Automatica 140 (2022) 110224

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Variable demand and multi-commodity flow in Markovian network equilibrium[☆]

Yue Yu[a,*,1], Dan Calderone[c], Sarah H.Q. Li[b], Lillian J. Ratliff[c], Behçet Açıkmeşe[b]

[a] *Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, 78712, United States of America*
[b] *William E. Boeing Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, 98195, United States of America*
[c] *Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, 98195, United States of America*

ARTICLE INFO

Article history:
Received 15 July 2020
Received in revised form 26 October 2021
Accepted 7 February 2022
Available online 18 March 2022

Keywords:
Wardrop equilibrium
Markov decision process
Network optimization

ABSTRACT

Markovian network equilibrium generalizes the classical Wardrop equilibrium in network games. At a Markovian network equilibrium, each player of the game solves a Markov decision process instead of a shortest path problem. We propose two novel extensions of Markovian network equilibrium by considering (1) variable demand, which offers the players a quitting option, and (2) multi-commodity flow, which allows players to have heterogeneous ending time. We further develop dynamic-programming-based iterative algorithms for the proposed equilibrium problems, together with their arithmetic complexity analysis. Finally, we illustrate our network equilibrium model via a multi-commodity ride-sharing example, and compare the computational efficiency of our algorithms against the state-of-the-art optimization software MOSEK over extensive numerical experiments.

1. Introduction

Network equilibrium problems arise in a variety of applications, such as resource allocation and routing in communication or transportation networks (Bertsekas, 1998; Bürger, Zelazo, & Allgöwer, 2014; Rockafellar, 1984; Xiao, Johansson, & Boyd, 2004). Among the most well-studied examples is the Wardrop equilibrium model in routing games (Beckmann, McGuire, & Winsten, 1956; Correa & Stier-Moses, 2010; Gartner, 1980a, 1980b; Patriksson, 1994). In this model, users in a transportation network are assumed to choose routes with the cost that they perceive as the lowest, *i.e.*, each user solves a shortest path problem, under the prevailing traffic conditions (Correa & Stier-Moses, 2010). With this assumption, the resulting equilibra are characterized by the Wardrop equilibrium principle: the cost of all the routes actually used is equal, and less than those which would be experienced by a single user on any unused route (Wardrop & Whitehead, 1952).

To ensure their practical relevance, it is often necessary to incorporate stochasticity into network equilibrium problems. For example, the stochastic user equilibrium (SUE) model (Fisk, 1980; Liu, He, & He, 2009; Sheffi & Powell, 1982) considers independent stochastic error on the route cost perceived by the users, leading to a user distribution based on the logit (Dial, 1971) or probit model (Daganzo & Sheffi, 1977); see Patriksson (1994, Sec. 2.8.1) and Cominetti, Facchinei, and Lasserre (2012) for a detailed discussion. Unfortunately, the SUE model presents several drawbacks: it requires computationally expensive route enumeration, and is not suited for problems with overlapping routes (Baillon & Cominetti, 2008).

To address these drawbacks, different network models consider different types of stochasticity. In particular, Ahipaşaoğlu, Arıkan, and Natarajan (2019), Baillon and Cominetti (2008) introduced a Markovian network equilibrium model where users are assumed to choose, instead of routes, sequences of actions with the accumulated cost that they perceive as the lowest. Each action is accompanied by a deterministic outcome and a stochastic cost. For example, each vehicle in a transportation network is assumed to choose a sequence of arcs, where each arc leads to a deterministic transition to the next node in the network and a stochastic amount of travel time (Baillon & Cominetti, 2008). On the other hand, Calderone and Sastry (2017a, 2017b) proposed a different stochastic network equilibrium model. Unlike the one in Baillon and Cominetti (2008), each action is accompanied by a stochastic outcome and a deterministic cost. For example, an aircraft flying in stormy weather is assumed to choose a sequence of waypoints

[☆] This work is supported by National Science Foundation, United States of America Award 1736582. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Rong Su under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.
E-mail addresses: yueyu@utexas.edu, yueyu@uw.edu (Y. Yu), djcal@uw.edu (D. Calderone), sarahli@uw.edu (S.H.Q. Li), ratliffl@uw.edu (L.J. Ratliff), behcet@uw.edu (B. Açıkmeşe).

[1] This work was completed while Yue Yu was a graduate student at University of Washington.

https://doi.org/10.1016/j.automatica.2022.110224
0005-1098/

to fly towards, where each choice costs a deterministic amount of fuel usage and is accompanied by a stochastic change in the weather condition (Nilim & El Ghaoui, 2005). As a result, instead of a shortest path problem, each user solves a Markov decision process (MDP) (Bertsekas, 1996; Puterman, 1994), where the cost of different actions is determined by the prevailing choices of all users. This model has found a variety of applications in modern transportation including ridesharing and parking (Calderone, 2017; Li, Yu, Calderone, Ratliff, & Açrkmeşe, 2019).

Although the results in Calderone and Sastry (2017a, 2017b) serve as a first step toward a general class of stochastic dynamic network equilibra, it has the following limitations: (a) it does not incorporate many important features of Wardrop equilibrium, such as variable demand and multi-commodity flow and (b) its solution method relies exclusively on off-the-shelf optimization software, which does not fully exploit the problem structure. We address these limitations by making the following contributions.

(1) We develop novel extensions to the Markovian network equilibrium model by considering (a) variable demand, which offers the users a quitting option, and (b) multi-commodity flow, which allows the users to have heterogeneous ending time.
(2) We design novel dynamic-programming-based algorithms for the Markovian network equilibrium problems, together with arithmetic complexity analysis. Our algorithms outperform state-of-the-art optimization software MOSEK in extensive numerical experiments.

The rest of the paper is organized as follows. We first review some background on MDP in Section 2, then present our variable demand and multi-commodity flow equilibrium models in Section 3. Section 4 focuses on developing efficient iterative algorithms for our equilibrium problems.[2] In Section 5, we first illustrate the equilibrium models in Section 3 via a multi-commodity ride-sharing example, then compare the algorithms in Section 4 against commercial software Mosek. Finally, we conclude with discussions and comments on the future directions in Section 6.

Throughout the paper we will use the following notation: $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_+$ denotes the set of nonnegative real numbers, and $\mathbb{N}$ denotes the set of positive integers; $[N]$ denotes the set $\{1, 2, \ldots, N\}$ for integer N; a_{ijk} denotes the ijkth component of a three-dimensional tensor $a \in \mathbb{R}^{n_1\times n_2\times n_3}$, and analogously, a_{ij} denotes the ijth entry of for a two-dimensional tensor (matrix) $a \in \mathbb{R}^{n_1\times n_2}$. Given $b_1, \ldots, b_N \in \mathbb{R}$, we say $(b^\star, i^\star) = \min_{i\in[N]} b_i$ if $b^\star = \min_{i\in[N]} b_i$ and $i^\star \in \operatorname{argmin}_{i\in[N]} b_i$.

