Adaptive learning algorithms for traffic games with naive users

Toshihiko Miyagi\textsuperscript{a}, Genaro Peque, Jr.\textsuperscript{c}, Junya Fukumoto\textsuperscript{b}

\textsuperscript{a} Professor, Graduate School of Information Science, Tohoku University, Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan
\textsuperscript{b} Associate Professor, Graduate School of Information Science, Tohoku University, Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan
\textsuperscript{c} PhD Student, Graduate School of Information Science, Tohoku University, Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan

**Abstract**

In this paper, we consider a traffic game where many atomic agents try to optimize their utilities by choosing the route with the least travel cost, and propose an actor-critic-based adaptive learning algorithm that converges to $\epsilon$-Nash equilibrium with high probability in traffic games. The model consists of an N-person repeated game where each player knows his action space and the realized payoffs he has experienced but is unaware of the information about the action(s) he did not select. We formulate this traffic game as a stochastic congestion game and propose a naive user algorithm for finding a pure Nash equilibrium. An analysis of the convergence is based on Markov chain. Finally, using a single origin destination network connected by some overlapping paths, the validity of the proposed algorithm is tested.

© 2013 The Authors. Published by Elsevier Ltd. Open access under CC BY-NC-ND license.

Selection and peer-review under responsibility of Delft University of Technology

**Keywords:** traffic games; atomic user; naive user algorithm; pure Nash equilibrium; network simulation

1. Introduction

This paper is motivated by behavioral experiments on travelers’ route-choice that was conducted by Selten et al. [18]. Their experimental results clarify the problems involved in the assumptions that has been made in the traditional procedures for modeling of travelers’ route choice behaviors: 1) each user knows his utility function, 2) each user knows other users’ utility functions, 3) each user observes other users’ behavior, 4) each user can compute the minimum path based on the average travel time on each link, and 5) all users know that there exists an equilibrium in the system. We introduce the concept of a naive user, for whom we don’t require any of the above assumptions to describe. On the other hand, we define an informed-user as a person that assumptions (1) and (3) can be applied to.

This paper is also stimulated by an agent-based route choice model based on cellular automata for vehicle traffic flow [15,16], in which travelers’ trip costs are determined by simulation-based experience not by artificial link cost functions and are stochastically changeable. In order to relate the results of traffic flow simulation to the route choice behavior of the traveler, we need a model on how each traveler forms his expectation from realized travel times that are obtained by day-to-day travel experiences.

The purpose of this paper is to propose a naive user algorithm which describes the route-choice behaviors of a user in a traffic network comprised of a number of discrete, interactive decision-makers. We describe the traffic game as a state-independent stochastic game. This implies that our model is essentially a repeated game. The learning process of the naive user is closer to the so-called reinforcement learning [2, 19] though it differs in the way that simultaneous moves of more than three persons are involved in the game so that the process is inherently non-stationary. A naive user algorithm is conceptually similar with a particle SUE [16], however, our approach includes a forward-looking behavior in trip costs and doesn’t make the assumption that there is an equilibrium in the system. In fact, we show in this paper that there doesn’t always exist a pure Nash equilibrium.

For the naive user problem with non-atomic games, Leslie and Collins proposed the multiple-timescales algorithm [10] and the individual Q-learning algorithm [11]. Cominetti et al. [5] used almost the same approach...
as Leslie and Collins and proved that a logit learning rule converges to a unique equilibrium point under the condition on the dispersion parameter (hereafter we call it as a logit-parameter) included in the logit choice model. The individual Q-learning algorithms [11] and the algorithm adopted in Cominetti et al. [5] are called the payoff-based algorithm where only the payoff evolution process is allowed to evolve. In addition, in both approaches, the determination of the logit-parameter is inconsistent with the assumption of the naive user in the sense that the determination of logit-parameter requires more than the information that an individual can obtain through his experiences. Marden et al. [12] proposed a simple payoff-based algorithm for the naive user problem that converges to a pure Nash equilibrium with probability $\rho < 1$. Their model is characterized by a model-free approach and by a realized payoff-based model, not the estimated payoff-based used in this paper. Our approach is similar with their method, but provides a more general model that includes their model as a special case. The convergence proof is also different.

