Dynamic Path Planning and Traffic Light Coordination for Emergency Vehicle Routing

Hend Kamal Gedawy

Advisors:

M. Bernardine Dias, Ph.D., Assistant Research Professor, Robotics Institute

Khaled Harras, Ph.D., Assistant Teaching Professor, Computer Science Department

Abstract— **An ambulance or fire truck arriving a couple of seconds late can be the difference between life and death for some. As different technologies emerge, various approaches to expediting the movement of emergency vehicles have evolved. Horns, sirens and flashing lights were early attempts that are no longer sufficient in most places to clear traffic on the emergency vehicle's route. In these situations, traffic signal preemption has made it possible to guide traffic to move in favor of clearing the emergency vehicle's route. Early traffic signal preemption approaches depended on direct communication between an emergency vehicle's signal emitter and a corresponding signal receiver on the traffic light it was approaching. Accordingly, the location of the vehicle could be detected. Later, (D)GPS was used to more accurately locate the emergency vehicle. This solution was further enhanced by using efficient or even optimal path planning algorithms to choose the route of the emergency vehicle. In the state-of-the-art in emergency vehicle routing, online static route selection is combined with traffic-light preemption to make emergency vehicle travel faster and safer along the chosen optimal path.**

In this thesis, we propose an enhancement to the state-of-the-art approaches for reducing the emergency vehicle's travel time. Our hypothesis is that combining traffic signal preemption with dynamic path planning will increase the efficiency of routing an emergency vehicle. We implement a graph version of the D*Lite informed search algorithm to efficiently and dynamically plan optimal paths for the emergency vehicle while taking into consideration the real-time updates of congestion levels and other delays to travel time. To further improve our solution, we propose a traffic light preemption strategy that seeks to ensure fast and safe travel of the emergency vehicle while, as a secondary priority, maximizes other traffic flow through the intersection. We evaluate our hypothesis through analytical experiments using our implementation of D* Lite, and further validate our proposed solution through scenarios developed using the VISSIM specialized microscopic traffic simulator [15]. The results validate our hypothesis demonstrating that dynamic path planning can improve travel time under uncertain congestion conditions, and that incorporating an appropriate traffic light preemption mechanism can further improve travel time for an emergency vehicle; potentially saving lives.

Keywords—– **Emergency vehicle routing, Traffic signal preemption, Dynamic path planning, Signal phase selection**

I. Introduction

Expedient movement of emergency vehicles to and from the scene of an accident can greatly improve the probability that lives will be saved. There are some potential barriers to this expedient movement that range from synchronized operation of traffic lights to the traffic conditions in the current and nearby intersections. Since traffic lights can control traffic flow at intersections, they can guide the flow of traffic to favour movement in the direction that best suits the emergency vehicle. This factor has been a key component for preventing traffic-related delays of emergency vehicles.

Since the 1960s, as different technologies were introduced, the approaches taken to solve the problem of effective emergency vehicle routing accordingly evolved. Initial strategies of using noisy sirens were deemed insufficient to clear traffic that blocked the path of an emergency vehicle. Thus, traffic lights were coordinated to move traffic in directions that would clear congestion on the route of the emergency vehicle. The main idea is to communicate the presence of the emergency vehicle to the relevant traffic lights and notify the traffic light of the emergency vehicle's position and direction of travel. Detectors for strobe lights and sirens were attached to traffic lights to enable them to recognize the emergency vehicles. However, these approaches required clear line of sight which was difficult to maintain in bad weather conditions, near curves, or most importantly, when obstacles prevented this line-of-sight detection. To address these difficulties, radio and microwave signal transceivers were deployed to improve the communication between the emergency vehicles and the traffic lights [2][3][4][5][6][7]. When Differential Global Positioning System (D)GPS technology emerged it was possible to track the position of emergency vehicles more accurately and indicate the arrival of an emergency vehicle at an intersection much earlier [8][9].

A later improvement that was added to emergency vehicle routing to further reduce its travel time was static path planning, where algorithms were employed to choose the fastest path to the destination based on the congestion information available at the time of planning the path. This approach was adopted by Kim and Kwon whose solution is to use Dijkstra's algorithm for static path planning for the emergency vehicle and dynamically preempt the traffic lights as the vehicle travels along its route [9]. Their solution provides an online route selection module that when queried checks the current traffic conditions and statically recalculates the least cost path from the vehicle's location to the goal.

Although path planning is an enhancement of previous approaches in that it views the route in its entirety and not just the local impediments, path planning approaches have adopted a static perspective on route planning. This ignores the possibility that costs are constantly changing because the level of congestion keeps changing over time. While Kim and Kwon [9] provide an online route selection module, it is not until a replanning request is made that the current changes in traffic conditions is taken into consideration. This is not efficient if the replanning occurs after the vehicle is stuck in a congested route. Also, it is neither time efficient nor effective to plan from scratch, which is what happens with Dijkstra's algorithm. Algorithms that locally replan by modifying previous search results have been shown to replan one to two orders of magnitudes faster than planning from scratch [10].

Accordingly, one possible development that can be built on the previous approaches to further improve the emergency vehicle's travel time is to efficiently and dynamically plan the emergency vehicle's route depending on the updated traffic conditions. Hence, this paper presents a solution that combines dynamic path planning with a corresponding traffic light preemption plan. The dynamic path planning uses the D* Lite informed search algorithm that can efficiently and optimally plan and replan according to changing costs in the traffic network. The cost of a given route translates to the travel time of that route since the main goal is to reduce the emergency vehicle's delay. For a chosen route, a preemption strategy is used to select a signal phase that reduces the emergency vehicle's delay and, when possible, maximizes flow for other vehicles approaching the intersection.

We evaluate our work at two levels. First, an analytical evaluation is conducted using the graph version of D*Lite we implemented. Extra costs are assigned to nodes or intersections when preemption is not in effect, and extra costs are assigned to edges to represent the effects of congestion. Our analytical results clearly demonstrate reduced travel times for emergency vehicles when using dynamic path planning combined with preemption; especially in the presence of unforeseen congestion in the road network. Our second level of evaluation is based on evaluating appropriate scenarios using the high fidelity VISSIM microscopic traffic simulator [15].

The rest of this thesis is organized as follows. In Section II we present our related work and discuss the limitations in the current state-of-the-art solutions. Afterwards, we explain our proposed solution in detail in Section III. We then evaluate our solution and results in Section IV. Finally, in Section V, we conclude and discuss future work.

II. Related Work

A. Existing Approaches

In this section we review the state of the art in emergency vehicle routing techniques.

