Unified Route Choice Framework and Empirical Study in Urban Traffic Control Environment

Xiao-Feng Xie
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
xfxie@cs.cmu.edu

Yiheng Feng
Department of Systems and Industrial Engineering
The University of Arizona
Tucson, AZ 85721
yihengfeng@email.arizona.edu

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@cs.cmu.edu

K. Larry Head
Department of Systems and Industrial Engineering
The University of Arizona
Tucson, AZ 85721
klhead@email.arizona.edu

6249 words + 4 figures + 1 tables

November 15, 2013

ABSTRACT

Route choice systems (RCS) and traffic control systems (TCS) constitute two major approaches to mitigating congestion in urban road networks. The interaction between signal control and route choice is often captured in the combined traffic assignment and control problem, whose solution has many practical applications. In this paper, we consider this interaction from a narrower RCS perspective, and focus on distributed route choice models for operational rather than planning purposes. Our goal is to analyze the relative performance of alternative route choice models as different assumptions are made about the type of TCS in use in the urban road network. To this end, we define a unified agent-based framework for formulating different route choice models, and integrate this framework with a microscopic traffic simulation environment. Within our framework, each agent's memory is updated repeatedly (daily) to reflect available prior individual and social experience, and then a route is chosen by a probabilistic sequential decision-making process that is a function of the agent's updated current memory. Several previously developed route choice models from the literature are implemented using the framework, and their performance, along with some additional hybrid models that are suggested by the modeling framework, is evaluated on two simulated real-world TCSs: (1) a 32-intersection road network in Pittsburgh, PA running a fixed, SYNCHRO-generated coordinated signal control plan, and (2) the same road network running with the SURTRAC adaptive TCS. The results show that specific route choice models perform differentially when applied in conventional and adaptive traffic signal control settings, and that better overall network performance for all route choice models is achieved in the adaptive signal control setting. Our unified framework also makes it possible to analyze the performance impact of route choice model components, and to formulate better performing hybrid models.

INTRODUCTION

In urban road networks, traffic congestion is a major problem, resulting in significant costs for drivers through wasted time and fuel, detrimental impact to the environment due to increased vehicle emissions, and increased needs for infrastructure upgrades.

It is generally recognized that traffic control systems (TCSs) (*1, 2, 3, 4*) and route choice systems (RCSs) (*5, 6, 7, 8*) are two major technologies for dealing with traffic congestion. In any signalized road network, the delay incurred by a TCS at intersections significantly affects the cost of travel through the network, and the collective route choices of drivers utilizing the network produce flow patterns, which in turn influence the design of the TCS and its control plans. From a planning perspective, this loop is commonly referred to the combined traffic assignment and control problem (*9, 10*). Broadly, techniques for solving this problem have many practical applications, e.g., in supporting policy evaluation in urban planning, in providing route guidance operations, and in identifying optimal combinations of signal settings and routing patterns (*10*).

In this paper, we consider the interaction between TCSs and RCSs from the perspective of agent-based route choice models, which builds on concepts from game theory. The problem is formulated as a congestion game (*11, 12, 13, 14, 15*), a form of commuting problem where a set of agents (drivers) must repeatedly select travel routes from an origin location to a destination location. The set of possible routes are shared resources, and the cost to each driver depends on the route chosen and the number of others that have chosen the same route in the same time period. The game is repeated day by day, and drivers try to adapt their route choices according to the available information. The user equilibrium (UE) and system optimal (SO) represent performance optimums in the fully noncooperative and fully cooperative cases, respectively. However, these optimums assume that all agents have complete information and make perfect decisions, which is rarely achievable in the real world. Due to their perception errors, agents should consider an imperfect, or noisy, rational expectations equilibrium (*16*). As also indicated by early work in stochastic UE (*6, 7*) and boundedly rational UE (*17*), agents still have a tendency toward cost minimization, but not necessarily choose the lowest. Instead, agents try to quickly adapt to the set of correlated equilibria (*13*), where the probability distribution over the choice set is obtained using shared information. Based on a concept of equilibrium similar to that adopted in the stochastic traffic assignment literature, an agent-based framework has the power to naturally model heterogeneous individual behavior under dynamic flow conditions, although there is an additional challenge for the system to reach desired equilibria based on (myopic) individual decisions, under essentially decentralized operating conditions.

Various route choice models (RCMs) have been proposed. In the regret matching strategy (*13*), agents may depart from the current choice with probabilities proportional to their *regret* for not choosing other particular choices in the past steps. This simple adaptive strategy can converge to a correlated equilibrium. The exploration-replication policy (ERP) (*12*) is a distributed routing policy that approximates the Wardrop equilibrium. It uses an adaptive sampling rule that is inspired by replicator dynamics, which amplifies the choice of low-cost routes. To reduce the computational burden of fictitious play (FP), the fading memory joint strategy FP (JSFP) is proposed (*14*), and convergence is established to a pure Nash equilibrium in potential games. The linear reward-inaction (LRI) algorithm (*11*) is used for distributed learning of the equilibrium in a linear Wardrop game. In discrete choice theory (*5*), multinomial logit (MNL) models have been extensively studied, and some variants can handle the underlying i.i.d. assumption to some extent (*18*). In (*19*), an agent-based model is proposed to account for the heterogeneity of different drivers using the Dirichlet distribution, the conjugate prior of the multinomial distribution, where the concentration parameters are obtained over time using Bayesian learning. This model has been shown to converge to a UE and is able to handle flow disruptions in a single-commodity network.

The theoretical convergence properties of all of the above models have been studied and demonstrated in idealized settings, and little attempt has been made to understand their performance in real-world traffic networks. Our goal in this paper is to investigate this general question, with particular emphasis on how various route choice models interact with different traffic control systems or plans. To this end, we define a unified agent-based framework for formulating different (distributed) route choice models, and integrate this framework with a microscopic traffic simulation environment. Within our framework, each agent's memory is updated repeatedly (daily) to reflect available prior individual and social experience, and then a route is chosen by a probabilistic sequential decision-making process that relies on the agent's updated memory. A set of existing route choice models are implemented using the framework, along with some additional hybrid models, and these models are then empirically evaluated in two simulated real-world settings: (1) a 32-intersection road network in Pittsburgh, PA running with a fixed, SYNCHRO-generated coordinated signal control plan and (2) the same road network running with the SURTRAC adaptive traffic control system (*3, 4, 20*). Furthermore, our unified framework makes it possible to separately analyze the performance impact of route choice models and their components, and provides a basis for formulating better performing hybrid route choice models.

