# Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto

Samah El-Tantawy, *Student Member, IEEE*, Baher Abdulhai, *Member, IEEE*, and Hossam Abdelgawad

***Abstract*—Population is steadily increasing worldwide, resulting in intractable traffic congestion in dense urban areas. Adaptive traffic signal control (ATSC) has shown strong potential to effectively alleviate urban traffic congestion by adjusting signal timing plans in real time in response to traffic fluctuations to achieve desirable objectives (e.g., minimize delay). Efficient and robust ATSC can be designed using a multiagent reinforcement learning (MARL) approach in which each controller (agent) is responsible for the control of traffic lights around a single traffic junction. Applying MARL approaches to the ATSC problem is associated with a few challenges as agents typically react to changes in the environment at the individual level, but the overall behavior of all agents may not be optimal. This paper presents the development and evaluation of a novel system of multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC). MARLIN-ATSC offers two possible modes: 1) independent mode, where each intersection controller works independently of other agents; and 2) integrated mode, where each controller coordinates signal control actions with neighboring intersections. MARLIN-ATSC is tested on a large-scale simulated network of 59 intersections in the lower downtown core of the City of Toronto, ON, Canada, for the morning rush hour. The results show unprecedented reduction in the average intersection delay ranging from 27% in mode 1 to 39% in mode 2 at the network level and travel-time savings of 15% in mode 1 and 26% in mode 2, along the busiest routes in Downtown Toronto.**

***Index Terms*—Adaptive traffic signal control, game theory, microsimulation modeling, multi-agent reinforcement learning, multi-agent system, reinforcement learning.**

Manuscript received October 2, 2012; revised February 11, 2013; accepted March 7, 2013. Date of publication April 16, 2013; date of current version August 28, 2013. This work was supported by the University of Toronto, Toronto, ON, Canada, through the Connaught Scholarship and the Ontario Graduate Scholarship. The Associate Editor for this paper was B. Chen.

S. El-Tantawy is with the Intelligent Transportation Systems Center and Testbed, University of Toronto, Toronto, ON M5S 1A4, Canada (e-mail: samah.el.tantawy@utoronto.ca).

B. Abdulhai is with the Toronto Intelligent Transportation Systems Center and Testbed, Civil Engineering Department, University of Toronto, Toronto, ON M5S 1A4, Canada (e-mail: baher.abdulhai@utoronto.ca).

H. Abdelgawad is with the Civil Engineering Department, University of Toronto, Toronto, ON M5S 1A4, Canada (e-mail: hossam.abdelgawad@alumni.utoronto.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2255286

## I. Introduction

POPULATION is steadily increasing worldwide; consequently, the demand for mobility is increasing, particularly during good economic times. When growth in social and economic activities outpace growth in transportation infrastructure, congestion is inevitable. Severe congestion and long commute hours plague many large urban areas around the word, and the Greater Toronto, ON, Canada, Area is no exception. Congestion wastes time, hampers social and economic activities, and harms the environment, which all deteriorate the quality of our lives. Adaptive traffic signal control (ATSC) has the potential to efficiently alleviate traffic congestion by adjusting signal timing parameters in response to traffic fluctuations to achieve a certain objective (e.g., minimize delay); therefore, it has great potential to outperform both pretimed and actuated control [1]. Employing ATSC strategies at the local level (isolated intersection) might limit their potential benefits. Therefore, optimally controlling the operation of multiple intersections simultaneously can be synergetic and beneficial. However, such integration certainly adds more complexity to the problem. Coordination has been typically approached in a centralized way (e.g., Split Cycle Offset Optimization Technique (SCOOT) [2] and TUC [3]), which is only feasible if communication channels among all intersections and the central control location are available, which is resource demanding. The Sydney Coordinated Adaptive Traffic System (SCATS) [4] is another example of an adaptive signal control system that is a hierarchical and distributed system in which an area is divided into smaller subsystems (in the range of one to ten intersections) that independently perform. PRODYN [5], Optimized Policies for Adaptive Control [6], and RHODES [7] are also examples of adaptive systems that are decentralized, but their relatively complex computation schemes make their implementation costly [8].

The coordination mechanism in the given systems is employed along an arterial (where the major demand is). Although it is important to efficiently operate traffic signals along arterials where the major demand is (e.g., progression), it is also important to consider the network-wide effect of such operation. In a signalized urban network setting, considering a network-wide objective has the potential to improve overall network performance and mobility and to reduce emissions.

As an alternative, coordination can be plausibly achieved using reinforcement learning and game-theoretic approaches [8]. Reinforcement learning (RL) has shown good potential for self-learning closed-loop optimal traffic signal control in a stochastic traffic environment [9], [10]. RL has the added advantage of being able to perpetually learn and improve service over time. In RL, a traffic signal represents a control agent that interacts with the traffic environment in a closed-loop system to achieve optimal mapping between the environment's traffic state and the corresponding optimal control action, offering an optimal control law. Mapping from states to actions is also referred to as the control policy. The agent iteratively receives a feedback reward for actions taken and adjusts the policy until it converges to the optimal control policy. Applying RL to a transportation network of multiple signalized intersections is associated with some challenges. Agents typically react to changes in the environment at the individual level, but the overall behavior of all agents may not be optimal. Each agent is faced with a moving-target learning problem, in which the agent's optimal policy changes as the other agents' policies change over time [8]. Game theory provides tools to model multiagent systems as a multiplayer game and provide a rational strategy to each player in a game. Multiagent reinforcement learning (MARL) is an extension of RL to multiple agents in a stochastic game (SG; i.e., multiple players in a stochastic environment). The decentralized traffic control problem is an excellent testbed for MARL due to the inherited dynamics and stochastic nature of the traffic system [8], [11], which is our focus in this paper.

Despite recent approaches employing MARL in an SG, MARL faces many challenges. First is the exponential growth in the state–action space with the increase in the number of agents. Second is that the majority of the MARL-based ATSC in the literature assume that agents independently learn, in which case each agent individually acts in its local environment without explicit coordination[1] with other agents in the environment. Although this simplifies the problem, it limits their usefulness in case of a network of agents. For example, in oversaturated traffic conditions, queues could easily propagate from a downstream intersection (agent) and spills back to upstream intersections (agents) in a network-wide cascading fashion; such cases require network-wide multiagent coordination, as discussed earlier. Thus, flexible and computationally efficient approaches are becoming instrumental in controlling a network of agents, plausibly by employing heuristics and approximate approaches based on modifying the existing MARL techniques [8].

To address these limitations, we present a novel multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC) that offers the following features and characteristics: 1) *decentralized design* and *operation*, which is typically less expensive compared with the centralized system; 2) *scalable* to accommodate any network size; 3) *robust*, i.e., with no single point of failure; 4) *model-free*, i.e., does not require a model of the traffic system that is challenging to obtain; 5) self-*learning*, i.e., reduces human intervention in the operation phase after deployment (the most costly component of operating existing ATSCs); and 6) *coordinated*, i.e., by implementing mode 2 (integrated mode), which coordinates the operation of intersections in 2-D road networks (e.g., grid network) this is a new feature that is unprecedented in ATSC state of the art and practice. In addition, MARLIN-ATSC is tested on a large-scale simulated network of 59 intersections in Downtown Toronto using the input data (e.g., traffic counts, signal timings, etc.) provided by the City of Toronto.

