Unified Route Choice Framework

Specification and Application to Urban Traffic Control

Xiao-Feng Xie, Yiheng Feng, Stephen F. Smith, and K. Larry Head

The route choice system and the traffic control system (TCS) constitute two major approaches to mitigating congestion in urban road networks. The interaction between signal control and route choice is considered from a narrower route choice system perspective, with the focus on route choice models for operational purposes. The goal is to analyze the relative performance of alternative route choice models as different assumptions are made about the type of TCS in use. To this end, an agent-based framework for formulating different route choice models is defined, and this framework is integrated with a microscopic traffic simulation environment. Within the 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. Several previously developed route choice models from the literature are implemented with 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 systems: a 32-intersection road network in Pittsburgh, Pennsylvania, running with a SYNCHRO-generated coordinated timing plan and the same road network running with the scalable urban traffic control (SURTRAC) adaptive TCS. The results show that specific route choice models perform differentially when applied in conventional and adaptive traffic control settings and that better overall network performance for all route choice models is achieved in the adaptive control setting. This unified framework also makes it possible to analyze the performance impact of route choice model components and to formulate better-performing hybrid models.

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

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

The interaction between TCSs and RCSs is considered from the perspective of agent-based route choice models (RCMs), which build on concepts from game theory. The problem is formulated as a congestion game (*11–15*), a form of commuting problem in which 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 who 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 and system optimal represent performance optima 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. Because of their perception errors, agents should consider an imperfect, or noisy, rational expectations equilibrium (*16*). As also indicated by early work in the stochastic user equilibrium (*6, 7*) and the boundedly rational user equilibrium (*17*), agents still have a tendency toward cost minimization but do not necessarily choose the lowest. Instead, agents try to quickly adapt to the set of correlated equilibria (*13*), in which the probability distribution over the choice set is obtained by 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 RCMs have been proposed. In the regret matching strategy, agents may depart from the current choice with probabilities proportional to their regret for not choosing other particular choices in the past steps (*13*). This simple adaptive strategy can converge to a correlated equilibrium. The exploration–replication policy (ERP) is a distributed routing policy that approximates the Wardrop equilibrium (*12*). 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, the fading memory joint strategy fictitious play (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 models have been extensively studied, and some variants can handle the underlying independent and identically distributed assumption to some extent (*18*). In work by Feng and Head,

X.-F. Xie and S. F. Smith, Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213. Y. Feng and K. L. Head, Department of Systems and Industrial Engineering, University of Arizona, 1127 East James E. Rogers Way, Tucson, AZ 85721. Corresponding author: X.-F. Xie, xfxie@cs.cmu.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2466, Transportation Research Board of the National Academies, Washington, D.C., 2014, pp. 105–113.
DOI: 10.3141/2466-12

an agent-based model is proposed to account for the heterogeneity of different drivers by using the Dirichlet distribution, a conjugate prior of the multinomial distribution, where the concentration parameters are obtained over time with Bayesian learning (*19*). This model has been shown to converge to a user equilibrium and is able to handle flow disruptions in a single-commodity network.

The theoretical convergence properties of all of the foregoing models have been studied and demonstrated in idealized settings, and little attempt has been made to understand their performance in real-world traffic networks. The goal in this study is to investigate this general question, with particular emphasis on how various RCMs interact with different traffic control systems or plans. To this end, a unified agent-based framework for formulating different (distributed) RCMs is defined, and this framework is integrated with a microscopic traffic simulation environment. Within the 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 RCMs are implemented with the framework, along with some additional hybrid models, and these models are then empirically evaluated in two real-world settings: (*a*) a 32-intersection road network in Pittsburgh, Pennsylvania, running with a fixed, SYNCHRO-generated coordinated signal control plan and (*b*) the same road network running with the scalable urban traffic control (SURTRAC) adaptive TCS (*3, 4, 20*). As will be shown, the unified framework makes it possible to separately analyze the performance impact of RCMs and their components and provides a basis for formulating better-performing hybrid RCMs.

PROBLEM FORMULATION

For a traffic network, overall performance is determined by the interaction of two basic systems. The TCS allocates green times for traffic 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, and each phase has a duration that can range between a minimum and maximum. Various TCSs have been proposed to minimize travel time for drivers (*1–3*).

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 pregenerated by various strategies (*7, 8, 21*). Let A be a set of drivers (vehicles), each of whom travels between a given O-D pair rs departing at a given time t. The RCS is used by 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, each edge $e \in E$ is associated with a nondecreasing latency function $l_e(x) \geq 0$, where $x \geq 0$ is the flow on edge e for a specific time period (*11, 14, 16*). Analytic models—for example, the Bureau of Public Roads function (*18*)—provide a suitable means to calculate l_e for planning applications. However, in the presence of the 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.