PROBLEM FORMULATION

For a traffic network, overall performance is determined by the interaction of two basic systems. The TCS allocates green times for vehicular flows passing through intersections; and the RCS dynamically assigns routes for vehicles traveling on the roads in the network.

The TCS is used for controlling traffic lights at intersections. For each intersection with a set of entry and exit roads, the traffic light cycles through a sequence of phases, and each phase governs the right of way for a set of compatible movements from entry to exit roads. For traffic control, a *signal sequence* contains a sequence of phases and associated durations. For the phase switching process, there are timing constraints for safety and fairness: the yellow light after each phase runs for a fixed duration, while each phase has a duration that can range between a minimum and maximum. Various TCSs (*1, 2, 3*) have been proposed to minimize the travel time for drivers.

Let the traffic network be represented by a directed graph $G = (V, E)$, which contains a set of nodes (intersections) V and a set of edges (roads) E. Let RS be the set of origin-destination (O-D) pairs (or commodities (*11*)), and $rs \in RS$ represents an O-D pair. Let K_{rs} be a fixed set of routes between an O-D pair rs, where each route contains a set of adjacent edges that connect rs. The route sets for all O-D pairs can be pre-generated by some strategies (*7, 8, 21*) [1]. Let A be a set of drivers (vehicles), of whom each travels between a given O-D pair rs, departing at a given time t. The RCS is used for helping each driver $a \in A$ to decide his route $k \in K_{rs}$. Then these vehicles contribute to dynamic traffic flows on the edges over time.

The travel time on a route is equal to the sum of the travel times of its edges. As in a congestion game (*11, 14, 16*), each edge $e \in E$ is associated with a non-decreasing latency function $l_e(x) \geq 0$, where $x \geq 0$ is the flow on edge e for a specific time period. Analytic models, e.g.,

[1] A classification of generation models can be found in (*7*). The route sets should include most efficient routes, and some alternative routes to bypass possibly congested links, for different flow situations in the network.

the Bureau of Public Roads (BPR) function (*18*), provide a suitable means for calculating l_e for planning applications. However, in the presence of TCS, travel flows are not continuous, and the traffic network is a complex system with nonlinear dynamics. To be more realistic in this operational setting, a microscopic traffic simulator is used for the travel time calculation.

We consider a day-to-day evolution process. On day n, each driver can decide to stay on the same route as the previous trip, or change to an alternative route, prior to his departure. Let tt_{an} be the travel time of the driver a (on day n), the average travel time of all drivers (TT_n) is

$$TT_n = \sum_{a \in A} tt_{an}/|A|. \quad (1)$$

This evolution process is similar to the combined traffic assignment and control problem (*9, 22*). In this paper, we focus more on the side of evaluating different strategies in a unified framework of RCS, which feed their outcomes as presumably optimized input instances for TCS, and gain limited experience from the outcomes of TCS day by day. Nevertheless, the significance of the optimization capability of TCS in the whole loop will be evaluated as well.

ROUTE CHOICE FRAMEWORK

For RCS, it is natural to consider an decentralized, agent-based framework, where route choices are made by *agents*. Agent-based modeling promotes situatedness, robustness, and scalability. It is natural in the era of mobile computing, as smart phones and in-vehicle navigation systems are broadly adopted. We assume there is no explicit central coordination, although agents can access limited social information, which might be available from information service providers (ISPs) through wireless or vehicle-to-infrastructure (V2I) communication.

Basic Framework

Specifically, the route choice framework contains an ISP [2] and a set of autonomous agents. For each agent, the basic decision making capability arises from the interaction between memory and behavior (*23*), under the bounded rationality assumption (*24*). Each day, each agent first manages its *memory* by a *memory updating process* over a set of *updating rules* using available individual and social experience, then chooses a route by a *probabilistic sequential decision making process* over a set of *decision rules* that are instantiated with the memory elements of specific *types*.

This framework is designed to be extensible, where memory elements and (updating and decision) rules [3] are basic *components* that can be added/removed.

All agents are homogeneous in that they possess the same components. However, the agents are essentially heterogeneous during the day-by-day execution. First, they might have significantly different memory instances. For example, the individual travel times are different not only due to different departure times but also because of inherent disturbance in the complex network with TCS. Second, the stochastic nature in their run-time decisions will lead to the execution of different decision rules that work on different memory elements (each encodes individual knowledge).

During each day, each agent $a \in A$ chooses a route $k \in K_{rs}$ for a specific O-D pair rs, and departs at a specific time $t \in [1, T]$. On day $(n+1)$, an agent can obtain limited social information from an ISP and its own travel experience in day n. Let A_{rs} be the group of agents traveling between

[2] Using a single ISP is for the simplicity of description. Multiple ISPs can be naturally considered. Furthermore, an ISP can provide rather accurate social information, e.g., $(\widehat{\mathbf{TT}}_n^{rs}, \widehat{\mathbf{FF}}_n^{rs})$, given a reasonable large sample size.

[3] In this paper, all rules are represented using the generic type "R", and defined with specific input parameters.

rs. Each agent $a \in A_{rs}$ has its own travel experience $(\widehat{tt}_{an}^{rs}, \widehat{k}_{an}^{rs})$, i.e., the individual travel time and the chosen route during day n. The ISP will provide it with the social information $(\widehat{\mathbf{TT}}_n^{rs}, \widehat{\mathbf{FF}}_n^{rs})$, i.e., the travel times and the proportions for all routes between rs, for the agents in A_{rs}.

In the following description, we will focus on the components related to the decision making of each agent $a \in A_{rs}$ during day $(n+1)$. For simplicity of description, the symbols a (for an agent), rs (for an O-D pair), and n (for a day) will be dropped if the information is not necessarily.

Memory and Updating Process

Memory is a basic component for supporting the learning process. Specifically, memory is used for storing and retrieving a fixed list of elements, where each element contains certain historical experience of a specific *type*. As in a Markov chain process, memory elements are initialized and then updated day by day. Different memory elements might hold different properties, according to their types and how are they updated in the day-by-day process.

