Uludağ University Journal of The Faculty of Engineering, Vol. 22, No. 2, 2017 RESEARCH

DOI: 10.17482/uumfd.338803

SIMULATION OF VEHICLES' GAP ACCEPTANCE DECISIONS USING REINFORCEMENT LEARNING

Bekir Oğuz BARTIN [*]

Received:08.04.2016; revised:25.04.2017; accepted:01.08.2017

Abstract: This paper presents the use of reinforcement learning approach for modeling vehicles' gap acceptance decisions at a stop-controlled intersection. The proposed formulation translates a simple gap acceptance decision into a reinforcement learning problem, assuming that drivers' ultimate objective in a traffic network is to optimize wait-time and safety. Using an off-the-shelf simulation tool, drivers are simulated without any notion of the outcome of their decisions. From multiple episodes of gap acceptance decisions, they learn from the outcome of their actions, i.e., wait-time and safety. A real-world traffic circle simulation network developed in Paramics simulation software is used to conduct experimental analyses. The results show that drivers' gap acceptance behavior in microscopic traffic simulation models can easily be validated with a high level of accuracy using Q-learning reinforcement-learning algorithm.

Keywords: Reinforcement learning, traffic simulation, gap acceptance, Paramics, traffic circle

Araçların Kritik Aralık Kabul Kararlarının Pekiştirmeli Öğrenmeyle Simülasyonu

Öz: Bu çalışma pekiştirmeli öğrenme yöntemini kullanarak araçların kritik aralık kabul kararlarının basit bir T-kavşakta modellemesini sunmaktadır. Önerilen yaklaşım araçların ulaşım ağlarındaki nihai amaçlarının bekleme sürelerini ve risklerini optimize edeceklerini varsayarak basit bir kritik aralık kabul kararını pekiştirmeli öğrenme problemine çevirmektedir. Trafik simülasyon yazılımında sürücüler kararlarının yol açacağı sonuçları bilmeyerek hareket eder, fakat bir çok simülasyon episodu sonrasında eylemlerinin sonuçlarını bekleme süresi ve risk şeklinde öğrenmeye başlarlar. Gerçek bir dönel kavşağın Paramics trafik simülasyon modeli deneysel analizler için kullanılmıştır. Elde edilen sonuçlara göre kullanılan bu simülasyon modeli "Q-learning" öğrenme yöntemi kullanılınca sürücülerin kritik aralık kabul kararlarının doğrulaması kolaylıkla yapılabilmektedir.

Anahtar Kelimeler: Pekiştirmeli öğrenme, mikroskobik trafik simülasyonu, kritik aralık kabulu, Paramics, dönel kavşak

1. INTRODUCTION

Microscopic traffic simulation tools allow transportation analysts to design traffic facilities and obtain detailed performance measures of existing designs, as well as to assess the impact of proposed design alternatives. The value of these tools, however, lies in their ability to stochastically simulate drivers' behavior, such as lane changing, car following, gap acceptance and route choice. The functions or rules that govern drivers' decisions in simulation software tools need to be validated and calibrated to reproduce the observed field conditions. Despite the advances in computing power and the ability of available simulation tools to represent complex driver behavior, simulation modeling and analysis is still a long and painstaking procedure, requiring extensive field data for validation/calibration and time.

[*] Altınbaş Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, İnşaat Mühendisliği Bölümü, İstanbul.
İletişim Yazarı:Bekir Oğuz BARTIN (bekir.bartin@altınbas.edu.tr)

There has been much work documented in the literature focusing on proper validation/calibration of traffic simulation models, how to use the results for decision making, and to forecast the impact of alternative operations or geometric designs (Sacks et al., 2002; Barton et al., 2002; Biller and Nelson, 2002; Henderson, 2003; Dowling et al., 2004; Bartin et al., 2006; Iyer et al., 2010). The value of any simulation software package lies in its ability to correctly simulate complex vehicle behavior. However, no matter how accurate the underlying models are, they need to be modified based on the characteristics of the study area. For example, Bartin et al. (2006) presented the shortcomings of the default gap acceptance model of Paramics when used to simulate traffic circles. It was shown that an arduous validation and calibration process based on detailed field data was required to modify the default gap acceptance model embedded in Paramics for accurately modeling a traffic circle. A similar validation/calibration process is surely needed when using any simulation software tool for analyzing any type of traffic facility.

1.1. Motivation

This study stems from the quest for alternative methods for validating/calibrating simulation models, which are so-called "model-free" approaches that do not rely on default functions or rules embedded in simulation software packages, rather use learning agents that adapt to the simulation network through iterated runs. It should be noted that a "model-free" approach does not entirely dismiss the functions or the rules that govern drivers' decisions in a simulation software tool, but replaces the ones that impact the performance of a specific traffic facility the most. For example, if the studied facility is a traffic circle or a roundabout, the most essential driver behavior that impacts its operational performance is drivers' gap acceptance decisions. Then, the default gap acceptance model of the simulation tool is replaced, and instead agents learn this behavior via learning methods in iterative simulation runs. As a result, they develop a "network specific" gap acceptance behavior. Another example is toll plazas, where the performance of these facilities is linked to vehicles' toll lane selection decisions. Then, in the "model-free" approach drivers' lane selection decisions in the simulation software tool are replaced by facility specific lane selection decisions learned through iterative simulation runs. In both cases, the remaining functions that govern the secondary driver behaviors, such as lane changing, car following and route choice remain default as provided by the simulation software package.

1.2. Objectives

The objective of this paper is to demonstrate the feasibility of a validation/calibration process for microscopic traffic simulation models, which does not rely on any underlying model or function that governs a particular driver decision/behavior. The proposed idea is to train drivers (agents) using a reinforcement learning method during simulation until they make decisions that are in accordance with the observed driver behavior. Agents adapt to the network and events during the simulation runs and make decisions based on their assigned objectives. During training, agents' objectives are twofold: (1) minimize time spent traveling and (2) minimize risk. Agents learn from their experience and improve their decision making progressively.

Q-learning reinforcement learning method is used to train drivers for their gap acceptance behavior. For this purpose, the simulation network of a traffic circle in New Jersey developed by Bartin et al. (2006) is borrowed for the simulation analyses presented in this paper. The circle has four yield-controlled intersections and one one-way stop-controlled intersection. Bartin et al. (2006) previously used binary probit models estimated using field data to model drivers' gap acceptance decisions. In the current paper, the stop-controlled intersection of the stated traffic circle is used as the test-bed for evaluating the feasibility of the proposed approach.

Paramics simulation software is used in the analyses presented in this paper. Paramics is an advanced suite of software tools for microscopic traffic simulation developed by Quadstone

Limited (Paramics Website, 2017). It is used to model the movement and behavior of individual vehicles on urban and highway road networks. Accurate geometry of the network and coding of links in Paramics are important for simulation results because the driver's behavior relies on characteristics of drivers and vehicles, the interactions between vehicles and network geometry. It has the ability to obtain detailed state variable information on each vehicle on time scales with better than second-by-second accuracy. The basic input data for the simulation are a road network and time dependent traffic demand (origin destination demand matrix). The software is controlled via a graphical user interface, which visualizes the network and simulated traffic in two or three dimensions. The most important feature of the software is its Application Programming Interface (API) feature, which allows users to customize many of its underlying simulation models. Paramics API is coded in C++ programming language. This feature allows modelers control the decisions and movements of individual vehicles and incorporate customized functionalities and test their own models instead of using the default, in-built traffic models.

