9-2021

Routing policy choice prediction in a stochastic network: Recursive model and solution algorithm

Tien MAI
Singapore Management University, atmai@smu.edu.sg

Xinlian YU
Southeast University

Song GAO
University of Massachusetts Amherst

Emma FREJINGER
University of Montreal

Citation

MAI, Tien; YU, Xinlian; GAO, Song; and FREJINGER, Emma. Routing policy choice prediction in a stochastic network: Recursive model and solution algorithm. (2021). *Transportation Research Part B: Methodological*. 151, 42-48.
Available at: https://ink.library.smu.edu.sg/sis_research/6215

Routing policy choice prediction in a stochastic network: Recursive model and solution algorithm

Tien Mai[a], Xinlian Yu[b], Song Gao[c], Emma Frejingerd[d]

a School of Computing and Information Systems, Singapore Management University, Singapore
b School of Transportation, Southeast University, China
c Department of Civil & Environmental Engineering, University of Massachusetts Amherst, United States of America
d Department of Computer Science and Operational Research and CIRRELT, Université de Montréal, Canada
Corresponding author: atmai@smu.edu.sg (T. Mai), xinlianyu@seu.edun.cn (X. Yu)

Published in Transportation Research Part B: Methodological, (2021 September), 151, 42-58. DOI: 10.1016/j.trb.2021.06.016

Abstract:

We propose a Recursive Logit (STD-RL) model for routing policy choice in a stochastic time-dependent (STD) network, where a routing policy is a mapping from states to actions on which link to take next, and a state is defined by node, time and information. A routing policy encapsulates travelers' adaptation to revealed traffic conditions when making route choices. The STD-RL model circumvents choice set generation, a procedure with known issues related to estimation and prediction. In a given state, travelers make their link choice maximizing the sum of the utility of the outgoing link and the expected maximum utility until the destination (a.k.a. value function that is a solution to a dynamic programming problem). Existing recursive route choice models and the corresponding solution approaches are based on the assumption that network attributes are deterministic. Hence, they cannot be applied to stochastic networks which are the focus of this paper.

We propose an efficient algorithm for solving the value function and its gradient, critical for parameter estimation. It is based on partitioning the state space and decomposing costly matrix operations into a series of simpler ones. We present numerical results using a synthetic network and a network in Stockholm, Sweden. The estimation running time has a 20-30 times speed-up due to matrix decomposition. The estimated model parameters have realistic interpretations. Specifically, travelers are more likely to be adaptive to realized travel times during a longer trip, and more sensitive to travel time when travel time variability is higher. The STD-RL model performs better in predicting route choices than the RL model in a corresponding static and deterministic network.

Keywords: Routing policy choice, Stochastic time-dependent networks, Recursive logit, Decomposition algorithm

1. Introduction

Predicting traffic flow in road networks is of importance in many transport applications. Discrete choice models based on the random utility maximization framework are often used for this purpose and model parameters can be estimated by maximizing likelihood defined over trajectory data. The route choice modeling literature mainly focuses on networks where link or path attributes are static and deterministic even though most models can be extended to a deterministic and time-dependent case. A recent exception is de Moraes Ramos et al. (2020) that compares static and dynamic travel time representations in terms of estimation and prediction results.

The assumption of deterministic network attributes is restrictive since many transportation systems are inherently uncertain, for instance, due to incidents and weather. Real-time traffic information can allow travelers to adapt their route choices to unfolded traffic conditions and thus potentially mitigate the adverse impact of uncertainty for travelers. In recent years, the advances in telecommunication and information technologies coupled with the prevalence of smartphones and in-vehicle navigation systems have made real-time traffic information increasingly a norm, instead of a privilege. It is so prevalent that the diversion of traffic to local communities by Waze users has become newspaper headlines as residents complain about disruptions (Littman, 2019). It is therefore imperative that route choice models incorporate travelers' adaptation to traffic information (see, e.g., Zhang et al., 2018; Moghaddam et al., 2019; Zhang et al., 2019; Jiang et al., 2020, for some recent studies on route choice with traffic information). Existing recursive route choice models and corresponding solution algorithms cannot be used for this purpose as they crucially depend on network attributes being deterministic.

In this paper, we focus on modeling adaptive route choices (i.e., routing policy choices) in stochastic and time-dependent (STD) networks where link travel times are random variables with time-dependent distributions. This is considerably more challenging than modeling non-adaptive route choices, i.e., path choices, in deterministic networks. We propose a Recursive Logit (RL) model (see, e.g., Fosgerau et al., 2013; Nassir et al., 2019; de Moraes Ramos et al., 2020, for its application in deterministic auto and transit networks) for an STD network and present estimation and prediction results for a case study in the city of Stockholm. In the following we provide a background on related literature and outline the paper contributions.

1.1. Background and contributions

Ding-Mastera et al. (2019) tackle the adaptive route choice modeling problem in an STD network with revealed preference (RP) data, following a series of previous work on adaptive route choice based on synthetic and laboratory data (e.g., Gao et al., 2008; Razo and Gao, 2013). The adaptive behavior is encapsulated in the definition of a routing policy, which is a mapping from states to actions on which link to take next, and a state is defined by node, time and information. A routing policy can be realized as different paths depending on the realizations of travel times; a path is a special routing policy where the action is independent of time or information. Travelers are assumed to choose a routing policy at the origin, instead of a fixed path. They generate a choice set of routing policies for each origin–destination (OD) pair by generalizing a number of path choice set generation algorithms, and develop a Policy-Size Logit model where Policy Size is a generalization of Path Size (Ben-Akiva and Ramming, 1998) to account for overlapping of routing policies.

The disadvantages of the approach in Ding-Mastera et al. (2019) are those inherent to choice set generation for route choice modeling. Even in the simpler network setting of deterministic travel times, it is challenging to design models that can be consistently estimated and that can produce accurate forecasts in short computational time, in particular for large-scale networks. The literature can be divided into path-based and link-based (recursive) models. The former relies on choice sets of paths that are sampled using a path generation algorithm, typically a kind of repeated shortest path search (Prato, 2009, presents a survey). While consistent parameter estimates can be obtained by correcting the utilities for the choice set sampling (e.g., Frejinger et al., 2009; Guevara and Ben-Akiva, 2013; Lai and Bierlaire, 2015), it is unclear how to accurately define path choice sets for prediction. Ad-hoc procedures may lead to inaccurate forecasts and counter-intuitive accessibility measures (Nassir et al., 2014; Zimmermann et al., 2017). Moreover, sampling choice sets of paths is computationally costly and it does not scale well, since the number of paths connecting each OD pair increases exponentially with the size of the network. The disadvantage of sampling alternatives becomes more acute in modeling routing policy choice, since calculating an optimal routing policy in an STD network, a basic procedure in the process, is more computationally expensive than calculating the shortest path in a deterministic network. Furthermore, a possible sampling bias has not yet been investigated.

Motivated by these issues, a link-based RL model is proposed by Fosgerau et al. (2013) that allows to estimate path choice models in deterministic networks without sampling any choice sets of paths. Moreover, the RL model is straightforward and fast to use for prediction. The model has been extended to deal with correlated utilities and to relax the Independence from Irrelevant Alternative (IIA) assumption (Mai et al., 2015, 2018; Mai, 2016). The RL model is based on the dynamic discrete choice framework, where the choice of path corresponds to a sequence of link choices. Travelers make their link choice maximizing the sum of the utility of the outgoing link and the expected maximum utility until the destination (a.k.a. value function that is a solution to a dynamic programming problem).

The objective of this paper is to develop STD-RL, an RL model for STD networks, so that routing policy choice models can be estimated without sampling any choice sets and used for prediction at a reasonable computational cost. The key challenge lies in the solution of the value function, which in this case is more expensive to solve than for deterministic networks. Indeed, the state space is large due to the representation of joint realizations of random travel times and discretization of the study period into small time intervals. Unlike in the deterministic case, state transition probabilities are not degenerate, which renders a non-linear system rather than a linear system as in the deterministic case.

This paper makes both methodological and empirical contributions. Firstly, we propose an RL model for STD networks and we propose an algorithm for solving the value function. The key to the efficient solution algorithm is to decompose the state space, which is finite but very large, into two sub-spaces: one corresponding to the physical network, and one that contains information. This allows us to decompose the computation into simple matrix operations. Secondly, we present estimation and prediction results for a network in Stockholm, Sweden, and demonstrate the computational efficiency, modeling realism and accurate predictive performance of the proposed model and solution algorithm.

1.2. Paper structure

The remainder of the paper is structured as follows. In Section 2, we delineate the STD-RL model for adaptive route choice and define the corresponding log-likelihood function. We describe the decomposition method for maximum likelihood estimation in Section 3. Section 4 presents numerical results from a synthetic network and a case study using GPS trajectory data in a sub-network of Stockholm, Sweden. Finally, conclusions and future research directions are presented in Section 5.

2. A recursive logit model in stochastic time-dependent networks

In this section we describe the recursive model formulation for routing policy choice in networks where travel times are uncertain. The model is based on the same behavioral assumptions as previous work on routing policy choice (e.g., Ding-Mastera et al., 2019). Namely, we assume that travelers have access to real-time traffic information and they may adapt their route choices en route based on this information. This means that their choice cannot be represented by a simple path choice at the origin. Consider for example a traveler who, en route, observes congestion and adjusts her route choice based on her new expectation of travel time to the destination. The behavioral assumption is hence that the travelers are forward-looking — they plan their route to the destination and adapt their plan according to the information they receive. Crucial in this context is how to formalize information access. In this section we therefore start by introducing the network representation and describe how we formalize the information concept. We note that it corresponds to the perfect online information (POI) representation in Gao and Chabini (2006). We then introduce the recursive logit formulation that we propose (Section 2.2). In Section 2.3 we define choice probabilities and discuss maximum likelihood estimation for this new model. As the routing policy choice is latent, we hence link the recursive routing policy choice model to the observed *paths*.

2.1. The stochastic network and traveler information

We consider a physical transportation network represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ and $\mathcal{A}$ are the sets of nodes and links, respectively. For each node $k \in \mathcal{N}$, we denote by $A(k)$ the set of outgoing nodes from k. The study period, e.g., AM peak, is discretized into equal-length intervals. A time interval is represented by a non-negative integer, $t = \lfloor \tilde{t}/\delta \rfloor$, where $\tilde{t}$ is the original time and the length of a time interval δ is equal to the shortest link travel time. The set of time indices, $\mathcal{T}$, starts from 0 and covers the study period.