1) Direct communication techniques: The main idea of the direct communication solutions is to attach a device to the emergency vehicles to communicate with a suitable receiver in the traffic lights' control system [2][3][4][5][6][7]. Early systems depended on strobe lights emitted from the vehicle being detected by optical receivers on the traffic lights. Then directional microphones were used to detect the sirens of the emergency vehicle. However, these two approaches required clear line of sight. This requirement limits their functionality. When different obstacles or bad weather conditions affect the communication, i.e. the signal, traffic lights fail to correctly detect the presence of the emergency vehicle near the intersection. Later, microwave and radio signal transceivers were used for earlier detection of the emergency vehicle since these signals could reach further and could overcome the clear line of sight limitation introduced in the previous approaches.

2) (D)GPS-dependent approaches: The next generation of approaches took advantage of the Differential Global Positioning System (D)GPS for real-time data of the actual vehicle location. This helped to detect the location of the emergency vehicle more accurately and start the traffic light preemption process earlier. Radio transceivers were used to communicate the GPS data between the vehicle and each traffic light the vehicle approached. The (D)GPS transmitted a signal either directly to an intersection controller or to a central server that could dynamically preempt traffic lights along the vehicle's route [8][9].

3) Adding static path planning: Lights, radio transmitters, and DGPS all address the need for preempting traffic signals. More recent work has suggested that traffic preemption techniques should be combined with efficient path planning [9]. Path planning avoids delaying the emergency vehicle in a congested or otherwise delayed route by choosing the best (fastest) route for the vehicle given the current best information. To apply path planning to the emergency vehicle routing problem, the traffic network is abstracted into nodes representing intersections and links representing roads. The costs assigned to the links are estimates of travel time along these links and they depend on different factors, including the level of congestion on these links. This approach was taken by two researchers Kim and Kwon, who used Dijkstra's algorithm for static path planning for the emergency vehicle and designed a preemption scheme for the traffic lights along the emergency vehicle's route [9]. Their solution provides an online route selection module that when queried checks the current traffic conditions and statically recalculates the least cost path from the vehicle's location to the goal.

B. Limitations

The main limitation in the previous approaches is that path planning approaches have adopted a static perspective on route planning which ignores the uncertainty in the level of congestion which could change in many ways due to events such as accidents or adverse weather conditions. Even though Kim and Kwon provide an online route selection module, it needs to be triggered by a replanning request from the driver of the emergency vehicle and it cannot repair its plan; but instead must plan from scratch. This is a limitation of the choice of Dijkstra's algorithm for path planning. We propose to address this limitation by using the D* Lite optimal dynamic planning algorithm coupled with an appropriate traffic light preemption mechanism.

III. Proposed Solution

A. Assumptions

Before delving into the details of the solution, we explore some basic assumptions on which the proposed solution builds. First, we assume that traffic lights are controlled by a central server with an overall view of the entire traffic network. This is not a strong assumption as long as we can get updated congestion information during runtime of the emergency vehicle, and if we are allowed to execute our preemption mechanism on any traffic light as needed. Our second assumption is that we have uninterrupted connectivity between the emergency vehicle and the central server. This can be done through many technologies which in cases might be expensive but worth deploying for emergencies. Our third assumption is the existence of a reliable congestion detection system that can measure the congestion level on the roads of the network and report it to the central server. Fourth, we assume the emergency vehicle has a GPS (or other) tracking system that can accurately determine its current location at

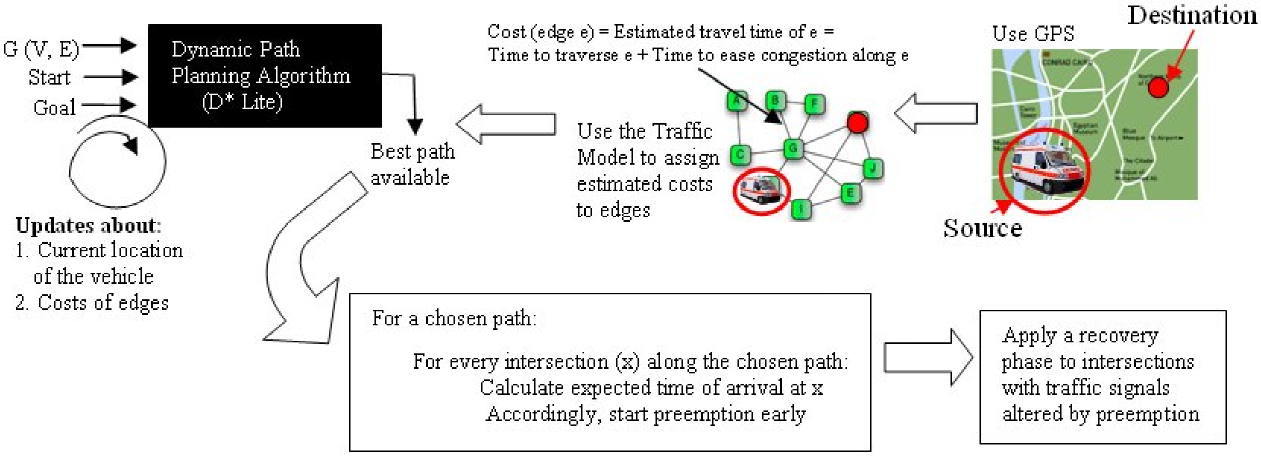

G (V, E)
Start
Goal
Dynamic Path Planning Algorithm (D* Lite)
Updates about:
1. Current location of the vehicle
2. Costs of edges
Best path available
Cost (edge e) = Estimated travel time of e = Time to traverse e + Time to ease congestion along e
Use the Traffic Model to assign estimated costs to edges
A
B
F
C
G
J
E
I
Use GPS
Destination
Source
For a chosen path:
For every intersection (x) along the chosen path:
Calculate expected time of arrival at x
Accordingly, start preemption early
Apply a recovery phase to intersections with traffic signals altered by preemption

Figure 1: Proposed Solution Overview

any point during its traverse and send this information back to the server. Our fifth and final assumption is access to a traffic model that provides useful estimates of traffic flow patterns so we can seed our path planning algorithm and preemption mechanism with reasonable default values.