The paper is organized as follows. Section 2 presents the related work that utilize reinforcement learning methods in traffic simulation studies. Section 3 describes the problem formulation, first with a simple case then extends it to a more complex gap acceptance decision-making scenario. Section 4 presents the traffic simulation model used for evaluating the feasibility of the proposed approach, and discusses the validation results. Section 5 presents the conclusions and the anticipated future work.

2. RELATED WORK

Reinforcement learning (RL) is a subset of machine learning that studies how an autonomous agent acts in an environment and learns to choose optimal actions to achieve an assigned goal (Mitchell, 1997). Similar to humans, agents learn from success and failure through reward and punishment (Russell and Norvig, 2003). RL methods have been used to solve problems such as mobile robot motion, industrial manufacturing, and combinatorial search problems for applications such as computer chess and backgammon games.

RL methods have been used to model drivers' macroscopic level decision making, such as travel and route choices of drivers (Ozbay et al., 2001; Ozbay et al. 2002; Nagel, 2004; Yanmaz-Tuzel and Ozbay, 2009; Yanmaz-Tuzel, 2010), yet less attention has been paid to modeling microscopic decision-making of vehicles in traffic simulation networks.

The majority of the available literature on using RL in traffic simulation mainly focused on efficient traffic signal control operations. It was demonstrated that the control policy governing an individual traffic signal or a set of signals can be improved in a stochastic traffic environment using RL methods (Abdulhai and Kattan, 2003). Traffic signals, acting as agents, map between traffic states and the corresponding control actions, and by continuously receiving rewards for actions taken, adjust the policy until they reach the optimal one.

For example, Wiering (2000) employed a set of multi-agent model-based RL systems for traffic light control for optimizing the signal control decision of a series of traffic signals. The developed RL systems learn the value functions by estimating the expected wait time for vehicles given different traffic light settings. In his experiment he used a small network of six signalized intersections, and reported that the RL systems can outperform non-adaptable systems. Bingham (2001) demonstrated the use of neural networks by fine-tuning the membership functions of a fuzzy traffic signal controller. In this work, the neural learning algorithm used was RL, giving rewards for successful system behavior and punishes poor behavior, where the objective of the learning is to minimize vehicle delays caused by the signal control policy. Bingham (2001) used a rather small network, and reported a reduction in delay in the range of 3-6%. Abdulhai et al. (2003) used Q-learning algorithm for optimizing the operations of an isolated two-phase traffic signal controlling the intersection of two two-lane roadways. With the use of RL, they reported delay reductions in the range of 38-44% compared

to pretimed signal control policy. Bull et al. (2004) used the learning classifier systems based RL approach for optimal signal control policy for an isolated traffic signal, and reported promising results. Bazzan et al. (2010) employed multi-agent RL for controlling a set of traffic lights. In order to manage the dimensionality problem due to increased number of states and actions, they proposed to have agents organized in groups of limited size where the number of joint actions is reduced. These formed groups were then coordinated by another agent, acting as a supervisor. Their experiment included 36 signals with 12 assigned supervisors and their results demonstrate reduced number of stopped vehicles compared to agents learning by using individual Q-learning. Arel et al. (2010) used a multi-agent RL framework to obtain an efficient traffic signal control policy. In their study they introduce a Q-learning algorithm with a feedforward neural network for value function approximation. Their numerical experiment included a small network of five signalized intersections, where the central intersection was controlled by the RL algorithm and the other four was operated using the longest-queue-first policy. It was shown that the RL-based signal significantly outperformed the other policies in terms of average delay per vehicle and average number of intersection blockings. Rezaee et al. (2012) investigated the use of RL approach for freeway ramp-metering. They employed the k-nearest neighbor (k-NN) technique to tackle the dimensionality problem. They tested their approach using a traffic simulation model of a freeway in Toronto, developed in Paramics simulation software. The results showed that their approach can reduce the total network travel time by 44% compared to the status-quo, meaning no control, and by 17% compared to ALINEA, a well-known ramp metering algorithm. Bombol et al. (2012) also investigated the benefits of using Q-learning in finding the optimal policy of an adaptive signal control. For their experimental analysis they used the simulation model of an isolated traffic signal in VISSIM simulation software, and demonstrated the reduction in average delay due to RL method compared to fixed and actuated signal control policies. Ozan (2012) and Ozan et al. (2014) used a modified RL approach for solving dynamic user equilibrium network problem. In their novel approach, a two level programming technique was employed, where in the first level dynamic user equilibrium link flows were obtained from Dynasmart-P and Dynus-T, respectively, and in the higher level signal timings are obtained by the modified RL method. The modified RL method was evaluated in a medium sized network consisting of 23 links and six signalized intersections modeled in TRANSYT-7F. System performance index, defined as the sum of a weighted linear combination of vehicle delay and number of stops per unit time was used to drive the Q-learning based RL. It was determined that the proposed approach yielded superior performance index when compared with the signal control policies determined by genetic algorithm and hill-climbing based optimization tools. In a more recent study, Ozan et al. (2015) demonstrated the use of this modified RL approach for finding optimum signal timings in coordinated signalized networks for a set of fixed link flows. El-Tantawy et al. (2013) used the multi-agent RL approach for integrated network of adaptive traffic signal controllers and demonstrated its advantages on a large-scale simulation network of Lower Downtown Toronto network modeled in Paramics simulation software. Their results showed that average intersection delay reduced by 39% and travel time savings increased by 26 % using their multi-agent RL approach.

Applying the RL approach to a different problem, Vanhulsel et al. (2009) used an extended RL algorithm to fit activity schedules based on diary data of travelers. They evaluated three distinct approaches: a generic Q-learning approach, a Q-learning approach including bucket-brigade reward distribution, which is similar to the one proposed by Holland (1976) and a RL approach improved with a regression tree-based function approximator. Their results showed that all three approaches were able to determine activity schedules that match the input schedules, and that the RL approach that employs regression tree function approximator was able to obtain a better solution than the previous two Q-learning approaches.

In the studies presented above, the common approach is to represent traffic signals as agents. Thus far, to the best of the author's knowledge, there have been a limited number of studies in the literature that utilized RL methods for simulating drivers' microscopic decisions in traffic simulation networks, where vehicles as agents learn from the feedback from their surroundings during iterative simulation runs. For example, Moriarty and Langley (1998) applied RL using the artificial neural network method to generate vehicles' lane selection strategies through trial and error interactions with the traffic environment, in an effort to demonstrate efficient intelligent vehicle and highway systems. They employed a global traffic performance function that is the difference between the actual and desired speeds averaged over several time steps and over all agents. The objective of the learning algorithm is to minimize this performance function. In their analyses, Moriarty and Langley (1998) used a customized traffic simulator that was coded for a 3.3-mile roadway stretch, a test-bed of their analyses. Later, Pendrith (2000) studied the same problem presented in Moriarty and Langley (1998), using the Q-learning method. The study area was a hypothetical 13.3 mile-long stretch of freeway with 200 cars, and was simulated using a customized simulator. Gelenbe et al. (2001) proposed learning agents that can adapt to their environment to model adaptive behavior of humans. Agents select tasks to be accomplished among a given set of tasks as the simulation progresses based on the observations of their surroundings and the information they receive from other agents. They studied the simulation of manned vehicles where the agents were assigned a goal of traversing a dangerous metropolitan grid safely and rapidly using goal based RL with neural networks. The ultimate goal of agents in this current paper, i.e. safety and wait-time, is borrowed from Moriarty and Langley (1998).