Link travel times are jointly distributed, time-dependent random variables, which leads to an STD network. A support point ω^r, $r = 1, 2, \dots, R$, is defined as a distinctive vector of joint realization of link travel times at all times during the study period with a dimension $|\mathcal{A}||\mathcal{T}|$. The joint distribution of the time-dependent link travel time variables is thus represented by the set of support points $\{\omega^1, \omega^2, \dots, \omega^R\}$.

The set of states is $S = \{(k, t, \mathbf{q}) \mid \ k \in \mathcal{N}, t \in \mathcal{T}, \mathbf{q} \in \mathbf{Q}(t)\}$, where k is the node, t the time and $\mathbf{q}$ the event collection. Assume the information that a traveler has at node k and time t is represented by realized travel time values revealed to the traveler (through, say, a smartphone app) on a subset of time-dependent link travel time random variables. The corresponding index set in a support point is denoted $\mathcal{I}(k, t)$. Let $\{\pi_i, i \in \mathcal{I}(k, t)\}$ represent the set of realized link travel times. An event collection, $\mathbf{q}$, is a representation of the information that is conducive to the calculation of attribute values and state transition probabilities (Gao and Chabini, 2006). It is composed of support points that are consistent with realized link travel times revealed to the traveler at node k and time t, that is, $\mathbf{q} = \{\omega^r | \omega^r_i = \pi_i, \forall i \in \mathcal{I}(k, t)\}$. As the traveler moves in the network and collects potentially more information, the size of $\mathcal{I}(k, t)$ increases or remains the same, and as a result the size of the event collection decreases or remains the same. That is, the network becomes potentially less stochastic. When the event collection becomes a singleton, the network is deterministic. Define $\mathbf{Q}(t)$ as the set of all possible event collections at time t, and $\mathbf{Q}(t)$ can be generated a priori for a given process of information progression over time. Note that any predictive information, whether provided by Google Maps or deduced by experienced drivers, is based on observed travel times in the past. We do not differentiate by the entities who perform the prediction, but go to the sources and define the information by realized travel times. Furthermore, we denote the destination node by $d \in \mathcal{N}$, and by S^d the corresponding set of absorbing destination states, $S^d = \{(d, t, \mathbf{q}) \mid \ t \in \mathcal{T}, \mathbf{q} \in \mathbf{Q}(t)\}$.

We illustrate the support point, event collection and routing policy concepts in a small example. It is based on the information reported in Fig. 1 and Table 1. We assume that travelers' information only comes from their own experience on traversed links. Moreover, for the sake of simplicity, we assume a time-independent network so the focus lies is on the stochastic travel times and the role of information. The path (k_1, k_2, k_4) represents a highway subject to congestion, while (k_2, k_3, k_4) is a detour from the highway starting from a midpoint k_2.

There are two support points, ω^1 and ω^2, each represented by a vector of joint realization of link travel times in Table 1. Support point ω^1 can be viewed as the "free flow" scenario where travel times on both highway links are low, while ω^2 can be viewed as the "congested" scenario where travel times on both highway links are high. The detour has a fixed travel time regardless of the support point.

Consider a traveler going from k_1 to k_4. There is just one possible event collection $\{\omega^1, \omega^2\}$ at the origin k_1. The event collection $\{\omega^1, \omega^2\}$ then splits into two event collections, $\{\omega^1\}$ and $\{\omega^2\}$, at node k_2, when the traveler learns of the realized travel time on (k_1, k_2). Information on traversed links thus helps the traveler to make inferences about the future. Specifically, if the travel time on (k_1, k_2) is one, the traveler can infer that s/he is in support point ω_1 and the travel time on (k_2, k_4) is also one. Thus, it is better

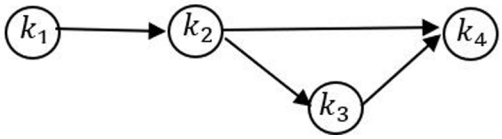

Fig. 1. An illustrative network.

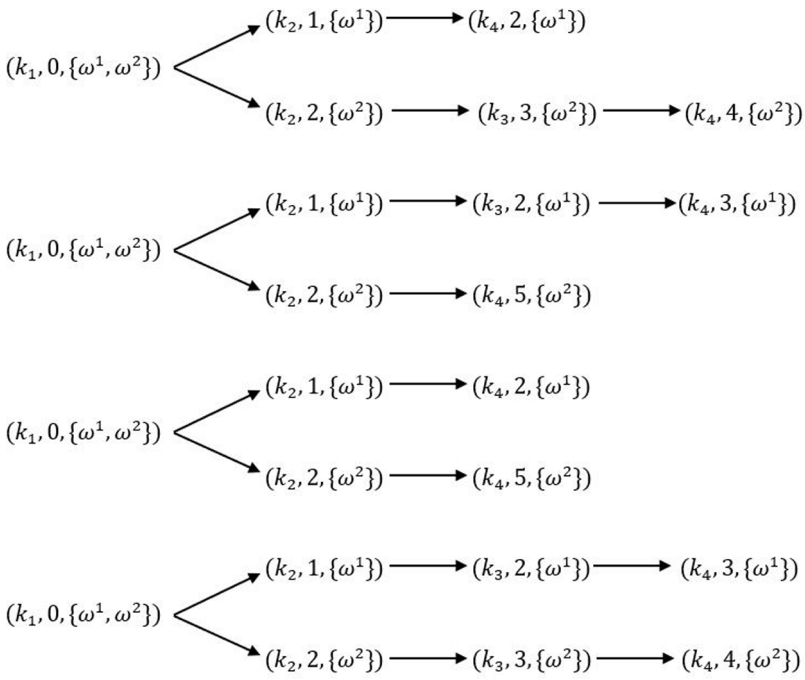

Fig. 2. Illustration of all the possible routing policies.

Table 1
Stochastic link travel times.

Link(s)	Support Points	
	ω^1	ω^2
(k_1, k_2)	1	2
(k_2, k_4)	1	3
$(k_2, k_3), (k_3, k_4)$	1, 1	1, 1

to stay on the highway. If the travel time on (k_1, k_2) is two, the traveler can infer that s/he is in support point ω_2 and the travel time on (k_2, k_4) is three. Thus, it is better to take the detour.

There are four possible routing policies in this example. The traveler starts from an initial state $(k_1, 0, \{\omega^1, \omega^2\})$, arriving at node k_2 with two possible states, $(k_2, 1, \{\omega^1\})$ or $(k_2, 2, \{\omega^2\})$, due to the stochastic link travel time on link (k_1, k_2). At state $(k_2, 1, \{\omega^1\})$, the traveler may continue on the highway, arriving at the destination with state $(k_4, 2, \{\omega^1\})$, or take the detour and arrive at destination with state $(k_4, 3, \{\omega^1\})$. Similarly, at state $(k_2, 2, \{\omega^2\})$, the traveler continue on the highway, arriving at the destination with state $(k_4, 5, \{\omega^2\})$, or take the detour and arrive at the destination with state $(k_4, 4, \{\omega^2\})$.

The four policies are illustrated in Fig. 2. The upper two are truly adaptive, in the sense that the decision at node k_2 depends on realized travel time on link (k_1, k_2), in other words, event collection $\{\omega^1\}$ or $\{\omega^2\}$. Travelers who follow a truly adaptive routing policy therefore could take different paths, depending on the manifested network conditions. The lower two routing policies correspond to paths, in that the decision at node k_2 does not change with the event collection. Note that a truly adaptive routing policy is not necessarily good. For example, the second policy takes the highway while it is congested and takes the detour when the highway is not congested.

2.2. The recursive logit model for routing policy choice

At each state $s \in S$, an instantaneous utility, $u(a|k,t,\mathbf{q};\beta) = v(a|k,t,\mathbf{q};\beta) + \mu\epsilon(a)$, is associated with the action of choosing node $a \in A(k)$. These utilities depend on the parameter vector β associated with attributes of the action. In order to simplify the notation, henceforth we omit the explicit dependence on β. Assuming a Logit choice model at each state, the random terms $\epsilon(a)$ are i.i.d. extreme value type I with zero mean and they are independent of everything else in the model. The traveler chooses the next node sequentially assuming that he/she maximizes the sum of the instantaneous utility $u(a|k,t,\mathbf{q})$ and the expected maximum downstream utility (until reaching the destination). The latter is defined by taking the continuation of this sequential process into account via the Bellman equation. Specifically, the expected maximum utility at state $(k,t,\mathbf{q})$, $V^d(k,t,\mathbf{q})$ with a destination node d, also known as the value function, is defined as follows:

$$V^d(k,t,\mathbf{q}) = \begin{cases} \mathbb{E}\left[\max_{a\in A(k)}\left\{v(a|k,t,\mathbf{q}) + \mu\epsilon(a) + \rho\sum_{\mathbf{q}'\in\mathbf{Q}(t')} P\left(\mathbf{q}' \mid \mathbf{q}\right) V^d\left(a,t',\mathbf{q}'\right)\right\}\right], & \forall (k,t,\mathbf{q}) \in S\backslash S^d, \\ 0, & \forall (k,t,\mathbf{q}) \in S^d, \end{cases} \tag{1}$$

where $0 < \rho \leq 1$ is a discount factor, $t' = t + \tau(a|k,t,\mathbf{q})$ is the arrival time at the chosen node a and $\tau(a|k,t,\mathbf{q})$ is the travel time between node k and node a conditional on state $(k,t,\mathbf{q})$. The definition of $\tau(a|k,t,\mathbf{q})$ depends on the information the traveler has access to. If we assume POI, that is, $\mathcal{I}(k,t) = \mathcal{A}\times\{0,1,\ldots,t\}$, the traveler knows realized travel times on all links up to and including time t, and $\tau(a|k,t,\mathbf{q})$ and t' are both deterministic values conditional on state $(k,t,\mathbf{q})$. This is a reasonable assumption with the present sensing and telecommunication technologies (Gao and Chabini, 2006).

