b$), then after applying phase rotation, the emergency vehicle gets the required phase [figure 3.14(b)]. But only phase rotation can not ensure the required phase in every case.

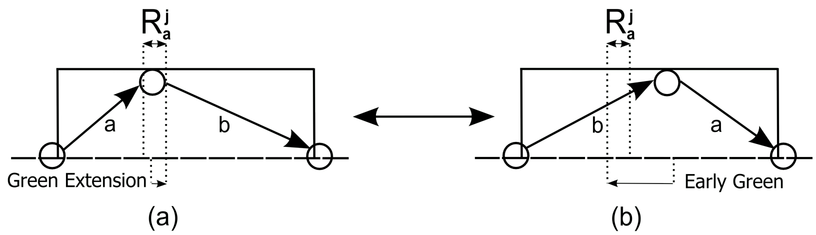


Figure 3.15: Green Extension vs Early Green for a phase inside a ring

In this example shown in figure 3.15, the emergency vehicle seeks for phase a at its arrival time. Regular schedule inside the ring [figure 3.15(a)] and the rotated schedule [figure 3.15(b)] do not serve the required phase a at its arrival time. We need to use either green extension or early green (after phase rotation) to get the desired phase. To do so, another decision variable λ_p is used here to compare between these two options.

```
1: if i = a then
2:     set k = a, l = b
3: else
4:     set k = b, l = a
5: end if
```

$$\lambda_p = \frac{\overline{R}_i^j - \overline{P}_k^x}{\underline{P}_l^x - \underline{R}_i^j} \quad [\text{when } \overline{R}_i^j > \overline{P}_k^x \text{ and } \underline{P}_l^x > \underline{R}_i^j] \tag{3.2}$$

$\underline{P}_i^x$ and $\overline{P}_i^x$ indicate the starting and the ending time of phase P_i^x. Here, x and i indicate the cycle number and the phase number respectively. Just as in *step 1*, λ_p

compares green extension $(\overline{R}_i^j - \overline{P}_k^x)$ and early green $(\underline{P}_l^x - \underline{R}_i^j)$. If $i = a$, then we need to rotate phases inside the ring to calculate early green (figure 3.15). On the other hand, if $i = b$, then we need to rotate the phases for green extension.

To decide which option to select, we can reuse the variable C from *step 1*. If $\lambda_p > C$, early green is selected, and if $\lambda_p <= C$, green extension is our selected strategy. Phase rotation should be applied accordingly, if it is needed based on the selection.

If the phase group is currently in service, then we can not use phase rotation as one of the phases is already executing. Instead of phase rotation, preemption can be useful along with green extension and early green strategies.

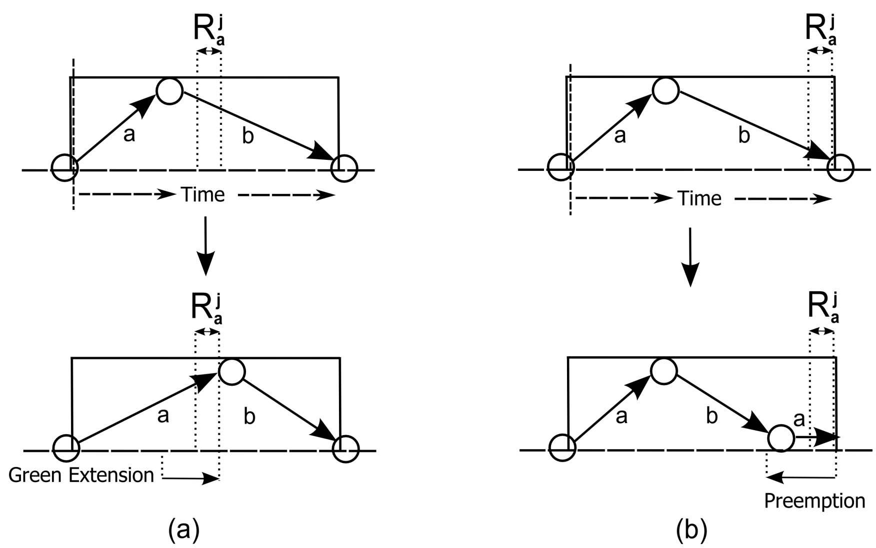


Figure 3.16: Green Extension (a) and Preemption (b) inside a ring

Figure 3.16 shows two different cases where an emergency vehicle requires a phase which is currently in service. Here in this example, the current schedule does not provide the required phase in both cases. To get the required phase, we need to use green extension [figure 3.16(a)] or preemption [figure 3.16(b)]. We need to select signal preemption if the emergency vehicle arrives, and the phase group is close to its end time.

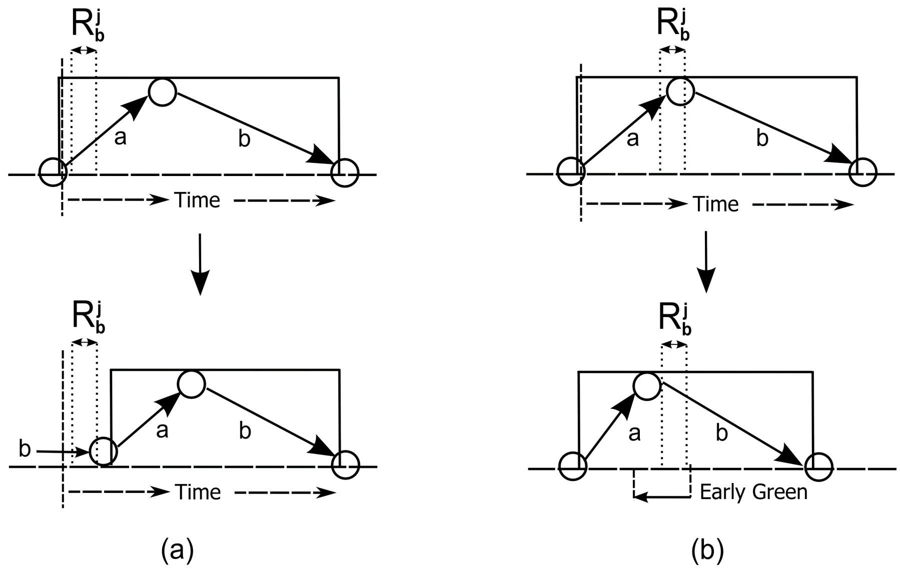


Figure 3.17: Preemption (a) and Early Green (b) inside a ring

Figure 3.17 shows the cases where an emergency vehicle requires the ending phase of the phase group while the starting phase is in service. We can provide early green here if it allows minimum green time to the current phase [figure 3.17(b)]. Otherwise, we need to suspend the current phase immediately to provide the required phase at its arrival time [figure 3.17(a)]. For any other case, we go for preemption if the emergency vehicle does not get the required phase at its arrival time after *step 2*.

The pseudocode in page 42 represents our single-request algorithm. At the beginning, the algorithm checks whether the request already has the required phase group or not. If it does, then it sends the request to CheckPhase function to get the required phase. If the request does not have the required phase group at its arrival, the request is sent to EarlyGreenForPhaseGroup or GreenExtensionForPhaseGroup to get the required phase group. After getting the phase group, the algorithm reaches to step 2. Here, CheckPhase function ensures the required phase for the request. At first, phase rotation is applied if it is needed to ensure the required phase. If phase

Algorithm 1 Single-Request Algorithm

```
procedure MAIN                                              ▷ //step 1
    if request R^j gets required phase group then
        CHECKPHASE()
    else
        if λ > C then
            EARLYGREENFORPHASEGROUP()
        else
            GREENEXTENSIONFORPHASEGROUP()
        end if
    end if
end procedure

procedure GREENEXTENSIONFORPHASEGROUP
    Apply GreenExtension on PhaseGroup
    on success CHECKPHASE()          ▷ //on failure preemption and continue
end procedure

procedure EARLYGREENFORPHASEGROUP
    Apply EarlyGreen on PhaseGroup
    on success CHECKPHASE()          ▷ //on failure preemption and continue
end procedure

procedure CHECKPHASE                                        ▷ //step 2
    Apply PhaseRotation (if needed)
    if request R^j gets required phase then
        Return
    else
        if λ_p > C then
            EARLYGREENFORPHASE()
        else
            GREENEXTENSIONFORPHASE()
        end if
    end if
end procedure

procedure GREENEXTENSIONFORPHASE
    Apply GreenExtension on Phase               ▷ //or preemption if failed
end procedure

procedure EARLYGREENFORPHASE
    Apply EarlyGreen on Phase                   ▷ //or preemption if failed
end procedure
```

rotation can not help here to get the required phase, EarlyGreenForPhase or GreenExtensionForPhase is selected based on the λ_p value. At this point, the emergency vehicle gets the required phase from one of these two functions.

After ensuring the required phase, if it is needed, we need to adjust the phase from the other ring. Same road phases (left turn only phase and through phase) need to be served together. (1,6), (2,5), (3,8) and (4,7) are the pair of same road phases where one phase is from ring 1 and the other one is from ring 2 in each pair. For instance, if the emergency vehicle seeks for phase 3 which is from ring 1, we need to serve phase 8 from ring 2 to clear the traffic (figure 3.7). For all cases, we need to maintain minimum green time and maximum green time for emergency vehicle.

3.2.1.1 Dealing With Uncertainty

Uncertainty associated with arrival time estimation plays a big role in this research. Using span of time instead of a point in time for arrival time is an effective way which we used here to ensure robust optimization. As the vehicle approaches the intersection, the level of uncertainty to reach the intersection decreases with time. If we consider the emergency vehicles which are arriving within next two cycles, then we can avoid the high level of uncertainty related to arrival time estimation. These strategies can help us to reduce the number of schedule change attempts in traffic schedules related to an error in arrival time estimation.

Although these strategies can help us to get a reliable arrival time for an emergency vehicle, there is always a scope for error in arrival time estimation. Any incident can happen while the emergency vehicle is traveling which may cause the vehicle arrive faster or later than the estimated arrival time. To ensure that the emergency vehicle gets the required phase at its arrival time, we need to track the emergency vehicle and estimate its arrival time repeatedly after a short interval.

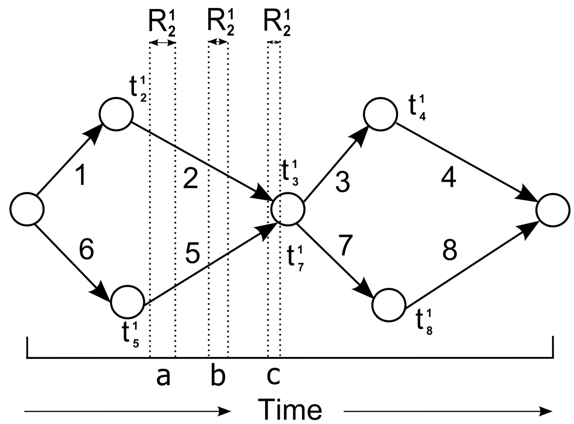


Figure 3.18: Error in arrival time estimation

Figure 3.18 shows an example where an emergency vehicle is seeking for phase 2 at its arrival. The variables a, b and c are three different estimations of arrival time calculated at three different times. For instance, estimated arrival time a is calculated here first, then it is changed to b and c respectively. Although estimation a and b are different from each other, we don't need to change our traffic signal schedule as the emergency vehicle gets the required phase in both cases. But for estimation c, the emergency vehicle does not get the required phase at its arrival, and we need to reschedule the traffic signal. If the update arrives while our algorithm is executing, the algorithm must return and we need to revert all the changes.

To reschedule the traffic signal, we need to undo any changes made associated with the erroneous arrival time estimation. We might not recover fully if any of the changes we made earlier is already executed or executing right now. For those cases, we need to recover the regular timing for upcoming phases. After undoing the changes which are made for the previous estimation, we can reschedule from scratch for the new estimation of arrival time.

3.2.2 Algorithm For Multiple Priority Requests

We have designed an algorithm in the previous section where the traffic signal control system can provide the required phase to a single emergency vehicle. In this section, this algorithm will be extended to fit multiple emergency requests. An example is shown here in figure 3.19 where two emergency vehicles are arriving at the intersection and seeking for different phases. They arrive within a very short time interval, and none of them get required phases with the current schedule.

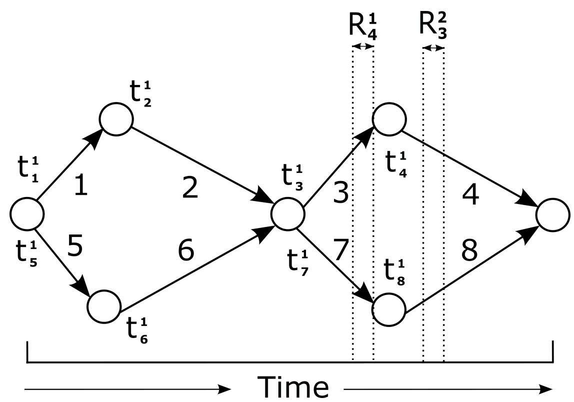


Figure 3.19: Multiple emergency requests

Although these two emergency vehicles are looking for conflicting phases, it is possible to design a traffic schedule which can provide required phases for both of them. But in some cases, especially when multiple emergency vehicles arrive almost at the same time and seek for conflicting phases, it is not possible to provide required phases for both of them. In that case, one must wait until the other one has passed through the intersection. To decide and prioritize among emergency vehicles, we have used priority weight (W^j) associated with each emergency request (R_i^j). The value of priority weight indicates the severity of the emergency needs based on the situation. Emergency Vehicle Dispatcher can determine relative priorities of situations and set the value of priority weights accordingly. A guideline should be available from traffic control center which can help a dispatcher to calculate the priority weight for an emergency vehicle.