[1]It is important to not confuse the coordination that is concerned about creating green wave along a certain corridor by adjusting the offset timing (defined as progression hereafter) with the mechanism between agents (signalized intersections) to coordinate their policies such that a certain objective is achieved for the entire traffic network (defined as coordination hereafter). In this paper, coordination refers to the latter one.

## II. From Single-Agent to Multiagent Reinforcement Learning

### A. RL

Typically, RL is concerned with a single agent operating in an environment so as to maximize its cumulative long-run reward. The environment is modeled as a Markov decision process (MDP), assuming that the underlying environment is stationary in which that the environment's state only depends on the agent's actions. The most common single-agent RL algorithm is $Q$-learning [12]. The $Q$-learning agent learns optimal mapping between the environment's state $s$ and the corresponding optimal control action $a$ based on accumulating rewards $r(s,a)$. Each state–action pair $(s,a)$ has a value called the $Q$-factor that represents the expected long-run cumulative reward for the state–action pair $(s,a)$. In each iteration, i.e., $k$, the agent observes current state $s$ and chooses and executes action $a$ that belongs to the available set of actions $A$; then, the $Q$-factor is updated according to the immediate reward $r(s,a)$ and the state transition to state $s'$ as follows [13]:

$$Q^k(s^k,a^k) = (1-\alpha)Q^{k-1}(s^k,a^k) + \alpha\left[r(s^k,a^k) + \gamma \max_{a^{k+1}\in A} Q^{k-1}(s^{k+1},a^{k+1})\right]$$

where $\alpha$ and $\gamma \in (0,1]$ are referred to as the learning rate and the discount rate, respectively.

The agent can simply choose the greedy action at each iteration based on the stored $Q$-factors, as follows:

$$a^{k+1} \in \arg\max_{a\in A}\left[Q(s,a)\right].$$

However, sequence $Q^k$ is proven to converge to the optimal value only if the agent visits the state–action pair for an infinite number of iterations [12]. This means that the agent must sometimes explore (try random actions) rather than exploit the best known actions. To balance the exploration and exploitation in $Q$-learning, algorithms such as $\epsilon$-greedy and softmax are typically used [13].

### B. MARL

MARL is an extension of RL to multiple agents (signalized intersections). The decentralized traffic signal control problem

is an excellent testbed for MARL due to the inherent dynamics and stochastic nature of the traffic system [8], [11]. The simplest way to extend RL to the MARL is to consider the local state and local action for each agent, assuming a stationary environment and that the agent's policy is the prime factor affecting the environment. However, MARL in the traffic environment is associated with some challenging issues because the traffic environment is nonstationary since it includes multiple agents learning concurrently, i.e., the effect of any agent's action on the environment depends on the actions taken by the other agents. Each agent is, therefore, faced with a moving-target learning problem because the best policy changes as the other agents' policies change, which accentuates the need for coordination among agents. Coordination can be achieved by considering the joint state and joint action for the other agents in the learning process. Moreover, given that all agents are simultaneously acting, the agents' choices of actions must be mutually consistent to achieve their common goal of optimizing the signal control problem. Therefore, the agents require a coordination mechanism to make the optimal decision from the possible joint actions (i.e., agents have to coordinate their choices/actions to reach a unique equilibrium policy). Agent coordination in this context is not to be confused with conventional traffic signal coordination that maximizes green bands, offsets, etc.

Markov games form the theoretical framework of MARL. A Markov game (known as an SG) is an extension of the MDP to multiagent environments. The game is played in a sequence of stages. At each stage, the game has a certain state in which the players select actions and each player receives a reward that depends on the current state and the chosen joint action. The game then moves to a new random state whose distribution depends on the previous state and the joint action chosen by the players. The procedure is repeated in the new state and continues for a finite or infinite number of stages. The agent's objective is to find a joint policy (known as equilibrium) in which each individual policy is a best response to the others, such as Nash equilibrium [14]. A comprehensive survey of MARL algorithms can be found in [15]. Examples of MARL approaches with a coordination mechanism are Optimal Adaptive Learning (OAL) [16] for cooperative games and Nonstationary Converging Policies (NSCP) algorithms [17] for general sum games. Coordination in OAL [16] and NSCP [17] algorithms is achieved by modeling the other agents' policies, and hence, the agent can act accordingly. However, the applicability of such approaches is limited to optimize a few traffic signal agents due to the obvious exponentially increasing joint space of states with the increase in the number of agents [8].

## III. Challenges of Applying Multiagent Reinforcement Learning for Adaptive Traffic Signal Control Systems

Thorpe [18] applied the state–action–reward–state–action (SARSA) RL algorithm to a simulated traffic light control problem. The results showed that the SARSA RL algorithm outperformed the fixed timing plans by reducing the average vehicle waiting time by 29%. Wiering [19] utilized model-based RL (with state transition models and state transition probabilities) to control traffic-light agents to minimize the waiting time of vehicles in a small grid network. The experimental results showed that RL systems outperform nonadaptive systems by 22% in waiting time. Abdulhai *et al.* [20] applied a model-free $Q$-learning technique to a simple two-phase isolated traffic signal in a 2-D road network. $Q$-learning for the isolated traffic-light controller outperformed the pretimed control scheme for the variable traffic flow case by around 44%. Camponogara and Kraus Jr. [21] formulated the traffic signal control problem as a distributed SG, in which agents employed a distributed $Q$-learning algorithm. When testing policy 3 (i.e., both agents run $Q$-learning), a 43% reduction in waiting time was achieved compared with policy 1 (assigns the same probability to all actions available to an agent). De Oliveira *et al.* [22] extended RL to multiple isolated traffic lights. They proposed an RL method called RL with context detection, which can handle stochastic traffic patterns that occur due to traffic dynamics. Richter *et al.* [23] applied the natural actor critic (NAC) algorithm to a $10 \times 10$ junction grid simulation network. NAC outperformed SAT (adaptive controller inspired by SCATS) by 20% reduction in average network travel time. Another example can be found in the work of Arel *et al.* [24], where RL is used to control the central intersection in a network of five intersections, whereas the other four intersections use the longest-queue-first heuristic. Li *et al.* [25] proposed an RL-based approach in which each agent considered the weighted sum of its local delay and its neighbors' delays as the outcome of its action. Salkham *et al.* [26] proposed a similar algorithm to provide adaptive and efficient urban traffic control. Medina and Benekohal [27] used $Q$-learning and an approximate DP algorithm to control the traffic signals in which the learning agent considered its local state in addition to information on the congestion levels of neighboring intersections.

In most of the previous studies, algorithms were applied to simplified scenarios and under strong assumptions in terms of traffic behavior by considering a simplified simulation environment [20]–[24] and/or assuming hypothetical traffic flows [18]–[24], [28], which does not necessarily mimic the reality in traffic networks. Moreover, the previous studies considered independent learning agents and considered no explicit mechanism for coordination.