The key difference with respect to a deterministic setting (e.g., as the one in Fosgerau et al., 2013) is the stochastic transition to the next state, in that the sum in (1) corresponds to the expected value of $V^d(a,t',\mathbf{q}')$ taken over $\mathbf{Q}(t')$, all event collections $\mathbf{q}'$ at time t'. The conditional probability of $\mathbf{q}'$ given $\mathbf{q}$, $P\left(\mathbf{q}'|\mathbf{q}\right)$, is derived in Gao and Chabini (2006) as follows:

$$P\left(\mathbf{q}' \mid \mathbf{q}\right) = \frac{P(\mathbf{q}'\cap\mathbf{q})}{P(\mathbf{q})} = \frac{\sum_{r|\omega^r\in\mathbf{q}'\cap\mathbf{q}} p_r}{\sum_{r|\omega^r\in\mathbf{q}} p_r}, \tag{2}$$

where p_r is the probability of support point r. Note that $\mathbf{q}' \subseteq \mathbf{q}$, that is, an event collection at a later time is no larger than that at the current time.

Under the assumption that $\epsilon(a)$ are i.i.d. extreme value type I, the value function is the well-known "logsum",

$$V^d(k,t,\mathbf{q}) = \begin{cases} \mu\ln\left(\sum_{a\in A(k)}\exp\left(\frac{1}{\mu}\left(v(a|k,t,\mathbf{q}) + \rho\sum_{\mathbf{q}'\in\mathbf{Q}(t')} P\left(\mathbf{q}' \mid \mathbf{q}\right) V^d(a,t',\mathbf{q}')\right)\right)\right), & \forall (k,t,\mathbf{q}) \in S\backslash S^d. \\ 0, & \forall (k,t,\mathbf{q}) \in S^d. \end{cases} \tag{3}$$

The transition probability from state $(k,t,\mathbf{q})$ to state $\left(a,t',\mathbf{q}'\right)$ is given by the product of the probability of choosing node a given state $(k,t,\mathbf{q})$ – $P^d(a|k,t,\mathbf{q})$ – and the probability of $\mathbf{q}'$ at $t' = t + \tau(a|k,t,\mathbf{q})$ given $\mathbf{q}$ – $P(\mathbf{q}'|\mathbf{q})$. The former is given by the Logit model. The latter is given by Eq. (2), and its dependence on the choice of node a is implicitly accounted for in the arrival time t' at node a. Combining these, the transition probability is

$$\begin{aligned} P^d(a,t',\mathbf{q}'|k,t,\mathbf{q}) &= P^d(a|k,t,\mathbf{q})P(t',\mathbf{q}'|\mathbf{q},a) \\ &= P^d(a|k,t,\mathbf{q})P(\mathbf{q}'|\mathbf{q}) \\ &= \frac{\exp\left(\frac{1}{\mu}\left(v(a|k,t,\mathbf{q}) + \rho\sum_{\mathbf{q}'\in\mathbf{Q}(t')} P(\mathbf{q}'|\mathbf{q})V^d(a,t',\mathbf{q}')\right)\right)}{\sum_{a'\in A(k)}\exp\left(\frac{1}{\mu}\left(v(a'|k,t,\mathbf{q}) + \rho\sum_{\mathbf{q}''\in\mathbf{Q}(t'')} P(\mathbf{q}''|\mathbf{q})V^d(a',t'',\mathbf{q}'')\right)\right)} P(\mathbf{q}'|\mathbf{q}) \\ &= \exp\left[\frac{1}{\mu}\left(-V^d(k,t,\mathbf{q}) + v(a|k,t,\mathbf{q}) + \rho\sum_{\mathbf{q}'\in\mathbf{Q}(t')} P(\mathbf{q}'|\mathbf{q})V^d(a,t',\mathbf{q}')\right)\right] P(\mathbf{q}'|\mathbf{q}). \end{aligned} \tag{4}$$

While the derivation in the case of an STD network follows the same steps as Fosgerau et al. (2013), the resulting model has different properties which leads to distinct challenges. The value function (3) is a non-linear system (as opposed to linear in the deterministic case) over a larger state space. Indeed, the state space is defined over all possible values of $(k,t,\mathbf{q})$ as opposed to all k for a deterministic and static network, or all (k,t) for a deterministic and dynamic network. Before describing a solution approach in Section 3, we next define choice probabilities and the associated likelihood function.

2.3. Choice probabilities and maximum likelihood estimation

A routing policy cannot be observed as it is a contingency plan in a decision maker's mind. An observation is thus the realization of a routing policy in a specific STD network, that is, a sequence of states $\sigma = \{(k_1,t_1,\mathbf{q}_1),\ldots,(k_I,t_I,\mathbf{q}_I)\}$ that corresponds to path $\{k_1,\ldots,k_I\}$. Consider the illustrative example from the previous section, where four routing policies are presented in Fig. 2. A modeler might observe a sequence $(k_1,0,\{\omega^1,\omega^2\}),(k_2,1,\{\omega^1\}),(k_4,2,\{\omega^1\})$, however, this could be the result of a traveler taking the first routing policy, which realizes as an observed path (k_1,k_2,k_4) in support point ω^1, or that of a traveler simply taking path (k_1,k_2,k_4), namely, the third routing policy.

The probability of σ given the initial state $(k_1, t_1, \mathbf{q}_1)$ is the joint probability of the sequence of subsequently observed states, in this case the product of probabilities (4):

$$P(\sigma) = \prod_{i=1}^{I-1} P^{k_I}(k_{i+1}, t_{i+1}, \mathbf{q}_{i+1} | k_i, t_i, \mathbf{q}_i). \tag{5}$$

As demonstrated in Fosgerau et al. (2013), in a deterministic network, the RL is equivalent to a Logit model at the origin with a universal choice set of paths. Such an equivalence does not hold for the proposed STD-RL model in an STD network. That is, (5) is not equivalent to a Logit model at the origin with a universal set of routing policies. The latter corresponds to the so-called "non-RL" model that is studied in Ding-Mastera et al. (2019). The reader is referred to Yu et al. (2020) for a detailed conceptual discussion with numerical examples on the comparison of the RL and non-RL models of routing policy choice. Moreover, the Independence from Irrelevant Alternatives property (IIA) does not hold for ratios of path probabilities (5). This is trivial to see as each probability in the product depends on value functions associated with all alternatives at each choice stage that do not cancel out in the product. In turn, the probabilities depend on the states of the network.

We use the nested fixed point algorithm (Rust, 1987) for maximum likelihood estimation of the instantaneous utility parameters β. In brief, this algorithm combines an outer iterative nonlinear optimization algorithm for searching over the parameter space with an inner algorithm solving the value function. The outer optimization problem maximizes the log-likelihood function defined over observations $n = 1, \dots, N$,

$$LL(\beta) = \sum_{n=1}^{N} \sum_{i=0}^{I_n - 1} \ln P^{k_{I_n}}(k_{i+1}^n, t_{i+1}^n, \mathbf{q}_{i+1}^n | k_i^n, t_i^n, \mathbf{q}_i^n), \tag{6}$$

where I_n is the number of observed states in observation n. We use first-order nonlinear optimization algorithms. We can compute the gradient of (6) via first-order derivatives of probabilities $P^{k_{I_n}}(k_{i+1}^n, t_{i+1}^n, \mathbf{q}_{i+1}^n | k_i^n, t_i^n, \mathbf{q}_i^n)$. The partial derivative $P^d(a, t', \mathbf{q}' | k, t, \mathbf{q})$ with respect to a parameter β is

$$\begin{aligned} &\frac{\partial \ln P^d(a, t', \mathbf{q}' | k, t, \mathbf{q})}{\partial \beta} \\ &= \frac{1}{\mu} \left(-\frac{\partial V^d(k, t, \mathbf{q})}{\partial \beta} + \frac{\partial v(a|k, t, \mathbf{q})}{\partial \beta} + \rho \sum_{\mathbf{q}' \in \mathbf{q}(t')} \frac{\partial V^d(a, t', \mathbf{q}')}{\partial \beta} P(\mathbf{q}' | \mathbf{q}) \right). \end{aligned} \tag{7}$$

Thus, $V^d(k, t, \mathbf{q})$ and $\frac{\partial V^d(k,t,\mathbf{q})}{\partial \beta}$ are needed to compute the log-likelihood and its gradient. Given that the state space can be very large, the value functions are destination specific and they need to be computed repeatedly, an efficient solution algorithm is crucial. For this purpose, we present a decomposition algorithm in the following section.

3. Decomposition algorithm for maximum likelihood estimation

The nested fixed point algorithm (Rust, 1987) combines an outer iterative nonlinear optimization algorithm (e.g., trust region or line search algorithm) for searching over the parameter space with an inner algorithm solving value functions. Given that the inner algorithm is called a large number of times, it is crucial to compute value functions quickly. Furthermore, gradients of value functions are needed for the search over the parameter space. In this section we start by defining the value function and the corresponding gradient. With the purpose of speeding up the computations, we then propose to decompose costly matrix operations into several simpler ones. We do this without changing the network representation, that is, we solve the value functions for all states.