A day-to-day evolution process is considered. On day n, each driver can decide to stay on the same route as the previous trip or change to an alternative route before his departure. Let $\hat{\mathrm{tt}}_{an}$ be the travel time of the driver a (on day n); the objective is to minimize the average travel time of all drivers ($\widehat{\mathrm{TT}}_n$):

$$\widehat{\mathrm{TT}}_n = \sum_{a \in A} \frac{\hat{\mathrm{tt}}_{an}}{|A|} \tag{1}$$

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

ROUTE CHOICE FRAMEWORK

For the RCS, it is natural to consider an agent-based framework, in which route choices are made by agents. Agent-based modeling promotes situatedness, robustness, and scalability. This framework is natural in the era of mobile computing, when smartphones and in-vehicle navigation systems are being broadly adopted. It is assumed that 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 communication.

Basic Framework

Specifically, the route choice framework contains an ISP 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 and 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 are basic components that can be added or 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 because of different departure times but also because of inherent disturbance in the complex network with the TCS. Second, the stochastic nature of 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 *rs*. Each agent $a \in A_{rs}$ has its own travel experience $(\hat{\mathrm{tt}}_{an}^{rs}, \hat{k}_{an}^{rs})$, that is, the individual travel time and the chosen route during day *n*. The ISP will provide it with the social information $(\widehat{\mathbf{TT}}_n^{rs}, \widehat{\mathbf{FF}}_n^{rs})$, that is, the travel times and the proportions for all routes between *rs*, for the agents in A_{rs}.

The following description focuses on the components related to the decision making of each agent $a \in A_{rs}$ during day $(n + 1)$. For simplicity, 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 necessary.

Memory and Updating Process

Memory is a basic component that supports the learning process of an RCM. Specifically, memory is used for storing and retrieving a fixed list of elements, in which each element summarizes a 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 RCMs are considered. Let *y* and $\boldsymbol{Y}$ 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 value *c*. Three subtypes of $\boldsymbol{Y}$ are considered: $\boldsymbol{C}$ is a generalized cost array, where $C(k) > 0$ indicates the cost for route *k*. $\boldsymbol{F}$ is a frequency array, where $F(k) > 0$ and $\sum_{k \in K_{rs}} F(k) = 1$. Also considered is an array of concentration parameters $\boldsymbol{D}$ used in a Dirichlet distribution (*19*), where $D(k) > 0$.

The updating rules that work on these generic types are first introduced and then used 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 a current EMA value $\tilde{y}$ using *y*, with the coefficient $\alpha \in [0, 1]$:

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

The EMA function can be easily realized in the array fashion, that is, by applying Equation 2 on individual elements of the EMA data array $\tilde{\boldsymbol{Y}}$ and the data array $\boldsymbol{Y}$:

$$\tilde{\boldsymbol{Y}} = R_{\mathrm{EMA}}^{\mathrm{vec}}\left(\tilde{\boldsymbol{Y}}, \boldsymbol{Y}|\alpha\right) \tag{3}$$

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

$$\tilde{Y}(k') = R_{\mathrm{EMA}}\left(\tilde{Y}(k'), y|\alpha\right) \tag{4}$$

The normalization functions $R_{\mathrm{NORM}}^{\mathrm{sum}}$ and $R_{\mathrm{NORM}}^{\mathrm{min}}$ obtain scaled arrays from an array $\boldsymbol{Y}$ 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 do nothing if the denominator is zero.

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

It is assumed that $\boldsymbol{F} = R_{\mathrm{MNL}}(-\theta \cdot \boldsymbol{D})$ (the multinomial logit rule R_{MNL} will be introduced later in Equation 8), where $\theta > 0$ is the dispersion parameter; the reverse-fitting rule, $R_{RF}(\boldsymbol{F})$, returns an array of concentration parameters $\boldsymbol{D}$ by reversely fitting from a frequency array $\boldsymbol{F}$:

$$D(k) = \frac{\log\left(\exp(\theta) \cdot \dfrac{F(k)}{F_{\min}}\right)}{\theta} \qquad \text{for } \forall k \in K_{rs} \tag{5}$$

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

In the linear reward–inaction updating rule (*11*), $R_{\mathrm{LRI}}(\boldsymbol{F}, \boldsymbol{C}, \hat{k})$, the frequency array $\boldsymbol{F}$ is updated by using a normalized cost array $\boldsymbol{C}_{\mathrm{norm}} = R_{\mathrm{NORM}}^{\mathrm{sum}}(\boldsymbol{C})$ and $\boldsymbol{Y} = R_{\mathrm{CI}}(\hat{k})$:

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

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

Memory Updating Process

The memory updating process initializes and updates the memory by using individual experience $(\hat{\mathrm{tt}}, \hat{k})$ 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. Here $\gamma = 1$. One might set $\gamma < 1$ to accommodate the fact that the access to social information may be limited. The types of $\hat{\mathrm{tt}}$, $\widetilde{\mathbf{TT}}$, and $\widetilde{\mathbf{FF}}$ are *c*, $\boldsymbol{C}$, and $\boldsymbol{F}$, respectively.