On the other hand, Kuyer *et al.* [29] found the only algorithm, to the best of authors' knowledge, that considered an explicit coordination mechanism between learning agents, extending the work of Wiering in [19] using the Max-plus algorithm. The Max-plus algorithm was used to estimate the optimal joint action by sending locally optimized messages among connected agents. However, the Max-plus algorithm was found computationally demanding as it requires negotiations between the agents to coordinate their actions. Due to the real-time nature of the ATSC problem, this forces the agents to report their current best action at any time even if the action found so were suboptimal. In addition, the use of a model-based RL approach adds unnecessary complexities compared with using a model-free approach such as $Q$-learning.

In conclusion, there are two major challenges associated with applying RL (MARL) to the ATSC problem, i.e., the need

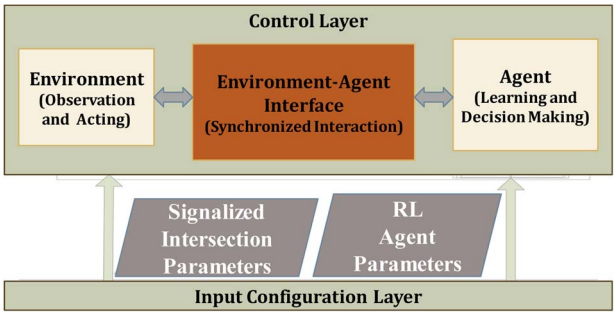


Fig. 1. MARLIN-ATSC platform.

for coordination and the cure of dimensionality. These major challenges are discussed as follows.

- ***Need for Coordination***: The need for coordination stems from the fact that the effect of any agent's action on the environment also depends on the actions taken by the other agents. Hence, the agents' choices of actions must be mutually consistent to achieve their intended effect [15].It can be concluded from the reviewed literature that the majority of the previous studies considered independent learning agents, such as that by De Oliveira *et al.* [22], Camponogara and Kraus Jr. [21], Bazzan [30], Richter *et al.* [23], Arel *et al.* [24], Wiering [19], Li *et al.* [25], and Salkham *et al.* [26]. Although Kuyer *et al.* [29] considered the two-level coordination, it suffers from the aforementioned limitations.
- ***Curse of Dimensionality***: Although there exist a few coordination-based MARL methods (e.g., OAL [16] and NSCP [17]), they suffer from the curse of dimensionality issue that arises because the state space is exponentially growing with the number of agents. Even in SG-based MARL approaches that are proven to optimally converge to the joint policy, each agent has to keep a set of tables whose size is exponential in the number of agents: $|S_1| \times \cdots \times |S_N| \times |A_1| \times \cdots \times |A_N|$, where $S_i$ and $A_i$ represent the state and action spaces for agent $i$, respectively. In addition to the dimensionality issue, these methods require each agent to observe the state of the whole system, which is infeasible in the case of transportation networks. In the following section, we introduce a new algorithm that maintains a coordination mechanism between agents without compromising the dimensionality of the problem.

## IV. Multiagent Reinforcement Learning For Integrated Network of Adaptive Traffic Signal Controllers Platform

The MARLIN-ATSC platform is shown in Fig. 1. The platform consists of two main layers. The first layer is an input configuration layer that is responsible for configuring and providing the necessary input to the second layer.

The configuration layer has two main roles: 1) It configures the simulation-based learning environment (model) such that the simulated environment closely matches the real-world environment and 2) configures RL-design parameters.

The second layer is a control layer that includes three interacting components, as shown in Fig. 1.

### A. Agent

The agent component implements the control algorithm; the agent is the learner and the decision-maker that interacts with the environment by first receiving the system's state and the reward and then selecting an action accordingly. A generic agent model is developed using Java Programming Language such that different levels of coordination, learning methods, state representations, phasing sequence, reward definition, and action selection strategies can be tested for any control task. In MARLIN-ATSC, agents can implement one of the following two control modes.

- **Independent Mode**: In this mode, each controller has an RL agent working independently of other agents using MARL for independent controllers (***MARL-I***), in which each agent implements a $Q$-learning algorithm [12].
- **Integrated Mode**: In this mode, each controller coordinates the signal control actions with the neighboring controllers by implementing a MARLIN learning algorithm.
- **MARLIN Learning Approach:** MARLIN presents a new control system that maintains an explicit coordination mechanism while addressing the curse of dimensionality problem for a large-scale network of connected agents by means of the following measures.
- **Exploiting the Principle of Locality of Interaction [31] Among Agents**: The principle of locality of interaction endeavors to estimate a local neighborhood utility that maps the effect of an agent to the global value function while only considering the interaction with its neighbors. Hence, it is sufficient to consider the neighbors' policies to find the best policy for the agent.
- **Utilizing the Modular *Q*-Learning Technique [32]**: Modular $Q$-learning partitions the state space to partial state spaces that consist of two agents. As a consequence, the size of the partial state space is always $|S|^2$ regardless of the number of agents and, therefore, results in a reasonable state space.

In MARLIN, each signalized intersection (agent) plays a game with all its adjacent intersections in its neighborhood. The agent has a number of learning modules; each corresponds to one game. The state and action spaces are distributed such that the agent learns the joint policy with one of the neighbors at a time, following the principle of modular $Q$-learning.

The following are the steps for the learning approach designed in MARLIN that is formally described in a pseudocode in Algorithm 1.

- If there are $|\mathrm{NB}_i|$ neighbors for agent $i$, there are $|\mathrm{NB}_i|$ partial state and action spaces for agent $i$. Each partial state space and action space consists of agent $i$ and one of the neighbors $\mathrm{NB}_i[j]$, s.t. $j \in \mathrm{NB}_i$ $(S_i, S_{\mathrm{NB}_i[j]}, A_i, A_{\mathrm{NB}_i[j]})$.
- Each agent $i$ builds a model that estimates the policy for each of its neighbors and is represented by matrix $M_{i,\mathrm{NB}_i[j]}$, s.t. $j \in \mathrm{NB}_i$, where the rows are joint states $S_i \times S_{\mathrm{NB}_i[j]}$, and the columns are the neighbor's actions

$A_{\mathrm{NB}_i[j]}$. Each cell $M_{i,\mathrm{NB}_i[j]}([s_i, s_{\mathrm{NB}_i[j]}], a_{\mathrm{NB}_i[j]})$ represents the probability that agent $\mathrm{NB}_i[j]$ takes action $a_{\mathrm{NB}_i[j]}$ at joint state $[s_i, s_{\mathrm{NB}_i[j]}]$ using the count of visits to the state–action $\upsilon([s_i^k, s_{\mathrm{NB}_i[j]}^k], a_{\mathrm{NB}_i[j]}^k)$ for the state–action pair $([s_i^k, s_{\mathrm{NB}_i[j]}^k], a_{\mathrm{NB}_i[j]}^k)$ [see (3)].