3.2.2.1 Locking The Phase For An Emergency Request

For multiple emergency vehicles with conflicting phase requests, we need a locking technique where we can lock the required phase for a request so that other requests can not change the phase.

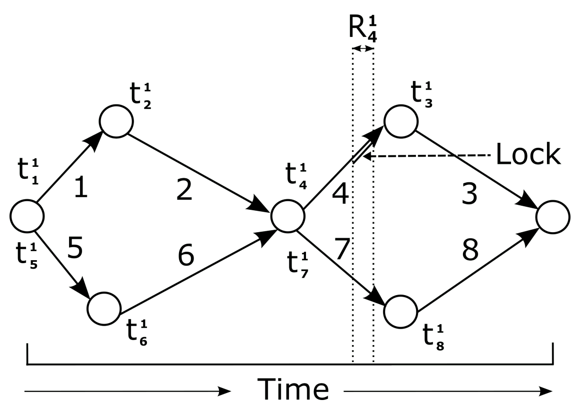


Figure 3.20: Phase locking for an emergency request

An example is shown in figure 3.20 where phase 4 is locked to serve an emergency request at its arrival time. After assigning the required phase for the first emergency vehicle, this phase needs a lock. With this lock, no other request can change this phase later. This way we can prevent disruption from future requests.

In this example here, the phase is locked during the estimated arrival time only. The emergency vehicle can also lock some extra time before its arrival for queue discharge at the intersection. We do not need to lock the entire phase. The phase which is locked by an emergency request can be truncated or extended on the unlocked portion for future requests.

Figure 3.21 shows another example with two emergency requests where phase a is locked for a request R_a^j [figure 3.21(a)], and this phase is truncated to make room for another request R_b^{j+1} [figure 3.21(b)]. Truncation of phase a does not have any

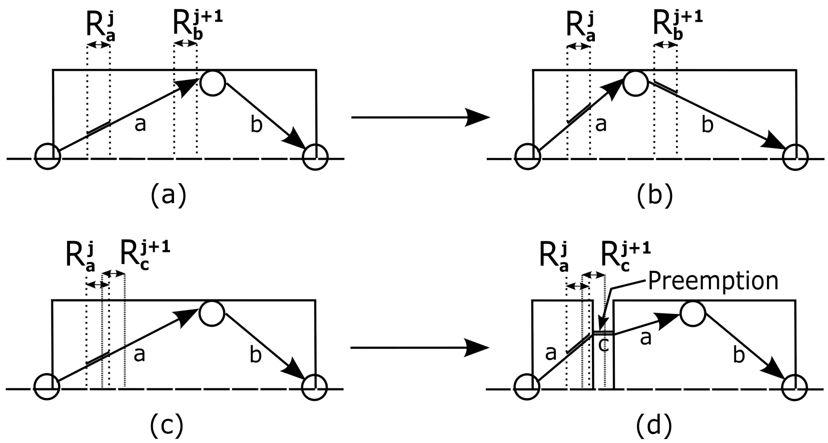


Figure 3.21: Phase locking for multiple emergency requests

impact on the locked portion. After locking, all future requests to make any change on the locked portion will be denied. In figure 3.21 (c), the second request seeks for a phase which is from different phase group and its estimated arrival time overlaps with request R_a^j. As phase a is locked, the second request can not change the locked portion. The request waits until the lock is open, and goes for preemption afterward as phase c can not be served using any other strategies [figure 3.21(d)].

3.2.2.2 A Greedy Approach For Multiple Emergency Requests

Now we have a traffic control algorithm for single request and a locking technique for multiple requests. If we sort all the active requests and assign schedule for all of them, we get an algorithm for multiple requests.

Algorithm 2 A Greedy Approach For Multiple Emergency Requests

```
Sort Active Request Queue based on priority weight
for each request in Active Request Queue do
    Assign schedule for the request using Single-Request Algorithm
    Lock the phase at arrival time
end for
```

Active request queue can be used as a data structure responsible for maintaining all the emergency requests which are not yet served. To avoid high uncertainty with arrival time estimation, we consider the active requests which are reaching the intersection within two cycles. Algorithm 2 sorts all the requests in the active request queue based on their priority weight. This means, higher priority requests get the schedule first. In the case of arrival time overlaps with conflicting phases, higher priority requests do not need to wait for completion of lower priority requests.

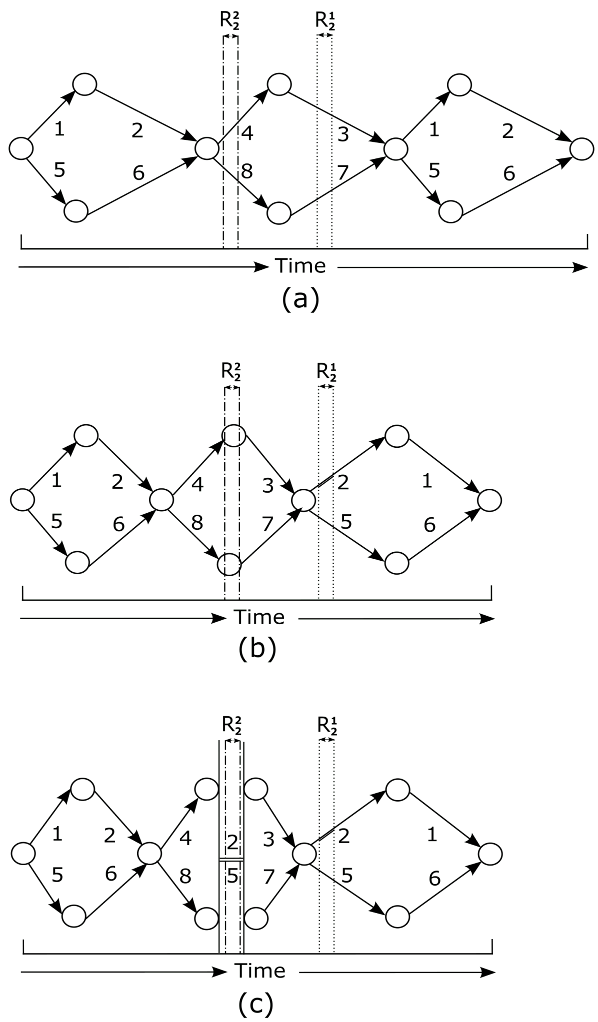


Figure 3.22: Assigning schedule using Algorithm 2

Figure 3.22 shows an example where two requests are assigned green time using our algorithm for multiple requests. These two requests are seeking for phase 2 where request R_2^2 arrives first, but has lower priority than request R_2^1. Our algorithm sorts the active request queue and request R_2^1 gets the first opportunity to get the green time at its estimated arrival time. In step 1, as request R_2^1 arrives on phase group 2 and seeking for phase group 1, our single request preemption algorithm needs to decide between green extension and early green strategy to apply on phase group. For instance, early green is to be selected here as early green requires less change on schedule. Applying phase rotation on the phase group ensures green time for request R_2^1, and our algorithm locks the phase at its estimated arrival time. Request R_2^2 does not get the required phase group, and it can not use green extension and early green. Green extension for the first phase group can not ensure minimum green time for phases from the second phase group as R_2^1 locked phase 2 at its arrival time. Early green is also denied as it prevents the phase groups to maintain minimum green time. Request R_2^2 gets desired green time by using preemption.

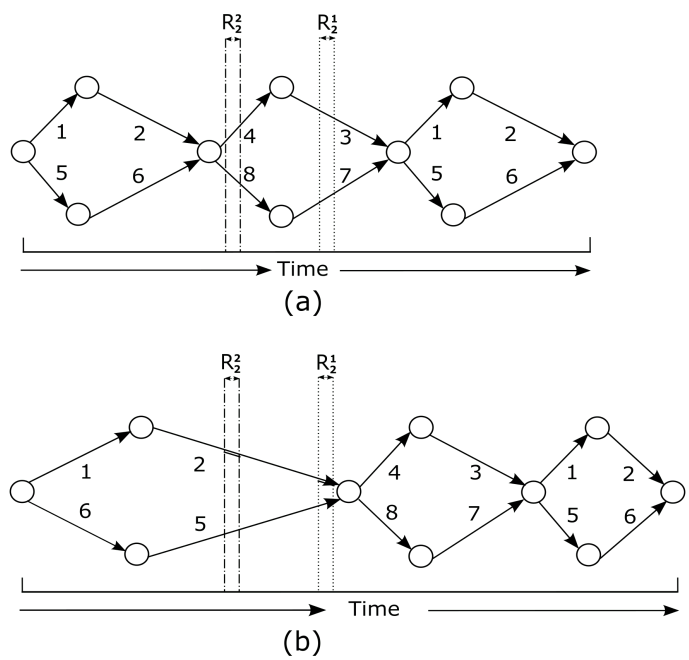


Figure 3.23: A better way to assign schedule for multiple emergency requests

The same requests can be served in a different way (figure 3.23). Instead of early green for phase group, if we select green extension for request R_2^1, resulting schedule comes out a lot different than what we have in figure 3.22. Request R_2^2 does not need to use preemption here, and disruption is less than the earlier example. This second schedule works better particularly in peak hours when no green time is wasted even for an extension of regular green time. In peak hours, all roads are packed with vehicles and it is not desirable to use phases with low time-spans. Because it needs some time to get natural traffic flow after a phase starts. Also, each phase change requires yellow and all red signal to serve, and it slows down traffic flow. Sequential low time-span phases can create some confusions among motorists. Moreover, preemption is not a desired strategy if there is another option to apply. For all these reasons, the schedule we have in figure 3.23 is a better choice than the earlier one. For a single request, our single request priority algorithm works fine. But for multiple priority requests, using a greedy approach to select the best option for an individual request may not be a good choice. If we can consider other choices, then the resulting schedule may have lower schedule disruption.

3.2.2.3 Schedule Disruption Value

Multiple emergency requests can be served in different ways, and our goal is to find a way where priority is maintained while keeping disruption to general traffic at a minimum level. We maintain priority by sorting the active request queue based on priority weight (W^j) and assigning green time to requests accordingly. Now, we need to find a way where we can identify a schedule which has the lowest disruption to the traffic.

Figure 3.24 shows an example of a regular traffic signal schedule at a four-way intersection. Let us consider a case where we have some emergency requests to serve during the operation of this schedule. If this signal schedule can not serve all of the

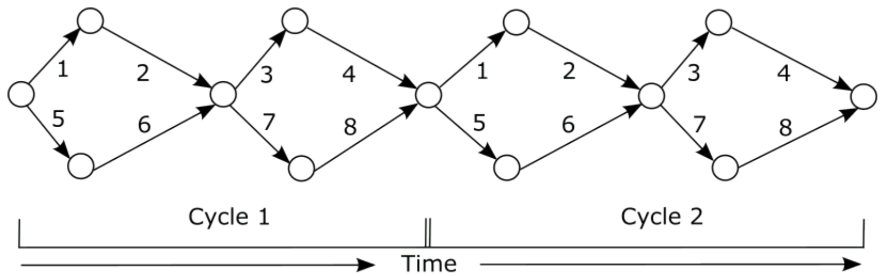


Figure 3.24: An example of a traffic signal schedule at an intersection

requests, we need to adjust phase timings. Figure 3.25 and figure 3.26 show two different candidate solutions for these emergency requests which are derived from the regular schedule (figure 3.24). Let us assume, both of these signal schedule solutions serve all of these requests. Now we need to choose one of these solutions which has the least impact on general traffic. Schedule disruption value can help us here to identify the better solution. Between these two candidate solutions, our selected schedule would be the one which has the lowest schedule disruption value.