3.1. Computation of the value function and its gradient by value iteration

We consider the stochastic network with the set of states $S = \{(k, t, \mathbf{q}) |\ k \in \mathcal{N}, t \in \mathcal{T}, \mathbf{q} \in \mathbf{Q}(t)\}$. For notation simplicity, the destination index is dropped in the remainder of the presentation. For each state $s = (k, t, \mathbf{q}) \in S$ we define mappings $n(s) = k \in \mathcal{N}$, $\pi(s) = t$, and $i(s) = \mathbf{q}$, i.e., $n(s)$ refers to the physical node in the transportation network associated with state s, $\pi(s)$ refers to the time, and $i(s)$ refers to the event collection $\mathbf{q} \in \mathbf{Q}(t)$. We define the set of links in the stochastic network as $\mathcal{A}^S = \{(s, s') |\ n(s') \in A(n(s)), P(i(s')|i(s)) > 0\}$, where $P(i(s')|i(s))$ is defined in (2). In other words, two nodes s and s' in S are connected if their corresponding physical nodes are connected, and the transition probability from $i(s)$ to $i(s')$ is greater than zero. For notation simplicity, we define $W(s'|s)$ such that

$$W(s'|s) = \begin{cases} P(i(s')|i(s)) & \text{if } (s, s') \in \mathcal{A}^S \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The value function as defined in (3) can be written as

$$V(s) = \begin{cases} \mu \ln \left(\sum_{a \in A(n(s))} \exp \left(\frac{1}{\mu} v(a|s) + \frac{\rho}{\mu} \sum_{\substack{s' \in S \\ n(s')=a}} V(s') W(s'|s) \right) \right) & s \in S \backslash S^d \\ 0 & s \in S^d. \end{cases} \tag{9}$$

If we define a vector Z of size $|S|$ with elements $Z_s = \exp(\frac{V(s)}{\mu})$, $\forall s' \in S$, the value function can be rewritten as

$$Z_s = \begin{cases} \sum_{a \in A(n(s))} \exp\left(\frac{1}{\mu} v(a|s)\right) \prod_{\substack{s' \in S \\ n(s')=a}} Z_{s'}^{\rho W(s'|s)} & \text{if } s \in S \backslash S^d \\ 1 & \text{if } s \in S^d. \end{cases} \tag{10}$$

Now we define matrices M and U of size $|S| \times |\mathcal{N}|$

$$M_{sa} = \begin{cases} \exp\left(\frac{1}{\mu} v(a|s)\right) & \forall s \in S, a \in A(n(s)) \\ 0 & \text{otherwise,} \end{cases} \quad U_{sa} = \begin{cases} \prod_{\substack{s' \in S \\ n(s')=a}} Z_{s'}^{\rho W(s'|s)} & \forall s \in S, a \in A(n(s)) \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

as well as a vector b of size $|S|$ such that $b_s = 0$, if $s \in S \backslash S^d$, and $b_s = 1$ if $s \in S^d$. The recursive formulation in (10) can then be written as

$$Z = (M \circ U)e + b, \tag{12}$$

where e is a vector of size $|\mathcal{N}|$ with unit entries, and $\circ$ is the element-by-element product. Based on this recursive formulation, we could compute the value of Z by simple value iteration. More precisely, at the beginning (iteration $l = 0$), we start with an initial value Z^0. At iteration l, given that the current value of vector Z is $Z^{(l)}$, we update the next value Z^{l+1} by performing the two following steps.

1. Compute $U^{(l)}$ by (11), using $Z^{(l)}$,
2. $Z^{(l+1)} \leftarrow (M \circ U^{(l)})e + b$.

We stop the iterative process when a fixed point solution is found, i.e., $\|Z^{(l+1)} - Z^{(l)}\| \leq \gamma$ for a given threshold $\gamma > 0$. The method converges to a unique fixed point solution after $\mathcal{O}(\log \gamma^{-1})$ iterations, if $\rho < 1$ (Rust, 1987).

The first-order derivatives of the log-likelihood function are critical for the maximum likelihood estimation. Similar to the computation of the value function, we formulate the computation of the gradient as matrix operations, which then can be decomposed into a sequence of simpler steps. The following proposition shows that the Jacobian of Z in (12) can be obtained by solving a system of linear equations.

Proposition 1. *For each parameter β_j, the gradient vector of Z w.r.t. β_j is a solution to the following system of linear equations*

$$\frac{\partial Z}{\partial \beta_j}(I - D) = J^j e, \tag{13}$$

where I is the identity matrix of appropriate size, J^j and D are matrices of size $(|S| \times |\mathcal{N}|)$ and $(|S| \times |S|)$, respectively, with entries

$$J^j_{sa} = \frac{1}{\mu} K_{sa} \frac{\partial v(a|s)}{\partial \beta_j}, \ \forall s \in S, a \in \mathcal{N} \tag{14}$$

$$D_{ss'} = \frac{\rho K_{sa} W(s'|s)}{Z_{s'}}, \ \forall s, s' \in S, a = n(s'), \tag{15}$$

where K is matrix of size $|S| \times |\mathcal{N}|$ with entries

$$K_{sa} = e^{\frac{1}{\mu} v(a|s)} \prod_{\substack{s' \in S \\ n(s')=a}} Z_{s'}^{\rho W(s'|s)}, \ \forall s \in S, a \in N.$$

Proof. The proof is straightforward taking the first-order derivative of Z_s given in (12). See Appendix. ■

Moreover,

$$\frac{\partial V(s)}{\partial \beta_j} = \frac{\mu}{Z_s} \frac{\partial Z_s}{\partial \beta_j}, \ \forall s \in S,$$

which allows us to compute the gradient of the log-likelihood function using (6) and (7). We note that the linear system in (13) has a unique solution if $(I - D)$ is invertible. We do not have explicit conditions for the invertibility of $I - D$, as they depend, e.g., on the magnitudes of parameters β. We note, however, that provided the magnitudes are large enough, it is invertible.

3.2. Decomposition-based estimation algorithm

While the value function and its gradient can be computed by solving (12) and (13), it is costly due to the large number of states. Matrix U (11) for the computation of Z (12), and matrices J (14) and D (15) for computing $\partial Z / \partial \beta_j$ (13) are critical. In the following, we present a way to compute these matrices through a series of simple matrix operations. The underlying idea is to partition the state space into two subsets: one that characterizes the physical network (deterministic attributes) and the other the probabilistic transitions between states (event collection and travel time distributions).

First, to facilitate the exposition, we define a matrix $F(|S| \times |S|)$ of the state transition probabilities

$$F_{ss'} = W(s'|s), \ \forall s, s' \in S.$$

Let T denote a matrix of size $|S| \times |\mathcal{N}|$ with entries

$$T_{sk} = \begin{cases} 1 & \text{if } k = n(s), \\ 0 & \text{otherwise,} \end{cases} \quad \forall s \in S,$$

recall that $k = n(s)$ represents the node in the physical network associated with s.

Given any matrix X of size $|S| \times |S|$, we denote by $\ln(X)$ and $\exp(X)$ two matrices of the same size with elements

$$[\exp(X)]_{ss'} = \begin{cases} \exp(X_{ss'}) & \text{if } (s, s') \in \mathcal{A}^S \\ 0 & \text{otherwise,} \end{cases} \text{ and } [\ln(X)]_{ss'} = \begin{cases} \ln(X_{ss'}) & \text{if } (s, s') \in \mathcal{A}^S \\ 0 & \text{otherwise.} \end{cases}$$

If X is a matrix of size $|S| \times |\mathcal{N}|$, then $\ln(X)$ and $\exp(X)$ are defined as

$$[\exp(X)]_{sa} = \begin{cases} \exp(X_{sa}) & \text{if } a \in A(n(s)) \\ 0 & \text{otherwise,} \end{cases} \text{ and } [\ln(X)]_{sa} = \begin{cases} \ln(X_{sa}) & \text{if } a \in A(n(s)) \\ 0 & \text{otherwise.} \end{cases}$$

Finally, if X is a vector of size $|S|$, then $\ln(X)$ and $1/X$ are also of size $|S|$ with entries $[\ln(Z)]_s = \ln(Z_s)$, $[1/Z]_s = 1/Z_s$, $\forall s \in S$.

We are now ready to show our main result in Proposition 2 which leads to Algorithms 1 and 2 for computing the value function and the Jacobian, respectively. The algorithms make use of a series of simple matrix operations. While the matrices remain large in size, they are sparse. Importantly, the computational complexity of sparse operations is proportional to the number of non-zero elements and the row and column sizes.

Proposition 2. *The matrices U, J^j, D defined in (11), (14) and (15) can be computed as*

$$\ln(U) = \left[F \times diag\left(\ln(Z)\right)\right] \times T$$
$$J^j = H^j \circ (M \circ U)$$
$$D = \left[\left(K \times T^T\right) \circ F\right] \times diag(1/Z),$$

where H^j is a matrix of size $|S| \times |\mathcal{N}|$ with entries $H^j_{sa} = \partial v(a|s)/\partial \beta_j$, $\forall s \in S, a \in \mathcal{N}$, $diag(Z)$ is a square diagonal matrix with the elements of vector Z on the main diagonal.

Proof. See Appendix. ■

Algorithm 1: Computing the value function V

 # 1. Initializing the value function and the discount factor
$Z = Z^0$; $\rho \leq 1$; $\gamma > 0$
 # 2. Value iteration
do
  $Z_{prev} = Z$
  $Y = \left[F \times \text{diag}\left(\ln(Z)\right)\right] \times T$;
  $U = \exp(Y)$;
  $Z = (M \circ U)e + b$;
while $||Z_{prev} - Z|| < \gamma$;
$V = \mu \ln(Z)$;

Algorithm 2: Computing the Jacobian of the value function Z

$Y = \left[F \times \text{diag}\left(\ln(Z)\right)\right] \times T$;
$K = M \circ \exp(Y)$;
$J^j = H^j \circ (M \circ U)$;
$D = \left[\left(K \times T^{\mathrm{T}}\right) \circ F\right] \times \text{diag}(1/Z)$;
foreach q **do**
  Solve $\frac{\partial Z}{\partial \beta_j}(I - D) = J^j e$ to obtain $\frac{\partial Z}{\partial \beta_j}$;
  $\frac{\partial V}{\partial \beta_j} = \mu \left(\frac{\partial Z}{\partial \beta_j}\right) \circ (1/Z)$;

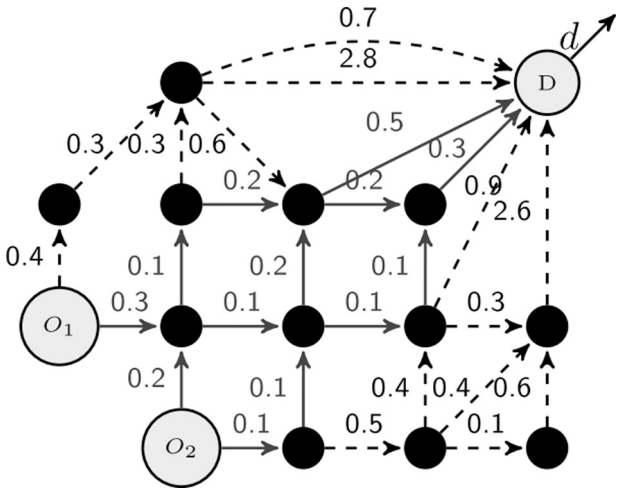

Fig. 3. Small network used for synthetic data generation.
Source: Adapted from Mai (2016b).

