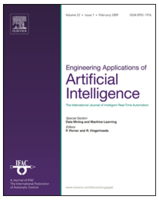
Engineering Applications of Artificial Intelligence 29 (2014) 134–151

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework

Mohamed A. Khamis [a,*], Walid Gomaa [a,b]

[a] Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology (E-JUST), P.O. Box 179, New Borg El-Arab City, Postal Code 21934 Alexandria, Egypt
[b] Alexandria University, Alexandria 21544, Egypt

ARTICLE INFO

Article history:
Received 30 July 2013
Received in revised form
4 December 2013
Accepted 9 January 2014
Available online 1 February 2014

Keywords:
Adaptive optimization
Multi-objective optimization
Reinforcement learning
Exploration
Traffic signal control
Cooperative multi-agent system

ABSTRACT

In this paper, 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. 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. 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. 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 $\simeq 8$ and 6 times lower respectively when using the multi-objective controller.

1. Introduction

In this paper, 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 schools 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 paper, 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

[*] Corresponding author. Tel.: +20 3 309 4075; Mobile: +20 100 638 2428; fax: +20 3 459 9520.
E-mail addresses: mohamed.khamis@ejust.edu.eg (M.A. Khamis), walid.gomaa@ejust.edu.eg (W. Gomaa).

0952-1976/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2014.01.007

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.

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, Pizam, 1999), 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 their behavior leading to a high rate of accidents, considerable time losses, and a high rate of vehicle emissions which severely affects the environment. Those problems have a considerable negative effect on the country economy. Thus, in this paper, 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 Section 7.

In 2010, traffic costs (based on time loss and fuel consumption) about $115 billion in the US based on 439 urban areas (Schrank et al., 2011). In the same year, 32,885 people died in accidents in the US (U.S. Department of Transportation, 2012). In Egypt, traffic problems are responsible for more than 25,000 accidents in 2010 with more than 6000 deaths per year (CAPMAS). Deaths per million driving kilometers in Egypt is about 34 times greater than in the developed countries (Abbas, 2004). This value is about 3 times greater than countries in the Middle East region (Abbas, 2004). The authors expect that this value is much worse in 2011–2013 due to the political upheaval in Egypt.

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 (Shoham and Leyton-Brown, 2010; De-Oliveira and Camponogara, 2010). 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) configuration is very promising (as proposed in Bazzan, 2009).

In the current paper, we adopt a MARL framework in a cooperation-based configuration to comply with the distributed nature and complexity of the problem. Our work is a significant extension of the framework developed by Wiering (2000) and Wiering et al. (2004). Wiering's controller, namely TC-1, represents a *pioneering* step in the use of real-time reinforcement learning framework in modeling traffic signal control. TC-1 outperforms traditional controllers (e.g., random, fixed time, longest queue, most cars). Moreover, TC-1 has proved its effectiveness and efficiency when being applied to large scale traffic networks. In contrast, other controllers based on reinforcement learning, e.g., Thorpe and Anderson (1996) and Abdulhai et al. (2003) suffer from exponential state-spaces when applied to large scale traffic networks. In addition, many latter researchers, e.g., Houli et al. (2010), Kuyer et al. (2008), Schouten and Steingröver (2007), Iša et al. (2006), and Steingröver et al. (2005), use TC-1 as a *benchmark* for performance evaluation. Each of these controllers contribute to TC-1 from a different prospective. For instance, in Schouten and Steingröver (2007), the authors overcome the *partial observability* of the traffic state-space, while we assume that the state-space is *fully-observable*, i.e., the agent can perfectly sense its environment.

Nevertheless, as will be explained latter, we tackle some problems in which TC-1 fails to adapt with. This includes: (1) stable adaptation to the limited-time congestion periods (using Bayesian probability interpretation), (2) advanced reward formulation to adapt with the continuous-time continuous-space simulation platform, and (3) using a multi-objective reward formulation in an additive manner to optimize multiple performance indices.

Moreover, we evaluate the performance of our proposed controller in comparison with two adaptive control strategies which are also based on AI methods: Self-Organizing Traffic Lights (SOTL) (Cools et al., 2008) (that outperforms a traditional green wave controller) and a Genetic Algorithm (GA) (Wiering et al., 2004).

Particularly, our objective in this paper is to develop a traffic control framework with the following characteristics: (1) inherently distributed through the use of a vehicle-based *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, and (5) higher confidence in the validity of the proposed traffic signal controller: through the use of a more realistic simulator as a testbed that is achieved by implementing the IDM acceleration model (Treiber et al., 2000) in the GLD vehicle traffic simulator (Wiering et al., 2004); 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 signal turns green. This behavior, on the other side, causes some kind of sign oscillation when being applied on the underlying RL model as will be shown later in Section 4 (which we called the *Zeno phenomenon*[1]) which results from the very slow acceleration of back vehicles when the traffic signal is just turning green.

Preliminary results of this work have been published in Khamis et al. (2012a,b) and Khamis and Gomaa (2012). In this paper, we provide a more detailed description and improvements on the multi-objective function. Such improvements boost the performance of the multi-objective controller, particularly when being compared to the

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

underlying single objective one. In addition, we present a novel cooperative hybrid exploration that is *more adaptive* to the changing dynamics in road conditions, and improves the trip waiting time of vehicles during transient periods. We also present a survey of the state-of-the-art work.

The remaining part of this paper is organized as follows. The related work of urban traffic signal controllers is discussed in Section 2. A background on the adopted traffic signal control and simulation models is presented in Section 3. The proposed framework including the improvements on the traffic signal control and simulation models is presented in Section 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 Section 5. Our multi-objective RL traffic signal control framework is discussed in Section 6. Section 7 presents the experiments conducted under this framework. This section includes the results of the experiments, discussion about these results, and how those results can be validated. Finally, Section 8 concludes the paper and proposes some directions for future work.

2. Related work

There have been several approaches proposed in the literature for traffic signal control. The two broad classes of these controllers are traditional control paradigms and adaptive control paradigms. On the one hand, the simplest intuitive type of traffic 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 (Febbraro et al., 2004; List and Cetin, 2004), Model Predictive Control (MPC) (De-Oliveira and Camponogara, 2010; Lin et al., 2011), 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 Rezaee et al. (2012), 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., Heung et al. (2005) and Sen and Head (1997). 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., Sen and Head (1997).

2.1. AI-based traffic signal controllers

Modern traffic signal controllers tend to be *more adaptive* to the current traffic conditions than traditional controllers (e.g., fixed-time controllers). 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 the 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, e.g., Smith and Chin (1995), Srinivasan et al. (2006), fuzzy logic, e.g., Gokulan and Srinivasan (2010), Wenchen et al. (2012), evolutionary algorithms, e.g., Lertworawanich et al. (2011), Sánchez-Medina et al. (2010). 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.

Generally, most of the previous works that are based on ML paradigms 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) and (2) a novel adaptive cooperative exploration technique. We discuss these approaches in detail in Section 5.

2.2. RL-based traffic signal controllers

The application of RL in the context of traffic signal control is pioneered by Thorpe and Anderson (1996). This approach is based on a State-Action-Reward State-Action (SARSA) RL algorithm. This approach 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 Steingröver et al. (2005), 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. Most RL-based traffic signal controllers proposed in the literature have junction-based full state representation (e.g., Abdulhai et al., 2003; El-Tantawy and Abdulhai, 2012; Medina and Benekohal, 2012). This suffers from the curse of dimensionality, the state-action space is estimated at the size of 10^{101} (as mentioned in Prashanth and Bhatnagar, 2011).

In our work, we adopted a different approach that is a *vehicle-based* state-space representation (Wiering, 2000). 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*.

In El-Tantawy and Abdulhai (2012), Medina and Benekohal (2012), 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 (Medina and Benekohal, 2012) uses the max-plus algorithm which is computationally demanding.

These approaches are based on *model-free* RL algorithms (e.g., SARSA, *Q*-learning) in which the learning process is not guided by a state transition probability model. Although less computations per traffic signal decision are 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 applications lacking real-time decision making, e.g., mine sweeping using robots. 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.

