الجامعة المصرية اليابانية للعلوم والتكنولوجيا

E-JUST

Egypt - Japan University of Science and Technology

エジプト日本科学技術大学

ADAPTIVE MULTI-OBJECTIVE REINFORCEMENT LEARNING FOR TRAFFIC SIGNAL CONTROL BASED ON COOPERATIVE MULTI-AGENT FRAMEWORK

A THESIS

Submitted to the Graduate School of
Electronics, Communication and Computer Engineering,
Egypt-Japan University of Science and Technology (E-JUST)

In Partial Fulfillment of the Requirements for the Degree
of
Doctor of Philosophy
in
Computer Science and Engineering

by

Mohamed AbdElAziz Khamis Omar

August 2013

Adaptive Multi-Objective Reinforcement Learning for Traffic Signal Control Based on Cooperative Multi-Agent Framework

Submitted by
Mohamed AbdElAziz Khamis Omar
For The Degree of

Doctor of Philosophy
in
Computer Science and Engineering

Supervision Committee **Signature**

Assoc. Prof. Ahmed El-Mahdy, Computer Sci. and Eng., E-JUST, Alex., Egypt ....
Assist. Prof. Walid Gomaa, Computer Sci. and Eng., E-JUST, Alex., Egypt ....
Prof. Amin Shoukry, Computer and Systems Eng., Fac. of Eng., Alex. Univ., Egypt ....

Examination Committee **Approved**

Prof. Amin Shoukry, Computer and Systems Eng., Fac. of Eng., Alex. Univ., Egypt ....

Prof. Kazunori Ueda, Computer Sci. and Eng., Waseda Univ., Tokyo, Japan ....

Prof. Ikuo Takeuchi, Computer Sci. and Eng., Waseda Univ., Tokyo, Japan ....

Assoc. Prof. Ahmed El-Mahdy, Computer Sci. and Eng., E-JUST, Alex., Egypt ....

Dr. Eng. Ahmed I. Mosa, Ministry of Transport. Advisor for Transport. Planning ....

Prof. Ahmed Abo Ismail

Vice President for Education and Academic Affairs (Provost)

Summary

The traffic of vehicles in urban areas has many problems that include the increase of traffic congestion and psychological stress of drivers leading to high rate of accidents, considerable time losses, and high rate of vehicle emissions. Accordingly, those problems have a considerable negative effect on the country economy. One possible solution for the traffic problem which has a great impact in tackling those negative effects is using *traffic signal control*.

Hence, in this thesis, we focus on computing a consistent traffic signal configuration at each junction that optimizes multiple performance indices, i.e., multi-objective traffic signal control. The multi-objective function includes minimizing trip waiting time, total trip time, and junction waiting time. Moreover, the multi-objective function includes maximizing flow rate, satisfying green waves for platoons traveling in main roads, avoiding accidents especially in residential areas, and forcing vehicles to move within moderate speed range of minimum fuel consumption.

In particular, we formulate our multi-objective traffic signal control as a Multi-Agent System (MAS). Traffic signal controllers have a distributed nature in which each traffic signal agent acts individually and possibly cooperatively in a MAS. In addition, agents act autonomously according to the current traffic situation without any human intervention. Thus, we develop a multi-agent multi-objective Reinforcement Learning (RL) traffic signal control framework that simulates the driver's behavior (acceleration/deceleration) continuously in space and time dimensions.

This framework has two main challenges; the formulation of the learning task as a multi-objective function and the application of this multi-objective learning framework on continuous space-time models. Particularly, the proposed framework is based on a multi-objective sequential decision making process whose parameters are estimated based on the Bayesian interpretation of probability. Using this interpretation together with a novel adaptive cooperative exploration technique, the proposed traffic signal controller can make real-time adaptation in the sense that it responds effectively to the changing road dynamics.

These road dynamics are simulated by the Green Light District (GLD) vehicle traffic simulator that is the testbed of our traffic signal control. GLD is an open-source traffic simulator that facilitates the traffic signal control research by using customizable road networks. We have implemented the Intelligent Driver Model (IDM) acceleration model in the GLD traffic simulator. The change in road conditions is modeled by varying the traffic demand probability distribution and adapting the IDM parameters to the adverse weather conditions.

For better performance evaluation, we added new performance indices based on collaborative learning to the GLD traffic simulator. These performance indices are based on estimating the remaining time of vehicles till arriving to their destinations rather than depending only on the elapsed time in the whole trip of the arrived vehicles.

Moreover, we added new performance indices, i.e., Measures Of Effectiveness (MOEs) to evaluate the various system objectives, e.g., average number of trip stops, average speed, percentage of arrived vehicles, percentage of rejected vehicles, maximum and average queue lengths, etc.

Under the congested and free traffic situations, the proposed multi-objective controller significantly outperforms the underlying single objective controller which only minimizes the trip waiting time (i.e., the total waiting time in the whole vehicle trip rather than at a specific junction). For instance, the average trip and waiting times are lower $\simeq 8$ and 6 times respectively when using the multi-objective controller.

To my parents and brothers who have always been my support in life
I dedicate this work.

Acknowledgements

I would like to show my sincere gratitude to my university Egypt-Japan University of Science and Technology (E-JUST). I would like to show my gratitude to the International Business Machines (IBM) Corporation for granting me the prestigious IBM PhD Fellowship for three consecutive years. I am grateful to my supervisor Dr. Walid Gomaa for his direct supervision and guidance throughout my PhD period. Special thanks are due to Prof. Ahmed El-Mahdy, the director of the Smart City Facility at E-JUST, for fruitful discussions. I would like to thank Dr. Hisham El-Shishiny, the Manager of IBM Center for Advanced Studies in Cairo and the World Wide Leader of IBM on Campus Program, for fruitful discussions.

Special thanks are due to Prof. Amin Shoukry, Prof. Ikuo Takeuchi and Prof. Kazunori Ueda for giving me the chance to present my work in several events (e.g., JEC-ECC 2012 conference, E-JUST-Waseda CSE tele-workshop, etc.) and feedback me with valuable comments.

I would like to thank Prof. Marco Wiering for providing me with the latest source code of the Green Light District (GLD) traffic simulator and answering my questions concerning the GLD. In addition, I would like to thank Prof. Martin Treiber for helping me understanding the Intelligent Driver Model (IDM) model. Moreover, I would like to thank Prof. Carlos Gershenson, Prof. Bram Bakker, Prof. Shimon Whiteson, Merlijn Steingröver, and Gaston Escobar for making the task of understanding the GLD traffic simulator easier.

Finally, I would like to show my gratitude to my department professors, my colleagues and the whole E-JUST staff members who helped me throughout my whole work and made it a rich experience from which I have learned a lot.

Table of Contents

TABLE OF CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

AA	Accidents Avoidance.
AI	Artificial Intelligence.
AJWT	Average Junction Waiting Time.
ANNs	Artificial Neural Networks.
ATT	Average Trip Time.
ATWT	Average Trip Waiting Time.
CF	Congestion Factor.
DP	Dynamic Programming.
EAs	Evolutionary Algorithms.
FC	Fuel Consumption.
FL	Fuzzy Logic.
FR	Flow Rate.
GAC	Gain Adapted by Congestion.
GAs	Genetic Algorithms.
GDP	Gross Domestic Product.
GLD	Green Light District.
GW	Green Wave.
IDM	Intelligent Driver Model.
ITS	Intelligent Transportation Systems.
MARL	Multi-Agent Reinforcement Learning.

ABBREVIATIONS

MAS	Multi-Agent System.
MDP	Markov Decision Process.
ML	Machine Learning.
MOBIL	Minimizing Overall Braking decelerations Induced by Lane changes.
MOEs	Measures Of Effectiveness.
MOGA	Multi-Objective Genetic Algorithm.
MPC	Model Predictive Control.
MS	Moderate Speed.
OPAC	Optimization Policies for Adaptive Control.
POMDP	Partially Observable Markov Decision Process.
RHODES	Real-Time Hierarchical Optimized Distributed and Effective System.
RL	Reinforcement Learning.
SARSA	State-Action-Reward State-Action.
SBC	State Bit for Congestion.
SCATS	Sydney Coordinated Adaptive Traffic System.
SCOOT	Split Cycle and Offset Optimization Technique.
SOTL	Self-Organizing Traffic Lights.
V2I	Vehicle-to-Infrastructure.

Chapter 1

Introduction

In this thesis, we focus on computing a consistent traffic signal configuration at each junction that optimizes multiple performance indices (i.e., multi-objective traffic signal control). Traffic signal control can be viewed as a multi-objective optimization problem. The multi-objective function can have a global objective for the entire road network or there may be different objectives for the different parts of the road network (e.g., maximize safety especially in residential and school areas), or even different times of the day for the same part of the road network.

Construction of a new infrastructure is expensive, thus the generally acceptable solution is to improve the utilization of the existing resources by moving towards Intelligent Transportation Systems (ITS) for traffic management and control. Traffic control is a set of methods that are used to enhance the traffic network performance by, for example, controlling the traffic flow to minimize congestion, waiting times, fuel consumption and avoid accidents. Traffic control generally includes the following components; controlling the traffic signals in urban areas, ramp-metering in highways, enforcing variable speed limits (according to vehicles types), supporting the drivers with route guidance based on the up-to-date traffic status using some kind of navigation systems (e.g., GPS), enforcing overtaking rules, and using driver-assistance systems (e.g., adaptive cruise control). In this thesis, we particularly focus on controlling traffic signals in urban areas.

Another two important components of the ITS are traffic modeling and traffic simulation. Traffic modeling is the formulation of rigorous mathematical models that represent the various dynamics of the traffic system. This includes drivers' behavior in acceleration, deceleration, lane changing, phenomena such as rubbernecking, and behavior change under different weather conditions. Traffic simulation is the virtual emulation of the traffic system on digital computers. Traffic simulators are used for experimentation and validation of the underlying traffic models and traffic control mechanisms.

1.1 Motivation

Intelligent traffic control has many challenges that include the continuing increase in the number of vehicles (it is expected that 70% of the people worldwide will live in urban areas by 2050 [1]), the high dynamics and non-stationarity of the traffic network, and the nonlinear behavior of the different components of the control system. Nowadays, the different types of transportation means (specifically vehicles in urban areas) have major problems that governments are facing in both developing and developed countries. Traffic of vehicles in urban areas, specifically, has many problems that include increase of traffic congestion, psychological stress of drivers that affects the drivers' behavior leading to high rate of accidents, considerable time losses, and high rate of vehicle emissions which severely affects the environment. Those problems have a considerable negative effect on the country economy. Thus, in this thesis, the proposed traffic signal controller tackles most of those problems (e.g., minimizes the waiting time of vehicles) as will be shown by the performance evaluation in Chapter 6.

In 2010, traffic costs (based on time loss and fuel consumption) about $115 billion in the US based on 439 urban areas [2]. In the same year, 32,885 people died in accidents in the US [3]. According to [4], the phase 2 congestion analysis ingoing by the World Bank estimates that congestion in Cairo will cost Egypt yearly in the range of 40-50 billion EGP through direct and indirect means. The economic costs of congestion includes [4] travel delays, wasted fuel, health impacts (due to polluted air and accidents), and economic productivity. Particularly, the yearly economic cost of Cairo traffic congestion could reach up to 4% of Egypt Gross Domestic Product (GDP) [4]. Moreover, in Egypt, traffic problems are responsible for more than 25,000 accidents in 2010 with more than 6,000 deaths per year [5]. Deaths per million driving kilometers in Egypt is about 34 times greater than in the developed countries [6]. This value is about 3 times greater than countries in the Middle East region [6]. We expect that this value is much worse in 2011-2013 due to the political upheaval in Egypt.

1.2 Problem Definition

Recently, some computer science tools and technologies have been used to address the traffic signal control complexities. Among these is the MAS framework whose characteristics are similar in nature to the traffic problem [7, 8]. Such characteristics include distributivity, autonomy, intelligibility, on-line learnability, and scalability. In particular, the formulation of the traffic signal control problem as a Multi-Agent Reinforcement Learning (MARL) is very promising (as proposed in [9]). Hence, in this thesis, we adopt a MARL framework [10], in a cooperation-based configuration, to comply with the distributed nature and complexity of the problem. This particular version proves its effectiveness when being applied to large scale traffic networks. In contrast, other controllers e.g., [11, 12] suffer from exponential state-space when being applied to large scale traffic networks. Thus, the main technical challenges of traffic signal control in urban areas are:

1. Traffic signal control application on large scale traffic networks.

2. Being adaptive to changing traffic dynamics.

3. Being multi-objective, i.e., optimizes multiple performance indices at a time.

4. Traffic signal control validation/simulation on continuous space-time models.

In this thesis, we present how we tackle these challenges (especially when the traffic signal control framework is based on RL). Particularly, our objective in this thesis is to develop a traffic signal control framework with the following characteristics:

1. Inherently distributed through the use of a *multi-agent system*; there are two types of agents:

   - **Traffic junction agents** (active computing agents) which are responsible for the decision making process (i.e., deciding on the proper traffic signal configuration) according to the information collected from the vehicle agents.
   - **Vehicle agents** (passive agents) which support the decision making process by communicating the necessary information to the junction agents.

2. Online sequential decision making framework where decisions are taken in real-time for signal splitting based on multiple optimization criteria:

   - The core of the applied mechanism is based on Dynamic Programming (DP) which is very-well suited for sequential decision making tasks.
   - The real-time optimization and decision making is done incrementally by integrating the online learning with DP through the use of reinforcement learning.

3. Effectively and efficiently handle the inherent complexity of the problem, the uncertainties involved, the incompleteness of information, the absence of a rigorous modeling of the traffic volume and the general dynamics:

   - Through the use of stochastic and statistical tools to predict the unknown parameters and provide an up-to-date model of the current traffic conditions.

4. Adaptive system in the sense that it responds effectively to the road dynamics (variations in traffic demand, changing weather conditions, etc.):
    - Through the use of a Bayesian approach for estimating the parameters of the underlying Markov Decision Process (MDP) and the use of an adaptive cooperative hybrid exploration technique.
5. Higher confidence in the validity of the proposed traffic signal controller:
    - Through the use of a more realistic simulator as a testbed.
    - This is achieved by implementing the IDM acceleration model [13] in the GLD vehicle traffic simulator [14].
    - Moving from the unrealistic discrete-time discrete-space simulation platform to a continuous-time continuous-space one.

The discrete-time discrete-space simulation platform was unrealistic in the sense that the first waiting vehicle jumps once the traffic signal turns green. Now, by applying the more realistic IDM acceleration model, the vehicle takes the normal time to decelerate when a traffic signal turns red and accelerates back again to cross the junction when the traffic signal turns green. This behavior, on the other side, causes some kind of sign oscillation when being applied on the underlying learning model as will be shown later in Chapter 4 (which we called the *Zeno phenomena* [1]) which results from the very slow acceleration of back vehicles when the traffic signal is just turning green.

1.3 Contributions

The thesis contributions are:

1. Proposing a multi-objective RL traffic signal controller with the following objectives:

[1] A *Zeno phenomena* occurs due to the infinitesimal motion of a particle continuously within the same state.

- Minimizing the Average Trip Waiting Time (ATWT).
- Minimizing the Average Trip Time (ATT) especially in main roads.
- Minimizing the Average Junction Waiting Time (AJWT).
- Maximizing the Flow Rate (FR).
- Satisfying Green Wave (GW)[1] for platoons traveling in main roads.
- Accidents Avoidance (AA) in residential areas.
- Forcing vehicles to move within Moderate Speed (MS) range of minimum Fuel Consumption (FC).

2. Using the Bayesian probability interpretation to estimate the unknown parameters of the MDP of the traffic control model. This estimation allows the multi-objective controller to make real time (online) adaptation in the sense that it responds effectively to the changing environment and the non-stationarity of the road network.

3. Proposing a novel cooperative hybrid exploration technique which is more adaptive to the changing dynamics in road conditions, i.e., improves the trip waiting time of vehicles during transient periods (e.g., due to rush hours).

4. Analyzing and fixing some crucial problems that appeared in the original RL traffic control model [10], particulary when applying the IDM time-continuous acceleration model [13]. This acceleration model is more realistic than the previous contributions which are applied on time-discrete models, e.g., [10, 15, 16].

5. Checking the traffic signal controller against various driver behaviors (i.e., acceleration/deceleration) and traffic demands which are resulting from simulating some adverse weather conditions.

[1] A green wave is achieved when consequent traffic signals are set to green when a platoon of vehicles are approaching the traffic signals. This phenomena usually occurs under a free traffic condition.

6. Adding new performance indices based on collaborative learning to the GLD traffic simulator [14] for better performance evaluation. These performance indices are based on estimating the remaining time of vehicles till arriving to their destinations rather than depending only on the elapsed time in the whole trip of the arrived vehicles.

7. Adding new performance indices (MOEs) to the GLD traffic simulator to evaluate the system performance (e.g., average speed, average number of trip stops, average number of trip absolute stops, percentage of arrived vehicles, percentage of rejected vehicles, maximum and average queue lengths.)

8. Presenting a survey of the state-of-the-art work.

1.4 Thesis Organization

The thesis is organized as follows. The related work on urban traffic signal controllers is presented in Chapter 2. A background on the adopted traffic signal control and simulation models is presented in Chapter 3. The proposed framework including the improvements on the traffic signal control and simulation models is presented in Chapter 4. Traffic non-stationarity is tackled by two models; MDP parameter estimation using the Bayesian probability interpretation and a novel adaptive cooperative hybrid exploration technique. These two models are presented in Chapter 5. Our multi-objective RL traffic signal control framework and performance evaluation of the whole system are presented in Chapter 6. Finally, Chapter 7 concludes the thesis and proposes some directions for future work. Appendix A gives background on the reinforcement learning, while Appendix B gives background on the Minimizing Overall Braking decelerations Induced by Lane changes (MOBIL) lane changing model which we implemented in the GLD.

Chapter 2

Related Work

2.1 Introduction

There have been several approaches proposed in the literature for traffic signal control. For instance, the well-known Sydney Coordinated Adaptive Traffic System (SCATS) [17] and Split Cycle and Offset Optimization Technique (SCOOT) [18] are based on complex mathematical models to optimize the traffic signal switching times. These systems have improved the traffic performance in many countries. However, as mentioned in [19, 20], these systems have the overhead of advanced personnel training, problems of gaining acceptance, generally centralized and suffer from inefficient handling of saturated traffic conditions due to the *inadequate real-time adaptability.* In addition, human intervention in unexpected situations (e.g., accidents) is in many cases unavoidable.

Other approaches such as the Optimization Policies for Adaptive Control (OPAC) [21] and Real-Time Hierarchical Optimized Distributed and Effective System (RHODES) [22] calculate the traffic signal switching times by solving *dynamic optimization problem* in real-time. As mentioned in [20, 23], these systems are hard for large scale deployment due to *exponential complexities.* In this chapter, we give further classifications for the state-of-the-art traffic signal controllers.

2.2 Traditional vs. Adaptive Traffic Signal Controllers

The two broad classes of the traffic signal controllers are: traditional control paradigms and adaptive control paradigms. On the one hand, the simplest intuitive type of traffic signal control is to allow every traffic direction to pass for a fixed amount of time. This of course ignores the dynamics and the high variability of the traffic network. Thus, this strategy can result in very poor utilization of the traffic system and inefficient usage of the available resources.

On the other hand, traffic signal controllers based on robust models, e.g., petri-nets [24, 25], Model Predictive Control (MPC) [8, 26], etc., are hard to design and require a complete match with the actual traffic network dynamics for optimal traffic signal control. In particular, as mentioned in [27], any uncertainty or mismatch in the network model will result in a suboptimal performance of the MPC. Hence, these models are rigid and non-adaptive to non-modeled variations.

Some traffic signal controllers are based on the dynamic programming algorithmic paradigm, e.g., [28, 29]. DP is inherently a paradigm for sequential decision making hence it is very well suited to the nature of traffic signal control. However, most traffic signal controllers based on DP are applied on an isolated junction, thus it does not take into account the inter-dependability between the different parts of the traffic network. In addition, most traffic prediction is based on historical traffic data that is taken in the same time of the day during which traffic is being controlled, e.g., [29].

2.3 MAS-Based Traffic Signal Controllers

The formulation of the traffic signal control problem as a MAS tackles the inherent complexity of this problem (i.e., exponential state-space of the whole system). In particular, the MAS formulation is a distributed way of modeling the problem, i.e., no centralized state of the network; we do not need to model all possible states for the system.

Rather, each junction agent only observes the partial state of the intersection under-control, e.g., [30].

2.3.1 Single Agent vs. Multi-Agent Learning

There are two possible configurations for formulating the traffic signal control problem as a MAS:

- A *single agent* is responsible for controlling the traffic at one junction independently from the other junctions, e.g., [10–12].
- A *multi-agent system* where there is a joint traffic signal configuration between neighboring agents, e.g., [16, 31, 32]. For instance, in [31], each agent plays a game with all its neighboring junctions; the state-space and action-space are distributed such that the agent learns the joint policy with one of its neighbors.

In the work presented in this thesis, every agent is responsible for controlling a specific junction in the traffic network. The learning in this model is *single-agent* RL that is optimized to allow for small state-space. Nevertheless, the cooperation between neighboring agents is done on two levels:

- The *transient state* (e.g., due to rush hours) are transferred from one agent to its neighbors in order to change their exploration policy.
- The *degree of congestion* in the next lane after a vehicle crosses the current junction and joins the lane of the next junction is considered in the traffic signal decision.