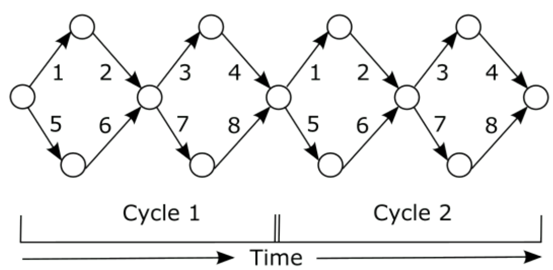


Figure 3.25: A traffic signal schedule solution (phases compressed)

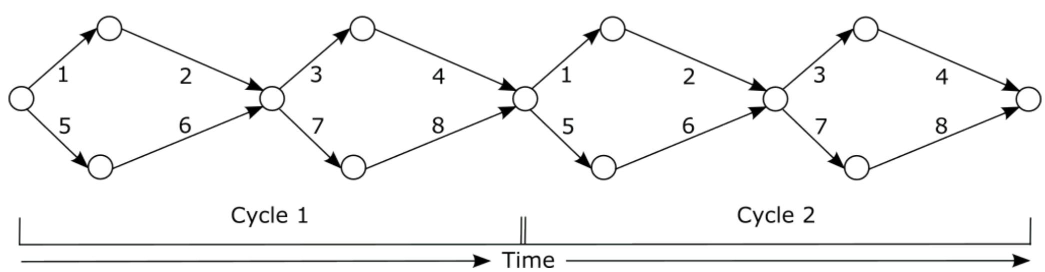


Figure 3.26: Another traffic signal schedule solution (phases stretched)

Schedule Disruption Value (sdv) calculation: Every phase in an intersection has a time-span which can be a preset value, or it can be determined based on current traffic demand. In adaptive signal control, the phase duration increases if the flow of traffic is high. There are some other aspects which are considered while calculating the phase duration. The average vehicle delay needs to be minimized, and offset value (figure 2.1) requires to be maintained for a green wave.

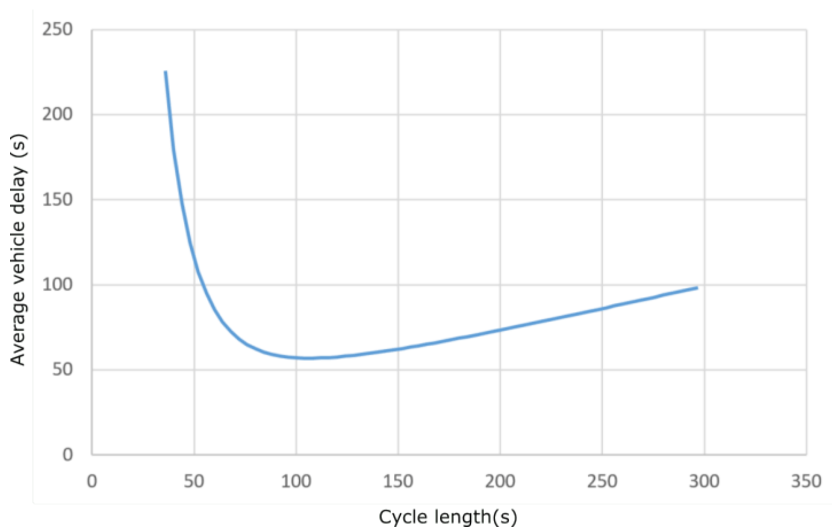


Figure 3.27: Relationship between average vehicle delay and cycle length [3]

Figure 3.27 shows the relationship between average vehicle delay and cycle length [3]. As traffic signal cycles consist of phases, this figure can be interpreted as the relationship between phase duration and average delay. If the phase duration is too short, too much time will be lost for yellow time, clearance interval, and vehicle starting delay. This reduces intersection capacity and thus the average delay increases. On the other hand, if the phase duration is extended for too long, vehicles from other approaches wait for too long which results in an increase of average delay. So we need to find an optimal time duration for each phase where the average vehicle delay is at a minimum level. And this optimal duration is set as the phase duration at an intersection. The phase duration may change slightly when offset value is added to cycle length to maintain signal coordination. After considering these aspects, traffic signal controller (or traffic engineer) sets the duration for each of the phases. Regular signal cycles use these phases to serve general traffic.

Now if the phase duration is changed from this standard value, it affects average vehicle delay and signal coordination. Schedule disruption value represents the overall effect for the deviation from the regular phase duration. To calculate this value for a typical four-way intersection, we propose the following equation.

$$sdv = \sum_{x=0}^{1} \sum_{i=1}^{8} max(1, \Delta d_i^x).\Delta v_i^x \tag{3.3}$$

Here,

$$schedule\ disruption\ value,\ sdv$$

$$average\ vehicle\ delay,\ d$$

$$phase\ duration,\ v$$

$$phase\ number,\ i$$

$$cycle\ number,\ x$$

$$\Delta d = d_{candidate} - d_{regular}$$

$$\Delta v = |v_{candidate} - v_{regular}|$$

$d_{candidate}$ and $d_{regular}$ represent average vehicle delay for a phase of the candidate solution and for the corresponding regular phase respectively. Similarly, $v_{candidate}$ and $v_{regular}$ represent phase duration for a phase of the candidate solution and for the corresponding regular phase respectively. Δd is the increased average vehicle delay from the regular phase for a candidate solution phase. Likewise, Δv is the change in phase duration from the regular phase for the candidate solution phase. Δv is always a positive value as it represents the duration which is added or cut from the regular phase. If Δd is too small (i.e. less than 1s) for a phase of a candidate solution, we just take Δv to calculate *sdv* for this phase. As we are considering next two cycles, *sdv* represents the combined effect of all the upcoming phases during this period. Equation 3.3 starts $\Delta d.\Delta v$ calculation from the first phase (*phase* 1) of the first cycle (*cycle* 0) of the candidate solution and adds up values for each of the phases up to the last phase (*phase* 8) of the second cycle. If our calculation starts from the middle of a cycle, ending point should be at the same position after two cycles (i.e. sixteen upcoming phases for a typical four-way intersection). The calculation of average vehicle delay (d) is described at the end of this thesis (Appendix A.1).

3.2.2.4 Traffic Control Algorithm For Multiple Emergency Requests

Using a schedule disruption value, we can develop an algorithm which selects the best schedule for multiple emergency requests which introduces the least amount of delay. In the multiple-request algorithm on page 55, we consider all priority strategies for each emergency vehicle in the active request queue and selects the schedule with the least amount of schedule disruption. For each request, the algorithm does not choose an option between green extension and early green immediately. Instead, it advances to next steps and calculates schedule disruption value at the end for each of these choices. If it fails to apply a priority strategy (i.e., green extension or early green) for any request, it applies preemption. Green extension and early green can fail to apply for a request for different reasons. If the desired time-span is already locked for higher priority emergency vehicle, the lower priority vehicle needs to wait until the high priority vehicle clears the intersection. Then the vehicle will get preemption if it does not get the desired phase normally. Besides, we need to

Algorithm 3 Multiple-Request Algorithm

```
procedure Main
    Set Lowest-schedule-disruption-value = a high value
    Sort Active Request Queue based on priority weight
    ProcessMultipleRequests(1)
    Execute Saved Schedule
end procedure

procedure ProcessMultipleRequests(index j)
    if j > request.Length then                ▷ //when all requests are scheduled
        Schedule-disruption-value = CalcScheduleDisruptionValue(schedule)
        if Schedule-disruption-value < Lowest-schedule-disruption-value then
            Save Schedule
            Lowest-schedule-disruption-value = Schedule-disruption-value
        end if
        Return
    end if
    CheckPhaseGroup(j)
end procedure

procedure CheckPhaseGroup(index j)
    if request R^j gets required phase group then
        CheckPhase(j)
    else
        GreenExtensionForPhaseGroup(j)
        EarlyGreenForPhaseGroup(j)
    end if
end procedure

procedure GreenExtensionForPhaseGroup(index j)
    Apply GreenExtension on PhaseGroup
    on success CheckPhase(j)              ▷ //on failure preemption and continue
end procedure

procedure EarlyGreenForPhaseGroup(index j)
    Apply EarlyGreen on PhaseGroup
    on success CheckPhase(j)              ▷ //on failure preemption and continue
end procedure
```

```
procedure CheckPhase(index j)
    Apply PhaseRotation (if needed)
    if request R^j gets required phase then
        Lock the phase at arrival time
        ProcessMultipleRequests(j + 1)
    else
        GreenExtensionForPhase(j)
        EarlyGreenForPhase(j)
    end if
end procedure

procedure GreenExtensionForPhase(index j)
    Apply GreenExtension on Phase (if no conflict)        ▷ //or preemption if failed
    Lock the phase at arrival time
    ProcessMultipleRequests(j + 1)
end procedure

procedure EarlyGreenForPhase(index j)
    Apply EarlyGreen on Phase (if no conflict)            ▷ //or preemption if failed
    Lock the phase at arrival time
    ProcessMultipleRequests(j + 1)
end procedure

procedure CalcScheduleDisruptionValue(schedule)
    Schedule-disruption-value = 0
    for each phases which are changed do
        Schedule-disruption-value += deviation from regular phase
    end for
    Return Schedule-disruption-value
end procedure
```

maintain minimum green time and maximum green time for the emergency vehicle. If a priority strategy fails to maintain any of these two constraints, it needs to select preemption instead for this request. If preemption is selected at phase group (e.g., GreenExtensionForPhaseGroup), the algorithm does not need to proceed to phase level (e.g., CheckPhase); since preemption assures the desired phase at desired time.

The algorithm works the following way. At the very beginning, the schedule sorts all requests based on their priority weights. The request with the highest priority weight is sent to the ProcessMultipleRequests function. This function is the entry point for each request and simply sends the request to CheckPhaseGroup. This function is the step 1 of the single-request algorithm and checks whether the request already has the required phase group or not. If it does, then it sends the request to CheckPhase function to get the required phase. If the request does not have the required phase group at its arrival, the request is sent to GreenExtensionForPhaseGroup and EarlyGreenForPhaseGroup functions to the get required phase group using green extension and early green. Unlike the single-request algorithm, we are considering both of these techniques. As a result, branching occurs, and we are proceeding with both of the branches. In the single-request algorithm, depending on the value of λ, it invokes only one of these two functions.

The next step is similar to the single-request algorithm. Each of these functions sends the request to CheckPhase after ensuring phase group for the request. The role for the CheckPhase function is to ensure the required phase for the request which resembles the second step of our single-request algorithm. If the request already gets the phase, we just need to ensure that the corresponding phase from phase pairs is served concurrently with the phase (figure 3.7). Phase rotation on the other ring can help us here. If the request does not get the phase, phase rotation is applied to ensure that the request gets the required phase (figure 3.14). If the request does not get the phase with phase rotation technique (figure 3.15), the request is sent to GreenExtensionForPhase and EarlyGreenForPhase to achieve the required phase (figure 3.16). At this stage, another branching may occur as we are considering early green and green extension both. In our single-request algorithm, λ_p selects only one of these functions. This is why we do not have branching in the single-request algorithm.

After getting the required phase, the request locks the schedule at its arrival time, and ProcessMultipleRequests is called again to process the next request which has the highest priority in the active request queue. Locking is another technique which is introduced in the multiple-request algorithm. We do not need any locking mechanism in the single-request algorithm as the algorithm deals with only one request.

When all the requests are served, this function calculates the schedule disruption value for the branch and saves the schedule if the disruption value is lower than previous branches. Otherwise, the algorithm simply proceeds to other unfinished branches without saving the schedule. When all branches are constructed, the saved schedule has the lowest schedule disruption value to execute.

Figure 3.28 shows a state space tree for an emergency request using our multiple-request algorithm. At the first step, the algorithm looks for the available priority choices for the phase group. If the regular schedule does not serve green time to the phase group, then we can apply green extension or early green to make sure that the phase group gets a green signal at its arrival time. The next step ensures the desired phase gets a green signal at the emergency vehicle's arrival time. If we have two choices for phase group and if we again get two choices in the next step for each of these two choices, we end up with four possible choices for a single request. Next

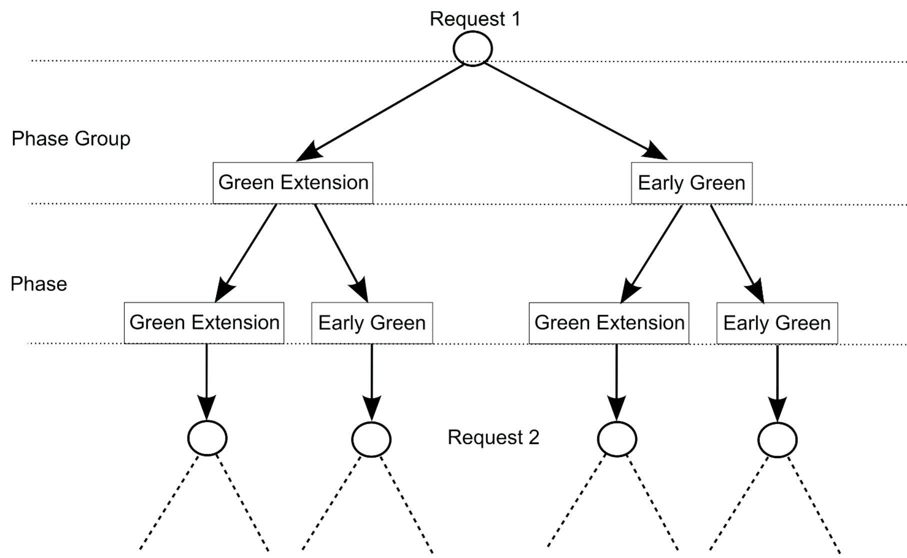


Figure 3.28: State space using our multiple-request algorithm

request is considered for all of these choices, and the decision tree continues until all of the requests are considered. Then the algorithm selects the combination of all the choices from the first request to the last having the lowest schedule disruption value.