Table 1 shows the implementation for the memory updating process. Here the memory contains five elements, $\{\tilde{\mathrm{tt}}, \widetilde{\mathbf{TT}}, \widetilde{\mathbf{FF}}, \tilde{\boldsymbol{F}}_{\mathrm{LRI}}, \tilde{\boldsymbol{D}}\}$, with the types shown in the fifth column. Each element will be

TABLE 1 Implementation for Memory Updating Process

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

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 the upper rows, if available. One might add elements with other properties by using different updating rules.

Here $\tilde{\mathbf{tt}}$, $\widetilde{\mathbf{TT}}$, and $\widetilde{\mathbf{FF}}$ are information sources, which are expected to have small perception errors over time. The updating of $\tilde{\mathbf{tt}}$ 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, $\tilde{\boldsymbol{F}}_{\mathrm{LRI}}$ and $\tilde{\boldsymbol{D}}$ are updated directly and indirectly through multinomial logit to represent route choice probabilities. During the initialization, $\widetilde{\mathbf{FF}}$ provides a "warm start" for $\tilde{\boldsymbol{F}}_{\mathrm{LRI}}$ directly and for $\tilde{\boldsymbol{D}}$ using R_{RF}. Both are updated over time, which represents a basic learning mechanism toward a form of (correlated) equilibrium with other agents.

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 $k \in K_{rs}$ by using the elements in the memory. PS-DMP contains an ordered list of subprocesses, that is, [$\mathrm{sp}_1, \ldots, \mathrm{sp}_m, \ldots, \mathrm{sp}_M$], in which each subprocess (sp) is described as a pair (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 subprocesses from sp_1 to sp_M. For each sp, its decision rule instance is executed with the associated probability. Suppose that the current subprocess 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, the decision rule of sp_m either was not executed (with the associated probability) or was returned undecidable, in which case the process continues to execute the next subprocess, sp_{m+1}. PS-DMP simply returns $\hat{k}$ 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 on the basis of 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 subspace in the decision space. Probabilistic parallel execution is supported for rules with associated probability values, and these rules can cooperate (through use of the memory) across iterations (*25*).

Decision Rules

Various decision rules have been proposed from perspectives of game theory, discrete choice theory, transportation science, and constraint optimization. Just as updating rules relate to use of data to learn, decision rules relate to use of data to decide between routes.

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

The random-move decision rule (R_{RND}) returns $k \in K_{rs}$ 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 $\hat{k}$ in specific situations and return undecidable otherwise. The ε-inertia decision rule ($R^{\varepsilon}_{\mathrm{IN}}$) (*19*) returns $\hat{k}$ if $(C(\hat{k}) - C(k^*))/C(\hat{k}) \le \varepsilon$, in which $\varepsilon \ge 0$ 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^{n}_{IN} becomes a best-inertia decision rule (R^{B}_{IN}), which returns $\hat{k}$ if $\hat{k} \equiv k^*$ (*19*). The δ-inertia decision rule (R^{δ}_{IN}) (*19*) returns $\hat{k}$ with probability $C(k^*)/C(\hat{k})$. The absolute-inertia decision rule (R^{A}_{IN}) simply returns $\hat{k}$.

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

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

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

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

The regret-matching decision rule (*13*), $R_{\mathrm{RM}}(\boldsymbol{C}, c)$, has two inputs—the average cost array $\boldsymbol{C}$ on all routes and the average individual cost c experienced by the agent—and a parameter $v \ge 1$. The R_{RM} rule has the probability p_k to switch to route k for $\forall k \ne \hat{k}$:

$$p_k = \frac{\max(0, c - C(k))}{(v \cdot c)} \qquad \text{for } \forall k \ne \hat{k} \tag{9}$$

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 (*14*). Here $(v \cdot c)$ is used to set the parameter $\mu > 0$ used in the original algorithm (*13*). The μ-value should be sufficiently large to ensure $\sum_{k \ne \hat{k}} p_k < 1$ and guarantee the convergence to the set of correlated equilibria, but a too-large value reduces the speed of convergence (*13*). Choice of a fixed value as done by Hart and Mas-Colell (*13*) might be quite difficult, since c for different agents in different O-D pairs varies significantly.