2.3.2 State Definition in a Multi-Agent System

In a MAS-based traffic signal control framework, a system state can be represented on three levels:

- A *vehicle state* in which a vehicle makes a transition from one state (i.e., current position) to the next state (i.e., next position) due to a transition signal (i.e., red or green).
- A *controller state* in which every junction represents the positions of the vehicles in its ingoing lanes to be the current junction state.
- A *system state* in which the partial states gathered from junction agents are integrated together to represent the whole system state (with an exponential state-space complexity).

In the work presented in this thesis, we adopt a *vehicle-based* state-space representation [10]. As mentioned in [33], in this state-space representation, the global system state is *approximated* by decomposing it into local states based on the vehicles traveling in the network. In addition, this state-space representation has a number of advantages [33]:

1. The number of states grows *linearly* in the number of lanes and cells, thus will scale well for large networks.
2. Another vehicle information (rather than the vehicle location) can be included in the state representation, e.g., vehicle speed, vehicle destination, etc.
3. Due to the direct correlation between the size of the state-space and the convergence of the Q-value [1], the Q-values will converge relatively fast compared to methods were the increase in the size of the state-space is exponential. Note that in this representation the Q-table is partitioned across the controllers.

2.4 AI-Based Traffic Signal Controllers

Modern traffic signal controllers tend to be *more adaptive* to the up-to-date traffic conditions than traditional traffic signal controllers (e.g., fixed-time traffic signal controllers).

[1] Refer to Appendix A for a background on RL.

That is if a change occurs in the network dynamics (due to accidents, rush hours, etc.), those traffic signal controllers change accordingly the traffic signal configuration by the way that optimizes the various performance indices (e.g., waiting time, queue lengths, etc.). These controllers are mainly based on Artificial Intelligence (AI) approaches, specifically based on Machine Learning (ML) techniques. There are two broad classes of ML techniques; *parametric* and *non-parametric*. On the one hand, *non-parametric* ML techniques can be used to implicitly capture the control model from the training data. On the other hand, *parametric* ML techniques find the optimal estimated value for the control model parameters (e.g., cycle time, offsets, splits, etc.) based on the training data.

For instance, parametric learning models are robust in the sense that there is no need for a complete mathematical model of the environment. Such controllers include Artificial Neural Networks (ANNs), e.g., [34, 35], Fuzzy Logic (FL), e.g., [36, 37], Evolutionary Algorithms (EAs), e.g., [38,39]. However, most of these approaches have the same problem of being only applied on small scale traffic networks. Moreover, most controllers are hard to be applied on large scale traffic networks due to computational space and time constraints. For instance, as mentioned in [40], traffic signal control methods based on FL, e.g., [41] are more suitable to control traffic at an isolated junction.

As mentioned in [40], EAs such as Genetic Algorithms (GAs) and Ant Algorithms, e.g., [42, 43] can not be easily applied for online optimization of large scale traffic coordinated control due to their characteristics of random search and implicit parallel computing. These methods will spend huge time to converge to the optimal traffic signal decision for large scale networks. This may result in offline decision making that is not suitable to the online traffic signal control application.

Generally, most of the previous work that is based on ML are non-adaptive in the sense that the dynamics of the environment is assumed to be non-changing (i.e., stationary). Particularly, after reaching steady state, the above learning algorithms can effectively converge to reasonable optimal configuration. However, if the road conditions change (due to rush hours, weather conditions, etc.), these methods fail to adapt to the new conditions, hence the performance indices might overshoot. In our traffic signal control framework, we handle the traffic non-stationarity using:

1. Bayesian probability interpretation for estimating the parameters of the MDP; this estimation was found to be more stable, robust, and adaptive to the changing environment dynamics.

2. A novel adaptive cooperative exploration technique.

We discuss these approaches in details in Chapter 5.

2.5 RL-Based Traffic Signal Controllers

Another class of AI traffic signal controllers are based on RL (e.g., [10–12, 44, 45]). In RL methods, each agent learns how to control a traffic signal through its interaction with the environment and gain some feedback (reward signal). Through a trial-and-error process, the agent learns a policy that optimizes the cumulative reward it gains over time. This process is based on a sequential online decision making process.

The application of RL in the context of traffic signal control is pioneered by Thorpe and Anderson [11]. This approach is based on a State-Action-Reward State-Action (SARSA) RL algorithm. A discrete state-action space is used to represent the states and actions. A state is described by the queue length, the vehicles positions in the queue, and the elapsed time of the current traffic signal. A single controller is trained on a single junction. After training is completed, the state of this single controller is replicated to all junctions of the traffic network. This system outperformed both fixed and rule-based controllers in some realistic simulation with changing speed.

In [46], the authors applied Q-learning to dynamically control traffic signals at an isolated junction. The traffic state includes the green phase index, the green signal lasting time, the traffic volume during green phase, the mean number of queued vehicles during red phase, and the traffic flow trend prediction. The learner action is to change the green phase to the red phase, or to extend the green phase until the next decision point. The reward is the traffic volume during the green phase divided by the waiting time increase during the red phase.

In [12], the authors applied Q-learning to a traffic signal control system. For an isolated junction, a state includes the queue lengths on the four main roads and the elapsed phase time. The actions include extending the current phase or switching to the next phase. The reward (a penalty in this case) is the total delay occurred between successive decision points by vehicles in the queues of the four main roads. The delay is a power function of the queue length. The reward is the weighted summation of the rewards of all isolated junctions. The reward of the main road is weighted more heavily.

The methods described above suffer from the growth in the number of states when scaling to larger networks even when some state information is excluded (e.g., the controller uses only the queue length). Thus, those methods were only applied to relatively small scale traffic networks (e.g., a single junction) or training a single junction and using the same policy for a number of junctions.

In [16], the authors extended Wiering RL model for traffic signal control [10] by using max-plus and coordination graphs. This work implements an explicit coordination mechanism between the learning junction agents. The max-plus algorithm is used to estimate the optimal joint action by sending the locally optimized messages between neighboring junctions. However, as mentioned in [31], the max-plus algorithm is computationally demanding and therefore the agents report their current best action at anytime even if the action found so far is sub-optimal. In particular, this coordination-based mechanism comes at the cost of adding one more dimension to the underlying Wiering RL model for maintaining the joint action-space. Moreover, the max-plus technique will not outperform Wiering RL model in two cases [33]:

1. The relatively large amount of *local* traffic (anti-cooperation based configuration).
2. All junctions are connected to a number of edge nodes (i.e., not connected to other junctions).

In [20], the authors proposed a collaborative RL approach using a local adaptive round robin phase switching model at each junction. Each junction collaborates with neighboring junctions in order to learn appropriate phase timing based on traffic patterns. In [47], the authors exploited the *natural actor-critic* algorithm which is based on four RL methods, i.e., policy gradient, natural gradient, temporal difference, and least-square temporal difference. The authors extended the state-space of the agent to include the state of other agents to control a 10×10-junction grid. In [48], a distributed traffic signal control method using ML-based neural networks have been proposed. In this approach, RL is used to control *only* the central junction in a network of 5 junctions while the other 4 junctions use the longest-queue-first algorithm and collaborate with the central agent by providing it with the local traffic statistics. However, due to the large state-space of junction-based methods (as will be detailed in the next sub-section), neural networks are used for better searching the state-space.

2.5.1 Exponential Explosion of Junction-Based State-Space Representation

Almost all RL-based traffic signal controllers are inherently based on MAS frameworks. Those systems could be classified according to their state-space representation into: *junction-based* state-space representation, e.g., [12, 31] and *vehicle-based* state-space representation, e.g., [10, 33]. Most RL-based traffic signal controllers proposed in the literature have *junction-based* full state representation, e.g., [11, 12, 31, 32]. For instance, in [12], a lookup table is required to store the Q-values for every possible (s, a)-tuple (as mentioned in [49]). This formulation is useful in small state-action space, however, it is computationally demanding for large traffic networks that involve multiple junctions. For instance, as mentioned in [49], the state-action space of a 3 × 3 grid is estimated at the size of 10^{101}. Moreover, as mentioned in [49], for a small traffic network consisting of 2 junctions with 10 signalized lanes with 20 vehicles per lane, the size of the state-action tuples (and hence the size of the $Q(s, a)$ lookup table) is in the order of 10^{14}. Note that in the case of *independent* learning agents, the number of junctions is *independent* of the exponential explosion of the state-action space *per junction*, i.e., the complexity scale *linearly* in the number of junctions. However, there is still a state-action space problem for large traffic networks.

The RL traffic signal control approach proposed by Thorpe and Anderson [11] is based on a *junction-based* state-space representation which represents all possible traffic configurations around a junction. In particular, each junction learns a Q-value that maps all possible traffic configurations to total waiting times of all vehicles around the junction. As mentioned in [15], this representation quickly leads to a very large state-space, because there are many possible configurations of vehicles waiting in the ingoing lanes of any junction. In [31, 32], the authors proposed Q-learning algorithms for traffic signal control with explicit coordination mechanisms among neighboring junctions. However, both works are based on *junction-based* state-space representation which consumes large space as discussed earlier. In addition, the latter work [32] uses the max-plus algorithm which is computationally demanding.

In our work, we adopted a different approach that is a *vehicle-based* state-space representation [10]. In this representation, the number of states will grow linearly in the number of lanes and vehicles positions and thus will scale well for large networks. The traffic signal decision is made by combining the estimated gain (e.g., waiting time) of all vehicles around a junction. Note that each vehicle does not have to represent its estimated gain itself (this can be done by the traffic junction) but the representation is *vehicle-based.*

2.5.2 Model-based vs. Model-free RL Traffic Signal Controllers

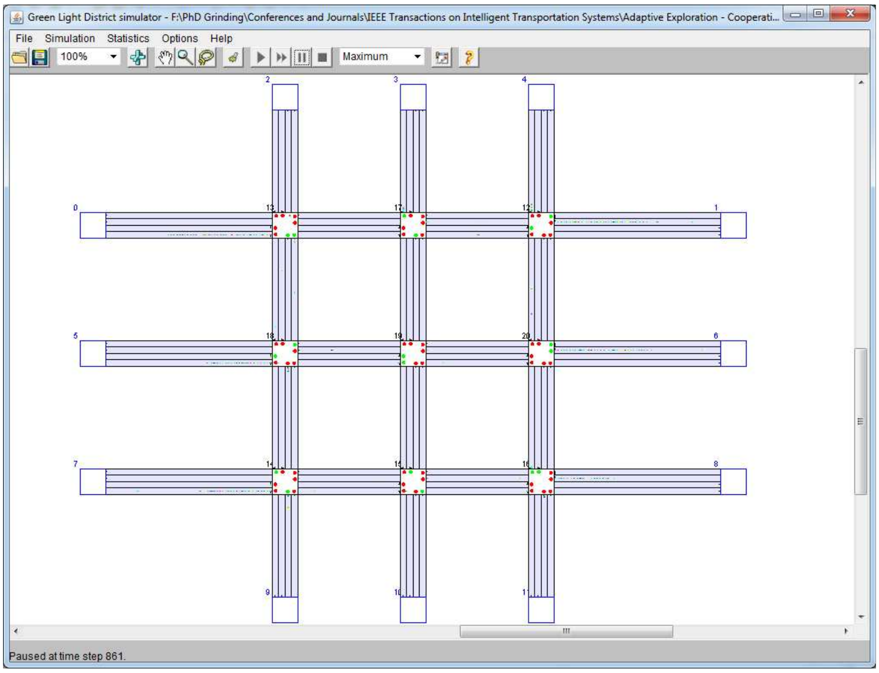
There are two categories for learning an optimal policy [33]: *model-based* methods or DP methods (as the one adopted in this thesis) and *model-free* methods, e.g., Monte Carlo methods, Temporal Difference Learning (e.g., SARSA [11], Q-learning [12]).

In *model-free* RL methods, the learning process is not guided by a state transition probability model. Although less computations per traffic signal decision is required by *model-free* RL methods relative to *model-based* ones, the convergence time is much smaller in *model-based* RL methods because the learning process is guided by a state transition probability model. The *model-free* RL methods may be more convenient in some domains, e.g., robotics applications where the computation and power capabilities of robots may be limited, while the number of iterations required for reaching the optimal policy is not demanding in the applications lacking real-time decision making, e.g., mine sweeping using robots. In *model-based* RL methods, e.g., the model presented by Wiering in [10], the overhead of the state transition probability is minimized by using a simple lookup table that maintains *few possible next states* at each time step where the vehicle can either: stay at the same state, move ahead in the current lane, or cross the junction.

In this thesis, we adopt the version of *model-based* RL proposed in [10] which proves its effectiveness when being applied to large scale traffic networks. In particular, we learn the state transition probability model using the Bayesian probability interpretation. Our proposed model is *continuously adaptive*, assuming that Prior (n+1) = Posterior (n), as will be shown in details in Chapter 5. Hence, we find that *model-based* RL methods (e.g., value iteration) are more convenient for traffic signal control in which investing more computations per traffic signal decision is not a demanding issue (considering the computation capabilities of junction agents) while reaching faster to the optimal learned values of traffic signal configurations is demanding in real-time traffic signal control.

2.6 Wiering-Based Traffic Signal Controllers

For testing and experimentation of our traffic signal control framework, we use the GLD traffic simulator [14], see Figure 2.1. The GLD simulator was initially based on a very simple discrete-time discrete-space model of traffic dynamics.


Green Light District simulator - F:\PhD Grinding\Conferences and Journals\IEEE Transactions on Intelligent Transportation Systems\Adaptive Exploration - Cooperati...
File
Simulation
Statistics
Options
Help
100%
Maximum
2
3
4
0
13
17
12
1
5
18
19
20
6
7
14
15
16
8
9
10
11

Paused at time step 861.

Figure 2.1: GLD vehicle traffic simulator: traffic network with 12 edge nodes and 9 traffic signal nodes.

Three previous extensions to the GLD traffic simulator have been implemented with simple acceleration models. The first extension is due to Cools *et al.* [50] that proposes a simple rule-based acceleration model based on the distance to the front vehicle. The second extension is due to Schouten and Steingröver [51] that allows the vehicles to change their speeds following either a Uniform or a Gaussian distribution. The third extension is due to Kuyer *et al.* [16] who implement the same technique of Gaussian distribution while using different values of speed thresholds. All the three extensions are inherently discrete with respect to both the time and space domains.

An important concern in any traffic simulator is the generation of populations of vehicles at different parts of the traffic network (i.e., simulating the traffic demand). Two extensions have been added to the GLD in this context. Escobar *et al.* [52] assume fixed generation frequency over extended periods of time, the generation frequency can be changed over non-overlapping intervals, the schedule of such change is specified in an XML file. Steingröver *et al.* [15] implement the same technique through a screen graphical interface.

Two extensions have been added to the GLD to achieve traffic green waves. The first is implemented by Escobar *et al.* [52]. This work proposes a very simple rule-based method for implementing green waves which depends on successive green signals over consecutive junctions with offsets. These offsets are determined based on the average vehicles speeds between the junctions. This is implemented over fixed periods of time. Since only the two opposite directions of the main road can have green waves simultaneously, traffic in the side roads will be delayed even when the traffic flow on the main road is very low. The second extension was implemented by Cools *et al.* [50]. They propose a more robust rule-based technique for implementing green waves. The integrity of a platoon of vehicles is achieved by preventing the tail of the platoon from being cut (when switching the traffic signal), while allowing the division of long platoons (in case there is a demand on the intersecting lanes) in order to prevent platoons from growing too much.

Our traffic signal control framework handles the drawbacks of the previously mentioned extensions to the GLD: acceleration model, traffic demand simulation, green wave implementation, etc. as will be shown in the proposed framework Chapter 4.

2.7 Multi-Objective Based Traffic Signal Controllers

To the best of our knowledge, few learning-based approaches are existing for multi-objective urban traffic signal control (e.g., [38]). On the one hand, the majority of these methods are based on either neuro-fuzzy or Multi-Objective Genetic Algorithm (MOGA). However, as mentioned in [53], the use of FL is not sufficient to represent the real-time traffic uncertainties. Also, ANNs and GAs require many computations and their parameters are difficult to be determined.

On the other hand, some traffic signal controllers that are *junction-based* (e.g., [12]) implement RL models in which the reward is a function in both the total delay and the queue length. However, as mentioned previously, *junction-based* methods suffer from exponential state-space. In [10], Wiering proposes two controllers called TC-2 and TC-3. The number of vehicles waiting in the queue at the next traffic signal is considered in the Q-function. The state representation is the same as in TC-1 (the original model of Wiering). However, as mentioned in [15], the proposed Q-function leads to an unusual adaptation of the real-time dynamic programming update in Equation 3.3. In addition, the $Q(s, a)$'s usually will not converge but instead keep oscillating between different values.

Houli *et al.* [44] present a multi-objective RL traffic signal control model. However, the traffic adaptation is done offline by activating one objective function at a time according to the current number of vehicles entering the network per minute. Steingröver *et al.* [15] present two traffic signal controllers, namely State Bit for Congestion (SBC) and Gain Adapted by Congestion (GAC). Traffic junctions take into account congestion information from neighboring junctions. This extension allows the agents to learn different state transition probabilities and value functions when the outgoing lanes are congested (i.e., optimizes the flow rate while optimizing the primary objective; trip waiting time). However, adding a new bit to indicate the degree of congestion in the next lane increases the state-space and slows the learning process. On contrary, in our model, the state-space representation is the same in size as the underlying traffic signal controller [10].

GAC [15] does not learn anything permanent about congestion, also this approach can not be easily generalized. In addition, the weight of the Q-value (i.e., congestion degree in the next lane) is neither a part of the state definition nor a function in the possible next states. On contrary, in our model, we weight the reward of the flow rate objective and learn this weight by defining it as a function in the possible next states as will be shown in Chapter 6. In the next Chapter 3, we present the underlying traffic signal control and simulation models for our multi-objective traffic signal control framework.

Chapter 3

Background

3.1 Introduction

In this chapter, we present a background on the multi-agent reinforcement learning (MARL) model. Afterwards, we discuss the Wiering MARL model for traffic signal control [10]. Finally, we present the GLD vehicle traffic simulator model [14] that is the testbed used to evaluate the proposed traffic signal control framework.

3.2 Multi-Agent Reinforcement Learning

The objective of RL is for the agent to learn how to take actions in a dynamic environment based on the past experiences of interactions between the agent and its surrounding environment. This learning process is done by maximizing some payoff function or minimizing some cost function [54]. The learning occurs iteratively and is performed through a trial-and-error process where the agent receives some reinforcement signals (rewards or penalties) as a result of the actions taken by the agent based on the current perception of the environment. The agent can be either a software program or a robot that can perceive its environment through sensors and act upon it through actuators.

In RL, the agent is assumed to be *rational* as it will always choose an action which optimizes some performance measure given what it knows so far [33]. In addition, the agent is *autonomous* as it chooses its actions according to what it has observed and learned so far from its own experience [33].

A performance speed-up can be achieved by the distributed computation of a MARL model (where the agents can exploit the decentralized structure of a task) [55]. Another advantage of a MAS formulation is that when one or more agents fail, the remaining agents can take over some of their tasks; this implies that MARL is inherently *robust* [55]. Furthermore, the design of most MASs allows an easy insertion of new agents into the system leading to a high degree of *scalability* [55].

One important challenge in a MARL-based system is the *curse of dimensionality* that is caused by the *exponential* growth of the size of the joint state-action space [55]. Matarić [56] classifies the many challenges of MARL in dynamic environments into two main categories: (1) managing the complexity in the size of the state-space, and (2) structuring and assigning the reinforcement signals.

One more important point is that agents may be of *self-interest* or *fully cooperative.* In the former, agents act to achieve their individual goals and have no need to cooperate in order to achieve a common goal [33]. In cooperative MAS, several agents attempt, through their interaction, to jointly solve tasks or to maximize a common utility [57]. Kuyer *et al.* [33] focus on *fully cooperative* traffic control agents that act to achieve a common goal (minimize the waiting time of traveling vehicles in a traffic network). In such a system, the payoff function (the reward) is global (shared by all agents). The optimal individual action based on individual payoff may not be the optimal one for a group of agents. The agents coordinate their actions in order to find the optimal joint action [33]. For more details about the RL, the reader is referred to Appendix A.

3.3 Wiering RL Traffic Signal Control Model

We adopt the RL model developed by Wiering [10] for traffic signal control. Each junction is controlled by an active [1] intelligent agent that learns a policy for signal splitting through a guided trial-and-error life interaction process with the environment to online optimizing some criteria (e.g., minimizing the waiting time of vehicles). This approach is *vehicle-based*, that is, the state of the system is local and microscopic.

In Wiering's approach, the state of the vehicle at a particular junction consists of the following pieces of information:

1. The *traffic light* of the lane in which the vehicle is moving or waiting, denoted tl.

2. The *position* in which the vehicle is currently at, denoted p.

3. The *destination* towards which the vehicle is traveling, denoted des.

Thus, the vehicle current and next states can be denoted by $s = [tl, p, des]$ and $s' = [tl', p']$, respectively, where the vehicle final destination does not change by the state transition. In a real-world application, drivers/vehicles can send the information required by the junction controller agent (i.e., position and destination) for the junction to estimate the vehicle gain from the traffic signal decision. This can be achieved by using some kind of sensors (e.g., sensors in smart phones) through a Vehicle-to-Infrastructure (V2I) communication protocol.

This approach is essentially a *model-based value-iteration* technique where the *state transition probability* is *continually* estimated to guide the learning and optimization process. The state transition probability is represented by a lookup table $\Pr(s, a, s')$ where a is the action of the traffic signal (i.e., red or green) that causes the vehicle to move from state s to the next state s'.

[1] Despite we consider the junction as the *active* agent and the vehicle as the *passive* agent, our model is still *vehicle-based* not junction-based as the state definition is on the vehicle level.