2. Preliminaries and background

A T-horizon Markov decision process (MDP) is defined by a set of states $[S]$, a set of actions $[A]$, a cost tensor $c \in \mathbb{R}^{T\times S\times A}$, and a transition probability tensor $P \in [0, 1]^{S\times A\times S}$, where $T, S, A \in \mathbb{N}$ denote the number of time steps, states, and actions, respectively. Further, $c_{tsa} \in \mathbb{R}$ denotes the cost of choosing action $a \in [A]$ in state $s \in [S]$ at time $t \in [T]$, and $P_{sas'} \in [0, 1]$ denotes the probability of transition from state $s \in [S]$ to $s' \in [S]$ when choosing action $a \in [A]$. To find the optimal sequence of action that minimizes the expected accumulated cost, one can solve either one of the two following linear programs:

$$\begin{aligned}\min_{y}\ & \sum_{t,s,a} c_{tsa}y_{tsa}\\ \text{s.t.}\ & \sum_{a} y_{1sa} = p_{1s},\\ & \sum_{a} y_{t+1,sa} = p_{t+1,s} + \sum_{s',a} P_{s'as}y_{ts'a},\quad \forall t \in [T-1],\\ & 0 \le y_{tsa},\quad \forall t \in [T], s \in [S], a \in [A].\end{aligned} \tag{1}$$

$$\begin{aligned}\max_{v}\ & \sum_{t,s} p_{ts}v_{ts}\\ \text{s.t.}\ & v_{Ts} \le c_{Tsa},\\ & v_{ts} \le c_{tsa} + \sum_{s'} P_{sas'}v_{t+1,s'},\quad \forall t \in [T-1], s \in [S], a \in [A].\end{aligned} \tag{2}$$

Here $p \in \mathbb{R}_+^{T\times S}$ is such that $p_{1s} > 0$ for some $s \in [S]$. If $\sum_{s\in[S]} p_{1s} = 1$ and $p_{ts} = 0$ for all $1 \le t \le T$ and $s \in [S]$, then p_{1s} represents the probability of starting the MDP in state s. Variable y_{tsa} in optimization (1) represents the probability of choosing action a in state s at time t, and variable v_{ts} in optimization (2) represents the expected accumulated cost between time t and time T starting from state s.

One of the most popular numerical methods for linear program (1) and (2) is *dynamic programming*, given by the following Algorithm 1 and Algorithm 2.

Algorithm 1 Backward induction

Require: P, c, T.
Ensure: v, π.
1: Let $(v_{Ts}, \pi_{Ts}) = \min_{a\in[A]} c_{Tsa},\quad \forall s \in [S]$.
2: **for** $t = T-1, T-2, \ldots, 1$ **do**
3: $\quad (v_{ts}, \pi_{ts}) = \min_{a\in[A]} c_{tsa} + \sum_{s'} P_{sas'}v_{t+1,s'}, \forall s \in [S]$
4: **end for**

Algorithm 2 Forward induction

Require: π, p, P, T.
Ensure: y.
1: Initialize $y = 0$, let $y_{1s\pi_{1s}} \leftarrow p_{1s}, \forall s \in [S]$.
2: **for** $t = 1, 2, \ldots, T-1$ **do**
3: $\quad y_{t+1,s\pi_{t+1,s}} \leftarrow p_{t+1,s} + \sum_{j} P_{j\pi_{tj}s}y_{tj\pi_{tj}}, \forall s \in [S]$
4: **end for**

Let (v, π) be the output of Algorithm 1 with input (P, c, T), and y be the output of Algorithm 2 with input (π, p, P, T), then one can verify that such a solution pair (y, v) directly satisfies the Karush–Kuhn–Tucker (KKT) conditions (Ben-Tal & Nemirovski, 2001, Thm. 1.3.3) of linear program (1) and (2). If we define the *sparsity level* of an MDP as follows

$$\sigma = \max\{N_1, N_2\}/S, \tag{3}$$

where $N_1 = \max_{s,a} \left|\{s'|P_{sas'} > 0\}\right|$ and $N_2 = \max_{s',a} \left|\{s|P_{sas'} > 0\}\right|$, then σS measures the maximum number of states connected by the transition kernel P. Further, it is straightforward to check that Algorithm 1 costs $O(\sigma TS^2A)$ arithmetic operations, and Algorithm 2 costs $O(\sigma TS^2)$ arithmetic operations.

3. Markovian network equilibrium

MDP routing games combine the idea of MDP together with classical routing games (Calderone & Sastry, 2017b). In an MDP routing game, a fixed amount of players with the same planning

[2] Due to the limit of space, we omit some of the proof details in Section 4, and include them in Yu, Calderone, Li, Ratliff, and Açıkmeşe (2021).

horizon choose sequences of actions that they perceive as achieving the lowest expected accumulated cost under the prevailing choices of other players. The equilibra of an MDP routing game, termed *Markovian network equilibra*, are similar to the Wardrop equilibra in a routing game where a fixed amount of players with the same destination choose routes that they perceive as the shortest under the prevailing choices of other players.

We introduce two extensions to MDP routing games that allow the amount of players to vary and the planning horizon to differ. We will also show that, under some mild assumptions, the corresponding Markovian network equilibra can be computed using convex optimization.

3.1. Variable demand

One limitation of the MDP routing games in Calderone and Sastry (2017b) is the assumption that the total amount of players is fixed. However, an important feature in network games is to allow the total amount of players to vary, or equivalently, to provide the players with a quitting action (Patriksson, 1994, Sec. 2.1.2). Aiming to address this limitation, we propose the following *MDP routing game with variable demand*.

Game 1. *At each time $t \in [T]$, p_{ts} new players start the game from state $s \in [S]$. Among these p_{ts} players, each one can choose to:*

(1) quit the game immediately at the cost of $\psi_{ts}(z_{ts})$,
(2) take action $a \in [A]$ at the cost of $\phi_{tsa}(y_{tsa})$ and reach state $s' \in [S]$ with probability $P_{sas'}$ at time $t+1$, then repeat such process till $t = T$, when the player ends the game after choosing the last action.

Here z_{ts} and y_{tsa} denote the total amount of players choosing to quit the game in state s at time t, and, respectively, taking action a in state s at time t.

Remark 1. Game 1 is a special case of mean field games over graphs (Gomes, Mohr, & Souza, 2009, 2010; Guéant, 2011, 2015; Tanaka, Nekouei, Pedram, & Johansson, 2020). The interactions among different players are mediated by a mean field, described by function ϕ_{tsa} and function ψ_{ts} for all $t \in [T], s \in [S], a \in [A]$.

Intuitively, one can interpret Game 1 as a competitive market model. The supply side corresponds to the stochastic environment, providing the option of playing or quitting the game. The demand side corresponds to the amount of players that decide to play the game, which changes with the expected accumulated cost of the playing option according to function ψ_{ts} for all $t \in [T]$ and $s \in [S]$.

Remark 2. If the quitting option is not available, then Game 1 reduces to an MDP routing game with fixed demand, introduced in Calderone and Sastry (2017b). On the other hand, if the transition in Game 1 is deterministic, *i.e.*, for each $s \in [S]$ and $a \in [A]$, there exists $s' \in [S]$ such that $P_{sas'} = 1$, then Game 1 reduces to a classical single-commodity routing game with a variable demand Patriksson (1994, Sec. 2.2.3). Particularly, each player solves an MDP with deterministic transitions, which is equivalent to a shortest path problem.

The Wardrop equilibrium principle is a key characterization of the equilibra of network games (Correa & Stier-Moses, 2010; Patriksson, 1994). The principle states that, at equilibra, only the strategies with the lowest cost are actually used. Does this principle apply to Game 1? As we show in the following, the answer is affirmative.

First, we make the following assumptions on Game 1.