- Each agent $i$ learns the optimal joint policy for agents $i$ and $\mathrm{NB}_i[j]$ $\forall$ $j \in \{1, \ldots, |\mathrm{NB}_i|\}$ by updating the $Q$-values that are represented by a matix of $|S_i \times S_{\mathrm{NB}_i[j]}|$ rows and $|A_i \times A_{\mathrm{NB}_i[j]}|$ columns, where each cell $Q_{i,\mathrm{NB}_i[j]}([s_i, s_{\mathrm{NB}_i[j]}],[a_i, a_{\mathrm{NB}_i[j]}])$ represents the $Q$-value for a state–action pair in the partial spaces corresponding to the pair of connected agents $(i, \mathrm{NB}_i[j])$.
- Each agent updates $Q$-values $Q_{i,\mathrm{NB}_i[j]}([s_i, s_{\mathrm{NB}_i[j]}], [a_i, a_{\mathrm{NB}_i[j]}])$ using the value of the best-response action taken in the next state. The best-response value $(\mathrm{br}_i^k)$ is the maximum expected $Q$-value at the next state, which is calculated using the models for other agents [see (4)].
- Each agent decides its action without direct interaction with the neighbors. Instead, the agent uses the estimated models for the other agents and act accordingly. Agent $i$ chooses the next action using a simple heuristic decision procedure, which bias action selection toward actions that have the maximum expected $Q$-value over its neighbors $\mathrm{NB}_i$. The likelihood of $Q$-values is evaluated using the models of the other agents, i.e., $M_{i,\mathrm{NB}_i[j]}$, estimated in the learning process [see (6)].

**Algorithm 1: MARLIN Learning**

**Initialization at time $k = 0$:**
**For each agent $i$**, $i \in \{1, 2, \ldots, N\}$:
**For each neighbor $j \in \{1, 2, \ldots, |\mathrm{NB}_i|\}$**
Initialize $s_i^0$, $a_i^0$, $a_{\mathrm{NB}_i[j]}^0$

$$M_{i,\mathrm{NB}_i[j]}^0\left([s_i, s_{\mathrm{NB}_i[j]}], a_{\mathrm{NB}_i[j]}\right) = 1/|A_{\mathrm{NB}_i[j]}|,$$

$$Q_{i,\mathrm{NB}_i[j]}^k\left([s_i, s_{\mathrm{NB}_i[j]}], [a_i, a_{\mathrm{NB}_i[j]}]\right) = 0$$

**End for**
**End for**
**For each time step $k$, do**:
**For each agent $i$, $i \in \{1, 2, \ldots, N\}$, do:**
**For each neighbor** $\mathrm{NB}_i[j]$, $j \in \{1, 2, \ldots, |\mathrm{NB}_i|\}$ **do**:
a. Observe $a_{\mathrm{NB}_i[j]}^k$, $s_i^{k+1}$ $s_{\mathrm{NB}_i[j]}^{k+1}$, and $r_i^k$
b. Update $M_{i,\mathrm{NB}_i[j]}$

$$M_{i,\mathrm{NB}_i[j]}\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], a_{\mathrm{NB}_i[j]}^k\right) = \frac{\upsilon\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], a_{\mathrm{NB}_i[j]}^k\right)}{\sum_{a_{\mathrm{NB}_i[j]} \in A_{\mathrm{NB}_i[j]}} \upsilon\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], a_{\mathrm{NB}_i[j]}\right)} \quad (3)$$

c. Choose the maximum expected $Q$-value at state $s_{\mathrm{NB}_i[j]}^{k+1}$

$$\mathrm{br}_i^k = \max_{a_i \in A_i}\left[\sum_{a_{\mathrm{NB}_i[j]} \in A_{\mathrm{NB}_i[j]}} Q_{i,\mathrm{NB}_i[j]}^k \times \left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], [a_i, a_{\mathrm{NB}_i[j]}]\right) \times M_{i,\mathrm{NB}_i[j]}\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], a_{\mathrm{NB}_i[j]}\right)\right] \quad (4)$$

d. Update $Q_{i,\mathrm{NB}_i[j]}$

$$Q_{i,\mathrm{NB}_i[j]}^k\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], \left[a_i^k, a_{\mathrm{NB}_i[j]}^k\right]\right) = (1-\alpha) Q_{i,\mathrm{NB}_i[j]}^{k-1} \times \left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], \left[a_i^k, a_{\mathrm{NB}_i[j]}^k\right]\right) + \alpha\left[r_i^k + \gamma \mathrm{br}_i^k\right] \quad (5)$$

Decide

$$a_i^{k+1} = \arg\max_{a_i \in A_i}\left[\sum_{j \in \{1,2,\ldots,|\mathrm{NB}_i|\}} \sum_{a_{\mathrm{NB}_i[j]} \in A_{\mathrm{NB}_i[j]}} \times Q_{i,\mathrm{NB}_i[j]}^k\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], [a_i, a_{\mathrm{NB}_i[j]}]\right) \times M_{i,\mathrm{NB}_i[j]}\left(\left[s_i^k, s_{\mathrm{NB}_i[j]}^k\right], a_{\mathrm{NB}_i[j]}\right)\right] \quad (6)$$

**End For**
**End For**
**End For**

### B. Simulation Environment

The simulation environment component models the traffic environment. In this paper, Paramics, which is a microscopic traffic simulator, is used to model traffic environment [33]. Paramics models stochastic vehicle flow by employing speed regulations, car-following, gap acceptance, and overtaking rules. Paramics provides three methods of traffic assignment that could be employed at different levels, i.e., "all-or-nothing" assignment, stochastic assignment, and dynamic feedback assignment. In this application, a dynamic stochastic traffic assignment was used where 1) random noise was added to the travel cost to account for heterogeneity among drivers' perception of travel cost, and 2) a dynamic feedback interval was used to update route travel times for familiar drivers in the simulation. Paramics application programming interface functions were used to construct the state, execute the action, and calculate the reward for each signalized intersection.

Some of the main challenges in deigning any RL system are the design of the state, action, and reward definitions. In [34], a comprehensive investigation of these key issues in RL-based signal control for isolated intersections is conducted. The state, action, and reward definitions recommended in [34] and [35] are adopted in this paper as follows. (For more details on the definitions, see [34].)

- ***State Definition: Queue Length***: The agent's state is represented by a vector of $2 + P$ components, where $P$ is the number of phases. The first two components are 1) index of the current green phase and 2) elapsed time of the current phase. The remaining $P$ components are the maximum queue lengths associated with each phase.

- ***Action Definition: Variable Phasing Sequence***: The agent is designed to account for a variable phasing sequence in which the control action is either to extend the current phase or to switch to any other phase according to the fluctuations in traffic, possibly skipping unnecessary phases. Therefore, this algorithm is an acyclic timing scheme with a variable phasing sequence in which not only the cycle length is variable but the phasing sequence is also not predetermined. Hence, the action is the phase that should be in effect next.
- ***Reward Definition: Reduction in the Total Cumulative Delay***: The immediate reward for a certain agent is defined as the reduction (saving) in the total cumulative delay associated with that agent, i.e., the difference between the total cumulative delays of two successive decision points. The total cumulative delay at time $k$ is the summation of the cumulative delay, up to time $k$, of all the vehicles that are currently in the intersections' upstreams. If the reward has a positive value, this means that the delay is reduced by this value after executing the selected action. However, a negative reward value indicates that the action results in an increase in the total cumulative delay.

### C. Interface

The interface component manages the interactions between the agent and the simulation environment by exchanging the state, reward, and action. The interaction between the agent and the environment is associated with the following design elements.

- A synchronized interaction between the agent and the environment was designed to ensure that the simulation environment is *held* while the agent is performing the learning and decision-making processes and, finally, produces the action that should be executed by a simulation environment. At the same time, the agent should be *on hold* until the action is executed in the environment, and the resultant state and the reward are measured.
- The system was designed such that the interaction frequency is variable for each agent. The interaction occurs at each specified time interval (1 s in this research) as long as the current green for a signalized intersection that is associated with an agent $i$ exceeded the minimum green time. Otherwise, the interaction starts after the minimum green.