A naïve user algorithm proposed here is a natural extension of the informed-user algorithms by Miyagi and Peque [13] and aims to provide a combined learning algorithm for finding a pure Nash equilibrium in the congested network with atomic users. Our algorithm includes a simultaneous updating scheme of payoff learning and strategy learning with autonomously updating logit-parameter. This means that the estimated payoffs must be consistent with those that are used in the mixed strategies. However, the users doesn’t know the payoffs that they have not selected, and the cost functions as well. Therefore, we cannot use a fixed point algorithm proposed by Cantrarella [4]. Our algorithm relies on best response with the estimated payoffs that are recursively updated by a stochastic approximation scheme. We will show that the algorithm converges to a $\varepsilon$-Nash equilibrium. The convergence proof is based on Markov chain.

In our naive user algorithms, we use a logit model derived from stochastic fictitious play. The logit-parameter is recursively updated based on regret defined by the realized payoffs [7]. Therefore, if each user successfully selects a better action, the regret asymptotically converges to a very small number, and identification ability of each user to select the best action is improved.

This paper is organized as follows. In section 2, we present the notation and definition related to traffic games. Section 3 is the main part of this paper; therein we discuss the naive-user algorithm and show its convergence property. Section 4 shows simulations of the test network with a single origin-destination and 3 routes, we simulated the case of both using linear and non-linear link cost functions, respectively. In addition, in order to show that the algorithm is useful in a generic user equilibrium problem, computational examples in cases of networks with multi-user, step-cost function and cost function with asymmetric Jacobian, respectively are presented. In the last section, we provide our concluding remarks.

2. Notation and definition

Let $I = \{1, \ldots, i, \ldots, n\}$ and $A^i = \{1, \ldots, k, \ldots, m^i\}, \forall i \in I$, denote the set of players and the set of actions of player $i$, respectively. We interchangeably use a notation $a' \in A^i$ and $k \in A^i$ to represent a generic element of $A^i$. We use the conventional notation $a^{-i} \in A^{-i}$ to describe the action taken by the opponents of $i$, $a^{-i}$, where $A^{-i}$ is the action set of the opponent(s) of player $i$. The action profile is a vector denoted by $a = (a', \ldots, a', \ldots, a^N) \in A$, or $a = (a', a^{-i}) \in A$ where $A = A' \times \cdots \times A^N$. The payoff (or utility) of player $i$ in a one-shot game is determined by the function $u^i : A \to \mathbb{R}$.

Suppose that at the end of each stage, player $i$ observes a sample $U^i_t$ which is a realized payoff that player $i$ receives at stage $t$. We assume that the realized payoff consists of the average payoff $u^i(a', a^{-i})$ and a choice specific random term $\varepsilon^i(a')$. That is,

$$U^i_t = u^i(a'_t, a^{-i}) + \varepsilon^i_t(a'_t)$$  \hspace{1cm} (1)

where $u^i(\cdot)$ is the real-valued function and $\varepsilon^i(a'_t)$ is a component of the random variables vector $\varepsilon'_i = (\varepsilon_{i1}', \varepsilon_{i2}', \ldots, \varepsilon_{im^i}')$. An analyst knows the functional form of each user’s payoff function, $u^i(a'_t)$, however each of the users doesn’t.

Consider a discrete time process $\{U^i_t\}_{t=0}^\infty$ of vectors. At each stage $t$, all players having observed the past realizations $U^i_1, \ldots, U^i_{t-1}$, choose the actions $a^i_t$ in $A$. The objective of player $i$ is to maximize the expected payoff define by

$$\mathbb{E}\left[ \liminf_{t \to \infty} \frac{1}{t} \sum_{s=1}^{t} U^i_s \right]$$  \hspace{1cm} (2)
A mixed strategy $\pi_i(a')$ represents the probability that player $i$ chooses an action $a'$ at time $t$, i.e.

$$\pi_i(a') = P[a_i = a'] \tag{3}$$

**Definition 1 (Naive user).** Each user doesn’t know his/her payoff function and those of other travelers as well. In addition, each user does not know the state and the number of participants of the game. The only information available to him/her is the realized payoff that he/she has used at that day, $U_i(t)$.