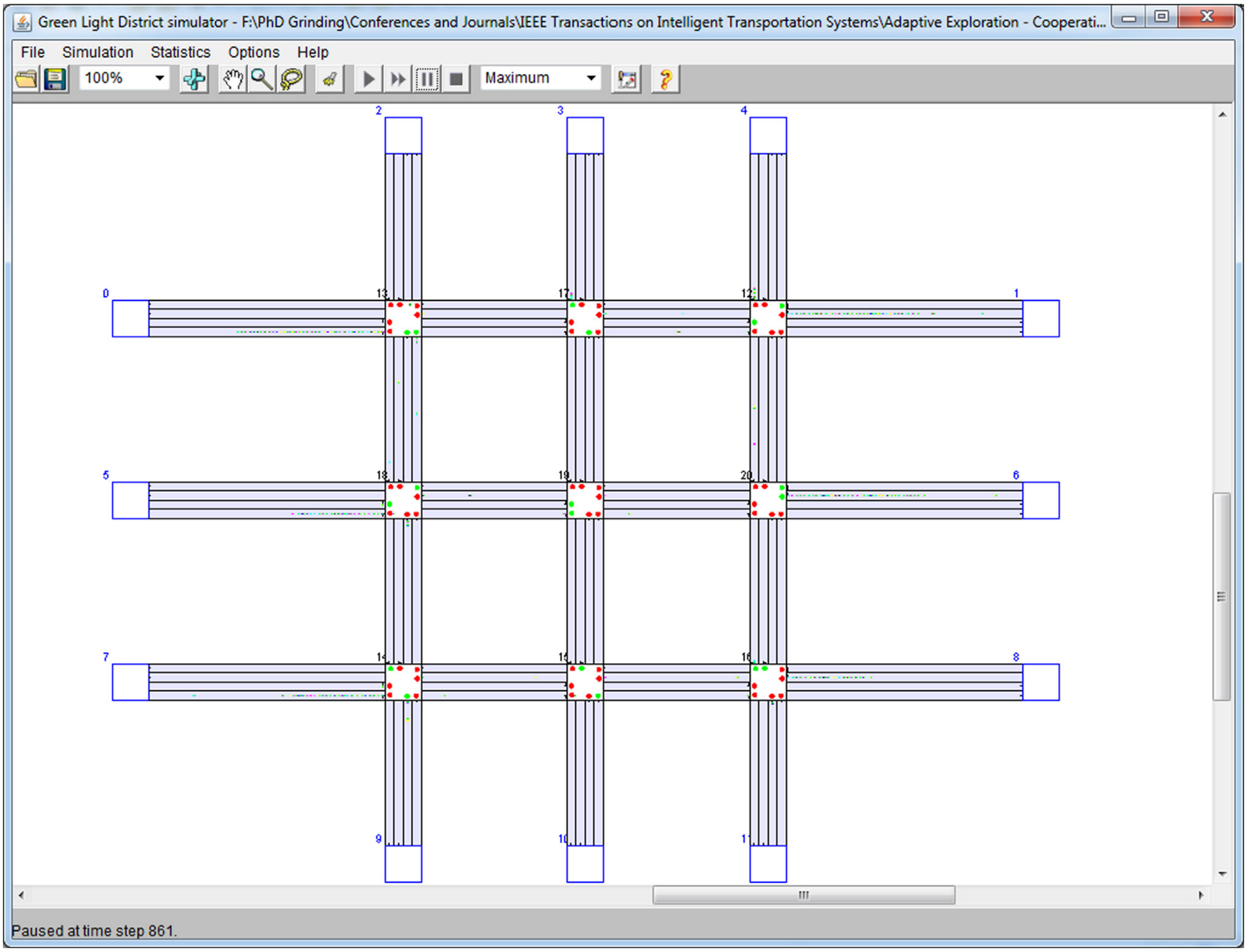
In this paper, we adopted the version of model-based RL presented in Wiering (2000). This particular version proves its effectiveness when being applied to large scale traffic networks.

In Kuyer et al. (2008), the authors extended Wiering RL model for traffic signal control (Wiering, 2000) 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 the neighboring junctions. However, as mentioned in El-Tantawy and Abdulhai (2012), 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 Salkham et al. (2008), 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 Richter et al. (2007), 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 Arel et al. (2010), a distributed traffic signal control method using ML-based neural networks has 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, neural networks are used for better searching the state-space.

2.3. Wiering-based traffic signal controllers

For testing and experimentation of our traffic signal control, we use the GLD traffic simulator (Wiering et al., 2004), see Fig. 1. The GLD simulator was initially based on a very simple discrete-time


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.

Fig. 1. GLD vehicle traffic simulator: traffic network with 12 edge nodes and 9 traffic signal nodes.

discrete-space model of traffic dynamics. Three previous extensions to the GLD traffic simulator have been implemented with simple acceleration models. The first extension is due to Cools et al. (2008) 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 (2007) that allows the vehicles to change their speed following either a Uniform or a Gaussian distribution. The third extension is due to Kuyer et al. (2008) who implement the same technique of Gaussian distribution 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. (2004) 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. (2005) 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. (2004). 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 speed of vehicles 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. (2008). 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 Section 4.

2.4. 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., Lertworawanich et al. (2011). On the one hand, the majority of these methods are based on either neuro-fuzzy or Multi-Objective Genetic Algorithms (MOGA). However, as mentioned in Faye et al. (2012), the use of fuzzy logic is not sufficient to represent the real-time traffic uncertainties. Also, neural networks and genetic algorithms require many computations and their parameters are difficult to be determined. In addition, as mentioned in Liu (2007), traffic signal control methods based on fuzzy logic are more suitable to control traffic at an isolated intersection. Also, evolutionary algorithms such as genetic algorithms will spend huge time to converge to the optimal traffic signal decision for large scale networks. On the other hand, some traffic signal controllers that are *junction-based*, e.g., Abdulhai et al. (2003) 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 Wiering (2000), the author 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 Steingröver et al. (2005), the proposed Q-function leads to an unusual adaptation of the real-time dynamic programming update in Eq. (1). In addition, the $Q(s,a)'s$ usually will not converge but instead keep oscillating between different values.

Houli et al. (2010) 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. (2005) 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 the contrary, in our model, the state-space representation is the same in size as the underlying traffic signal controller (Wiering, 2000). GAC (Steingröver et al., 2005) does not learn anything permanent about congestion, also this approach cannot be easily generalized. In Section 3, we present the underlying traffic signal control and simulation models for our multi-objective traffic signal control framework.

3. Background: traffic signal control and simulation models

3.1. Wiering RL traffic signal control model

In Khamis et al. (2012a,b) and Khamis and Gomaa (2012), we adopted the RL model developed by Wiering (2000) for traffic signal control. Each junction is controlled by an active[2] 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, and (3) the *destination* towards which the vehicle is traveling, denoted des. 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 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'. These probabilities are estimated based on the *frequentist* interpretation of probability: $\Pr(s,a,s') = C(s,a,s')/C(s,a)$ 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. In Khamis et al. (2012a), we used the *Bayesian* probability interpretation to estimate the parameters of these probabilities. This estimation was found to

[2] 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.

be more stable, robust, and *continuously adaptive* to the changing environment dynamics. We discuss this approach in Section 5.

The original model (Wiering, 2000) 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

$$Q(s,a)=\sum_{s'}\Pr(s,a,s')(R(s,a,s')+\gamma V(s')), \tag{1}$$

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 (Wiering, 2000), $R(s,a,s')=1$ in case the vehicle waits at the same position, otherwise equals 0. In Khamis et al. (2012b) and Khamis and Gomaa (2012), we proposed 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 Section 6.

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

$$V(s)=\sum_{a}\Pr(a|s)Q(s,a). \tag{2}$$

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[3] (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).

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 speed of vehicles, etc.

3.2. 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*. In our work, we chose to extend the moreVTS vehicle traffic simulator (Cools et al., 2008) that is based on the GLD traffic signal simulation platform (Wiering et al., 2004). This is due to the following reasons: (1) the GLD is a *widely used* open source traffic simulator, e.g., used by Cools et al. (2008), Steingröver et al. (2005), Kuyer et al. (2008), and Prashanth and Bhatnagar (2011), (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 that are already available in the GLD with the ability to add new performance indices, and (4) the visual ability to edit/create traffic networks and schedule traffic demands through a graphical interface, see Fig. 1.

4. Proposed framework

Despite the aforementioned capabilities of the GLD, it still contains severe drawbacks resulting from oversimplifications that we fixed in our previous work. We briefly mention our fixes here, and refer the reader to the original papers for more details (Khamis et al., 2012a,b; Khamis and Gomaa, 2012).

4.1. 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. In Khamis et al. (2012a), we implemented the more realistic IDM acceleration model (Treiber et al., 2000) that is used to simulate, in continuous-time and continuous-space, the acceleration and deceleration of vehicles. The vehicle acceleration $\mathrm{d}v/\mathrm{d}t$ depends on (1) the current velocity[4] v, (2) the distance to the front vehicle s, and (3) the difference in velocity Δv that is positive when approaching the front vehicle; the acceleration is given by

$$\frac{\mathrm{d}v}{\mathrm{d}t}=a\left[1-\left(\frac{v}{v_0}\right)^{\delta}-\left(\frac{s^*}{s}\right)^2\right],$$
$$s^*=s_0+\min\left[0,\left(vT+\frac{v\Delta v}{2\sqrt{ab}}\right)\right]. \tag{3}$$

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

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:

$$\text{speed}_{\text{new}}=\text{speed}_{\text{old}}+\text{acceleration}_{\text{IDM}}\times\delta t,$$
$$\text{position}_{\text{new}}=\text{position}_{\text{old}}-\text{speed}_{\text{new}}\times\delta t. \tag{4}$$

Note that in the GLD, the vehicles position values are decreasing as vehicles move from its source nodes towards the junctions. This clarifies the negative sign in the position update, Eq. (4). Afterwards, the simulator gathers all the needed statistics from the traffic network such as the average waiting time and the average queue length. 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. Afterwards, the traffic signal controllers decide on the new actions at all junctions

[3] 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.

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

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).

4.2. IDM impact on the RL traffic signal control model

In Khamis and Gomaa (2012), we analyzed and fixed some crucial problems that appeared in the original RL traffic signal control model (Wiering, 2000), particularly 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 a 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.