Here some generic types used in existing route choice models are considered. Let y and $\mathbf{Y}$ be a single value for a specific route and an array, respectively, that are defined on all routes in K_{rs}. The only subtype of y considered here is a generalized *cost* [4] value c. Three subtypes of $\mathbf{Y}$ are considered: $\mathbf{C}$ is a generalized *cost array*, where $C(k) \geq 0$ indicates the cost for route k. $\mathbf{F}$ is a *frequency array*, where $F(k) > 0$ and $\sum_{k \in K_{rs}} F(k) = 1$. We also consider an array of concentration parameters $\mathbf{D}$ used in Dirichlet distribution (*19*), where $D(k) > 0$.

We first introduce a list of updating rules that work on these generic types, and then use them to update specific elements in the memory, with specific individual and social experience.

Updating Rules

For time series data y, the new value $\tilde{y}$ of the exponential moving average (EMA) is obtained by updating an current EMA value $\tilde{y}$ with the new value y, with the coefficient $\alpha \in [0,1]$:

$$\tilde{y} = R_{EMA}(\tilde{y}, y|\alpha) = \alpha \cdot y + (1-\alpha) \cdot \tilde{y}. \tag{2}$$

The EMA function can be easily realized in the array fashion, i.e.,

$$\widetilde{\mathbf{Y}} = R_{EMA}^{vec}(\widetilde{\mathbf{Y}}, \mathbf{Y}|\alpha), \tag{3}$$

by applying Eq. 2 on individual elements of the EMA data array $\widetilde{\mathbf{Y}}$ and the data array $\mathbf{Y}$.

The $R_{EMA}^{index}(\widetilde{\mathbf{Y}}, y, k'|\alpha)$ function only updates the dimension k' of $\widetilde{\mathbf{Y}}$, which has

$$\widetilde{Y}(k') = R_{EMA}(\widetilde{Y}(k'), y|\alpha). \tag{4}$$

The *normalization* functions R_{NORM}^{sum} and R_{NORM}^{min} obtain scaled arrays from an array $\mathbf{Y}$ by respectively dividing each of its elements with the sum of all its elements $\sum_{k \in |K_{rs}|} Y(k)$ and the minimal value $\min_{k \in |K_{rs}|} Y(k)$. They will just do nothing if the denominator is zero.

There are some *indexing*-style functions. The *better-indexing* function, i.e., $R_{BI}(\mathbf{C}, c)$, returns an array $\mathbf{Y}$, where $Y(k) = 1$ if $C(k) < c$ for $\forall k$, and $Y(k) = 0$ otherwise. The *current-indexing* function, i.e., $R_{CI}(k')$, returns an unit array $\mathbf{Y}$, where $Y(k') = 1$ and $Y(k) = 0$ for $\forall k \neq k'$.

Assuming $\mathbf{F} = R_{MNL}(-\theta \cdot \mathbf{D})$ (the multinomial logit rule R_{MNL} will be introduced later in Eq. 8), where $\theta > 0$ is the dispersion parameter, the *reverse-fitting* rule, i.e., $R_{RF}(\mathbf{F})$, returns an

[4] Note that *cost* and *utility* values can be mutually transformed in simple ways.

array of concentration parameters $\mathbf{D}$ by reversely fitting from a frequency array $\mathbf{F}$, i.e.,

$$D(k) = \log(\exp(\theta) \cdot F(k)/F_{min})/\theta, \quad \text{for } \forall k \in K_{rs}, \tag{5}$$

where F_{min} is the minimal value in $\mathbf{F}$.

In the *linear reward-inaction* updating rule (*11*), i.e., $R_{LRI}(\mathbf{F}, \mathbf{C}, \widehat{k})$, the frequency array $\mathbf{F}$ is updated using a normalized cost array $\mathbf{C}_{norm} = R^{sum}_{NORM}(\mathbf{C})$ on all routes, and $\mathbf{Y} = R_{CI}(\widehat{k})$:

$$F(k) = F(k) + \beta \cdot (1 - C_{norm}(k)) \cdot (Y(k) - F(k)), \tag{6}$$

where $\beta \in (0,1)$ is a precision parameter.

Memory Updating Process

The memory updating process initializes and updates the memory using individual experience $(\widehat{tt}, \widehat{k})$ and social information $(\widehat{\mathbf{TT}}, \widehat{\mathbf{FF}})$. Individual experience is available each day, whereas social information is available on the first day but only with the probability γ afterward. In this paper, we simply set $\gamma = 1$. One might set $\gamma < 1$ to accommodate the fact that the access to social information may be limited. The types of $\widehat{tt}$, $\widehat{\mathbf{TT}}$, and $\widehat{\mathbf{FF}}$ are c, $\mathbf{C}$, and $\mathbf{F}$, respectively.

Table 1 gives the implementation for the memory updating process. Here the memory contains five elements, i.e, $\{\widetilde{tt}, \widetilde{\mathbf{TT}}, \widetilde{\mathbf{FF}}, \widetilde{\mathbf{F}}_{LRI}, \widetilde{\mathbf{D}}\}$, with the types shown in the fifth column. Each element will be initialized by the obtained data in the second column. Afterward, the updating processes are activated row by row, with probability in the fourth column. Each element might be updated by multiple updating rules. For each updating process, its inputs might come from new experience (either individual or social experience) and the memory elements in upper rows, if available. One might add elements with other properties using different updating rules.

Here $\widetilde{tt}$, $\widetilde{\mathbf{TT}}$, and $\widetilde{\mathbf{FF}}$ are information sources, which are expected to have small perception errors over time. The updating of $\widetilde{tt}$ and $\widetilde{\mathbf{TT}}$ is more agent-oriented to account for the individual difference, whereas the updating of $\widetilde{\mathbf{FF}}$ is more social-oriented for a less-biased estimation.

For each agent, $\widetilde{\mathbf{F}}_{LRI}$ and $\widetilde{\mathbf{D}}$ are updated directly and indirectly through MNL to represent route choice probabilities. During the initialization, $\widetilde{\mathbf{FF}}$ provides a "warm-start" for $\widetilde{\mathbf{F}}_{LRI}$ directly and for $\widetilde{\mathbf{D}}$ using R_{RF}. Both of them are updated over time, which represents the basic learning mechanism toward a form of (correlated) equilibrium with other agents.