In standard game theory, each player is assumed to have a belief that her opponents’ behave independently accordingly to mixed strategies so that the average payoff in the mixed strategy space is written as:

$$u'(\pi) = \mathbb{E}_i[u'(a)] = \sum_{a \in \mathcal{A}} u'(a) \prod_{j \neq i} \pi_j(a')\tag{4}$$

(A mixed) Nash equilibrium is achieved when each player plays a best response to the opponent strategies, so that

$$u'(\pi_i) = \max_{a' \in \mathcal{A}} u'(a', \pi_i) \tag{5}$$

**Definition 2 (Pure Nash equilibrium).** A pure Nash equilibrium of a game is defined as an action profile that satisfies the condition:

$$u'(a_i, a') = \max_{a' \in \mathcal{A}} u'(a', a_i') \tag{6}$$

A traffic game with atomic flow was proposed by Rosenthal [17] and is well known as a (deterministic) congestion game. Congestion game is a special case of potential game [14], and has a pure strategy Nash equilibrium. Congestion game is also known to be a special case of weakly acyclic game [20, 21] and has the property that for any action $a \in \mathcal{A}$, there exists a better reply path starting at $a$ and ending at some pure Nash equilibrium of the game [12, 20]. On the other hand, the properties of stochastic traffic games are not so much known. We are interested in the learning rule such that when used by all the users, stage-by-stage play converges to pure Nash equilibrium play most of the time.

3. Adaptive learning algorithms for congestion games

3.1. Flow and cost

We begin with flow conservation equations in traffic games with atomic flow. For simplicity purposes, throughout this paper we will restrict our attention to a single origin-destination (O-D) pair connected by paths (routes). Each player shares a set of facilities (or link) of the transportation network, $L$. The action space of player $i \in I$ is $\mathcal{A} = 2^L$. That is, a player’s action corresponds to choosing a path $k \in \mathcal{A}$ that consists of a set of link on that path, $\ell \in k$. The number of visits to path $k \in \mathcal{A}$ by player $i$ at time $t$ is denoted by a 0-1 variable and defined by

$$z_{k,i}^t = \mathbf{1}[a_i^t = k] \tag{7}$$

where $\mathbf{1}[*]$ is the indicator function that takes the value of 1 if the statement in the parenthesis is true, and zero, otherwise. Therefore, the number of players who use link $\ell$ is a positive integer value and given as

$$f_{i,\ell} = \sum_{k \in \mathcal{A}} \delta_{i,k} z_{k,i}^t, \quad \forall \ell \in L \tag{8}$$

where $\{\delta_{i,k}\}$ is an element of the link-path incidence matrix.

Equation (8) can be equivalently expressed by using link-path incidence matrix as:

$$f_{i,\ell} = \sum_{i \in I} \sum_{k \in \mathcal{A}} \delta_{i,k} z_{k,i}^t$$

An analyst may formulate link travel cost on $\ell \in L$ as a real-valued non-decreasing function, $C_i(\ell)$. Travel cost of path $k \in \mathcal{A}$ is defined as

$$c_k^i = \omega \sum_{k \in \mathcal{A}} \delta_{i,k} C_i(\ell), \tag{9}$$

where $\omega$ denotes the value of time of user $i$. We define the payoff of path $k$ as $u_k^i = -c_k^i$. Thus, the payoff function is no longer continuous with respect to the flow. By using the experienced payoff of user $i$ at time $t$, the action specific average payoff, $U_i^t$, received by user $i$ at time $t$ (excluded) is given by

$$U_k^i = \frac{1}{Z_{k,i}} \sum_{t=0}^{\infty} U_k^i z_{k,i}^t$$
where \( Z^i_t(k) \) denotes the number of visits of path \( k \) up to \( t \) defined as:
\[
Z^i_{k,t} = \sum_{s=0}^{t-1} \pi^i_{k,s}
\]

Then we can show that
\[
\hat{u}^i_{k,t} = \hat{u}^i_{k,t-1} + \frac{1[\alpha^i_{k,t} = k]}{Z^i_{k,t}} (U^i_{t-1} - \hat{u}^i_{k,t-1})
\]

(10)

Proof is shown in Appendix A.