Assumption 1. We assume that $p \in \mathbb{R}_+^{T\times S}$, $P \in [0,1]^{S\times A\times S}$ and $\sum_{s'} P_{sas'} = 1$ for all $s \in [S]$, $a \in [A]$. Further, the function $\phi_{tsa} : [0, \rho] \to \mathbb{R}$ and function $\psi_{ts} : [0, \rho] \to \mathbb{R}$ are continuous and strictly increasing over their respective domains, where $\rho = \sum_{t,s} p_{ts}$.

With these assumptions, we now introduce the following pair of primal–dual optimization problems associated with Game 1.

$$
\begin{aligned}
\min_{y,z}\ & \sum_{t,s,a}\int_0^{y_{tsa}} \phi_{tsa}(\alpha)d\alpha + \sum_{t,s}\int_0^{z_{ts}} \psi_{ts}(\alpha)d\alpha\\
\text{s.t.}\ & \sum_a y_{1sa} = p_{1s} - z_{1s},\\
& \sum_a y_{t+1,sa} = p_{t+1,s} - z_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a},\ \ \forall t \in [T-1],\\
& 0 \le y_{tsa},\ \ 0 \le z_{ts} \le p_{ts},\ \ \forall t \in [T], s \in [S], a \in [A].
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\max_{u,v,w,\lambda}\ & \sum_{t,s} p_{ts}(v_{ts} - \lambda_{ts}) - \sum_{t,s,a}\int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha)d\alpha\\
& - \sum_{t,s}\int_{\psi_{ts}(0)}^{w_{ts}} \psi_{ts}^{-1}(\alpha)d\alpha\\
\text{s.t.}\ & v_{Ts} \le u_{Tsa},\\
& v_{ts} \le u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'},\ \ \forall t \in [T-1],\\
& v_{ts} \le w_{ts} + \lambda_{ts},\ \ 0 \le \lambda_{ts},\ \ \forall t \in [T], s \in [S], a \in [A].
\end{aligned} \tag{5}
$$

In particular, the constraint $0 \le z_{ts} \le p_{ts}$ allows the number of players choosing to quit the game in state s at time t to vary within interval $[0, p_{ts}]$. If variable z_{ts} is zero and function ϕ_{tsa} is constant-valued for all $t \in [T], s \in [S], a \in [A]$, *i.e.*, the quitting option is removed and the cost of each action does not depend on y in Game 1, then one can verify that optimization (4) will reduce to (1) and optimization (5) will reduce to (2).

The following theorem shows that the solutions of the optimizations in (4) and (5) satisfy the Wardrop equilibrium principle in Game 1.

Theorem 1. *Suppose Assumption 1 holds, (y, z) solves (4), and (u, v, w, λ) solves (5), then for any $p_{ts} > 0$,*

$$
\begin{aligned}
&\text{if } z_{ts} = 0,\ \text{ then } v_{ts} \le \psi_{ts}(p_{ts}),\\
&\text{if } 0 < z_{ts} < p_{ts},\ \text{ then } v_{ts} = \psi_{ts}(z_{ts}),\\
&\text{if } z_{ts} = p_{ts},\ \text{ then } v_{ts} \ge \psi_{ts}(0).
\end{aligned} \tag{6}
$$

Further, if $y_{Tsa} > 0$ for some $s \in [S]$ and $a \in [A]$, then

$$
(v_{Ts}, a) = \min_{a' \in [A]} \phi_{Tsa'}(y_{Tsa'}). \tag{7}
$$

If $y_{tsa} > 0$ for some $t \in [T-1]$, $s \in [S]$ and $a \in [A]$, then

$$
(v_{ts}, a) = \min_{a' \in [A]} \phi_{tsa'}(y_{tsa'}) + \sum_{s'} P_{sa's'} v_{t+1,s'}. \tag{8}
$$

Proof. See Appendix A.1.

Theorem 1 shows that an equilibrium of Game 1 that satisfies the Wardrop equilibrium principle not only exists, but can be computed by solving optimization (4) and (5). In particular, if action a is chosen in state s at time t by any player at equilibrium, *i.e.*, $y_{tsa} > 0$, then action a must be optimal in the sense of Algorithm 1. On the other hand, Eq. (6) says that if some players choose the quitting option in state s at time t at equilibrium, *i.e.*, $z_{ts} > 0$, then the cost of playing is no more than quitting, *i.e.*, $v_{ts} \ge \psi_{ts}(p_{ts})$. Similarly, if some players choose to play, *i.e.*, $z_{ts} < p_{ts}$, then the cost of playing is no more than quitting, *i.e.*, $v_{ts} \le \psi_{ts}(p_{ts})$. Therefore, Theorem 1 indeed describes a Wardrop equilibrium where no individual player can benefit from unilaterally switching to alternative actions.

3.2. Multicommodity flow

Another limitation of the MDP routing games in Calderone and Sastry (2017b) is that all players are assumed to end their game at the same time, which is analogous to the single-commodity routing game where all vehicles have the same destination. Aiming to address this limitation, we propose the following multi-commodity MDP routing game, where players can have heterogeneous ending time, denoted by set $\mathbb{T}$. We assume, without loss of generality, that $\mathbb{T} \subset [T]$ and $T \in \mathbb{T}$.

Game 2. *At each time $t \in [T]$, p_{ts}^τ new players with a common ending time $\tau \in \mathbb{T}$ ($\tau \ge t$) start the game from state s. Each of these players can choose the action a at the cost of $\phi_{tsa}(\sum_{\tau,\tau\ge t} y_{tsa}^\tau)$ and reach state s' with probability $P_{sas'}$ at time $t+1$, then repeat this process till $t=\tau$, when the player ends the game after choosing the last action. Here y_{tsa}^τ denotes the total amount of players that end the game at time τ and choose action a in state s at time t.*

Remark 3. If $\mathbb{T} = \{T\}$, then Game 2 reduces to a MDP routing game introduced in Calderone and Sastry (2017b). On the other hand, if the transition in Game 2 is deterministic, *i.e.*, for each $s \in [S]$ and $a \in [A]$, there exists $s' \in [S]$ such that $P_{sas'} = 1$, then Game 2 reduces to the traditional multi-commodity routing game with a fixed demand Patriksson (1994, Sec. 2.1.1). Particularly, the state where a player starts and ends the game forms an origin–destination pair, which is jointly determined by the starting state and the deterministic transition.

Similar to Game 1, the equilibrium of Game 2 can also be computed by solving convex optimization problems, as we show in the following.

First, we make the following assumptions on Game 2.

Assumption 2. We assume $T \in \mathbb{T} \subseteq [T]$, $p_{ts}^\tau \in \mathbb{R}_+$ for all $\tau \in \mathbb{T}$, $t \ge \tau$ and $s \in [S]$; $P \in \mathbb{R}_+^{S\times A\times S}$ and $\sum_{s'} P_{sas'} = 1$ for all $s \in [S]$, $a \in [A]$. Further, the function $\phi_{tsa} : [0,\rho] \to \mathbb{R}$ is continuous and strictly increasing, where $\rho = \sum_\tau \sum_{t\le\tau,s} p_{ts}^\tau$.