TABLE 1 : Implementation for the memory updating process

Memory	Initialization	Updating Process	Probability	Type
$\widetilde{tt}$	$\widehat{tt}$	$R_{EMA}(\widetilde{tt}, \widehat{tt} \mid \alpha = 0.5)$	1.0	c
$\widetilde{\mathbf{TT}}$	$\widehat{\mathbf{TT}}$	$R^{vec}_{EMA}(\widetilde{\mathbf{TT}}, \widehat{\mathbf{TT}} \mid \alpha = 0.01)$	γ	$\mathbf{C}$
		$R^{index}_{EMA}(\widetilde{\mathbf{TT}}, \widehat{tt}, \widehat{k} \mid \alpha = 0.5)$	1.0	
$\widetilde{\mathbf{FF}}$	$\widehat{\mathbf{FF}}$	$R^{sum}_{NORM}(R^{vec}_{EMA}(\widetilde{\mathbf{FF}}, \widehat{\mathbf{FF}} \mid \alpha = 0.5))$	γ	$\mathbf{F}$
		$R^{sum}_{NORM}(R^{vec}_{EMA}(\widetilde{\mathbf{FF}}, R_{CI}(\widehat{k}) \mid \alpha = 0.01))$	1.0	
$\widetilde{\mathbf{F}}_{LRI}$	$\widehat{\mathbf{FF}}$	$R^{sum}_{NORM}(R_{LRI}(\widetilde{\mathbf{F}}_{LRI}, \widetilde{\mathbf{TT}}, \widehat{k} \mid \beta = 0.01))$	1.0	$\mathbf{F}$
$\widetilde{\mathbf{D}}$	$R_{RF}(\widehat{\mathbf{FF}})$ $(\theta = 0.05)$	$R^{vec}_{EMA}(\widetilde{\mathbf{D}}, R_{RF}(\widehat{\mathbf{FF}}) \mid \alpha = 0.01)$	γ	$\mathbf{D}$
		$R^{min}_{NORM}(\widetilde{\mathbf{D}} + R^{sum}_{NORM}(R_{BI}(\widetilde{\mathbf{TT}}, \widehat{tt})))$	1.0	

Each agent can adjust the weights of social information through adjusting α values by using moving averages to update memory. The balance of using individual and social knowledge due to different updating behavior in the learning process might be an interesting topic in future research.

Probabilistic Sequential Decision Making Process

The probabilistic sequential decision making process (PS-DMP) returns a route $k \in K_{rs}$ using the elements in the memory. PS-DMP contains an ordered list of sub-processes, i.e., [sp_1, ..., sp_m ..., sp_M], in which each sub-process sp is described in a tuple (*Decision Rule, Probability*). Each *decision rule* either selects a route k in the route set or returns *undecidable* if it cannot find one.

PS-DMP then runs through the sequence of its sub-processes from sp_1 to sp_M. For each sp, its decision rule instance is executed with the associated probability. Suppose the current sub-process is sp_m, the total process is terminated and returns k if the current sp_m returns a valid route $k \in K_{rs}$. Otherwise, it means that the decision rule of sp_m is not executed (with the associated probability) or returns *undecidable*, and the process continues to execute the next sub-process sp_{m+1}. PS-DMP simple returns $\widehat{k}$ if the final decision of sp_M is still *undecidable*.

PS-DMP follows the style of fast and frugal heuristics (*24*). Each decision rule is (fast and) frugal based on limited information. PS-DMP is especially focused on supporting both sequential and probabilistic parallel execution of multiple decision rules. Sequential execution is supported by allowing rules to be *undecidable*, where some of them might only work well within a small sub-space in the decision space. Probabilistic parallel execution is supported for rules with associated probability values, and these rules can cooperate (through the memory) across iterations (*25*).

We will first introduce decision rules, and then describe the implementation of PS-DMP.

Decision Rules

Various decision rules have been proposed from perspectives of game theory, discrete choice theory, transportation science, and constraint optimization. As updating rules are related to using data to learn, decision rules are related to using data to decide between routes.

The *best-move* decision rule, i.e., $R_B(\mathbf{C})$, always returns k^*, i.e., the action with the lowest cost in a given cost array $\mathbf{C}$. This rule might be used to quickly reach a local optimum in constraint optimization (*26*). This rule is similar to the best response in the fading memory JSFP (*14*).

The *random-move* decision rule (R_{RND}) returns an action $k \in [1, |K_{rs}|]$ at random. This rule can be used to escape from a local optimum (*26*), and to explore new, unused actions (*12*).

Some *inertia*-style decision rules stay at $\widehat{k}$ in specific situations, and returns *undecidable* otherwise. The ε*-inertia* decision rule (R^{ε}_{IN}) (*19*) returns $\widehat{k}$ if $(C(\widehat{k}) - C(k^*))/C(\widehat{k}) \leq \varepsilon$, in which $\varepsilon \geq 0$ is a relative threshold value related to the perception error. This rule has also been used to prevent unnecessary migrations caused by probabilistic effects. If $\varepsilon = 0$, R^{ε}_{IN} becomes a *best-inertia* decision rule (R^{B}_{IN}), which returns $\widehat{k}$ if $\widehat{k} \equiv k^*$ (*19*). The δ*-inertia* decision rule (R^{δ}_{IN}) (*19*) returns $\widehat{k}$ with probability $C(k^*)/C(\widehat{k})$. The *absolute-inertia* decision rule (R^{A}_{IN}) simply returns $\widehat{k}$.

The *proportional* decision rule, i.e., $R_P(\mathbf{F})$, has the selection probabilities on each route $k \in K_{rs}$, with respect to a frequency array $\mathbf{F}$ (or an *action probability vector* (*11*)) on K_{rs}:

$$p_k = \frac{F(k)}{\sum_{k' \in K_{rs}} F(k')}, \quad \text{for } \forall k \in K_{rs}. \tag{7}$$

The *multinomial logit* decision rule (*5*), i.e., $R_{MNL}(\mathbf{C})$, has the selection probabilities on each route $k \in K_{rs}$, with respect to a generalized cost array $\mathbf{C}$ on all the routes in K_{rs}:

$$p_k = \frac{\exp(-C(k))}{\sum_{k' \in K_{rs}} \exp(-C(k'))}, \text{ for } \forall k \in K_{rs}. \tag{8}$$

The *regret-matching* decision rule (*13*), i.e., $R_{RM}(\mathbf{C}, c)$, has two inputs, i.e., the average cost array $\mathbf{C}$ on all routes and the average individual cost c experienced by the agent, and a parameter $\nu \geq 1$. The R_{RM} rule has the probability p_k to switch to route k for $\forall k \neq \widehat{k}$,

$$p_k = \max(0, c - C(k))/(\nu \cdot c), \text{ for } \forall k \neq \widehat{k}, \tag{9}$$

where $c - C(k)$ can be interpreted as the "regret" for not choosing k in the past steps. It has been shown that regret matching is strongly related to fictitious play (FP) (*14*). Here $(\nu \cdot c)$ is used to set the parameter $\mu > 0$ used in the original algorithm (*13*). The μ value should be sufficiently large to ensure $\sum_{k \neq \widehat{k}} p_k < 1$ and guarantee the convergence to the set of correlated equilibria, but a too large value reduces the speed of convergence (*13*). Choosing a fixed value as in (*13*) might be quite difficult, since c for different agents in different O-D pairs varies significantly.