4. Numerical experiments

In this section we present two sets of numerical results. First, we use synthetic data to illustrate the difference between the proposed STD-RL model and the deterministic RL model developed in Fosgerau et al. (2013). Second, we use revealed preference (RP) data collected in Stockholm, Sweden to compare the same two models and assess their respective performance in a real large-scale setting. Results from synthetic data show differences between the stochastic and deterministic models as functions of systematically varied parameters, while results from RP data shed light on the pros and cons of the two models as they are tested with real-world data.

4.1. Synthetic data

The proposed STD-RL model is more complex and, as we detail in Section 4.2.4, computationally more costly to estimate, than a deterministic RL model. It is therefore relevant to investigate if a deterministic RL model can be a good approximation of the STD-RL one. The objective of this section is to illustrate how different the deterministic RL model results are in terms of parameter estimates and prediction accuracy, if the true model is an STD-RL one. For this purpose, we use a static network and focus on comparing stochastic versus deterministic settings. We start by presenting the network and the synthetic data generation, followed by an analysis of the results.

Network scenarios. The network (adapted from Mai, 2016b) is shown in Fig. 3. It consists of 15 nodes and 27 links with two origins O_1 and O_2 and one destination D. There are two support points (probability 0.5 each) defining the link travel time distribution. They represent one uncongested scenario and one congested scenario. The number associated with each link in the figure represents the uncongested travel times. A link with a congested travel time different from the uncongested one is depicted in solid line. We define the congested travel time by multiplying the uncongested time of each link by a factor $\gamma(\Delta+1)$, where γ is a random number uniformly distributed in $[0.5, 1.5]$, and $\Delta > 0$ is a scalar and is the same over links. The use of γ here is to avoid the situation that the congested and uncongested link travel times are proportional. We can hence interpret Δ as the level of stochasticity of the network. To illustrate the difference between the STD-RL and deterministic RL models, we present results for different levels of stochasticity, $\Delta \in [0, 5]$. We consider a deterministic RL model, denoted as RE-RL, which is based on a deterministic network with link travel times set as the averages of congested and uncongested ones.

Synthetic data generation. The STD-RL model is assumed to be the true model with a linear-in-parameters function of two attributes, namely, link travel time (TT) and link constant (LC). The latter is included to penalize intersections. We fix the values of the associated parameters to -2.0 and -0.5 for travel time and link constant, respectively. For each level of stochasticity, each OD pair and each scenario, we generate 500 observations. Thus, we have 2000 observations for each level of stochasticity. Furthermore, we divide each of the generated sets of observations into two sets: one used for estimation (80% of the observations) and one (20% of the observations) used to assess prediction accuracy.

Parameter estimates. The observations have been generated assuming STD-RL as a true model, therefore we analyze if the deterministic RL model is nevertheless able to recover the true parameter values. For this purpose we present in the left-hand graph of Fig. 4 the p-values of t-tests where the null hypothesis is that the estimated parameter values are equal to the true ones. As expected, for the deterministic model, the null hypothesis cannot be rejected for the STD-RL models as well as for the deterministic

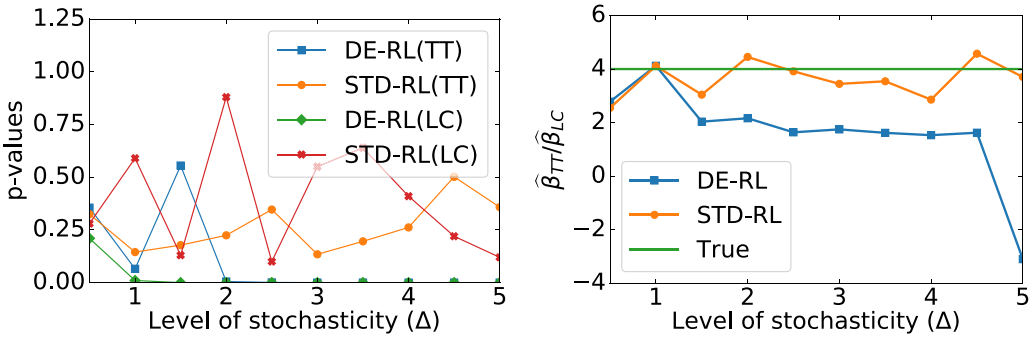

Fig. 4. Comparisons of parameter estimates between the stochastic RL (STD-RL) and deterministic RL (DE-RL) models.

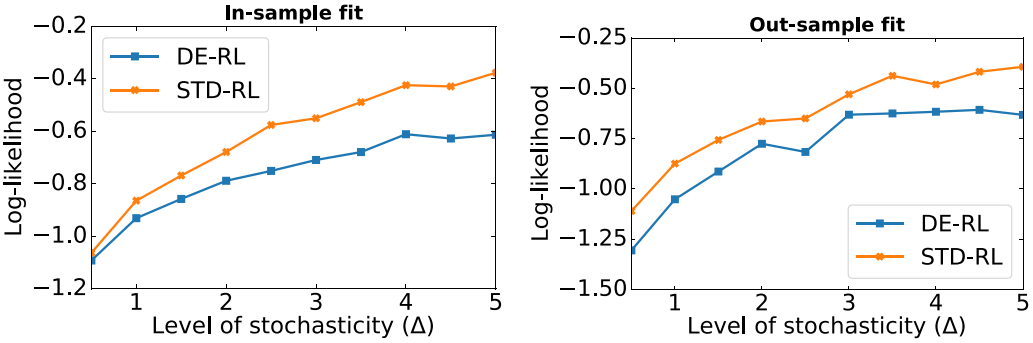

Fig. 5. In and out-of-sample fit comparisons between the stochastic RL (STD-RL) and deterministic RL (DE-RL) models.

model when the level of stochasticity is relatively low ($\Delta \leq 2$). However, for more stochastic settings the parameter estimates of the deterministic RL model are significantly different from the true values. Furthermore, based on the right-hand graph in Fig. 4 we analyze the extent of the differences compared to the true model by plotting parameter ratios. The ratios of the STD-RL model are close to the true ratio, while those given by the deterministic model decrease with increasing values of Δ. In other words, when average link travel times are poor approximations of the true travel times (uncongested and congested scenarios), the parameter estimates can have a strong bias. The travel time parameter even has the wrong sign for $\Delta = 5$. This is consistent with the findings in de Moraes Ramos et al. (2020) albeit in a different setting. They observe a strong bias when comparing parameter estimates from static and dynamic networks using revealed preference data. Biased parameter estimates can have important practical implications if they are used to analyze, for example, value of time measures.

In- and out-of-sample fit. We evaluate the in- and out-of-sample fits given by the STD-RL and DE-RL models. Fig. 5 presents two graphs: in-sample fit (left-hand side) and out-of-sample fit (right-hand side). For each value of Δ (x-axis), we compare the deterministic RL and STD-RL models. Similar to the parameter estimates, the in- and out-of-sample fit are quite similar when the network has a low level of stochasticity but the difference increases with increasing value of Δ. Again we make an observation similar to one made in de Moraes Ramos et al. (2020) when they compare static and dynamic networks on real data: The difference in prediction performance is less pronounced than the difference in parameter estimates. The parameters of deterministic RL model are estimated such that the model achieves the best log-likelihood value. The relative prediction performance is hence improved at the price of biased parameter estimates. Nevertheless, as expected, the prediction performance is worse than that of the true model.

4.2. Revealed preference data

This section is devoted to a numerical analysis of the proposed model using a dataset collected in Stockholm, Sweden. We start by describing the network and the corresponding observed trajectories. We then present estimation and prediction results followed by a discussion on computing times.

4.2.1. Path observations and STD network definition

We consider a subnetwork of Stockholm, Sweden, which includes the Arlanda airport area, the northeast inner city, and the connecting corridor. The data consists of 500 path observations obtained by GPS tracking during morning peak hours (Mondays through Fridays, 6:30 to 9 AM) over a period of 56 days (November 1, 2012 through January 18, 2013). The GPS data was map-matched as part of another project (see, Rahmani and Koutsopoulos, 2013, for details). Furthermore, the data was used for modeling

routing policy choice in Ding-Mastera et al. (2019). We refer to these papers for more details about the data collection, including a map of the study area.

The graph representation of the physical network comprises 2772 nodes and 5447 links. The challenge in defining the related STD network resides in the travel time distributions. For this purpose we rely on the work of Rahmani et al. (2015) which we summarize in the following.

There are 56 support points, one for each day covered by the path observations. Each support point comprises travel times on all links over all 10-second time intervals. A non-parametric method and map-matched GPS data are used to compute the link travel times per time interval for each support point, hence producing an empirical, joint travel time distribution. A link is treated as deterministic when there is not enough variation of travel time over time and day, or not enough observations to derive reliable travel time estimates. In this case, a single mean travel time is estimated across all days and time intervals. The resulting STD network has 619 stochastic links (11.4% of the total number of links in the physical network representation). The effective study period length for a given destination is determined by the maximum duration of observed trips to the destination. Therefore, the number of states ranges from 33,476 to 445,315, and the number of links in the state network between 65,364 and 844,285, depending on the destination.

4.2.2. Instantaneous utility function specification

We specify a linear-in-parameter instantaneous utility function of individual n for link $(k,a) \in \mathcal{A}$ at time t with information $(t,\mathbf{q})$ as

$$\begin{aligned} v^n(a|k,t,\mathbf{q}) = \mathrm{DL}^n\Big(\beta^1_{TT}\mathrm{TT}(a|k,t,\mathbf{q})\mathrm{DHV}(a|k) + \beta^2_{TT}\mathrm{mTT}(a|k)(1-\mathrm{DHV}^n(a|k))\Big) \\ + \ \big(1-\mathrm{DL}^n\big)\,\beta_{mTT}\mathrm{mTT}(a|k) + \beta_{LC}\mathrm{LC}(a|k) + \beta_{PS}\mathrm{PS}^n(a|k). \end{aligned} \tag{16}$$

DL^n and DHV^n are both binary variables, and together they determine which travel time variable and the associated parameter apply to an observed trip by individual n. Results from Ding-Mastera et al. (2019) show that travelers are adaptive in longer trips and not so much in shorter ones. Therefore, we define the dummy-long-trip variable, DL^n, which is equal to 1 if the travel time of trip n is greater then 15 min, and 0 otherwise. It is also conceivable that travelers are more likely to be adaptive when travel time variability is high, since the value of information and adaptation is high in reducing uncertainty. Therefore, we define the dummy-high-variance-travel-time variable, $\mathrm{DHV}^n(a|k)$, which is equal to 1 if the standard deviation of travel time on link (k,a) is more than 20% of the average travel time on link (k,a) over all time periods and support points, and 0 otherwise. As such, the instantaneous utility function, $v^n(a|k,t,\mathbf{q})$, contains the realized travel time $\mathrm{TT}(a|k,t,\mathbf{q})$ and the associated parameter β^1_{TT} only when the trip is longer than 15 min ($\mathrm{DL}^n = 1$) and the travel time variability on link (k,a) is large enough ($\mathrm{DHV}^n(a|k) = 1$). Otherwise, the average travel time variable, $m\mathrm{TT}(a|k)$, averaged over all time periods and support points, is used, with a trip length-specific parameter, that is, β^2_{TT} for a long trip with low variability ($\mathrm{DL}^n = 1$, $\mathrm{DHV}^n(a|k) = 0$), and $\beta_{m\mathrm{TT}}$ for a short trip ($\mathrm{DL}^n = 0$).