Recall that there are *few possible next states* at each time step where the vehicle can either stay at the same state, move ahead in the current lane, or cross the current junction. These probabilities are estimated based on the *frequentist* interpretation of probability:

$$\Pr(s, a, s') = \frac{C(s, a, s')}{C(s, a)}, \tag{3.1}$$

where $C(s, a, s')$ counts the number of transitions (s, a, s') and $C(s, a)$ counts the number of times a vehicle was in state s and action a was taken.

$\Pr(a|s)$ represents the probability that the traffic signal *tl* is red or green given that a vehicle is at state s. $C(s)$ counts the number of times a vehicle was in state s. Thus, the probability $\Pr(a|s)$ is given by:

$$\Pr(a|s) = \frac{C(s, a)}{C(s)}. \tag{3.2}$$

We use the *Bayesian* probability interpretation to estimate the parameters of these probabilities. This estimation was found to be more stable, robust, and *continuously adaptive* to the changing environment dynamics. We discuss this approach in Chapter 5. Note that, in case we want to transfer from the simulation environment to a real case, we can make use from the learned *state transition probabilities* of the possible next states (considering the same network structure and traffic dynamics).

The original model [10] optimizes the cumulative waiting time of all vehicles till arriving at their destinations. Thus, the Q-function represents the estimated waiting time for a vehicle at state s until it arrives to its destination in case the action of the current traffic signal is a and is given by:

$$Q(s, a) = \sum_{s'} \Pr(s, a, s')(R(s, a, s') + \gamma V(s')), \tag{3.3}$$

where γ is a discount factor ($0 < \gamma < 1$) that discounts the influence of the previously learned V-values and ensures that the Q-values are bounded.

The reward function $R(s, a, s')$ is the immediate scalar reward. In the single objective controller proposed in the original work [10], $R(s, a, s') = 1$ in case the vehicle waits at the same position, otherwise equals zero. We propose a more elaborate design for the reward function that is well-suited for a multi-objective traffic signal control framework. The proposed multi-objective reward function is discussed in Chapter 6.

The V-function represents the estimated average waiting time for a vehicle at state s till leaving the traffic network regardless the current traffic signal action and is given by:

$$V(s) = \sum_{a} \Pr(a|s) Q(s, a). \tag{3.4}$$

The controller at each junction sums up the gains $Q(s, red) - Q(s, green)$ of all vehicles waiting at the current junction and chooses the traffic signal configuration (consistent green lights on all directions of the junction) with the maximum cumulative gain. In the proposed multi-objective traffic signal control framework, we adopt the same gain definition of vehicles.

The possible traffic signal configurations (i.e., possible phases) represent the consistent green lights on all directions of the junction that do not cause any possible accidents between the crossing vehicles. Consider a junction controlling the traffic between 4 intersecting roads. Each road consists of 4 lanes, in which the ingoing lanes per each road are one lane for turning left and one lane for going straight or turning right. According to this setting, there exist 8 possible traffic signal configurations [1] (4 possible configurations for the traffic signals of each road to be green for left and straight/right directions and 4 possible configurations for the traffic signals of each opposite roads to be green for left and straight/right directions).

[1] Note that the 8 possible traffic signal configurations per junction in the adopted model differ from the number of phases at an ordinary traffic signal, i.e., green-amber-red.

For a fixed time controller, all possible phases should at least be green once within a cycle. In our multi-objective framework, we do not estimate the optimal phase length, but rather, at each time step the junction agent chooses (based on the current traffic situation) either to extend the current phase or to begin another possible traffic signal configuration. In addition, the decision is based on all vehicles in the lane, i.e., not only the vehicles queued at the traffic signals, this setting is much consistent with the nature of the multi-objective function, i.e., formulation and evaluation of some objectives, e.g., average trip time, average vehicles speed, etc.

3.4 GLD Traffic Signal Simulation Model

In order to examine the proposed traffic signal control framework, some experimentation platform is needed, that is a *traffic simulator*. The widely used microscopic traffic flow simulation programs (e.g., VISSIM, Paramics, etc.) do not give the researcher an easy free way to create novel traffic signal controllers. In our work, we chose to extend the moreVTS vehicle traffic simulator [50] that is based on the GLD traffic signal simulation platform [14]. This is due to the following reasons:

1. The GLD is a *widely used* open source traffic simulator, e.g., used by [15,16,49,50].

2. The ability to compare the proposed traffic signal controller with other major traffic signal controllers implemented over the GLD.

3. Collecting statistics from a set of performance indices (MOEs) that are already available in the GLD with the ability to add new performance indices.

4. The visual ability to edit/create traffic networks and schedule traffic demands through graphical interface, see Figure 2.1.

The GLD is a microscopic traffic signal simulation model such that it models the behavior of the individual vehicles. The infrastructure mainly consists of *roads* and *nodes*.

Any *road* connects *two* nodes and can have *several* lanes in each direction. The length of any road is expressed in *cells*. A *node* can be either a *junction* where traffic signals are acting or an *edge node*. Two types of agents occupy this infrastructure: *vehicles* and *traffic signals controllers*. Agents act *autonomously* and are updated every time step.

Vehicles enter the traffic network at *edge nodes*. Each *edge node* has a probability to generate a vehicle at every time step $\in [0, 1]$ (i.e., 1 means a vehicle is generated every time step from the edge node, while 0 means no vehicle will be generated). A destination from the other edge nodes is assigned to each generated vehicle. The GLD user can adjust the distribution of destinations for each edge node.

3.5 Conclusion

The characteristics of a MAS are similar in nature to the traffic signal control problem. In particular, the formulation of the traffic signal control problem as a MARL model is very promising. In this thesis, we adopt a MAS framework to comply with the distributed nature and complexity of the problem in a cooperation-based configuration (as will be shown in Chapter 5).

In this chapter, we discuss the adopted traffic signal control model [10]. We use the Bayesian probability interpretation to estimate the unknown parameters of the MDP. We discuss in Chapter 5 how this estimation is more stable, robust, and adaptive to the changing environment dynamics. We propose a more elaborate design for the reward function in Chapter 6 which is well-suited for a multi-objective traffic signal control when being compared to the underlying single objective control (which optimizes only the trip waiting time). In this chapter, we discuss as well the GLD vehicle traffic simulation model. Despite of its mentioned capabilities, the GLD needs some contributions in order to give higher confidence in the validity of the proposed traffic signal control framework. Our contributions to the GLD are discussed in the next Chapter 4.

Chapter 4

Proposed Framework

4.1 Introduction

In this chapter, we present the proposed framework that is implemented on the top of the traffic signal control and simulation models discussed in the previous Chapter 3. This includes moving towards a continuous-time/continuous-space simulation platform, applying the IDM acceleration model on the RL traffic signal control model, simulating the impact of the adverse weather conditions on traffic flow and traffic demand, adding the vehicle generation probability distributions that allow for variability and non-stationarity, introducing the proposed exploration schema, and adding new performance indices.

4.2 Continuous-Time and Continuous-Space Simulation Platform

The GLD is a discrete-time discrete-space simulation platform that is based on *cellular automata* in which each road is represented by *discrete* cells. A road cell can be occupied by a vehicle or can be empty. We implemented the more realistic IDM acceleration model [13] that is used to simulate, in continuous-time and continuous-space, the acceleration and deceleration of vehicles. The vehicle acceleration dv/dt depends on:

- The *current velocity* v [1].
- The *distance* to the front vehicle s.
- The *difference in velocity* Δv that is positive when approaching the front vehicle.

Thus, the *acceleration* is given by:

$$\begin{aligned} \frac{\mathrm{d}v}{\mathrm{d}t} &= a\Big[1 - \Big(\frac{v}{v_0}\Big)^{\delta} - \Big(\frac{s^*}{s}\Big)^2\Big], \\ s^* &= s_0 + min\Big[0, \Big(vT + \frac{v\Delta v}{2\sqrt{ab}}\Big)\Big]. \end{aligned} \tag{4.1}$$

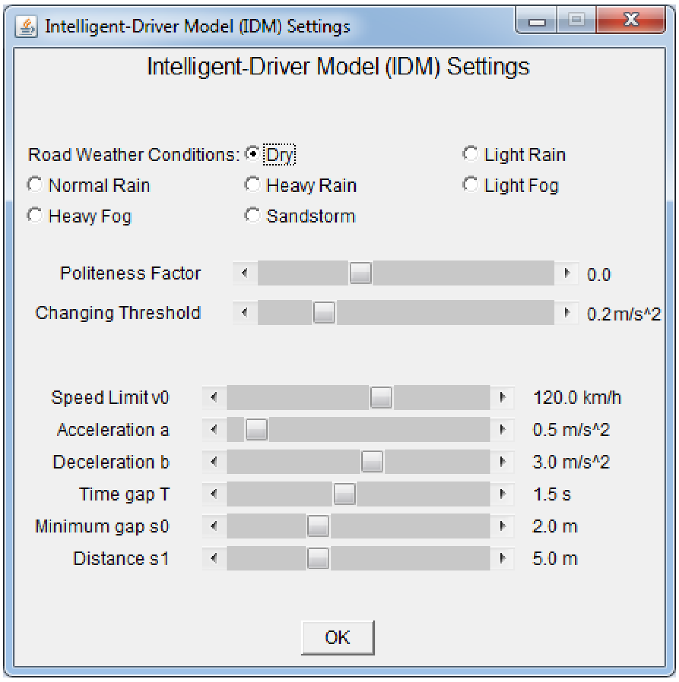
The acceleration model consists of two terms: the *desired acceleration* when the road is *free* $a[1-(\frac{v}{v_0})^{\delta}]$, and the *braking deceleration* when there is a *front vehicle* $-a[(\frac{s^*}{s})^2]$. The parameters of the IDM driver model are [13]:

1. The *desired velocity* on a free road, v_0.
2. The *desired safety time headway*, T.
3. The *acceleration* in usual traffic, a.
4. The *braking deceleration* in usual traffic, b.
5. The *minimum bumper-to-bumper distance*, s_0.
6. The *acceleration exponent*, δ.

We adopt the default values of those parameters and the range of each parameter to be as the traffic simulator available at http://www.traffic-simulation.de which applies the IDM

[1] In the rest of the thesis, we refer to the vehicle absolute velocity by the *vehicle speed* which is always positive.

acceleration model. Figure 4.1 represents the place where the GLD user can change the values of those parameters.


Intelligent-Driver Model (IDM) Settings

Intelligent-Driver Model (IDM) Settings

Road Weather Conditions: ● Dry ○ Light Rain
○ Normal Rain ○ Heavy Rain ○ Light Fog
○ Heavy Fog ○ Sandstorm

Politeness Factor 0.0
Changing Threshold 0.2m/s^2

Speed Limit v0 120.0 km/h
Acceleration a 0.5 m/s^2
Deceleration b 3.0 m/s^2
Time gap T 1.5 s
Minimum gap s0 2.0 m
Distance s1 5.0 m

Figure 4.1: Intelligent Driver Model (IDM) settings in the GLD.

Accordingly, there are 3 clocks in our traffic signal control framework that need to be synchronized: (1) the IDM modeler time, (2) the traffic signal controller time, and (3) the GLD simulator time. The 3 clocks are synchronized every δt as follows. First, the IDM modeler updates the state of all vehicles in the entire traffic network where the new positions are calculated as follows:

$$\begin{aligned} \text{speed}_{\text{new}} &= \text{speed}_{\text{old}} + \text{acceleration}_{\text{IDM}} \times \delta t, \\ \text{position}_{\text{new}} &= \text{position}_{\text{old}} - \text{speed}_{\text{new}} \times \delta t. \end{aligned} \tag{4.2}$$

Note that in the GLD, the vehicles positions values are decreasing as vehicles move from its source nodes towards the junctions. This clarifies the negative sign in the position update in Equation 4.2. Afterwards, the simulator gathers all the needed statistics from the traffic network such as the average waiting time, the average queue length, etc. The controller updates the state transition of each vehicle and recalculates the $Q(s, a)$'s and $V(s)$'s. Then the simulator updates the traffic network screen visualization. Afterward, the traffic signal controllers decide on the new actions at all junctions of the network by calculating how every traffic signal should be switched. The new traffic signal configurations are applied by switching the traffic signals to their appropriate values. Finally, the simulator schedules the next state for the next time step (e.g., new vehicles join the network following the scheduled traffic demand).

4.3 IDM Impact on the RL Traffic Signal Control Model

We analyzed and fixed some crucial problems that appeared in the original RL traffic signal control model [10], particulary when applying the IDM acceleration model. As a result of the control being still discrete in nature, many IDM state transitions (potentially infinite) correspond to one state transition with respect to the control (the controller perceives the lane as an extension of discrete cells whereas the IDM views it as a continuous stretched line - recall that the vehicle position is part of the controller state definition).

As a result, some ambiguity appears in the definition of the reward function $R(s, a, s')$. In particular, if the reward value is depending on the distance traveled by the vehicle, then there will be different immediate reward values for the same controller state transition. We solved this problem by averaging the reward values gained over time. For instance, if the obtained rewards in the time steps $t_1, t_2, \ldots, t_n$ due to state transitions from a state s to a next state s' are $r_1, r_2, \ldots, r_n$, respectively, then the immediate reward of the state transition from s to s' will change from $\sum_{i=1}^{n-1} r_i/(n-1)$ to $\sum_{i=1}^{n} r_i/n$.

Another issue is the sign oscillation problem (a *Zeno* phenomena) that results from the infinitesimally slow acceleration of back vehicles when the traffic signal is just turning green. In this case, the $Q(s, green)$'s of those *stationary* vehicles will increase that decreases the cumulative gain and accordingly forces the traffic signal to switch back to red (too early) before any vehicle can cross the junction. We solved this issue by giving those *stationary* vehicles some penalty smaller than the one given when the traffic signal is red, e.g. $R(s, a, s')$ for back stationary vehicles when the signal is green equals 0.3 instead of one [1].

4.4 Simulating Adverse Weather Conditions Impact on Traffic Flow

Adverse weather conditions have a *substantial impact* on traffic speed, capacity, and flow rate [58–60]. However, the adaptation effects in the actual longitudinal driving behavior models need more investigation and a data-driven mathematical model is needed (which is interesting for prospective investigation).

The results presented in [61] show that adverse weather conditions (specifically fog) led to a *decrease* in the *desired speed*, v_0. A *substantial increase* is observed in both of the *distance* to the front vehicle, s_0, and the *minimum headway*, T. The *maximum acceleration*, a, and *deceleration*, b, *highly decrease* after the start of the adverse weather condition. We have implemented in the GLD the impacts of weather on the IDM acceleration model parameters. The simulated weather impacts on the speed parameters comply with the results presented in [61]. Moreover, the implemented weather conditions can represent those conditions in Egypt, specifically the weather of Alexandria city that include; light rain, normal rain, heavy rain, light fog, heavy fog, and sandstorm with 5%, 10%, 12%, 25%, 30%, 36% speed reduction relative to the dry weather, respectively.

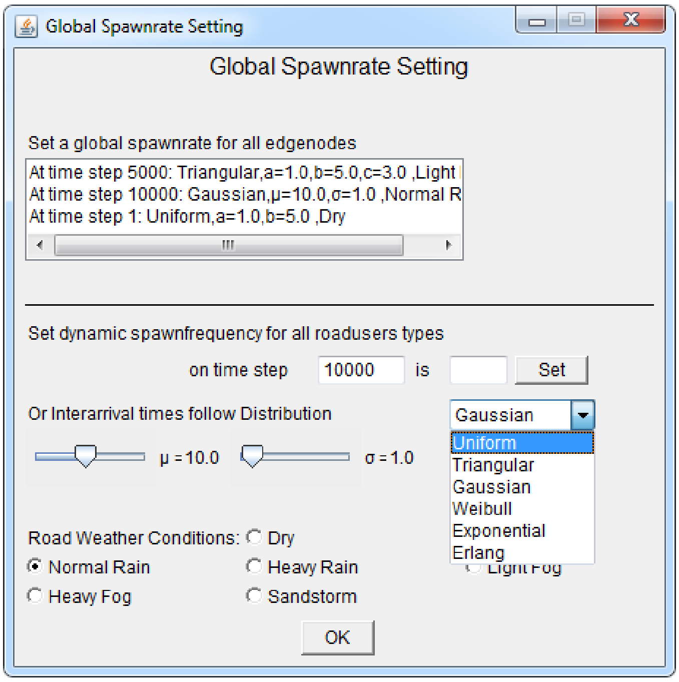
In [59], Cools *et al* study the impact of weather conditions on traffic intensity. This

[1] Note that all the reward values are then scaled (multiplied by 10) for better discrimination between the reward values in case the traffic signal is red or green.

study showed that snowfall, rainfall, and wind speed diminish traffic intensity while high temperatures increase traffic intensity. We simulate these two impacts by low and high vehicle generation frequencies, respectively.

4.5 Traffic Demand Probability Distributions

The traffic demand in the GLD traffic simulation model [14] is implemented by generating a uniform random number every simulation time step and checking its value against a fixed traffic demand rate ∈ $[0, 1]$. In order to allow for variability and non-stationarity, we have implemented in the GLD varying probability distributions of the inter-arrival times of input vehicles, see Figure 4.2.



Global Spawnrate Setting

Global Spawnrate Setting

Set a global spawnrate for all edgenodes

At time step 5000: Triangular,a=1.0,b=5.0,c=3.0 ,Light
At time step 10000: Gaussian,μ=10.0,σ=1.0 ,Normal R
At time step 1: Uniform,a=1.0,b=5.0 ,Dry

Set dynamic spawnfrequency for all roadusers types

on time step 10000 is

Or Interarrival times follow Distribution: Gaussian

μ = 10.0

σ = 1.0

Road Weather Conditions: ○ Dry ● Normal Rain ○ Heavy Rain ○ Heavy Fog ○ Sandstorm ○ Light Fog

Figure 4.2: Vehicle generation probability distributions in the GLD.

We categorize the vehicle generation distributions as following:

- *Fixed* (no inter-arrival probability distribution) or *Probabilistic* (not fixed; one of our contributions).

- *Static* (the same vehicle generation distribution for all time intervals) or *Dynamic* (not static; one of our contributions in the *probabilistic* case).

We check if the generated random number after the last arrival equals one time step or if the time step of the last arrival plus the random number generated after the last arrival equals the current time step, we do the following:

1. Generate new vehicle.
2. Set the time step of the last arrival to the current time step.
3. Generate a new random number following the specified inter-arrival distribution.

For modeling the vehicle inter-arrival probability distribution, we use the following continuous distributions: Uniform, Triangular, Exponential, Erlang, Weibull, and Gaussian. Here, we give some examples on the *dynamic* generation probability distributions:

First Example:

1. Initially the current time step equals 0.
2. The time step of the last arrival is the current time step minus 1, i.e., equals -1.
3. The last generated random number equals 1, i.e., the inter-arrival time between the two vehicles is 1 time step.

Thus, the time step of the last arrival plus the last generated random number equals 0 (that is the current time step). Then, a vehicle is generated at time step 0 and the time step of the last arrival is the current time step.

Second Example:

1. The current time step equals 130.

2. The time step of the last arrival equals 120, and the generation frequencies are defined as following:

   - From time step 120 - To time step 129: the generation frequency is fixed to 0.4.
   - From time step 130 - To time step 140: the inter-arrival distribution is $\mathcal{U}(a = 0, b = 1)$, and the generated random number equals 1.

Then, a vehicle is generated at time step 130.

Third Example:

1. The current time step equals 120.
2. The time step of the last arrival equals 100, and the generation frequencies are defined as following:

   - From time step 100 - To time step 115: the inter-arrival distribution is $\mathcal{N}(\mu = 20, \sigma = 0.5)$, and the generated random number equals 20.
   - From time step 115 - To time step 120: the inter-arrival distribution is $Weibull(k = 1, \lambda = 1)$, and the generated random number equals 3.

Then, two vehicles are generated at time steps 120 and 123. Following the Poisson process, the inter-arrival time can not be equal 0, i.e., no two vehicles can be generated in the same time step on the same lane.

4.6 Exploration Policy

In the underlying traffic signal control model [10], a random traffic signal configuration can be chosen with a small probability $\epsilon = 0.01$ for the exploration of the state-action space. In [62], we also used ϵ-exploration, though we found that it is better to start initially with high exploration rate (where there is still no much knowledge about the *optimal gain values* to be exploited) and decrease the exploration rate *gradually* in time; the exploration rate was given by:

$$\epsilon_t = exp(-t/k_t), \tag{4.3}$$

where t is the current simulation time step and k_t is the Boltzmann temperature factor that decays by time till being fixed at the value of 1. k_t is used to increase the exploration effect initially where all traffic signal configurations will have approximately the same probability to be green. k_t decreases *gradually* where all traffic signal configurations will be selected according to their cumulative gain (i.e., exploitation of the learned values) after $t \simeq 400$ time steps. We chose to start at $k_t = 100$ and then decrease by 1 every 10 time steps until reaching $k_t = 1$ (similar to the rate proposed in [63]).

We propose a novel hybrid exploration technique that uses softmax exploration to better respond to transient periods (e.g., due to congestion at rush hours). This exploration technique is discussed in details in Chapter 5.

4.7 Fixing the Next States Definition in the GLD

The implementation of the underlying traffic signal control model loops on all the *possible* next states s' according to the free positions ahead of a vehicle at state s in the *current* time step. Particularly, this implementation assumes the next states by discretizing the free distance between the vehicle and the front one. Thus, the sum of the transition probabilities of these next states is not a must equal to 1 because the probability should be calculated and updated based on the *actually experienced* next states.

Hence, this implementation is improper and we instead loop on all the next states that are *actually experienced* (e.g., by other vehicles) starting from the same state s. The sum of these state transition probabilities equals 1. The main aim of this update is the correction of the model implementation in calculating $Q(s, a)$ (i.e., not a must to enhance the results of the various performance indices).