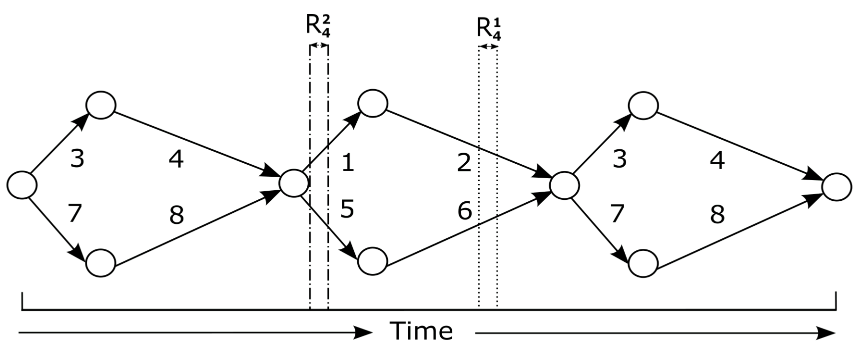


Figure 3.29: A schedule with multiple requests

An example schedule is shown in figure 3.29 where two requests are asking for the same phase, and none of them get the required phase. The operation process to assign a schedule using our algorithm is described in figure 3.30. The algorithm takes the first request, and two solution branches are created using green extension and early green. The solution branch from green extension gets the schedule for the

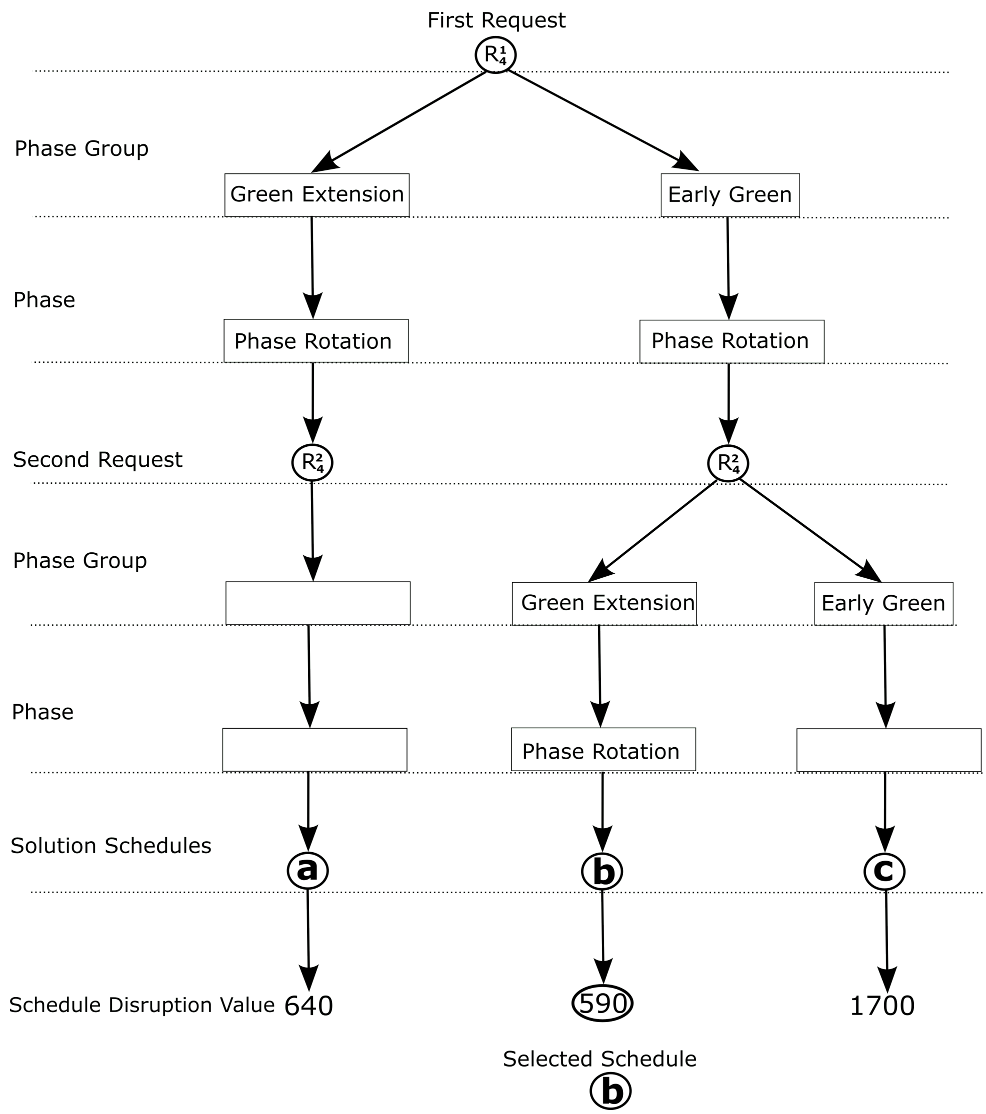


Figure 3.30: An example of multiple-request algorithm operation

first request, and covers the second request without any modification which creates a candidate solution [figure 3.31 (a)]. The other solution branch from early green creates another two branches for the second request and two candidate solutions [figure 3.31 (b) and figure 3.31 (c)] are produced. Among these three candidate solutions (a, b, and c), solution b [figure 3.31 (b)] has the lowest disruption value which means b is our selected solution to execute.

The time complexity of our multiple-request algorithm is $O(4^{n+1})$ which is not

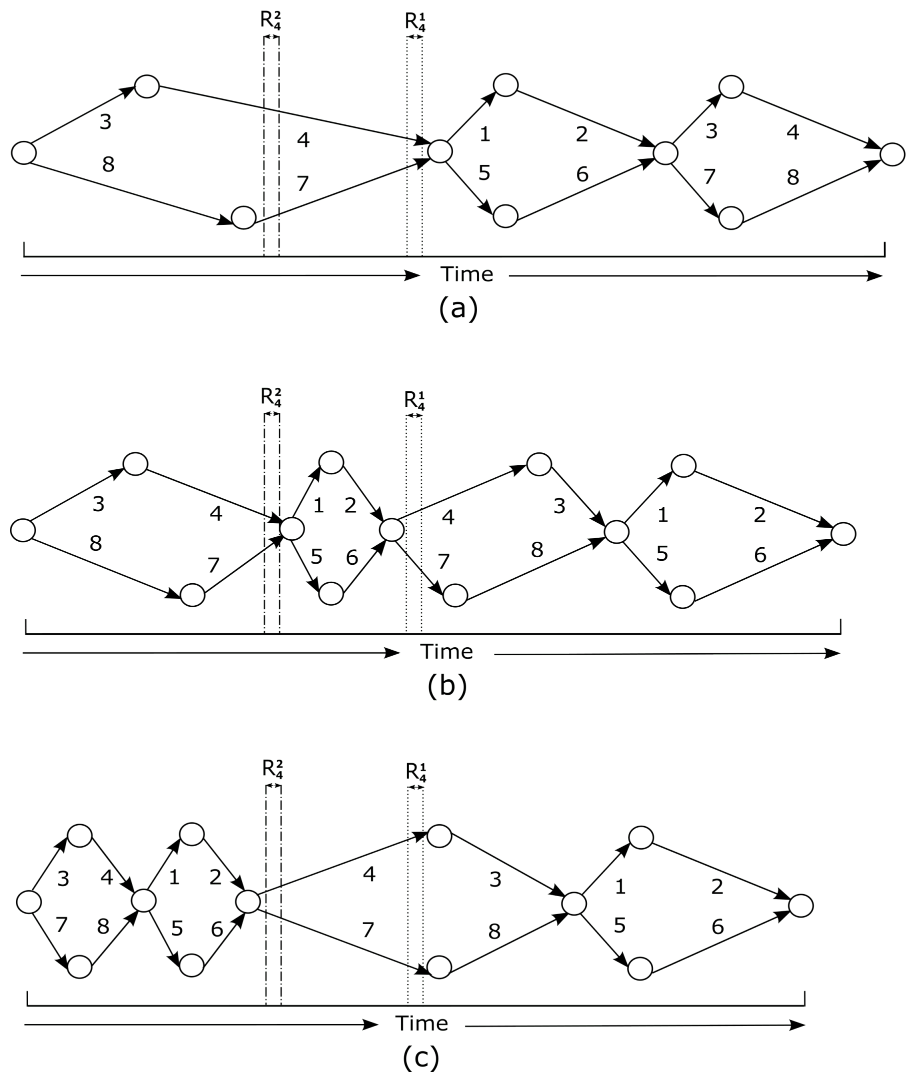


Figure 3.31: Solution schedules after the multiple-request algorithm operation

overwhelming where the number of emergency vehicle requests, n is relatively low (i.e. less than 5). For a scenario where a lot of emergency vehicles arrive at the same intersection at the same time, we need to optimize the algorithm. Grouping the requesting vehicles or using a different algorithmic approach can be two different ways to optimize the algorithm for this kind of situation.

3.2.2.5 Grouping Emergency Vehicles

In a four-way intersection, we can get emergency vehicles from these four road sections. We can group all the emergency vehicles which come from the same road section within a very short time interval and consider them as a single request. In this case, this request will get the priority value equal to the highest priority vehicle in the group. If there is more than one emergency vehicle waiting to get the green indication, it is not practical to consider only the highest priority vehicle. It is more time saving to clear all the emergency vehicles which arrive simultaneously from the same direction.

3.2.2.6 Algorithm Optimization: Branch And Bound Approach

Schedule disruption value can be interpreted as an extra waiting time for general traffic due to emergency vehicle movements. Vehicles are required to pull over and make a complete stop until the emergency vehicle passes. For multiple emergency requests, we need to make changes to the traffic signal schedule for each of the emergency vehicles. And for each of these changes, schedule disruption value increases. Using this information, we can lower the effective branching factor of the state space tree depicted in figure 3.28. Let us consider each of our choices as a node. If we branch through the nodes with minimum schedule disruption value, it is very likely to get the optimal solution much faster. The following algorithm describes a branch and bound approach where all the nodes with larger schedule disruption values are discarded when we get a candidate solution.

A node is created for each of the choices to serve a request, R^j. Each node represents the state of the schedule after serving a request along with the corresponding schedule disruption value. New nodes are put in a priority queue called Open List which sorts the nodes based on ascending order of the schedule disruption value. The lower the value of schedule disruption, the higher priority a node has in the queue.

Algorithm 4 A Branch And Bound Approach (Branch-And-Bound Algorithm)

```
procedure MAIN
    Sort Active Request Queue based on priority weight
    Set Lowest-schedule-disruption-value = a high value
    Set n = CREATENODE (top request of the Active Request Queue, 0, schedule)
    initialize the Open List                          ▷ //another priority queue
    put n on the Open List
    while the open list is not empty do
        find the node with the least dValue on the open list, call it "q"
        pop q off the open list
        PROCESSREQUEST(q)
    end while
    Execute Saved Schedule
end procedure

procedure CREATENODE(request R^j, dValue, schedule)
    Set the values              ▷ //dValue represents Schedule Disruption Value
    Return node
end procedure

procedure PROCESSREQUEST(node q)
    CHECKPHASEGROUP(q)
end procedure

procedure CHECKPHASEGROUP(node q)
    if request R^j(node q) gets required phase group then
        CHECKPHASE(q)
    else
        GREENEXTENSIONFORPHASEGROUP(q)
        EARLYGREENFORPHASEGROUP(q)
    end if
end procedure

procedure GREENEXTENSIONFORPHASEGROUP(node q)
    Apply GreenExtension on PhaseGroup
    on success CHECKPHASE(q)
end procedure

procedure EARLYGREENFORPHASEGROUP(node q)
    Apply EarlyGreen on PhaseGroup
    on success CHECKPHASE(q)
end procedure
```

```
procedure CheckPhase(node q)
    Apply PhaseRotation (if needed)
    if request R^j(node q) gets required phase then
        AssignGreenTime(q)
    else
        GreenExtensionForPhase(q)
        EarlyGreenForPhase(q)
    end if
end procedure

procedure GreenExtensionForPhase(node q)
    Apply GreenExtension on Phase (if no conflict)        ▷ //or preemption if failed
    AssignGreenTime(q)
end procedure

procedure EarlyGreenForPhase(node q)
    Apply EarlyGreen on Phase (if no conflict)            ▷ //or preemption if failed
    AssignGreenTime(q)
end procedure

procedure AssignGreenTime(node q)
    Lock the phase at arrival time of request R^j (node q)
    dVal = CalcScheduleDisruptionValue(schedule)
    if request R^j(node q) is the last request on the Active Request Queue then
        pop all the nodes off the Open List which have larger dValue than dVal
        if dVal < Lowest-schedule-disruption-value then
            Save Schedule
            Lowest-schedule-disruption-value = dVal
        end if
    else
        o = CreateNode(next request from Active Request Queue, dVal, schedule)
        put o on the Open List
    end if
end procedure

procedure CalcScheduleDisruptionValue(schedule)
    Schedule-disruption-value = 0
    for each phases which are changed do
        Schedule-disruption-value += deviation from normal phase
    end for
    Return Schedule-disruption-value
end procedure
```

The highest priority node will be popped off from the Open List and will get the chance to reschedule traffic signal to serve next request from the active request queue. Strategy selection process is similar to the multiple-request algorithm where we don't pick between candidate solutions immediately, rather move forward and after serving all requests, we select the solution with the least schedule disruption value.