The *exploration-replication* decision rule (*12*), i.e., $R_{ER}(\mathbf{C}, \mathbf{F})$, is inspired by the replicator dynamics in evolutionary game theory, which amplifies the probabilities of using actions with lower costs. The rule uses two inputs, i.e., a cost array $\mathbf{C}$ and a frequency array $\mathbf{F}$ on the route set K_{rs}. The execution consists two steps. First, it samples a route $k' \in K_{rs}$ based on R_{RND} with the probability b, and $R_P(\mathbf{F})$ with the probability $(1-b)$, using the frequency array $\mathbf{F}$ as the input. Then it choose the route k_s with the following probability:

$$p_{k'} = \frac{\max(0, C(\widehat{k}) - C(k'))}{d \cdot (C(\widehat{k}) + a)}. \tag{10}$$

where $a \geq -C(k)$ for $\forall k \in K_{rs}$ is used to prevent a negative cost, d is the relative slope that denotes an upper bound on the elasticity of the latency functions l_e on all edges. For example, $d = 4$ for the function $l_e(x) \sim x^4$ used in (*18*). By default, $a = 0$, since $C(k) \geq 0$ for $\forall k$ in our setting.

Implementation

The implementation proceeds in two steps. The first step is to define a list of instances of decision rules, where each instance is *instantiated* with specific elements in the memory as the inputs, with specific setting parameter values, and with a unique name to be called later.

- (R1): $R_{IN}^{\varepsilon}(\widetilde{\mathbf{TT}})$ with $\varepsilon = 0.1$.
- (R2): $R_{IN}^{\delta}(\widetilde{\mathbf{TT}})$.
- (R3): R_{IN}^{A}.
- (R4): $R_B(\widetilde{\mathbf{TT}})$.
- (R5): $R_P(\widetilde{\mathbf{F}}_{LRI})$.
- (R6): $R_{MNL}(-\theta \cdot \widetilde{\mathbf{D}})$ with $\theta = 0.05$ (In Table 1, R_{RF} uses the same θ value).
- (R7): $R_{RM}(\widetilde{\mathbf{TT}}, \widetilde{tt})$ with $\nu = 10$.
- (R8): $R_{ER}(\widetilde{\mathbf{TT}}, \widetilde{\mathbf{FF}})$ with $a = 0$, $b = 0$, and $d = 4$.

The second step is to define individual PS-DMP cases, where each case can be viewed as a stand-alone route choice model (RCM), based on the instances of decision rules.

- (LRI): [(R5, 1)]. It is the linear reward-inaction (LRI) algorithm (*11*).

- (ERP): [(R3, 31/32), (R8, 1)]. It is the exploration-replication policy (ERP) (*12*).
- (RM): [(R7, 1)]. This is a variant of the regret-matching (RM) model (*13*).
- (ABM): [(R1, 1), (R2, 1), (R6, 1)]. This is a variant of the agent-based model in (*19*).
- (LRI2): [(R1, 1), (R2, 1), (R5, 1)]. LRI is hybridized with two sub-processes in ABM.
- (ERP2): [(R1, 1), (R2, 1), (R8, 1)]. ERP is hybridized with two sub-processes in ABM.
- (ABM-B): [(R1, 1), (R2, 1), (R4, 1/2), (R6, 1)]. Here ABM is hybridized with R4. As instantiated in the framework, R4 is similar to the best response in JSFP (*14*).
- (ABM-BI): [(R3, 3/4), (R1, 1), (R2, 1), (R4, 1/2), (R6, 1)]. Here ABM-B is hybridized with R3. R3 is corresponding to *inertia*, a probabilistic reluctance to change choices (*14*).

Let us take ABM-B as an example for describing the execution. Here sp_1=(R1, 1), which simply executes R1 with probability 1. If R1 returns $\widehat{k}$, the total process is terminated and returns $\widehat{k}$; otherwise R1 returns *undecidable*, and the process goes to sp_2. If R2 also returns *undecidable*, then R4 is only executed with probability 1/2, otherwise R6 is executed. Here R1 and R2 are executed sequentially, whereas R4 and R6 are executed in a probabilistic parallel fashion.

EXPERIMENTS

In this section, experiments are designed to evaluate the route choice models implemented in the unified framework, based on two traffic control systems, in a microscopic simulation environment.

Experimental Setup

We evaluate the performance of algorithms in simulation using an open-source microscopic road traffic simulator, Simulation of Urban Mobility (SUMO)[5].

Fig. 1 shows a real-world road network (*3*) of the downtown area of Pittsburgh, PA. We considered a time-of-day period corresponding to a mid-day period. This period has an average of 4786 vehicles per hour. The total simulation duration is four hours to reduce the side bias. Initial routes and flows were generated using SUMO, based on the turning movement counts at all intersections. A total of 3234 routes are generated.

Two traffic control systems (TCSs) were considered. For the "fixed" TCS, the coordinated signal timing plans used to control the Pittsburgh downtown network, including cycle times, splits, and offsets for all intersections, were provided to us by the city. These plans were generated originally using SYNCHRO, given the traffic signal constraints and turning movement counts in the time-of-day period. This TCS is used to evaluate the capability of RCS in the context of fixed traffic signal timings.