B. Solution Components

The proposed solution consists of two main components: dynamic path planning and preemption. As shown in Figure 1, the destination and current location of the emergency vehicle is determined using data from the GPS tracking system. We then model the traffic network between the source and destination as a graph where nodes represent intersections and edges represent the roads in between. Costs are assigned to both nodes and edges. Since the goal of the whole solution is to reduce the emergency vehicle's travel time, the cost of a particular edge is translated to the time needed to travel along that road and the cost of a node is the time to cross that intersection. This estimated travel time includes reasoning about congestion. Once costs are assigned to edges and nodes, the graph is provided as an input to the dynamic path planning algorithm; i.e. D* Lite. The algorithm keeps receiving updates about the current location of the emergency vehicle using GPS data and the current costs of the edges using the assumed congestion detection system, and accordingly outputs the best route available at any point in time. After identifying a path, the preemption process can start early depending on the time the vehicle is expected to arrive at intersections along that route. Finally, for every intersection affected by the preemption process, a recovery phase is applied to the traffic lights to restore back their normal operation once the emergency vehicle passes through.

1) Dynamic path planning: D* Lite is an informed or heuristic search algorithm invented by Koenig and Likhachev in 2002 and since then has been used in different applications for dynamic path planning. It can be customized for a specific domain using an appropriate heuristic. This heuristic is selected specific to the problem domain and guides the search. It is an enhancement of another informed search algorithm called A*, and specifically Lifelong Planning A*. Most of D* Lite properties are inherited from Lifelong Planning A*. However, unlike Lifelong Planning A* that does forward search, D* Lite does backward search. It searches starting from the goal to the start state. This property makes repairing the current plan easier in D* Lite because the goal stays fixed and the start node keeps moving around [11]. D* Lite is a simpler version of the D* algorithm previously invented by Stentz [16].

D* Lite works as follows. It maintains two estimates of node's costs. One is called g and is an estimate of the objective function value while the other, rhs, is one-step look-ahead estimate of the objective function value. A node is considered to be consistent as long as these two estimates are equal. If a node is not consistent it is added to a priority queue for processing. Nodes are added and processed each depending on its priority or key (k) which is defined depending on the heuristic and the two estimates of the node, as shown in (1)[10].

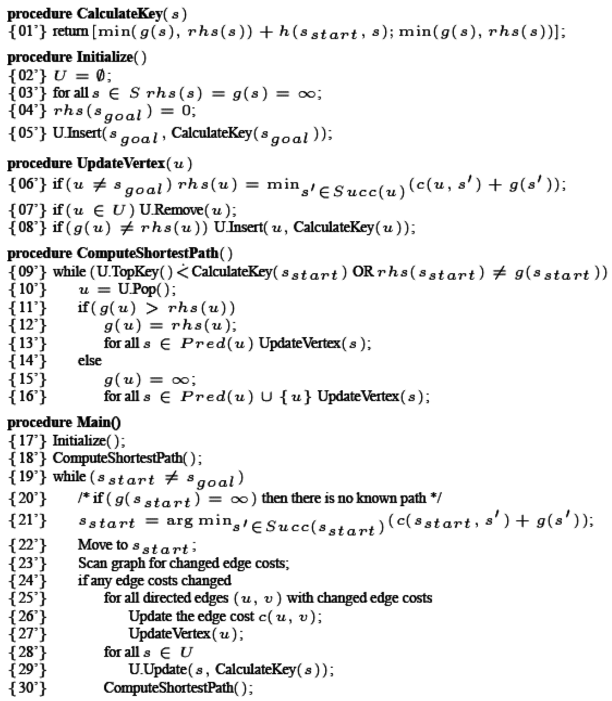
$$k(s) = [k_1(s); k_2(s)] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$k_1(s) = \min(g(s), rhs(s)) + h(s_{start}, s\ ) \ldots\ldots\ldots\ldots(2)$$

$$k_2(s) = \min(g(s), rhs(s)) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

$$k(s) \le k'(s) \text{ iff either } k_1(s) \le k_1'(s) \text{ or}$$
$$k_1(s) = k_1'(s) \text{ and } k_2(s) \le k_2'(s) \ldots\ldots\ldots\ldots\ldots(4)$$

Figure 2 shows pseudo code for the D* Lite algorithm [10]. First, `the algorithm calls Initialize(), line {17'}, which initializes g and rhs values of the nodes. It sets all of them to infinity since their cost is unknown yet but the rhs value of goal is set to zero. The priority queue now has only the goal vertex since it is the only inconsistent vertex. Its g value is infinity and its rhs value is zero. Then, the algorithm computes the shortest path, as shown in line {18'} from the current start vertex to the goal. If the goal is not yet reached, then one step or transition is made along the shortest path towards the goal and the start state is updated to be the current vertex that we stepped to, lines {21'-22'}. The algorithm then checks for edges with changed costs, {23'}, and for each of these it calls UpdateVertex() procedure to update the rhs value of vertices affected by the changed costs. Accordingly, the priority queue is updated to contain all the inconsistent nodes. In line {29'} the priorities of all vertices in the priority queue are updated and shortest path is recalculated to incorporate the changes, line {30'}.



```
procedure CalculateKey($s$)
{01'} return $[\min(g(s), rhs(s)) + h(s_{start}, s); \min(g(s), rhs(s))]$;
procedure Initialize()
{02'} $U = \emptyset$;
{03'} for all $s \in S$ $rhs(s) = g(s) = \infty$;
{04'} $rhs(s_{goal}) = 0$;
{05'} U.Insert($s_{goal}$, CalculateKey($s_{goal}$));
procedure UpdateVertex($u$)
{06'} if ($u \neq s_{goal}$) $rhs(u) = \min_{s' \in Succ(u)}(c(u, s') + g(s'))$;
{07'} if ($u \in U$) U.Remove($u$);
{08'} if ($g(u) \neq rhs(u)$) U.Insert($u$, CalculateKey($u$));
procedure ComputeShortestPath()
{09'} while (U.TopKey() $\dot{<}$ CalculateKey($s_{start}$) OR $rhs(s_{start}) \neq g(s_{start})$)
{10'}     $u$ = U.Pop();
{11'}     if ($g(u) > rhs(u)$)
{12'}         $g(u) = rhs(u)$;
{13'}         for all $s \in Pred(u)$ UpdateVertex($s$);
{14'}     else
{15'}         $g(u) = \infty$;
{16'}         for all $s \in Pred(u) \cup \{u\}$ UpdateVertex($s$);
procedure Main()
{17'} Initialize();
{18'} ComputeShortestPath();
{19'} while ($s_{start} \neq s_{goal}$)
{20'}     /* if ($g(s_{start}) = \infty$) then there is no known path */
{21'}     $s_{start} = \arg\min_{s' \in Succ(s_{start})}(c(s_{start}, s') + g(s'))$;
{22'}     Move to $s_{start}$;
{23'}     Scan graph for changed edge costs;
{24'}     if any edge costs changed
{25'}         for all directed edges $(u, v)$ with changed edge costs
{26'}             Update the edge cost $c(u, v)$;
{27'}             UpdateVertex($u$);
{28'}         for all $s \in U$
{29'}             U.Update($s$, CalculateKey($s$));
{30'}         ComputeShortestPath();
```

