

Variable Demand and Multi-commodity Flow in Markovian Network Equilibrium

Yue Yu, Dan Calderone, Sarah H. Q. Li, Lillian J. Ratliff, Behçet Açıkmeşe

Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, 78712

Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, 98195

Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, 98195

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 state-of-the-art optimization software Mosek over extensive numerical experiments.

Key words: Wardrop equilibrium, Markov decision process, network optimization

1 Introduction

Network equilibrium problems arise in a variety of applications, such as resource allocation and routing in communication or transportation networks [Rockafellar, 1984, Bertsekas, 1998, Xiao et al., 2004, Bürger et al., 2014]. Among the most well-studied examples is the Wardrop equilibrium model in routing games [Beckmann et al., 1956, Gartner, 1980a, Gartner, 1980b, Correa and Stier-Moses, 2010, Patriksson, 1994]. In this model, users in a transportation network are assumed to choose routes with cost that they perceive as the lowest, *i.e.*, each user solves a shortest path problem, under the prevailing traffic conditions [Correa and Stier-Moses, 2010]. With this assumption, the resulting equilibria are characterized by the Wardrop equilibrium principle: the

cost of all the routes actually used are equal, and less than those which would be experienced by a single user on any unused route [Wardrop and Whitehead, 1952].

To ensure their practical relevance, it is often necessary to incorporate stochasticity into the network equilibrium problems. For example, the stochastic user equilibrium (SUE) model [Fisk, 1980, Sheffi and Powell, 1982, Liu et al., 2009] considers independent stochastic error on the route cost perceived by the users, leading to user distribution based on the logit [Dial, 1971] or probit model [Daganzo and Sheffi, 1977]; see [Patriksson, 1994, Sec. 2.8.1] and [Cominetti et al., 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 due to its assumption of independent route cost.

To address these drawbacks, different network models consider different type of stochasticity. In particular, [Baillon and Cominetti, 2008, Ahipaşaoglu et al., 2019] introduced a Markovian network equilibrium model where users are assumed to choose, instead of routes, sequences of actions with accumulated cost that they perceive as the lowest. Each action is accompanied by a deterministic outcome and a stochastic cost. For

¹ Y. Yu is with the Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas, 78712 (e-mail: yueyu@utexas.edu). S. H. Q. Li, and B. Açıkmeşe are with the William E. Boeing Department of Aeronautics & Astronautics, University of Washington, Seattle, Washington, 98195 (e-mail: yueyu@uw.edu, sarahli@uw.edu, behcet@uw.edu). D. Calderone and L. J. Ratliff are with the Department of Electrical & Computer Engineering, University of Washington, Seattle, Washington, 98195 (e-mail: djcal@uw.edu, ratliff1@uw.edu).

example, each vehicle in a transportation network is assumed to choose a sequence of arcs, where each arc leads to deterministic transition to the next node in the network and a stochastic amount of travel time [Baillon and Cominetti, 2008].

On the other hand, [Calderone and Sastry, 2017b, Calderone and Sastry, 2017a] 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 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 and El Ghaoui, 2005]. As a result, instead of a shortest path problem, each user solves a Markov decision process (MDP) [Puterman, 1994, Bertsekas, 1996], 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].

Although the results in [Calderone and Sastry, 2017b, Calderone and Sastry, 2017a] serves as a first step toward a more general class of stochastic dynamic network equilibrium model, 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 users having heterogeneous ending time.
- (2) We design novel dynamic-programming-based algorithms for Markovian network equilibrium problems with detailed arithematical 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 revisit 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. Section 5 first illustrates the equilibrium models in Section 3 via a multi-commodity ride-sharing example, then compares the algorithms in Section 4 against commercial software Mosek. Finally, we conclude with discussions and comments on the future directions of research 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 $\{1, 2, \dots, N\}$ for integer N ; a_{ijk} denotes the ijk -th component of the three-dimensional tensor $a \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and analogously, a_{ij} for the two-dimensional case. Given $b_1, \dots, b_N \in \mathbb{R}$, we say $(b^*, i^*) = \min_{i \in [N]} b_i$, if $b^* = \min_{i \in [N]} b_i$ and $i^* \in \operatorname{argmin}_{i \in [N]} b_i$.