With these assumptions, we now introduce the following pair of primal–dual optimization problems associated with Game 2.

$$\begin{aligned}
\min_{\{y^\tau\}_{\tau\in\mathbb{T}}} \quad & \sum_{t,s,a} \int_0^{\sum_{\tau,\tau\ge t} y_{tsa}^\tau} \phi_{tsa}(\alpha)d\alpha \\
\text{s.t.} \quad & \sum_a y_{1sa}^\tau = p_{1s}^\tau \\
& \sum_a y_{t+1,sa}^\tau = p_{t+1,s}^\tau + \sum_{s',a} P_{s'as} y_{ts'a}^\tau, \quad \forall t \in [\tau-1], \\
& 0 \le y_{tsa}^\tau, \quad \forall \tau \in \mathbb{T}, t \in [\tau], s \in [S], a \in [A]
\end{aligned} \tag{9}$$

$$\begin{aligned}
\max_{u,\{v^\tau\}_{\tau\in\mathbb{T}}} \quad & \sum_{t,s} \sum_{\tau,\tau\ge t} p_{ts}^\tau v_{ts}^\tau - \sum_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha)d\alpha \\
\text{s.t.} \quad & v_{\tau s}^\tau \le u_{\tau sa}, \\
& v_{ts}^\tau \le u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}^\tau, \quad \forall t \in [\tau-1], \\
& \forall \tau \in \mathbb{T}, s \in [S], a \in [A]
\end{aligned} \tag{10}$$

The following theorem shows that the solutions to optimization problems (9) and (10) satisfy the Wardrop equilibrium principle in Game 2.

Theorem 2. *Suppose Assumption 2 holds, y solves (9), and (u, v) solves (10). If $y_{\tau sa}^\tau > 0$ for some $\tau \in \mathbb{T}$, $s \in [S]$, $a \in [A]$, then*

$$(v_{\tau s}^\tau, a) = \min_{a'\in[A]} \phi_{\tau sa'}\Big(\sum_{\tau,\tau\ge t} y_{\tau sa'}^\tau\Big). \tag{11}$$

If $y_{tsa}^\tau > 0$ for some $\tau \in \mathbb{T}$, $t \in [\tau-1]$, $s \in [S]$, $a \in [A]$, then

$$(v_{ts}^\tau, a) = \min_{a'\in[A]} \phi_{tsa'}\Big(\sum_{\tau,\tau\ge t} y_{tsa'}^\tau\Big) + \sum_{s'} P_{sa's'} v_{t+1,s'}^\tau. \tag{12}$$

Proof. See Appendix A.2.

Theorem 2 shows that an equilibrium of Game 2 that satisfies the Wardrop equilibrium principle not only exists, but can be computed by solving optimization (9) and (10). In particular, the equations in (11) and (12) characterize a multi-commodity flow Wardrop equilibrium in the sense that no individual player can benefit from unilaterally switching to alternative actions at any time.

4. Efficient algorithms via linearization

In this section, we develop efficient numerical algorithms for the network equilibrium problems introduced in the previous section. We will first show that the linearized versions of problem (4) and problem (9) can be solved via Algorithm 1 and Algorithm 2. This observation motivates efficient iterative algorithms with detailed arithmetical complexity. Due to the limit of space, we omit some of the proof details in this section and include them in Yu et al. (2021).

We will use the following notation to simply our later discussions. Given $y, u \in \mathbb{R}^{T\times S\times A}$ and $z, w \in \mathbb{R}^{T\times S}$, we let $\phi(y), \phi^{-1}(u) \in \mathbb{R}^{T\times S\times A}$ and $\psi(z), \psi^{-1}(w) \in \mathbb{R}^{T\times S}$ be such that $[\phi(y)]_{tsa} = \phi_{tsa}(y_{tsa})$, $[\phi^{-1}(u)]_{tsa} = \phi_{tsa}^{-1}(u_{tsa})$, $[\psi(z)]_{ts} = \phi_{ts}(z_{ts})$, and $[\psi^{-1}(w)]_{ts} = \psi_{ts}^{-1}(w_{ts})$ for all $t \in [T]$, $s \in [S]$, $a \in [A]$. We also let $\underline{u}, \overline{u} \in \mathbb{R}^{T\times S\times A}$ and $\underline{w}, \overline{w} \in \mathbb{R}^{T\times S}$ be such that $\underline{u}_{tsa} = \phi_{tsa}(0)$, $\overline{u}_{tsa} = \phi_{tsa}(\rho)$, $\underline{w}_{ts} = \psi_{ts}(0)$ and $\overline{w}_{ts} = \psi_{ts}(\rho)$ for all $t \in [T]$, $s \in [S]$, $a \in [A]$.

4.1. Linearization and dynamic programming

If we approximate the objective function in (4) using its linearization at $u \in \mathbb{R}^{T\times S\times A}$ and $w \in \mathbb{R}^{T\times S}$, we obtain the following optimization:

$$\begin{aligned}
-g(u,w) = \min_{y,z} \quad & \sum_{t,s,a} u_{tsa} y_{tsa} + \sum_{t,s} w_{ts} z_{ts} \\
\text{s.t.} \quad & \text{constraints in problem (4).}
\end{aligned} \tag{13}$$

Similarly, if we approximate the objective function in (9) using its linearization at $u \in \mathbb{R}^{T\times S\times A}$, we obtain the following

$$\begin{aligned}
-h(u) = \min_{y^\tau,\tau\in\mathbb{T}} \quad & \sum_{t,s,a} \sum_{\tau,\tau\ge t} u_{tsa} y_{tsa}^\tau \\
\text{s.t.} \quad & \text{constraints in problem (9).}
\end{aligned} \tag{14}$$

The following two lemmas show that both optimization (13) and optimization (14) can be solved using Algorithm 1 and 2.

Lemma 1. *Suppose Assumption 1 holds. Let $(\hat{v}, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, T). Let $\hat{z}_{ts} = p_{ts}$ if $\hat{v}_{ts} > w_{ts}$, and $\hat{z}_{ts} = 0$ if $\hat{v}_{ts} \le w_{ts}$ for all $t \in [T]$, $s \in [S]$. In addition, let $\hat{y}$ be the output of Algorithm 2 with input $(\hat{\pi}, p - \hat{z}, P, T)$. Then $-g(u,w) = \sum_{t,s,a} u_{tsa}\hat{y}_{tsa} + \sum_{t,s} w_{ts}\hat{z}_{ts}$. Further, for any $u' \in \mathbb{R}^{T\times S\times A}$ and $w' \in \mathbb{R}^{T\times S}$,*

$$\begin{aligned}
& g(u', w') - g(u, w) \\
\ge & \sum_{t,s,a} (u'_{tsa} - u_{tsa})(-\hat{y}_{tsa}) + \sum_{t,s} (w'_{ts} - w_{ts})(-\hat{z}_{ts}).
\end{aligned}$$

Proof. See Yu et al. (2021, Appx. A.3).

Lemma 2. *Suppose Assumption 2 holds. Let* $(\hat{v}^\tau, \hat{\pi})$ *be the output of Algorithm 1 with input* (P, u, τ), $\hat{y}^\tau$ *be the output of Algorithm 2 with input* $(\hat{\pi}^\tau, p^\tau, P, \tau)$. *Then* $-h(u) = \sum_{t,s,a}\sum_{\tau,\tau\geq t} u_{tsa}\hat{y}^\tau_{tsa}$. *Further, for any* $u' \in \mathbb{R}^{T\times S\times A}$,

$$h(u') - h(u) \geq \sum_{t,s,a}\sum_{\tau,\tau\geq t}(u'_{tsa} - u_{tsa})(-\hat{y}^\tau_{tsa}).$$

Proof. See Yu et al. (2021, Appx. A.4).