The exploration–replication decision rule (*12*), $R_{\mathrm{ER}}(\boldsymbol{C}, \boldsymbol{F})$, 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, a cost array $\boldsymbol{C}$ and a frequency array $\boldsymbol{F}$, on the route set K_{rs}. The execution has two steps. First it samples a route $k \in K_{rs}$ based on R_{RND} with the probability b, and $R_P(\boldsymbol{F})$ with the probability $(1 - b)$, using the frequency array $\boldsymbol{F}$ as the input. Then it chooses the route k with the following probability:

$$p_k = \frac{\max\left(0, C\left(\hat{k}\right) - C(k)\right)}{d \cdot \left(C\left(\hat{k}\right) + a\right)} \tag{10}$$

where $a \ge -C(k)$ for $\forall k \in K_{rs}$ is used to prevent a negative cost and 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 $l_e(x) \propto x^4$ is used by Chen et al. (*18*). By default, $a = 0$, since $C(k) > 0$ for $\forall k$ in the current setting.

Implementation

The implementation proceeds in two steps. The first step is to define a list of instances of decision rules, in which 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 given later:

R1. $R^{\varepsilon}_{\mathrm{IN}}(\widetilde{\mathbf{TT}})$ with $\varepsilon = 0.1$,
R2. $R^{\delta}_{\mathrm{IN}}(\widetilde{\mathbf{TT}})$,
R3. R^{A}_{IN},
R4. $R_B(\widetilde{\mathbf{TT}})$,
R5. $R_P(\tilde{\boldsymbol{F}}_{\mathrm{LRI}})$,
R6. $R_{\mathrm{MNL}}(-\theta \cdot \tilde{\boldsymbol{D}})$ with $\theta = 0.05$ (in Table 1, R_{RF} uses the same θ-value),
R7. $R_{\mathrm{RM}}(\widetilde{\mathbf{TT}}, \tilde{\mathrm{tt}})$ with $v = 1$, and
R8. $R_{\mathrm{ER}}(\widetilde{\mathbf{TT}}, \widetilde{\mathbf{FF}})$ 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 RCM, on the basis of the instances of decision rules:

- LRI. [(R5, 1)] is the linear reward–inaction algorithm (*11*);
- ERP. [(R3, 31/32), (R8, 1)] is the exploration–replication policy (*12*);
- RM. [(R7, 1)] is a variant of the regret-matching model (*13*);
- ABM. [(R1, 1), (R2, 1), (R6, 1)] is a variant of the agent-based model in work by Feng and Head (*19*);
- LRI2. [(R1, 1), (R2, 1), (R5, 1)] is hybridized with subprocesses in the ABM;
- ERP2. [(R1, 1), (R2, 1), (R8, 1)] is hybridized with subprocesses in the ABM;
- ABM-B. [(R1, 1), (R2, 1), (R4, 1/2), (R6, 1)] is hybridized with R4; as instantiated in the framework, R4 is similar to the best response in JSFP (*14*); and
- ABM-BI. [(R3, 3/4), (R1, 1), (R2, 1), (R4, 1/2), (R6, 1)]; ABM-B is hybridized with R3, which corresponds to inertia, a probabilistic reluctance to change choices (*14*).

ABM-B is taken as an example for describing the execution. Here sp_1= (R1, 1), which simply executes R1 with probability 1. If R1 returns $\hat{k}$, the total process is terminated and returns $\hat{k}$; 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 RCMs implemented in the unified framework on the basis of two real-world TCSs in a microscopic traffic simulation environment.

Experimental Setup

The performance of the algorithms in the simulation is evaluated by using an open-source microscopic road traffic simulator, the simulation of urban mobility (*27*).

Figure 1 shows a real-world road network in the downtown area of Pittsburgh (*3*). Among a total of 32 intersections, there are 22 intersections with two phases (of which 4 have a pedestrian phase),