The agent was designed to learn off-line through a simulation environment (such as the microsimulation model employed in the experiments) before field implementation. After convergence to the optimal policy, the agent can either be deployed in the field by mapping the measured state of the system to optimal control actions directly using the learned policy or continue learning in the field by starting from the learned policy.

## V. Experimental Results

### A. Testbed Network

MARLIN-ATSC is tested on a simulated network of the Lower Downtown Toronto network. The lower downtown of Toronto is the core of the City of Toronto. The lower downtown of Toronto in this study is bounded to the South by the Queens Quay corridor, to the West by Bathurst Street, to the East by the Don Valley Parkway, and to the North by Front Street. Toronto is the oldest, densest, and most diverse area in the region, and its downtown core contains one of the highest concentrations of economic activity in the country. This paper demonstrates large-scale application of MARLIN-ATSC on a simulated replica of the lower downtown core. A base-case (BC) simulation model for the lower downtown core was originally developed using Paramics in the Intelligent Transportation Systems Center and Testbed, University of Toronto, for the year 2006. In this application, the model is further refined to reflect the signal timing sheets provided by the City of Toronto.[2] The analysis period considered in this application is the A.M. peak hour, which has around 25 000 vehicular trips.

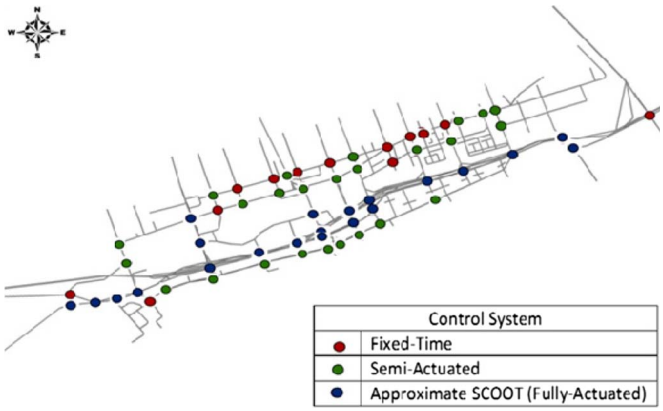


Fig. 2. Currently implemented signal control systems.

### B. Benchmarks

It is typically difficult to find a benchmark for large-scale traffic signal control problems given that the operational details of most traffic control systems are not easily available for obvious commercial reasons. The performance of the MARLIN-ATSC approach is compared with the BC scenario in which traffic signals, as defined and operated by the City of Toronto, are a mix of fixed-time control, semiactuated control, and SCOOT control, as shown in Fig. 2. It is worth noting that, due to the limited technical details about the operation of SCOOT, it is approximated in this thesis as an enhanced fully-actuated control, in which loop detectors are placed on all approaches, and extension times are conducted second by second.

### C. Results and Discussion

Results are reported for BC control systems (existing conditions), MARL-I (represents MARLIN-ATSC Independent Mode with no communication between agents), and MARLIN (represents MARLIN-ATSC Integrated Mode with coordination between agents).

[2]The contact person is Rajnath Bissessar; City of Toronto—Transportation Services, Manager of the Urban Traffic Control System (UTCS).

TABLE I
NETWORK-WIDE MOE IN THE NORMAL SCENARIO

| MOE \ System | BC | MARL-I | MARLIN | % Improvements MARL-I Vs. BC | % Improvements MARLIN Vs. BC | % Improvements MARLIN Vs. MARL-I |
|---|---|---|---|---|---|---|
| Average Intersection Delay (sec/veh) | 35.27 | 25.72 | 22.02 | 27.06% | 37.57% | 14.41% |
| Throughput (veh) | 23084 | 23732 | 24482 | 2.81% | 6.06% | 3.16% |
| Avg Queue Length (veh) | 8.66 | 6.60 | 5.88 | 23.77% | 32.07% | 10.88% |
| Std. Avg. Queue Length (veh) | 2.12 | 1.62 | 1.47 | 23.37% | 30.74% | 9.61% |
| Avg. Link Delay (sec) | 9.45 | 8.50 | 5.04 | 10.07% | 46.73% | 40.76% |
| Avg. Link Stop Time (sec) | 2.74 | 2.57 | 2.02 | 5.95% | 26.06% | 21.38% |
| Avg. Link Travel Time (sec) | 16.81 | 15.81 | 12.32 | 5.97% | 26.70% | 22.05% |
| CO2 Emission Factor (gm/km) | 587.28 | 421.34 | 412.21 | 28.26% | 29.81% | 2.17% |

The performance of each control system is evaluated based on the following measures of effectiveness:

- average delay per vehicle (s/veh);
- average maximum queue length per intersection (veh);
- average standard deviation of queue lengths across approaches (veh);
- number of completed trips;
- average $CO_2$ emission factors (gm/km);
- average travel time for selected routes (min).

Table I compares the performance of the BC against the MARLIN-ATSC system with and without communication among agents, i.e., MARLIN and MARL-I, respectively.

The analysis of the results shown in Table I leads to the following findings.

- The two MARLIN-ATSC algorithms result in lower average delay, higher throughput, shorter queue length, and stop time compared with those from the BC. The most notable improvements are those in the average delay (38% MARLIN versus BC), in the standard deviation of the average queue length (31% MARLIN versus BC), and in $CO_2$ emission factors (30% MARLIN versus BC).
- These substantial improvements are due to not only the intelligence of the RL algorithm but also the coordination mechanism between the agents to reach a network-wide set of actions that minimize long-term delay. This coordination results in the so-called "metering" effect from the upstream intersection to the downstream intersection while accounting for the queues and delays at the downstream intersection. In fact, the tangible savings in the standard deviation in the queue length is interesting because this means balanced queue among all intersection approaches.
- MARL-I outperforms the BC in all the measures of effectiveness (MOEs), most notably are the average intersection delay (27%) and the $CO_2$ emission factor (28%). However, comparing MARLIN with MARL-I, it is found that the latter experiences relatively higher delays because the actions in MARL-I are only based on locally collected data and, thereby, results in more vehicles retained in the network at the end of the simulation (6% throughput improvement in MARLIN versus 2.8% throughput improvement in MARL-I).

Table I shows a very promising overall performance of MARLIN. However, as shown in the wide range of average delays among intersections, the improvements at some intersections are much higher than the network averages. Therefore, the spatial distribution of percentage improvement is presented in Fig. 3.

It is important to study the effect of various control systems on travel time and travel-time variability for selected key routes in the lower downtown core of Toronto. Eight key routes are defined, as shown in Fig. 4.

Route travel times and standard deviation in travel time for the BC, MARL-I, and MARLIN scenarios are presented in Table II. The routes in Table II are arranged in descending order from the worst to the best in terms of percentage improvements in average route time for MARLIN versus BC. To further study the route travel times within the simulation hour, the travel times for selected routes are plotted in Fig. 4. The analysis in Table II and Fig. 4 leads to the following conclusions.

- It is clear that MARLIN outperforms MARL-I and BC in all routes. The percentage improvements range from 4% in route 1 to 30% in route 8. MARL-I outperforms BC in almost all cases; the percentage improvements range from 3% in route 5 to 15% in route 6 with the exception of route 8, in which the BC scenario performs better than MARL-I.