3.2. Stochastic congestion games

In a naive user environment, since users will always judge their best action based on the estimated payoffs \( \hat{u}^i(k) \), they would take a probabilistic rather than a deterministic behavior. Following Fudenberg and Levine [6], we define the smooth best response function obtained by solving
\[
x^i_t = \arg \max_{x \in \Delta} \left\{ \sum_{k \in A'} \pi^i_{k,t} \hat{u}^i_{k,t} + \mu \psi'(\pi^i_{k,t}) \right\}
\]

(11)

where \( \mu > 0 \) is a person-dependent, smoothing parameter and \( \psi' : \Delta(A') \rightarrow \mathbb{R} \) is a private information of player \( i \), which is a smooth, strictly differentiable concave function. A typical example of the private information function is the entropy function
\[
\psi'(\pi^i_t) = -\sum_{k \in A'} \pi^i_{k,t} \log \pi^i_{k,t}
\]

Then the following logit functions are derived:
\[
x^i_t = \frac{\exp\left\{ \hat{u}^i_{k,t} / \mu^i \right\}}{\sum_{m \in d'} \exp\left\{ \hat{u}^i_{m,t} / \mu^i \right\}} , k \in A', i \in I
\]

(12)

Since we don’t know the explicit relations between \( x \) and \( \hat{u} \) in (12), we cannot find the equilibrium state of the choice structure. In addition to this, in order for \( \{x^i_t\}_{t=0} \) to be the best response for user \( i \), it is necessary that the recursively estimated payoffs asymptotically approximate the realized payoffs. To achieve this, we use the stochastic approximation theory [3, 9].

Note that (10) includes a person-dependent random variable \( Z^i_{k,t} \) that changes according to user \( i \’s \) choice history. We approximate (10) with the following updating scheme:
\[
\hat{u}^i_{k,t+1} = \hat{u}^i_{k,t} + \frac{1[\alpha^i_{k,t} = k]}{x^i_t} (U^i_{t} - \hat{u}^i_{k,t})
\]

(13a)

where for each \( i \), \( \{ \lambda^i_t \}_{t=0} \) is a deterministic sequence satisfying
\[
\sum_{t=0}^{\infty} \lambda^i_t = \infty, \sum_{t=0}^{\infty} (\lambda^i_t)^2 < \infty
\]

(13b)

The convergence property of (13a) can be approximated by ordinary differential equations (ODE) [9]. If the system is in a stationary process wherein each user acts independently and assumes that other users keep their previous actions, the estimated payoffs will converge to the expected values of the realized payoffs (see Appendix B for a detailed discussion). In a more general case where all users change their actions simultaneously, the system exhibits a non-stationary process. Leslie and Collins [10,11] proved that the parameter \( \lambda \) in (13a) should be a user specific parameter \( \{ \lambda^i_t \} \) and the following additional condition is required,
\[
\frac{\lambda^i_t}{\lambda^{i+1}_t} \rightarrow 0 \text{ as } t \rightarrow \infty
\]

(13c)

Using the theorem on the perturbed ODE proposed by Benaim [1], Leslie and Collins [11] provided the following lemma:

Lemma 1. The values \( \hat{u}^i_t \) resulting from (13a) with the user-specific parameter \( \lambda \) converges almost surely to a connected internally chain-recurrent set of the flow defined by the singularly perturbed ODE provided that the \( \hat{u}^i_t \) remain bounded for all time.
3.3. A naive user algorithm

3.3.1. Markov Learning Process

We consider a general class of congestion games where each player takes a better response strategy rather than a best response. Better response may be rational in cases where each user’s behavior is perturbed due to private information and/or that link cost is dependent on the flows on other links. Our approach is based on the Markov learning model which describes the transition process where a state of the game is the level of a learner. At each time step \( t = 1, 2, \ldots \), let \( s_t \) denote the state of the game. We define the state variable \( s \) as the number of learners, that is, the players whose the best replies in two consecutive steps are in accordance with and the \( (n-s) \) out of the \( n \) players as non-learners. We label the action choice of non-learners, zero. On the other hand, a learner is the player who has succeeded in finding the best reply. The learner is going to stay with the action in the succeeding steps. We label the action of the learner, one. Figure 1 shows the state transitions for three persons with zero and one actions. This traffic game is a weakly acyclic game \([20, 21]\) and does not necessarily have the finite improvement property because the graph includes a cycle.