Two other variables are included in $v^n(a|k,t,\mathbf{q})$. The link constant $\mathrm{LC}(a|k)$, is designed to penalize intersections, and is equal to 1 for every $(k,a) \in \mathcal{A}$. The Policy Size, $\mathrm{PS}^n(a|k)$ is a heuristic correction to the utility function to penalize overlapping routing policies. The attribute is similar to the Link Size (LS) attribute proposed by Fosgerau et al. (2013) for deterministic networks. We use the fraction of flow that starts at the origin of trip n and arrives at the current node k as an indicator of the level of overlapping for link (k,a). The vector of node flow F of size $|\mathcal{N}|$ is computed by solving the flow conservation equation, $F^n = G^n + (\widetilde{P}^n)^{\mathrm{T}}F^n$, such that $F^n = [I - (\widetilde{P}^n)^{\mathrm{T}}]^{-1}G^n$. We note that $[I - (\widetilde{P}^n)^{\mathrm{T}}]$ is always invertible (Baillon and Cominetti, 2008). The vector of demand G^n of size $|\mathcal{N}|$ has the entry at the origin of trip n equal to 1, and 0 otherwise. $\widetilde{P}^n$ is a matrix of averaged node probabilities computed based on the policy choice model as follows:

$$\widetilde{P}^n(a|k) = \sum_{t\in\mathcal{T}}\frac{1}{|\mathcal{T}|}\sum_{\mathbf{q}\in\mathbf{Q}(t)}\frac{1}{|\mathbf{Q}(t)|}\sum_{t',\mathbf{q}'}P^n(a,t',\mathbf{q}'|k,t,\mathbf{q}),\ \forall a,k\in\mathcal{N}, \tag{17}$$

where $P^n(a,t',\mathbf{q}'|k,t,\mathbf{q})$ is computed based on the utility function defined in Eq. (16), with the term $\beta_{PS}\mathrm{PS}^n(a|k)$ excluded, and parameters $[\beta^1_{TT},\beta^2_{TT},\beta_{mTT},\beta_{LC}] = [-1.8,-3.7,-3.6,-1.0]$. We then set $PS^n(a|k) = F^n(k),\ \forall k,a\in\mathcal{N}$.

We note that both LC and PS variables can be artificially inflated if a node does not represent a real intersection (i.e., a node with both indegree and outdegree equal to one). It is therefore advisable to include a pre-processing step to remove artificial nodes from the network.

4.2.3. Estimation and prediction results

We report estimation results for both STD-RL and DE-RL models for the sake of comparison. For the STD-RL model, we use the aforementioned utility specification of the STD-RL model with different values of the discount factor, $\rho = 0.98, 0.95, 0.90, 0.85$, and for the DE-RL model we employ the following utility function

$$v^n(a|k) = \beta_{aTT}\mathrm{aTT}(a|k) + \beta_{LC}\mathrm{LC}(a|k) + \beta_{LS}\mathrm{LS}^n(a|k), \tag{18}$$

where $\mathrm{aTT}(a|k)$ is the average travel time on link (k,a). This is the same as $\mathrm{mTT}(a|k)$ but we use different notations to distinguish between the two RL models. $\mathrm{LS}^n(a|k)$ are the Link Size attributes (Fosgerau et al., 2013) computed based on an RL model with the utility function $v(a|k) = -3.6\cdot\mathrm{aTT}(a|k) - 1.0\cdot\mathrm{LC}(a|k)$. We do not include any discount factor in the DE-RL model.

In Table 2 we report the estimation results for both RL models. For the STD-RL, we note that the parameter ratios are stable across different ρ values and the interpretation are hence the same. Intuitively, a unit travel time is penalized more on shorter

Table 2
Estimation results for the STD-RL and RE-DL models.

	STD-RL Value of discount factor ρ				DE-RL
	0.98	0.95	0.90	0.85	
$\hat{\beta}^1_{TT}$	−3.20	−3.19	−3.14	−3.11	–
Rob. Std. Err.	0.28	0.27	0.28	0.28	–
Rob. t-test(0)	−11.43	−11.91	−11.26	−11.11	–
$\hat{\beta}^2_{TT}$	−1.16	−1.14	−1.11	−1.10	–
Rob. Std. Err.	0.22	0.22	0.21	0.21	–
Rob. t-test(0)	−5.27	−5.15	−5.33	−5.24	–
$\hat{\beta}_{mTT}$	−4.21	−4.18	−3.98	−3.98	–
Rob. Std. Err.	0.33	0.32	0.36	0.35	–
Rob. t-test(0)	−12.93	−11.00	−11.37	−11.37	–
$\hat{\beta}_{aTT}$	–	–	–	–	−1.65
Rob. Std. Err.	–	–	–	–	0.67
Rob. t-test(0)	–	–	–	–	−2.48
$\hat{\beta}_{LC}$	−1.03	−1.02	−0.98	−0.98	−3.89
Rob. Std. Err.	0.04	0.04	0.05	0.05	0.28
Rob. t-test(0)	−25.75	−23.01	−20.95	−19.60	−14.1
$\hat{\beta}_{PS}$	−0.12	−0.12	−0.12	−0.12	–
Rob. Std. Err.	0.02	0.02	0.02	0.02	–
Rob. t-test(0)	−6.00	−5.58	−5.63	−6.00	–
$\hat{\beta}_{LS}$	–	–	–	–	−1.07
Rob. Std. Err.	–	–	–	–	0.42
Rob. t-test(0)	–	–	–	–	−25.5
Log-likelihood	−3385.01	−3354.68	−3329.42	−3315.14	−3006.17

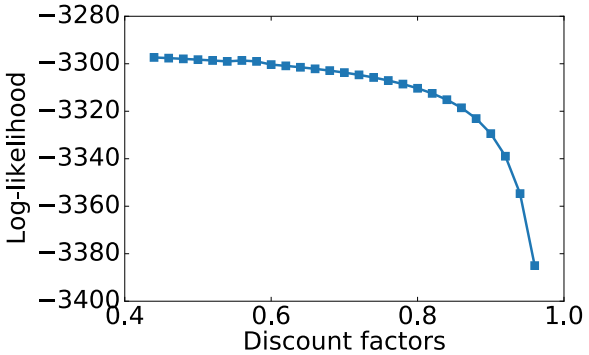

Fig. 6. Final log-likelihood values with discount factor from 0.50 to 0.98.

trips than on longer ones ($\hat{\beta}_{mTT}$ compared to $\hat{\beta}^1_{TT}$ and $\hat{\beta}^2_{TT}$). Moreover, a unit travel time on a high-variance link is over two times more penalized than that on a low-variance link (comparing $\hat{\beta}^1_{TT}$ and $\hat{\beta}^2_{TT}$). We note that the Policy Size parameter has its expected negative sign, penalizing overlapping among routing policies. For the DE-RL model, all the parameter estimates also have their expected signs. We note that the final log-likelihood values from the two models cannot be directly compared, as the choice set of routing policies in the STD-RL model and the choice set of physical routes in the DE-RL model are different.

We now turn our attention to the values of the discount factors. Unlike Oyama and Hato (2017), who analyze the degree of myopic route choice behavior by estimating discount factors in deterministic networks, we cannot infer the values from data. Indeed, this leads to a highly nonlinear constrained and costly optimization problem exhibiting numerical issues. Instead we fix the values of the discount factor to analyze the resulting parameter estimates and the numerical behavior of the model. Fig. 6 shows the final log-likelihood for varying discount factor values. We note that, as expected, there are numerical issues fixing the discount factor to 1 or below 0.44. In the former case, there is no proof that the fixed point solution exists, which we observe numerically. In the latter case, the close to myopic behavior implied by the factor results in numerical issues, possibly because the observations do not align with that assumption.

As a further step in analyzing the difference across the values of the discount factor, we investigate numerically the number of cycles that occur in the physical network. For this purpose we estimate six models using $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.98\}$, and simulate 200 paths for each observed OD pair with the resulting models. In total, we simulate 100,000 paths and Table 3 reports the number of simulated paths without and with cycles. We note that approximately 98% of the simulated paths are cycle-free, for

Table 3
Number of paths with cycles over 100,000 simulated paths.

Discount factor	Number of paths with cycles						
	0	1	2	3	4	5	6
0.50	98,020	1,937	29	8	2	1	3
0.60	97,968	1,989	23	14	2	2	2
0.70	97,990	1,964	33	8	2	1	2
0.80	98,026	1,948	23	2	1	0	0
0.90	98,123	1,862	14	1	0	0	0
0.98	97,896	2,082	21	1	0	0	0

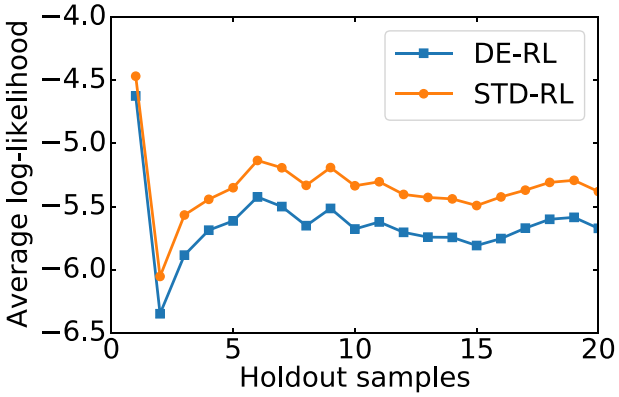

Fig. 7. Average log-likelihood values over holdout samples.

all values of ρ. For $\rho = 0.50, 0.60, 0.70$, there are a few paths with 4, 5 and 6 cycles, while for $\rho \geq 0.9$ there is no path having more than 3 cycles. Nevertheless, the results indicate that in this network a model with low discount factor does not have the issue of generating a significant share of paths with cycles. We note that if a given application requires cycle free path predictions, the path distribution can be truncated to exclude paths with cycles. Indeed, cycles can be easily detected. Since the share is small, this does not significantly impact the distribution.