2 Preliminaries and background

A T -horizon 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 $a \in [A]$. In order to find the optimal sequence of action that minimizes the expected accumulated cost, one can solve either one of the two following linear programs²:

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

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

where $p \in \mathbb{R}_+^{T \times S}$ 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 $1 \leq t \leq T$ 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 . In general, optimization (1) and (2) can be interpreted as the linear optimal distribution [Rockafellar, 1984, Sec.7A] and differential problem [Rockafellar, 1984, Sec.7E] defined on a T -layered *Markovian network* (see Fig. 1), where p_{ts} represents the divergence on state node s in the t -th layer, y_{tsa} represents the flow from state s to action a in the t -th layer, and v_{ts} represents the potential on state s in the t -th layer.

² Compared with the formulation in [Puterman, 1994], the linear program here also allows $p_{ts} > 0$ when $t > 1$.

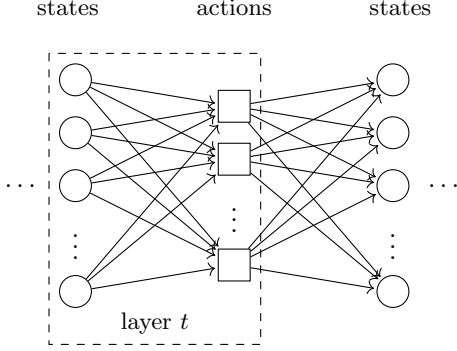


Fig. 1. A Markovian network

The following lemma shows that solutions of optimizations (1) and (2) satisfy the dynamic programming principle.

Lemma 1 ([Puterman, 1994]) Suppose y solves (1), and v solves (2). If $y_{tsa} > 0$ for any $t \in [T], s \in [S], a \in [A]$, then

$$(v_{Ts}, a) = \min_{a' \in [A]} c_{Tsa'},$$

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

for all $t \in [T - 1]$.

Perhaps the most efficient solution algorithm for problem (1) and (2) is *dynamic programming*, given by the following Algorithm 1 and Algorithm 2.

Algorithm 1 Backward induction

Input: P, c, T .

Output: v, π .

- 1: Let $(v_{Ts}, \pi_{Ts}) = \min_{a \in [A]} c_{Tsa}, \forall s \in [S]$.
 - 2: **for** $t = T - 1, T - 2, \dots, 1$ **do**
 - 3: $(v_{ts}, \pi_{ts}) = \min_{a \in [A]} (c_{tsa} + \sum_{s'} P_{sa's'} v_{t+1, s'}), \forall s \in [S]$
 - 4: **end for**
-

Algorithm 2 Forward induction

Input: π, p, P, T .

Output: y .

- 1: Initialize $y = 0$, let $y_{1s\pi_{1s}} \leftarrow p_{1s}$ for all $s \in [S]$.
 - 2: **for** $t = 1, 2, \dots, T - 1$ **do**
 - 3: $y_{t+1, s\pi_{t+1, s}} \leftarrow p_{t+1, s} + \sum_j P_{j\pi_{tj}s} y_{tj\pi_{tj}}, \forall s \in [S]$
 - 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 easily show that such solution pair (y, v) directly satisfies the Karush-Kuhn-Tucker

(KKT) conditions [Rockafellar, 1970, Thm. 28.3] of (1) and (2), hence it is an optimal primal-dual solution pair. If we define the *sparsity level* of an MDP as follows

$$\sigma = \max\{N_1, N_2\}/S \in [1/S, 1], \quad (3)$$

where $N_1 = \max_{s,a} |\{s' | P_{sas'} > 0\}|$ and $N_2 = \max_{s',a} |\{s | P_{sas'} > 0\}|$, then σS measures the maximum number of states connected by the transition kernel P . Further, it is straightforward to check that Algorithm 1 costs $O(\sigma TS^2 A)$ arithmetic operations, and Algorithm 2 costs $O(\sigma TS^2)$ arithmetic operations. In addition, Algorithm 1 and Algorithm 2 can be implemented as convolutional neural networks that allows efficient parallel computation [Tamar et al., 2016].