4.2. Iterative algorithms using linearization

We now develop iterative algorithms for the optimization problems in Section 3 using linearization. We will use the following notion of ϵ-*optimal solutions*: given a constrained optimization, we say a solution is ϵ-optimal $\epsilon \in \mathbb{R}_+$ if it satisfies all the constraints and the objective function value evaluated at this solution is at most ϵ away from the optimal value. We also make the following assumptions on Games 1 and 2.

Assumption 3. Function $\phi_{tsa} : [0, \rho] \to \mathbb{R}$ and function $\psi_{ts} : [0, \rho] \to \mathbb{R}$ are L-Lipschitz continuous over their respective domains for all $t \in [T]$, $s \in [S]$ and $a \in [A]$.

Assumption 4. Function $\phi_{tsa} : [0, \rho] \to \mathbb{R}$ is L-Lipschitz continuous over its domain for all $t \in [T]$, $s \in [S]$ and $a \in [A]$.

Remark 4. Assumptions 3 and 4 are mild assumptions on the differentiability of the corresponding functions. For example, if ϕ_{tsa} is continuously differentiable, then using the mean value theorem one can show that Assumption 4 is satisfied by choosing

$$L \geq \max_{\alpha\in[0,\rho]} |\phi'_{tsa}(\alpha)|, \quad \forall t \in [T], s \in [S], a \in [A].$$

Algorithm 3 Frank–Wolfe method

Require: $p, P, \phi, \psi, T, \{\alpha^k\}$, initial value for y, z.

for $k = 1, 2, \ldots, K$ **do**

$\quad(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1$(P, \phi(y), T)$.

$\quad\hat{z}_{ts} = \begin{cases} p_{ts}, & \hat{v}_{ts} > \psi_{ts}(z_{ts}) \\ 0, & \hat{v}_{ts} \leq \psi_{ts}(z_{ts}) \end{cases}, \quad \forall t \in [T], s \in [S]$

$\quad\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p - \hat{z}, P, T)$.

$\quad y \leftarrow y - \alpha^k(y - \hat{y})$

$\quad z \leftarrow z - \alpha^k(z - \hat{z})$

end for

Algorithm 4 Multicommodity Frank–Wolfe method

Require: $p, P, \phi, \mathbb{T}, \{\alpha^k\}$, initial value for y^τ for all $\tau \in \mathbb{T}$.

for $k = 1, 2, \ldots, K$ **do**

$\quad\hat{y}_{tsa} = \sum_{\tau,\tau\geq t} y^\tau_{tsa}, \quad \forall t \in [T], s \in [S], a \in [A]$

$\quad(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow$Alg. 1$(P, \phi(\hat{y}), \tau), \quad \forall \tau \in \mathbb{T}$

$\quad\hat{y}^\tau \leftarrow$ Alg. 2$(\hat{\pi}^\tau, p^\tau, P, \tau), \quad \forall \tau \in \mathbb{T}$

$\quad y^\tau \leftarrow y^\tau - \alpha^k(y^\tau - \hat{y}^\tau), \quad \forall \tau \in \mathbb{T}$

end for

Based on Lemmas 1 and 2, we propose to solve optimization (4) and (9) using the Frank–Wolfe method (Frank & Wolfe, 1956), which repeatedly solves the linearized versions of (4) and (9). We summarize the Frank–Wolfe method for optimization (4) and (9) in Algorithm 3 and, respectively, Algorithm 4. The following theorem provides the overall arithmetic complexity analysis of Algorithm 3 and Algorithm 4.

Theorem 3. *Let* σ *be given by (3). If Assumptions 1 and 3 hold, then Algorithm 3 with* $\alpha^k = \frac{2}{k+1}$ *gives an* ϵ*-optimal solution of optimization (4) in* $O(\frac{\sigma TS^2A}{\epsilon})$ *arithmetic operations. Similarly, if Assumptions 2 and 4 hold, then Algorithm 4 with* $\alpha^k = \frac{2}{k+1}$ *gives an* ϵ*-optimal solution of optimization (9) in* $O(\frac{\sigma T^2S^2A}{\epsilon})$ *arithmetic operations.*

Proof. See Yu et al. (2021, Appx. A.5).

The idea of the proof is to combine the iteration complexity of Frank–Wolfe method (Bubeck et al., 2015, Thm. 3.8) together with the arithmetic complexity of Algorithm 1 and Algorithm 2.

What about optimization (5) and (10)? Observe that the optimization in (5) is equivalent to the following

$$\max_{u,w} \; -g(u,w) - \sum_{t,s,a}\int_{\phi_{tsa}(0)}^{u_{tsa}} \phi^{-1}_{tsa}(\alpha)d\alpha - \sum_{t,s}\int_{\psi_{ts}(0)}^{w_{ts}} \psi^{-1}_{ts}(\alpha)d\alpha \tag{15}$$

where

$$\begin{aligned} -g(u,w) = \max_{v,\lambda} \; & \sum_{t,s} p_{ts}(v_{ts} - \lambda_{ts}) \\ \text{s.t. } & \text{constraints in (5).} \end{aligned} \tag{16}$$

One can show that the optimization in (16) is exactly the dual problem of the linear program in (13). Since the constraint sets in (13) and (16) are both nonempty, the optimal value of (13) and (16) is the same (Ben-Tal & Nemirovski, 2001, Thm. 1.3.3). In other words, (5) can be equivalently written as follows

$$\begin{aligned} \max_{u,w} \; & -g(u,w) - \sum_{t,s,a}\int_{\phi_{tsa}(0)}^{u_{tsa}} \phi^{-1}_{tsa}(\alpha)d\alpha - \sum_{t,s}\int_{\psi_{ts}(0)}^{w_{ts}} \psi^{-1}_{ts}(\alpha)d\alpha \\ \text{s.t.} \; & -g(u,w) \text{ is the optimal value of (13).} \end{aligned} \tag{17}$$

Using similar reasoning, we can rewrite (10) as follows

$$\begin{aligned} \max_{u} \; & -h(u) - \sum_{t,s,a}\int_{\phi_{tsa}(0)}^{u_{tsa}} \phi^{-1}_{tsa}(\alpha)d\alpha \\ \text{s.t.} \; & -h(u) \text{ is the optimal value of (14).} \end{aligned} \tag{18}$$

Algorithm 5 Subgradient method

Require: $P, p, \phi, \psi, T, \{\alpha^k\}$, initial value for u, w.

for $k = 1, 2, \ldots, K$ **do**

$\quad(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1(P, u, T)

$\quad\hat{z}_{ts} = \begin{cases} p_{ts}, & \hat{v}_{ts} > w_{ts} \\ 0, & \hat{v}_{ts} \leq w_{ts} \end{cases}, \quad \forall t \in [T], s \in [S]$

$\quad\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p - \hat{z}, P, T)$.