4.8 New Performance Indices

The main performance measure in the GLD depends on the *average delay* of the vehicles. The *junction delay* of a vehicle is calculated as follows:

$$\begin{aligned} \text{Junction Delay} = &\ (\text{Time Step the Vehicle Crosses the Junction} \\ &- \text{Time Step the Vehicle Joins the Junction Lane}) \\ &- (\text{Lane Length/Lane Maximum Speed}). \end{aligned} \tag{4.4}$$

In our work, we define the proper *junction waiting time* of a vehicle as follows:

$$\begin{aligned} \text{Junction Waiting Time} = &\ \text{Time Step the Vehicle Crosses the Junction} \\ &- \text{Time Step the Vehicle Joins the Junction Waiting Queue}, \end{aligned} \tag{4.5}$$

where joining the junction waiting queue is counted once the vehicle speed drops beyond a specific threshold, 0.36 km/h [1].

In order to examine the performance of the proposed multi-objective controller, we need more elaborate performance indices than those originally implemented in the GLD. Some performance indices in the GLD are inefficient. The original *average trip waiting time* proved to be insufficient because all vehicles not arrived yet to their destinations (for any reason, e.g., due to congested traffic) are not incorporated in the statistics.

We include all vehicles even those that have not yet arrived to their destinations by adding for those vehicles the *expected trip waiting time* $V(s)$ to the total waiting time

[1] In the traffic simulator available at www.traffic-simulation.de which applies the IDM acceleration model, the minimum value of the desired velocity v_0 in the "traffic light" scenario is 1 km/h. Thus, we set the *stop speed* to be lower than half this value (to be equal to 0.36 km/h).

they have experienced so far. The *total waiting time* that a vehicle has experienced equals the *summation* of the waiting times at the junctions that the vehicle has already crossed in Equation 4.5. We call this policy the *co-learning* technique for calculating the performance indices.

- Given the set of entered vehicles is V_{entered}.
- The set of vehicles that entered but have not arrived yet is $V_{\text{notArrived}}$.
- The set of nodes (junctions or edge nodes) crossed by the vehicle v is N_{crossed}.
- The time step at which the vehicle v crosses the node n is t_{cross}.
- The time step at which the vehicle v joins the waiting queue of the node n is t_{joinQn}.
- The current time step is t_{current}.
- The time step at which the vehicle v joins the waiting queue of the current node is $t_{\text{joinQcurrent}}$.
- The expected trip waiting time of the vehicle $v \in V_{\text{notArrived}}$ is $V(s_{\text{current}})$.
- The number of vehicles entered so far is N_{entered}.

Then, the $ATWT_{\text{colearn}}$ is given by:

$$\begin{aligned} ATWT_{\text{colearn}} = & \Bigg[\sum_{v \in V_{\text{entered}}} \sum_{n \in N_{\text{crossed}}} (t_{\text{cross}} - t_{\text{joinQn}}) + \sum_{v \in V_{\text{notArrived}}} \Big((t_{\text{current}} - t_{\text{joinQcurrent}}) \\ & + V(s_{\text{current}}) \Big) \Bigg] / N_{\text{entered}}. \end{aligned} \tag{4.6}$$

For the $ATWT_{\text{colearn}}$, the reward function is $R(s, a, s') = 1$ if the vehicle waits at its position, otherwise, $R(s, a, s') = 0$. If the vehicle waits at the current position, i.e., $\Delta p \simeq 0$ (that leads to higher ATWT), then it will be penalized by the reward value.

Moreover, we have also implemented the ATT_{colearn} which is given by:

- Given the set of arrived vehicles is V_{arrived}.
- The time step at which the vehicle v arrives its destination is t_{arrive}.
- The time step at which the vehicle v starts its trip is t_{start}.
- The expected trip time of the vehicle $v \in V_{\text{notArrived}}$ is $V(s_{\text{current}})$.

Then, the ATT_{colearn} is given by:

$$ATT_{\text{colearn}} = \left[\sum_{v \in V_{\text{arrived}}} (t_{\text{arrive}} - t_{\text{start}}) + \sum_{v \in V_{\text{notArrived}}} \Big((t_{\text{current}} - t_{\text{start}}) + V(s_{\text{current}}) \Big) \right] / N_{\text{entered}}. \tag{4.7}$$

For the ATT_{colearn}, the reward function is $R(s, a, s') = 1$ (such that every time step the vehicle trip time increases by one). Despite we have implemented as well the $AJWT_{\text{colearn}}$ performance index, it is not logically meaningful as the *co-learning* technique for calculating the performance indices is more convenient to the trip-based statistics (using the expected remaining value till the end of the trip).

In the original GLD, the vehicles waiting in edge nodes (due to overfull ingoing lanes) do not enter the traffic network and consequently are not incorporated in many performance measures (e.g., ATWT, ATT, etc.). We solved this problem by rejecting the vehicles that are queued in edge nodes and use the *percentage of rejected vehicles* as a more reasonable performance index. Moreover, we added the *relative throughput* performance index in the GLD. This performance index equals the total number of arrived vehicles divided by the total number of entered vehicles. In addition, we added the *average speed* performance index. This performance index equals the total distance traveled by all vehicles (either have arrived or have not arrived yet) divided by the total time spent in the network.

In order to evaluate the performance of the *green wave* objective, we added the *average number of trip absolute stops* performance index. Once the vehicle joins the waiting queue (i.e., its speed drops beyond 0.36 km/h, as mentioned earlier), we count 1 vehicle stop, and once the vehicle joins the next waiting queue after crossing the current junction, this count will be 2 vehicle stops.

Since the vehicle stops increase the vehicle emission and oil consumption (as mentioned in [44]), we added the *average number of vehicles trip stops* performance index to evaluate the performance of the fuel consumption objective. This performance index equals the sum of all vehicles stops in the whole trip divided by the number of arrived vehicles. We only depend on the *number of vehicles stops* to evaluate the *fuel consumption* objective. However, a more advanced evaluation of the *fuel consumption* should be a weighted function of the *number of vehicles stops*, maximum and minimum change in vehicles speeds, etc. This can be done as a future work.

The performance indices presented in this section represent the *measures of effectiveness* (MOEs) used in evaluating the performance/effectiveness of the proposed traffic signal control framework relative to other controllers (as will be shown in Chapter 6).

4.9 Discussion

As mentioned in this chapter, the three clocks of the modeler, controller, and simulator are synchronized every δt. In our work, we set δt=0.25 second. Both of the traffic simulators available at www.traffic-simulation.de which applies the IDM acceleration model and the Paramics traffic simulator use the same value of δt=0.25 second. A dissociation between the three timers is recommended, e.g., increasing the δt of the controller for decreasing the overhead of frequent traffic signal decision. On contrary, we can decrease the δt of the simulator for smooth and realistic traffic motion.

One proposed solution to the Zeno phenomena[1] is using a *"minimum green time"*. The traffic signal controller can ignore the first transient period and check the traffic status after the stationary vehicles start to move (become non-stationary). However, this may weaken the power of the RL that should decide according to the learned knowledge (i.e., without using a *"minimum green time"*) to extend the current traffic signal configuration adaptively. Another proposed solution to this phenomena is increasing the traffic signal controller decision clock, i.e., δt=0.25 second. This will allow the controller to take its decision after the stationary vehicles move. Moreover, it may be rather better that the vehicles, *in general*, inform the controller with their new locations after they move. This will remove the controller overhead of checking every small period of time (e.g., δt=0.25 second) whether all vehicles have moved or not.

4.10 Conclusion

In this chapter, we present the proposed framework that is implemented on the top of the underlying traffic signal control and simulation models. In addition, we show that using RL for solving control tasks optimization in continuous-space (specifically in the traffic signal control domain) has some challenges that affect the reward design of the model.

Moreover, in this chapter, we introduce how the *exploration technique* used by the RL traffic signal controller can lead to better response to transient periods (e.g., due to congestion at rush hours) which is discussed in details in the next Chapter 5. Finally, we show that the RL traffic signal controller can make use from the *learned knowledge* for better evaluating the performance of the traffic network.

[1] Recall that the Zeno phenomena occurs in the underlying RL traffic signal control model when applying the continuous-time/continuous-space IDM acceleration model.

Chapter 5

Handling Traffic Network Non-Stationarity

5.1 Introduction

In this chapter, we propose the Bayesian probability interpretation for estimating the unknown parameters of the probabilities of the MDP. This estimation allows the traffic signal controller to make real time (online) adaptation in the sense that it responds effectively to the changing environment dynamics and non-stationarity of the road network. Moreover, we propose a novel cooperative hybrid exploration technique which is more adaptive to the changing dynamics in road conditions, i.e., improves the trip waiting time of vehicles during transient periods (e.g., due to congestion at rush hours).

5.2 Stationary vs. Non-Stationary Environments

In [33], Kuyer discusses the *stationary* versus the *non-stationary* environments. In a stationary environment, as an assumption the dynamics is always *fixed*. This means that the transition probability $\Pr(s'|s,a)$ is fixed such that the next state s' is either unique for a given action a and state s or there is some probability distribution over the possible next states which is fixed over time.

This assumption is violated in real-world systems where $\Pr(s'|s,a)$ for a given action a and state s changes over time due to the change in the environment dynamics. In the non-stationary environment, agents will need to learn the whole history even for environment dynamics which have been previously experienced since the policy that was computed is no longer valid when the dynamics change.

5.3 MDP Parameters Estimation Using Bayesian Probability Interpretation

In [10,14], Wiering et al. learn the transition probability functions $\Pr(s'|s,a)$ and $\Pr(a|s)$ by counting the number of vehicles facing the same traffic situation (s,a). Each next state s' has a bias probability $\Pr(s'|s,a)$ for a given action a and state s where the summation of those probabilities equals one. We implement the case where $\Pr(s'|s,a)$ changes over time due to the change in the environment dynamics (e.g., due to congestion at rush hours). In order to make the transition probabilities non-stationary (adaptive), we estimate the weights of the next states/actions using the Bayesian probability interpretation depending on the current road conditions. The bias probabilities of the possible next states s' can be simulated by changing the corresponding generation frequency, e.g., decreasing the traffic in some situation (one possible next state) while increasing the traffic in another situation (other possible next state).

In our approach, the current estimation becomes the *prior* for the next time step. This estimation is more stable and *more adaptable* to the changing environment dynamics. That is if a change occurs in the network dynamics (due to accidents, rush hours, etc.) the controller using this probability estimation can handle the traffic efficiently by the way that optimizes the various performance indices (e.g., waiting time, queue lengths).

The idea behind this state transition probability estimation is based upon the simple Bayes' rule: Let A and B be two events, then the posterior density of A given B has the following formula:

$$\Pr(A|B) = \Pr(B|A)\Pr(A)/\Pr(B). \tag{5.1}$$

Let P be a random variable representing an *estimator* of some unknown parameter. In the proposed traffic signal control framework, such a parameter can be either:

1. One of the parameters of $\Pr(a|s)$ which is the posterior probability of taking action a given state s, or

2. One of the parameters of $\Pr(s'|s,a)$ which is the transition probability of being in the next state s' given the state/action pair (s,a).

Following, we give an example for illustration. Fix some state s, then $\Pr(a|s)$ has one parameter P for the probability of $a = RED$. For every time index t, let $I_t = \{j \leq t\colon\ \textit{state s is occupied at time j}\}$. For every $n = |I_t| \in \mathbb{N}$, define the Bernoulli random variable X_n as follows:

$$X_n = \begin{cases} 1 & a = RED\ at\ time\ k = \max I_t, \\ 0 & o.w., \end{cases} \tag{5.2}$$

that is $\bar{X}_n$ is a sequence of Bernoulli random variables defined at the time indices where the state s is occupied by a vehicle. When X_{n+1} is defined, we estimate P by recursively applying the Bayesian inference rule as follows:

$$\text{Posterior}(n+1) = \frac{\text{Likelihood}(n+1)\text{Prior}(n+1)}{\text{Normalizing Factor}(n+1)}. \tag{5.3}$$

We take Prior$(n+1)$ = Posterior(n). Let $\bar{X}_{n+1} = (X_1, \ldots, X_{n+1})$. Then we have

$$\Pr(P_{n+1}|\bar{X}_{n+1}) = \frac{\Pr(\bar{X}_{n+1}|P_{n+1})\Pr(P_{n+1})}{\Pr(\bar{X}_{n+1})} = \eta \Pr(\bar{X}_{n+1}|P_{n+1})\Pr(P_{n+1}|\bar{X}_n), \tag{5.4}$$

where η is the normalization factor. Solving the above recursive equation with the assumption that $\bar{X}_{n+1}$ are independent random variables,

$$\begin{aligned}\Pr(P_{n+1}|\bar{X}_{n+1}) &= \alpha \prod_{i=1}^{n+1} \Pr(\bar{X}_i|P_{n+1})\Pr(P_{n+1}|\bar{X}_0); \Pr(P_{n+1}|\bar{X}_0) = 1 \\ &= \alpha \prod_{i=1}^{n+1}\prod_{j=1}^{i} \Pr(X_j|P_{n+1}) = \alpha \prod_{i=1}^{n+1}\prod_{j=1}^{i} P_{n+1}^{X_j}(1-P_{n+1})^{(1-X_j)}.\end{aligned} \tag{5.5}$$

For an easier differentiation, we find $\ln\Pr(P_{n+1}|\bar{X}_{n+1})$:

$$\begin{aligned}\ln\Pr(P_{n+1}|\bar{X}_{n+1}) &= \ln\alpha + \sum_{i=1}^{n+1}\ln(\prod_{j=1}^{i} P_{n+1}^{X_j}(1-P_{n+1})^{(1-X_j)}) \\ &= \ln\alpha + \sum_{i=1}^{n+1}\sum_{j=1}^{i}\Big[X_j \ln P_{n+1} + (1-X_j)\ln(1-P_{n+1})\Big].\end{aligned} \tag{5.6}$$

Differentiating with respect to P_{n+1} and equating to 0, where $\ln\Pr(P_{n+1}|\bar{X}_{n+1})$ and consequently $\Pr(P_{n+1}|\bar{X}_{n+1})$ are maximum:

$$\begin{aligned}\frac{\partial \ln\Pr(P_{n+1}|\bar{X}_{n+1})}{\partial P_{n+1}} &= \sum_{i=1}^{n+1}\sum_{j=1}^{i}\Big[\frac{X_j}{P_{n+1}} - \frac{(1-X_j)}{(1-P_{n+1})}\Big] \\ &= \sum_{i=1}^{n+1}\sum_{j=1}^{i} X_j - \sum_{i=1}^{n+1}\sum_{j=1}^{i} P_{n+1} \\ &= \sum_{i=1}^{n+1}\sum_{j=1}^{i} X_j - \frac{P_{n+1}(n+1)(n+2)}{2} \\ &= 0.\end{aligned} \tag{5.7}$$

The posterior probability P_{n+1} as a function of $n+1$ is given by:

$$P_{n+1} = \frac{2}{(n+1)(n+2)} \sum_{i=1}^{n+1} \sum_{j=1}^{i} X_j. \tag{5.8}$$

Assuming that $P_n = P$, we get the following formula for the estimator P:

$$P = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=1}^{i} X_j. \tag{5.9}$$

5.4 Adaptive Cooperative Hybrid Exploration

The IDM acceleration model causes congestion at the outer parts of the network (roads connecting source nodes with junctions) than at the inner parts of the network (roads connecting inner junctions) causing an unstable load in the whole traffic network. This problem was not clear before because in the original speed implementation [14], every vehicle does not take the normal time to decelerate and then to accelerate back again (e.g., a waiting vehicle jumps once the traffic signal turns green).

One of the proposed solutions for the congestion problem near source nodes is using more elaborate exploration policy. In this thesis, we propose a hybrid exploration technique based on both ϵ-exploration and softmax exploration. In softmax exploration, the traffic signal decision is chosen proportionally to the gain values:

$$\text{Weight of Traffic Signal Configuration Number } i = \frac{exp(g_i)}{\sum_{g_i} exp(g_i)}, \tag{5.10}$$

where g_i is the *cumulative gain* of the vehicles in the lanes of the traffic signal configuration number i. This hybrid exploration is *more adaptive* to the transient periods, particularly when a main road has very high congestion for some period of time (e.g., due to accidents or rush hours) while the side roads have much lower traffic demand. In this case, using ϵ-exploration solely leads to semi-permanent domination of the main road that causes long waiting times to the vehicles in the side roads.

Thus, we propose at every time step each junction decides whether to use the network-level "default" ϵ-greedy exploration ($\epsilon = 0.01$ as proposed in [10]) or to use softmax exploration. We found that the softmax exploration gives better trip waiting time results in case the gain of some traffic signal configuration exceeds the gain of any other configuration by 20% of its value (i.e., domination that might lead to blockage of the other possible configurations if ϵ-greedy exploration is used).

This hybrid exploration technique requires an explicit coordination between a junction agent and its neighboring junctions. A junction (or one of its direct neighbors) is said to be in a transient state if the cumulative gain of all vehicles in this junction keeps increasing (or decreasing) with 10% of its current value for 10 (or more) consecutive time steps. The cooperation is used to check if some junction is in a transient state, then this transient state will be most likely *transferred soon* to some neighboring junction; thus during this period it is preferable for the junction to use the softmax exploration.

We have proposed another kind of cooperation in [62] that depends on transferring the learned Q-values (with some decaying cooperation factor) from the ingoing lanes of a junction to the outgoing lanes. The proposed Q-function of every vehicle crossing from the ingoing lane of one junction to specific outgoing lane is given by:

$$Q_{\text{new}} = (1 - \alpha_t)Q_{\text{own}} + \alpha_t Q_{\text{transferred}}, \tag{5.11}$$

where $\alpha_t \in [0, 1]$ is the agent's learning rate. The Q-function can be reformulated to be as follows:

$$Q_{\text{new}} = Q_{\text{own}} + \alpha_t[Q_{\text{transferred}} - Q_{\text{own}}]. \tag{5.12}$$

In case $\alpha_t = 1$, Q_{new} will be updated to $Q_{\text{transferred}}$, in case $\alpha_t = 0$, we will completely ignore $Q_{\text{transferred}}$, and in case $\alpha_t \in]0, 1[$, we will give $Q_{\text{transferred}}$ some credit, but also consider the knowledge learned so far. As a starting value, we set $\alpha_t = k_t/100$ such that α_t will decrease as the Boltzmann temperature parameter k_t falls down (as proposed in [63]). Every crossing vehicle will take a weighted $Q_{\text{transferred}}$ (i.e., $Q(s, red)$ or $Q(s, green)$) from the last position the vehicle hits before crossing the previous junction.

This method leads to better performance in the transient period, however, we find that the steady state is worse. The new cooperative hybrid exploration technique improves both the transient and steady state periods. Note that in the proposed cooperation method the reward functions of neighboring agents are *independent*; this is mainly due to the reward function is *vehicle-based.*

5.5 Results

The original experiments in [14] were done with fixed vehicles generation rates. In our experiment, the generation probability distributions change over time in order to simulate the realistic road conditions of varying traffic demand entering and exiting a city (e.g., due to rush hours). We use the traffic network in Figure 5.1 for experimentation. This network consists of 12 edge nodes, 15 traffic signal nodes and 1 node without a traffic signal. There are 36 roads each consists of 2 lanes in each direction. We assume that all vehicles have equal length and number of passengers. We set the γ discount factor to 0.9 and set a random traffic signal decision chance to 0.01 for exploration purposes [1]. All edge nodes are set to the same vehicle generation rates and each edge node has an equal chance of being the destination of a new vehicle except its source node.

[1] In this experiment, we use the default ϵ-exploration technique as in [10] in order to focus on the performance gain achieved by the Bayesian probability interpretation. The performance gain achieved from using the adaptive cooperative hybrid exploration technique is mentioned on the top of the multi-objective traffic signal control framework presented in the next Chapter 6.

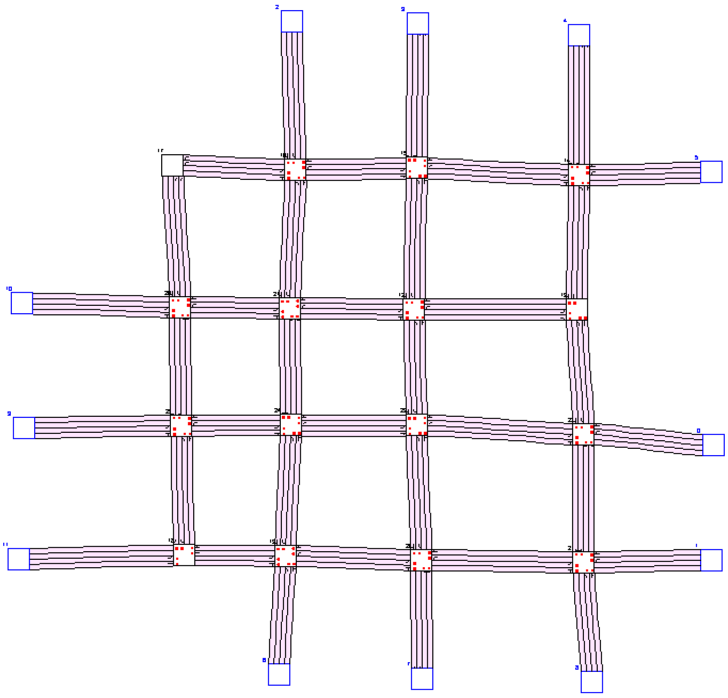


Figure 5.1: Traffic network with 12 edge nodes, 15 traffic signal nodes, and 1 node without a traffic signal.