3 Markovian network equilibrium

By combining MDP together with classical routing games, Calderone and Sastry [Calderone and Sastry, 2017b] proposed MDP routing games where a fixed amount of players with the same planning horizon choose sequences of actions that they perceive as achieving the lowest expected accumulated cost under the prevailing choices of other players. Such games are similar to routing games 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.

In this section, we introduce two generalizations to MDP routing games that allow the amount of players to vary and the planning horizon to differ. We will also show that, under mild assumptions, the equilibria of such games can be computed efficiently using convex optimization.

3.1 Variable demand

One limitation of the MDP routing games in [Calderone and Sastry, 2017b] is the assumption that 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 variable demand MDP routing games.

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,

where 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 et al., 2009, Gomes et al., 2010, Guéant, 2011, Guéant, 2015, Tanaka et al., 2020]. The interactions among different players is 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 decided to play the game, which changes with the expected accumulated cost of the playing option according to curve 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 is 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 transition, which is equivalent to a shortest path problem.

The Wardrop equilibrium principle is a key characterization of the equilibria of network games [Patriksson, 1994, Correa and Stier-Moses, 2010]. The principle states that, at equilibria, 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 $\phi_{tsa} : [0, \rho] \rightarrow \mathbb{R}$ and function $\psi_{ts} : [0, \rho] \rightarrow \mathbb{R}$ 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.

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

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

In particular, the constraint $0 \leq z_{ts} \leq p_{ts}$ 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, under Assumption 1, the solution to the optimizations in (4) and (5) satisfy an equilibrium condition of Game 1. Similar to the Wardrop equilibrium principle, this equilibrium condition implies that no individual player can benefit from unilaterally switching its actions.

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

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

Further, if $y_{tsa} > 0$, then

$$\begin{aligned} (v_{Ts}, a) &= \min_{a' \in [A]} \phi_{Tsa'}(y_{Tsa'}), \\ (v_{ts}, a) &= \min_{a' \in [A]} \phi_{tsa'}(y_{tsa'}) + \sum_{s'} P_{sa's'} v_{t+1,s'}, \end{aligned} \quad (7)$$

for all $t \in [T-1]$.

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, equation (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., $v_{ts} \geq \psi_{ts}(p_{ts})$. 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., $v_{ts} \leq \psi_{ts}(p_{ts})$. 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, denote by \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 who have a common ending time $\tau \in \mathbb{T}$ with $\tau \geq t$, start the game from state s . Each of these players can choose the action a at the cost of $\phi_{tsa}(\sum_{\tau, \tau \geq t} y_{tsa}^\tau)$ 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 who plan to 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 start and end the game form a 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 $\tau \in \mathbb{T}$, $t \geq \tau$ 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 $\phi_{tsa} : [0, \rho] \rightarrow \mathbb{R}$ is continuous and strictly increasing, where $\rho = \sum_{\tau} \sum_{t \leq \tau, s} p_{ts}^\tau$.

With these assumptions, we now introduce the following pair of primal-dual optimization problems associated with Game 2. Notice that if $\mathbb{T} = \{T\}$, then they reduce to optimization (1) and (2), respectively.

$$\begin{aligned} \min_{\{y^\tau\}_{\tau \in \mathbb{T}}} \quad & \sum_{t,s,a} \int_0^{\sum_{\tau, \tau \geq t} y_{tsa}^\tau} \phi_{tsa}(\alpha) d\alpha \\ \text{s.t.} \quad & \sum_a y_{tsa}^\tau = p_{ts}^\tau \\ & \sum_a y_{t+1,sa}^\tau = p_{t+1,s}^\tau + \sum_{s',a} P_{s'as} y_{ts'a}^\tau, \quad t \in [\tau-1], \\ & 0 \leq y_{tsa}^\tau, \quad \forall \tau \in \mathbb{T}, t \in [\tau], s \in [S], a \in [A] \end{aligned} \quad (8)$$

$$\begin{aligned} \max_{u, \{v^\tau\}_{\tau \in \mathbb{T}}} \quad & \sum_{t,s} \sum_{\tau, \tau \geq t} p_{ts}^\tau v_{ts}^\tau - \sum_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha \\ \text{s.t.} \quad & v_{ts}^\tau \leq u_{tsa}, \\ & v_{ts}^\tau \leq u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}^\tau, \quad t \in [\tau-1], \\ & \forall \tau \in \mathbb{T}, s \in [S], a \in [A] \end{aligned} \quad (9)$$

The following theorem shows that, under Assumption 2, the solutions to optimization problems (8) and (9) satisfy the equilibrium condition of Game 2. Similar to the Wardrop equilibrium principle, this equilibrium condition implies that no individual player can benefit from unilaterally switching actions.