Figure 2: D* Lite Pseudo Code [10]

We chose D* Lite for applying dynamic path planning to emergency vehicle routing for a number of principled reasons. First, it is provably complete, optimal, and can handle dynamic cost changes [10]. This feature is significant for vehicle path planning based on traffic conditions, such as congestion level, that can suddenly change at anytime. Second, D* Lite implements fast replanning because it does not plan from scratch but rather fixes the current plan considering those edges affected with the changes and only the ones that matter to the least cost path. During emergencies, this fast replanning can save the emergency vehicle unnecessary time spent planning from scratch. Third, D* Lite makes no assumptions about how the costs of the edges are changing. This is necessary for accurate routing of emergency vehicles because the cost or travel time of the roads can increase dramatically due to sudden and unexpected incidents, like car accidents. Thus, we should not assume anything about how costs are changing in this case.

To apply D* Lite to emergency vehicle's path planning, we implemented a graph version of the algorithm which was needed to incorporate the intersections, the roads and their associated costs. It is written in JAVA and intersections are modelled as Node objects while the roads are defined as Edge objects. Since D* Lite requires an admissible heuristic to guarantee an optimal path, we chose our heuristic to be an estimate of the lowest possible travel time on a link or road in the network or graph (which is computed as the link distance divided by the Vehicle's maximum speed). This heuristic is admissible, i.e. does not overestimate cost, because it estimates the travel time based only on distance and the maximum speed of the vehicle. While there are other factors, like congestion, that might affect travel time of a link or road, these factors do not always exist. Thus, adding them to the heuristic would be an overestimation and would make the heuristic inadmissible. Our objective function is travel time since we wish to minimize the travel time of the emergency vehicle as it travels from its start to destination.

2) Signal preemption: Signal preemption involves extending the current phase of a traffic light, or ending the current phase and switching to another phase to allow fast and safe travel of the emergency vehicle through that intersection. We address two main questions related to preemption. The first question is how should preemption be done? The second is when or how early should preemption start?

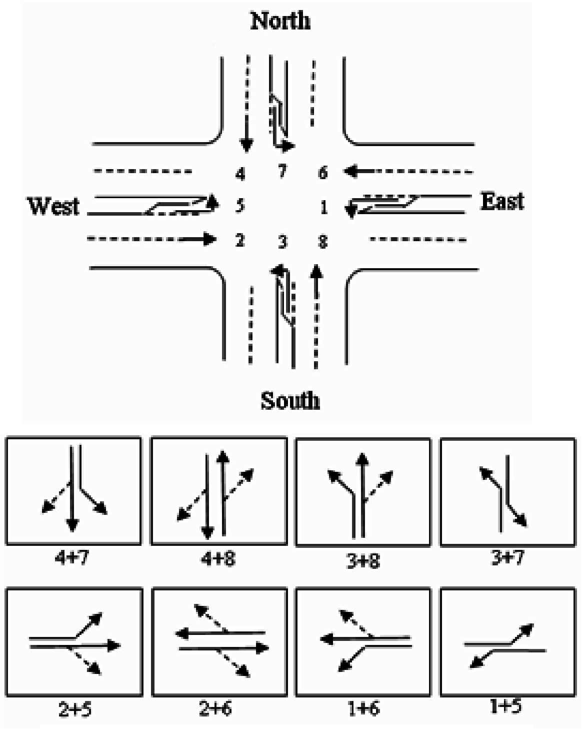


Figure 3: Defining traffic signal phases [9]

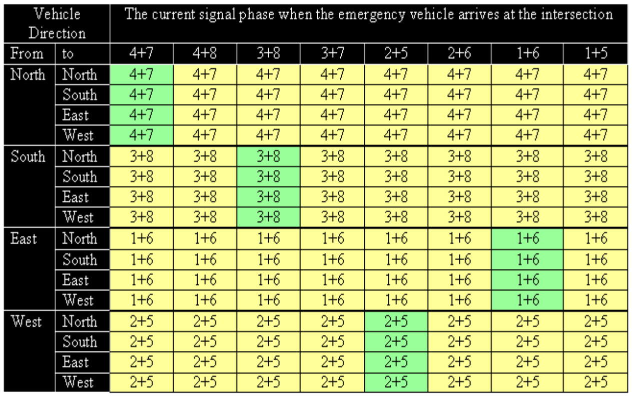
How should preemption be done?

Deciding how preemption is done mainly depends on satisfying the expedient movement of an emergency vehicle and then, if possible, maximizing the flow of other traffic at the intersection. Maximizing the flow on the other approaches, i.e. other than the one the emergency vehicle is coming from, of the intersection is not possible unless doing this does not affect the emergency vehicle's delay and safety. With this goal in mind, the traffic lights phase selection depends on the congestion level, the current phase of the traffic lights when preemption is triggered, and on the intersection approach from which the vehicle is coming and to which the vehicle is heading. Figure 4 shows an example of traffic light phases that can be used to explain the phase selection algorithm. These phases were borrowed from the work by Kim and Kwon [9]. We use the same example but develop our own preemption plan.

In this example, the possible destinations of a vehicle coming from any of the four sides of the intersection are indicated by numbers. A phase reflects the set of traffic lights, indicated by numbers that have a green light at the intersection. As shown in Figure 4, the phases' cycle starts with (4+7) which means that traffic lights 4 and 7 are green and every other traffic light at the intersection is red.

Now that the traffic lights phases are identified, it is important to consider how to control these phases to achieve the aforementioned preemption goal. This control depends on the level of congestion as we demonstrate below.

To begin with, if the level of congestion is high, then delay can be imposed onto the emergency vehicle's route. Traffic, therefore, needs to be cleared around the emergency vehicle as much as possible. Given the current phase of the traffic lights and the emergency vehicle's origin/destination, we built Table 1 to indicate the best phase selection to ensure fast and safe travel of the emergency vehicle through the intersection.