The results of this experiment are averaged over ten independent runs. Every run has a seed equals its starting computer clock time (in milliseconds) and consists of 50,000 time steps. At time step 1, the generation distribution is set to $Weibull(k = 20, \lambda = 20)$ (i.e., at maximum a vehicle is generated every 17 time steps and at minimum a vehicle is generated every 21 time steps; this vehicle generation rate can lead to a free traffic situation). At time steps 5000, 10000 and 15000, the generation rate is fixed to 0.1, 0.2 and 0.3 respectively. At time step 20000, the generation distribution is set to $\mathcal{U}(a = 2, b = 4)$ (i.e., at maximum a vehicle is generated every 2 time steps and at minimum a vehicle is generated every 4 time steps; this vehicle generation rate can lead to a congested traffic situation). At time step 25000, the generation rate is fixed to 0.37 that represents the rush hour peak. The same initial behavior is repeated again; at time step 30000, the generation distribution is set to $Weibull(k = 20, \lambda = 20)$. At time steps 35000, 40000 and 45000, the generation rate is fixed to 0.1, 0.2 and 0.3 respectively.

As mentioned in [64], 3 performance measures are considered good indicators for measuring the congestion inside a traffic network [1]:

- **Average Trip Waiting Time (ATWT)**: represents the amount of average time spent by the vehicles waiting (i.e., not driving) in the whole trip.

- **Average Junction Waiting Time (AJWT)**: represents the average time a vehicle has to wait at each junction. This measure is a good indicator of the congestion level in the traffic network.

- **Total Waiting Queue Length (TWQL)**: The new generated vehicles entering the traffic network at any source node are put in a waiting queue if the lanes leading from this source node is fully congested. This measure is calculated by summing the waiting queue lengths at all source nodes. When there are unoccupied positions for the queued vehicles, i.e., congestion is being resolved, the total waiting queue length will decrease.

Figures 5.2, 5.3, and 5.4 compare the 3 performance indices of the Bayesian-based controller versus the TC-1 frequentist-based controller [10] with the pre-set dynamic generation distributions.

[1] The three performance indices ATWT, AJWT and TWQL that are used in this experiment are the default performance indices available in the GLD traffic simulator [14]. Our proposed performance indices (i.e., enhanced performance indices, co-learning performance indices, and new performance indices) are used in the next Chapter 6 to evaluate the performance of the multi-objective traffic signal controller.

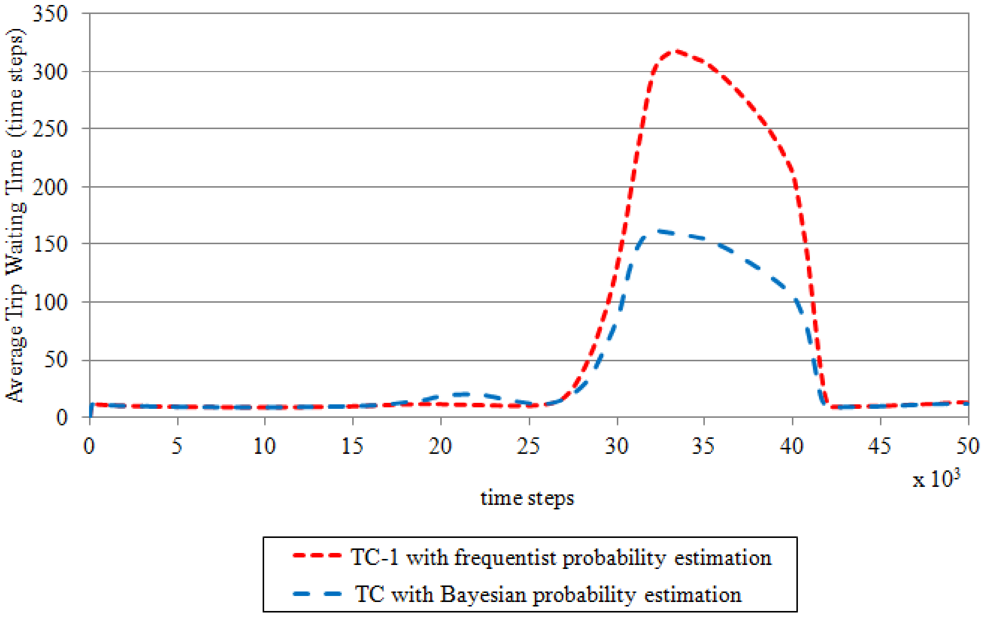


Figure 5.2: Average trip waiting time.

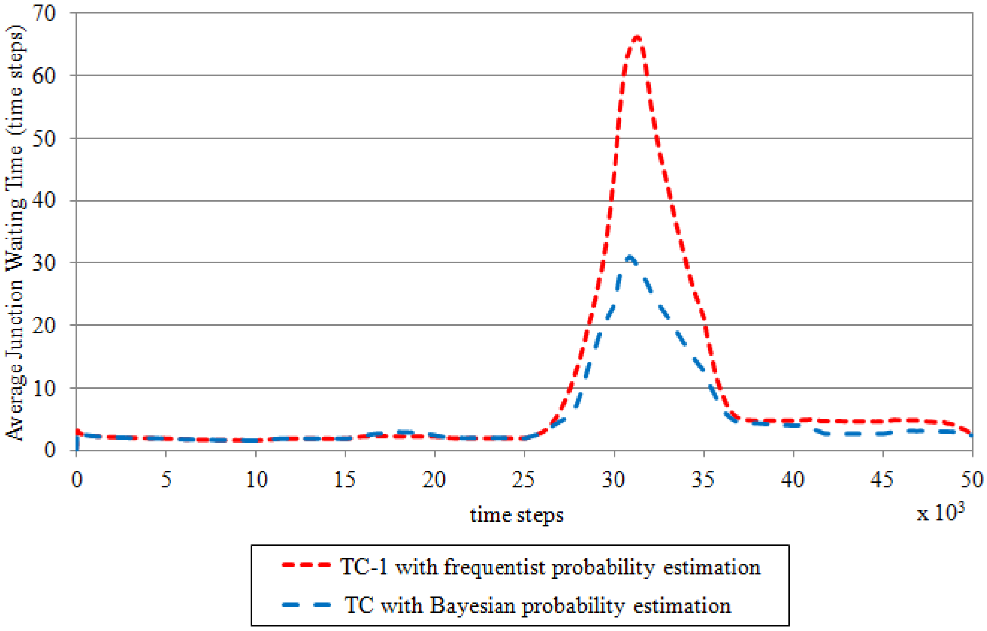


Figure 5.3: Average junction waiting time.

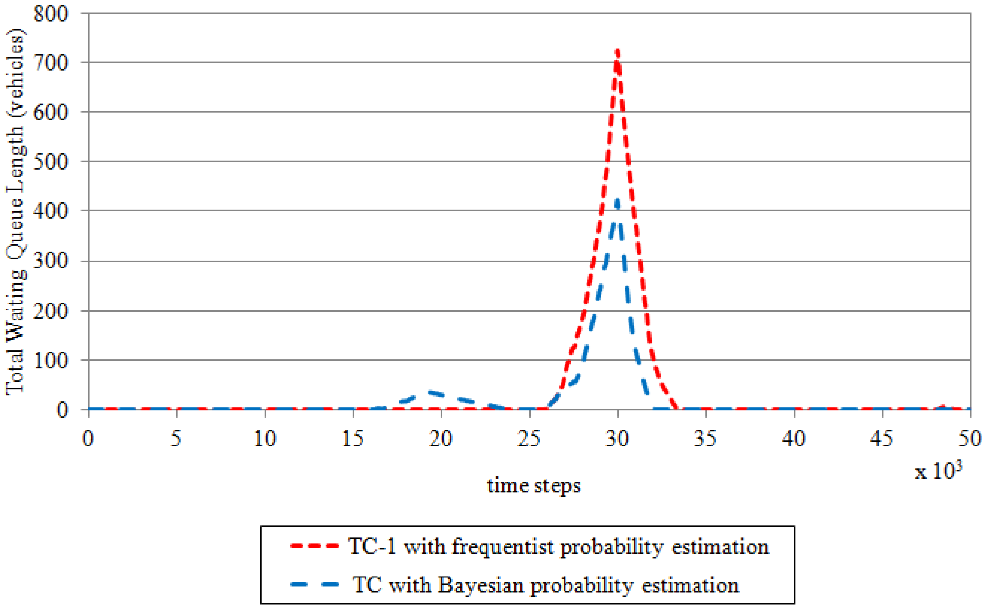


Figure 5.4: Total waiting queue length.

5.6 Validation

The mathematical model of estimating the parameters of the MDP based on the Bayesian probability interpretation represents one sort of system validation. In Equation 5.9, the agent takes the whole history into consideration in the learning process and gives higher weight to the initial experiences than the most recent ones. Since non-stationarity in the traffic network (e.g., due to accidents, rush hours, etc.) lasts for some limited time (i.e., transient periods), the system performance will be more stable and not much affected with these abrupt changes. Under these congested traffic periods, the Bayesian-based controller significantly outperforms the TC-1 frequentist-based controller. This is due to some next states (e.g., vehicles stay in the same positions due to congestion) have higher weights; this situation is quickly detected by the Bayesian-based controller.

5.7 Discussion

We noticed that in the low traffic congestion, both of the Bayesian-based controller and the TC-1 frequentist-based controller almost have the same performance. This is due to the transition probabilities to the next states of each vehicle position have equal chance. In addition, we noticed that when a traffic congestion lasts for a long time period, the two controllers almost have the same performance. This is due to the Bayesian-based controller has the overhead of learning the whole history. This can be avoided by learning partially from the history. For now, this is rare to happen in real-world scenarios, since congestion lasts for some limited time. Moreover, we noticed that giving higher weights to the initial experiences can lead the Bayesian-based controller to ignore limited pulses of traffic non-stationarity. Nevertheless, this is an unrealistic situation, since in real-world, traffic non-stationarity lasts for some period of time (e.g., rush hours).

5.8 Conclusion

In this chapter, we propose the Bayesian probability interpretation for estimating the parameters of the state transition probabilities of the MDP. We show that this estimation allows the traffic signal controller to make real time (online) adaptation in the sense that it responds effectively to the changing environment dynamics and the non-stationarity of the road network. As mentioned in the discussion section, under some conditions the two controllers may have the same performance, e.g., lengthy congested traffic or free flowing traffic. However, the Bayesian-based controller significantly outperforms the frequentist-based controller under limited time non-stationarities (which is typical to rush hours and accidents).

In addition, we show that how a novel cooperative hybrid exploration technique can be more adaptive to the changing dynamics in road conditions, i.e., improves the trip waiting time of vehicles during transient periods (e.g., due to rush hours).

Chapter 6

Multi-Objective RL for Traffic Signal Control

6.1 Introduction

Traffic signal control can be viewed as a multi-objective optimization problem. The multi-objective function can have a global objective for the entire road network or there may be different objectives for different parts of the road network (e.g., accidents avoidance especially in residential and school areas), or even different times of the day for the same part of the road network.

On the one hand, up to our knowledge, almost all traditional traffic signal control methods are single objective. On the other hand, as mentioned in [65], little work has been done in multi-objective RL with some exceptions, e.g., [66–68]. Thus, the framework proposed in this thesis is considered a novel contribution to the area of using multi-objective RL especially in the domain of traffic signal control.

In our model, we had two alternatives for implementing the multi-objective RL traffic signal control. The first is to use a separate Q-function for each objective, the second is consolidating all rewards in one Q-function. We decided to use the second alterative that is more suitable for the *vehicle-based* approach where each vehicle has two representative values $Q(s, red)$ and $Q(s, green)$.

In particular, similar to the underlying traffic signal control model [10], s is the state of the vehicle and $\Pr(s, a, s')$ is the state transition probability; both values are the same for the various objectives with respect to the same vehicle. The innovative part in this model specifically (and in the RL generally) is the design of the *reward function.* The *consolidated reward values* represent the core of the model which lead to the *final estimated gain* of every vehicle which affects the decision of the traffic signal controller.

6.2 Labeled Roads

There are different road types that vary in their speed limits, purposes and priorities. In urban traffic, as proposed in [69], the concern is on specific road types that are: major arteries, minor arteries, and local roads. Major arteries have large traffic volumes and represent the city entry and exit points. Their speed limits are usually within a range of 60-70 km/h. Minor arteries usually facilitate traffic flow from one major artery to another, and are generally shorter than major arteries. They are partially residential roads with local destinations such as schools. Their speed limits are usually within a range of 55-70 km/h. Local roads have low speed limits and usually carry low volumes of traffic. Hence, in our experiments a road connecting two nodes can only be either a major artery (main road) or a minor artery (side road). In the proposed multi-objective traffic signal control framework, the *reward* is function in the road type. For instance, the traffic junction agent learns to minimize the trip time in main roads while avoid accidents in resedential/side roads.

6.3 Multi-Objective Traffic Signal Control Model

Since adverse weather conditions affects the traffic demand, safety, and the traffic flow operations [70], we choose the traffic objectives accordingly. The proposed multi-objective function is given by:

$$\begin{aligned} Q(s,a) = \sum_{s'} \Pr(s,a,s')\Big[\big(&R_{\text{ATWT}}(s,a,s') + R_{\text{ATT}}(s,a,s') + R_{\text{AJWT}}(s,a,s') \\ &+ CF(s,a,s') \times R_{\text{FR}}(s,a,s') + R_{\text{GW}}(s,a,s') \\ &+ R_{\text{AA}}(s,a,s') + R_{\text{MS}}(s,a,s')\big) + \gamma V(s')\Big]. \end{aligned} \tag{6.1}$$

Let the distance traveled by the vehicle in the current time step be equal to Δp (always positive). The first reward represents the ATWT (the same as the single objective of Wiering's approach) and is given by: $R_{\text{ATWT}}(s,a,s')$ equals 10 or 3 in case the traffic signal is red or green respectively with $\Delta p \simeq 0$, otherwise equals 0. The reward values are scaled to the unit of 10 instead of 1 (as was initially proposed in [10]) to better discriminate the reward values in case the traffic signal is red or green.

The second reward represents the ATT. For instance, if the vehicle waits at the current position, i.e., $\Delta p \simeq 0$ (that leads to higher ATT), then it will be penalized by the reward value. In main roads, our controller enforces the ATT objective to dominate by using a *stronger* reward function:

$$R_{\text{ATT}}(s,a,s') = C_{\text{ATT}} \times (1 - 2^{-\Delta^2 p}). \tag{6.2}$$

In side roads (e.g., residential areas in which the main objective is to *avoid accidents*), the controller uses a *weaker* ATT reward function:

$$R_{\text{ATT}}(s,a,s') = C_{\text{ATT}} \times (1 - 2^{-\Delta p}). \tag{6.3}$$

C_{ATT} equals 10 or -10 in case the traffic signal is red or green respectively. Since the individual vehicle gain equals $Q(s, red) - Q(s, green)$, the reward has *negative* value when the traffic signal is green.

The third reward represents the AJWT. If the vehicle waits at the current junction, i.e., $tl' = tl$ (that leads to higher AJWT), then it will be penalized by the reward value. The AJWT reward function is given by: $R_{\mathrm{AJWT}}(s, green, s') = 0$ in case $tl' \neq tl$, otherwise equals 10 (the AJWT will increase if the current lane has red signal or is congested with green signal).

The fourth reward represents the flow rate (FR) in which we consider the *spatial queuing* that considerably affects neighboring junctions performances. If there is high congestion in the next lane, then the vehicle will be penalized by the reward value. The FR reward function is given by: $R_{\mathrm{FR}}(s, green, s') = 10$ in case $tl' \neq tl$, otherwise equals 0. Assume the number of blocks [1] taken by the waiting vehicles in the next lane [2] and the length of the next lane to be N and L respectively. Let $W = N/L$, then the Congestion Factor (CF) is given by [44]:

$$CF(s, green, s') = \begin{cases} 0, & \text{if } W \leq \theta, \\ 10 \times (W - \theta), & \text{if } \theta < W \leq 1, \\ 2, & \text{if } W > 1. \end{cases} \tag{6.4}$$

θ is a threshold whose best value equals 0.8 (as mentioned in [15]). For instance, if $N = 9$ meters and $L = 10$ meters, then $CF(s, green, s') = 1$ (the traffic signal controller try to minimize the FR when the next lane is congested). If $tl' \neq tl$, $CF(s, green, s')$ will decrease when the next lane at tl' is free. In this case, $Q(s, green)$ will decrease and thus the cumulative gain will increase (recall that a vehicle gain equals $Q(s, red) - Q(s, green)$)

[1] Like moreVTS [50], we set 1 block = 1 meter.

[2] Such kind of information can be coordinated between the neighboring junctions; each junction has such information through V2I communication with surrounding vehicles.

and accordingly the green phase length will be longer that allows more traffic to pass through, i.e., increasing vehicles flow rate.

The fifth reward represents achieving a traffic green wave (GW) and is implemented by checking the following conditions:

1. The current lane is part of a *main road.*

2. The current traffic signal is *green.*

3. The number of vehicles within distance ω from the traffic junction is $\in [1, \mu]$,

Then, $R_{\mathrm{GW}}(s, green, s') = -10$, otherwise equals 0. The best parameters values are $\omega = 25$ meters (as proposed in [50]) and $\mu = 3$ vehicles. Unlike the original RL model [10] that considers only the gain of the *waiting* vehicles when taking a traffic signal decision, our controller considers as well the *approaching* vehicles. In this case, the red signals might switch to green even before the vehicles reach the junctions creating an *emergent green wave* (the vehicles need not to slow down or stop at all). That occurs due to the increase of $Q(s, red)$ for the *approaching* vehicles.

The sixth reward represents the accidents avoidance (AA). The impact of an accident (i.e., vehicles moving with *very slow* speed or *stationary* at a short distance e beyond a *green* traffic signal) is propagated to the vehicles crossing the green signal. In this case, our controller uses a stronger AA reward function regardless of the road type:

$$R_{\mathrm{AA}}(s, a, s') = C_{\mathrm{AA}} \times \frac{1}{\Delta^2 p + 1}. \tag{6.5}$$

The best value of the short distance e beyond the traffic junction is 10 meters (as proposed in [71]). In residential and schools areas, our controller alleviates driver's aggressiveness by using the following AA reward function:

$$R_{\mathrm{AA}}(s, a, s') = C_{\mathrm{AA}} \times \frac{1}{\Delta p + 1}. \tag{6.6}$$

C_{AA} equals 10 or -10 in case the traffic signal is red or green respectively. This reward function assures that $Q(s, green)$ will increase at *high* vehicle speeds that decreases the gain leading the traffic signal to switch to red (i.e., forces vehicles to decelerate that helps in accidents avoidance in residential and schools areas). Note that in the simulation environment, the IDM acceleration model is a *collision-free* model [13]. Thus, we cannot measure efficiently the performance of the AA objective, e.g., by using *number of accidents* performance index. However, other performance indices still can give good indication, e.g., *average speed of vehicles.*

The seventh reward represents forcing vehicles to move within moderate speed (MS) range of minimum fuel consumption. The emission rates per kilometer are very high at very low average speeds [72]. On contrary, when vehicles travel at much higher speeds, they need very high engine loads, which consume more fuel, and which therefore lead to high emission rates [72]. Hence, the emissions-speed curve has a specific parabolic shape, with high emission rates on both ends and low emission rates at moderate speeds of around 65-97 km/h [72]. Thus, if the distance traveled per time step (resulting in the motion from a controller state s to a next state s') is smaller or greater than the moderate speed limits (for main roads is 60-70 km/h and for side roads is 55-70 km/h), we set $R_{\mathrm{MS}}(s, a, s')$ to C_{MS} or -C_{MS} respectively, otherwise equals 0. C_{MS} equals 10 or -10 in case the traffic signal is red or green respectively.

6.4 Results

The original experiments in [14] were done with *fixed* vehicles generation rates. In our experiment, the generation probability distributions change over time in order to simulate the realistic road conditions of varying traffic demand entering and exiting a city due to rush hours, accidents, and weather conditions. We use the traffic network in Figure 2.1 for experimentation. This network consists of 12 edge nodes and 9 traffic signal nodes.

There are 6 roads each of 2 lanes in each direction. The 3 horizontal roads are the main roads (where there is higher possibility of traffic *green wave* creation) and the 3 vertical roads are the side roads. We assume that all vehicles have equal length and number of passengers. The γ discount factor is set to 0.9. The duration of each simulation time step is 0.25 second. The results of this experiment are averaged over ten independent runs. Every run has a seed equals its starting computer clock time (in milliseconds) and consists of 100,000 time steps which is about 400 minutes.

As mentioned in [49], the proportion of vehicles flowing in a main road to those on a side road is in the ratio of 100:5 (this setting is close to real-life traffic scenarios on many busy corridors and grid networks). Accordingly, we set the default generation rate of the main and side roads to 0.04 (576 vehicles per hour [1]) and 0.002 ($\simeq$ 30 vehicles per hour) respectively. We set the default weather condition in the main and side roads to *normal rain* and *sandstorm* respectively and the IDM *desired velocity* parameter v_0 to 108 km/h and 77 km/h respectively. We set the speed limit of the main and side roads to 60 km/h and 55 km/h respectively.

In order to clarify the case where the vehicles in the side roads will wait for very long times, i.e., main road *domination*, when the controller uses ϵ-greedy exploration, we schedule the destination frequency such that 90% of the traffic demand generated from the source edge node of a main road will exit from its destination edge node. The remaining 10% of the generated traffic demand will exit uniformly from the other 10 edge nodes. We use the same destination frequency for the side roads.