3. PROBLEM FORMULATION

An agent can observe the *state* of its surroundings, and it can choose from a set of *actions* it can perform to change its current state. Its task is to learn an optimal control strategy, usually referred to as a *policy*, for choosing actions that achieve its assigned objective. The objective of the agent can be defined by a *reward* function that assigns an immediate payoff (a numerical value) to each action the agent takes for distinct states (Mitchell, 1997).

In the context of making decisions in vehicular traffic, states are the surroundings of a driver, e.g. which lane the driver is in, where the other vehicles are located, their speeds, etc. Actions are the set of decisions a driver can make, such as turn right, left, brake, accelerate, and change lanes.

However, the dynamic nature of vehicular traffic makes the representation of the possible states and actions somewhat harder. Gap acceptance of a driver at a stop-controlled intersection is therefore a fitting problem domain, because the set of states and actions are relatively limited. For example, when a vehicle is at a stop sign searching for a possible gap to cross safely, its actions are either to (1) accept the gap or (2) reject the gap and wait for an acceptable gap

The problem domain for a driver searching for an acceptable gap consists of a population of cars on the primary roadway approaching the intersection at various speeds. The vehicle can detect within its visibility the speeds and distances of the approaching cars on the main roadway.

3.1. A Simple Case

The view of the surroundings from the perspective of a vehicle (henceforth called "agent") at a stop sign can be represented as shown in Figure 1.

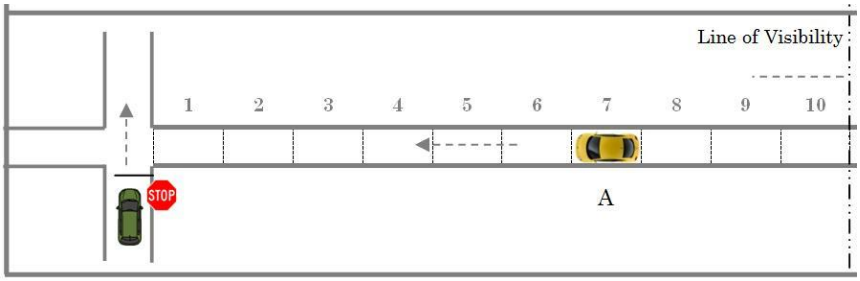


Figure 1:
Representation of an agent's surroundings as a grid network

There are three states in the simple case presented in Figure 1. The first state is the beginning of the decision process, where the agent senses the vehicle on the primary roadway, along with its location and speed. This can be named state "A". The other two possible states are mutually exclusive. The second state, in this example when the agent crosses the intersection (accepts the gap), is the *goal state* (state X). The third state is when vehicle A on the primary roadway crosses the intersection, in other words the agent rejects the gap. Because the agent cannot observe the environment beyond the line of visibility, the third state is constructed with the assumption that there is a vehicle "F" immediately after the visibility line (State F).

The definition of states in this context is different from how they are defined in the literature. The only state that is visible to agents is the one that exists in the present time (state A). All other states are constructed by agents as they predict the future events based on the dynamics of vehicles in their surroundings. Therefore, while the first state is a current (visible) state, the second and third states are the possible future states.

The state diagram of the agent is shown in Figure 2.

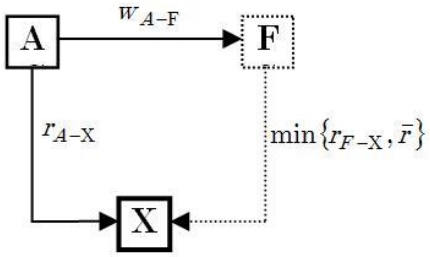


Figure 2:
State Diagram of the Agent

Rewards of each action for distinct states are shown in the diagram. If the agent decides to transfer from state A to X, accepting the gap, it will receive a reward, r_{A-X} , which denotes the risk of collision. If the agent decides to shift from state A to F, in other words if it rejects the gap, it will receive a reward, w_{A-F} , which denotes the time the agent has to wait until vehicle A clears the intersection. The next action is to transfer from state F to the goal state, X, where the associated reward is $\min\{r_{F-X}, \bar{r}\}$. $\bar{r}$ stands for the conventionally accepted risk of agents. It is updated throughout the training process as follows:

$$\bar{r} = \hat{r} \cdot \beta + r \cdot (1 - \beta) \tag{1}$$

where, $\hat{r}$ is the value of $\bar{r}$ in the previous trial, r is the risk taken in the present trial, and β is a constant $0 < \beta < 1$, preferably close to 1, indicating that the agents assign more weight to recent experiences.

The use of $\bar{r}$ prevents agents from choosing unrealistic gaps. No matter how small the gaps are at a given instance the agents know that, as human drivers do, if they wait there will be another gap that is conventionally acceptable.

The value of $\bar{r}$ changes with each action taken by the agent, and smooths out throughout the training process, and becomes stable, as shown later in the numerical analyses.

Most studies in the literature estimate drivers' gap acceptance probability using the first available gap only (Bartin et al. (2006), Mahmassani and Sheffi (1981), Teply et al. (1997), Polus et al. (2005), Hamed et al. (1997), Polus et al. (2003), Gattis and Low (1999)). One exception is the study by Daganzo (1981) where the formulation considers multiple gaps available to the driver in the decision process. Pollatschek et al. (2002) combines wait-time and risk of crossing during the decision process, but only considers the risk of the first available gap.

The premise of the proposed formulation is that the agent chooses an action based not only on available gaps but also the risks associated with the probable actions. It can either accept the gap and receive a reward, r_{A-X} , or it can wait for vehicle A to clear the intersection then cross, thus receiving rewards w_{A-F} and r_{F-X} , consecutively.

3.2. Learning Rewards

In most RL methods, the rewards of actions for distinct states are assumed to be known by the agent. In the formulation presented in this paper it is assumed that agents learn the rewards of their actions for various states by trial-and-error. For example, in Figure 1, at the beginning of the training period, the agent only knows the location of vehicle A (cell 7 in Figure 1), but it does not know the time it would take for vehicle A at cell 7 to clear the intersection, or the risk of collision if it attempts to cross. It can observe r_{A-X} when it transfers from state A to X. Similarly, it can observe w_{A-F} only when it shifts from state A to F.

Suppose that the primary roadway is represented as a grid network as shown in Figure 1, and that there are $i = 1...n$ cells within the visibility distance of the agent, with each cell of size, Δ.

Let us denote two vectors $\mathbf{w}$ and $\mathbf{c}$ both with size n. $\mathbf{w}$ and $\mathbf{c}$ store the average wait-time (w) and collision risk (r) of each cell as observed by the agent.The definition of wait-time, w, is straightforward: gap in seconds. As to risk of collision, r, the following formula developed by Ozbay et al. (2008) is employed.

$$r = \frac{\left\| \vec{V}_1^c + \vec{V}_2^c \right\|}{2} \times \frac{1}{T_{PET}} \tag{2}$$

where, $\vec{V}_1^c$ is the speed of vehicle on the secondary road crossing the intersection at conflict point (mph).$\vec{V}_2^c$ is the speed of vehicle on the primary road clearing the intersection at conflict point (mph).T_{PET} is the post-encroachment time, defined as the time lapse between vehicles on the primary and secondary road arriving at the conflict point (seconds).

Equation (2) suggests that the risk of collision becomes higher as the time lapse between vehicles at the conflict point decreases, i.e. near collision.