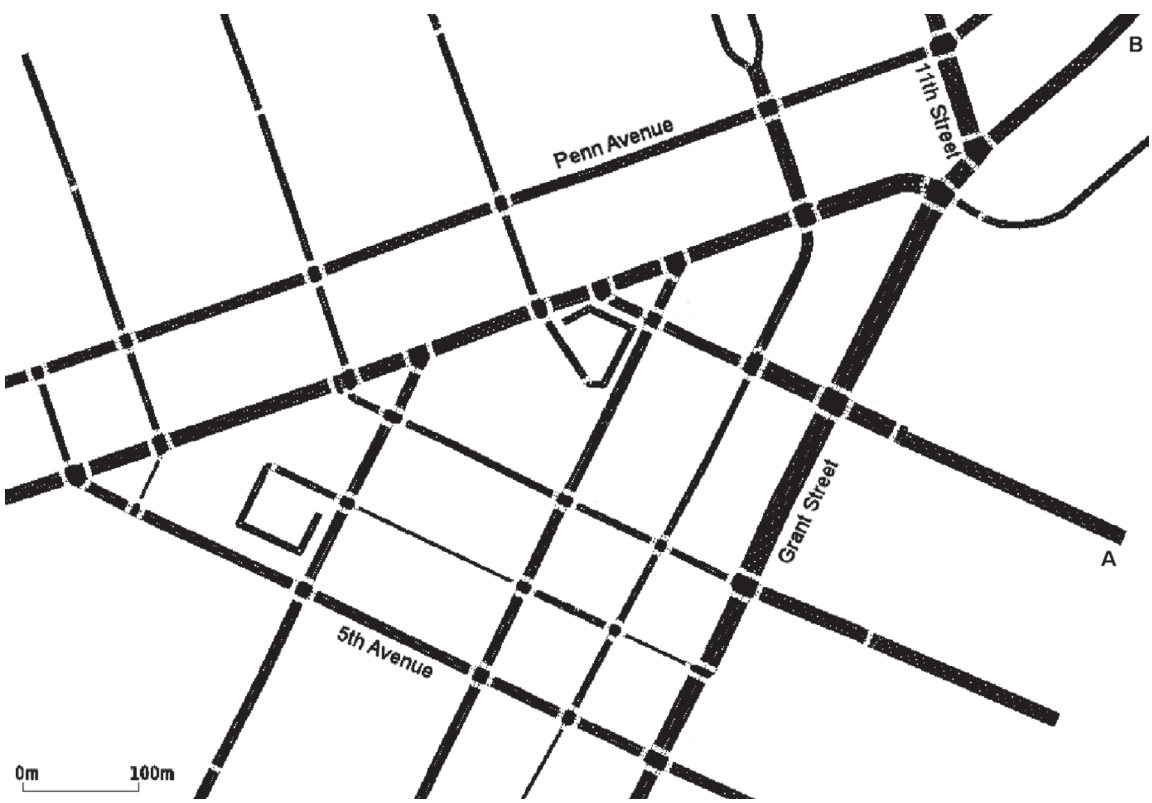

FIGURE 1 Downtown Pittsburgh traffic network with 32 intersections (*3*).

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. A time-of-day period corresponding to a midday period was considered. This period has an average of 4,786 vehicles per hour. The total simulation duration is 4 h to reduce the side bias. Initial routes and flows were generated with the simulation of urban mobility on the basis of the turning movement counts at all intersections. A total of 3,234 routes were generated.

Two TCSs were considered. For the fixed TCS, the coordinated signal timing plans currently used to control the Pittsburgh downtown network, including cycle times, splits, and offsets for all intersections, were provided by the city public works department. These plans were generated originally with 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 the 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, 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 (*3*). 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 SURTRAC system running in the East Liberty area of Pittsburgh since 2012 and has reduced the average travel time through the pilot site by over 25% (*4*). Further details can be found elsewhere (*3, 4, 20*). This system is well suited for the combined RCS and TCS process since it can generate 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 the 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 here is to evaluate the performance of different RCMs in a unified environment. First a basic use of the RCS, to quickly reach an equilibrium in the network, is considered. Figure 2 shows the results for six models, starting from basic flows that correspond to an imperfect equilibrium obtained by myopic behavior of human drivers. The ERP and LRI were not included here because of their much slower convergence. With ABM-B, the adaptive