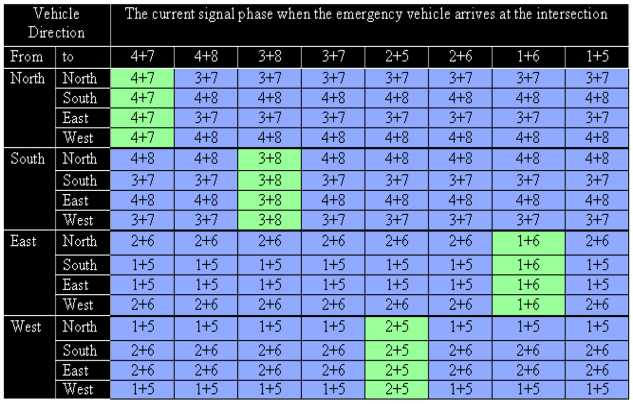


Vehicle Direction		The current signal phase when the emergency vehicle arrives at the intersection							
From	to	4+7	4+8	3+8	3+7	2+5	2+6	1+6	1+5
North	North	4+7	4+7	4+7	4+7	4+7	4+7	4+7	4+7
	South	4+7	4+7	4+7	4+7	4+7	4+7	4+7	4+7
	East	4+7	4+7	4+7	4+7	4+7	4+7	4+7	4+7
	West	4+7	4+7	4+7	4+7	4+7	4+7	4+7	4+7
South	North	3+8	3+8	3+8	3+8	3+8	3+8	3+8	3+8
	South	3+8	3+8	3+8	3+8	3+8	3+8	3+8	3+8
	East	3+8	3+8	3+8	3+8	3+8	3+8	3+8	3+8
	West	3+8	3+8	3+8	3+8	3+8	3+8	3+8	3+8
East	North	1+6	1+6	1+6	1+6	1+6	1+6	1+6	1+6
	South	1+6	1+6	1+6	1+6	1+6	1+6	1+6	1+6
	East	1+6	1+6	1+6	1+6	1+6	1+6	1+6	1+6
	West	1+6	1+6	1+6	1+6	1+6	1+6	1+6	1+6
West	North	2+5	2+5	2+5	2+5	2+5	2+5	2+5	2+5
	South	2+5	2+5	2+5	2+5	2+5	2+5	2+5	2+5
	East	2+5	2+5	2+5	2+5	2+5	2+5	2+5	2+5
	West	2+5	2+5	2+5	2+5	2+5	2+5	2+5	2+5

Table 1: Traffic signal phases selected when level of congestion is high

The rows represent the directions of the emergency vehicle at the intersection, while the columns describe which of the eight phases has the green lights on when the preemption is triggered. Given these two inputs, the intersection of the row and the column gives the best phase that should be green for the emergency vehicle to pass through quickly and safely. The results of the table can be summarized as follows: regardless of the current traffic light phase when the emergency vehicle arrives, and regardless of its destination on the intersection, all traffic lights on the approach it is coming from has to be green. This is reflected in the four phases: (4+7) if the vehicle is approaching the intersection from the North, (3+8) if it is coming from South, (2+5) if it is coming from West and (1+6) if it is coming from East. Doing this ensures that the emergency vehicle would not get delayed by traffic on its route. The green coloured cells represent the case where the preemption is triggered when the optimal phase is on so we just extend the green light for the phase until the emergency vehicle has cleared the intersection.

On the other hand, if the level of congestion is low then the emergency vehicle's expedient movement is quite easy to achieve so traffic flow on other approaches of the intersection can be maximized as well. Thus, the phase selection criteria becomes as follows. If the current phase allows the vehicle to get through to its desired approach in the intersection, we extend the phase. Otherwise, we pick the phase that allows the emergency vehicle through the intersection and maximizes vehicle flow on the other sides of the intersection. Applying these criteria to the eight defined phases, given the current phase when preemption is initiated and the desired origin/destination intersection approaches of the emergency vehicle, we get Table 2.



Vehicle Direction		The current signal phase when the emergency vehicle arrives at the intersection							
From	to	4+7	4+8	3+8	3+7	2+5	2+6	1+6	1+5
North	North	4+7	3+7	3+7	3+7	3+7	3+7	3+7	3+7
	South	4+7	4+8	4+8	4+8	4+8	4+8	4+8	4+8
	East	4+7	3+7	3+7	3+7	3+7	3+7	3+7	3+7
	West	4+7	4+8	4+8	4+8	4+8	4+8	4+8	4+8
South	North	4+8	4+8	3+8	4+8	4+8	4+8	4+8	4+8
	South	3+7	3+7	3+8	3+7	3+7	3+7	3+7	3+7
	East	4+8	4+8	3+8	4+8	4+8	4+8	4+8	4+8
	West	3+7	3+7	3+8	3+7	3+7	3+7	3+7	3+7
East	North	2+6	2+6	2+6	2+6	2+6	2+6	1+6	2+6
	South	1+5	1+5	1+5	1+5	1+5	1+5	1+6	1+5
	East	1+5	1+5	1+5	1+5	1+5	1+5	1+6	1+5
	West	2+6	2+6	2+6	2+6	2+6	2+6	1+6	2+6
West	North	1+5	1+5	1+5	1+5	2+5	1+5	1+5	1+5
	South	2+6	2+6	2+6	2+6	2+5	2+6	2+6	2+6
	East	2+6	2+6	2+6	2+6	2+5	2+6	2+6	2+6
	West	1+5	1+5	1+5	1+5	2+5	1+5	1+5	1+5

Table 2: Traffic signal phases selected when level of congestion is low

An example to clarify the algorithm under low congestion conditions is when the vehicle is heading from North to South, i.e. second row in Table 2, on the intersection while the green signal phase is (3+8), i.e. third column. Considering the phase choices available to get a vehicle from North to South on the intersection, there are two phases, (4+8) and (4+7). Since the level of congestion is low, emergency vehicle's delay is not a concern in either case. However, delaying other traffic on the other sides of the intersection is a concern. Choosing (4+8) would allow vehicles coming from South and heading East or North to move through. However, choosing (4+7) would keep these vehicles waiting unnecessarily. Thus, the Table outputs (4+8) in this case. The rest of the cells are filled by applying the same analysis.

Comparing the two tables, the green cells are the cases on which they match. In other words, these are where high or low congestion makes no difference in our phase selection decision. This result indicates that preemption phase selection is not affected by congestion in one eighth of the scenarios and therefore congestion needs to be taken into consideration in most preemption scenarios.

When should preemption start?