Vectors $\mathbf{w}$ and $\mathbf{c}$ are updated every time the agent chooses an action for a given state. The learning is terminated when:

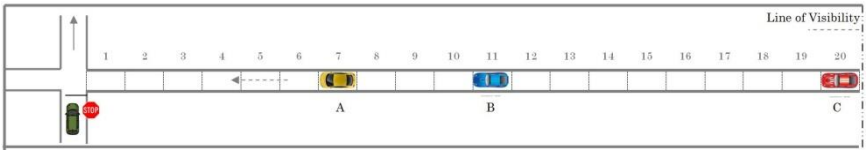
$$\begin{aligned} &\left(\mathbf{w}(i) - \hat{\mathbf{w}}(i)\right)/\hat{\mathbf{w}}(i) \le \varepsilon \qquad \forall i = 1...n \\ &\left(\mathbf{c}(i) - \hat{\mathbf{c}}(i)\right)/\hat{\mathbf{c}}(i) \le \varepsilon \qquad \forall i = 1...n \\ &(\bar{r} - \hat{r})/\hat{r} \le \varepsilon \end{aligned} \tag{3}$$

where, $\hat{\mathbf{w}}$ and $\hat{\mathbf{c}}$ are the vectors, $\hat{r}$ is the conventionally accepted risk obtained in the previous

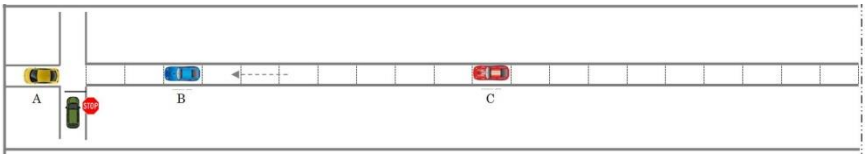
trial, and ε is the percent error.

3.3. Multiple Gaps

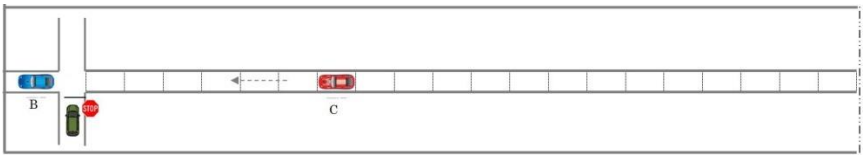
Let us now consider the case where the agent is confronted by multiple vehicles within its visibility distance, as shown in Figure 3a.



(a) First decision



(b) Second decision



(c) Third decision

Figure 3:
Representation of Multiple Vehicles on the Primary Roadway

The agent predicts the future states by using the time when the first vehicle on the primary roadway clears the intersection. In other words, by using $\mathbf{w}(7)$, the average observed gap of cell 7, it calculates the time when vehicle A clears the intersection. Figure 3(b) depicts the estimated location of vehicles B and C when vehicle A clears the intersection. Similarly, by using $\mathbf{w}(3)$, the agent constructs the future location of vehicle B, as shown in Figure 3(b).

Figure 4 shows the states and actions diagram for the initial decision of the agent.

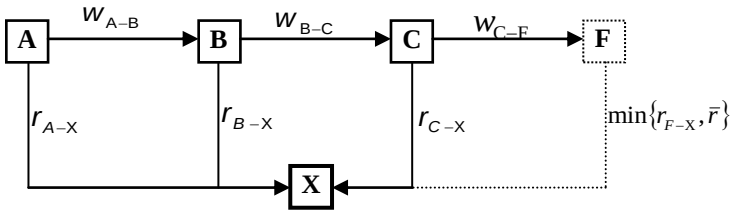


Figure 4:
State Diagram of the Agent's First Decision

3.4. Algorithm

Q-learning is a standard RL method that finds an optimal set of actions to achieve an assigned goal by learning action-value representation, instead of learning utilities. Detailed information on the Q-learning method and its applications can be found in Mitchell (1997), Russell and Norvig (2003) and Sutton and Barto (1998). In this paper, the simplest version of

Q-learning is used with undiscounted rewards. The Q-function is updated by the following algorithm (Mitchell, 1997):

For each state, *s*, and action, *a*, initialize the table entry $\hat{Q}(s,a)$ to zero.
Do Forever
Select a random initial state, *s*.
Do until reaching the goal state
From the current state *s*, select an action *a* and execute it.
- Receive an immediate reward *r*
- Observe next state s'
- Update the table entry $\hat{Q}(s,a)$ as follows:

$$\hat{Q}(s,a) \leftarrow r + \gamma \cdot \max_{a'} \hat{Q}(s',a')$$

where, γ is discount factor, $0 \le \gamma \le 1$

- Set the next state as the current state, $s \leftarrow s'$

End Do
End Do

In the Q-learning algorithm the agent learns to choose actions that will maximize its expected reward based not only on the immediate actions but also on future actions.

Note that the above algorithm assumes that the agent learns forever. For practical purposes, if the table entry $\hat{Q}(s,a)$ does not vary from one episode to another, the learning should be terminated.

It should be mentioned that the rewards in Figure 4, namely w and r , are not in the same scale. Therefore, a scale parameter, α , is used to so that r is commensurable with w . For example, while executing Q-learning for the diagram in Figure 4, the total reward of the path A-B-X (selecting the gap of vehicle B) is calculated by $w_{A-B} + \alpha \cdot r_{B-X}$.

A smaller α implies a risk-prone agent where the risk of collision has a smaller importance in choosing actions.

In the context of the presented problem, the common goal of all agents is to *minimize* the weighted combination of wait-time and collision risk. Therefore, in the algorithm presented above the rewards, w and $\alpha \cdot r$ are expressed as negative values. At the beginning of the training a small α value can be assigned to help vehicles exploring.

3.5. Gap Acceptance Decision Process

In the multiple-gap case shown in Figure 3, the agent first executes the Q-learning algorithm for the diagram shown in Figure 4. If the optimal path as calculated by the algorithm is transferring from state A to X, then it accepts the gap and updates the $\mathbf{c}$ vector. If the optimal path to the goal state X is different from A-X, then it waits for vehicle A to clear the intersection and updates the $\mathbf{w}$ vector. Once vehicle A clears the agent will confront yet another present state, similar to Figure 3(b), and predict future states accordingly. Therefore, the agent executes the Q-learning algorithm when they clear the intersection or when there is a new vehicle in the visibility distance.

The flowchart of agent's decision process is given in Figure 5.