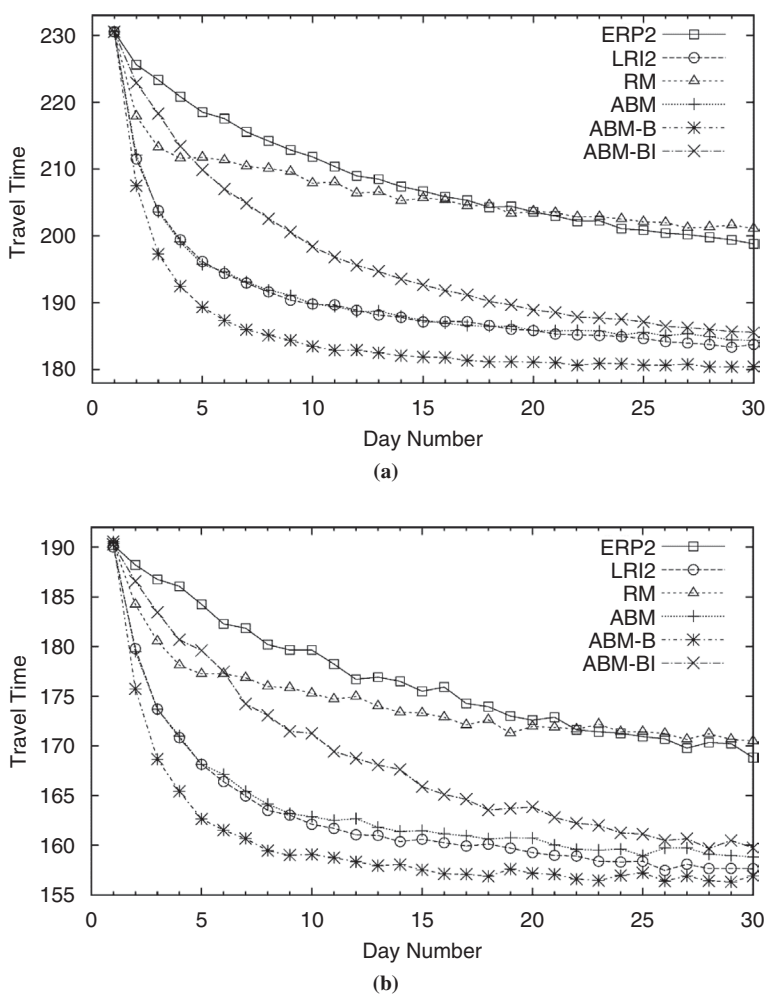

FIGURE 2 Basic flow: starting from flows based on real movement counts: (*a*) fixed-time TCS and (*b*) adaptive TCS.

route choice behavior reduced average travel time by 21.7%, and the adaptive TCS produced a further reduction of 13.4% on the average travel time. Four of the six models, except for ERP2 and RM, approached 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. The four models that were most effective in the basic test were considered. Here the initial condition is the previous equilibrium solution obtained by ABM-B. For Figures 3 and 4, the flows from A to B (Figure 1) were increased by 100% and 200%, which corresponds to 315 and 730 more vehicles per hour. These vehicles are distributed on the routes between A and B at random.

In this experiment, larger differences were observed under the fixed TCS condition, which appeared to be more congested. LRI2 and ABM converged relatively slowly, especially for the higher-disturbance case in Figure 4*a*. ABM-BI could quickly approach 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 than ABM-B.

The adaptive TCS is now the main force for resolving the congestion in the network. The traffic system is in congested and noncongested states in Figure 3 and in overcongested 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 be due to the fact that the real-time adaptation leads to flexible capacity control for reducing the risk of congestion when operating at near equilibrium generated by the RCS.

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

Some difference in the main route-changing rules can also be found, 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 work by updating choice probabilities directly and indirectly. Actually, the correlated equilibrium is a collective result of individual choice probabilities (*13*). For the class of R5 and R6, it might be easier to learn 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 operation can be seen from the slower convergence rates