How early preemption should be triggered for a given intersection approach also differs depending on congestion level. The base case is when there is no congestion. In this case, the factors affecting how early preemption should occur are the time the vehicle is expected to arrive at the traffic light and the phase transition time, i.e. time needed to end the current traffic signal phase and switch to another one. A safety margin is also needed for ensuring no delay affects the vehicle's travel time. If the vehicle is approaching intersection traffic light X and expected to be at the stop line of the traffic light at time τ and the transition time is *Trans*, then time to start preemption at X , or

$$p_{nc} = \tau - Trans - safety\ m\arg in \text{ ……..………………..(5)}$$

On the other hand, if congestion is introduced into the network, then preemption should start earlier considering the congestion effect on the vehicle's travel time. For a given approach to an intersection, there are two possible delays that can affect the travel time of vehicles on that approach. This is based on a new time dependent travel time estimation model, developed by Linu and Ma, that has proven to be quite accurate [12]. The first delay is the signal delay which they define as the time spent at the traffic light due to red light. However, since preemption occurs before the vehicle is at the traffic light, this delay will not affect the emergency vehicle. The second delay is the queuing delay, $t_q(\tau)$, which they defined as the time needed to clear queue of vehicles between the vehicle and the traffic light stop line at time τ. This delay is very likely during congestion and it can affect the emergency vehicle's travel time. Thus, taking this delay into account, preemption at X now becomes (6):

$$p_c = p_{nc} - t_q(\tau) \quad \text{……………………………..…………..(6)}$$

$$t_q(\tau) = l_1 + h \times n_q(\tau) \quad \text{……………………………………….(7)}$$

$$n_q(\tau) = \begin{cases} \max\{0,\ A(\tau) - D(t_G^0) - s \times (\tau - t_G^0)\}, & \textit{when green is on} \\ A(\tau) - D(t_G^0 + g), & \textit{otherwise} \end{cases} \quad \text{..(8)}$$

According to Linu and Ma, queuing delay at time τ can be estimated by equation (7), depending on the number of vehicles in the queue (8) [12]. h is the saturation headway or the constant headway achieved once a stable moving queue is established and is defined in (seconds/vehicle)[14]. When the traffic stream starts, the first several vehicles consume more than h (seconds/vehicle). Adding the incremental headways of these vehicles gives the value l_1, the Start-up lost time [14]. $n_q(\tau)$ is the number of vehicles in the queue at time τ which is the difference between the arrival counts and the departure counts. Detectors are used before the signal stop line and at the stop line to calculate these counts. $A(\tau)$ represents the arrival count at time τ. t_G^0 is the start of the green time of the current cycle, g is the green time of the current cycle, and $D(t_G^0)$ and $D(t_G^0 + g)$ are the departure counts at the start of the current green, and the end of the current green respectively. s is the saturation flow rate or the number of vehicles that can enter the intersection in a single lane if the signal were always green for that lane and vehicles were never stopped [14].

IV. EVALUATION

A. Analytical Evaluation

We first evaluate our hypothesis analytically using our D*Lite graph-based implementation where costs are assigned to edges for queuing delay due to congestion and costs are assigned to nodes for signal delay. These test cases were run with two graphs, as shown in Figure 4. One is simple with one decision point between the source, a, and the destination, d. The other graph is more complex where there are many routes that can be taken between the start, A, and destination M. The numbers on the edges indicate the length of the network roads in Kilometres. The emergency vehicle's max speed is assumed to be 120 Km/hr. To test cost and intersection delay, when preemption is not done, a 3 minutes cost was assigned to nodes, or intersections, between the start and the destination. In cases where there is no congestion, queuing delay is assumed to be zero. When there is congestion, a 7 minutes estimated cost is assigned to some edges in the network to indicate the queuing delay mounting at these edges. As a lower bound in terms of delay, preemption is assumed to be done perfectly leaving no congestion along the emergency vehicle's route.

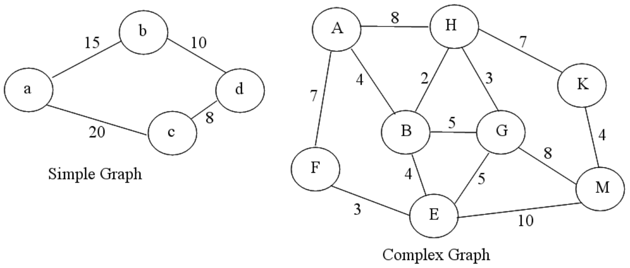


Figure 4: Road Networks Used in Testing

Overall, as results indicate in Figures 5 and 6, dynamic path planning shows significant improvement over static path planning. The difference is more prominent in the complex graph scenario. When there is zero congestion, then static and dynamic path planning give the same travel time results as one would expect.

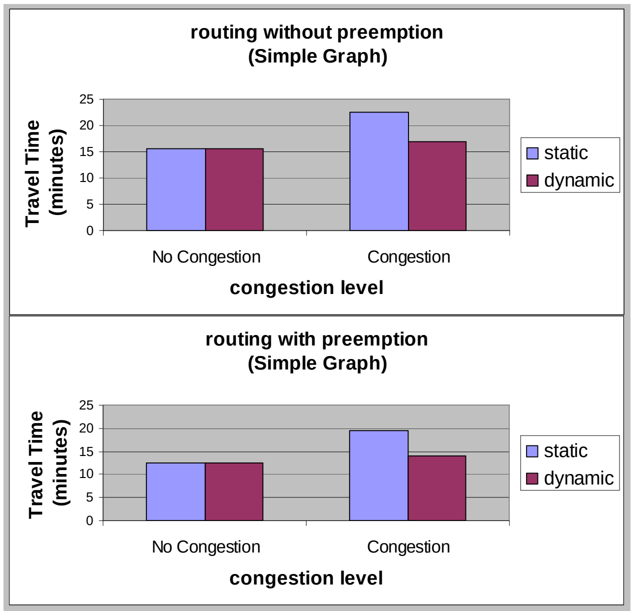


Figure 5: Analytical Result for Simple Graph

However, when congestion occurs, a clear travel time improvement is observed with dynamic path planning. The most obvious difference in performance is in the complex case where there is no preemption and a lot of congestion. This analytical result indicates that even in cities where doing preemption is not feasible, dynamic routing alone can make a significant difference. When preemption is added a clear cut in travel time takes place in both dynamic and static routing but dynamic path planning saves more time because it preempts the least cost path.