In the worst case scenario, the branch-and-bound algorithm has the same time and space complexity as the multiple-request algorithm. The main difference between the multiple-request algorithm and the branch-and-bound algorithm is that the multiple-request algorithm makes a complete search to find the schedule. In our branch and bound algorithm, we calculate the schedule disruption value at every stage and discard those nodes which do not qualify to be a candidate solution. So for average cases, our branch-and-bounce algorithm is likely to get the solution much faster than the multiple-request algorithm.

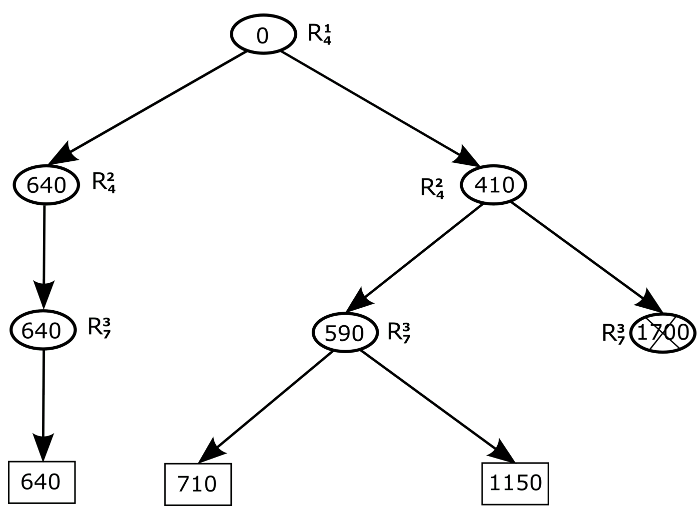


Figure 3.32: An example of the branch-and-bound algorithm operation

Figure 3.32 shows an example where three emergency vehicle requests are needed to be served using the branch-and-bound algorithm. Each of the oval shapes represents a node in this figure. The value inside the oval shape is the schedule disruption value

(sdv) of the node. First, the algorithm takes the highest priority request and makes a node with it. This node gets processed, and the required phase can be assigned by two ways which mean another two nodes for the second request. Unlike in the multiple-request algorithm, we calculate schedule disruption value here at this stage. These two nodes are then sent to the Open List. The top node ($sdv = 410$) pops off from the Open List to get processed, and it creates another two nodes for the third request which are put on the Open List. The top node from the Open List at this stage ($sdv = 590$) produces two candidate solutions. All nodes having larger sdv value than our candidate solutions are discarded from the Open List. The next node from the Open List gets processed, and the algorithm continues until the Open List is empty. Using this approach we can discard some branches (one node is discarded here with $sdv = 1700$) which have higher schedule disruption values and get the solution quicker.

The solution from the branch-and-bound algorithm is always optimal. Let us assume that a solution exists among one of these discarded nodes. This means, after serving all the requests, the solution has the least schedule disruption value. Changing the traffic signal schedule for any new emergency vehicle will never decrease the value of schedule disruption. So the final solution with this discarded node will have a higher schedule disruption value than our selected solution, which is a contradiction to our previous assumption. Discarded nodes always have a higher schedule disruption value than our candidate solution. So the discarded nodes can not have any optimal solution.

3.2.2.7 Recovery From Arrival Time Estimation Error

Keeping the schedule disruption value as low as possible has a substantial benefit in recovery from arrival time estimation error. Our algorithm selects the solution which needs the least amount of change in traffic signal schedule. If we need to revert our solution to a regular schedule, this needs the least amount of change as well. This means, reverting and rescheduling the traffic signal is easier with the least amount of schedule disruption.

Let us consider the previous example from figure 3.23. If the estimated arrival time of any of the emergency vehicle changes, we may need to reschedule the traffic signal to serve the emergency requests. Figure 3.33 describes such kind of scenario where the estimated arrival time changes while the resultant schedule is in service. In this example, the resultant solution is applied to schedule the traffic signal to serve green indications for request R_2^1 and R_2^2 [figure 3.33(a)]. Figure 3.33(b) shows the changed estimated arrival time of request R_2^1 while traffic signal served phase 1 and phase 6 already from the resultant solution. Request R_2^1 does not get the desired phase at its newly estimated arrival time in this example. To reschedule, we need to revert the schedule to its regular state [figure 3.33(c)]. Here, the signal schedule can not be reverted fully as phase 1 and phase 6 are served already. After this revert process, our algorithm reschedules the traffic signal to serve both of these requests [figure 3.33(d)].

The same rescheduling process is applicable if any new request comes within next two cycles while a solution is in progress. We need to put the request in the active request queue and return to regular traffic schedule as much as possible. We can generate a new solution using the latest active request queue, and traffic controller can implement the new solution.

3.2.3 Implementation Issues

Our traffic control algorithm may face some issues during implementation. Some of these issues are described below.

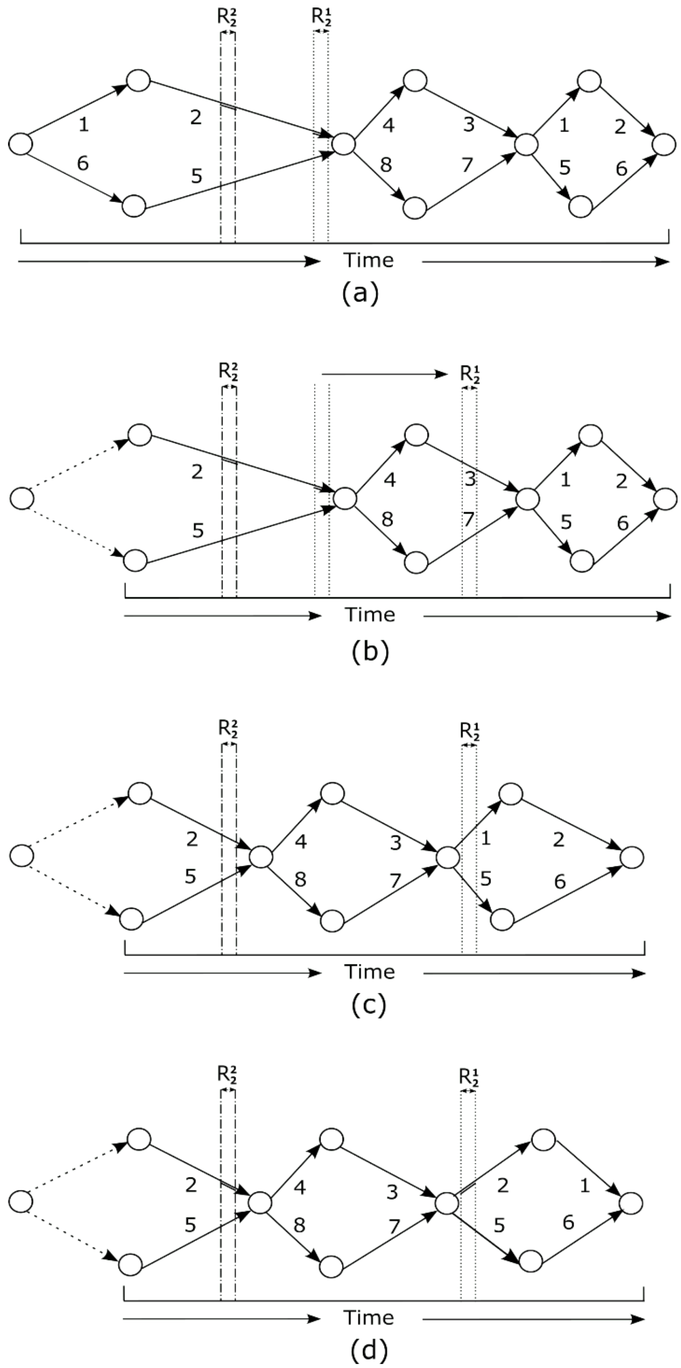


Figure 3.33: Recovery from arrival time estimation error

- For multiple emergency requests, our algorithm ensures that the vehicle with the highest priority gets the desired phase at its arrival time. Other emergency vehicles which have comparatively lower priority values and request for conflicting phases, need to wait until the emergency vehicle with the highest priority pass through the intersection safely. At this stage, it is unclear how a driver from an emergency vehicle will react when s/he requests for preemption, and it is not granted immediately. Drivers may assume that the system is not working, and may attempt to pass the intersection if they are not informed about the presence of another emergency vehicle. By communicating with them, connected vehicle technology can help us here to avoid this safety risk. Using dedicated short-range communications technology(DSRC), the traffic controller can communicate with these emergency vehicles and advise them to wait or maintain an optimum speed in case of conflicting emergency requests.

- To ensure a free flow of the emergency vehicle, we need to make sure that there is no traffic congestion on its way through the intersection. In the case of congestion, we need to serve green signal earlier than the emergency vehicle arrival time. For this in our calculation, we can include the *queue discharge time* with the emergency vehicle arrival time. Recall that the queue discharge time is the extra green time on the required phase which clears traffic congestion to ensure a free flow of the emergency vehicle. This means, if the emergency vehicle arrival time spans from $\underline{R}_i^j$ to $\overline{R}_i^j$ and the queue discharge time is q^j, we need to serve the green signal from $\underline{R}_i^j - q^j$ to $\overline{R}_i^j$ for the desired phase.

Most of the embedded devices do not have large processing capabilities. So it could be tough to accommodate parallel processing into the system. This is why these algorithms in this thesis were designed to run sequentially. The traffic controller itself is responsible to invoke the algorithm. In case of any parallel call, the algorithm can return and wait for the completion of the previous call.

3.3 Simulation Evaluation

To evaluate our proposed algorithm, a set of experiments was conducted using VISSIM microscopic traffic simulator. VISSIM was developed by Planung Transport Verkehr (PTV) in Karlsruhe, Germany and currently, is one of the market leaders in microscopic transportation simulation. It simulates every detail of individual entities in a traffic flow model and is being used for assessing and solving various transportation problems [67] [68].

For these experiments, a program was written in C# to reschedule the traffic signal. This program takes the current signal schedule and the emergency requests as the input and generates a new traffic schedule using our algorithm. The output schedule is then fed into the VISSIM simulator program to assess the effect on general traffic.

Minimum green time and maximum green time for emergency vehicles are set as 10s and 100s respectively. For each phase, 3s of yellow time and 1s of clearance interval are added after each green time. Measurement methods for the average vehicle delay (Appendix A.1) and C (Appendix A.2) are discussed at the end of this thesis. For early green implementation, we have truncated green time from all current and upcoming phases proportional to their phase duration. Similarly, we added an equally proportional amount of time duration to these phases for a green extension.

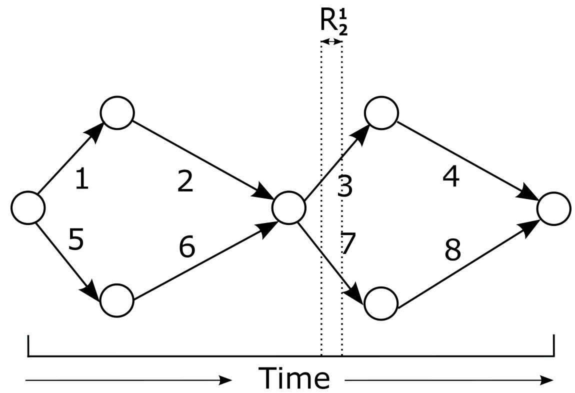
Figure 3.34: Intersection models on a major(left) and a minor (right) arterial road

Two four-way intersections were modeled in VISSIM to analyze average vehicle delays, average queue length, and intersection throughput in case of traffic signal preemption (fig. 3.34). Our first intersection was modeled on a major arterial road (Prince Philip Drive near Health Sciences Centre in St. John's, NL) and the second one was modeled on a minor arterial road (Elizabeth Avenue and Portugal Cove Road intersection in St. John's, NL). Intersections on arterial roads were chosen to examine vehicle delays as these are the major thoroughfare with signal lights in an urban area.

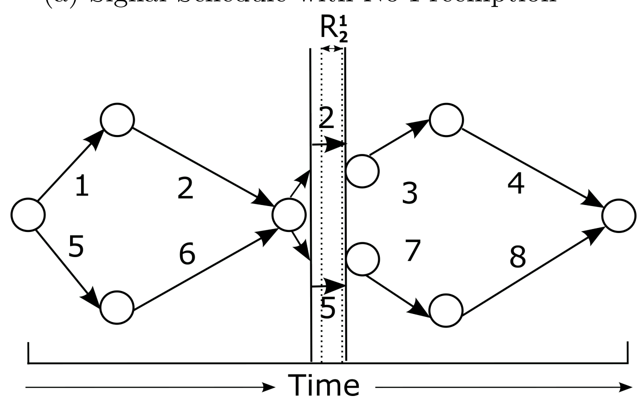
Each run of the experiment includes a 5 min of warm up time to allow traffic to stabilize before we start collecting the data. We considered low, medium, and high traffic volumes to measure the delay output at the intersection. Only those traffic signal cycles which have served at least one emergency vehicle are considered for delay measurements.