Another issue is the sign oscillation problem (a *Zeno* phenomenon) 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.[5]

4.3. Traffic demand probability distributions

The traffic demand in the GLD traffic simulation model (Wiering et al., 2004) is implemented by generating a uniform random number at every simulation time step and checking its value against a fixed traffic demand rate $\in [0,1]$. In order to allow for variability and non-stationarity, we have implemented in the GLD varying probability distributions of the inter-arrival times of the input vehicles in Khamis et al. (2012a).

4.4. Exploration policy

In the underlying traffic signal control model (Wiering, 2000), a random traffic signal configuration can be chosen with a small probability $\varepsilon = 0.01$ for the exploration of the state-action space. In Khamis and Gomaa (2012), we also used ε-exploration, though we found that it is better to start initially with a 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 $\varepsilon_t = \exp(-t/k_t)$ 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. In the current paper, 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 detail in Section 5.

4.5. 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 to be equal to 1 because the probability should be calculated and updated based on the *actually experienced* next states. Hence, this implementation is improper and in Khamis and Gomaa (2012) 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.

4.6. 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:

$$\begin{aligned}\text{Junction delay} = (&\text{Time step the vehicle crosses the junction}\\ &-\text{Time step the vehicle joins the junction lane})\\ &-(\text{Lane length}/\text{Lane maximum speed}). \qquad (5)\end{aligned}$$

In Khamis et al. (2012b), we defined the proper *junction waiting time* of a vehicle as follows:

$$\begin{aligned}\text{Junction waiting time} = &\text{ Time step the vehicle crosses the junction}\\ &-\text{Time step the vehicle joins the junction waiting queue}, \qquad (6)\end{aligned}$$

where joining the junction waiting queue is counted once the vehicle speed drops beyond a specific threshold, 0.36 km/h[6] (Khamis and Gomaa, 2012).

In Khamis et al. (2012b), we criticized the inefficiency of the GLD performance indices. The original *average trip waiting time* (ATWT) 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 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 Eq. (6). We call this policy the *co-learning* technique for calculating the performance indices. We have also implemented the *co-learning average trip time* (ATT). For more details and mathematical derivations of the co-learning performance indices, the reader is referred to Khamis et al. (2012b). Despite we have implemented as well the *co-learning average junction waiting time* (AJWT) version, 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* (Khamis et al., 2012b) as a more reasonable performance index. Moreover, we added the *relative throughput* performance index (Khamis et al., 2012b) in the GLD. This

[5] 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.

[6] 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).

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 (Khamis et al., 2012b). 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 (Khamis and Gomaa, 2012). 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 Houli et al., 2010), we added the *average number of vehicles trip stops* performance index (Khamis and Gomaa, 2012) to evaluate the performance of the fuel consumption objective. This performance index equals the sum of all vehicle stops in the whole trip divided by the number of arrived vehicles.

5. Handling traffic network non-stationarity

5.1. MDP parameters estimation using Bayesian probability interpretation

We use the Bayesian probability interpretation for estimating the unknown parameters of the MDP probabilities instead of the frequentist interpretation that was originally proposed in Wiering (2000). 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, etc.) in the congested periods. 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:

$$\Pr(A|B) = \Pr(B|A)\Pr(A)/\Pr(B). \tag{7}$$

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 $I_t = \{j \le t : \textit{state } s \textit{ is occupied at time } j\}$. For every $n = |I_t| \in \mathbb{N}$, define the Bernoulli random variable X_n as follows:

$$X_n = \begin{cases} 1, & a = RED \textit{ at time } k = \max I_t, \\ 0, & o.w., \end{cases} \tag{8}$$

That is $\overline{X}_n$ 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:

$$\text{Posterior}(n+1) = \frac{\text{Likelihood}(n+1)\text{Prior}(n+1)}{\text{Normalizing factor}(n+1)}. \tag{9}$$

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

$$\begin{aligned}\Pr(P_{n+1}|\overline{X}_{n+1}) &= \frac{\Pr(\overline{X}_{n+1}|P_{n+1})\Pr(P_{n+1})}{\Pr(\overline{X}_{n+1})} \\ &= \eta \Pr(\overline{X}_{n+1}|P_{n+1})\Pr(P_{n+1}|\overline{X}_n),\end{aligned} \tag{10}$$

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

$$\begin{aligned}\Pr(P_{n+1}|\overline{X}_{n+1}) &= \alpha \prod_{i=1}^{n+1} \Pr(\overline{X}_i|P_{n+1})\Pr(P_{n+1}|\overline{X}_0); \Pr(P_{n+1}|\overline{X}_0) = 1 \\ &= \alpha \prod_{i=1}^{n+1}\prod_{j=1}^{i} \Pr(X_j|P_{n+1}) = \alpha \prod_{i=1}^{n+1}\prod_{j=1}^{i} P_{n+1}^{X_j}(1-P_{n+1})^{(1-X_j)}.\end{aligned} \tag{11}$$

For an easier differentiation, we find

$$\begin{aligned}\ln \Pr(P_{n+1}|\overline{X}_{n+1}) &= \ln \alpha + \sum_{i=1}^{n+1} \ln\left(\prod_{j=1}^{i} P_{n+1}^{X_j}(1-P_{n+1})^{(1-X_j)}\right) \\ &= \ln \alpha + \sum_{i=1}^{n+1}\sum_{j=1}^{i} [X_j \ln P_{n+1} + (1-X_j)\ln(1-P_{n+1})].\end{aligned} \tag{12}$$

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

$$\begin{aligned}\frac{\partial \ln \Pr(P_{n+1}|\overline{X}_{n+1})}{\partial P_{n+1}} &= \sum_{i=1}^{n+1}\sum_{j=1}^{i}\left[\frac{X_j}{P_{n+1}} - \frac{(1-X_j)}{(1-P_{n+1})}\right] \\ &= \sum_{i=1}^{n+1}\sum_{j=1}^{i} X_j - \sum_{i=1}^{n+1}\sum_{j=1}^{i} P_{n+1} \\ &= \sum_{i=1}^{n+1}\sum_{j=1}^{i} X_j - \frac{P_{n+1}(n+1)(n+2)}{2} \\ &= 0.\end{aligned} \tag{13}$$

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

$$P_{n+1} = \frac{2}{(n+1)(n+2)} \sum_{i=1}^{n+1}\sum_{j=1}^{i} X_j. \tag{14}$$

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

$$P = \frac{2}{n(n+1)} \sum_{i=1}^{n}\sum_{j=1}^{i} X_j. \tag{15}$$

5.2. Adaptive cooperative hybrid exploration

In this paper, 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: $\exp(g_i)/\sum_{g_i}\exp(g_i)$, 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 that each junction decides whether to use the network-level "default" ε-greedy exploration (ε=0.01 as proposed in Wiering, 2000) 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 Khamis and Gomaa (2012) that depends on transferring the learned Q-values (with some decaying cooperation factor) from the ingoing lanes of a junction to the outgoing lanes. 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.

6. Multi-objective RL model for traffic signal control

As mentioned in Jin and Sendhoff (2008), little work has been done in multi-objective RL with some exceptions, e.g., Gábor et al. (1998), Mannor and Shimkin (2004), and Natarajan and Tadepalli (2005). Thus, the framework proposed in this paper 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 (Wiering, 2000), 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.

The proposed multi-objective function is given by

$$\begin{aligned}Q(s,a) = \sum_{s'}\Pr(s,a,s')[(R_{\text{ATWT}}(s,a,s')+R_{\text{ATT}}(s,a,s')+R_{\text{AJWT}}(s,a,s')\\ +CF(s,a,s')\times R_{\text{FR}}(s,a,s')+R_{\text{GW}}(s,a,s')\\ +R_{\text{AA}}(s,a,s')+R_{\text{MS}}(s,a,s'))+\gamma V(s')]. \qquad (16)\end{aligned}$$

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 second reward represents the ATT. In this paper, we improve the ATT reward function that we previously proposed in Khamis and Gomaa (2012) in order to better discriminate the reward values in case the traffic signal is red or green. 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: $R_{\text{ATT}}(s,a,s') = C_{\text{ATT}} \times (1-2^{-\Delta^2 p})$. In side roads (e.g., residential areas in which the main objective is to *avoid accidents*), the controller uses a *weaker* ATT reward function: $C_{\text{ATT}} \times (1-2^{-\Delta p})$. 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 $R_{\text{AJWT}}(s, green, s') = 0$ 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 $R_{\text{FR}}(s, green, s') = 10$ in case $tl' \neq tl$, otherwise equals 0. Assume that the number of blocks[7] taken by the waiting vehicles in the next lane[8] 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 (Houli et al., 2010)

$$CF(s, green, s') = \begin{cases} 0 & \text{if } W \le \theta, \\ 10\times(W-\theta) & \text{if } \theta < W \le 1, \\ 2 & \text{if } W > 1. \end{cases} \qquad (17)$$

$\theta$is a threshold whose best value equals 0.8 (as mentioned in Steingröver et al., 2005). For instance, if $N=9$ m and $L=10$ m, then $CF(s, green, s') = 1$ (the traffic signal controller tries 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 $Q(s, red)-Q(s, green)$) 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*, and (3) the number of vehicles within distance ω from the traffic junction is $\in [1, \mu]$, then $R_{\text{GW}}(s, green, s') = -10$, otherwise equals 0. The best parameters values are $\omega = 25$ m (as proposed in Cools et al., 2008) and $\mu = 3$ vehicles. Unlike the original RL model (Wiering, 2000) 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 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). In this paper, we improve the safety reward function that we previously proposed in Khamis and Gomaa (2012) in order to better discriminate the reward values in case the traffic signal is red or green. 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: $R_{\text{AA}}(s,a,s') = C_{\text{AA}} \times (1/(\Delta^2 p+1))$. The best value of the short distance e beyond the traffic junction is 10 m (as proposed in Gershenson and Rosenblueth, 2009). In residential and schools areas, our controller alleviates driver's aggressiveness by using the following AA reward function: $C_{\text{AA}} \times (1/(\Delta p+1))$. 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

[7] Like moreVTS (Cools et al., 2008), we set 1 block = 1 m.