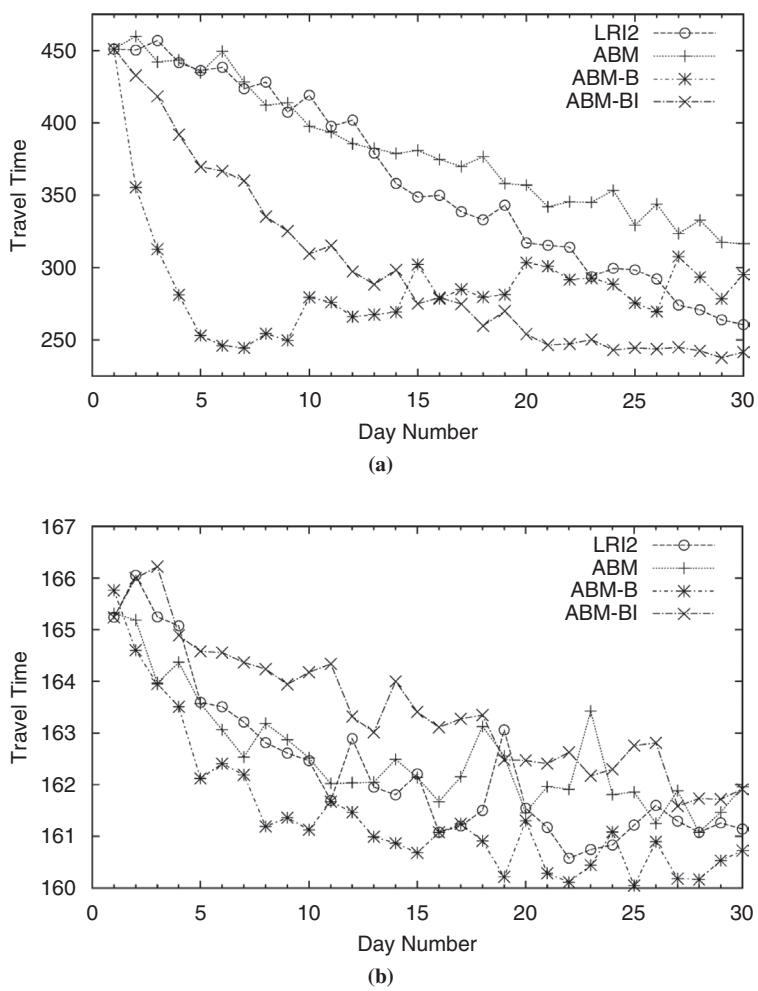

FIGURE 3 Disturbed flow: O-D flow from A to B increased by 100%: (*a*) fixed-time TCS and (*b*) adaptive TCS.

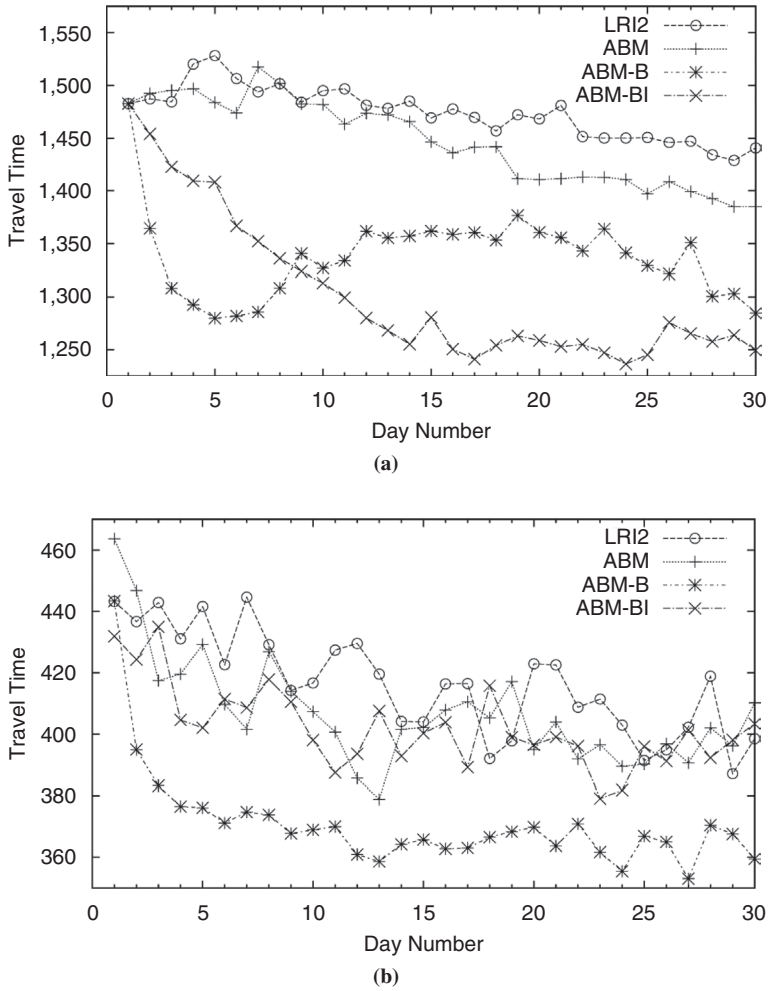

FIGURE 4 Disturbed flow: O-D flow from A to B increased by 200%: (*a*) fixed-time TCS and (*b*) adaptive TCS.

in Figures 3*a* and 4*a,* especially since the latter case has a higher disturbance.

It is also interesting to examine the use 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 Figures 3*a* and 4*a,* ABM-B reached a high-quality solution within a few iterations but did not retain a stable solution as time progressed.

A simple way to handle the instability problem is to introduce more inertia (*13, 14*). As shown in Figures 3*a* and 4*a,* ABM-BI could be approaching an approximated equilibrium. However, there is no free lunch. ABM-BI converged slower than ABM-B in the other four less congested cases.

Some effective hybrid models have been shown in the framework. It is natural to ask if there are ways to build more effective and stable RCMs in the huge combinatorial model space supported by the extensible framework, since 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 Figures 3*a* and 4*a,* 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

An agent-based route choice framework is proposed. The framework contains a set of 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 PS-DMP over some decision rules that are instantiated with the memory elements. On the basis of this framework, various RCMs can be configured in a combinatorial model space, supported by using a few memory elements and decision rules.