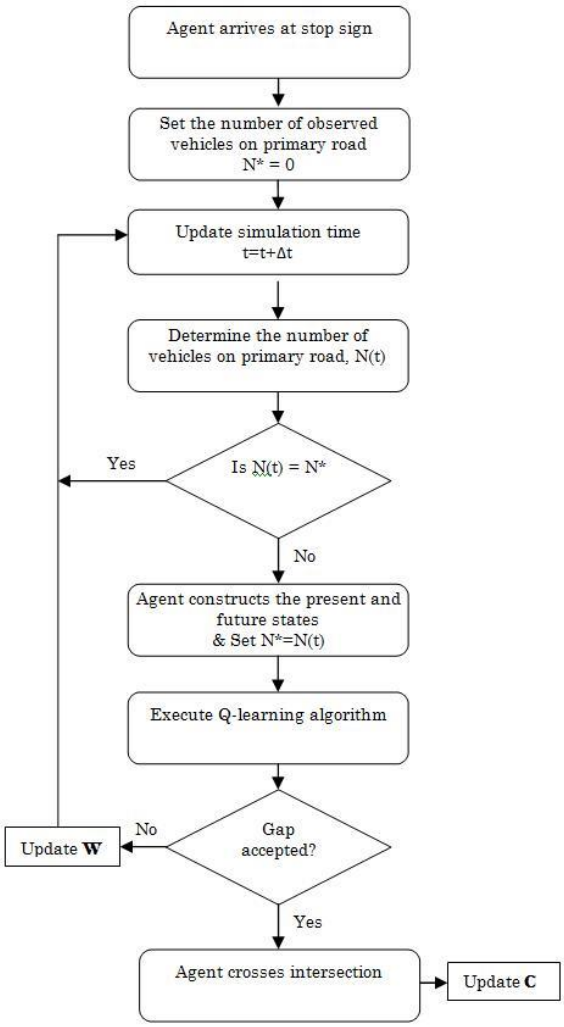


Figure 5:
Flowchart of Agent's Gap Acceptance Decision

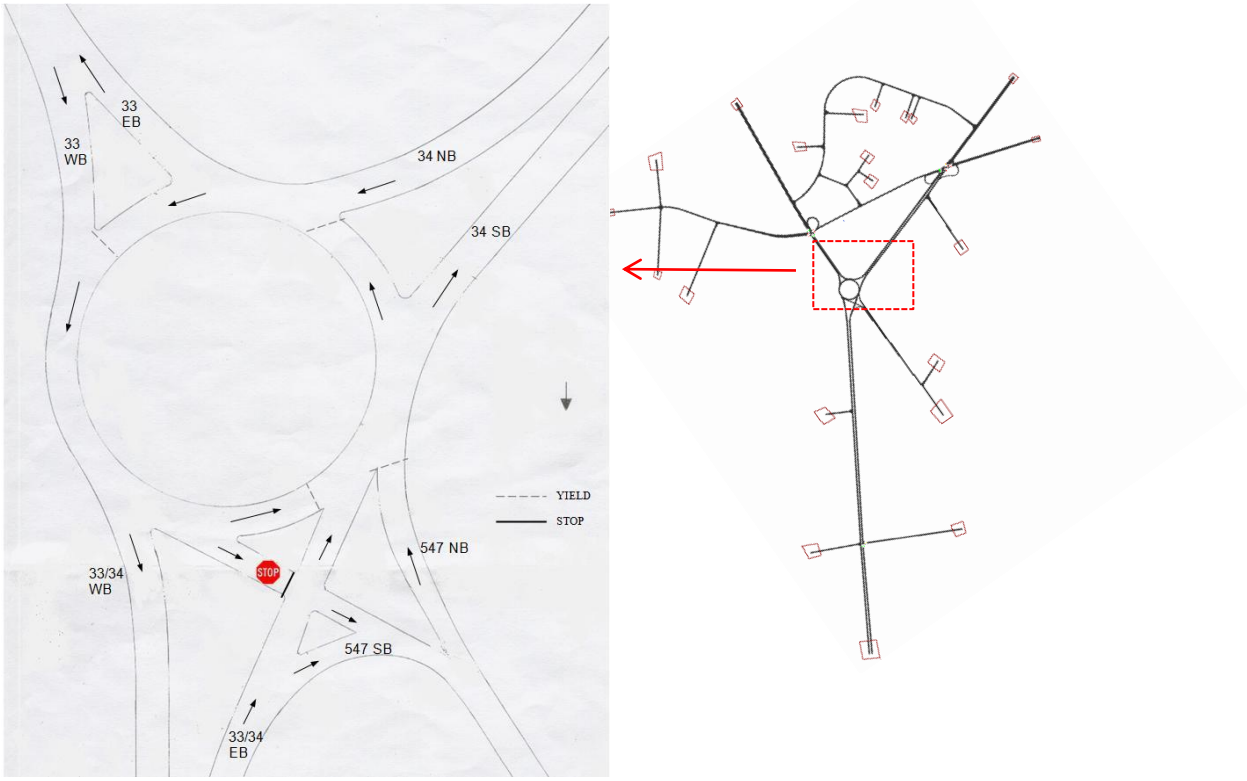
4. SIMULATION MODEL DEVELOPMENT

The simulation model used as a test-bed for this study is borrowed from Bartin et al. (2006). Below are the details of the study area, the collected field data and the description of the developed simulation model.

4.1. Study Area

Figure 6a shows the study circle, located in Wall Township, NJ. The circle has an unusual geometric and operational design. In modern roundabouts, circulating traffic has the right-of-way. However, in the case of the study circle, the traffic flows on Route 33 westbound, Route 34 northbound and Route 33/34 eastbound have the priority over the circulating traffic.

There are four yield-control intersections and one one-way stop-control intersection in the circle. The strikingly unconventional feature of the circle is the one-way stop controlled junction, where traffic comes to a full stop to exit the circle into Route 547 southbound direction. Particularly during the afternoon rush hours when the flow on Route 33/34 eastbound direction is high, drivers waiting for acceptable gaps at location 1 form a queue, which then builds up, and blocks the traffic flow on Route 33/34 westbound direction.



(a) (b)

Figure 6:
(a) Operational and geometric design of the study circle (b) Simulation Model in Paramics (Bartin et al., 2006)

4.2. Data Description

Field data collected by Bartin et al. (2006) include (1) vehicle counts with percentage distributions of trucks and passenger cars, (2) vehicle inter-arrival times at primary roads, (3) vehicle wait-times before yield signs and stop sign on secondary roads, and (4) gap acceptance/rejection times at the yield and stop signs.

Averages of wait-times and accepted gaps at the stop controlled intersection during the morning and afternoon peak periods are presented in Table 1.

Table 1. Observed data (Bartin et al, 2006)

Variable	**Morning Peak** (7 a.m. – 9 a.m.)	**Afternoon Peak** (3 p.m. – 5 p.m.)
Wait-times (sec)	7.64	13.9
Accepted Gaps (sec)	6.38	5.45

Note: Wait-times do not include queuing time

4.3. Simulation Model

The simulation model of the study circle was developed by Bartin et al. (2006) in Paramics microscopic traffic simulation software, shown in Figure 6b. The simulation model was validated / calibrated for the morning peak hours only, using the collected field data.

Bartin et al. (2006) modeled the gap acceptance behavior of vehicles at the uncontrolled intersections using a binary probit model. The variables of the probit model were (a) Accept: Dummy variable for acceptance behavior (1 if the gap is accepted, 0 otherwise) and (b) Gap: Time between consecutive vehicles at the approach. The authors developed a binary probit model for each uncontrolled intersection at the circle

Vehicles gap acceptance behaviors were then simulated using the API feature of Paramics, by controlling the movement of each vehicle within the simulation network. Basically, at each time step during simulation if a vehicle is within the link that has a yield or a stop sign, the API code checks the approach link associated with that sign. It then detects the leading vehicle on the approach link and calculates the approximate time, *g*, it would take the approaching vehicle to arrive at the junction. Thus, for every approaching vehicle the model calculates the probability of accepting the gap *g* at each location at each simulation time step using the developed probit model.

In this current study, the same simulation model is used yet the focus is the stop controlled junction. In other words, Paramics API for the stop-controlled junction is replaced with the RL approach described in Section 3, and the gap acceptance decision process shown in Figure 5 is simulated by controlling individual vehicles approaching the stop sign. The validation of the study location is performed both for the morning and the afternoon peak hours, as described in the next section.