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

residential and schools areas). Note that in the simulation environment, the IDM acceleration model is a *collision-free* model (Treiber et al., 2000). Thus, we cannot measure *efficiently* the performance of the AA objective, e.g., by using the *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. In this paper, we improve the fuel consumption reward function that we previously proposed in Khamis and Gomaa (2012) in order to better discriminate the reward values in case the traffic signal is red or green. 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 $R_{\text{MS}}(s,a,s')$ to C_{MS} or $-C_{\text{MS}}$, otherwise equals 0. C_{MS} equals 10 or -10 in case the traffic signal is red or green respectively.

7. Experimentation

7.1. Symmetric network: horizontal main roads with vertical side roads

We use the traffic network in Fig. 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) each of length equals 1120 m (2 entry links each of 300 m, 2 links between intersections each of 200 m, and 3 junctions each of 40 m) and the 3 vertical roads are the side roads each of length equals 920 m (2 entry links each of 200 m, 2 links between intersections each of 200 m, and 3 junctions each of 40 m). The road lengths and generation rates are chosen to simulate a high congestion in main roads with less traffic in side roads. This setting is made to show how the proposed traffic signal controller can tackle the possible long waiting vehicles in 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 s. The results of this experiment are averaged over 10 independent runs. Every run has a seed equals its starting computer clock time (in milliseconds) and consists of 100,000 time steps which are about 400 min.

As mentioned in Prashanth and Bhatnagar (2011), 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[9]) 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. For more details about the impact of weather conditions on the IDM acceleration model parameters, we refer the reader to Khamis et al. (2012b). We set the speed limit of the main and side roads to 60 km/h and 55 km/h respectively. For more details about the labeled roads and their characteristics, we refer the reader to Khamis and Gomaa (2012).

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 min 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 min (this corresponds to *extremely high* 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.

7.2. Results

Figs. 2–9 compare the performance of the multi-objective controller (using the Bayesian probability estimation) with hybrid exploration based on the *transient state* of the current and neighboring junctions (i.e., cooperation-based) versus the TC-1 controller (Wiering, 2000) (single objective with frequentist probability estimation using ε-exploration). The former controller is represented by blue long dashes, while the latter controller is represented by red dashes. Note that the achievements 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) (Cools et al., 2008) and a Genetic Algorithm (GA) (Wiering et al., 2004). 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 ($\phi_{\min}=5$ s). 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 m 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 $\in[1,\mu=3$ vehicles]) within a distance $\omega=25$ m 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, percentage of arrived vehicles, percentage of rejected vehicles (indicating network utilization), and the maximum queue length.

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*, Fig. 2, and the *co-learning ATWT*, Fig. 3, the mean values are $\simeq 8$ and 6 times lower respectively when using the multi-objective controller. Figs. 2 and 3[10] show that the multi-objective controller has much more stable response to the changing dynamics (occurring every 100 min). The response of the single objective controller to the transient periods is severe. Fig. 4 shows that the *average speed*

[9] This rate complies with the vehicles rate of Wetstraat at normal congestion periods which is provided by the *Ministry of the Brussels-Capital region* (Cools et al., 2008).

[10] 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.

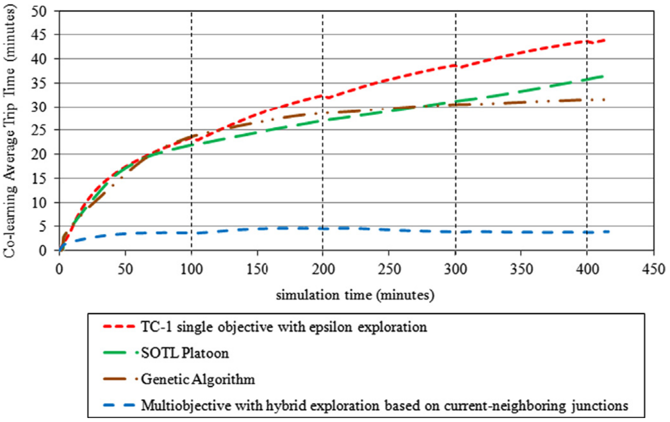


Fig. 2. Co-learning average trip time. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

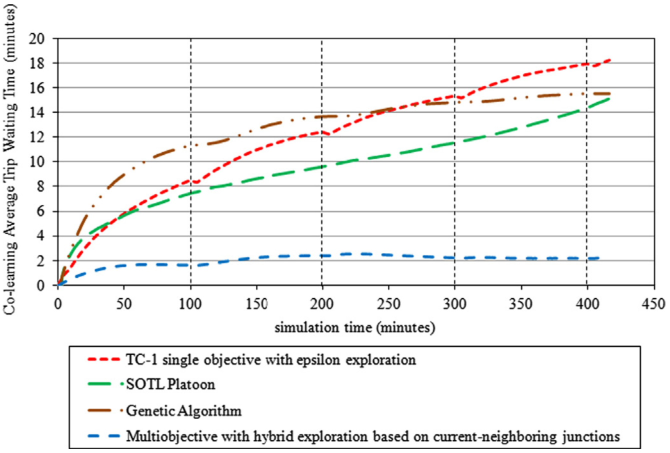


Fig. 3. Co-learning average trip waiting time. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

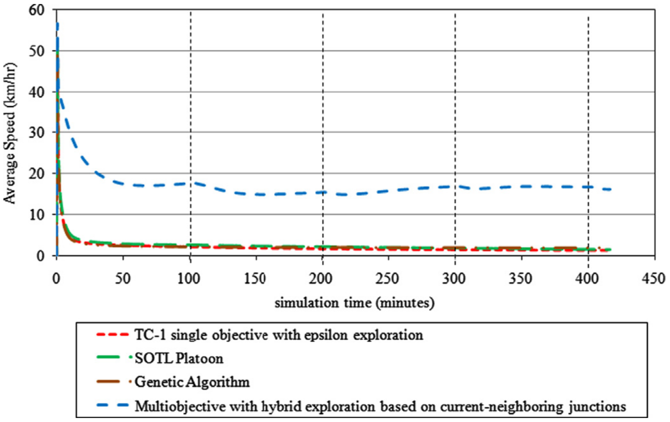


Fig. 4. Average speed. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

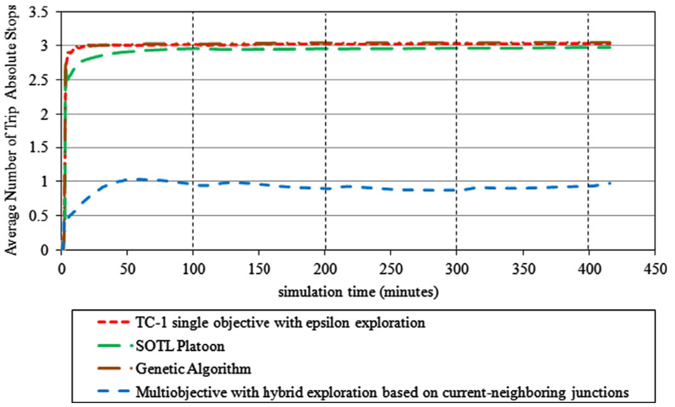


Fig. 5. Average number of trip absolute stops. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

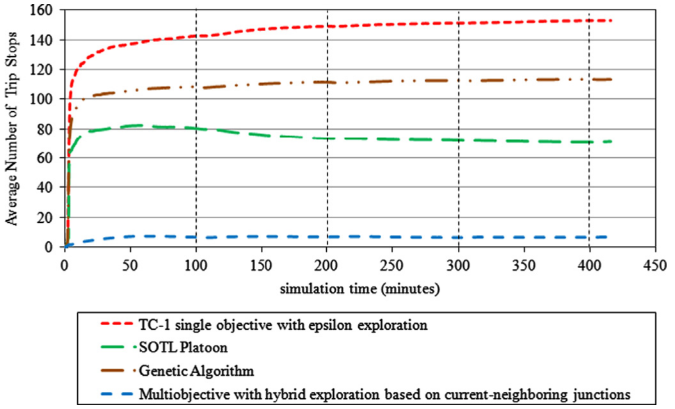


Fig. 6. Average number of trip stops. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

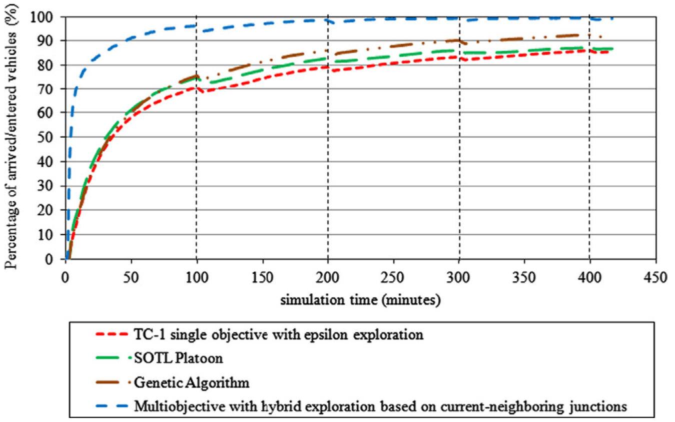


Fig. 7. Percentage of arrived to entered vehicles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

of vehicles is $\simeq 8$ times higher when using the multi-objective controller. This means lower congestion and faster arrival to destinations (that increases the driver′s satisfaction). Fig. 5 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*. Fig. 6 shows that the *vehicle stops* are $\simeq 22$ times lower when using the multi-objective controller. This will save fuel consumption and consequently is more environment friendly. Moreover, the *number of vehicle stops* can also be considered as a good measure of the *total delays* that encounter vehicles.