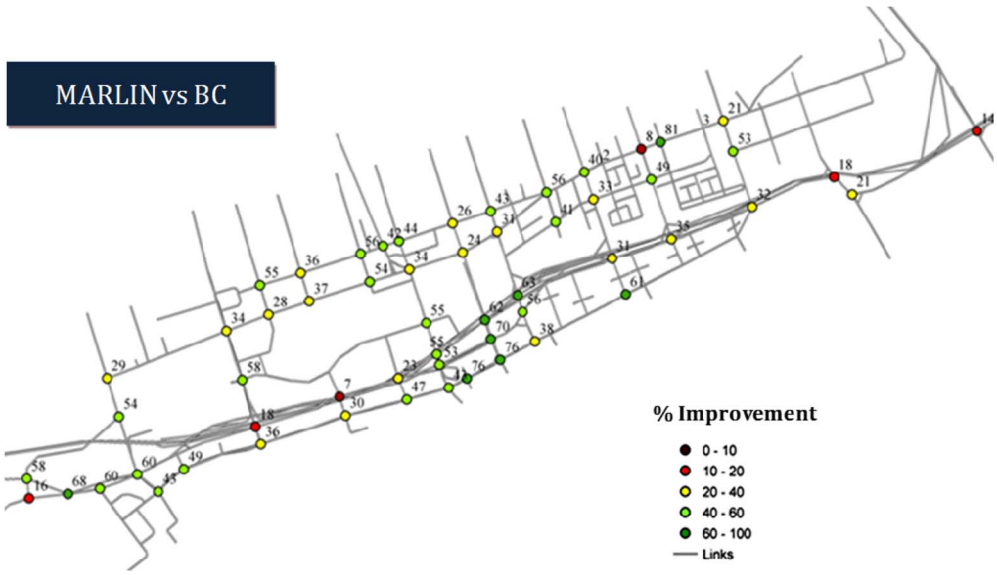


Fig. 3. Spatial distribution of percentage average delay improvements for MARLIN versus BC.

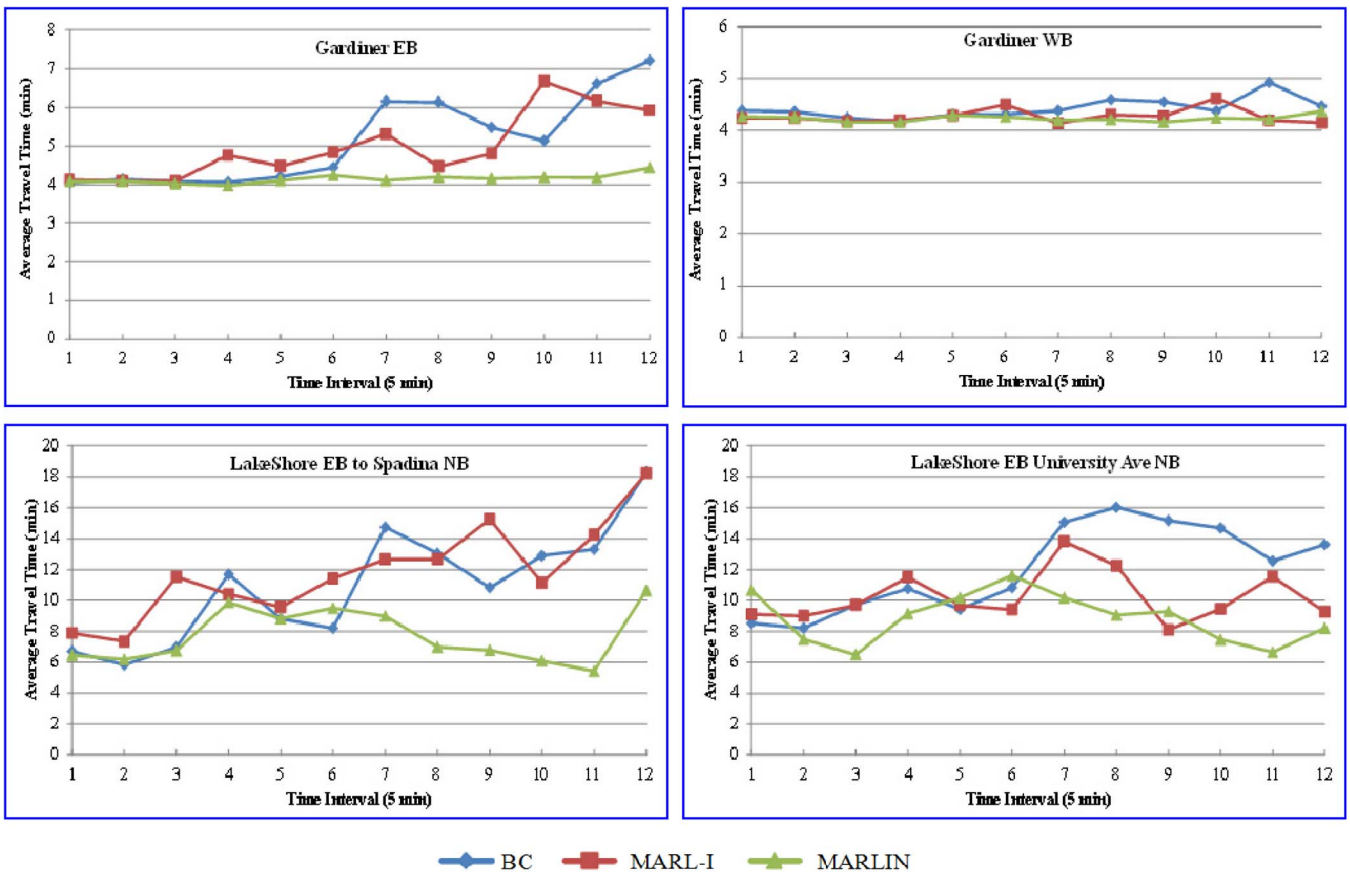


Fig. 4. Average route travel time for selected routes.

– It is interesting to find that the Gardiner Expressway East bound (EB) traffic (inbound) travel time improves by 19% in the MARLIN scenario. Alleviating the congestion on Spadina Street and York Street off-ramps contributes the most to these savings. This clearly shows the effect of downstream capacity on the freeway performance. For the Gardiner West bound (WB) direction, traffic was not as congested as the EB, but MARLIN still attains 4% improvement in average route travel times.
– The most congested routes appear to be routes 7 and 8, through which traffic originated at the west end of the study area and destined in the downtown core (Spadina Street and University Avenue). MARLIN achieves 30% and 26% improvements in routes 7 and 8, respectively,