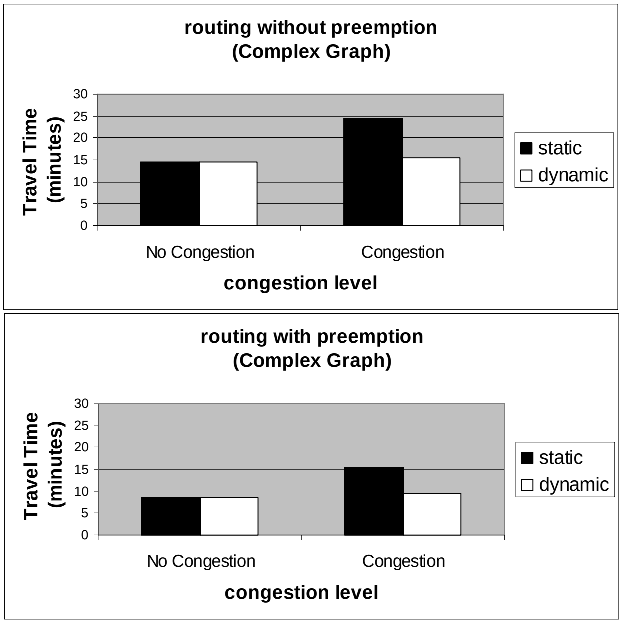


Figure 6: Analytical Results for Complex Graph

Another set of test cases on the complex graph were run to evaluate the average impact of using dynamic routing with preemption on the traffic network with emergency vehicles coming and heading to different parts of the network. To further reflect real-life scenarios, random roads or links are congested with random levels of congestion as well. To overcome the problem of having random start and goal points and to make the comparison meaningful, the numbers plotted represent an increase percentage relative to a base travel time specific to the case. The base case picked in each test case was the travel time value when there is no congestion and no signal delay. The results are reflected in Figure 7. The figure shows the results of ten runs of the test case where in each run the percentage increase of travel time over an optimal case is calculated. As the results show, preemption alone buys us more time than dynamic routing alone does. However, when both are combined the curve dives the most, compared to any other combination.

As shown in Figure 8, the next evaluation on the congested complex graph considered randomized signal and queuing delay costs and their effect on the emergency vehicle's travel time. Again, the vehicle's max speed was 120.

For the queuing delay, we fixed the signal delay at 3 minutes, when preemption does not exist, and randomly chose different queuing delay costs in a range between zero and seven minutes. The same links in the network were assigned the different queuing delay costs. As the results indicate, the vehicle's travel time with dynamic path planning stays constant as the queuing delay increases since the route is replanned and the high congested routes are avoided. However, for static routing the travel time increases with queuing delay since we stick to the same route even though it gets more and more congested. As the results also show adding preemption, in this case, saves around 5 to 10 minutes of the emergency vehicle's travel time.

For the signal delay, we fixed queuing delay at 7 minutes for a chosen set of edges and randomly chose different signal delay costs in a range between zero and three minutes. With preemption, assuming it is working perfectly, there is no signal delay and the vehicle is not stopped at red traffic lights. Thus, the curves stay constant. However, when preemption does not exist, signal delay has a major effect on the travel time. As the results also show, the travel time curve for static routing grows steeper than the dynamic routing curve when preemption does not exist.

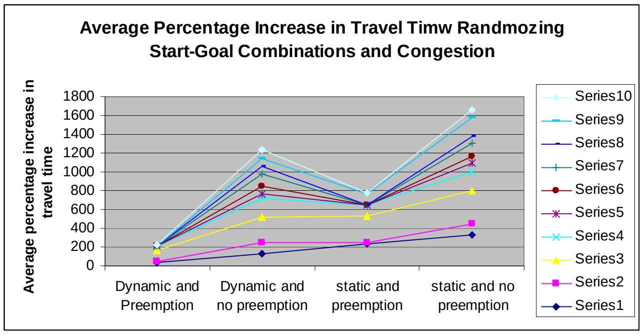


Figure 7: Randomized start-goal states and congestion

Another test that goes along with our hypothesis is shown in Figure 9. This is again a combination of randomized set of values for signal and queuing delays. The test proves that overall, regardless of the costs introduced into the network, dynamic path planning combined with pre-emption outperforms everything else.

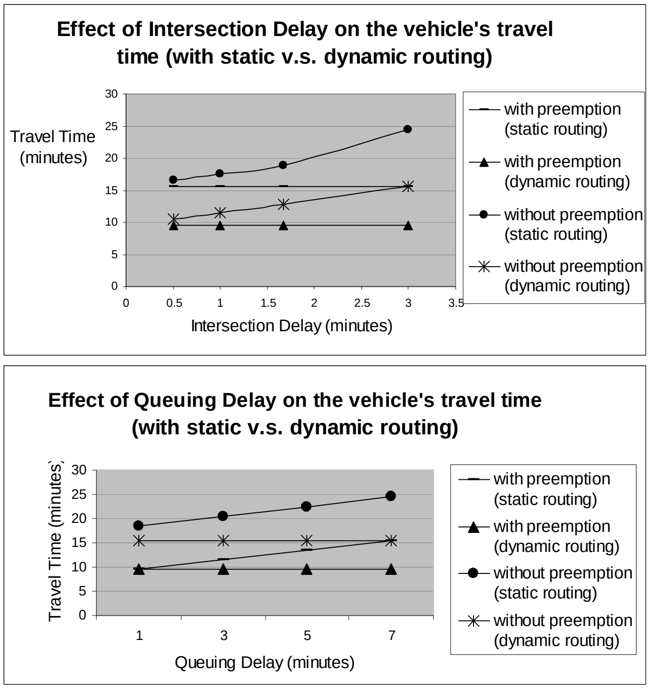


Figure 8: Second Set of Evaluations (Complex Graph with Congestion)

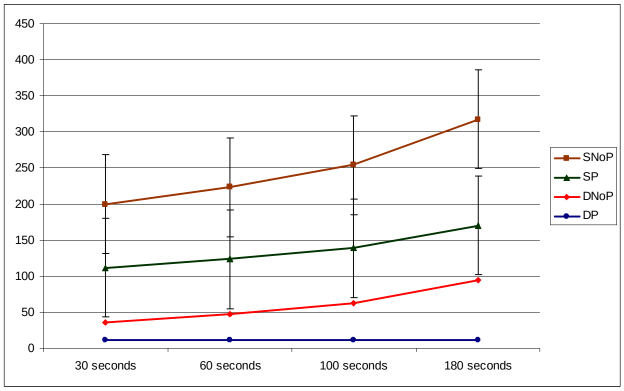


Figure 9: Intersection and Queuing Delay Effect

B. Evaluation via Simulation

At this point we understand that our analytical results abstract many details in a real traffic network. To further verify our results, we test our solutions and algorithms over a more realistic environment using a specialized simulator. The complementary set of results we present was conducted using the VISSIM microscopic traffic simulator which we have been exploring. VISSIM was developed in Germany by Planung Transport Verkehr (PTV) company. It simulates urban and highway traffic, including pedestrians, cyclists and motorized vehicles. It has been used to model different traffic engineering problems [13]. One major advantage of VISSIM over other existing traffic simulators is its flexible structure that allows continuous interaction between VISSIM’s main simulation module and an external module that controls every traffic light in the network [9]. Thus, for this research’s interest, the preemption algorithm and dynamic path planning algorithm can be separately coded into an external module that would continuously interact with VISSIM. VISSIM in turn gives simulated detector data to the external module.