Fig. 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.

Fig. 8 shows that the *rejected vehicles percentage relative to all generated vehicles* (i.e., generated but cannot join the network due

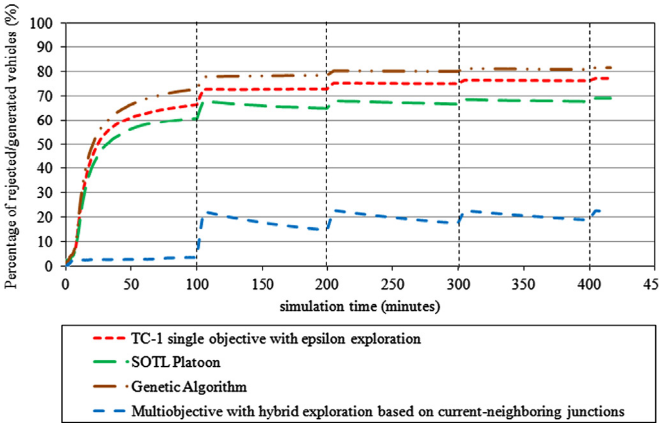


Fig. 8. Percentage of rejected to generated vehicles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

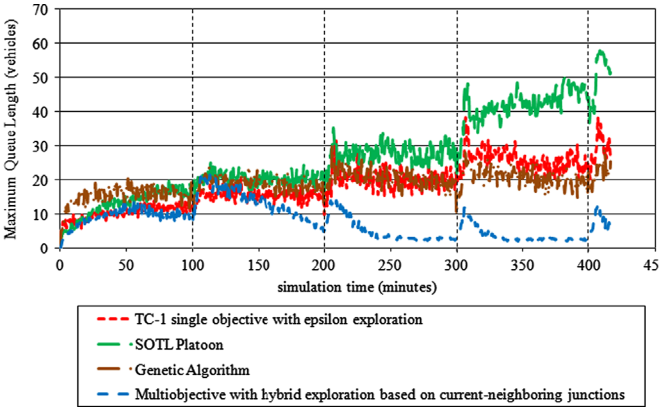


Fig. 9. Maximum queue length. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

to overfull ingoing lanes) is $\simeq 4$ times lower when using the multi-objective controller. This performance index is a good indicator of the *network congestion*, and accordingly the *network utilization*.

Fig. 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).

We use the *co-learning ATWT* performance index in order to show the impact of using the cooperative hybrid exploration (discussed in Section 5) on the long waiting times of vehicles in side roads when using the ε-exploration solely. Fig. 10 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 lower by $\simeq 10\%$ than the multi-objective controller using ε-exploration.

7.3. Validation

In order to better realize the contributions presented in this paper, here we give some insights about how the results presented in this paper can be validated. Firstly, the mathematical model of estimating the parameters of the MDP based on the Bayesian probability interpretation presented in Section 5 represents one sort of system validation. In Eq. (15), 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.

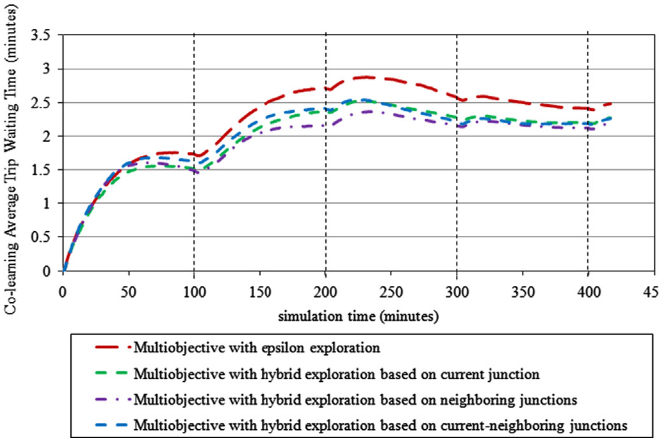


Fig. 10. Co-learning ATWT of the multi-objective controller with hybrid exploration transient state based on current junction, neighboring junctions, current–neighboring junctions, and with ε-exploration.

Secondly, the mathematical model of the accumulated reward (Q-function) formulation (Eq. (16)) 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.

Thirdly, comparing the performance of our multi-objective traffic signal controller with the *theoretically optimum* solution may be *computationally prohibitive*. Our multi-objective traffic signal controller is mainly based on *online decision making*. Whereas, it is *computationally demanding* to compute the *theoretical optimum* solution at every time step, e.g., using Little's law of *Queueing Theory* which may ignore some traffic related characteristics, e.g., the speed of vehicles and the inter-dependability between consecutive junctions, etc.. However, we can simply say that the *theoretical optimum* ATWT is *zero*. In addition, the *theoretical optimum* ATT can be calculated from the optimum average speed in main roads (equals 70 km/h $\simeq 20$ m/s) 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, Fig. 5. Thus, for the traffic network in Fig. 1, the *theoretical optimum* ATT equals 1120 m $\div$ 20 m/s = 56 s $\simeq 1$ min. In comparison with the performance of our multi-objective traffic signal controller (the mean value of ATT $\simeq 4$ min and the mean value of ATWT $\simeq 2$ min) considering the *dramatic change* in the traffic demand every 100 min; our traffic signal controller yields very good results.

Finally, 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 it yields lower fuel consumption especially when being compared to the performance of other controllers, Fig. 4. 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 Fig. 2 (i.e., $\simeq 4$ min); given that

Table 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 average traveled distance is 1120 m as mentioned previously. This yields some kind of validation for the presented results.

7.4. Discussion

The proposed multi-objective traffic signal controller does not overshoot at all in transient periods in Figs. 2 and 3. This is due to the triple effect of: (1) the reward function of the ATWT tackles the *Zeno* phenomenon discussed in Section 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 a function in the road type as discussed in Section 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 Section 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, and (3) using the novel adaptive cooperative exploration technique (discussed in Section 5) in which the impact of any transient period is propagated between the neighboring junctions to avoid very long waiting times 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), we rather 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 our 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 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 Jin and Sendhoff, 2008). 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.

Another issue worth discussing 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.

7.5. City center network: competing demands with non-parallel arterials

In this section, we apply the proposed traffic signal controller on a different *traffic pattern*, Fig. 11 (non-symmetric network with *competing demands* and not only parallel arterials). This traffic network complies with the city center traffic network presented in Wiering et al. (2004). The inner edge-nodes represent a city center.

The settings of this traffic network are similar to those in the main scenario traffic network, Fig. 1; the number of lanes in each direction, the length and the number of passengers of each vehicle, γ discount factor, and the duration of each simulation time step.

The horizontal and vertical roads highlighted by green are the main roads. The results of this experiment are averaged over 10 independent runs. Every run has a seed equals its starting computer clock time (in milliseconds) and consists of 50,000 time steps which are about 200 min.

The generation and destination rates are chosen to simulate *competing demands* where the default generation rate of all edge nodes is set to 0.01 (144 vehicles per hour). The default weather condition is set to *light fog* and the IDM *desired velocity* parameter v_0 to 90 km/h. We schedule the destination rate of every edge node to be equiprobable to the rest of the edge nodes, i.e., equals 1/8.

In order to simulate the *transient periods* at *normal* congestion periods (e.g., road users going to and leaving from the work), the traffic demand is changed every 100 min where the distribution of the inter-arrival time is set to 0.04 (576 vehicles per hour) continued for a period of 5 min. In these periods, we set the weather condition to *normal rain* and the IDM *desired velocity* parameter v_0 to 108 km/h. Dashed vertical lines clarify times at which changes occur in dynamics.