The "adaptive" system is a decentralized, schedule-driven TCS, which integrates traffic flow theory and artificial intelligence techniques. Each intersection is controlled by a local scheduler (*3*), which maintains a phase schedule that minimizes the total delay for vehicles traveling through the intersection and continually makes decisions to update the schedule according to a rolling horizon. The intersection scheduler communicates outflow information implied by its current schedule to its immediate neighbors, to extend visibility of incoming traffic and achieve network level coordination (*20*). This approach has been embedded into the scalable urban traffic control (SURTRAC) system running in the East Liberty area of Pittsburgh, PA since 2012, and has reduced the average travel time through the pilot site by over 25% (*4*). Further details can be found in (*3, 4, 20*). This system is well suitable for the combined RCS and TCS process, since it can gen-

[5]http://www.sumo-sim.org

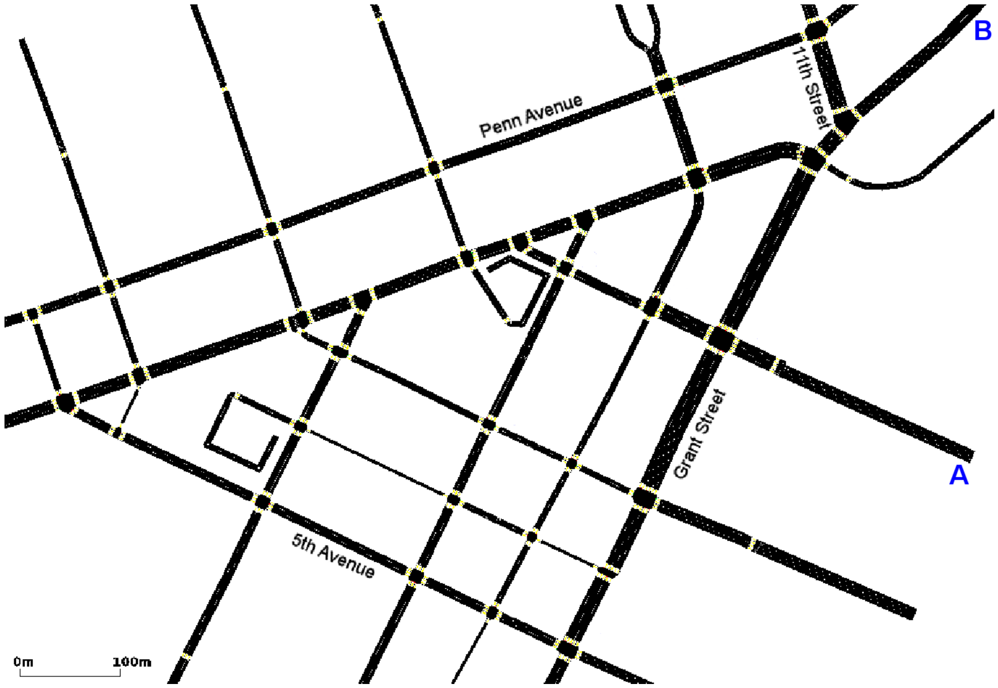


FIGURE 1 : Downtown Pittsburgh traffic network with 32 intersections: Among a total of 32 intersections, there are 22 intersections with two phases (of which 4 have a pedestrian phase), 8 intersections with three phases, 1 intersection with one phase plus a pedestrian phase, and 1 intersection with four phases. Most roads allow two-way traffic.

erate near-optimal signal control results quickly (normally milliseconds or less per intersection for each execution of the core algorithm), for each flow pattern generated by RCS. From the viewpoint of the iterative optimization and assignment procedure, this TCS provides an efficient optimization part for the dynamic assignment problem, in a realistic setting.

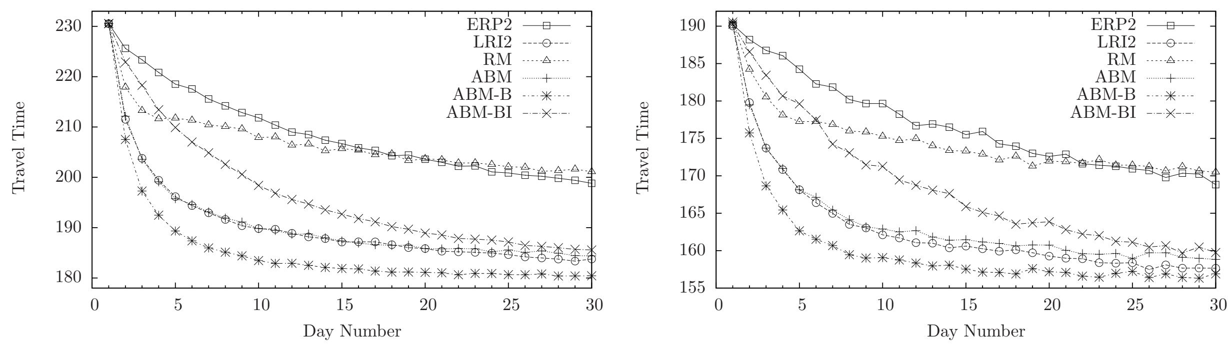
For all test cases, the day-by-day average travel times over 30 days are reported.

Results

The primary purpose is to evaluate the performance of different RCMs in a unified environment.

We first consider a basic usage of RCS, i.e., to quickly reach an equilibrium in the network. Figure 2 shows the results for six models, starting from basic flows that corresponding to an imperfect equilibrium obtained by myopic behavior of human drivers. ERP and LRI were not included here due to their much slower convergence. With ABM-B, the adaptive route choice behavior reduced average travel time by 21.7%, and the adaptive TCS produced a further reduction 13.4%, on the average travel time. Four of the six models, except for ERP2 and RM, approached to an approximate equilibrium quite fast, although ERP2 and RM also showed the convergence behavior to some extent.

It is also expected that an RCS might handle changing traffic flow between O-D pairs. We considered the four models that were effective in the basic test. Here the initial condition is the previous equilibrium solution obtained by ABM-B. For Figures 3 and 4, the flows from A to B (in Figure 1) were increased by 100% and 200%, which corresponds to 315 and 730 more vehicles per hour. These vehicles are distributed into the routes between A and B at random.



(a) The fixed TCS (b) The adaptive TCS

FIGURE 2 : Basic flow: Starting from the flows based on real movement counts.

There are larger differences under the fixed TCS, which appeared to be more congested. LRI2 and ABM converged relatively slow, especially for the higher disturbance case in Figure 4a. ABM-BI could quickly approach to an approximated equilibrium.

Under the adaptive TCS, the four RC models achieved much lower travel times, although they have noisy evolution curves. ABM-B achieved the best performance in the two disturbed flow cases, whereas ABM-BI converged slower as compared to ABM-B.