In order to simulate the *transient periods* in the main roads, the traffic demand is *dramatically* changed every 100 minutes where the distribution of the inter-arrival time is set to $\mathcal{U}(a = 2, b = 4)$, i.e., at maximum a vehicle is generated every 2 time steps (7200 vehicles per hour) and at minimum a vehicle is generated every 4 time steps (3600 vehicles per hour), continued for a period of 5 minutes (this corresponds to *extremely high*

[1] This rates complies with the vehicles rate of Wetstraat at normal congestion periods which is provided by the *Ministry of the Brussels-Capital region* [50].

congested traffic situation). In these periods, we set the weather condition to *dry* and the IDM *desired velocity* parameter v_0 to 120 km/h. Dashed vertical lines clarify times at which changes occur in dynamics.

Figures 6.1 through 6.10 compare the performance of the multi-objective controller (using the Bayesian probability interpretation) with hybrid exploration based on the *transient state* of the current and neighboring junctions (i.e., cooperation-based) versus the TC-1 controller [10] (single objective with frequentist probability interpretation using ϵ-exploration). The former controller is represented by blue long dashes, while the latter controller is represented by red square dots. Note that the contributions added to the GLD traffic simulator are applied on all controllers for *fair* performance evaluation.

We evaluate the performance of our proposed system in comparison with two adaptive control strategies which are also based on AI methods: Self-Organizing Traffic Lights (SOTL) [50] and a Genetic Algorithm (GA) [14]. Both controllers are already implemented in the GLD traffic simulator, namely "SOTL platoon" and "ACGJ-1" respectively. The SOTL controller turns a traffic signal to green if the time elapsed, since the signal turned red, reaches a certain threshold ($\phi_{min} = 5$ seconds). Given that the number of vehicles in the lane controlled by this traffic signal reaches another threshold ($\theta = 50$ vehicles) within a distance of 80 meters from the red signal. In the intersecting lane (which will be switched to red), the integrity of a *platoon of vehicles* is maintained by preventing the platoon tail from being cut (platoon tail $< \mu = 3$ vehicles) within a distance $\omega = 25$ meters from the green signal, while allowing the division of long platoons. The ACGJ-1 controller creates a *genetic population* every time step and tries to find the optimal city-wide configuration. The parameters of this algorithm are as follows: mutation factor $\mu = 0.05$, population size $s = 200$, and maximum number of generations $maxGen = 100$.

For performance evaluation, we use the following measures of effectiveness (MOEs): ATT, ATWT, average speed, average number of trip stops, average number of trip absolute stops, total arrived vehicles (network throughput), percentage of arrived vehicles, percentage of rejected vehicles (indicating network utilization), maximum and average queue lengths. Under the congested and free traffic situations (e.g., due to adverse weather conditions), our controller significantly outperforms the single objective controller.

For the *co-learning ATT*, Figure 6.1, and the *co-learning ATWT*, Figure 6.2, the mean values are lower $\simeq$ 8 and 6 times respectively when using the multi-objective controller. Figures 6.1 and 6.2 [1] show that the multi-objective controller has much more stable response to the changing dynamics (occurring every 100 minutes). The response of the single objective controller to the transient periods is severe.

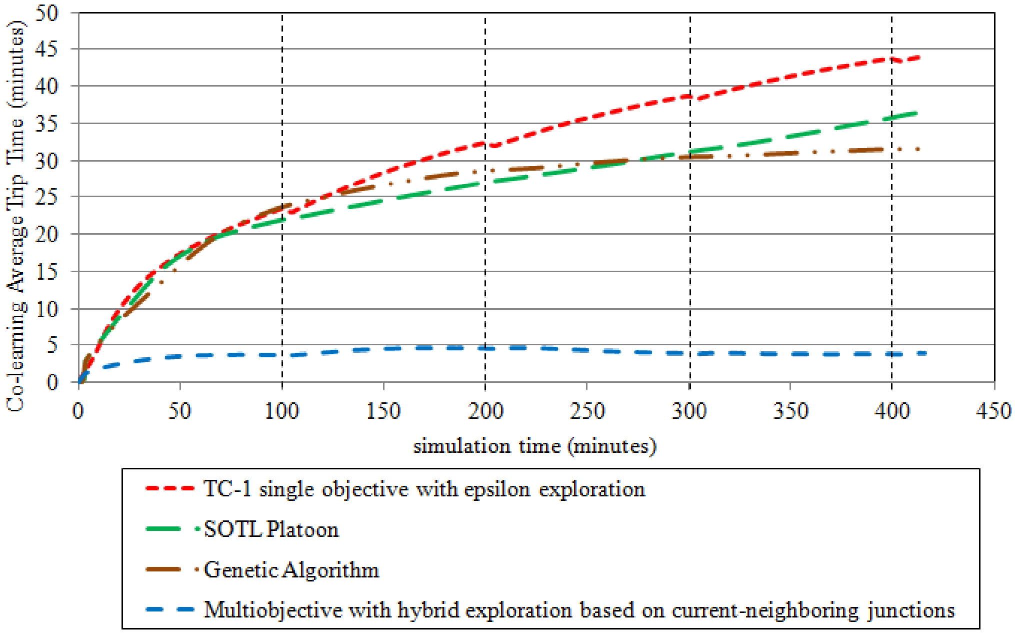


Figure 6.1: Co-learning average trip time.

[1] Note that for the SOTL and ACGJ-1 controllers, there is no estimator for the traveling vehicles in the co-learning performance indices, thus we measure the performance based on the arrived vehicles only.

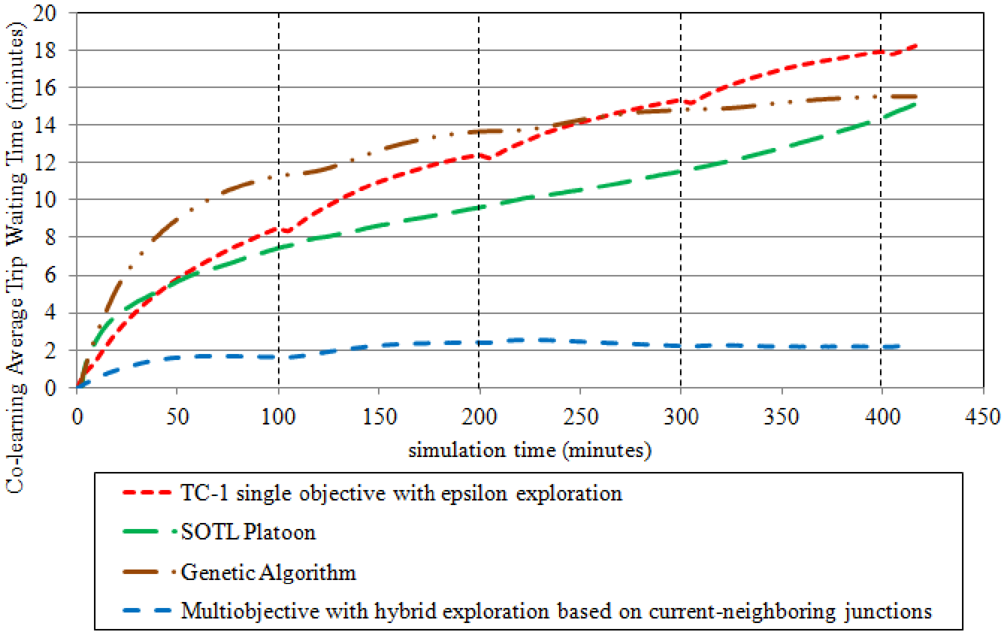


Figure 6.2: Co-learning average trip waiting time.

Figure 6.3 shows that the *average speed of vehicles* is higher $\simeq$ 8 times when using the multi-objective controller. This means lower congestion and faster arrival to destinations (that increases the driver's satisfaction). Figure 6.4 shows that when using the single objective controller, a vehicle stops at almost all junctions that the vehicle crosses before exiting the network ($\simeq$ 3 junctions). Whereas, when using the multi-objective controller, a vehicle stops on average at only 1 junction. This creates a traffic *green wave.* Figure 6.5 shows that the *vehicle stops* is lower $\simeq$ 22 times when using the multi-objective controller. This will save fuel consumption and consequently is more environment friendly. Moreover, the *number of vehicles stops* can be also considered as a good measure of the *total delays* that encounter vehicles.

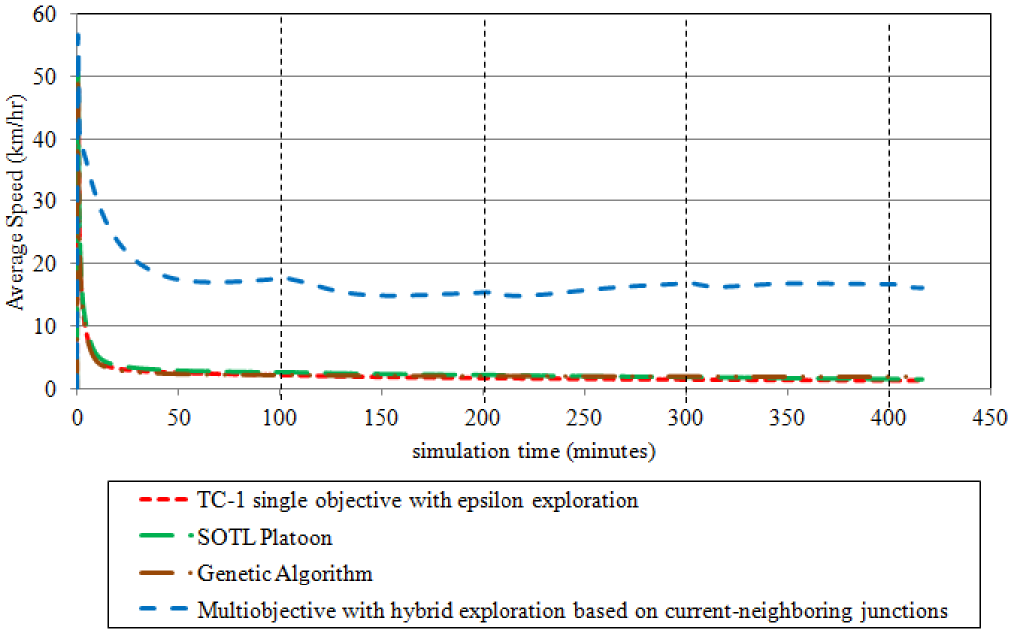


Figure 6.3: Average speed.

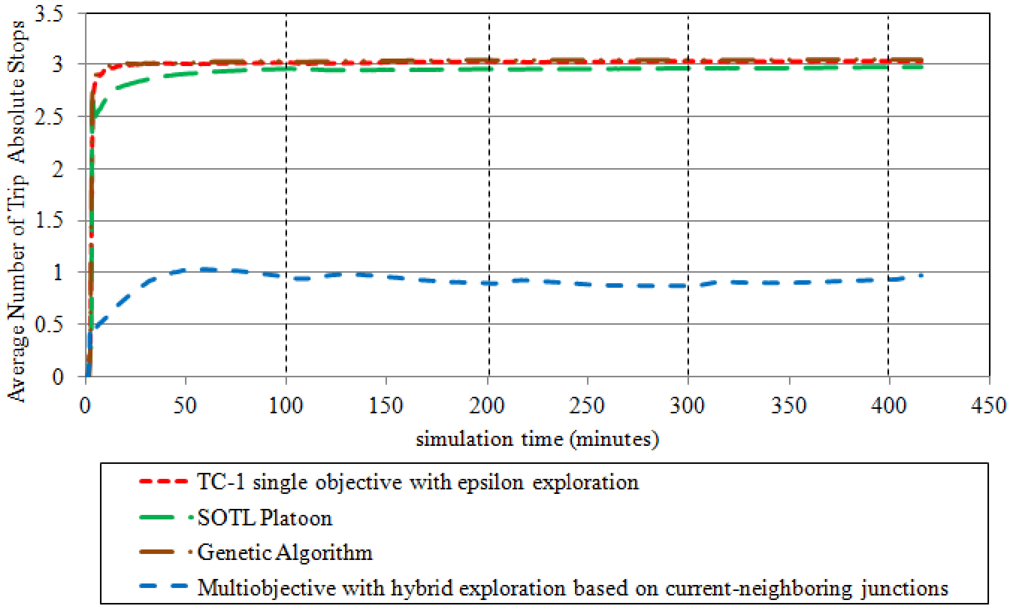


Figure 6.4: Average number of trip absolute stops.

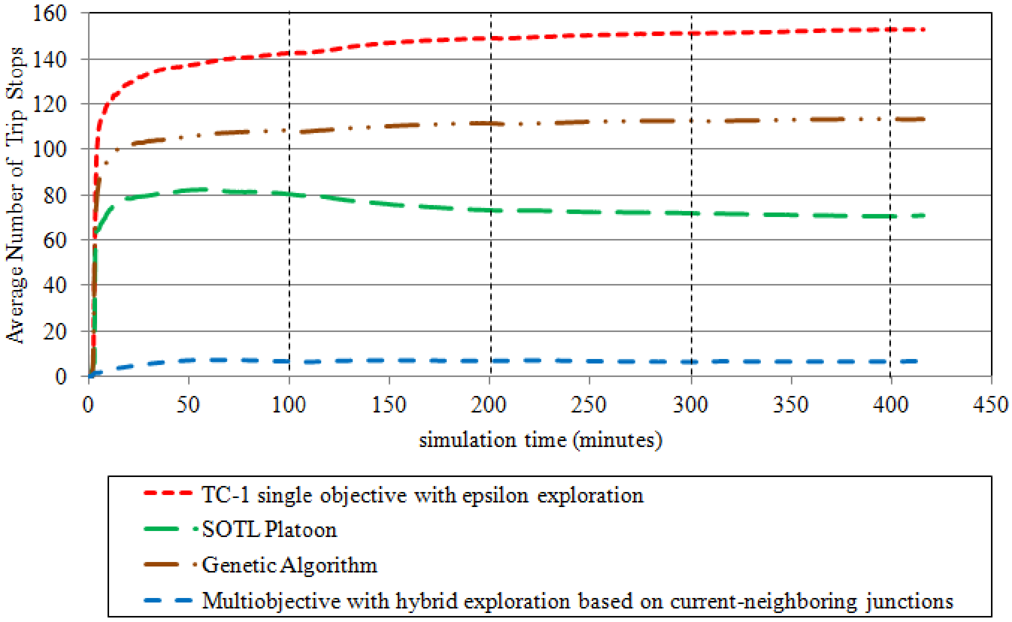


Figure 6.5: Average number of trip stops.

Figure 6.6 compares the *total number of arrived vehicles* (i.e., traffic network throughput) for both controllers. This performance index is useful especially in evaluating the *flow rate objective*; lower congestion rates means larger number of vehicles cross the traffic junctions and hence larger number of vehicles arrive to their destinations. Figure 6.7 shows that the mean value of the *arrived vehicles percentage* is higher by $\simeq$ 22% when using the multi-objective controller. This performance index is a good indicator of the *network throughput*, and accordingly the traffic flow rate. Figure 6.8 shows that the *rejected vehicles percentage relative to all generated vehicles* (i.e., generated but cannot join the network due to overfull ingoing lanes) is lower $\simeq$ 4 times when using the multi-objective controller. This performance index is a good indicator of the *network congestion*, and accordingly the *network utilization*.

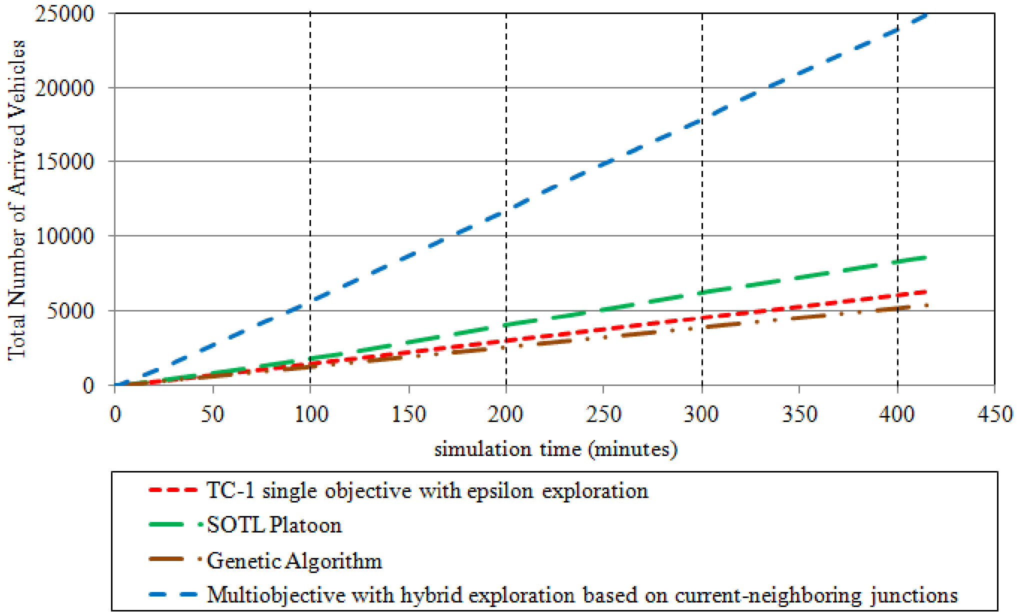


Figure 6.6: Total number of arrived vehicles.

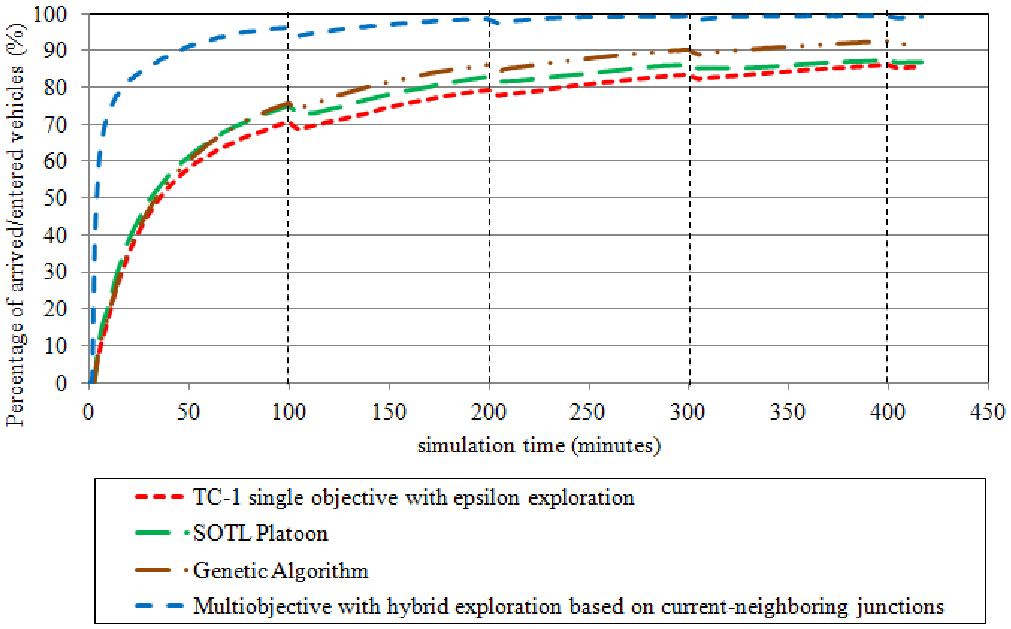


Figure 6.7: Percentage of arrived to entered vehicles.

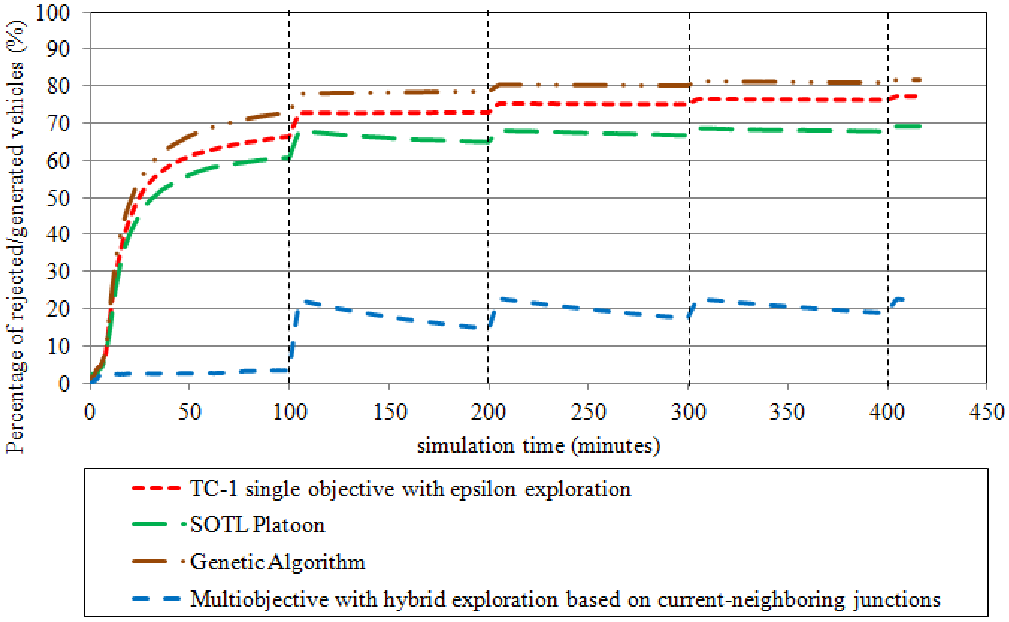


Figure 6.8: Percentage of rejected to generated vehicles.

Figure 6.9 shows that the mean value of the *maximum number of vehicles waiting at any junction* in the entire network is lower by $\simeq 10$ vehicles when using the multi-objective controller. This performance index is a good indicator of the *driver's comfort* (i.e., waiting in shorter queues). Figure 6.10 shows that the *average number of vehicles waiting at any junction* is better when using the multi-objective controller compared to the other controllers.