TABLE II
ROUTE TRAVEL TIMES FOR BC, MARL-I, AND MARLIN

| Route \ System | BC | MARL-I | MARLIN | % Improvements MARL-I Vs. BC | % Improvements MARLIN Vs. BC | % Improvements MARLIN Vs. MARL-I |
|---|---|---|---|---|---|---|
| 1- Gardiner WB | 4.42 | 4.27 | 4.23 | 3.35% | 4.35% | 1.04% |
| St Dev | 0.20 | 0.15 | 0.06 | 26.52% | 68.03% | 56.49% |
| 2- Front_WB | 5.55 | 5.34 | 5.10 | 3.81% | 8.15% | 4.51% |
| St Dev | 0.92 | 0.79 | 0.49 | 13.39% | 47.10% | 38.93% |
| 3- Front EB | 10.65 | 9.13 | 7.88 | 14.28% | 13.69% | 13.69% |
| St Dev | 2.15 | 1.22 | 0.60 | 43.26% | 72.27% | 51.13% |
| 4- LakeShore WB | 10.31 | 9.07 | 8.46 | 12.02% | 17.91% | 6.70% |
| St Dev | 1.03 | 0.69 | 0.50 | 33.30% | 51.09% | 26.67% |
| 5- Gardiner EB | 5.14 | 4.98 | 4.15 | 3.18% | 19.30% | 16.65% |
| St Dev | 1.15 | 0.86 | 0.12 | 24.59% | 89.70% | 86.34% |
| 6- LakeShore EB | 16.31 | 13.28 | 12.10 | 18.60% | 25.77% | 8.82% |
| St Dev | 3.74 | 1.37 | 1.37 | 63.38% | 63.49% | 0.31% |
| 7- LakeShore EB University to Ave NB | 12.05 | 10.24 | 8.87 | 15.04% | 26.38% | 13.36% |
| St Dev | 2.81 | 1.66 | 1.64 | 40.80% | 41.75% | 1.62% |
| 8- LakeShore EB to Spadina NB | 10.94 | 11.86 | 7.70 | -8.40% | 29.59% | 35.05% |
| St Dev | 3.75 | 3.07 | 1.75 | 18.25% | 53.44% | 43.05% |

which reflects the superior effect of 2-D coordination between agents.

– From observing the temporal distribution of route travel time across the simulation hour, it is generally found that MARLIN is stable and exhibits less variation compared with the BC and MARL-I scenarios. While the BC scenario exhibits the highest variability in travel time (as shown in the standard deviation values in Table II), MARL-I still shows some variations, most notably in the two most congested routes (routes 7 and 8). MARLIN shows stable route travel times in all routes.In terms of computational complexity, each agent (intersection) converges to the optimal policy with different convergence speeds. The average time required to converge to the minimum average delay per intersection is 60 simulation runs (1 h each). The computational time for each learning step (1 simulation/s) is 4.2 ms.

## VI. CONCLUSION AND FUTURE WORK

In this paper, previous studies that tackled the ATSC problem using MARL approaches have been reviewed, and the gaps in literature have been highlighted. The major challenges for using a MARL-based signal control system were the need for coordination and the curse of dimensionality. To attain the compromise of achieving coordination-based decentralized adaptive real-time control without suffering from the curse of dimensionality challenge that is associated with MARL techniques, a MARLIN-ATSC system has been presented. In this system, each agent plays a game with its immediate neighbors. Each agent learns and converges to the best response policy to all neighbors' policies. This paper has demonstrated the essence of MARLIN-ATSC on a large-scale urban network of 59 intersections in Downtown Toronto. Results were reported for BC control systems (represented existing field conditions using signal timing sheets provided by the City of Toronto), MARL-I (represented MARLIN-ATSC Independent Mode with no communication between agents), and MARLIN (represented MARLIN-ATSC Integrated Mode with coordination between agents). Results showed that MARL-I and MARLIN outperformed the BC in all the MOEs. However, comparing MARLIN with MARL-I, it was found that the latter experiences higher delays. In terms of route travel time, it was generally found that MARLIN exhibited less average route travel time and less variation of the temporal distribution across the simulation hour compared with the BC and MARL-I scenarios. The daily economic benefits (i.e., travel-time savings) were estimated to be around $53 000. MARLIN-ATSC would cost approximately $1.2 million to implement across a network of 59 intersections. Consequently, the payback period is 23 days.

To quantify the benefits of MARLIN-ATSC relative to existing ATSC systems such as SCOOT, without approximation, the following approaches could be used in the future: 1) Compare simulation-based measures of MARLIN with real-life observations and benefits of SCOOT for SCOOT-controlled intersections, and 2) use hardware-in-the-loop simulation methodologies to replicate the logic of SCOOT within the simulation software such as Paramics.

## ACKNOWLEDGMENT

The authors would like to thank the City of Toronto staff for providing the data for this research.

## REFERENCES

[1] W. R. McShane, R. P. Roess, and E. S. Prassas, *Traffic Engineering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[2] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "SCOOT—A traffic responsive method of coordinating signals," Transp. Road Res. Lab., Crowthorne, U.K., Tech. Rep., 1981.
[3] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to trafficresponsive network-wide signal control," *Control Eng. Pract.*, vol. 10, no. 2, pp. 183–195, Feb. 2002.
[4] A. G. Sims and K. W. Dobinson, "SCAT—The Sydney co-ordinated adaptive traffic system: Philosophy and benefits," presented at the Int. Symp. Traffic Control Systems, Berkeley, CA, USA, 1979.
[5] J. L. Farges, J. J. Henry, and J. Tufal, "The PRODYN real-time traffic algorithm," presented at the 4th IFAC/IFIP/IFORS Symp. Control Transp. Syst., Baden-Baden, Germany, 1983.