$\quad u \leftarrow \min\{\overline{u}, \max\{\underline{u}, u + \alpha^k\left(\hat{y} - \phi^{-1}(u)\right)\}\}$

$\quad w \leftarrow \min\{\overline{w}, \max\{\underline{w}, w + \alpha^k\left(\hat{z} - \psi^{-1}(w)\right)\}\}$

end for

Algorithm 6 Multi-commodity subgradient method

Require: $p, P, \phi, \mathbb{T}, \{\alpha^k\}$, initial value of u.

for $k = 1, 2, \ldots, K$ **do**

$\quad(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow$ Alg. 1$(P, u, \tau), \quad \forall \tau \in \mathbb{T}$

$\quad\hat{y}^\tau \leftarrow$Alg. 2$(\hat{\pi}^\tau, p^\tau, P, \tau), \quad \forall \tau \in \mathbb{T}$

$\quad\hat{y}_{tsa} = \sum_{\tau,\tau\geq t} y^\tau_{tsa}, \quad \forall t \in [T], s \in [S], a \in [A]$

$\quad u \leftarrow \min\{\overline{u}, \max\{\underline{u}, u + \alpha^k\left(\hat{y} - \phi^{-1}(u)\right)\}\}$

end for

Based on Lemmas 1 and 2, we propose to solve optimization (17) and (18) using the projected subgradient method, which repeatedly computes the slope of a linear underestimate, or *subgradient*, of function $g(u, w)$ and function $h(u)$, respectively. We summarize the projected subgradient method for optimization (17) and (18) in Algorithm 5 and, respectively, Algorithm 6. The following theorem shows the overall arithmetic complexity of Algorithm 5 and Algorithm 6.

Theorem 4. *Let σ be given by (3). If Assumptions 1 and 3 hold, then Algorithm 6 with $\alpha^k = \frac{2L}{k+1}$ gives an ϵ-optimal solution to (17) using $O(\frac{\sigma TS^2A}{\epsilon})$ arithmetic operations. Similarly, if Assumptions 2 and 4 hold, then Algorithm 6 with $\alpha^k \equiv \frac{2LT}{k+1}$ gives an ϵ-optimal solution to (18) in $O(\frac{\sigma T^2S^2A}{\epsilon})$ arithmetic operations.*

Proof. See Yu et al. (2021, Appx. A.6).

The idea of the proof is to combine the iteration complexity of projected subgradient method (Bubeck et al., 2015, Thm. 3.9) together with the arithmetic complexity of Algorithm 1 and Algorithm 2.

Theorems 3 and 4 show that the arithmetic complexity of solving Markovian network equilibrium problems is the same as solving the corresponding MDP in terms of their dependency on the problem dimension and sparsity (*i.e.*, T, S, A, σ). In other words, the generalization from MDP to Markovian network equilibrium does not worsen the curse of dimensionality.

5. Numerical examples

In this section, we first illustrate the equilibrium models in Section 3 using a ride-sharing example, then demonstrate the efficiency of the algorithms in Section 4 by comparing them against commercial software MOSEK over extensive numerical experiments.

5.1. Multicommodity ride-sharing game

We consider the game played by ride-sharing drivers in Seattle, competing for customers (Li et al., 2019). We first abstract the Seattle area as an undirected graph illustrated in Fig. 1, whose nodes denote various neighborhoods in Seattle, and edges denote available routes, labeled by its driving distance. We denote the set of neighboring nodes of node s as $\mathcal{N}_s$. We model the decision-making of an ride-sharing driver on a typical weekend night (7 pm–1 am) as an MDP defined as follows.

- Time steps: $t = 1, 2, \ldots, 36$ denotes the (end of) 10-minute-intervals between 7 pm and 1 am.
- States: $[S]$ correspond to different nodes in graph $\mathcal{G}$.
- Actions: in state s, $a_{s'}$ denotes picking up a waiting rider with destination s' for all $s' \in \mathcal{N}_s$; a_{wait} denotes waiting for a future rider.
- Transition kernel: we assume $P_{sas'} = 1$ if $a = a_{s'}$, $s' \in \mathcal{N}_s$, $P_{sas'} = 1/(|\mathcal{N}_s| + 1)$ if $a = a_{\text{wait}}$, $s' \in \mathcal{N}_s \cup \{s\}$, and all other entries of $P_{sas'}$ are zero. Here we use a uniform distribution over neighboring states to describe how the drivers relocate themselves while waiting.
- Cost: due to the competition among drivers, we assume the profit for picking up a rider decreases with the amount of drivers making the same offer, namely $f_{tss'} = \alpha + \beta(1 - \frac{y_{tsa_{s'}}}{\gamma_{tss'}})\text{dist}_{ss'}$ for all $t \in [T], s \in [S]$, where α and β are the baseline profit and, respectively, nominal profit per mile. We let $\text{dist}_{ss'}$ denotes distance(miles) between s and s', $\gamma_{tss'}$ denotes the rider demand from s to s' at time t, and finally $y_{tsa_{s'}}$ denotes the amount of drivers choosing action $a_{s'}$ in state s at time t. The cost of action a in state s is a function of y_{tsa} given by $\phi_{tsa}(y_{tsa}) = -f_{tss'}$ if $a = a_{s'}$, $s' \in \mathcal{N}_s$, and $\phi_{tsa}(y_{tsa}) = -\sum_{s' \in \mathcal{N}_s} P_{sas'} f_{tss'}$ if $a = a_{\text{wait}}$.

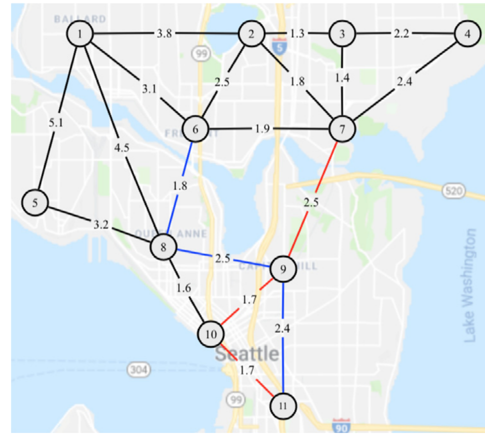


Fig. 1. Seattle transportation network and candidate LRT routes: 7-9-10-11 (red) and 6-8-9-11 (blue), map data ©Google 2021. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Values of $\gamma_{tss'}$.

$\gamma_{tss'}$	$s < 9 \le s'$	$s, s' \ge 9$	$s' < 9 \le s$	$s, s' < 9$
$1 \le t \le 6$	600	200	60	60
$7 \le t \le 30$	200	400	200	60
$31 \le t \le 36$	60	100	600	60

- Horizon: We assume that 10 drivers start working from each state every 10 min between 7 pm and 9 pm. Once started, each driver is assumed to only work for 4 consecutive hours to avoid fatigue, *i.e.*, $p_{ts}^{(t+24)} = 10$ for all $s \in [S]$ and $t \in [12]$.

The function ϕ_{tsa} defined above is linear with slope α, hence Assumption 2 is satisfied with $L = \alpha$. We also assume that each driver can travel between neighboring nodes within one time step in this simplified transportation network. In practice, such assumption can be ensured by adding more nodes to the network using a finer discretization of the interested area.

The equilibrium of the above game is a multi-commodity Markovian network equilibrium discussed in Section 3.2. We consider the scenario where $\alpha = 10, \beta = 0.2$, $\gamma_{tss'}$ is given in Table 1 and $\text{dist}_{ss'}$ is given in Fig. 1. We compute the driver number in the downtown area $\mathcal{D} = \{9, 10, 11\}$ by solving the optimization in (9) using commercial software Mosek (MOSEK ApS, 2019). Fig. 2 shows the results, where we can see that the driver number increases during $1 \le t \le 12$, then decreases during $24 \le t \le 36$. There are two sudden changes in the increasing/decreasing rate around $t = 7$ and $t = 31$, due to the corresponding changes in values of $\gamma_{tss'}$ in Table 1.