Before comparing the out-of-sample prediction performance of the STD-RL and the DE-RL models, we compare ratios of the parameter estimates. For this analysis we focus on the STD-RL model with a discount factor 0.95. For the DE-RL model, $\hat{\beta}_{aTT}/\hat{\beta}_{LC} = 0.42$. As we do not have the corresponding travel time attribute in the STD-RL model, we compare this value to three different ratios: $\hat{\beta}^1_{TT}/\hat{\beta}_{LC} = 3.12$, $\hat{\beta}^2_{TT}/\hat{\beta}_{LC} = 1.12$ and $\hat{\beta}_{mTT}/\hat{\beta}_{LC} = 4.10$. Similar to the results on synthetic data case, we see that different representations of travel time (static-deterministic versus stochastic) lead to substantially different parameter ratios, and hence different behavioral interpretations.

We further evaluate the out-of-sample prediction performance of the STD-RL model (again we use a discount factor of 0.95) and the DE-RL model. We use a cross-validation approach where the sample of observations is repeatedly (20 iterations in our case) divided into two sets by drawing observations at random with a fixed probability: 80% of the observations are used for estimation and 20% are used as holdout to compute predicted probabilities by applying the estimated model.

For each holdout sample i, we compute the average log-likelihood values of the holdout samples as $LL_i = 1/|HD_i| \sum_{\sigma_i \in HD_i} \ln P(\sigma_i|\hat{\beta}_i)$, where HD_i is the set of observations in holdout sample i and $\hat{\beta}_i$ are the parameter estimate from the corresponding training set. To have unconditional values, we compute the averages of LL_i over samples as $\overline{LL}_p = 1/p \sum_{i=1}^{p} LL_i$, for $p = 1, \ldots, 20$. These average log-likelihood values are plotted in Fig. 7, which clearly shows that the STD-RL model in the STD network outperforms the static-deterministic counterpart in terms of prediction performance.

4.2.4. Computing time

In this section, we briefly discuss the computing time required to estimate the STD-RL model. Our code is implemented in MATLAB 2015 and we use an Intel(R) 3.20 GHz machine with a x64-based processor. It runs with Window 8 64-bit operating system.

It requires 30 min to 2 h to compute a log-likelihood value, if the code is not parallelized. The optimization algorithm (i.e., the outer one) needs around 30 to 40 iterations to converge. Therefore, it takes about 15 to 30 h to estimate the model with a non-parallelized code. A 20 to 30 times speedup is achieved thanks to the decomposition method presented in Algorithms 1 and 2. Decomposing the matrix operations into a series of simpler ones hence leads to an important gain. Further speedup can be obtained by parallelizing the code. The exact speed up depends on a number of factors, such as the characteristics of the data set and the number of cores. As an example, we reduce the estimation time by a factor of seven when using a machine with eight CPUs.

Value iteration (Algorithm 1) needs 100 to 150 iterations. The computing time required to solve the value functions as well as their gradients varies between two and 100 s for each destination. In contrast, the DE-RL model for static and deterministic networks (Fig. 7) requires only approximately 0.04 s for any destination. This large difference in computing times is due to the size of the graph and the non-linearity of the value functions in the STD case. Importantly, the computational effort associated with simulating path predictions or computing link flows using these value functions is the same for DE-RL and STD-RL.

Notwithstanding the substantial additional computational effort associated with the STD-RL model, the bulk of it can be done offline. In this case the computational budget is typically less restricted (e.g., the budget for estimating models) than the online one (that is, when the predictions are used by an other algorithm, e.g., computing a traffic equilibrium).

5. Concluding remarks

We proposed STD-RL, a RL model for routing policy choice in an STD network, so that the choice of the next link out of a given node is conditional on the time of day and information on realized link travel times, reflecting the ever increasing role of real-time information on travelers' route choices. The model is a significant expansion from its counterparts in a deterministic – static or time-dependent – networks (e.g., Fosgerau et al., 2013; de Moraes Ramos et al., 2020). Conceptually, it incorporates the important behavior of traveler adaptation to real-time traffic conditions. Computationally, the resulting system of equations for value functions is challenging to solve due to its nonlinear nature and large size. We proposed a decomposition algorithm that reduced computing time by writing costly matrix operations as a series of simple ones.

We presented two sets of numerical results based on synthetic and revealed preference data, respectively. The results on synthetic data showed that the less complex deterministic RL model is not a good approximation if the network is highly stochastic and travelers behave according to a STD-RL model. In this case, the parameter estimates can have a strong bias. The results based on a subnetwork of Stockholm (Sweden) showed that the STD-RL model produced realistic parameter estimates, and that we achieved a significant computing time speedup due to decomposition and parallelization. The STD-RL model produced better out-of-sample fit than the DE-RL model in a corresponding deterministic, static network, suggesting a gain of prediction performance by capturing the stochasticity of travel times and travelers' adaptive behavior. The parameter ratios were substantially different comparing the DE-RL and STD-RL models. Thus, it is important to use the STD-RL model, if the purpose of a study is to interpret travel behavior (analyze parameter estimates) in a network with high travel time variability.

While the results showed good accuracy and that the model for stochastic networks is computationally tractable, the associated computing times are much higher than their deterministic counterpart. For large-scale applications, it is necessary to strike a balance between the realism of the representation of time and information and the computational efficiency of the estimation algorithm. We see three avenues for future research. First, improving the definition of the joint distribution of time-dependent link travel time variables. It should reflect conditions in real data while keeping the dimension as low as possible. A possible direction is to use a relatively small number of features to represent the realized traffic conditions, for example, travel times at strategic locations in the network instead of on all links. Second, further speed-up could be achieved by using heuristics to compute the value functions. In this context it can be worth exploring methodologies from approximate dynamic programming and reinforcement learning to solve the value functions approximately (e.g., Powell, 2011; Bertsekas, 2019). Third, future research can also be dedicated to avoiding a stochastic travel time representation. That is, improving the deterministic approximation of the stochastic travel times. This could potentially be done through the graph representation and the utility specifications. In this case, the model proposed in this paper constitutes an important baseline.

As highlighted in Zimmermann and Frejinger (2020) and Mai and Jaillet (2020), recursive route choice models and logit-based dynamic discrete choice modeling are linked to the literature on maximum causal entropy inverse reinforcement learning and regularized Markov Decision Processes (Ng et al., 2000; Ziebart et al., 2010; Geist et al., 2019). Future work will focus on exploring learning algorithms designed to speed up the training of inverse reinforcement algorithms as, e.g., Ermon et al. (2015).

CRediT authorship contribution statement

Tien Mai: Methodology, Software, Writing - original draft, Writing - review & editing. **Xinlian Yu:** Resources, Writing - original draft, Writing - review & editing. **Song Gao:** Conceptualization, Methodology, Writing - review & editing. **Emma Frejinger:** Conceptualization, Methodology, Writing - review & editing.

Acknowledgments

We thank three anonymous referees whose comments substantially helped us improve the paper. The second author gratelfully acknowledges the financial support from Chinese Scholarship Council. The fourth author acknowledges the funding through the Canada Research Chair on Demand forecasting and Optimization of Transport Systems.

Appendix

Proof of Proposition 1. We first derive the analytical formula of $\partial Z_s/\partial\beta$, $s \in S$, with respect to the model parameters β. From (12) we can write

$$Z_s = b_s + \sum_{a\in A(n(s))} e^{\frac{1}{\mu}v(a|s)} \prod_{\substack{s'\in S\\ n(s')=a}} Z_{s'}^{\rho W(s'|s)},\ \forall s \in S. \tag{19}$$

Let define a matrix $K(|S|\times|\mathcal{N}|)$ with entries

$$K_{sa} = e^{\frac{1}{\mu}v(a|s)} \prod_{\substack{s'\in S\\ n(s')=a}} Z_{s'}^{\rho W(s'|s)},\ \forall s \in S, a \in \mathcal{N}.$$

We have

$$\ln K_{sa} = \frac{1}{\mu}v(a|s) + \rho \sum_{\substack{s'\in S\\ n(s')=a}} W(s'|s)\ln Z_{s'},\ \forall s \in S, a \in \mathcal{N}. \tag{20}$$

Taking the first derivative of (20) with respect to a parameter β_j we obtain

$$\frac{\partial K_{sa}}{\partial\beta_j} = K_{sa}\left(\frac{1}{\mu}\frac{\partial v(a|s)}{\partial\beta_j} + \sum_{\substack{s'\in S\\ n(s')=a}} \frac{\rho W(s'|s)}{Z_{s'}}\frac{\partial Z_{s'}}{\partial\beta_j}\right). \tag{21}$$

Note that

$$Z_s = b_s + \sum_{\substack{a\in A(k)\\ k=t(s)}} K_{sa},\ \forall s \in S. \tag{22}$$

So if we take the derivative of (22) and substitute to (21), then we have

$$\begin{aligned}\frac{\partial Z_s}{\partial\beta_j} &= \sum_{\substack{a\in A(k)\\ k=t(s)}} K_{sa}\left(\frac{1}{\mu}\frac{\partial v(a|s)}{\partial\beta_j} + \sum_{\substack{s'\in S\\ n(s')=a}} \frac{\rho W(s'|s)}{Z_{s'}}\frac{\partial Z_{s'}}{\partial\beta_j}\right)\\ &= \sum_{\substack{a\in A(k)\\ k=t(s)}} \frac{1}{\mu}K_{sa}\frac{\partial v(a|s)}{\partial\beta_j} + \sum_{s'\in S}\left(\frac{\rho K_{sa}W(s'|s)}{Z_{s'}}\right)\frac{\partial Z_{s'}}{\partial\beta_j}.\end{aligned} \tag{23}$$

So, if we define two matrices $J(|S|\times|\mathcal{N}|)$, $D(|S|\times|\mathcal{N}|)$ as in (14) and (15), then (23) can be written in a matrix form as

$$\frac{\partial Z}{\partial\beta_j} = Je + D\frac{\partial Z}{\partial\beta_j}, \tag{24}$$

which is the desired system of linear equations. ■

We introduce three lemmas to support the proof of Proposition 2.