![State Transition Diagram](image)

Figure 1. A weakly acyclic property in the state transition

Let \( p_s \) denote the probability that a player chooses the best action in a state \( s \). Thus, \( (1 - p_s) \) represents the probability that a player will fail to choose the best action. Then, the transition probability is given as

\[
\Pr[s_{t+1} = s + 1 \mid s_t = 1] = \frac{n-s}{n} p_s, \text{ for } 0 \leq s \leq n-1
\]

The remaining transition probabilities are depicted in figure 2. In the figure we implicitly assume that

\[
\Pr[s_{t+1} = s' \mid s_t = 1] = 0, \text{ for } s' \notin \{s-1, s, s+1\}
\]

![Transition Probabilities Diagram](image)

Figure 2. Transition probabilities

Although an analyst knows the current state, each player does not know which state he is currently in. This Markov chain model has nice properties, it is irreducible and aperiodic, and converges to a steady transition probability \( \gamma \) that can be found by solving the following equation \([8]\).
3.3.2. Algorithm

We consider the process such that given the action profile and the resultant action values, \( \{a_t, U_t\} \), \( \{a_t, U_t\}_{t=0} \) are sequentially generated. Suppose that player \( i \) only knows the realized payoff \( U^i_t \) at each stage \( t > 0 \). Since behavioral strategies are time-dependent and simultaneously changed, users encounter a non-stationary environment. At the initial stage \( t = 0 \), each user randomly selects the initial action \( a^i_0 \). User \( i \) assigns it as his tentative best action (TBA) \( a^i_0 \). At each subsequent time step, a randomly selected user, say user \( i \), chooses his tentative best action \( k^* \) by exploration and includes it in the set of best actions \( B^i_t = \{ k^* \in A^i : k^* = \arg \max_{a \in A^i} \hat{u}^i(a) \} \) (15)

where a vector \( \hat{u}^i \) is given by (15). User \( i \) chooses his tentative best action \( k^* \) by the following rule:

\[
\pi^i_t(a') = \begin{cases} 
(1 - \epsilon)(a) + (1 - \epsilon), & \text{for } a' = k^* \\
\epsilon(1 - \pi^i_t(k^*))/|A^i| - 1, & \text{for all } a' \neq k^*
\end{cases}
\]

where \( \epsilon \) is an arbitrary small positive number. The choice probabilities of TBA is given as \( \pi^i_t(a') = K \exp\{\hat{u}^i(a')/\mu^i_t\}, \text{ for } a' = k^* \) (16)

where \( K \) is the normalization factor. The logit-parameter \( \mu^i_t \) is updated by
\
\mu^i_t = \mu^i_{t-1} + \frac{1}{t} (R^i_t - \mu^i_{t-1}), \text{ where } R^i_t = U^i_t - \bar{U}^i_t
\]

with \( \bar{U}^i_t = \frac{1}{t} \sum_{t=0}^{t-1} U^i_t \). (18)

Equation (18) means that the logit-parameter in our algorithm is autonomously updated according to the users’ experiences. The regret \( R^i_t \) approaches zero as stage \( t \) becomes a very large value [7]. The new action profile \( a^i_t \) is generated according to the mixed strategy \( \pi^i_t \). The first equation in (16) implies that user \( i \) assigns at least the probability \( \epsilon \pi^i_t(k^*) \) to each action and reinforces the TBA by assigning the increment. If we put \( \pi^i_t(a') = 1/|A^i| \), this scheme is equivalent to what is called the \( \epsilon \)-greedy algorithm in reinforcement learning [19]. Equation (16) together with (17) represents the re-planning (or adaptive learning) process of user \( i \) depending on the case if he has succeeded in choosing the best action or has failed.

An alternative for determining the best response set is as follows. Suppose that each user saves in his memory the payoffs that he has received up to \( t - 1 \) and searches the maximum value among them,

\[
U^i_{t,\max} = \max_{0 \leq s \leq t - 1} U^i(a^i_s), a^i_s \in A^i.
\]

The next action of user \( i \) is determined by comparing the actual payoff received, \( U^i(a^i_t) \), with the maximum received payoff, \( U^i_{t,\max} \), and is updated as follows:

\[
a^i_{t+1} = \begin{cases} 
a^i_t, & U^i(a^i_t) > U^i_{t,\max} \\
k^*, & U^i(a^i_t) \leq U^i_{t,\max}
\end{cases}
\]

If the TBA is repeated, the user stays there in the subsequent time periods. If the selected action according to the mixed strategy is not consistent with the TBA, each user chooses the TBA cautiously according to the conditional logit model (16) and (17).

3.4. Convergence property
Proposition. Suppose that the congestion game has a pure Nash equilibrium. Then the action profile \( a \), generated by the naive user algorithm converges to \( \varepsilon \)-Nash equilibrium with probability \( \rho < 1 \) for a sufficiently small \( \varepsilon \) and for all sufficiently large times \( t \).