We further consider a transportation network design problem as follows. Suppose that Seattle city council is considering two candidate light rail transit (LRT) routes, 7-9-10-11 and 6-8-9-11 (see Fig. 1), to alleviate the congestion in downtown area, and each candidate will reduce the demand of ride-sharing services (namely, value of $\gamma_{tss'}$) by 50% along its route. Our results allow comparison of the two candidates using simulation, and the results are also given in Fig. 2, which shows that route 6-8-9-11 is more effective that route 7-9-10-11 in terms of reducing the amount of drivers in $\mathcal{D}$.

5.2. Computation experiments

To demonstrate the efficiency of the algorithms developed in Section 4, we compare the computation time of our algorithms

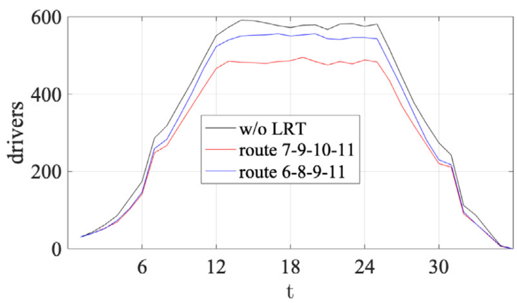


Fig. 2. Number of drivers in downtown area $\mathcal{D} = \{9, 10, 11\}$.

against commercial software Mosek, used in the previous section, over randomly generated examples. We use rand(a, b) to denote a random number sampled from uniform distribution over interval $[a, b]$ where $a, b \in \mathbb{R}$ and $a \leq b$. We let $P_{sas'} = \text{rand}(0, 1)$ for all $s \in [S], a \in [A]$, then normalized such that $\sum_{s'} P_{sas'} = 1$; $\phi_{tsa}(\alpha) = \text{rand}(1, 2)\alpha + \text{rand}(1, 2)$ for all $t \in [T], s \in [S], a \in [A]$; $p_{ts} = \text{rand}(0, 1)$ for all $s \in [S]$ if $t = 1$ and zero otherwise. In the variable demand case, we let $\psi_{ts}(\alpha) = \text{rand}(1, 2)\alpha - t + 21$ for all $t \in [T], s \in [S]$. In the multi-commodity flow case, we let $\mathbb{T} = \{5, 10\}$.

We fix $T = A = 10$ and let S range between 20 and 200, then test the computation time of Algorithm 3, Algorithm 5, Algorithm 4 and Algorithm 6, where all algorithms terminate when their objective function value agrees with the optimal one obtained by MOSEK with less than 0.5% relative error. The average computation time over 100 examples, along with corresponding 3-standard deviation interval are reported in Fig. 3.[3] All codes are in MATLAB and run on a 1.6 GHz laptop. From results in Fig. 3 we can see that, over the randomly generated 2000 examples, the subgradient method and the Frank–Wolfe method reduce the computation time consumed by Mosek by one and, respectively, two orders of magnitudes, at the price of a mere 0.5% of relative accuracy.

6. Conclusion

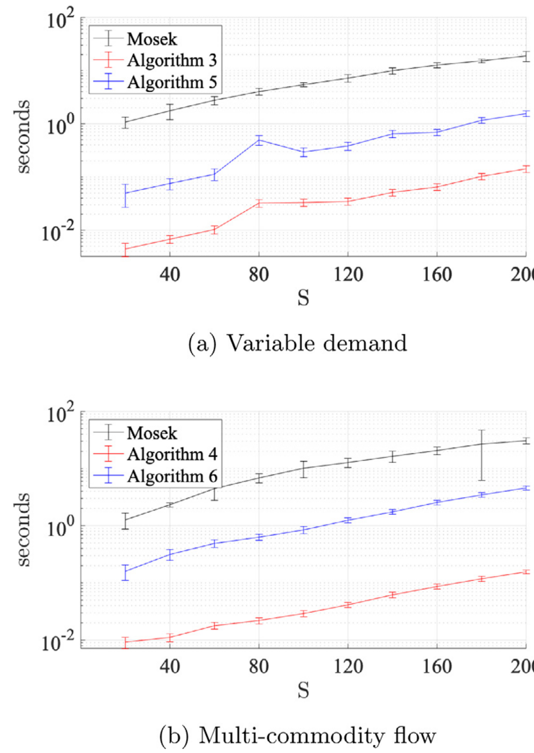
We study the variable demand and multi-commodity extensions in Markovian network equilibrium. We also propose efficient algorithms that outperform state-of-the-art commercial optimization software. However, the current work still has several limitations. For example, the cost of actions perceived by the players is assumed to be exact, rather than corrupted by stochastic noise, as considered in SUE model. Further, the ending time of each player is fixed at the beginning of the game. A more realistic assumption is to allow the players to change their ending time and recompute the equilibrium periodically. We aim to address these limitations in future work.

Appendix

A.1. Proof of Theorem 1

From the duality theorem of convex optimization (Rockafellar, 1970, Cor. 28.3.1) we know that, under Assumption 3, an optimal

[3] Since Mosek solves primal and dual problem simultaneously, we only report its solving time for optimization (4) and (9).



(a) Variable demand

(b) Multi-commodity flow

Fig. 3. Average computation time and 3-standard deviation intervals of 100 experiments with $T = A = 10$.

solution pair of optimization (4) and (5) necessarily satisfies the following KKT conditions

$$
\begin{aligned}
&v_{Ts} = \phi_{Tsa}(y_{Tsa}) - \mu_{Tsa},\\
&v_{ts} = \phi_{tsa}(y_{tsa}) + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa},\\
&v_{ts} = \psi_{ts}(z_{ts}) + \lambda_{ts} - \theta_{ts},\\
&y_{tsa}\mu_{tsa} = 0,\ \ z_{ts}\theta_{ts} = 0,\ \ \lambda_{ts}(z_{ts} - p_{ts}) = 0\\
&y_{tsa},\ z_{ts},\ \mu_{tsa},\ \theta_{ts},\ \lambda_{ts} \geq 0,
\end{aligned}
$$

for all $t \in [T], s \in [S], a \in [A]$. One can verify that the above conditions imply (6), (7) and (8).

A.2. Proof of Theorem 2

From the duality theorem of convex optimization (Rockafellar, 1970, Cor. 28.3.1) we know that, under Assumption 4, an optimal solution pair of optimization (9) and (10) necessarily satisfies the following KKT conditions

$$
\begin{aligned}
&v^{\tau}_{\tau s} = \phi_{\tau sa}\Big(\sum_{j, j\geq\tau} y^{j}_{\tau sa}\Big) - \mu^{\tau}_{\tau sa} = 0,\\
&v^{\tau}_{ts} = \phi_{tsa}\Big(\sum_{j, j\geq t} y^{j}_{tsa}\Big) + \sum_{s'} P_{sas'} v^{\tau}_{t+1,s'} - \mu^{\tau}_{tsa} = 0,\\
&y^{\tau}_{tsa}\mu^{\tau}_{tsa} = 0,\ \ y^{\tau}_{tsa},\ \mu^{\tau}_{tsa} \geq 0,
\end{aligned}
$$

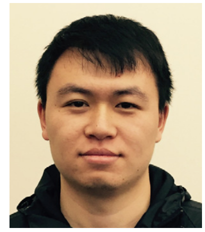
for all $t \in [\tau], \tau \in \mathbb{T}, s \in [S], a \in [A]$. One can verify that the above conditions imply (11) and (12).

References

Ahipaşaoğlu, S. D., Arıkan, U., & Natarajan, K. (2019). Distributionally robust Markovian traffic equilibrium. *Transportation Science*, *53*(6), 1546–1562.