The adaptive TCS is now the main force to resolve the congestion in the network. The traffic system is in congested and non-congested states in Figure 3, and over-congested and congested states in Figure 4, for the fixed and adaptive TCSs, respectively. For fixed-time TCSs, the loss of effectiveness as dynamic flow changes might be seen as an *aging* problem in the real world. The advance of the adaptive TCS might due to that the real-time adaptation leads to flexible capacity control for reducing the risk of congestion, as working at near equilibrium generated by RCS.

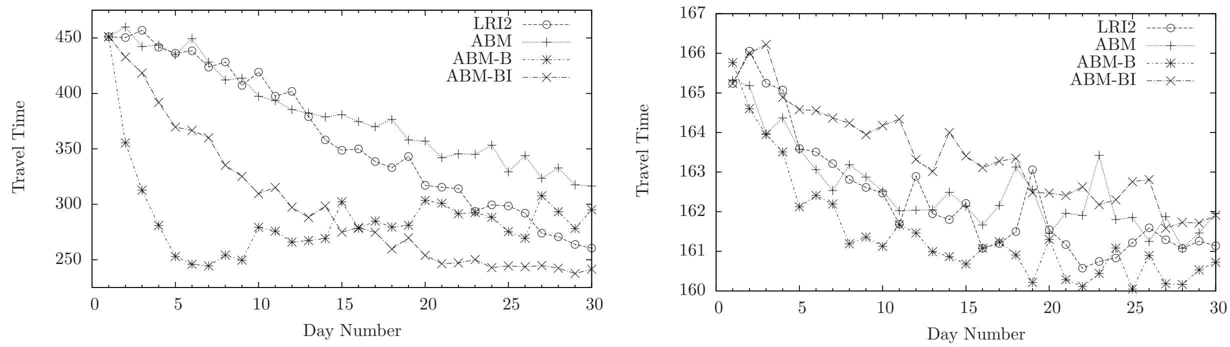
Furthermore, some knowledge can be gained by examining the difference of RCMs from their components, based on the unified framework. Compared to ERP and LRI, all six models strongly consider the cost information before agents change routes. ONly LRI has a weak usage of costs, and the proportional decision rule might return poorer routes. ERP sets a strong limitation, i.e., only 1/32 agents are activated each day (*12*), but that inertia rule is not cost-based.

We can also find some difference in main route-changing rules, which are R7 and R8 for RM and ERP2, and R5 and R6 for LRI2 and ABM variants. Both R7 and R8 primarily use the cost array $\widetilde{\mathbf{TT}}$ to directly make decisions, whereas R5 and R6 works by updating choice probabilities directly and indirectly. Actually, the correlated equilibrium (*13*) is a collective result of individual choice probabilities. For the class of R5 and R6, it might be easier to learning more precisely.

R5 and R6 use the frequency array $\widetilde{\mathbf{FF}}$ as a "warm start", which might deviate too far from the equilibrium state for heavily disturbed cases. This can be seen from the slower convergence rates in Figures 3a and 4a, especially as the latter case has a higher disturbance.

It is also interesting to examine the usage of R4, which behaves like the *best response* in the fading memory JSFP with inertia (*14*). R4 is used in ABM-B and ABM-BI. ABM-B achieved the best performance in four test cases. In Figure 3a and 4a, ABM-B reached a high-quality solution within a few iterations, but does not retain a stable solution as time moves on.

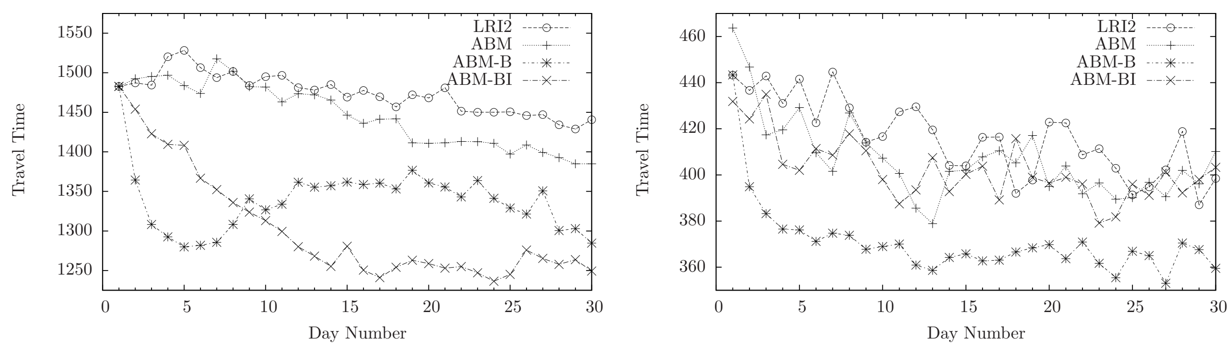
A simple way to handle the instability problem is to introduce more *inertia* (*13, 14*). As shown in Figure 3a and 4a, ABM-BI could approach to an approximated equilibrium. However,



(a) The fixed TCS

(b) The adaptive TCS

FIGURE 3 : Disturbed flow: The O-D flow from A to B is increased by 100%.



(a) The fixed TCS

(b) The adaptive TCS

FIGURE 4 : Disturbed flow: The O-D flow from A to B is increased by 200%.

there is no free lunch. ABM-BI converged slower than ABM-B in the other four less congested cases.

We have shown some effective hybrid models in the framework. It is naturally to ask if there are ways to build more effective and stable RCMs in the huge combinatorial model space supported by the extensible framework, as new memory elements and decision rules can be easily added, and various combinations of PS-DMP cases can be implemented. One possible way is online meta-learning (*25*) over multiple PS-DMP cases. For example, in Figure 3a and 4a, ABM-B might be used in the first several iterations, and ABM-BI is applied afterward. An intelligent, automatic selection procedure is required, by inferring from available information.

CONCLUSIONS

In this paper, an agent-based route choice framework was proposed. The framework contains a set of autonomous route choice agents. Each agent manages its own memory by a memory updating process, based on limited individual and social experience, and then selects a route by a probabilistic sequential decision making process over some decision rules that are instantiated with the memory elements. Based on this framework, various route choice models can be realized in a combinatorial model space, supported using a few memory elements and decision rules.