Lemma 3. *Given a matrix X of size $|S|\times|S|$, and a matrix $U(|S|\times|\mathcal{N}|)$ with entries $U_{sa} = \sum_{\substack{s'\in S\\ n(s')=a}} X_{ss'}$, $\forall s \in S, a \in \mathcal{N}$, then $U = X\times T$.*

Proof. Indeed, each element (s, a) of matrix $X \times T$ can be computed as

$$(X\times T)_{sa} = \sum_{w\in S} X_{sw}T_{wa}.$$

Moreover, according the definition of T we have, given $w \in S$, $T_{wa} = 1$ if and only if $n(w) = a$, so, we have the following result

$$(X\times T)_{sa} = \sum_{w\in S, n(w)=a} X_{sw} = U_{sa}.$$ ■

Lemma 4. *Given matrices X, Y of size $|S|\times|S|$, and a matrix $U(|S|\times|\mathcal{N}|)$ with entries $X_{ss'} = U_{sa}Y_{ss'}$, $\forall s, s' \in S$, $a = n(s')$, then $X = (U\times T^T)\circ Y$, where ^{T} is the transpose operator.*

Proof. Each element ss' of $(U\times T^{\mathrm{T}})\circ Y$ can be computed as

$$[(U\times T^{\mathrm{T}})\circ Y]_{ss'} = Y_{ss'}\sum_{a\in\mathcal{N}} U_{sa}T^{\mathrm{T}}_{as'} = Y_{ss'}U_{sa}T_{s'a} = Y_{ss'}U_{sa},$$

where $a = n(s')$. ■

Lemma 5. *Given matrices X, Y of size $|S| \times |S|$, and a vector $b(|S|)$ with entries $X_{ss'} = Y_{ss'} b_{s'}$, $\forall s, s' \in S$, then $X = Y \times diag(b)$.*

Proof. Each element ss' of $Y \times \text{diag}(b)$ can be computed as

$$[Y \times \text{diag}(b)]_{ss'} = \sum_{w \in S} Y_{sw} \text{diag}(b)_{ws'} = Y_{ss'} \text{diag}(b)_{s's'} = Y_{ss'} b_{s'}.$$

We obtain the result. ■

Proof of Proposition 2. First, we validate the formulation for $\ln(U)$. Each element of this matrix can be written as

$$\ln U_{sa} = \sum_{\substack{s' \in S \\ n(s')=a}} \rho W(s'|s) \ln Z_{s'}, \ \forall s \in S, a \in \mathcal{N}. \tag{25}$$

According to Lemmas 3 and 5, the matrix $\ln(U)$ can be computed as

$$\ln(U) = \left[F \times \text{diag}\left(\ln(Z)\right)\right] \times T.$$

Now we turn our attention to the computation of matrices J^j and D, defined in (14) and (15), which is necessary for the gradients of the value function as well as the log-likelihood function. We note that, $K = M \circ U$, so J^j can be written as

$$J^j = K \circ H^j = H^j \circ (M \circ U),$$

where H^j is a matrix of size $|S| \times |\mathcal{N}|$ with entries $\partial v(a|s)/\partial \beta_j$. Moreover, according to (15), matrix D can be computed based on matrix K (of size $|S| \times |\mathcal{N}|$) and matrices F (of size $|S| \times |S|$) and vector $Z(|S|)$. So, using Lemmas 4 and 5, D can be obtained through K, F, Z and T as follows

$$D = \left[\left(K \times T^{\mathrm{T}}\right) \circ F\right] \times \text{diag}(1/Z),$$

which is the desired equation. ■

References

Baillon, J.-B., Cominetti, R., 2008. Markovian traffic equilibrium. Math. Program. 111 (1–2), 33–56.

Ben-Akiva, M., Ramming, S., 1998. Lecture notes: Discrete choice models of traveler behavior in networks. In: Prepared for Advanced Methods for Planning and Management of Transportation Networks. Capri, Italy.

Bertsekas, D.P., 2019. Reinforcement Learning and Optimal Control. Athena Scientific.

de Moraes Ramos, G., Mai, T., Daamen, W., Frejinger, E., Hoogendoorn, S., 2020. Route choice behaviour and travel information in a congested network: Static and dynamic recursive models. Transp. Res. C 114, 681–693.

Ding-Mastera, J., Gao, S., Jenelius, E., Rahmani, M., Ben-Akiva, M.E., 2019. A latent-class adaptive routing choice model in stochastic time-dependent networks. Transp. Res. B 124, 1–17.

Ermon, S., Xue, Y., Toth, R., Dilkina, B., Bernstein, R., Damoulas, T., Clark, P., DeGloria, S., Mude, A., Barrett, C., Gomes, C.P., 2015. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in East Africa. In: Twenty-Ninth AAAI Conference on Artificial Intelligence.

Fosgerau, M., Frejinger, E., Karlstrom, A., 2013. A link based network route choice model with unrestricted choice set. Transp. Res. B 56, 70–80.

Frejinger, E., Bierlaire, M., Ben-Akiva, M., 2009. Sampling of alternatives for route choice modeling. Transp. Res. B 43 (10), 984–994.

Gao, S., Chabini, I., 2006. Optimal routing policy problems in stochastic time-dependent networks. Transp. Res. B 40 (2), 93–122.

Gao, S., Frejinger, E., Ben-Akiva, M., 2008. Adaptive route choice models in stochastic time dependent networks. Transp. Res. Rec. 2085, 136–143.

Geist, M., Scherrer, B., Pietquin, O., 2019. A theory of regularized markov decision processes. arXiv preprint arXiv:1901.11275.

Guevara, C.A., Ben-Akiva, M.E., 2013. Sampling of alternatives in multivariate extreme value (MEV) models. Transp. Res. B 48 (1), 31–52.

Jiang, G., Fosgerau, M., Lo, H.K., 2020. Route choice, travel time variability, and rational inattention. Transp. Res. B 132, 188–207.

Lai, X., Bierlaire, M., 2015. Specification of the cross nested logit model with sampling of alternatives for route choice models. Transp. Res. B 80, 220–234.

Littman, J., 2019. Waze hijacked L.A. in the name of convenience. Can anyone put the genie back in the bottle?. Los Angeles Mag..

Mai, T., 2016. A method of integrating correlation structures for a generalized recursive route choice model. Transp. Res. B 93 (1), 146–161a.

Mai, T., 2016b. Dynamic Programming Approaches for Estimating and Applying Large-Scale Discrete Choice Models (Ph.D. thesis). Université de Montréal, Montreal, Canada.

Mai, T., Bastin, F., Frejinger, E., 2018. A decomposition method for estimating recursive logit based route choice models. EURO J. Transp. Logist. 7, 253–275.

Mai, T., Fosgerau, M., Frejinger, E., 2015. A nested recursive logit model for route choice analysis. Transp. Res. B 75, 100–112.

Mai, T., Jaillet, P., 2020. A relation analysis of Markov decision process frameworks. arXiv preprint arXiv:2008.07820.

Moghaddam, Z.R., Jeihani, M., Peeta, S., Banerjee, S., 2019. Comprehending the roles of traveler perception of travel time reliability on route choice behavior. Travel Behav. Soc. 16, 13–22.

Nassir, N., Hickman, M., Ma, Z.-L., 2019. A strategy-based recursive path choice model for public transit smart card data. Transp. Res. B 126, 528–548.

Nassir, N., Ziebarth, J., Sall, E., Zorn, L., 2014. Choice set generation algorithm suitable for measuring route choice accessibility. Transp. Res. Rec.: J. Transp. Res. Board 2430, 170–181.

Ng, A.Y., Russell, S.J., et al., 2000. Algorithms for inverse reinforcement learning. In: ICML.

Oyama, Y., Hato, E., 2017. A discounted recursive logit model for dynamic gridlock network analysis. Transp. Res. C 85, 509–527.

Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality, second ed. Wiley.

Prato, C.G., 2009. Route choice modeling: Past, present and future research directions. J. Choice Model. 2, 65–100.

Rahmani, M., Jenelius, E., Koutsopoulos, H.N., 2015. Non-parametric estimation of route travel time distributions from low-frequency floating car data. Transp. Res. C 58, 343–362.

Rahmani, M., Koutsopoulos, H.N., 2013. Path inference from sparse floating car data for urban networks. Transp. Res. C 30, 41–54.

Razo, M., Gao, S., 2013. A rank-dependent expected utility model for strategic route choice with stated preference data. Transp. Res. C 27, 117–130.

Rust, J., 1987. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. Econometrica 55 (5), 999–1033.

Yu, X., Mai, T., Ding-Mastera, J., Gao, S., Frejinger, E., 2020. Modeling route choice with real-time information: Comparing the recursive and non-recursive models. arXiv e-prints, art. arXiv:2005.05783.
Zhang, C., Liu, T.-L., Huang, H.-J., Chen, J., 2018. A cumulative prospect theory approach to commuters' day-to-day route-choice modeling with friends' travel information. Transp. Res. C 86, 527–548.
Zhang, G., Wei, F., Ning, J., Ma, S., Yi, W., 2019. Information adoption in commuters' route choice in the context of social interactions. Transp. Res. A 130, 300–316.
Ziebart, B.D., Bagnell, J.A., Dey, A.K., 2010. Modeling interaction via the principle of maximum causal entropy. In: Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML'10), pp. 1255–1262.
Zimmermann, M., Frejinger, E., 2020. A tutorial on recursive models for analyzing and predicting path choice behavior. EURO J. Transp. Logist. 9 (2), 100004.
Zimmermann, M., Mai, T., Frejinger, E., 2017. Bike route choice modeling using GPS data without choice sets of paths. Transp. Res. C 75, 183–196.