Figure 3.35 shows an example of our experiments to evaluate using VISSIM traffic simulator. We have an emergency request and a signal schedule here as an input to our program. Three types of signal schedules are considered for this experiment: no-preemption [figure 3.35(a)], preemption [figure 3.35(b)] and corresponding schedule from our traffic control algorithm [figure 3.35(c)].

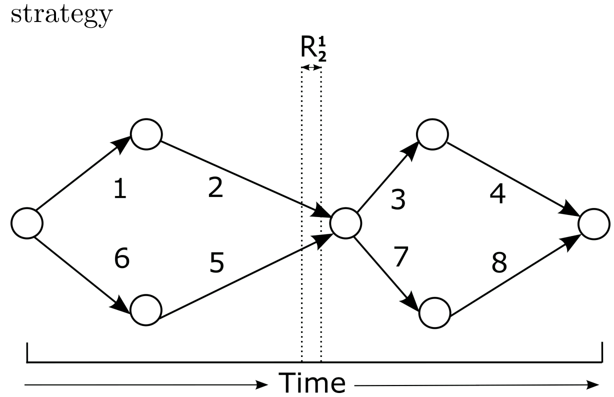
These signal schedules are implemented in VISSIM signal controller which is depicted in figure 3.36. For this example, we consider concurrent phases with same phase lengths. With no-preemption, our schedule from figure 3.35 can be implemented using four signal groups. Each signal group represents concurrent phase pairs in VISSIM software. Figure 3.36(a) represents the no-preemption strategy (e.g regular schedule). In figure 3.36 (b), signal group 3 and 6 represents the concurrent phase pair (3,7) which gets interrupted during preemption. The preempting phase pair (2,5) uses signal group 5 to serve a green signal to the EV. Figure 3.36 (c) represents the signal schedule from our traffic control algorithm. Here, concurrent phase pairs [(1,6) and



(a) Signal Schedule with No-Preemption



(b) Signal Schedule after applying Preemption strategy

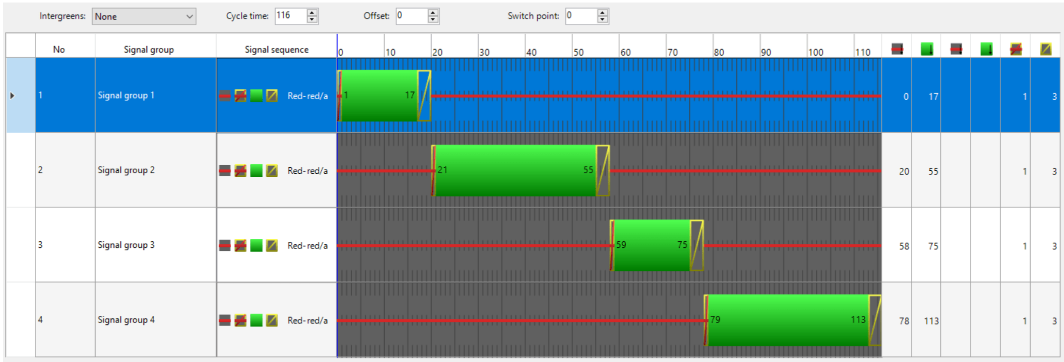


(c) Signal Schedule after using our algorithm

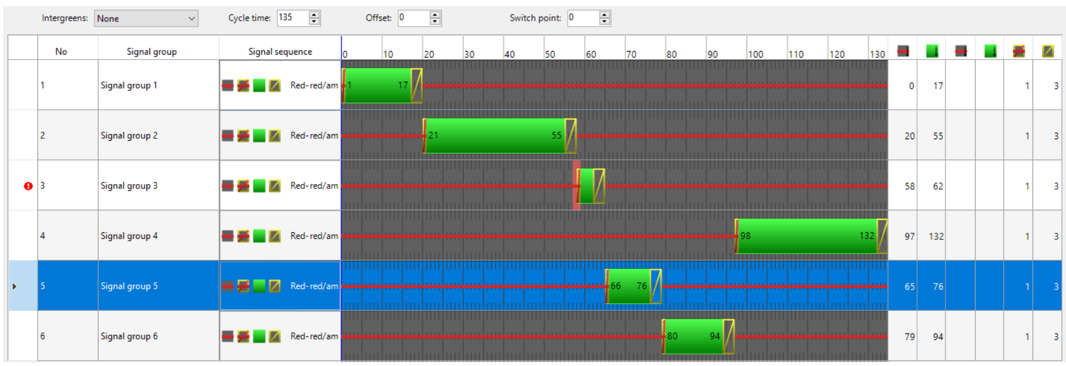
Figure 3.35: An experiment for simulation environment

(2,5)] from phase group 1 represents signal group 1 and signal group 2. Clearly, these two signal groups are stretched to render the green extension strategy.

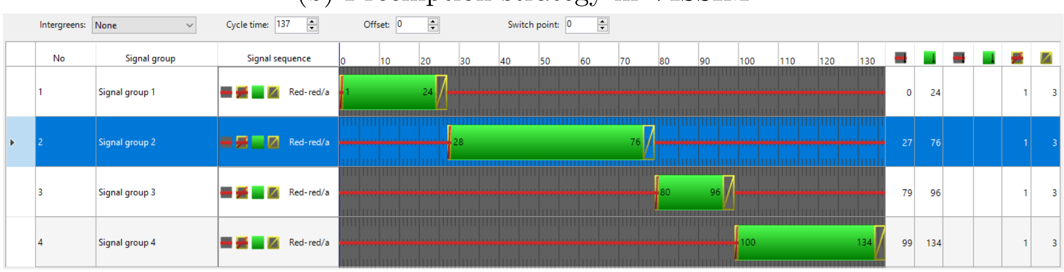
For each of the intersection we designed, we have implemented these traffic signal schedules. At this point, we provide traffic volume as input to the VISSIM simulator



(a) No-Preemption Schedule in VISSIM



(b) Preemption strategy in VISSIM



(c) Vissim Implementation of the schedule from our algorithm

Figure 3.36: VISSIM representation of signal schedules

to identify the impact of these schedules on general traffic. We have used different types of traffic volume at each intersection. We have categorized them as low, medium and high. Each intersection has a different capacity level. For the intersection with major arterial roads [figure 3.34 (a)], 400-900 vehicles per hour per approach to the intersection is categorized as low volume. Medium volume represents 900-1400 vehicles per hour per approach, and high volume represents 1400-2000 vehicles per hour per approach. More than 2000 vehicles per hour per approach exceeds the capacity of the intersection and creates an ever increasing traffic jam to the intersection. The other intersection has a slightly lower capacity than what is described here. The simulation is now ready to execute. We can get different types of output data from these experiments. Among these, we have collected average vehicle delay, average queue length, and average vehicle throughput to analyze the impact on general traffic.

Like the example described above, we have generated 30 different situations where emergency vehicle requests arrive at different phases, and different schedules are generated to collect the data. Each of the schedules is evaluated on each of the intersections using 13 different traffic volumes. For each schedule, corresponding schedules from the same situation (a schedule without preemption and another with preemption) are also evaluated to compare with the schedule from our algorithm. Schedules having single emergency request (20 situations) and multiple emergency requests (10 situations) are analyzed separately. All of our schedules consist of two cycles.

Single request: The schedules which serve a single emergency vehicle are analyzed here. The results of this category are important performance measures as this is the most frequent emergency request in effect.

Figure 3.37 shows the average vehicle delay for regular traffic at the intersection. Here, the y-axis represents the delay in seconds at different traffic volumes. Our algorithm is compared with emergency vehicle preemption and no-preemption scenarios. At low traffic volume, preemption does not have a significant impact on general traffic. Effect of preemption on general traffic is significant at medium and high traffic volumes where average vehicle delay is increased by around 30%.

Our algorithm works better than conventional preemption under medium and high traffic volume scenarios. Although we found 1-2% of extra vehicle delay at

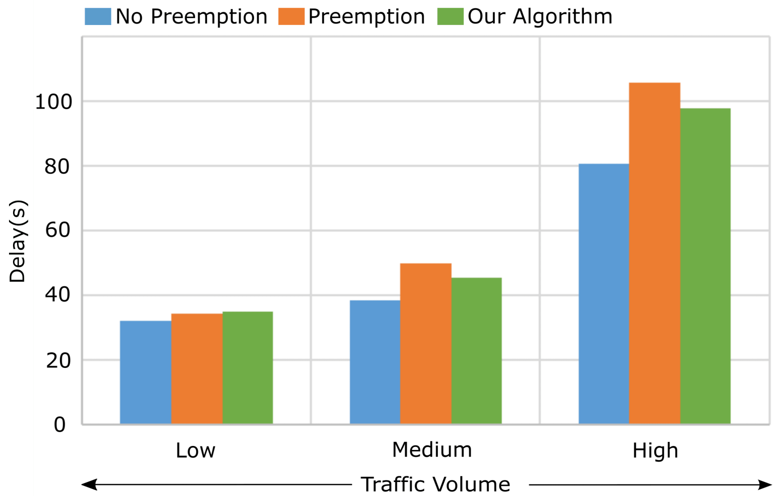


Figure 3.37: Vehicle delay at the intersection (single request)

lower traffic volumes compared to preemption, it overcomes the issue imposing less delay on medium and high traffic volumes. In both cases, our algorithm reduces the vehicle delay about 7-8% in comparison with conventional EVP (Emergency Vehicle Preemption).

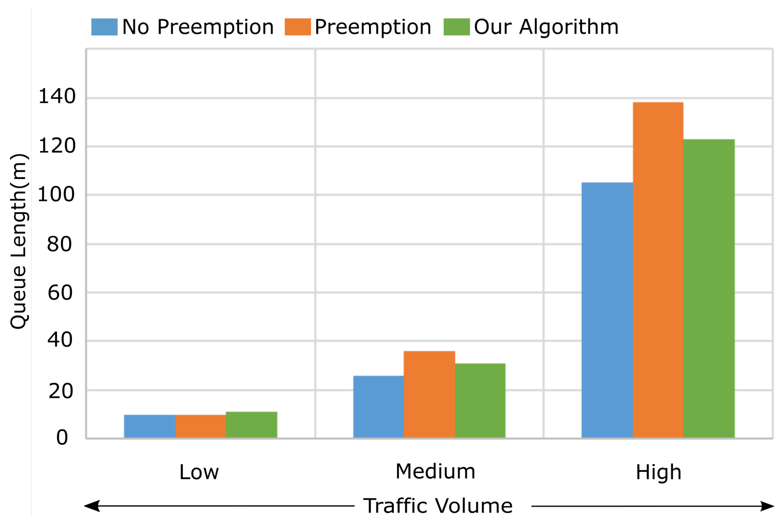


Figure 3.38: Queue length at the intersection (single request)

Average queue length at the intersection is another measure to understand the effect of preemption. Figure 3.38 depicts the queue length measurements for three different traffic volumes. Here, the y-axis represents arithmetic mean of queue lengths

[m] of all approaches at the intersection.

At lower traffic volume, our traffic control algorithm generates a slightly larger queue compared to preemption. On the other hand, it generates 11-14% smaller queues in comparison with preemption at medium and higher traffic conditions.

Intersection throughput is another measurement which measures traffic flow rate at the intersection. In these experiments, we counted vehicles passing through the intersection per minute to get the throughput value.

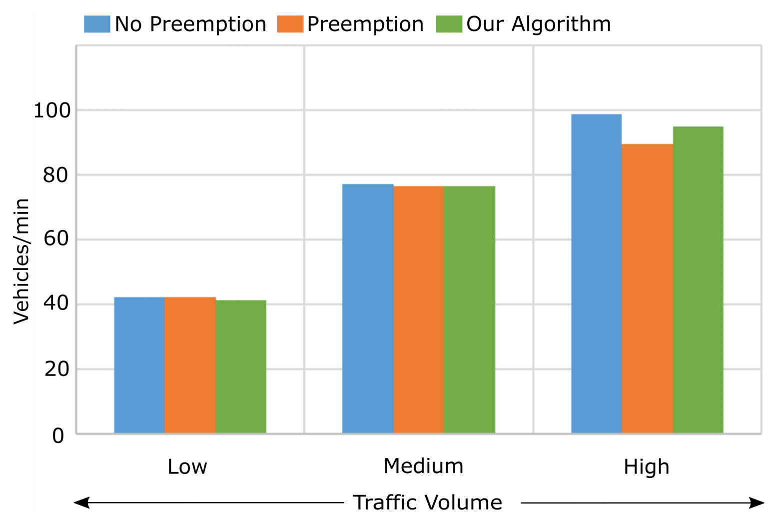


Figure 3.39: Throughput at the intersection (single request)

Figure 3.39 shows the intersection throughput values at different traffic volumes. Under low and medium traffic volume conditions, preemption and our traffic control algorithm does not have any significant impact on intersection throughput. Preemption reduces intersection throughput by 10% under high traffic volume conditions compared to no-preemption. Under the same traffic volume conditions, we found 6% higher traffic throughput compared to preemption. This means when the traffic volume reaches near the intersection capacity, preemption and our algorithm have some significant impacts on intersection throughput where our traffic control algorithm performs comparatively better.