Some state-of-the-art route choice models and their hybrids were implemented in the unified framework. These models were then empirically evaluated in a microscopic simulation environment using two real-world TCSs, i.e., a fixed, SYNCHRO-optimized coordinated signal control system and an adaptive, scalable urban traffic control (SURTRAC) system. We have shown some models converge quickly to an approximate equilibrium in a real-world network, while other are slow to converge in the the time frame considered. In the presence of disturbed flows, all models achieved better performance in the adaptive TCS, demonstrating that both RCS and TCS can contribute to reduced travel time. Overall, variations of the ABM route choice model hybridized with some best response and inertia behavior achieved the best performance. Based on the unified framework, properties of some choice rules in these route choice models could be obtained, according to the difference in their evolution curves. This knowledge might be useful to build more effective and stable models in the extensible framework.

There are several aspects of the current system that warrant further study. First, it might be extended to a Stackelberg game formulation (*22, 27*), where TCS is the leader and the users in RCS are followers, with some forms of pricing strategies to reduce the price of anarchy among multiple noncooperative commodities. Second, it is natural to extend our framework for investigating the system behavior under the influence of mixed cooperation and competition among multiple traffic ISPs (*15, 28*). Third, the agent-based framework is a natural platform to describe the heterogeneity of user behavior, e.g., (constrained) departure time choice (*29*) and risk preference (*30*), and to characterize route choice behavior and understand its dynamics in highly time-dependent flows.

ACKNOWLEDGEMENTS

This research was supported in part by the Traffic21 Initiative at Carnegie Mellon University, with support from the Hillman Foundation, the Heinz Endowments, and the CMU Robotics Institute.

REFERENCES

[1] Papageorgiou, M., C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, Vol. 91, 2003, pp. 2043–2067.

[2] Mirchandani, P. and L. Head. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C*, Vol. 9, 2001, pp. 415–432.

[3] Xie, X.-F., S. F. Smith, L. Lu, and G. J. Barlow. Schedule-driven intersection control. *Transportation Research Part C*, Vol. 24, 2012, pp. 168–189.

[4] Smith, S. F., G. J. Barlow, X.-F. Xie, and Z. B. Rubinstein. SURTRAC: Scalable urban traffic control. In *Transportation Research Board Annual Meeting*, 2013.

[5] McFadden, D. and K. Train. Mixed MNL models for discrete response. *Journal of Applied Econometrics*, Vol. 15, 2000, pp. 447–470.

[6] Daganzo, C. F. and Y. Sheffi. On stochastic models of traffic assignment. *Transportation Science*, Vol. 11, 1977, pp. 253–274.

[7] Prashker, J. N. and S. Bekhor. Route choice models used in the stochastic user equilibrium problem: a review. *Transport reviews*, Vol. 24, No. 4, 2004, pp. 437–463.

[8] Qian, Z. and H. Zhang. A hybrid route choice model for dynamic traffic assignment. *Networks and Spatial Economics*, Vol. 13, 2013, pp. 183–203.

[9] Smith, M. and R. Mounce. A splitting rate model of traffic re-routeing and traffic control. *Transportation Research Part B*, Vol. 45, 2011, pp. 1389–1409.

[10] Meneguzzer, C. Review of models combining traffic assignment and signal control. *Journal of Transportation Engineering*, Vol. 123, 1997, pp. 148–155.

[11] Barth, D., J. Cohen, O. Bournez, and O. Boussaton. Distributed learning of equilibria in a routing game. *Parallel Processing Letters*, Vol. 19, 2009, pp. 14–42.

[12] Fischer, S., H. Racke, and B. Vocking. Fast convergence to Wardrop equilibria by adaptive sampling methods. *SIAM Journal on Computing*, Vol. 39, 2010, pp. 3700–3735.

[13] Hart, S. and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, Vol. 68, 2000, pp. 1127–1150.

[14] Marden, J., G. Arslan, and J. Shamma. Joint strategy fictitious play with inertia for potential games. *IEEE Transactions on Automatic Control*, Vol. 54, 2009, pp. 208–220.

[15] Oh, J. and S. F. Smith. A few good agents: Multi-agent social learning. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 339–346.

[16] McKelvey, R. and T. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, Vol. 10, 1995, pp. 6–38.

[17] Mahmassani, H. and G.-L. Chang. On boundedly rational user equilibrium in transportation systems. *Transportation Science*, Vol. 21, 1987, pp. 89–99.

[18] Chen, A., S. Pravinvongvuth, X. Xu, S. Ryu, and P. Chootinan. Examining the scaling effect and overlapping problem in logit-based stochastic user equilibrium models. *Transportation Research Part A*, Vol. 46, 2012, pp. 1343–1358.

[19] Feng, Y. and L. Head. A hierarchical agent-based simulation for modeling traveler behaviors. In *Transportation Research Board Annual Meeting*, 2013.

[20] Xie, X.-F., S. F. Smith, and G. J. Barlow. Schedule-driven coordination for real-time traffic network control. In *International Conference on Automated Planning and Scheduling*, 2012, pp. 323–331.

[21] Frejinger, E., M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B*, Vol. 43, 2009, pp. 984–994.

[22] Chen, O. and M. Ben-Akiva. Game-theoretic formulations of interaction between dynamic traffic control and dynamic traffic assignment. *Transportation Research Record*, Vol. 1617, 1998, pp. 179–188.

[23] Anderson, J. R. Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, Vol. 29, No. 3, 2005, pp. 313–341.

[24] Brandstatter, E., G. Gigerenzer, and R. Hertwig. The priority heuristic: Making choices without trade-offs. *Psychological Review*, Vol. 113, No. 2, 2006, pp. 409–432.

[25] Smith-Miles, K. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, Vol. 41, 2008, p. 6.

[26] Liu, J., J. Han, and Y. Tang. Multi-agent oriented constraint satisfaction. *Artificial Intelligence*, Vol. 136, 2002, pp. 101–144.

[27] Swamy, C. The effectiveness of stackelberg strategies and tolls for network congestion games. *ACM Transactions on Algorithms*, Vol. 8, 2012, p. 36.

[28] Yang, I. and R. Jayakrishnan. Modeling framework to analyze effect of multiple traffic information service providers on traffic network performance. In *Transportation Research Board Annual Meeting*, 2013.

[29] Zhang, L., M. Mollanejad, C. Xiong, and S. Zhu. Agent-based approach for integrating departure time and dynamic traffic assignment models. In *Transportation Research Board Annual Meeting*, 2013.

[30] Razo, M. and S. Gao. A rank-dependent expected utility model for strategic route choice. *Transportation Research Part C*, Vol. 27, 2013, pp. 117–130.