Some state-of-the-art RCMs and their hybrids were implemented in the unified framework. These models were then empirically evaluated in a microscopic simulation environment by using two real-world TCSs: a fixed, SYNCHRO-optimized coordinated signal control system and an adaptive, SURTRAC system. Some models converged quickly to an approximate equilibrium in a real-world network, whereas other were slow to converge in the time frame considered. In the presence of disturbed flows, all models achieved better performance when the adaptive TCS was operating; this finding demonstrates that both the RCS and the TCS can contribute to reduced travel time. Overall, variations of the ABM RCM hybridized with best response and inertia behavior achieved the best performance. On the basis of the unified framework, properties of some

choice rules in these RCMs could be obtained according to the difference in their evolution curves. This knowledge is 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, 28*), in which the TCS is the leader and the users in the RCS are followers, with some forms of pricing strategies to reduce the price of anarchy among multiple noncooperative commodities (*29*). Second, it is natural to extend this framework to investigate system behavior under the influence of mixed cooperation and competition among multiple traffic ISPs (*15, 30*). Third, the agent-based framework is a natural platform for describing the heterogeneity of user behavior, for example, departure time choice (*31, 32*), online stochastic routing (*33*), travel time reliability (*34*), and risk preference (*35*), and for characterizing and understanding route choice behavior in time-dependent flows.

ACKNOWLEDGMENTS

This research was supported in part by the Traffic21 Initiative at Carnegie Mellon University (CMU), with grants from the Hillman Foundation, the Heinz Endowments, the Richard King Mellon Foundation, the T-SET University Transportation Center at CMU, 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, No. 6, 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. Presented at 92nd Annual Meeting of the Transportation Research Board, Washington, D.C., 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-Routing and Traffic Control. *Transportation Research Part B,* Vol. 45, No. 9, 2011, pp. 1389–1409.
10. Meneguzzer, C. Review of Models Combining Traffic Assignment and Signal Control. *Journal of Transportation Engineering,* ASCE, 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 *Proceedings of 7th International Joint Conference on Autonomous Agents and Multiagent Systems,* AAMAS 2008, May, Estoril, Portugal, Curran Publishing Services Ltd., Norwich, United Kingdom, 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, No. 18, 2012, pp. 1343–1358.
19. Feng, Y., and L. Head. A Hierarchical Agent-Based Simulation for Modeling Traveler Behaviors. Presented at 92nd Annual Meeting of the Transportation Research Board, Washington, D.C., 2013.
20. Xie, X.-F., S.F. Smith, and G.J. Barlow. Schedule-Driven Coordination for Real-Time Traffic Network Control. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling,* AAAI Press, Menlo Park, Calif., 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, No. 10, 2009, pp. 984–994.
22. Chen, O.J., and M.E. Ben-Akiva. Game-Theoretic Formulations of Interaction Between Dynamic Traffic Control and Dynamic Traffic Assignment. In *Transportation Research Record 1617,* TRB, National Research Council, Washington, D.C., 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. Krajzewicz, D., J. Erdmann, M. Behrisch, and L. Bieker. Recent Development and Applications of SUMO – Simulation of Urban MObility. *International Journal on Advances in Systems and Measurements,* Vol. 5, Nos. 3–4, 2012, pp. 128–138.
28. Swamy, C. The Effectiveness of Stackelberg Strategies and Tolls for Network Congestion Games. *ACM Transactions on Algorithms,* Vol. 8, 2012, p. 36.
29. Balcan, M.-F., A. Blum, and Y. Mansour. Circumventing the Price of Anarchy: Leading Dynamics to Good Behavior. *SIAM Journal on Computing,* Vol. 42, No. 1, 2013, pp. 230–264.
30. Yang, I., and R. Jayakrishnan. Modeling Framework to Analyze Effect of Multiple Traffic Information Service Providers on Traffic Network Performance. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2333,* Transportation Research Board of the National Academies, Washington, D.C., 2013, pp. 55–65.
31. Zhang, L., M. Mollanejad, C. Xiong, and S. Zhu. Agent-Based Approach for Integrating Departure Time and Dynamic Traffic Assignment Models. Presented at 92nd Annual Meeting of the Transportation Research Board, Washington, D.C., 2013.
32. Han, K., T.L. Friesz, and T. Yao. Existence of Simultaneous Route and Departure Choice Dynamic User Equilibrium. *Transportation Research Part B,* Vol. 53, 2013, pp. 17–30.
33. Du, L., S. Peeta, and Y.H. Kim. Online Stochastic Routing Incorporating Real-Time Traffic Information. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2334,* Transportation Research Board of the National Academies, Washington, D.C., 2013, pp. 95–104.
34. Carrion, C., and D. Levinson. Value of Travel Time Reliability: A Review of Current Evidence. *Transportation Research Part A,* Vol. 46, No. 4, 2012, pp. 720–741.
35. 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.

The Transportation Network Modeling Committee peer-reviewed this paper.