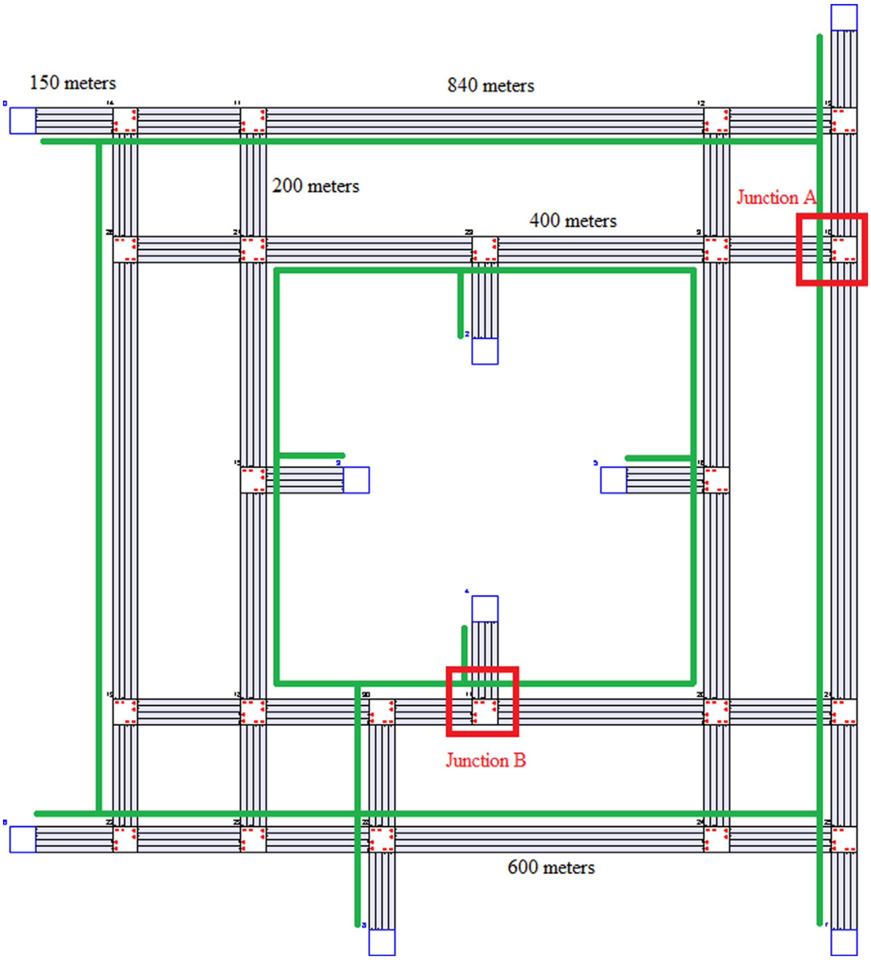


Fig. 11. Traffic network with 9 edge nodes and 22 traffic signal nodes – City center. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Figs. 12–15 show that the *average junction waiting time* (*AJWT*), the *average number of trip stops*, the *percentage of rejected to generated vehicles*, and the *average number of vehicles waiting at any junction* are better when using the multi-objective controller compared to the other controllers.

7.6. *Average green light percentage*

In order to determine how the proposed traffic signal controller deals with congestion specifically and how it behaves generally, we added a new performance index to the GLD traffic simulator, that is the *average green light percentage*[11] at each junction. This performance index represents the percentage of time that a specific *traffic light configuration*[12] at a specific junction is *green*. For space limitations, we mention here the *traffic signal operation* at two junctions only (A and B), Fig. 16, that are highlighted by red boxes in Fig. 11.

Figs. 17–20 show that the *average green light percentage* at junction A using the proposed multi-objective controller is better than the other controllers. For instance, the proposed controller, Fig. 17, responds effectively to the *transient periods* (occurring every 100 min) where it gives larger green time percentage to the *critical* configuration, i.e., the second configuration towards the city center. Nevertheless, at the same time, the proposed controller gives good chances to the vehicles in the other signal configurations to cross junction A. On the one hand, the TC-1 controller, Fig. 18, and the ACGJ-1 controller, Fig. 19, do not make the *sufficient distinction* to the *critical* configuration especially at *transient periods*. On the other hand, the SOTL controller, Fig. 20, over-discriminates the *critical* configuration, however, it almost blocks the third traffic signal configuration.

Figs. 21–24 show that the *average green light percentage* at junction B using the proposed multi-objective controller is better than the other controllers. For instance, the proposed controller, Fig. 21, responds effectively to the *transient periods* (occurring every 100 min) where it prioritizes the traffic signal configurations according to their directions to/from the city center with the following order: (1) the *third* configuration (the vehicles most probably are entering the city center), (2) the *first* configuration (the vehicles are leaving the city center at rush hours), and then (3) the *second* configuration (the vehicles may enter, leave, or move around the city center). The TC-1 controller, Fig. 22, the ACGJ-1 controller, Fig. 23, and the SOTL controller, Fig. 24, do not make such prioritization to the possible traffic signal configurations.

8. Conclusions and future work

8.1. *Conclusions*

In this paper, we present an adaptive multi-objective reinforcement learning system for traffic signal control based on a

[11] The *average* is considered due to sampling during the 50,000 time steps.

[12] Recall that the *traffic light configurations* represent the *consistent green lights* on all directions of a junction that do not cause any possible accidents between the crossing vehicles.

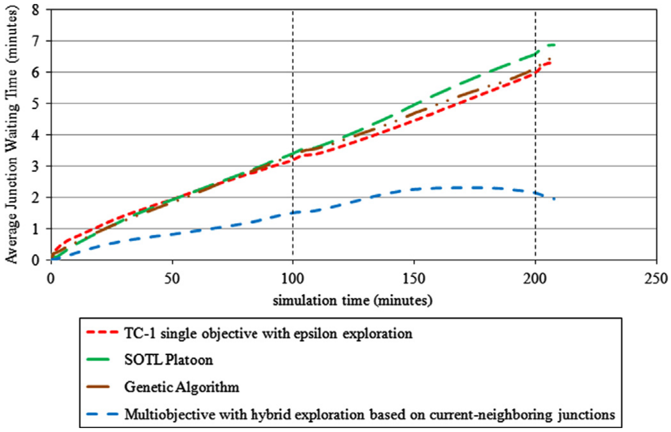


Fig. 12. Average junction waiting time – City center.

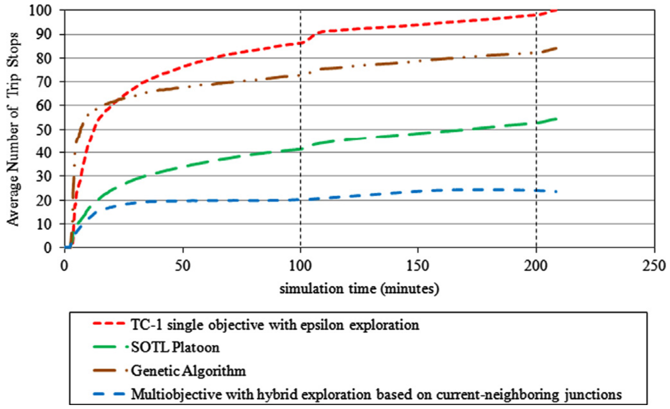


Fig. 13. Average number of trip stops – City center.

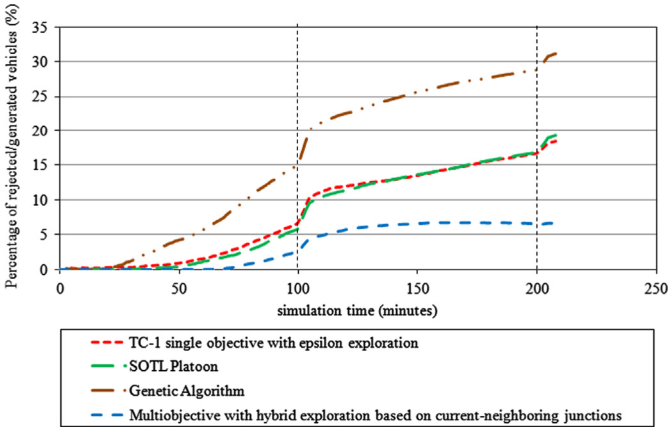


Fig. 14. Percentage of rejected to generated vehicles – City center.

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 is simulated by changing the traffic flow and the 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.)

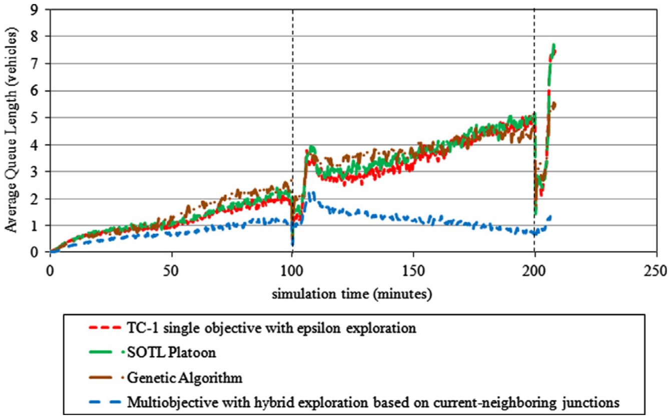


Fig. 15. Average queue length – City center.