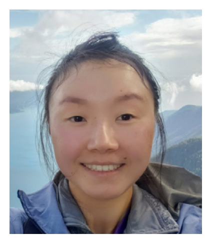
Baillon, J.-B., & Cominetti, R. (2008). Markovian traffic equilibrium. *Mathematical Programming*, *111*(1–2), 33–56.

Beckmann, M., McGuire, C. B., & Winsten, C. B. (1956). *Studies in the economics of transportation*. Yale University Press.
Ben-Tal, A., & Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM.
Bertsekas, D. P. (1996). *Neuro-dynamic programming*. Athena scientific Belmont.
Bertsekas, D. P. (1998). *Network optimization: continuous and discrete models*. Citeseer.
Bubeck, S., et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, *8*(3–4), 231–357.
Bürger, M., Zelazo, D., & Allgöwer, F. (2014). Duality and network theory in passivity-based cooperative control. *Automatica*, *50*(8), 2051–2061.
Calderone, D. J. (2017). *Models of competition for intelligent transportation infrastructure: parking, ridesharing, and external factors in routing decisions*. (Ph.D. thesis).
Calderone, D., & Sastry, S. (2017a). Infinite-horizon average-cost Markov decision process routing games. In *Proc. int. conf. intell. transp. syst.* (pp. 1–6). IEEE.
Calderone, D., & Sastry, S. (2017b). Markov decision process routing games. In *Proc. int. conf. cyber-phys. syst.* (pp. 273–279). ACM.
Cominetti, R., Facchinei, F., & Lasserre, J. B. (2012). Wardrop and stochastic user equilibrium. In *Modern optimization modelling techniques* (pp. 213–220). Springer.
Correa, J. R., & Stier-Moses, N. E. (2010). Wardrop equilibria. In *Wiley encyclopedia of operations research and management science*. Wiley Online Library.
Daganzo, C. F., & Sheffi, Y. (1977). On stochastic models of traffic assignment. *Transportation Science*, *11*(3), 253–274.
Dial, R. B. (1971). A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, *5*(2), 83–111.
Fisk, C. (1980). Some developments in equilibrium traffic assignment. *Transportation Research B*, *14*(3), 243–255.
Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, *3*(1–2), 95–110.
Gartner, N. H. (1980a). Optimal traffic assignment with elastic demands: A review, Part I. Analysis framework. *Transportation Science*, *14*(2), 174–191.
Gartner, N. H. (1980b). Optimal traffic assignment with elastic demands: A review, Part II. Algorithmic approaches. *Transportation Science*, *14*(2), 192–208.
Gomes, D. A., Mohr, J., & Souza, R. R. (2009). Discrete mean field games. arXiv preprint arXiv:0903.1620.
Gomes, D. A., Mohr, J., & Souza, R. R. (2010). Discrete time, finite state space mean field games. *Journal de Matheématiques Pures et Appliquées*, *93*(3), 308–328.
Guéant, O. (2011). From infinity to one: The reduction of some mean field games to a global control problem. arXiv preprint arXiv:1110.3441.
Guéant, O. (2015). Existence and uniqueness result for mean field games with congestion effect on graphs. *Applied Mathematics and Optimization*, *72*(2), 291–303.
Li, S. H., Yu, Y., Calderone, D., Ratliff, L., & Açrkmeşe, B. (2019). Tolling for constraint satisfaction in markov decision process congestion games. In *Proc. Amer. control conf.* (pp. 1238–1243). IEEE.
Liu, H. X., He, X., & He, B. (2009). Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics*, *9*(4), 485.
MOSEK ApS (2019). Mosek optimization toolbox for MATLAB. In *User's guide and reference manual, version 4*.
Nilim, A., & El Ghaoui, L. (2005). Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, *53*(5), 780–798.
Patriksson, M. (1994). *The traffic assignment problem: models and methods*. Courier Dover Publications.
Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
Rockafellar, R. T. (1970). *Convex analysis*. Princeton University Press.
Rockafellar, R. T. (1984). *Network flows and monotropic optimization*. John Wiley & Sons.
Sheffi, Y., & Powell, W. B. (1982). An algorithm for the equilibrium assignment problem with random link times. *Networks*, *12*(2), 191–207.
Tanaka, T., Nekouei, E., Pedram, A. R., & Johansson, K. H. (2020). Linearly solvable mean-field traffic routing games. *IEEE Transactions on Automatic Control*.
Wardrop, J. G., & Whitehead, J. I. (1952). Correspondence. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, *1*(5), 767–768.
Xiao, L., Johansson, M., & Boyd, S. P. (2004). Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, *52*(7), 1136–1144.
Yu, Y., Calderone, D., Li, S. H., Ratliff, L. J., & Açıkmeşe, B. (2021). Variable demand and multi-commodity flow in Markovian network equilibrium. arXiv preprint arXiv:1901.08731.



Yue Yu is a postdoctoral scholar at the Oden Institute for Computational Engineering and Sciences at The University of Texas at Austin. In 2020, Yue obtained his Doctor of Philosophy degree from the William E. Boeing Department of Aeronautics and Astronautics at University of Washington. Yue's research interests include optimization, optimization-based control, network games, and data-driven control.

Dan Calderone is a postdoctoral scholar and lecturer at the University of Washington, Seattle in Aeronautics and Astronautics and Electrical and Computer Engineering since 2018. He received his Ph.D. (2017) in Electrical Engineering and Computer Science at UC Berkeley. Calderone's research interests include game theory, Markov decision processes, network optimization, and learning in games. His teaching work focuses on developing visualizations for teaching linear algebra, convex optimization, control theory, and graph theory.



Sarah H.Q. Li is a Ph.D candidate at the University of Washington in Aeronautics and Astronautics Engineering since 2019. Her research interests include the application of game theory to competitive multi-agent systems in aerospace, Markov decision processes, and optimization algorithms. She is a Zonta Amelia Earhart Fellow. As a part of her Ph.D. and undergraduate degree, she has interned with Microsoft Research, Google Loon, ONERA, the French Aerospace Lab, Deutsche Elektronen-Synchrontron, and Macdonald Detwiler and Associates.

Lillian J. Ratliff is the Dhanani Endowed Assistant Professor in the Department of Electrical and Computer Engineering at the University of Washington. She also holds an Adjunct Assistant Professor position in the Allen School of Computer Science and Engineering and in the Aeronautics and Astronautics Department at UW. Prior to joining UW she obtained her Ph.D. (2015) in Electrical Engineering and Computer Sciences at UC Berkeley. Ratliff holds a MS (UNLV 2010) and BS (UNLV 2008) in Electrical Engineering as well as a BS (UNLV 2008) in Mathematics. Ratliff's research interests lie at the intersection of game theory, learning, and optimization. Ratliff is the recipient of an NSF Graduate Research Fellowship (2009), NSF CISE Research Initiation Initiative award (2017), and an NSF CAREER award (2019), and the ONR Young Investigator award (2020).

Behcet Acikmese is a professor at University of Washington (UW), Seattle. He received his Ph.D. in Aerospace Engineering from Purdue University. Before joining UW, he was a senior technologist at JPL and a lecturer at Caltech, and later a faculty member at the University of Texas at Austin. At JPL, he developed control algorithms for planetary landing, spacecraft formation flying, and asteroid and comet sample return missions. He developed the "flyaway" control algorithms in Mars Science Laboratory (MSL) mission, landed on Mars in 2012. His research interests include optimization and its applications in control theory, Markov decision processes, nonlinear and robust control, and game theory.