Sketch of proof: First, for the stationary probabilities, it holds that

\[
\gamma_{s} = \frac{s+1}{n-s} \frac{1}{p_s}, \quad 0 \leq s \leq n-1
\]

where \( p_s = 1 - \varepsilon(1 - x_s^*) \) and \( x_s^* \) is a choice probability which is assigned to the tentative best response path at state \( s \). Let \( \rho = \min_{0 \leq s \leq n} p_s \) be the lower bound of the probabilities of selecting a tentative best action. We can choose \( \varepsilon \) to ensure \( \eta = (1 - \rho)/\rho < 1 \). If we choose \( \varepsilon \leq 1/n \), then it holds that

\[
\gamma_0 \leq \cdots \leq \gamma_{n-1} \leq \gamma_n.
\]

Assume that the final state where all players achieve in finding the best responses occurs with probability one, that is, \( \gamma_n = \rho = 1 \). This involves \( \gamma_0 = \cdots = \gamma_{n-1} = 0 \) and that all users almost surely converge to pure Nash equilibrium. This can occur in the informed user problem, however, in the naive user problem \( \rho \) must be less than one. From (21), for a given \( \gamma_n = \rho \), we can build the sequential equations \( \{\gamma_{n-1}, \ldots, \gamma_0\} \) and the sum of those values

\[
H = \sum_{s=0}^{n-1} \gamma_s = \left( \sum_{s=0}^{n-1} \frac{n-1-s}{n} \eta^s + \cdots + n \frac{n-1}{2} \eta^s + n \eta^s \right) \rho
\]

Using the binomial formula, we have

\[
H = \rho \left\{ (1 + \eta)^n - 1 \right\}.
\]

Thus, we have

\[
\rho = \frac{1}{(1 + \eta)^n} = \left[ 1 - \varepsilon(1 - x^*) \right]^n < 1
\]

If the minimum choice probability that is assigned to the tentative best action \( x^* \) is close to one, then the probability of the state that all users succeed in finding their best paths, \( \rho \), approaches to very close value of one. However, adopting a very small value of \( \varepsilon \) brings in a negative effect. At the relatively early stages of the iteration a user will assign a very high probability to the wrong path and fixes it. The algorithm should be constructed to allow some users to be in their suboptimal states. If you add the lower ranked probabilities, which means that some users are allowed to achieve better responses, that is, \( \rho = \gamma_n + \gamma_{n-1} + \cdots + \gamma_0 \), then \( \varepsilon - \text{Nash equilibrium} \) can be achieved with relatively high probability.

4. Simulations

The proposed algorithm is applied to a single origin-destination (O-D) network using linear and non-linear cost functions on the links, respectively. The network has 5 links and 3 routes; this translates to 3 actions available for each player. Players must traverse from node O to node D and must do this repeatedly until they converge to a pure Nash equilibrium. The network models the complex interaction of players using the links and the proposed algorithm ensures that this complex interaction leads to an efficient use of more general networks.

4.1. Test network and link cost functions

We use a Braess-network shown in Figure 3. We pay attention to the single O-D case where the flow conservation equation is described as \( n = h_1 + h_2 + h_3 \), where \( n \) is the number of trips and \( h_k, k \in \{1,2,3\} \) denotes the \( i^{th} \) path flow. The following linear link cost functions are assumed:

\[
\text{link } 1: C_1 = 4 f_1, \text{link } 2: C_2 = 50 + f_2, \text{link } 3: C_3 = 50 + f_3, \text{link } 4: C_4 = 4 f_4, \text{ and link } 5: C_5 = 28 + f_5,
\]

where \( C_i(\cdot) \) is the travel cost of link \( i \) and \( f_i \) is the aggregated link flow on link \( i \). Under the condition that link costs are monotone increasing, and continuously differentiable with flow, there exists a unique Wardrop equilibrium that is consistent with a mixed Nash equilibrium. However, in network with atomic users, that is not
the case. We selected an example that there is a unique atomic UE. For the non-linear link cost functions, we assume the Bureau of Public Roads (BPR) type-cost function,

\[ C_{\ell}(f_{\ell}) = C_{0\ell}(1 + 0.15 \left( \frac{f_{\ell}}{K_{\ell}} \right)^4) \]  

(22)

where \( C_{0\ell} \) is the free flow travel time, \( K_{\ell} \) is the capacity of link \( \ell \).