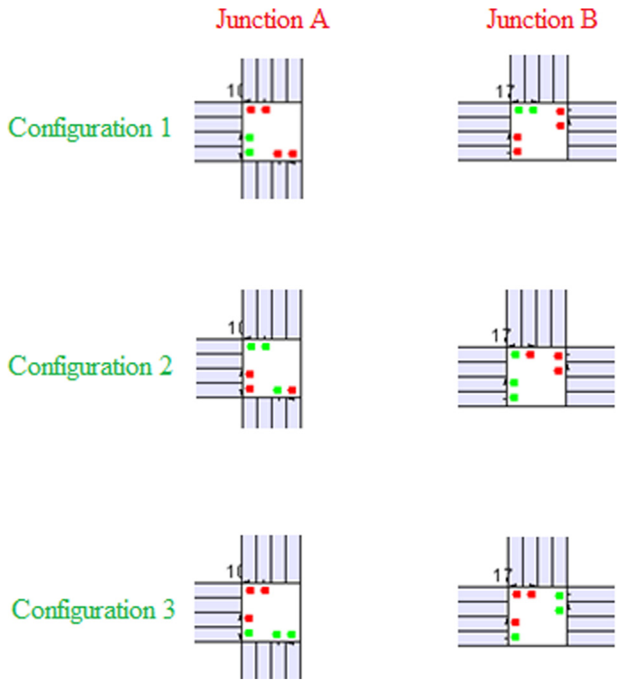


Fig. 16. Traffic signal operation at two junctions (A and B).

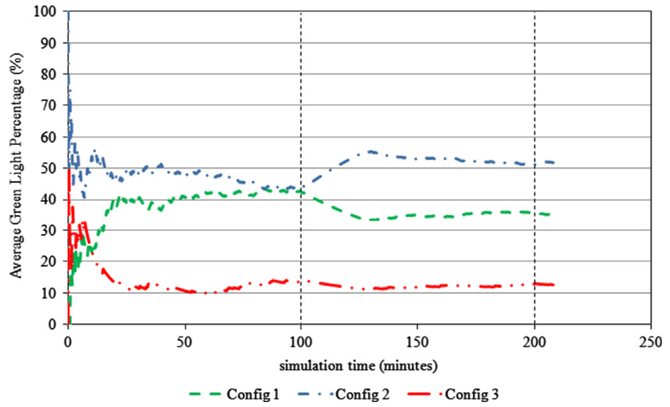


Fig. 17. Average green light percentage at junction A using the multi-objective controller.

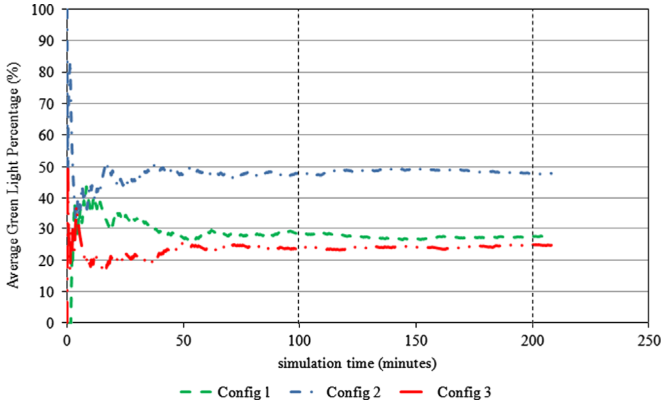


Fig. 18. Average green light percentage at junction A using the TC-1 single-objective controller.

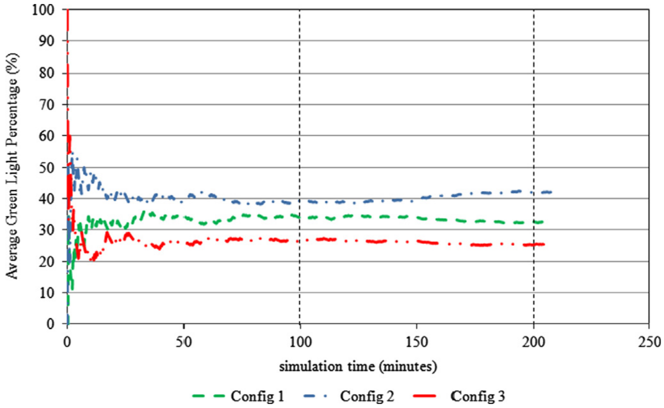


Fig. 19. Average green light percentage at junction A using the ACGJ-1 genetic-based controller.

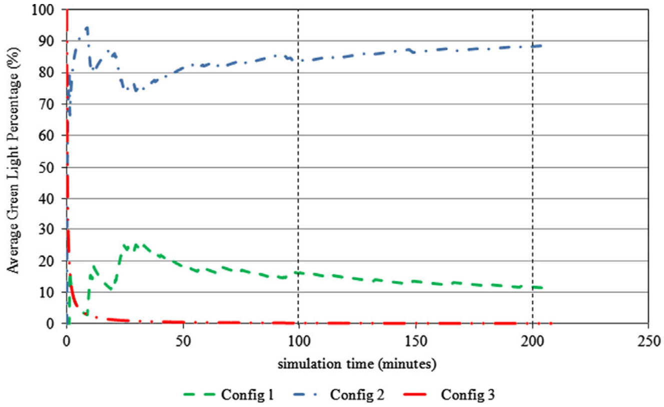


Fig. 20. Average green light percentage at junction A using the SOTL rule-based controller.

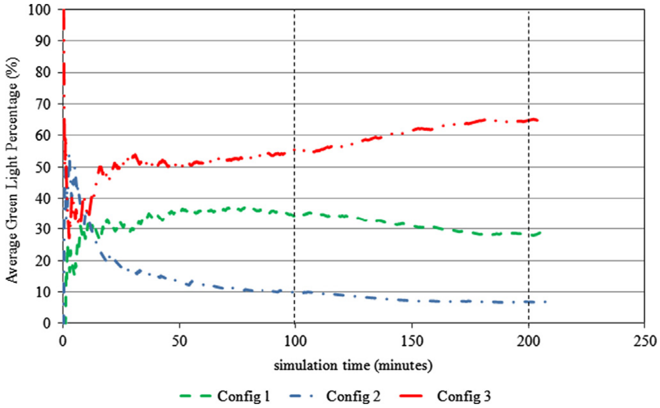


Fig. 21. Average green light percentage at junction B using the multi-objective controller.

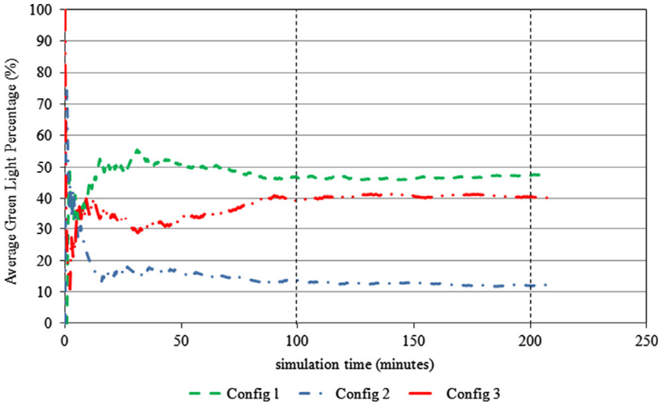


Fig. 22. Average green light percentage at junction B using the TC-1 single-objective controller.

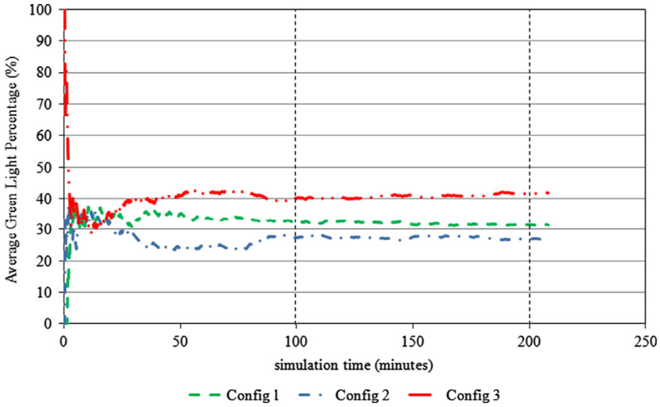


Fig. 23. Average green light percentage at junction B using the ACGJ-1 genetic-based controller.

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 $\simeq 8$ and 6 times lower respectively when using the multi-objective controller.

Finally, we show that the proposed traffic signal controller outperforms other controllers using the city center *traffic pattern with competing demands*. This traffic network is unlike the *typical* traffic pattern of main arterials and side roads (that leads to a *main road domination*) where the proposed traffic signal controller optimally behaves.

8.2. Future work

The work presented in this paper opens the door to a bulk of future work. For better organizing the suggested directions for future

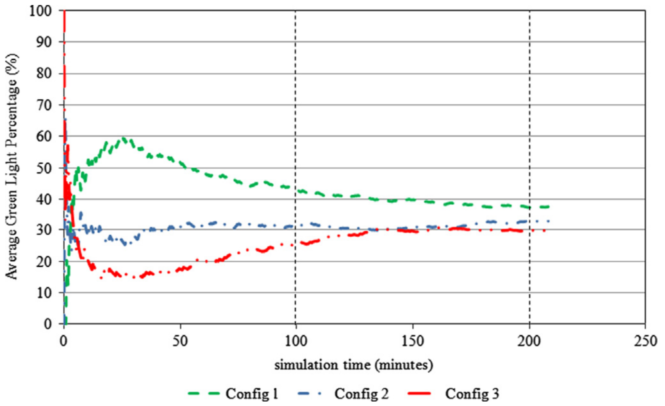


Fig. 24. Average green light percentage at junction B using the SOTL rule-based controller.

work, we categorize our ideas into future work in traffic signal control model and future work in traffic signal simulation model.