We use the *co-learning ATWT* performance index in order to show the impact of using the cooperative hybrid exploration (discussed in Chapter 5) on the long waiting times of vehicles in side roads when using the ϵ-exploration solely. Figure 6.11 compares the *co-learning ATWT* of the multi-objective controller with hybrid exploration based on the transient state of the current junction, the neighboring junctions, or the current-neighboring junctions versus the ϵ-exploration. The mean value of the multi-objective controller with hybrid exploration based on the current-neighboring junctions is better by $\simeq 10\%$ than the multi-objective controller using ϵ-exploration.

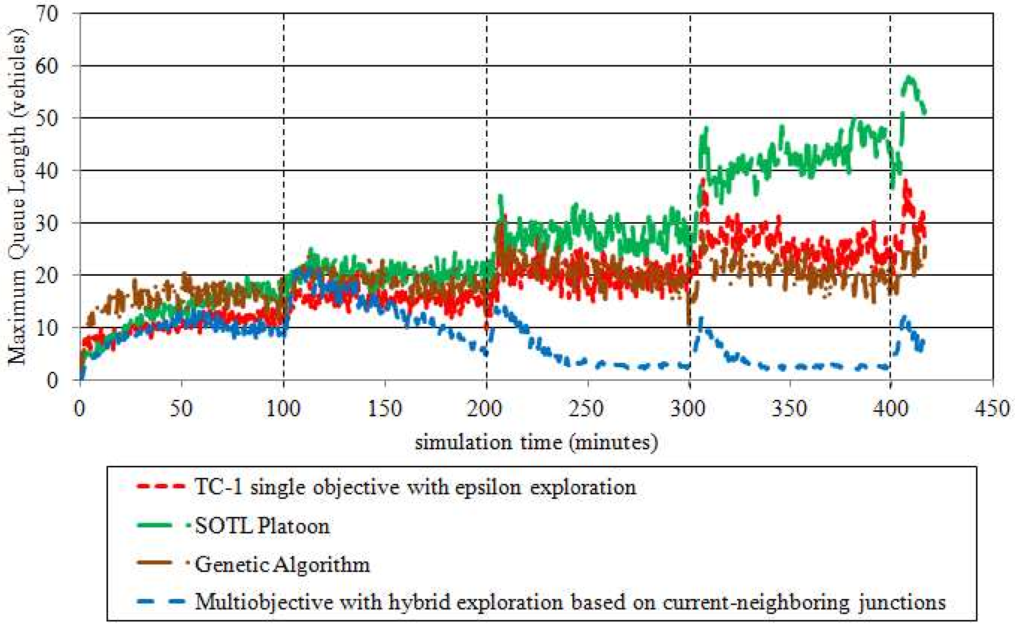


Figure 6.9: Maximum queue length.

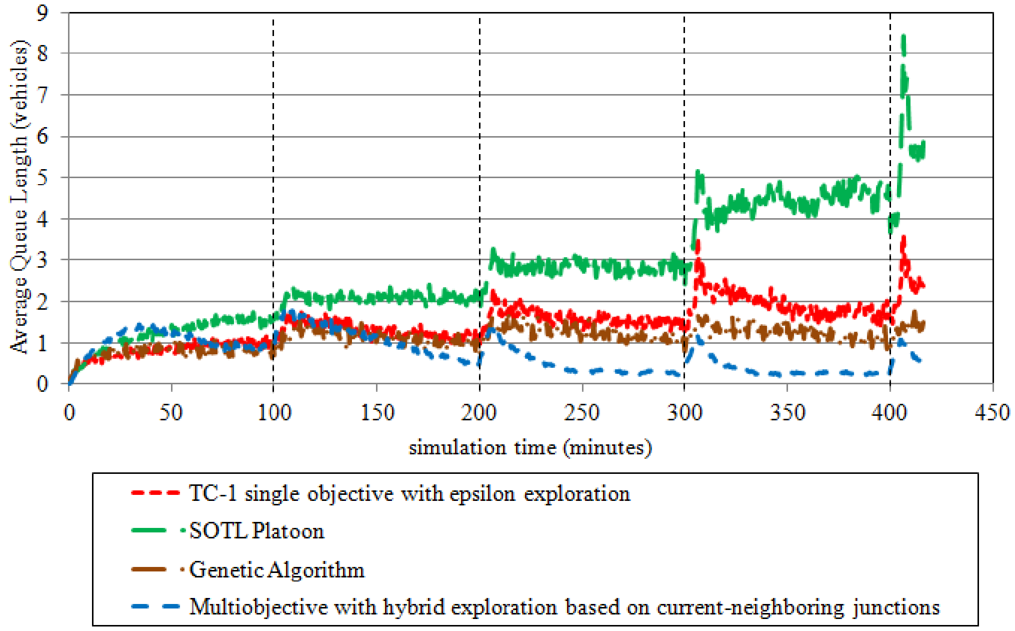


Figure 6.10: Average queue length.

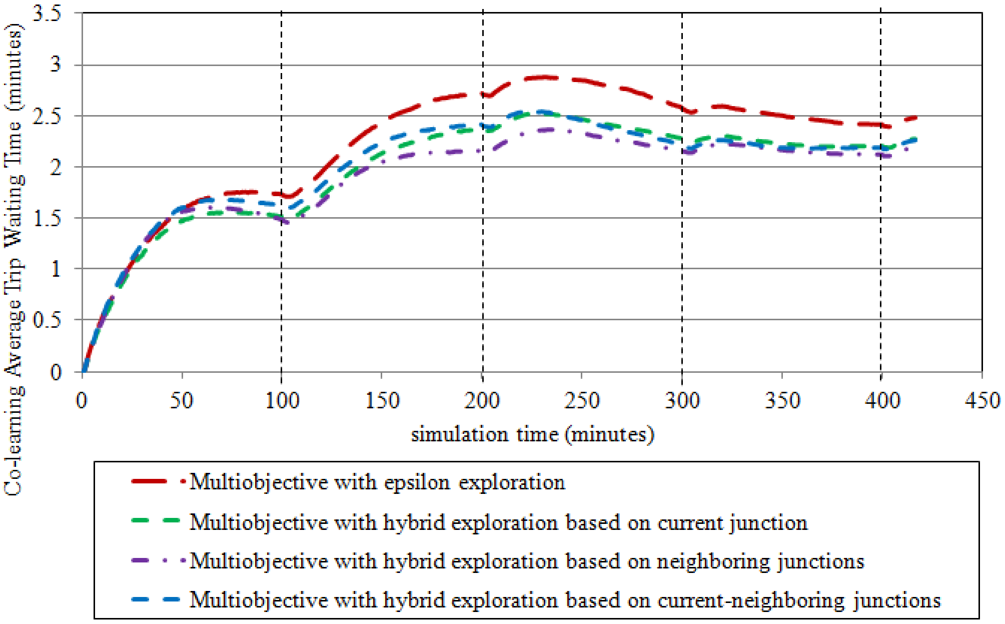


Figure 6.11: Co-learning ATWT of the multi-objective controller with hybrid exploration transient state based on: current junction, neighboring junctions, current-neighboring junctions, and with ϵ-exploration.

6.5 Validation

In order to better realize the contributions presented in this thesis, here we give some insights about how the results presented in this thesis can be validated. Firstly, the mathematical model of the accumulated reward (Q-function) formulation in Equation 6.1 in which the various reward functions (even of the conflicting objectives) work in harmony to optimize the final value function. This is generally achieved by decreasing $Q(s, green)$ or increasing $Q(s, red)$ and thus the cumulative gain will increase (recall that a vehicle gain equals $Q(s, red) - Q(s, green)$) and accordingly the green phase length will be longer that allows more deserving vehicles to cross the junction.

Secondly, comparing the performance of our multi-objective traffic signal controller with the *theoretically optimum* solution is *computationally prohibitive.* Our multi-objective traffic signal controller is mainly based on *online decision making*. Whereas, it is *computationally demanding* to compute the *theoretically optimum* solution at every time step, e.g., using Little's law of *Queueing Theory* which may ignore some traffic specific characteristics, e.g., vehicles speeds, inter-dependability between consecutive junctions, etc. However, we can simply say that the *theoretically optimum* ATWT is *zero*. In addition, the *theoretically optimum* ATT can be calculated from the optimum average speed in main roads (equals 70 km/h $\simeq$ 20 m/sec) and the average traveled distance (equals 1.12 km = 1120 m).

Note that the average traveled distance is calculated based on the destination frequencies (where 90% of the traffic demand generated from the source edge node of a main road will exit from its destination edge node.) Moreover, this average traveled distance complies with the average absolute number of vehicle stops, i.e., 3 stops, Figure 6.4. Thus, for the traffic network in Figure 2.1, the *theoretically optimum* ATT equals 1120 m $\div$ 20 m/sec = 56 sec $\simeq$ 1 min. In comparison with the performance of our multi-objective traffic signal controller (the mean value of the ATT $\simeq$ 4 min and the mean value of the ATWT $\simeq$ 2 min) considering the *dramatic change* in the traffic demand every 100 minutes; our traffic signal controller yields very good results.

Thirdly, the mean value of the average speed of our multi-objective controller is $\simeq$ 17 km/h. This value complies with the average speed in many mega cities which guarantees safety in urban areas. Moreover, this average speed value is not too low in the sense that yields low fuel consumption especially when being compared to the average speed of other controllers, Figure 6.3. In addition, the mean value of the average speed using our multi-objective controller (i.e., $\simeq$ 17 km/h) complies with the mean value of the ATT presented in Figure 6.1 (i.e., $\simeq$ 4 min); given that the average traveled distance is 1120 meters as mentioned previously. This yields some kind of validation for the presented results.

6.6 Discussion

The proposed multi-objective traffic signal controller does not overshoot at all in transient periods in Figures 6.1 and 6.2. This is due to the triple effect of:

1. The reward function of the ATWT tackles the *Zeno* phenomena discussed in Chapter 4 (giving stationary vehicles some penalty smaller than the one given when the traffic signal is red). In addition, the reward function of the ATT is function in the road type as discussed in Chapter 6 (in main roads, our controller enforces the ATT objective to dominate by using a stronger reward function).

2. Using the Bayesian probability interpretation for estimating the parameters of the underlying MDP which responds effectively to the traffic non-stationarity lasting for limited period of time. As mentioned in Chapter 5, the current estimation becomes the prior for the next time step. This estimation is more stable and more adaptable to the changing environment dynamics.

3. Using the novel adaptive cooperative exploration technique (discussed in Chapter 5) in which the impact of any transient period is propagated between the neighboring junctions to avoid very long waiting times of vehicles in side roads (i.e., main road domination).

Note that the objectives could be classified into three conflicting groups: (1) ATWT, ATT, AJWT, FR, GW, (2) AA, and (3) MS. In particular, to position our work in the scope of multi-objective reinforcement learning, we do not compute the Pareto front (that is computationally demanding) rather, we use multi-objective scalar optimization (i.e., scalar addition for the rewards representing the different objectives). For example, the Pareto front may include one optimal solution in which the trip time is minimized to the level that does not maximize the fuel consumption (in case a vehicle is moving too fast). The study of such points of optimality is subject to a future study.

Moreover, despite the proposed multi-objective traffic signal controller is based on *conflicting* objectives, the performance indices are *not conflicting*. For instance, the *number of vehicle stops* is decreased when using out multi-objective controller, that indicates lower fuel consumption, while the *trip time* is also decreased, that indicates a possibility of higher fuel consumption. However, we ignore this possibility because in urban areas the *trip time* is scarcely decreased to the level at which high amount of fuel is consumed.

Table 6.1 presents the mean values of the various MOEs when adding the objectives incrementally. This gives a better view about the impact of adding the reward function of every objective on the various performance indices. One interesting conclusion is that the addition of every reward function almost affects the entire set of MOEs, i.e., not just the corresponding MOE being optimized; this assures that machine learning is inherently a *multi-objective task* (as mentioned in [65]). Moreover, this opens the door to a future study of the impact of every individual objective, i.e., instead of being added incrementally. In addition, one can examine the performance when changing the order of adding the reward function of every objective. Finally, those proposed experiments should be tried on various traffic patterns; this can clearly show the impact of every objective under the specific conditions at which this objective optimally behaves.

Table 6.1: The mean values of the various MOEs when adding the objectives incrementally.

Objective/Index	ATWT	ATT	AJWT	Speed	Abs. Stops	Stops	Arrived%	Rejected%	Avg. Q
ATWT	0.14	2.43	0.03	26.59	0.52	1.76	96.06	15.48	0.04
ATT	4.06	5.80	0.74	12.46	0.86	6.93	94.75	15.07	1.52
AJWT	4.95	7.56	1.07	10.50	1.03	9.42	93.60	16.69	1.94
FR	5.45	8.36	1.21	9.80	1.06	9.94	93.23	17.44	2.12
GW	6.66	8.80	1.41	9.56	0.95	9.38	93.39	15.44	2.68
AA	2.64	5.01	0.54	14.60	0.99	7.46	94.85	16.62	1.00
MS	2.03	3.99	0.38	17.23	0.91	6.47	95.43	15.52	0.74

The proposed multi-objective traffic signal control framework includes the following parameters:

1. γ: the discount factor that is used to decrease the influence of the previously learned V-values (equals 0.9).

2. C_i: the parameter multiplied by the reward value to discriminate the reward values in case the traffic signal is red or green (equals 10 or -10).

3. θ: the threshold used in the congestion factor calculation (equals 0.8).

4. e: the short distance beyond the green traffic signal where vehicles are moving with very slow speed or stationary for the *accidents avoidance* objective (equals 10 meters).

5. ω: the distance from the traffic junction used for checking the tail of a platoon to achieve the *green wave* objective (equals 25 meters).

6. μ: the threshold of the number of vehicles representing the platoon tail (equals 3 vehicles).

Although almost all best values of the various parameters are taken from the literature, a study need to be conducted to check the impact of variations in those parameters values on the performance of the proposed multi-objective traffic signal controller. For instance, Table 6.2 repeats the data presented in Table 6.1 while changing the threshold of the number of vehicles representing the platoon tail; $\mu = 2$ vehicles. We can see that the impact of adding some related-objectives is better, e.g., AJWT, FR, and AA, while the impact of adding some other objectives is worse, e.g., ATWT, GW. Moreover, the final performance (of adding the entire seven reward functions) is better when setting $\mu = 3$ vehicles.

Table 6.2: The mean values of the various MOEs when adding the objectives incrementally; $\mu = 2$ vehicles.

Objective/Index	ATWT	ATT	AJWT	Speed	Abs. Stops	Stops	Arrived%	Rejected%	Avg. Q
ATWT	0.18	2.51	0.04	25.84	0.53	1.92	95.99	15.60	0.05
ATT	4.12	5.85	0.75	12.47	0.86	6.92	94.80	15.17	1.56
AJWT	4.71	7.20	1.02	11.19	1.01	9.00	93.83	16.35	1.90
FR	4.45	6.81	0.95	11.35	0.99	8.65	94.02	16.30	1.79
GW	6.95	9.47	1.56	9.30	0.99	9.76	93.03	15.82	2.89
AA	2.33	4.77	0.47	14.94	1.00	7.62	94.78	17.40	0.86
MS	2.19	4.27	0.42	16.30	0.93	6.76	95.29	15.63	0.82

Another issue worth discussion is studying the *time complexity* of the proposed multi-objective traffic signal control framework. On the one hand, in the work presented in this thesis, we did not optimize the *execution time* of the controller, e.g., using parallel programming techniques. However, the time complexity of the multi-objective controller versus the single objective one is comparable. This is mainly due to the scalar addition of the reward functions of the multi-objective controller. Thus, the high performance gain of the multi-objective controller (as shown by the various performance indices) does not come with a high computation cost. On the other hand, the proposed traffic signal controller is based on *online learning* and accordingly online decision making, thus there is no specific time threshold for reaching a terminal state. This is mainly due to the *continuous learning* of the changing environment dynamics.

6.7 Conclusion

We develop a multi-objective traffic signal control framework that is adaptive to the changing environment dynamics and non-stationarity of the road network. Traffic non-stationarity are simulated by changing the traffic flow and traffic demand resulting from changing the weather conditions. We show that using some innovative reward design (e.g., function in the vehicle speed), the various performance indices using the multi-objective controller significantly outperforms the underlying single objective controller.

In addition, we show in this chapter that using an advanced exploration technique (that was introduced in Chapter 5) can boost the performance of the RL-based traffic signal controller. We want to check the role of further exploration techniques in enhancing the various performance indices (i.e., not only the trip waiting time of vehicles). Another possible improvement is generalizing the role of exploration in enhancing the performance when the congested periods are continued over an extended course of time or when no change in dynamics occur for a long period of time (despite these are rare cases).

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we present an adaptive multi-objective reinforcement learning for traffic signal control based on cooperative multi-agent framework. We show that using RL for solving control optimization problems in continuous state-space (specifically in the traffic signal control domain) has some challenges that affect the reward design of the model. In addition, we show that using the Bayesian probability interpretation to estimate the parameters of the MDP probabilities can result in a better response to the traffic non-stationarity. Traffic non-stationarity are simulated by changing the traffic flow and traffic demand resulting from changing the weather conditions.

Generally, the application of multi-objective RL optimization is still a challenging task, and particularly, in the domain of traffic signal control. However, using an innovative reward design on a scalar-based form can greatly boost the various performance indices without the overhead of other computationally demanding techniques (e.g., using Max-plus, Pareto front optimization, etc.) Moreover, we show that the application of new exploration techniques that are adaptive to the current traffic conditions can greatly affect the performance of the traffic signal controller.

Under the congested and free traffic situations, the proposed multi-objective traffic signal controller significantly outperforms the underlying single objective controller. For instance, the average trip and waiting times are lower $\simeq 8$ and 6 times respectively when using the multi-objective controller.

7.2 Future Work

The work presented in this thesis opens the door to a bulk of future work. For better organizing the suggested directions for future work, we categorize our ideas into future work in traffic signal control model and future work in traffic signal simulation model.

7.2.1 Traffic Signal Control Model

Firstly, we want to investigate the multi-objective optimization using the *Pareto front* approach. This will be a challenging task in the domain of reinforcement learning traffic signal control. For instance, we may separate the benefits associated with each traffic objective and identify the trade-offs associated with each one. We can introduce the objectives one by one and show how these objectives affect the various performance indices.

Secondly, we want to check the *robustness/sensitivity* of the proposed multi-objective traffic signal controller due to *noisy* input provided by sensors, i.e., *partial observability* of state-space. In [51], the authors overcome the *partial observability* of the traffic state by estimating belief states and combining this with multi-agent variants of approximate Partially Observable Markov Decision Process (POMDP) solution methods. It was shown that the state transition model and value function could be estimated effectively under *partial observability.* Thirdly, we plan to control traffic signals in *roundabouts.* An initial idea is based on *game-theory*; every vehicle in every approach (i.e., road in the roundabout) will play a game with other vehicles in the other approaches. The precedence of roundabout crossing will be determined accordingly.

Finally, our long-term goal is to implement and test the proposed controller on *real traffic network* in Egypt. However, there are some *challenges* of deployment in Egypt, e.g., unlanned roads, chaotic driving behavior, etc. We can overcome the *unlanned roads* by *state-space approximation* to the vehicles positions. For the *chaotic driving behavior*, the traffic signal controller can learn the *non-stationarities* due to the *aggressive undisciplined* driving behavior in Egypt. We need to integrate the traffic control system with *sensors* (through loop detectors in roads, cameras, and/or communication with vehicles using GPS/Wi-Fi sensors). Generally, we still want to study the rest of *theoretical properties* of the proposed multi-objective traffic signal control framework.

7.2.2 Traffic Signal Simulation Model

Firstly, we want to examine the behavior of the proposed traffic signal controller when *simulating accidents risks* by generating *random perturbations* that lead to sudden braking or lane-changing manoeuvres. In addition, we need to examine the controller behavior when simulating an *accident* at some part of the traffic network (that need special handling from the traffic signal controller) while in another part of the network there is a *free-flowing* traffic. Secondly, we need to apply the proposed traffic signal controller on different *traffic patterns* (e.g., non-symmetric networks with non-parallel arterials) and analyze *traffic patterns* in which the proposed traffic signal controller optimally behaves.

Thirdly, we want to add in the GLD an *amount of fuel consumption* performance index. This performance index can be calculated from the IDM based fuel consumption model proposed by Treiber *et al.* [73] which directly calculates the *fuel consumption* and derived emission such as CO_2. However, we need first to add in the GLD some vehicle-specific characteristics, e.g., the power of vehicles engines.

Fourthly, we plan to use *learning-based* techniques to estimate the *optimal* values of the parameters of the IDM acceleration model based on the driving behavior in Egypt. Finally, we need to use a more *advanced* traffic simulator (rather than the GLD) to simulate a real traffic network and examine the proposed controller behavior accordingly. One proposed solution is the integration of the proposed traffic signal control framework with a *3D traffic simulator* that allows for *human drivers* as well as *agent vehicles*. This simulator has been being developed as a collaboration of our research team with the Prendinger Laboratory in the National Institute of Informatics (NII), Tokyo, Japan.

References

[1] A. Pizam, “Life and tourism in the year 2050,” *International journal of hospitality management*, vol. 18, no. 4, pp. 331–343, 1999.

[2] D. Schrank, T. Lomax, and B. Eisele, “TTI’s 2011 urban mobility report,” Texas Transportation Institute (TII), The Texas A&M University System, U.S. Dept. of Transportation, University Transportation Center for Mobility, TII Report Exhibit B-15, Sep. 2011.

[3] N. H. T. S. A. U.S. Department of Transportation, “2010 motor vehicle crashes: Overview,” NHTSA’s National Center for Statistics and Analysis, 1200 New Jersey Avenue SE., Washington, DC 20590, Traffic Safety Facts Research Note DOT HS 811 552, Feb. 2012.