Figure 3. Test network

4.2. Simulation for a basic network

Figure 4 depicts the flow variation among the three paths (the vertical axis represents flow). The resulting equilibrium flow vector is \( h^* = (4, 2, 4) \), which is the exact values of the Wardrop equilibrium flow on this network. Different from the informed user problem, the naive user problem path flows fluctuate for a long time but with diminishing flow-variation. This implies that it takes a long time for drivers to learn the correct values of the travel costs and that after a better response path converges to a pure NE, the best action periodically moves from the point to neighboring points around the NE point. This characterizes a connected internally chain-recurrent set of the best action.

Figure 4. Flow variations

In figure 5 each edge of each cell corresponds to a state of potential field. The figure shows a better response path starting from the big red dot to a unique absorbing point (a pure NE) designated by a green dot. Since the initial starting point is randomly selected, the better response path takes different trajectory during each simulation.
4.3. Multi-user network with different values of time

Figure 6 shows the flow variation of the same network with 10 users. In figure 4, users had the same value of time, \( \omega = 1 \). In figure 6, 4 of the 10 users had a value of time \( \omega = 1.5 \) while the remaining users had \( \omega = 1 \). The equilibrium in this network is \([3 4 3]\) where \([h_1 h_2 h_3]\) represents route 1, route 2 and route 3, respectively. Users with value of time \( \omega = 1.5 \) chose the 2nd route where a toll is to be paid while the remaining users choose the routes without a toll. This simulation shows that the users with a higher value of time are willing to pay to arrive at their destination faster.

![Figure 5. A better response path from starting point (red dot) to the final point (green dot)](image)

![Figure 6. Flow variations on a network with 10 users with different values of time](image)

4.4. Network with an asymmetric Jacobian link cost

In figure 7, we let the same network use asymmetric Jacobian link costs. Asymmetric Jacobian link costs are used to model multimodal and multiclass networks which provide a more realistic user interaction. In this network, we have 10 users with an equilibrium of \([5 1 4]\). The values of time for all users is \( \omega = 1 \). The following linear link cost functions were used:

\[ C = A + Bf \]

where
In figure 8, we used the same network with 10 users but this time using a step cost function. The equilibrium in this setting is \([4 \ 2 \ 4]\) and all the users have the same value of time \(\omega = 1\). The link costs assumed were the same except for link 5 which is dependent on the number of users, \(f_5\), where 

\[
C_5 = \begin{cases} 
17 & \text{if } f_5 \leq 2, \\
21 & \text{if } f_5 = 3, \\
28 & \text{if } f_5 \leq 4, \\
35 & \text{if } f_5 \leq 6, \\
15 + f_5 & \text{else}
\end{cases}
\]

Figure 8. Flow variations on a network with 10 users using a step cost function
5. Concluding remarks

A naive user algorithm simulates agents' route choice behavior in a realistic and plausible traffic condition where each agent determines its best action based on traffic information obtained through his own experiences. The learning process in the naive user problem is closer to the reinforcement learning in machine learning though it differs in the way that simultaneous moves of more than three persons are involved in the game so that the process is inherently non-stationary. This paper proposed a naive user algorithm for finding the ε-Nash equilibrium in a congested transportation network. Our algorithm is applicable to a large scale network by using the k-path minimum search. Dial algorithm is an alternative for determining action space. Once, the action space is determined, our algorithm no longer requires further exploration of finding minimum-paths. This may bring in a dramatic reduction of computing time. On the other hand, our algorithm needs a large number of iterations to adjust predicted action-values. Although stochastic approximation theory is useful to build the prediction of action-values, further updating is necessary to reduce computing time. In our model, logit-type model or model free approach is selective for finding the best action. A model free approach is recommendable for practical use but the setting of parameter associated with ‘inertia’ is still open to question. Our approach is an atomic-based model so that it is useful in bridging the route choice behaviors to the network simulator using a cellular automaton that are used in transportation planning packages like MATSim (Multi-Agent Transport Simulation) and TRANSIMS (Transportation Analysis and Simulation System). In this paper, we have restricted our attention to the atomic model, however, the same algorithm is also applicable to traffic games with non-atomic users.

Acknowledgements

This research is supported by MEXT Grants-in-Aid for Scientific Research, No. 22360201, for the term 2010-2012.

References


Appendix A.