However, configuring VISSIM is non-trivial and we were unable to master all of its features in the duration of this thesis work. In fact, VISSIM courses are offered at the graduate level and for professionals due to the complexity of its configuration, and mastering VISSIM often takes several years of practice. Nevertheless, we have developed an initial scenario of a Manhattan Grid network (zoomed in screenshots shown in Figure 10). This scenario sets the traffic lights to behave in the phases defined in Figure 3.

We simulated the dynamic path planning in VISSIM by modeling the VISSIM road network in our D* Lite implementation and manually feeding the resulting path to VISSIM. To measure the emergency vehicle’s delay under different traffic levels when dynamic versus static routing is used, different runs were conducted using VISSIM. We injected different traffic volumes on a specific route and measured the travel time it takes the emergency vehicle to get through. Since in static path planning, the planned path is never changed in response to traffic, the travel time cost increases as traffic along the chosen route increases. As Figure 11 indicates, the more the injected number of vehicles per hour, the more the travel time is. Now, comparing this to dynamic path planning, we notice a big difference and improvement in terms of travel time. The algorithm allows the vehicle to route and reroute avoiding this congested part of the network.

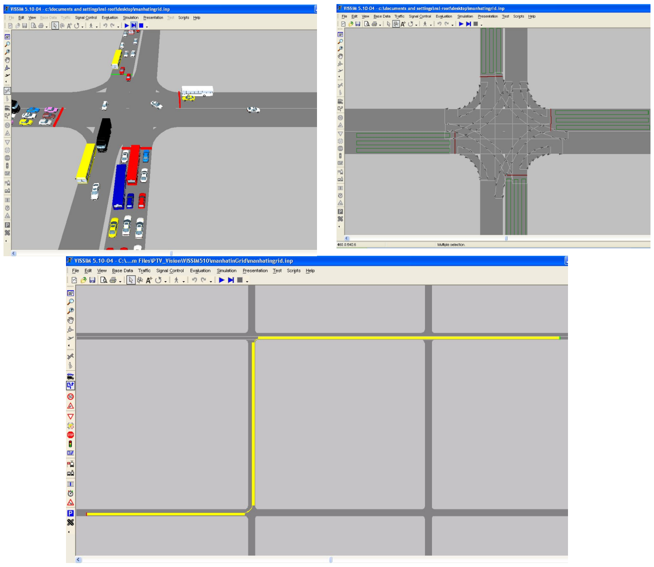


Figure 10: Snapshots of VISSIM Scenario

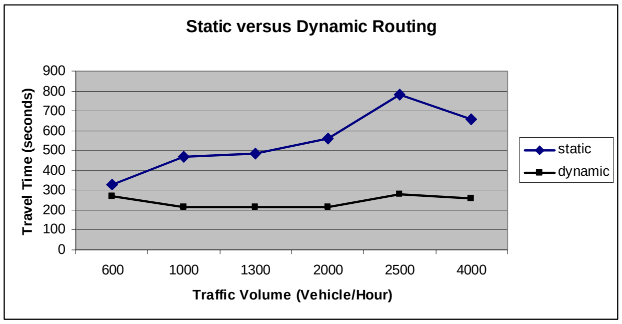


Figure 11: Static v.s. Dynamic Travel Time with Different Traffic Volumes.

V. Conclusion and Future Work

This thesis improves the state of the art in emergency vehicle routing by introducing dynamic path planning combined with traffic light preemption. As our results demonstrate, dynamic path planning has proven to reduce the emergency vehicle’s travel time. Even if not combined with preemption, dynamic path planning shows significant improvement over static path planning. Thus, in countries where doing preemption is not feasible, dynamic path planning can still play a significant role in saving lives.

There are three enhancements we are planning for future work. First, we plan to build a complete COM interface between the path planner, the preemption mechanism, and VISSIM. The second enhancement is considering the emergency vehicle’s vicinity in the preemption plan since extended congestion from neighbour intersections can affect the traffic flow at the intersection closest to the emergency vehicle. Doing preemption only along the emergency vehicle’s route, does not guarantee the smooth travel of the emergency vehicle. This has been a major limitation in all of the previous work. The third enhancement is handling multiple priorities at an intersection with the approach that we discussed in this research paper.

Acknowledgements

- I wish to acknowledge the support and encouragement of my advisors, Professor M. Bernardine Dias and Professor Khaled Harras for their support throughout the year. I also acknowledge the support of Professor Amer Shalaby, Associate Professor in the Department of Civil Engineering in the University of Toronto, who helped me a lot with his expertise in Traffic Engineering. I also would like to thank my supportive classmates, class of 2009, and my very supportive family.

References

[1] The National Transportation Safety Board and the Federal Highway Administration (2006). Traffic Signal Preemption for Emergency Vehicles. A CROSS-CUTTING STUDY.

[2] Schwab, Bruce B. (1959). Remote emergency traffic control system. United States Patent Office.

[3] Malach. (1975). Emergency Vehicle traffic controller. US patent.

[4] Anderson, George P. (1980). Emergency Vehicle traffic control system. United States patent.

[5] Mitchell, Wilbur L. Traffic Light control for Emergency Vehicles. United States patent.

[6] Robertson, Michael T. (1994). Traffic Light control means for emergency-type vehicles. United States Patent.

[7] King, Frederick N. (1982). Detection System for Emergency Vehicles. United States Patent.

[8] Brooke, O. (2005). Centralized traffic signal preemption and method of use. United States Patent.

[9] Kwon, E., & Kim, S. (2003). Development of dynamic route clearance strategies for Emergency Vehicle operations phase I.

[10] Koenig, S. & Likhachev, M. (2002). Improved fast replanning for robot navigation in unknown terrain. *Robotics and Automation.* Proceedings. ICRA '02. IEEE International Conference on , vol.1, no., pp. 968-975 vol.1, 2002

[11] Browning, B. & Dias, B. (2008). Artificial Intelligence. *PowerPoint presentation.*

[12] Koenig, Sven, & Likhachev, Maxim (2007). Time-Dependent Travel Time Estimation Model for Signalized Arterial Network. *Transportation Research Board 86th Annual Meeting.*

[13] Planung Transport Verkehr AG (2008). Vissim User Manual, V5.10. Germany

[14] Pernia J. (2006). Traffic Flow principles. *PowerPoint presentation.*

[15] http://www.vissim.com/

[16] Stentz, A. (1995) Optimal and Efficient Path Planning for Unknown and Dynamic Environments. *International Journal of Robotics and Automation*, Vol. 10, No. 3.