8.2.1. *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 Schouten and Steingröver (2007), 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.

Fourthly, 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).

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).

8.2.2. *Traffic signal simulation model*

Firstly, 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 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 3*D traffic simulator* that allows for *human drivers* as well as *agent vehicles*. This simulator has been developed as a collaboration of our research team with the Prendinger Laboratory in the National Institute of Informatics (NII), Tokyo, Japan.

Acknowledgment

This work is mainly supported by IBM PhD Fellowship and E-JUST Research Fellowship, and partially by Pharco Pharmaceuticals Corporation grant. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve this paper. Special thanks are due to Prof. Ahmed El-Mahdy, the director of the Smart City Facility at the Egypt-Japan University of Science and Technology (E-JUST), for fruitful discussions.

References

Abbas, K.A., 2004. Traffic safety assessment and development of predictive models for accidents on rural roads in Egypt. Accid. Anal. Prev. 36, 149–163.

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. ASCE J. Transp. Eng. 129, 278–285.

Arel, I., Liu, C., Urbanik, T., Kohls, A.G., 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4, 128–135.

Bazzan, A.L., 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Auton. Agents Multi-Agent Syst. 18, 342–375.

CAPMAS, 2010. Egypt Central Agency for Public Mobilization And Statistics (CAPMAS). (last accessed at 12 January 2013).

Cools, S.B., Gershenson, C., D′Hooghe, B., 2008. Self-organizing traffic lights: a realistic simulation. In: Advances in Applied Self-Organizing Systems, pp. 41–50.

De-Oliveira, L.B., Camponogara, E., 2010. Multi-agent model predictive control of signaling split in urban traffic networks. Transp. Res. Part C: Emerg. Technol. 18, 120–139.

El-Tantawy, S., Abdulhai, B., 2012. Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC). In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 319–326.

Escobar, G.D., Pastorino, M., Brey, G., Espinosa, M., 2004. Intelligent Argentinean TRAffic COntrol System (IATRACOS). Sourceforge repository.

Faye, S., Chaudet, C., Demeure, I., 2012. A distributed algorithm for adaptive traffic lights control. In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 1572–1577.

Febbraro, A.D., Giglio, D., Sacco, N., 2004. Urban traffic control structure based on hybrid petri nets. IEEE Trans. Intell. Transp. Syst. 5, 224–237.

Gábor, Z., Kalmár, Z., Szepesári, C., 1998. Multi-criteria reinforcement learning. In: Proceedings of the 15th International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, pp. 197–205.

Gershenson, C., Rosenblueth, D.A., 2009. Modeling Self-organizing Traffic Lights with Elementary Cellular Automata. Technical Report. Universidad Nacional Autónoma de México Ciudad University. Arxiv preprint arXiv:0907.1925.

Gokulan, B.P., Srinivasan, D., 2010. Distributed geometric fuzzy multiagent urban traffic signal control. IEEE Trans. Intell. Transp. Syst. 11, 714–727.

Heung, T.H., Ho, T.K., Fung, Y.F., 2005. Coordinated road-junction traffic control by dynamic programming. IEEE Trans. Intell. Transp. Syst. 6, 341–350.

Houli, D., Zhiheng, L., Yi, Z., 2010. Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network. J. Adv. Signal Process. (EURASIP), 7 pp.

Iša, J., Kooij, J., Koppejan, R., Kuijer, L., 2006. DOAS 2006 Project: Reinforcement Learning of Traffic Light Controllers Adapting to Accidents. Technical Report. Intelligent Autonomous Systems group, Informatics Institute, University of Amsterdam. Amsterdam, The Netherlands.

Jin, Y., Sendhoff, B., 2008. Pareto-based multiobjective machine learning: an overview and case studies. IEEE Trans. Syst. Man Cybern. C 38, 397–415.

Khamis, M.A., Gomaa, W., 2012. Enhanced multiagent multi-objective reinforcement learning for urban traffic light control. In: Proceedings of the IEEE 11th

International Conference on Machine Learning and Applications (ICMLA 2012), Boca Raton, FL, pp. 586–591.

Khamis, M.A., Gomaa, W., El-Mahdy, A., Shoukry, A., 2012a. Adaptive traffic control system based on Bayesian probability interpretation. In: Proceedings of the IEEE 2012 Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC 2012), Alexandria, Egypt, pp. 151–156.

Khamis, M.A., Gomaa, W., El-Shishiny, H., 2012b. Multi-objective traffic light control system based on Bayesian probability interpretation. In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 995–1000.

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

Lertworawanich, P., Kuwahara, M., Miska, M., 2011. A new multiobjective signal optimization for oversaturated networks. IEEE Trans. Intell. Transp. Syst. 12, 967–976.

Lin, S., Schutter, B.D., Xi, Y., Hellendoorn, H., 2011. Fast model predictive control for urban road networks via MILP. IEEE Trans. Intell. Transp. Syst. 12, 846–856.

List, G.F., Cetin, M., 2004. Modeling traffic signal control using petri nets. IEEE Trans. Intell. Transp. Syst. 5, 177–187.

Liu, Z., 2007. A survey of intelligence methods in urban traffic signal control. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) 7, 105–112.

Mannor, S., Shimkin, N., 2004. A geometric approach to multi-criterion reinforcement learning. J. Mach. Learn. Res. 5, 325–360.

Medina, J.C., Benekohal, R.F., 2012. Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy. In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 596–601.

Natarajan, S., Tadepalli, P., 2005. Dynamic preferences in multi-criteria reinforcement learning. In: Proceedings of the 22th International Conference on Machine Learning (ICML 2005), Bonn, Germany.

Pizam, A., 1999. Life and tourism in the year 2050. Int. J. Hosp. Manag. 18, 331–343.

Prashanth, L.A., Bhatnagar, S., 2011. Reinforcement learning with function approximation for traffic signal control. IEEE Trans. Intell. Transp. Syst. 12, 412–421.

Rezaee, K., Abdulhai, B., Abdelgawad, H., 2012. Application of reinforcement learning with continuous state space to ramp metering in real-world conditions. In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 1590–1595.

Richter, S., Aberdeen, D., Yu, J., 2007. Natural actor-critic for road traffic optimisation. In: Advances in Neural Information Processing Systems, vol. 19, pp. 1169–1176.

Salkham, A., Cunningham, R., Garg, A., Cahill, V., 2008. A collaborative reinforcement learning approach to urban traffic control optimization. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, pp. 560–566.

Sánchez-Medina, J.J., Galán-Moreno, M.J., Rubio-Royo, E., 2010. 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. 11, 132–141.

Schouten, R., Steingröver, M., 2007. Reinforcement learning of traffic light controllers under partial observability (Master′s Thesis). Faculty of Science University of Amsterdam, Amsterdam, The Netherlands.

Schrank, D., Lomax, T., Eisele, B., 2011. TTI′s 2011 Urban Mobility Report. TII Report Exhibit B-15. Texas Transportation Institute (TII), The Texas A&M University System, U.S. Department of Transportation, University Transportation Center for Mobility.

Sen, S., Head, K.L., 1997. Controlled optimization of phases at an intersection. Transp. Sci. 31, 5–17.

Shoham, Y., Leyton-Brown, K., 2010. Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations. Cambridge University Press, Cambridge, UK.

Smith, R.H., Chin, D.C., 1995. Evaluation of an adaptive traffic control technique with underlying system changes. In: Proceedings of the IEEE 27th Winter Simulation Conference (WSC 1995), Arlington, VA, pp. 1124–1130.

Srinivasan, D., Choy, M.C., Cheu, R.L., 2006. Neural networks for real-time traffic signal control. IEEE Trans. Intell. Transp. Syst. 7, 261–272.

Steingröver, M., Schouten, R., Peelen, S., Nijhuis, E., Bakker, B., 2005. Reinforcement learning of traffic light controllers adapting to traffic congestion. In: Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence Conference (BNAIC 2005), Brussels, Belgium, pp. 216–223.

Thorpe, T.L., Anderson, C.W., 1996. Traffic Light Control Using SARSA with Three State Representations. Technical Report. IBM Corporation.

Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62, 1805–1824.

U.S. Department of Transportation, N.H.T.S.A., 2012. 2010 Motor Vehicle Crashes: Overview. Traffic Safety Facts Research Note DOT HS 811 552. NHTSA′s National Center for Statistics and Analysis, Washington, DC.

Wenchen, Y., Lun, Z., Zhaocheng, H., Lijian, Z., 2012. Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and paramics simulation. In: Proceedings of the IEEE 15th International Conference on Intelligent Transportation Systems (ITSC 2012), Anchorage, AK, pp. 391–396.

Wiering, M., 2000. Multi-agent reinforcement learning for traffic light control. In: Proceedings of the 17th International Conference on Machine Learning (ICML 2000), pp. 1151–1158.

Wiering, M., Vreeken, J., van Veenen, J., Koopman, A., 2004. Simulation and optimization of traffic in a city. In: Proceedings of the IEEE Intelligent Vehicle symposium (IV 2004), Parma, Italy, pp. 453–458.