Multiple requests: For this section of our simulation, we have considered schedules which serve two emergency requests within the two-cycle time-span. Conflicting and non-conflicting requests are analyzed separately for our algorithm as these two types of requests generate significantly different schedules. Non-conflicting requests can be served without any delay while one emergency vehicle must wait in case of conflicting requests. This means, our algorithm needs preemption strategy for the second request unless the requested phase comes naturally. On the other hand, in conventional preemption, the controller makes two preemption calls whether the requests ask for conflicting phases or not. For this reason, conflicting and non-conflicting requests are analyzed together for preemption and no-preemption cases.

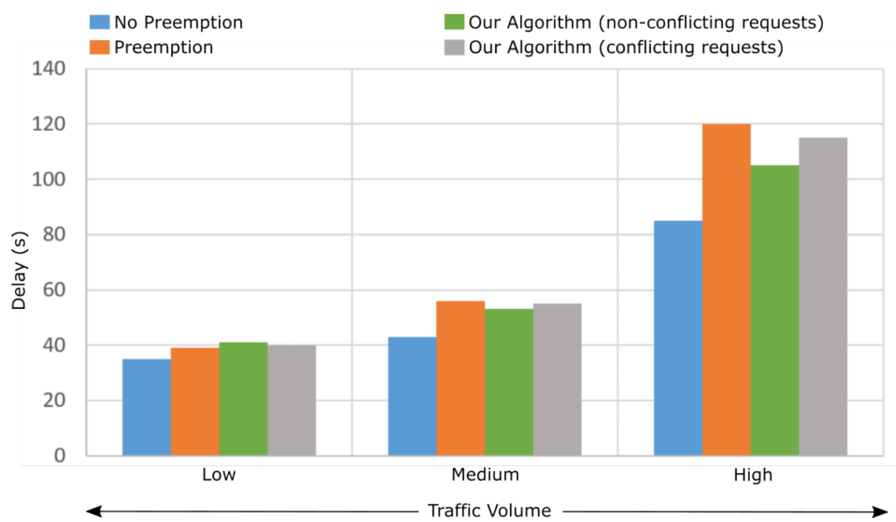


Figure 3.40: Vehicle delay at the intersection (multiple requests)

Figure 3.40 shows the average vehicle delay for regular traffic using schedules which serve two emergency requests. At low and medium traffic volumes, our algorithm does not make a significant improvement, but it reduces a worthwhile amount of delay in case of high traffic volume. Figure 3.41 depicts average queue length at the intersection for multiple request cases. For high traffic volumes, our algorithm shows a meaningful improvement particularly with non-conflicting requests. Figure

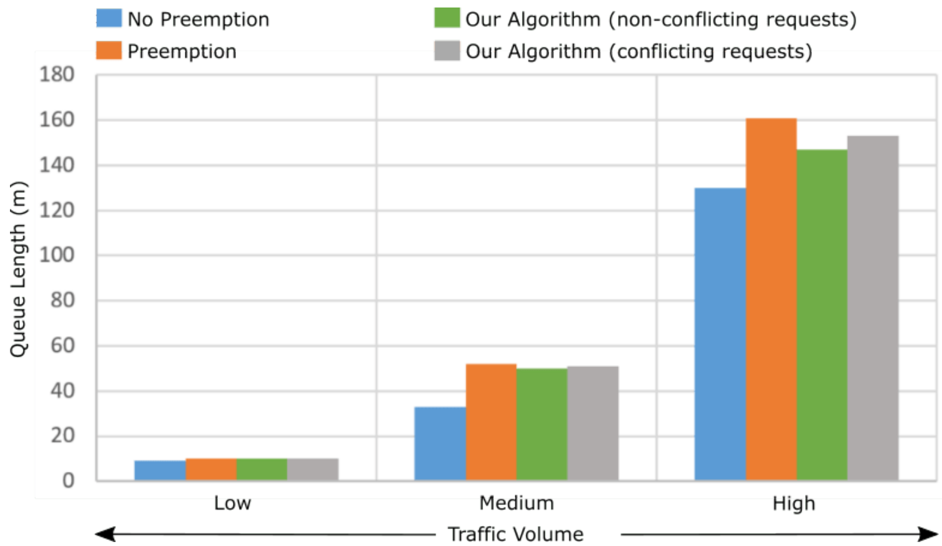


Figure 3.41: Queue length at the intersection (multiple requests)

3.42 portrays the throughput for multiple requests where a substantial amount of throughput is reduced for high traffic volume. Two emergency requests decrease the capacity of the intersection substantially than a single request. In case of throughput, our algorithm does not show extraordinary improvement with conflicting requests, but it shows better throughput than preemption, particularly with non-conflicting requests.

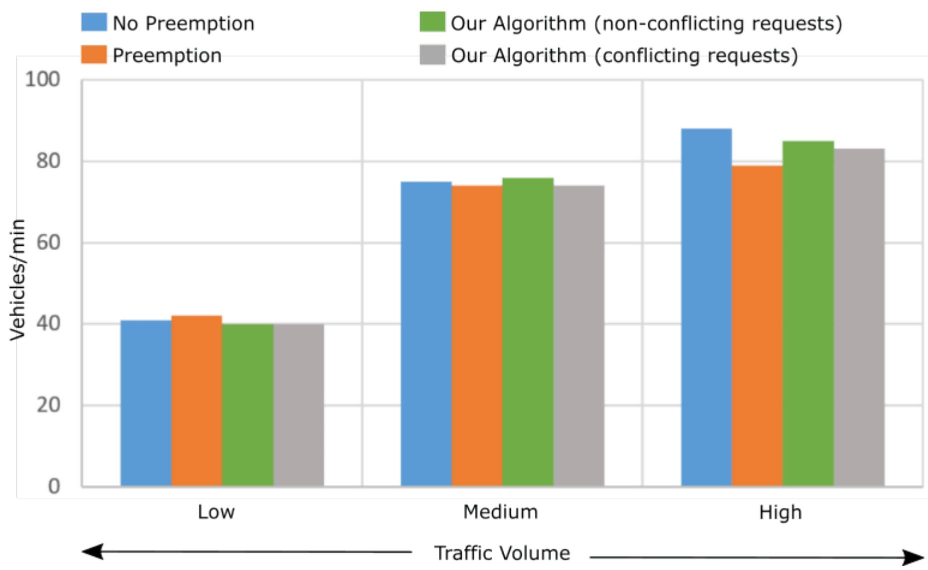


Figure 3.42: Throughput at the intersection (multiple requests)

Chapter 4

Conclusion

The advent of connected vehicle technology along with better arrival time prediction algorithms allow us to plan for a better emergency vehicle movement toward the intersection. Using TSP techniques at the right time, we can change the current schedule in such a way which will ensure the free movement of the emergency vehicle as well as keeping disruption to traffic flow at a minimum level. In the case of multiple simultaneous emergency requests, our algorithm provides a higher priority to the emergency vehicle which needs to respond faster and tries to keep the disruption level lower on traffic flow at the same time.

4.1 Summary

A lightweight traffic control algorithm is developed to implement priority control over emergency vehicles to deal with multiple emergency vehicle preemption scenarios. A priority queue is maintained for this purpose, and TSP techniques are used to alleviate the impact on traffic flow. Uncertainty regarding arrival time prediction is handled by using a span of time instead of a point in time. A branch and bound algorithmic approach is applied to identify an optimal solution which is easy to implement.

To analyze the performance of the algorithm, a set of experiments were conducted using the microscopic simulation software VISSIM. Compared with the conventional EVP (Emergency Vehicle Preemption), the proposed traffic control algorithm reduces average vehicle delay up to 8% under medium and heavy traffic conditions at the intersection.

4.2 Future Work

Our research work is focused on the improvement of emergency vehicle preemption control at isolated intersections. A future research direction includes the expansion of this work toward the operation of coordinated traffic signal control. Emergency vehicle preemption disrupts the coordination among the adjacent traffic signals. Different transition strategies can be implemented to recover the signal coordination. Obenberger and Collura [69] analyzed five most commonly available transition strategies to exit from the preemption control to the coordinated operation. Introducing exit phase is another way to recover from emergency vehicle preemption. An exit phase starts at the end of the emergency vehicle preemption and proceeds with a new phase which has the highest priority ignoring the normal phase. Yun et al. [70] evaluated fixed and dynamic exit phase controls during preemption control operations. These transition strategies and the exit phase control return the signal timing plan to a point where it would have been if no preemption was served. To identify the best recovery technique, we need to consider the conditions and constraints from each intersection and select the most suitable one which is in line with our traffic signal control algorithm.

Another future research direction is to improve the arrival time prediction techniques. Our research work relies heavily on the advancements of these techniques. With the recent progress in artificial intelligence and connected vehicle technologies, a new scope for improvement is available. Future research should investigate the contemporary ideas to develop an arrival time prediction technique particularly for the emergency vehicle with better accuracy.

Appendix A

A.1 Average Vehicle Delay (d)

Vehicle delay is related to driver's experience which expresses the excess time needed at an intersection. There are different kinds of delays related to an intersection [71]. Here in this thesis, we used approach delay for delay measurement. Approach delay includes three components - stopped delay, delay due to deceleration, and delay due to re-acceleration. It starts when a vehicle decelerates from the approach speed toward an intersection and ends when the vehicle gets back to the desired speed. Stopped delay is the time which is lost while a vehicle is stopped at an intersection waiting for the green signal indication. Average vehicle delay represents the average of all vehicles which arrive at the intersection during a time period.

The following equation represents the basic method to calculate the average vehicle delay for a phase [72].

$$d = d_1(PF) + d_2 \tag{A.1}$$

Here,

Average vehicle delay, d

Uniform delay, d_1

Progression adjustment factor, PF

Overflow delay, d_2

Our research was limited to isolated intersections and the progression adjustment factor for an isolated intersection is 1.0 ($PF = 1$) [72]. PF is different for coordinated traffic signal control.

A.1.1 Uniform Delay

Uniform delay represents the delay assuming uniform arrivals and stable flow of vehicles with no initial queue at the intersection. To estimate the uniform delay, we can use Webster's delay formulation which is widely accepted as a precise illustration of delay for uniform arrivals [3].

$$d_1 = \frac{0.5C_l(1 - \frac{g_i}{C_l})^2}{1 - [min(1, X)\frac{g_i}{C_l}]} \tag{A.2}$$

Here,

Uniform delay, d_1

Cycle length, C_l

Effective green time for phase i, g_i

Degree of saturation, X

A.1.1.1 Degree Of Saturation

Degree of saturation indicates the volume to capacity ratio.

$$X = \frac{v_i}{c_i} = \frac{v_i}{s_i(\frac{g_i}{C_l})} = \frac{v_i C_l}{s_i g_i} \tag{A.3}$$

Here,

Degree of saturation, X

Arrival flow (volume) for phase i, v_i (veh/h)

Capacity for phase i, c_i (veh/h)

Saturation flow rate for phase i, s_i (veh/h)

Saturation Flow Rate: Saturation flow rate for a phase indicates the number of vehicles which could pass through the intersection in an hour if the signal indication remained green for this phase all of the time. In this thesis, we have used 1800 veh/h as our saturation flow rate for each lane. A phase which operates on two lanes, has a saturation flow rate of 3600 veh/h.

Effective Green Time: Effective green time is the segment of actual green time during which vehicles can travel at saturation flow rate.

$$g_i = G_i + y_i \text{ - } \textit{lost time}$$

Here,

Effective green time, g_i

Actual green time, G_i

Yellow time, y_i

Lost time includes start-up lost time and ending lost time. Start-up lost time represents the time which is needed to discharge the stopped queue when the green indication is served. Ending lost time is the portion of yellow indication time which is not fully utilized to pass through the intersection. For simulation purpose, we have used 4s of total lost time for each phase.

A.1.2 Overflow Delay

Overflow delay represents the additional delay which occurs due to random overflow delay and continuous overflow delay. Random overflow delay indicates the delay which occurs due to the randomness of vehicle arrival. Some cycles may overflow randomly even if the traffic demand is lower than the capacity. Continuous overflow delay occurs when the capacity is lower than the actual traffic demand. The queue at the intersection grows continuously, and the total delay increases dramatically.

We can calculate the overflow delay using the following equation [73].

$$d_2 = 15t_e[(X-1) + \sqrt{(X-1)^2 + \frac{240X}{c_i t_e}}] \tag{A.4}$$

Here,

Overflow delay, d_2

Evaluation time, t_e *(minute)*

Degree of saturation, X

Capacity for phase i, $c_i = s_i(\frac{g_i}{C_l})$

For simulation, we have used $t_e = 10$ minutes.