5. VALIDATION OF THE STUDY CIRCLE

5.1. Assumptions

An initial scale parameter, α = 0.10 is used during the training runs. As stated earlier, a low initial value is assigned to α to allow the agents to explore different states and update the $\mathbf{c}$ vector faster. In other words, the agents place a low weight on risk during training and choose within the available gaps no matter how unreasonable they are, and experience "unrealistic" gaps to update the $\mathbf{c}$ vector. It should be noted that the selection of α does not affect the values of $\mathbf{w}$ and $\mathbf{c}$ vectors. The line of visibility is assumed to be 250 meters, an approximate value for this input parameter, selected based on field observations. The primary road is represented in the Q-learning algorithm by cells of size Δ = 5 meters. The constant, β, used to update $\bar{r}$ in Equation 1, is assumed 0.90.

Five vehicle categories are considered in the simulation network: one passenger car type and four truck types with varying sizes. It should be clear that vehicles with different sizes will have different collision risks, since their initial allowable acceleration abilities will affect the post encroachment times in Equation 2. Therefore, there are five different $\mathbf{c}$ vectors updated during the training runs.

It is also assumed that all vehicles within each category are homogeneous. In other words, they share the information stored in $\mathbf{c}$ vectors, making identical decision for identical state representations. However, all vehicles (both passenger cars and truck types) use the same conventionally accepted collision risk, $\bar{r}$, in the Q-learning algorithm.

5.2. Training Results

The agents are first trained in the simulation network with low demand volumes to update the $\mathbf{w}$ and $\mathbf{c}$ vectors. The demand is gradually increased until it meets the peak period demand of the study circle. The training is terminated when the conditions presented in Equation 3 are satisfied, with ε = 0.01 percent (no significant change).

Figure 7 shows the trend of $\bar{r}$ during the training runs when the constant, β, in Equation 1 is equal to 0.90, and the initial $\bar{r}$ is equal to 3,000, indicating a certain collision scenario.

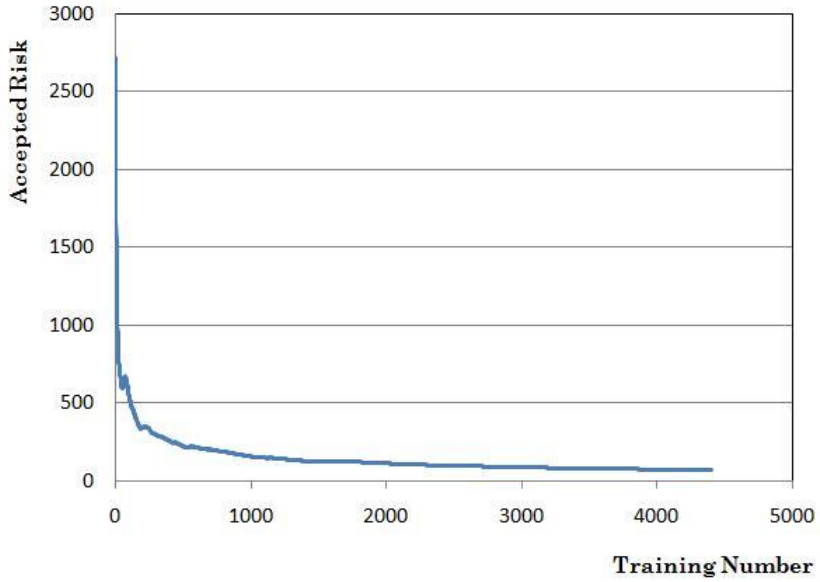


Figure 7:
Conventionally accepted risk, $\bar{r}$, during training

It should be noted that the same final value of $\bar{r}$ is obtained regardless of the initial value assigned.

5.3. Validation Results

The preliminary simulation runs showed that among the several input parameters, such as visibility, cell size, and the β constant, the scale parameter α has the highest impact on the results. Agents weigh wait-time and collision risk based on this parameter. Thus, when α is in the vicinity of 1.0, the agents experience unrealistically high wait-times leading to severe congestion. It is found that when $0.1 \leq \alpha \leq 0.2$ the simulation yields reasonable operational conditions, similar to those observed in the field data. Table 2 presents the output of simulation runs for the scale parameter, α =0.13.

Table 2. Simulation results

Variable	**Morning Peak** (7am-9am)	**Afternoon Peak** (3pm-5pm)
Wait-time (sec)	[7.16, 7.52]	[12.74,14.10]
Accepted Gaps (sec)	[6.32, 6.41]	[5.79, 5.87]

Note: The values in brackets indicate 95 percent confidence level of the output estimates.

Table 2 shows that some of the output ranges do not cover the corresponding observed output values presented in Table 1. However, it should be pointed out that the output collection in Paramics and the analyst's data extraction methods are never identical. Furthermore, it can be seen in Table 2 that none of the simulated output values are substantially out of range (less than 1 second). Therefore, the outputs of the simulation model can be assumed as valid.

Figure 8 and Figure 9 show the distributions of accepted gaps from simulation outputs and observed field data during the morning and afternoon peak periods, respectively.

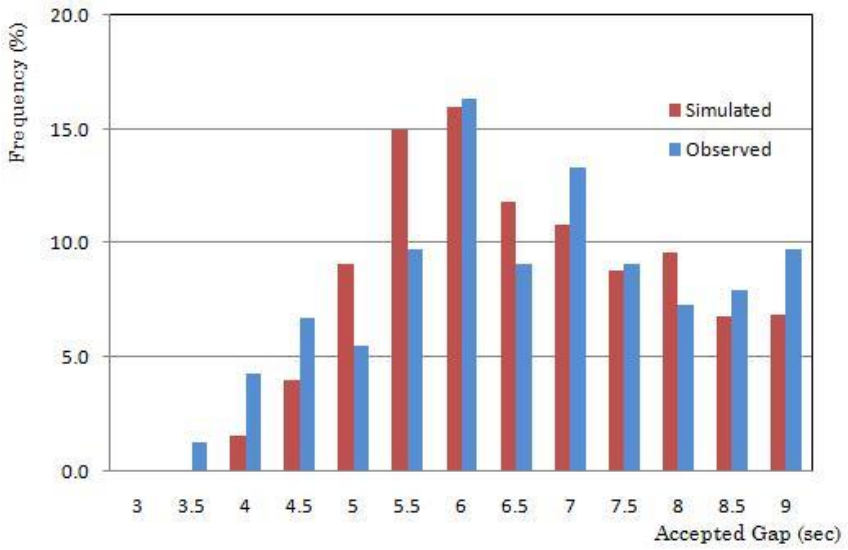


***Figure 8**:*
Distributions of accepted gaps from simulation outputs and observed data - Morning Peak

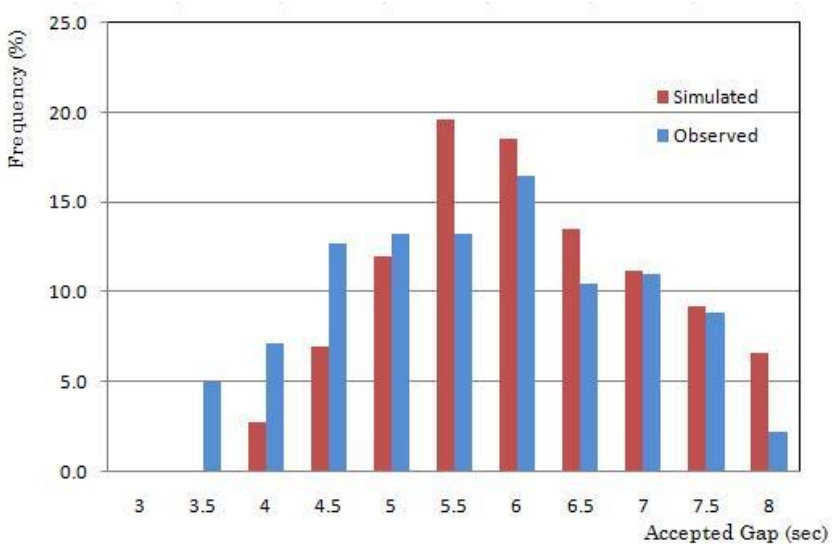


Figure 9:
Distributions of accepted gaps from simulation outputs and observed data - Afternoon Peak

These figures indicate that the distributions of accepted gaps of the agents are also in accordance with field data. The correlation coefficients of distributions are 0.86 and 0.81 for the morning and afternoon peak periods, respectively.