Equation (13a) can be developed as

\[ \hat{u}_{i,j}^t = \frac{1}{Z_{i,j}^t} \sum_{s=0}^{\infty} U_{i,j}^s 1_{|a_i^t = k} + \frac{U_{i,j}^t 1_{|a_i^t = k}}{Z_{i,j}^t} \]  

(A.1)
By substituting for
\[ \hat{u}_{k,t-1}^i = \frac{1}{Z_{k,t-1}^i} \sum_{s=0}^{t-2} U_{s}^i \mathbf{1}\{a_s^i = k\} \]

It follows that
\[
\begin{align*}
\hat{u}_{k,t}^i &= \frac{1}{Z_{k,t}^i} \left[ U_{k,t-1}^i \mathbf{1}\{a_{k,t-1}^i = k\} + Z_{k,t-1}^i \hat{u}_{k,t-1}^i \right] \\
&= \frac{1}{Z_{k,t}^i} \left[ U_{k,t-1}^i \mathbf{1}\{a_{k,t-1}^i = k\} + (Z_{k,t}^i - \mathbf{1}\{a_{k,t-1}^i = k\}) \hat{u}_{k,t-1}^i \right]
\end{align*}
\]

Thus, we have
\[
\hat{u}_{k,t}^i = \hat{u}_{k,t-1}^i + \frac{1}{Z_{k,t}^i} \left( \mathbf{1}\{a_{k,t-1}^i = k\} - \hat{u}_{k,t-1}^i \right) (U_{k,t-1}^i - \hat{u}_{k,t-1}^i)
\] (A.1)

Appendix B.

The expectation of the estimated action values (13a) is given as:

\[
\begin{align*}
\mathbb{E}_x \left[ \hat{u}_{k,t}^i - \tilde{u}_{k,t-1}^i \right] &= \lambda \mathbb{E}_x \left[ \mathbf{1}\{a_t^i = k\} / x_{k,t}^i \right] \mathbb{E}_x \left[ u'(a_{k,t-1}^i, a_{k,t-1}^i) - \hat{u}_{k,t-1}^i \right] \\
&= \lambda \left( u'(a_{k,t-1}^i, x_{k,t-1}^i) - \tilde{u}_{k,t-1}^i \right) = \lambda G_i^t(\hat{u}_{k,t-1}^i)
\end{align*}
\]

with
\[ G_i^t(\hat{u}_{k,t-1}^i) = u'(a_{k,t-1}^i, x_{k,t-1}^i) - \hat{u}_{k,t-1}^i \]

Let
\[ G_i^t(\hat{u}_{k,t-1}^i) = (G_i^t(\hat{u}_{k,t-1}^i), \ldots, G_i^t(\hat{u}_{k,t-1}^i)) \]

be a vector of \( G_i^t(\hat{u}_{k,t-1}^i) \), \( \forall k \in A^i \). Since, for any \( x \neq y \), it holds that
\[
\|G_i^t(x) - G_i^t(y)\| \leq \|x - y\|.
\]

\( G_i^t(\hat{u}_{k,t-1}^i) \) is Lipschitz continuous with \( L = 1 \).

The noise terms in (13a) is then given as
\[ \xi_{k,t}^i = \left[ \left( U_{k,t-1}^i - \hat{u}_{k,t-1}^i \right) \mathbf{1}\{a_t^i = k\} - G_i^t(a_{k,t-1}^i, \pi_{k,t-1}^i) \right] \]

\( \{\xi_{k,t}^i\} \) are martingale differences. Therefore, (B.1) is expressed by the following stochastic approximation equations:
\[ \hat{u}_{k,t}^i = \hat{u}_{k,t-1}^i + \lambda \left( G_i^t(a_{k,t-1}^i, \pi_{k,t-1}^i) + \xi_{k,t}^i \right) \] (B.2)

where the step size parameter \( \lambda \) satisfies the conditions (13b). Since \( G_i^t \) is Lipschitz continuous, we assume that \( G_i^t \) is bounded, then (B.2) is approximated by the ODE
\[
\frac{d\hat{u}_{k,t}^i}{dt} = u'(a_t^i, x^{-i}(\hat{u}_{k,t}) - \hat{u}_{k,t}^i(a_t^i)
\]

which clearly has a globally asymptotically stable fixed point \( x^{-i} \). That is, \( \|u'(a_t^i, x^{-i}(\hat{u}_{k,t}) - \hat{u}_{k,t}^i(a_t^i))\| \rightarrow 0 \) as \( t \rightarrow \infty \) a.s.