A.2 Calculation Of *C*

$$C = \frac{max(1, \Delta d_{early})}{max(1, \Delta d_{extension})} \tag{A.5}$$

Here,

$$\Delta d_{early} = d_{early} - d_{regular}$$

$$\Delta d_{extension} = d_{extension} - d_{regular}$$

$$Average\ vehicle\ delay\ using\ early\ green\ ,\ d_{early}$$

$$Average\ vehicle\ delay\ using\ green\ extension,\ d_{extension}$$

$$Average\ vehicle\ delay\ using\ regular\ schedule,\ d_{regular}$$

In this study, we have considered the delay factor to calculate the value of C. A traffic engineer can also consider emergency vehicle travel history data and intersection characteristics while determining this C value.

Bibliography

[1] S. R. Sunkari, R. J. Engelbrecht, and K. N. Balke. Evaluation of advance coordination features in traffic signal controllers. Technical report, Texas Transportation Institute, 2004.

[2] P. Koonce, L. Rodegerdts, K. Lee, S. Quayle, S. Beaird, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik. Traffic signal timing manual. Technical report, Federal Highway Administration, U.S. DOT, 2008.

[3] F. Webster and B. Cobbe. Traffic signals. *Road Research Laboratory Technical Paper No. 56, London, U.K.*, 1966.

[4] M. F. Hazinski and J. M. Field. 2010 American Heart Association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care science. *Circulation*, 122(Suppl):S639–S946, 2010.

[5] R. Sánchez-Mangas, A. García-Ferrrer, A. De Juan, and A. M. Arroyo. The probability of death in road traffic accidents. How important is a quick medical response? *Accident Analysis & Prevention*, 42(4):1048–1056, 2010.

[6] E. Nelson and D. Bullock. Impact of emergency vehicle preemption on signalized corridor operation: An evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, 1727:1–11, 2000.

[7] Fatal accident involving two ambulances. www.mirror.co.uk/news/uk-news/ambulance-driver-dead-3-injured-7855187, 2016.

[8] Fire truck collides with ambulance. www.nydailynews.com/news/national/12-injured-miami-fire-engine-collides-ambulance-article-1.2322198, 2015.

[9] Police car and ambulance collision. http://metro.co.uk/2016/08/21/police-car-and-ambulance-on-separate-999-calls-crash-at-junction-6081891/, 2016.

[10] Y. Yin. Robust optimal traffic signal timing. *Transportation Research Part B: Methodological*, 42(10):911–924, 2008.

[11] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[12] P. Mirchandani and L. Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.

[13] A. Stevanovic. *Adaptive traffic control systems: domestic and foreign state of practice*. Project 20-5 (Topic 40-03). Transportation Research Board, 2010.

[14] P. Lowrie. The Sydney coordinated adaptive traffic system-principles, methodology, algorithms. In *International Conference on Road Traffic Signalling, 1982, London, United Kingdom*, 207, 1982.

[15] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4), 1982.

[16] L. A. Klein, M. K. Mills, and D. R. Gibson. Traffic Detector Handbook: -Volume II. Technical report, Federal Highway Administration, U.S. DOT, 2006.

[17] ITS Canada. `https://www.itscanada.ca/it/`.

[18] Americans were stuck in traffic for 8 billion hours in 2015. `http://money.cnn.com/2016/03/15/news/us-commutes-traffic-cars/`.

[19] G. Dimitrakopoulos and P. Demestichas. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84, 2010.

[20] K. N. Qureshi and A. H. Abdullah. A survey on intelligent transportation systems. *Middle-East Journal of Scientific Research*, 15(5):629–642, 2013.

[21] Road Traffic Injuries: The Facts. `www.who.int/violence_injury_prevention/road_safety_status/2015/magnitude_A4_web.pdf`.

[22] 2015 Motor Vehicle Crashes: Overview. `www.crashstats.nhtsa.dot.gov/Api/Public/Publication/812318`.

[23] E. Uhlemann. Introducing connected vehicles [connected vehicles]. *IEEE Vehicular Technology Magazine*, 10(1):23–31, 2015.

[24] D. Schrank, B. Eisele, T. Lomax, and J. Bak. 2015 urban mobility Scorecard. *Texas A&M Transportation Institute. The Texas A&M University System*, 2015.

[25] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark. Connected vehicles: Solutions and challenges. *IEEE internet of things journal*, 1(4):289–299, 2014.

[26] J. F. Paniati and M. Amoni. Traffic Signal Preemption for Emergency Vehicles–A Cross-Cutting Study. *US Federal Highway Administration*, 2006.

[27] J. Bruner. Greening the Traffic Lights. `http://www.forbes.com/forbes/2008/0505/064.html`, 2008.

[28] J. Wang, W. Ma, and X. Yang. Development of degree-of-priority based control strategy for emergency vehicle preemption operation. *Discrete dynamics in nature and society*, 2013, 2013.

[29] J. Collura, J. Chang, E. W. Willhaus, and J. Gifford. Traffic Signal Preemption and Priority: Technologies, Past Deployments, and System Requirements. In *ITS America 11th Annual Meeting and Exposition, ITS: Connecting the Americas*, 2001.

[30] H. R. Smith, B. Hemily, and M. Ivanovic. Transit signal priority (TSP): A planning and implementation handbook. *ITS America*, 2005.

[31] H. Liu, W.-H. Lin, and C.-w. Tan. Operational strategy for advanced vehicle location system–based transit signal priority. *Journal of Transportation Engineering*, 133(9):513–522, 2007.

[32] C.-W. Tan, S. Park, H. Liu, Q. Xu, and P. Lau. Prediction of transit vehicle arrival time for signal priority control: Algorithm and performance. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):688–696, 2008.

[33] H. X. Liu, W. Ma, X. Wu, and H. Hu. Real-time estimation of arterial travel time under congested conditions. *Transportmetrica*, 8(2):87–104, 2012.

[34] D. Fagan and R. Meier. Intelligent time of arrival estimation. In *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*, pages 60–66. IEEE, 2011.

[35] S. Budge, A. Ingolfsson, and D. Zerom. Empirical analysis of ambulance travel times: the case of Calgary emergency medical services. *Management Science*, 56(4):716–723, 2010.

[36] B. S. Westgate, D. B. Woodard, D. S. Matteson, S. G. Henderson, et al. Travel time estimation for ambulances using Bayesian data augmentation. *The Annals of Applied Statistics*, 7(2):1139–1161, 2013.

[37] Z. Zhang, Q. He, J. Gou, and X. Li. Performance measure for reliable travel time of emergency vehicles. *Transportation Research Part C: Emerging Technologies*, 65:97–110, 2016.

[38] J. Wang, M. Yun, W. Ma, and X. Yang. Travel time estimation model for emergency vehicles under preemption control. *Procedia-Social and Behavioral Sciences*, 96:2147–2158, 2013.

[39] H. X. Liu, X. Wu, W. Ma, and H. Hu. Real-time queue length estimation for congested signalized intersections. *Transportation research part C: emerging technologies*, 17(4):412–427, 2009.

[40] C. Zhan, A. Gan, and M. Hadi. Prediction of lane clearance time of freeway incidents using the M5P tree algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1549–1557, 2011.

[41] J.-T. Kim and K. Courage. Evaluation and design of maximum green time settings for traffic-actuated control. *Transportation Research Record: Journal of the Transportation Research Board*, 1852:246–255, 2003.

[42] G. Zhang and Y. Wang. Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):164–173, 2011.

[43] Y. Tanaka, H. Yamada, S. Tamasaku, and H. Inaba. The fast emergency vehicle pre-emption system improved the outcomes of out-of-hospital cardiac arrest. *The American journal of emergency medicine*, 31(10):1466–1471, 2013.

[44] R. K. Kamalanathsharma and K. L. Hancock. Congestion-Based Emergency Vehicle Preemption. Technical report, Mid-Atlantic Universities Transportation Center, 2010.

[45] H. H. Teng, V. Kwigizile, G. Xie, M. S. Kaseko, and A. R. Gibby. The Impacts of Emergency Vehicle Signal Preemption on Urban Traffic Speed. In *Journal of the transportation research forum*, volume 49(1), pages 69–79. Transportation Research Forum, 2010.

[46] I. Yun, B. B. Park, C. K. Lee, and Y. T. Oh. Comparison of emergency vehicle preemption methods using a hardware-in-the-loop simulation. *KSCE Journal of Civil Engineering*, 16(6):1057–1063, 2012.

[47] I. Yun, M. Best, and B. Park. Evaluation of emergency vehicle preemption strategies on a coordinated actuated signal system using hardware-in-the-loop simulation. In *Transportation Research Board 86th Annual Meeting*, 07-2415, 2007.

[48] X. Qin and A. M. Khan. Control strategies of traffic signal timing transition for emergency vehicle preemption. *Transportation research part C: emerging technologies*, 25:1–17, 2012.

[49] W. Kang, G. Xiong, Y. Lv, X. Dong, F. Zhu, and Q. Kong. Traffic signal coordination for emergency vehicles. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 157–161. IEEE, 2014.

[50] H. Noori, L. Fu, and S. Shiravi. Connected-Vehicle-Based Traffic Signal Control Strategy for Emergency Vehicle Preemption. In *Transportation Research Board 95th Annual Meeting*, 16-6763, 2016.

[51] H. K. Gedawy, M. Dias, and K. Harras. Dynamic path planning and traffic light coordination for emergency vehicle routing. Carnegie Mellon School of Computer Science, 2012.

[52] A. F. Ray and D. F. Kupas. Comparison of crashes involving ambulances with those of similar-sized vehicles. *Prehospital Emergency Care*, 9(4):412–415, 2005.

[53] L. R. Becker, E. Zaloshnja, N. Levick, G. Li, and T. R. Miller. Relative risk of injury and death in ambulances and other emergency vehicles. *Accident Analysis & Prevention*, 35(6):941–948, 2003.

[54] S. Yasmin, S. Anowar, and R. Tay. Effects of drivers' actions on severity of emergency vehicle collisions. *Transportation Research Record: Journal of the Transportation Research Board*, 2318:90–97, 2012.

[55] C. A. Kahn, R. G. Pirrallo, and E. M. Kuhn. Characteristics of fatal ambulance crashes in the United States: an 11-year retrospective analysis. *Prehospital emergency care*, 5(3):261–269, 2001.

[56] P. Savolainen, T. Datta, I. Ghosh, and T. Gates. Effects of dynamically activated emergency vehicle warning sign on driver behavior at urban intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2149:77–83, 2010.

[57] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber. A VANET-based emergency vehicle warning system. In *2009 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2009.

[58] G. Unibaso, J. Del Ser, S. Gil-Lopez, and B. Molinete. A novel CAM-based traffic light preemption algorithm for efficient guidance of emergency vehicles. In *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 74–79. IEEE, 2010.

[59] V. de Cózar, J. Poncela, M. Aguilera, M. Aamir, and B. S. Chowdhry. Cooperative vehicle-to-vehicle awareness messages implementation. In *Wireless Sensor Networks for Developing Countries*, pages 26–37. Springer, 2013.

[60] Y.-S. Huang, Y.-S. Weng, and M. Zhou. Design of traffic safety control systems for emergency vehicle preemption using timed petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2113–2120, 2015.

[61] L. Head, D. Gettman, and Z. Wei. Decision model for priority control of traffic signals. *Transportation Research Record: Journal of the Transportation Research Board*, 1978:169–177, 2006.

[62] Q. He, K. L. Head, and J. Ding. Multi-modal traffic signal control with priority, signal actuation and coordination. *Transportation Research Part C: Emerging Technologies*, 46:65–82, 2014.

[63] Q. He, K. Head, and J. Ding. Heuristic algorithm for priority traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board*, 2259:1–7, 2011.

[64] W. Ma, Y. Liu, and X. Yang. A dynamic programming approach for optimal signal priority control upon multiple high-frequency bus requests. *Journal of Intelligent Transportation Systems*, 17(4):282–293, 2013.

[65] M. Zlatkovic, A. Stevanovic, and P. Martin. Development and evaluation of algorithm for resolution of conflicting transit signal priority requests. *Transportation*

Research Record: Journal of the Transportation Research Board, 2311:167–175, 2012.

[66] A. Ben-Tal and A. Nemirovski. Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[67] M. Fellendorf and P. Vortisch. Microscopic traffic flow simulator VISSIM. In *Fundamentals of traffic simulation*, pages 63–93. Springer, 2010.

[68] P. VISION. PTV VISSIM 8 User Manual, 2015.

[69] J. Obenberger and J. Collura. Transition strategies to exit preemption control: State-of-the-practice assessment. *Transportation Research Record: Journal of the Transportation Research Board*, 1748:72–79, 2001.

[70] I. Yun, C. K. Lee, Y. T. Oh, et al. Investigation on the exit phase controls for emergency vehicle preemption. *KSCE Journal of Civil Engineering*, 15(8):1419–1426, 2011.

[71] W. R. Reilly and C. C. Gardner. Technique for measuring delay at intersections. *Transportation research record*, 644:1–7, 1977.

[72] S. Teply, D. Allingham, D. Richardson, and B. Stephenson. Canadian capacity guide for signalized intersections. *Canadian Institute of Transportation Engineers*, 2008.

[73] H. Kua. *Modified Whiting Delay Model for Signalized Intersections*. University of Alberta, 1990.