5.4. Discussion

Although the results shown are promising, there are several caveats to the formulation presented in this paper:

Homogeneous agents: The presented reinforcement-learning algorithm assumes only five different agent types, depending on the vehicle type (one passenger car and four truck types). Therefore, there is no variability in the decisions made while accepting gaps. In other words, when faced with identical states, agents of the same type will make identical decisions. It would be more realistic to assume different values for α and $\bar{r}$, and different visibility ranges for different agents. However, such information is not possible to gather from field data. It should

be mentioned that the similar limitation exists when the gap acceptance decision is modeled using a probabilistic model, such as the probit model used in Bartin et al. (2006).

Perception Errors: It is assumed that the agents have perfect information of the gaps on the primary roadway. However, in reality, drivers are expected to have flawed perceptions of vehicles' speeds, especially when the approaching vehicles are farther away from the intersection. Therefore, they may reject gaps that should be accepted or vice versa.

Effect of Delay on Gap Acceptance: It is also assumed that agents' decisions do not change with how long they wait at the intersection (both wait-time and queue time). One would assume that parameter α decreases as agents' delay increases, indicating that they become impatient and choose smaller gaps. However, there is not a clear consensus in the literature that supports the unambiguous effect of delay on gap acceptance. While Bartin et al. (2006), Maze (1981) and Ashton (1971) do not find significant effects from delay on gap acceptance behavior, the results presented by Mahmassani and Sheffi (1981), Polus et al. (2005) and Hamed et al. (1997) indicate otherwise.

The lack of more complex behavior explained above leaves α as the only calibration parameter, which reduces the validation / calibration process significantly. Once the training runs are complete, the value of α is changed iteratively to match the simulation outputs with the observed field data.

6. CONCLUSIONS

In this paper, the application of RL in validation/calibration of a microscopic traffic simulation model is proposed. In particular, the validity of a validation/calibration process that does not rely on any underlying model for a particular driver decision / behavior has been investigated. The underlying idea is to train drivers (agents) using the Q-learning method during the simulation until they make decisions that comply with observed driver behavior. Agents adapt to the network and events during the simulation runs, and make decisions based on assigned objectives. During training, agents' objectives are to minimize time spent traveling and to minimize risk. Agents learn from their experience and improve their decision making progressively.

The proposed approach is studied for modeling vehicles' gap acceptance decisions at a stop-controlled intersection of a traffic circle that was modeled and validated in Paramics traffic simulation software by Bartin et al. (2006). It was argued in Bartin et al. (2006) that extensive field data and time are required to estimate realistic gap acceptance models in traffic simulation models.

In this paper drivers are simulated without any notion of the outcome of their decisions. Throughout multiple episodes of gap acceptance decisions, agents learn from the outcome of their actions, i.e., wait-time and safety. Results show that by using a RL algorithm, drivers' gap acceptance behavior can easily be validated with high accuracy.

The proposed approach can be applied in simulating various other traffic facilities where complex vehicle decisions dictate the operational performance, and the data required for estimating their mathematical models are hard to collect, such as lane selection at toll plazas, lane selection ahead of an off ramp, traffic signal, merge point, etc.

The anticipated future work will include the extension of the proposed approach using other available learning methods. In specific, the author expects to adopt Learning Classifier Systems (LCS) in their future analyses. LCS ties RL and genetic algorithms. LCS was first proposed in his seminal work by Holland (1976). In general terms, LCS can be regarded as an adaptive system that learns to perform the best action based on the received input from its environment. The “best" action generally means the action that will receive the most reward or reinforcement from the environment. "Input", on the other hand, means the environment as sensed by the system (agent), usually depicted in binary values. The set of available actions by the agent depends on the decision context, for instance turn left, right, stop and so on. LCS can be thought

of a simple model of an intelligent agent in its environment. In short, the use of LCS method in simulating learning agents is anticipated to generate a "model-free" approach to validate/calibrate traffic simulation models. Therefore, as an extension to this study, vehicles' gap acceptance decisions at the studied stop-controlled junction will be simulated using the LCS method and the results will be compared.

REFERENCES

1. Abdulhai, B. and Kattan, L. (2003) Reinforcement Learning: Introduction to Theory and Potential for Transport Applications, Canadian Journal of Civil Engineering, 30, 981-991. doi: 10.1139/l03-014

2. Abdulahi, B., Pringle, R. and Karakoulas, G. J. 2003. Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering. Vol. 129. No.3. pp. 278-285. doi: 10.1061/(ASCE)0733-947X(2003)129:3(278)

3. Arel, I., Liu, C., Urbanik, T. and Kohls, A. G. (2010) Reinforcement learning based multi-agent system for network traffic signal control, IET Intelligent Transportation Systems, 4(2), 128–135. doi: 10.1049/iet-its.2009.0070

4. Ashton, W. D. (1971) Gap acceptance problems at a traffic intersection, Applied Statistics, 20(2), 130-138. doi: 10.2307/2346461

5. Bartin, B., Ozbay, K., Yanmaz-Tuzel, O. and List, G. (2006) Modeling and Simulation of Unconventional Traffic Circles, *Transportation Research Journal: Journal of the Transportation Research Board,* 1965, 201-209. doi: 10.3141/1965-21

6. Barton, R. R., and Schruben, L. W. (2001) Resampling methods for input modeling, Proceedings of the 2001 Winter Simulation Conference, 1, 372–378. doi: 10.1109/WSC.2001.977303

7. Bazzan, A. L. C., Oliveira, D. and Silva, B. C. (2010) Learning in groups of traffic signals, *Engineering Applications of Artificial Intelligence*, 23, 560-568. doi: 10.1016/j.engappai.2009.11.009

8. Bingham, E. (2001) Reinforcement learning in neurofuzzy traffic signal control, *European Journal of Operation Research*, 131, 232-241. doi: 10.1016/S0377-2217(00)00123-5

9. Bombol, K., Koltovska, D. and Veljanovska, K. (2012) Application of reinforcement learning as a tool of adaptive traffic signal control on isolated intersections, *IACSIT International Journal of Engineering and Technology*, 4(2), 126 -129. doi: 10.7763/IJET.2012.V4.332

10. Bull, L., Sha'Aban, J., Tomlinson, A., Addison, J.D. and Heydecker, B. G. (2004) Towards distributed adaptive control for road traffic junction signals using learning classifier systems, In: Bull, L, (ed.) Applications of Learning Classifier Systems, 279-299. Springer: New York. doi: 10.1007/978-3-540-39925-4

11. Daganzo, C. (1981) Estimation of gap acceptance parameters within and across the population from direct roadside observation, *Transportation Research Part B*, 15B, 1-15. doi: 10.1016/0191-2615(81)90042-4

12. Dowling, R., Skabardonis, A. and Alexiadis, V. (2004) Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software, FHWA Contract DTFH61-01-C-00181, FHWA.