[4] Z. Nakat and S. Herrera, “Cairo traffic congestion study,” Report prepared by ECORYS Nederland BV and SETS Lebanon for the World Bank and the Government of Egypt, Phase 1 - Final Report 71852, Nov. 2010.

[5] Egypt Central Agency for Public Mobilization And Statistics (CAPMAS), 2010. Last accessed at 12th Jan 2013. [Online]. Available: http://www.capmas.gov.eg/reports/trans/frm_trans6.aspx

[6] K. A. Abbas, "Traffic safety assessment and development of predictive models for accidents on rural roads in Egypt," *Accident Analysis & Prevention*, vol. 36, no. 2, pp. 149–163, 2004.

[7] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations.* Cambridge Univ. Press, 2010.

[8] L. B. de Oliveira and E. Camponogara, "Multi-agent model predictive control of signaling split in urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 120–139, 2010.

[9] A. L. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, pp. 342–375, 2009.

[10] M. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proc. of the 17th International Conf. on Machine Learning (ICML 2000)*, 2000, pp. 1151–1158.

[11] T. L. Thorpe and C. W. Anderson, "Traffic light control using SARSA with three state representations," IBM Corporation, Tech. Rep., 1996.

[12] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *ASCE Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, May 2003.

[13] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[14] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman, “Simulation and optimization of traffic in a city,” in *Proc. IEEE Intelligent Vehicle symposium (IV 2004)*, Parma, Italy, Jun. 2004, pp. 453–458.

[15] M. Steingröver, R. Schouten, S. Peelen, E. Nijhuis, and B. Bakker, “Reinforcement learning of traffic light controllers adapting to traffic congestion,” in *Proc. of the 17th Belgium-Netherlands Conference on Artificial Intelligence Conference (BNAIC 2005)*, Brussels, Belgium, Oct. 17–18, 2005, pp. 216–223.

[16] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, “Multiagent reinforcement learning for urban traffic control using coordination graphs,” *Machine Learning and Knowledge Discovery in Databases*, pp. 656–671, 2008.

[17] P. Lowrie, “SCATS: The Sydney Co-ordinated Adaptive Traffic System-principles, methodology, algorithms,” in *Proc. IEE International Conference on Road Traffic Signalling*, London, United Kingdom, 1982, pp. 67–70.

[18] P. Hunt, D. Robertson, R. Bretherton, and M. Royle, “The SCOOT on-line traffic signal optimisation technique,” *Traffic Eng. & Control*, vol. 23, no. 4, pp. 190–192, 1982.

[19] K. Fehon, “Adaptive traffic signals are we missing the boat?” *ITE District 6 Annual Meeting*, 2004.

[20] A. Salkham, R. Cunningham, A. Garg, and V. Cahill, “A collaborative reinforcement learning approach to urban traffic control optimization,” in *Proc. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, Australia, Dec. 2008, pp. 560–566.

[21] N. H. Gartner, “Opac: A demand-responsive strategy for traffic signal control,” *U.S. Dept. Transportation, Transportation Research Record 906*, 1983.

[22] P. Mirchandani and L. Head, “A real-time traffic signal control system: architecture, algorithms, and analysis,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, Dec. 2001.

[23] V. Dinopoulou, C. Diakaki, and M. Papageorgiou, “Applications of the urban traffic control strategy TUC,” *European Journal of Operational Research*, vol. 175, no. 3, pp. 1652–1665, 2006.

[24] A. D. Febbraro, D. Giglio, and N. Sacco, “Urban traffic control structure based on hybrid petri nets,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 224–237, Dec. 2004.

[25] G. F. List and M. Cetin, “Modeling traffic signal control using petri nets,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 3, pp. 177–187, Sep. 2004.

[26] S. Lin, B. D. Schutter, Y. Xi, and H. Hellendoorn, “Fast model predictive control for urban road networks via MILP,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 846–856, Sep. 2011.

[27] K. Rezaee, B. Abdulhai, and H. Abdelgawad, “Application of reinforcement learning with continuous state space to ramp metering in real-world conditions,” in *Proc. IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012)*, Anchorage, AK, Sep. 16–19, 2012, pp. 1590–1595.

[28] T. H. Heung, T. K. Ho, and Y. F. Fung, “Coordinated road-junction traffic control by dynamic programming,” *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 3, pp. 341–350, Sep. 2005.

[29] S. Sen and K. L. Head, “Controlled optimization of phases at an intersection,” *Transportation science*, vol. 31, no. 1, pp. 5–17, 1997.

[30] R. van Katwijk, "Multi-agent look-ahead traffic-adaptive control," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, 2008.

[31] S. El-Tantawy and B. Abdulhai, "Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)," in *Proc. IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012)*, Anchorage, AK, Sep. 16–19, 2012, pp. 319–326.

[32] J. C. Medina and R. F. Benekohal, "Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy," in *Proc. IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012)*, Anchorage, AK, Sep. 16–19, 2012, pp. 596–601.

[33] L. Kuyer, "Multiagent reinforcement learning and coordination for urban traffic control using coordination graphs and max-plus," Master's thesis, Intelligent Autonomous Systems group Faculteit der Natuurwetenschappen, Wiskunde en Informatica, Universiteit van Amsterdam, 2008.

[34] R. H. Smith and D. C. Chin, "Evaluation of an adaptive traffic control technique with underlying system changes," in *Proc. IEEE 27th Winter Simulation Conference (WSC 1995)*, Arlington, VA, Dec. 3–6, 1995, pp. 1124–1130.

[35] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 261–272, Sep. 2006.

[36] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 714–727, Sep. 2010.

[37] Y. Wenchen, Z. Lun, H. Zhaocheng, and Z. Lijian, "Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and paramics simulation," in *Proc.*

IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, Sep. 16–19, 2012, pp. 391–396.

[38] P. Lertworawanich, M. Kuwahara, and M. Miska, "A new multiobjective signal optimization for oversaturated networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 967–976, Dec. 2011.

[39] J. J. Sánchez-Medina, M. J. Galán-Moreno, and E. Rubio-Royo, "Traffic signal optimization in "La Almozara" district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 132–141, Mar. 2010.

[40] Z. Liu, "A survey of intelligence methods in urban traffic signal control," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 7, no. 7, pp. 105–112, 2007.

[41] C. P. Pappis and E. H. Mamdani, "A fuzzy logic controller for a traffic junction," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 10, pp. 707–717, 1977.

[42] Z. Liu and S. Li, "Immunity genetic algorithms based adaptive control method for urban traffic network signal," *Control Theory & Applications*, vol. 23, no. 1, pp. 119–125, 2006.

[43] Y. Wen and T. Wu, "Real-time rolling horizon optimization of urban traffic control based on ant algorithm," *Control and Decision*, vol. 19, pp. 1057–1059, 2004.

[44] D. Houli, L. Zhiheng, and Z. Yi, "Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network," *Journal on Advances in Signal Processing (EURASIP)*, vol. 2010, p. 7, 2010.

[45] M. A. Khamis, W. Gomaa, and H. El-Shishiny, "Multi-objective traffic light control system based on bayesian probability interpretation," in *Proc. IEEE 2012 Interna-*

tional Conference on Intelligent Transportation Systems (15th ITSC), Sep. 16–19, 2012, pp. 995–1000.

[46] M. Shoufeng, L. Ying, and L. Bao, "Agent-based learning control method for urban traffic signal of single intersection," *Journal of Systems Engineering*, vol. 17, no. 6, pp. 526–530, 2002.

[47] S. Richter, D. Aberdeen, and J. Yu, "Natural actor-critic for road traffic optimisation," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1169–1176, 2007.

[48] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[49] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412–421, Jun. 2011.

[50] S.-B. Cools, C. Gershenson, and B. DHooghe, "Self-organizing traffic lights: A realistic simulation," *Advances in Applied Self-Organizing Systems*, pp. 41–50, 2008.

[51] R. Schouten and M. Steingröver, "Reinforcement learning of traffic light controllers under partial observability," Master's thesis, Faculty of Science University of Amsterdam, Amsterdam, The Netherlands, Aug. 2007.

[52] G. D. Escobar, M. Pastorino, G. Brey, and M. Espinosa. (2004, Dec.) Intelligent Argentinean TRAffic COntrol System (IATRACOS). Sourceforge repository. [Online]. Available: http://sourceforge.net/projects/morevts/files/moreVTS/moreVTSv1.0/

[53] S. Faye, C. Chaudet, and I. Demeure, “A distributed algorithm for adaptive traffic lights control,” in *Proc. IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012)*, Anchorage, AK, Sep. 16–19, 2012, pp. 1572–1577.

[54] E. Yang and D. Gu, “Multiagent reinforcement learning for multi-robot systems: A survey,” Dept. of Computer Science, University of Essex, Tech. Rep., 2004.

[55] L. Buşoniu, R. Babuška, and B. De Schutter, “Multi-agent reinforcement learning: An overview,” *Innovations in Multi-Agent Systems and Applications-1*, pp. 183–221, 2010.

[56] M. J. Matarić, “Reinforcement learning in the multi-robot domain,” *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997.

[57] L. Panait and S. Luke, “Cooperative multi-agent learning: The state of the art,” *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[58] L. Goodwin, “Weather impacts on arterial traffic flow,” *Mitretek Systems Inc.*, 2002.

[59] M. Cools, E. Moons, and G. Wets, “Assessing the impact of weather on traffic intensity,” *Weather, Climate, and Society*, vol. 2, no. 1, pp. 60–68, 2010.

[60] H. Alhassan and J. Ben-Edigbe, “Highway capacity prediction in adverse weather,” *Journal of Applied Sciences*, vol. 11, pp. 2193–2199, 2011.

[61] R. Hoogendoorn, G. Tamminga, S. Hoogendoorn, and W. Daamen, “Longitudinal driving behavior under adverse weather conditions: adaptation effects, model performance and freeway capacity in case of fog,” in *Proc. IEEE 13th International Conference on Intelligent Transportation Systems (ITSC2010)*, 2010, pp. 450–455.

[62] M. A. Khamis and W. Gomaa, “Enhanced multiagent multi-objective reinforcement learning for urban traffic light control,” in *Proc. IEEE 11th International Conference*

on Machine Learning and Applications (ICMLA 2012), Boca Raton, Florida, Dec. 12–15, 2012, pp. 586–591.

[63] C. Brooks. (2003, May) Course of software agents and electronic commerce, project 3: Learning in games/business plan. proj3.pdf. [Online]. Available: http://www.cs.usfca.edu/~brooks/S03classes/cs486/

[64] J. Iša, J. Kooij, R. Koppejan, and L. Kuijer, "Reinforcement learning of traffic light controllers adapting to accidents," Design and organisation of autonomous systems, University of Amsterdam, 2006.

[65] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 3, pp. 397–415, May 2008.

[66] Z. Gábor, Z. Kalmár, and C. Szepesári, "Multi-criteria reinforcement learning," in *Proc. of the 15th International Conf. on Machine Learning (ICML 1998)*, Madison, Wisconsin, Jul. 24-27, 1998, pp. 197–205.

[67] S. Mannor and N. Shimkin, "A geometric approach to multi-criterion reinforcement learning," *Journal of Machine Learning Research*, vol. 5, pp. 325–360, 2004.

[68] S. Natarajan and P. Tadepalli, "Dynamic preferences in multi-criteria reinforcement learning," in *Proc. of the 22th International Conf. on Machine Learning (ICML 2005)*, Bonn, Germany, 2005.

[69] Google Map Maker User Guide. [Online]. Available: http://support.google.com/mapmaker/

[70] T. Maze, M. Agarwai, and G. Burchett, "Whether weather matters to traffic demand, traffic safety, and traffic operations and flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1948, pp. 170–176, 2006.

[71] C. Gershenson and D. A. Rosenblueth, “Modeling self-organizing traffic lights with elementary cellular automata,” Universidad Nacional Autónoma de México Ciudad Univ., Tech. Rep. Arxiv preprint arXiv:0907.1925, 2009.

[72] M. Barth and K. Boriboonsomsin, “Traffic congestion and greenhouse gases,” *TR News*, vol. 268, 2010.

[73] M. Treiber, A. Kesting, and C. Thiemann, “How much does traffic congestion increase fuel consumption and emissions? Applying a fuel consumption model to NGSIM trajectory data,” in *87th Annual Meeting of the Transportation Research Board, Washington, DC*, 2008.

[74] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* Cambridge Univ. Press, 1998, vol. 1, no. 1.

[75] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[76] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model MOBIL for car-following models,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1999, pp. 86–94, 2007.

Appendix A

Reinforcement Learning

A.1 Agent and Environment

An agent interacts with its environment at each discrete time step $t = 0, 1, 2, 3, \ldots$ where at each time step, the agent perceives some representation s_t of the current state of the environment where $s_t \in S$ and S is a finite set of all possible states. The agent then selects an appropriate action $a_t \in A$ where A is a finite set of all possible actions that can be taken by the agent. Once the agent takes an action, the agent perceives a new state $s + 1$ at the next time step $t + 1$ and receives a scalar reward r_{t+1}. The action at time t depends on what the agent has perceived so far (states and actions) and what it expects to perceive in the future. If s_t is the agent *perception* at time t, then in order to take an *optimal* action at time t, the agent exploits the *complete* history of states and actions up to time t. This process can be formulated as follows:

$$\pi(s_0, a_0, s_1, a_1, s_2, a_2, \ldots, s_t) = a_t \tag{A.1}$$

The function π is called the *policy* of the agent. This policy maps the *complete* history of states and actions up to time t to the *optimal* action at time t.

A.2 Markov Decision Process

If the current state includes all the relevant information from the *past*, then it is said to have the *Markov property*. Formally, if the current state depends on the *entire* history of events that led to it, then the probability of reaching some state s_{t+1} at time t depends on the *complete* history of dynamics until time t. Hence, this probability can be formulated as follows:

$$\Pr(s_{t+1}|s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, \ldots, s_0, a_0) \tag{A.2}$$

However, if the state is said to have the *Markov property* then:

$$\Pr(s_{t+1}|s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, \ldots, s_0, a_0) = \Pr(s_{t+1}|s_t, a_t) \tag{A.3}$$

The state s_t includes all the relevant information until time t. Therefore, the probability of reaching some state s_{t+1} depends only on the dynamics at time t. This allows the agent to predict the next state and also the expected reward it will receive in that state given only the *current* state and action.

A.3 Value Functions

An RL agent not only maximizes the *immediate* rewards but rather chooses actions that are *optimal* in the long-run. If an agent follows a policy π at state s, then the *expected accumulative reward* by following π is given by:

$$U^\pi(s) = E_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s], \tag{A.4}$$

where $E_\pi[.]$ is the *expectation* operator which averages over all rewards and stochastic transitions and $\gamma \in [0, 1]$ is a *discount rate* which ensures that the sum is *finite.*

Assuming an MDP, $U^{\pi}(s)$ can also be expressed in terms of *state transitions probabilities*. Equation A.4 can then be rewritten in a *recursive* fashion as:

$$U^{\pi}(s) = r + \gamma \sum_{s'} \Pr(s'|s,a) U^{\pi}(s'), \tag{A.5}$$

where r is the reward at state $s_0 = s$ (i.e., r_0 in Equation A.4). Equation A.5 is called the *Bellman equation* (for a complete proof, the reader is referred to [74]). Given the above, the *optimal utility* for an agent in state s is given by:

$$U^{*}(s) = \max_{\pi} U^{\pi}(s) = r + \gamma \max_{a} \sum_{s'} \Pr(s'|s,a) U^{*}(s') \tag{A.6}$$

Similarly, we can define an *optimal action-value function* $Q^{*}(s,a)$ which indicates the desirability of taking action a at state s as the *discounted future reward* the agent receives for taking action a at state s:

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{A.7}$$

$$Q^{\pi}(s,a) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^{t} r_t | s_0 = s, a_0 = a] \tag{A.8}$$

$$Q^{*}(s,a) = r + \gamma \sum_{s'} \Pr(s'|s,a) \max_{a'} Q^{*}(s',a') \tag{A.9}$$

A policy π which satisfies Equations A.6 and A.9 is said to be an *optimal policy* since the agent will obtain the *maximum reward* on the long-run by following it. Note that there can be a number of policies for a given task. However, all policies will share a *unique* $Q^{*}(s,a)$ and $U^{*}(s)$

A.4 Learning an Optimal Policy

A.4.1 Value Iteration

Value Iteration is a *model-based* method based on DP for computing the *optimal policy* when the *transition model* is available. We begin with random values $U(s)$ and iteratively repeat Equation A.6 which results in the following assignment operation:

$$U(s) := r(s) + \gamma \max_a \sum_{s'} \Pr(s'|s,a)U^*(s') \tag{A.10}$$

Hence, the Bellman equation is turned into an *update rul*e. The agent can then select the *optimal action* at each state according to a *greedy policy*:

$$\pi^*(s) = \arg\max_a \sum_{s'} \Pr(s'|s,a)U^*(s') \tag{A.11}$$

In case the transition model is known *a priori*, the agent can compute its *optimal policy* before actually taking an action. Otherwise, the agent must *experience* the world in order to learn the model.

A.4.2 Q-learning

Q-learning [75] is a *model-free* method in which the agent has *no access* to the *transition model.* In Q-learning, the agent estimates $Q^*(s,a)$ by *continuously* interacting with the environment in a form of *trial-and-error* process. Like in *Value Iteration*, the agent begins with random value estimates and after each action receives a tuple $\langle s, a, r, s' \rangle$ where s is the current state, a is the action taken at state s, r is the current reward, and s' is the resulting state after taking the action a.

For each tuple, the agent can estimate the corresponding *action-value* as:

$$Q(s,a) := (1-\lambda)Q(s,a) + \lambda[r + \gamma \max_{a'} Q(s',a')], \tag{A.12}$$

where $\lambda \in [0,1]$ is the agent *learning rate.* If all state-action pairs are visited *infinitely often* and the learning rate decreases over time, Q-learning has been shown to *converge* to the *optimal* $Q^*(s,a)$ with *probability 1.* The *optimal policy* is then given by:

$$\pi^*(s) = \max_{a} Q^*(s,a) \tag{A.13}$$

The material presented in this Appendix is prepared from [33].

APPENDIX A

Appendix B

MOBIL Lane Changing Model

In the GLD traffic simulation model [14], once a lane is selected, the vehicle cannot switch to a different lane. We added the MOBIL lane changing model [76] to the GLD traffic simulator. Lane changes depend on two criteria [76]:

1. *Safety criterion*: satisfied if the IDM braking deceleration of the back vehicle on the target lane B' after a possible change $acc' = acc'_{IDM}$ does not exceed a certain threshold b_{save}:
$$acc'(B') > -b_{save} \tag{B.1}$$

2. *Incentive criterion*: evaluated by weighting *my own advantage* on the target lane measured by the increased acceleration or decreased braking deceleration $acc'(M') - acc(M)$ against the *disadvantage of other drivers* $[acc(B) + acc(B')] - [acc'(B) + acc'(B')]$ weighted with a *politeness factor* p whose values are typically less than 1:
$$acc'(M') - acc(M) > p[acc(B) + acc(B') - acc'(B) - acc'(B')] + a_{thr} \tag{B.2}$$

APPENDIX B

An additional lane-changing threshold a_{thr} has been added to balance the above equation. Different human driving behaviors can be modeled by varying the politeness factor p [76]:

1. $p > 1$: a very humane behavior,
2. $p \in]0, 0.5]$: a realistic behavior such that the advantages of other drivers have a lower priority but are not neglected,
3. $p = 0$: a purely selfish behavior, and
4. $p < 0$: an aggressive personality who discomforts other drivers even at the cost of own disadvantage.

According to the GLD traffic simulation model, the vehicle will change lane under some restrictions:

1. The vehicle can go further to the same next lane scheduled by its *driving policy* as if it remains in the original lane.
2. The vehicle must stick to the road rules, e.g., left lane users will turn left, thus the vehicle cannot change to the right-turning lane even there exists an accident in the left-turning lane (also in order not to congest the right-turning lane as well).

List of Publications

- **Mohamed A. Khamis**, Walid Gomaa, Ahmed El-Mahdy, and Amin Shoukry, **Adaptive Traffic Control System Based on Bayesian Probability Interpretation**, in *Proc. of the 2012 IEEE Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC 2012)*, Alexandria, Egypt, March 6 - 9 , 2012 , pp. 151 - 156.
- **Mohamed A. Khamis**, Walid Gomaa, and Hisham El-Shishiny, **Multi-Objective Traffic Light Control System Based on Bayesian Probability Interpretation**, in *Proc. of the 15th IEEE Intelligent Transportation Systems Conference (ITSC 2012)*, Anchorage, Alaska, September 16-19, 2012, pp. 995-1000.
- **Mohamed A. Khamis** and Walid Gomaa, **Enhanced Multiagent Multi-Objective Reinforcement Learning for Urban Traffic Light Control**, in *Proc. of the 11th IEEE International Conference on Machine Learning and Applications (ICMLA 2012)*, Boca Raton, Florida, December 12-15, 2012, pp. 586-591.
- **Mohamed A. Khamis** and Walid Gomaa, **Adaptive Multi-Objective Reinforcement Learning with Hybrid Exploration for Traffic Signal Control Based on Cooperative Multi-Agent Framework**, Elsevier Engineering Applications of Artificial Intelligence (revision under review).

APPENDIX B

Declaration

I herewith declare that I have produced this work without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. The thesis work was conducted by Mohamed AbdElAziz Khamis Omar under the supervision of Dr. Walid Gomaa at the *Egypt Japan University of Science and Technology (E-JUST)* university.

Alexandria, Egypt,

December 1, 2013

Mohamed AbdElAziz Khamis Omar,