[6] N. H. Gartner, "OPAC: A demand-responsive strategy for traffic signal control," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 906, pp. 75–81, 1983.
[7] K. L. Head, P. B. Mirchandani, and D. Sheppard, "Hierarchical framework for real-time traffic control," *Transp. Res. Rec.*, vol. 1360, pp. 82–88, 1992.
[8] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents Multi-Agent Syst.*, vol. 18, no. 3, pp. 342–375, Jun. 2009.
[9] B. Abdulhai and L. Kattan, "Reinforcement learning: Introduction to theory and potential for transport applications," *Can. J. Civil Eng.*, vol. 30, no. 6, pp. 981–991, Dec. 2003.
[10] S. El-Tantawy and B. Abdulhai, "An agent-based learning towards decentralized and coordinated traffic signal control," in *Proc. 13th IEEE ITSC*, 2010, pp. 665–670.
[11] S. El-Tantawy and B. Abdulhai, "Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers (MARLIN-OTC)," *Transp. Lett.: Int. J. Transp. Res.*, vol. 2, pp. 89–110, Apr. 2010.
[12] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.
[13] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.
[14] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. London, U.K: Classics Appl. Math., 1999.
[15] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
[16] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. 15th Nat. Conf. Artif. Intell./10th Conf. Innov. Appl. Artif. Intell.*, Madison, WI, USA, 1998, pp. 746–752.
[17] M. Weinberg and J. S. Rosenschein, "Best-response multiagent learning in non-stationary environments," in *Proc. 3rd Int. Joint Conf. Autonom. Agents Multiagent Syst.*, 2004, pp. 506–513.
[18] T. Thorpe, "Vehicle traffic light control using sarsa," M.S. thesis, Comput. Sci. Dept., Colo. St. Univ., Fort Collins, CO, USA, 1997.
[19] M. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 1151–1158.
[20] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *J. Transp. Eng.*, vol. 129, no. 3, pp. 278–285, Apr. 2003.
[21] E. Camponogara and W. Kraus, Jr., "Distributed learning agents in urban traffic control," in *Proc. 11th Portuguese Conf. Artif. Intell.*, 2003, pp. 324–335.
[22] D. De Oliveira, A. L. C. Bazzan, B. C. da Silva, E. W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva, and L. Lamb, "Reinforcement learning-based control of traffic lights in non-stationary environments: A case study in a microscopic simulator," in *Proc. EUMAS*, 2006, pp. 31–42.
[23] S. Richter, D. Aberdeen, and J. Yu, "Natural actor-critic for road traffic optimisation," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007.
[24] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, Jun. 2010.
[25] T. Li, D. B. Zhao, and J. Q. Yi, "Adaptive dynamic programming for multi-intersections traffic signal intelligent control," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, 2008, pp. 286–291.
[26] A. Salkham, R. Cunningham, A. Garg, and V. Cahill, "A collaborative reinforcement learning approach to urban traffic control optimization," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, 2008, pp. 560–566.
[27] J. C. Medina and R. F. Benekohal, "Q-learning and approximate dynamic programming for traffic control—A case study for an oversaturated network," presented at the Transp. Res. Board Annu. Meet., Washington, DC, USA, 2012, Paper 12-4103.
[28] L. Shoufeng, L. Ximin, and D. Shiqiang, "Q-Learning for adaptive traffic signal control based on delay minimization strategy," in *Proc. IEEE Int. Conf. Netw. Sens. Control*, 2008, pp. 687–691.
[29] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graph," in *Proc. 19th Eur. Conf. Mach. Learn.*, 2008, pp. 656–671.
[30] A. L. C. Bazzan, "A distributed approach for coordination of traffic signal agents," *Autonom. Agents Multi-Agent Syst.*, vol. 10, no. 1, pp. 131–164, Jan. 2005.
[31] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo, "Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs," in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, pp. 133–139.
[32] N. Ono and K. Fukumoto, "Multi-agent reinforcement learning: A modular approach," in *Proc. 2nd Int. Conf. Multi-Agent Syst.*, 1996, pp. 252–258.
[33] "Quadstone Paramics," Paramics Microscopic Traffic Simulation Software, 2012. [Online]. Available: http://www.paramics-online.com
[34] S. El-Tantawy and B. Abdulhai, "Comprehensive analysis of reinforcement learning methods and parameters for adaptive traffic signal control," presented at the Transp. Res. Board, Washington, DC, USA, 2011.
[35] S. El-Tantawy and B. Abdulhai, "Neighborhood coordination-based multi-agent reinforcement learning for coordinated adaptive traffic signal control," presented at the Transp. Res. Board, Washington, DC, USA, 2012.

**Samah El-Tantawy** (S'12) received the B.S. degree in electrical and communication engineering from Cairo University, Giza, Egypt, in 2004; the M.Sc. degree in engineering mathematics from Cairo University; and the Ph.D. degree in intelligent transportation systems from the University of Toronto, Toronto, ON, Canada, in 2012.

She is a Postdoctoral Fellow with the Intelligent Transportation Systems Laboratory and Testbed, University of Toronto. She has published a few journal papers and ten conference papers. She is the holder of a U.S. provincial patent.

Dr. El-Tantawy was the Vice President of the Institute of Transportation Engineers Student Chapter, University of Toronto, in 2010 and 2011; a Member of the Women in ITS group and of the IEEE Women in Engineering; and a friend of the Traffic Signal Systems Transportation Research Board Committee. She received four industrial scholarships (Intelligent Transportation Systems Canada, Transportation Association of Canada, Canadian Transportation Research Forum, and Canadian Institute of Transportation Engineers) and three provincial scholarships [two Ontario Graduate Scholarships (OGS) and one OGS in Science and Technology]. After her M.Sc. studies, she received the Ph.D. Connaught Scholarship from the University of Toronto. For her Ph.D. research, she developed a coordinated traffic signal control system using game theory concepts and multiagent reinforcement learning approaches (MARLIN-ATSC), which won her the MaRS Innovation Proof of Principle funding for 2012 to conduct system integration for MARLIN into a real controller and field implementation system requirements.

**Baher Abdulhai** (M'01) was born in Cairo, Egypt, in 1966. He received the Ph.D. degree in engineering from the University of California, Irvine, CA, USA, in 1996.

He is a Professor of civil engineering with the University of Toronto, Toronto, ON, Canada, and the Director of the Toronto Intelligent Transportation Systems Center. He has authored and coauthored nine book chapters, 46 journal papers, and 110 refereed conference papers on various intelligent transportation systems topics. He specializes in traffic control and management, traveler information systems, emergency evacuation optimization, dynamic road pricing, work zone traffic management, and pervasive and mobile intelligent transportation systems applications. His research employs intelligent transportation systems to reduce congestion, improve travel time and travel-time reliability, and enhance safety for travelers. His research encompasses open transportation service innovation and network-enabled platforms.

Dr. Abdulhai served on the Board of Directors of the Government of Ontario Transit Authority from 2004 to 2006. From 2005 to 2010, he served as a Canada Research Chair in intelligent transportation systems. From 2008 to 2011, he served as the Chair of the Board of the University of Toronto Urban Transportation Research and Advancement Center. From 2010 to 2012, he was the President of the ONE-ITS research society (one-its.net). He received several awards, including the IEEE Outstanding Service Award in 2006, the IEEE Outstanding Service Award, the Early Career Teaching Excellence Award, the Canada Foundation for Innovation New Opportunities Award, and the Ontario Innovation Trust New Opportunities Award. In 2005, the Intelligent Transportation Systems Center won the Ontario Showcase Merit Award of Excellence and the National GTEC Bronze Medal Award. His research, together with H. Abdelgawad, on emergency evacuation optimization won the International Transportation Forum Award, Leipzig, Germany, in 2010.

**Hossam Abdelgawad** received the B.Sc. degree in civil engineering and the M.Sc. degree in highway and traffic engineering from Cairo University, Giza, Egypt, and the Ph.D. degree in intelligent transportation systems from the University of Toronto, Toronto, ON, Canada, in June 2010.

He is currently the Manager of the Toronto Intelligent Transportation Systems Center and Testbed, University of Toronto. He is a Paramics Accredited User with ample experience in building, calibrating, and validating models using Paramics Microsimulation and other softwares such as VISSIM and DynusT. He is an expert in intelligent transportation systems and transport modeling and has presented his work at numerous international transportation conferences and published several book chapters and journal papers. He has a wide range of experience in intelligent transportation systems, advanced traffic management, transportation modeling, microscopic/mesoscopic traffic simulation, multimodal evacuation planning, and traffic signal optimization, including two years of experience in airport design, highway, and traffic engineering. Much of his professional career has been devoted to the development and refinement of tools/algorithms for real-time traffic management, multimodal evacuation planning, emergency evacuation management and optimization, artificial intelligence applications in transportation, demand management, pedestrian modeling, and crowd management.

Dr. Abdelgawad received the International Transport Forum Award for Transport and Innovation in 2010, which is a competition among 52 countries, for his work on emergency evacuation and disaster management of large cities. This included the development of an integrated multimodal emergency evacuation plan for the entire City of Toronto. He has won 15 scholarships/awards and has recently been featured in media streams/articles, including CBC's Living Cities, the University Affairs Magazine, the University of Toronto Civil Engineering Magazine, and University of Toronto Gradlife.