13. EI-Tantawy, S., Abdulhai, B. and Abdelgawad, H. (2013) Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC):

Methodology and Large-Scale Application on Downtown Toronto, *IEEE Transactions on Intelligent Transportation Systems*, 14(3), 1140-1150. doi: 10.1109/TITS.2013.2255286

14. Gattis, J. L. and Low, S. (1999) Gap acceptance at atypical stop-controlled intersections, *Journal of Transportation Engineering*, 123(3), 201-207. doi: 10.1061/(ASCE)0733-947X(1999)125:3(201)

15. Gelenbe, E. Seref, E. and Xu, Z. (2001) Simulation with learning agents, *Proceedings of the IEEE*, Vol. 89 (2), 148-157. doi: 10.1109/5.910851

16. Hamed, M. M., Easa, S. M. and Batayneh, R. R. (1977) Disaggregate gap-acceptance model for unsignalized T-intersections, *Journal of Transportation Engineering*, 123(1), 36-42, doi: 10.1061/(ASCE)0733-947X(1997)123:1(36)

17. Holland, J. H. (1976) Adaptation, In Rosen & Snell (eds) Progress in Theoretical Biology, 4. Plenum.

18. Iyer, S., Ozbay, K. and Bartin, B. (2010) Ex Post Evaluation of Calibrated Simulation Models of Significantly Different Future Systems, *Transportation Research Record: Journal of the Transportation Research Board*, 2161, 49-56. doi: 10.3141/2161-06

19. Mahmassani, H. and Sheffi, Y. (1981) Using gap acceptance sequences to estimate gap acceptance functions, *Transportation Research Part B*, 15B, 143-148. doi: 10.1016/0191-2615(81)90001-1

20. Maze, T. (1981) A probabilistic model of gap acceptance behavior, *Transportation Research Record*, 795, 8-13.

21. Mitchell, T. M. (1997) Machine Learning, McGraw Hill Higher Education.

22. Moriarty, D. E. and Langley, P. (1998) Learning cooperative lane selection strategies for highways, Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, 684-691, July, Madison, Wisconsin, United States.

23. Nagel, K. (2004) Route learning in iterated transportation studies, *Human Behaviour and Traffic Networks*, 305-318. doi: doi.org/10.1007/978-3-662-07809-9

24. Ozan, C. (2012) Dynamic User Equilibrium Urban Network Design Based on Modified Reinforcement Learning Method" (in Turkish), PhD Thesis. Pamukkale University, Science and Technology Institute, Civil Engineering Department, Transportation Division, Denizli, Turkey.

25. Ozan, C., Ceylan, H. and Haldenbilen, S. (2014) Solving network design problem with dynamic network loading profiles using modified reinforcement learning method, Proceedings of the 16th Meeting of the EURO Working Group on Transportation, Procedia - Social and Behavioral Sciences, 111, 38-47. doi: 10.1016/j.sbspro.2014.01.036

26. Ozan, C., Baskan, O., Haldenbilen, S. and Ceylan, H. (2015) A modified reinforcement learning algorithm for solving coordinated signalized networks, *Transportation Research Part C: Emerging Technologies*, 54, 40-55. doi: 10.1016/j.trc.2015.03.010.

27. Ozbay, K., Datta, A. and Kachroo, P. (2001) Modeling Route Choice Behavior Using Stochastic Learning Automata, *Transportation Research Record*, 1752, 38-46. doi: 10.3141/1752-06

28. Ozbay, K., Datta A. and Kachroo, P. (2002) Application of Stochastic Learning Automata for Modeling Departure Time and Route Choice Behavior. Transportation Research Record, 1807, 154-162. doi: 10.3141/1807-19

29. Ozbay, K., Yang, H., Bartin, B. and Mudigonda, S. (2008) Derivation and validation of a new simulation-based surrogate safety measure, *Transportation Research Record*, 2083, 103-113. doi: 10.3141/2083-12

30. Paramics Website. Access address: http://www.paramics-online.com/ (Accessed on April 7, 2017)

31. Pendrith, M. D. (2000) Distributed reinforcement learning for a traffic engineering application, Proceedings of the fourth international conference on Autonomous agents, 404-411, June 03-07, Barcelona, Spain. doi: 10.1145/336595.337554

32. Pollatschek, M.A., Polus, A. and Livneh, M. (2002) A Decision Model for Gap Acceptance and Capacity at Intersection, *Transportation Research Part B*, 36, 649-663. doi: 10.1016/S0191-2615(01)00024-8

33. Polus, A., Lazar, S. S. and Livneh, M. (2003) Critical gap as a function of waiting time in determining roundabout capacity, *Journal of Transportation* Engineering, 129(5), 504-509. doi: 10.1061/(ASCE)0733-947X(2003)129:5(504)

34. Polus, A., Shiftan, Y., and Shmueli-Lazar, S. (2005) Evaluation of the waiting-time effect on critical gaps at roundabouts by a logit model, *European Journal of Transport and Infrastructure Research*, **5**(1), 1-12.

35. Rezaee, K., Abdulahi, B. and Abdelgawad, H. (2012) Application of reinforcement learning with continuous state space to ramp metering in real-world conditions, 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, Alaska, USA. doi: 10.1109/MITS.2012.2217592.

36. Russell, S. J. and Norvig, P. (2003) Artificial intelligence: A modern approach, Prentice Hall series in artificial intelligence. Upper Saddle River, N.J.: Prentice Hall/Pearson Education.

37. Sacks, J., Rouphail, N. M., Park, B., Thakuriah, P., Rilett, L R., Spiegelman, C. H. and Morris, M. D. (2002) Statistically-Based Validation of Computer Simulation Models in Traffic Operations and Management, *Journal of Transportation and Statistics*, 5(1), 1-24.

38. Sutton, R. S. and Barto, A.G. (1998) Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA.

39. Teply, S., Abou-Henaidy, M. and Hunt, J. D. (1997) Gap acceptance behavior – aggregate and logit perspectives: Part 1, *Traffic Engineering and Control*, 37(9), 474-482.

40. Vanhulsel, M., Janssens, D., Wets, G. and Vanhoof, K. (2009) Simulation of sequential data: An enhanced reinforcement learning approach, Expert Systems with Applications. 36, 8032-8039. doi: 10.1016/j.eswa.2008.10.056

41. Yanmaz-Tuzel, O. (2010) Modeling traveler behavior via day-to-day learning dynamics, Ph.D. Thesis, Rutgers, The State University of New Jersey.

42. Yanmaz-Tuzel, O. and Ozbay, K. (2009) Chapter 19: Modeling Learning Impacts on Day-to-Day Travel Choice, Transportation and Traffic Theory 2009: Golden Jubilee, 387-403. doi: 10.1007/978-1-4419-0820-9_19

43. Wiering, M.A. (2000) Learning to control traffic lights with multi-agent reinforcement learning, First World Congress of the Game Theory Society, Utrecht, Netherlands, Basque Country University and Foundation, Spain.