Theorem 2 Suppose Assumption 2 holds, y solves (8), and (u, v) solves (9). If $y_{tsa}^\tau > 0$ for any $\tau \in \mathbb{T}$, $s \in [S]$, $a \in [A]$, then

$$\begin{aligned} (v_{ts}^\tau, a) &= \min_{a' \in [A]} \phi_{tsa'} \left(\sum_{\tau, \tau \geq t} y_{tsa'}^\tau \right), \\ (v_{ts}^\tau, a) &= \min_{a' \in [A]} \phi_{tsa'} \left(\sum_{\tau, \tau \geq t} y_{tsa'}^\tau \right) + \sum_{s'} P_{sa's'} v_{t+1,s'}^\tau, \end{aligned} \quad (10)$$

for all $t \in [\tau-1]$.

Proof See Appendix A.2.

Theorem 2 shows that a Wardrop equilibrium of Game 2 not only exists, but can be found by solving optimization problems (8) and (9). In particular, the equations in (10) characterize a multi-commodity flow Wardrop equilibrium in the sense that no individual player can benefit from using alternative actions before his/her ending time τ for all $\tau \in \mathbb{T}$.

4 Efficient algorithms via linearization

In this section, we develop efficient iterative algorithms for the network equilibrium problems introduced in the previous section. In particular, we first prove that the linearized versions of problem (4) and problem (8) can both be solved in closed form via Algorithm 1 and Algorithm 2. This observation motivates efficient iterative algorithms that enjoy detailed arithmetical complexity analysis.

We will use the following notation to simplify 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

$$\begin{aligned} [\phi(y)]_{tsa} &= \phi_{tsa}(y_{tsa}), \quad [\phi^{-1}(u)]_{tsa} = \phi_{tsa}^{-1}(u_{tsa}), \\ [\psi(z)]_{ts} &= \psi_{ts}(z_{ts}), \quad [\psi^{-1}(w)]_{ts} = \psi_{ts}^{-1}(w_{ts}), \end{aligned} \quad (11)$$

for all $t \in [T], s \in [S], a \in [A]$. We also let $\underline{u}, \bar{u} \in \mathbb{R}^{T \times S \times A}$ and $\underline{w}, \bar{w} \in \mathbb{R}^{T \times S}$ be such that

$$\begin{aligned} \underline{u}_{tsa} &= \phi_{tsa}(0), \quad \bar{u}_{tsa} = \phi_{tsa}(\rho), \\ \underline{w}_{ts} &= \psi_{ts}(0), \quad \bar{w}_{ts} = \psi_{ts}(\rho), \end{aligned} \quad (12)$$

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

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

Observe that the above optimization is a modification to problem (1), by including an additional variable z . This suggest that (13) may also be solved using dynamic programming, which is confirmed by the following lemma.

Lemma 2 *Suppose Assumption 1 holds. Let $(\hat{v}, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, T) , and*

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

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 $w' \in \mathbb{R}^{T \times S}$,

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

Proof See Appendix A.3.

Similarly, if we approximate the objective function in (8) using a linear function, we obtain the following

$$\begin{aligned} -h(u) &= \min_{y^\tau, \tau \in \mathbb{T}} \sum_{t, s, a} \sum_{\tau \geq t} u_{tsa} y_{tsa}^\tau \\ \text{s.t.} \quad &\text{constraints in problem (8)}. \end{aligned} \quad (14)$$

where $u \in \mathbb{R}^{T \times S \times A}$ is the approximation parameter, $-h(u)$ is the optimal value of (14). The following lemma shows that optimization (14) can be solved using Algorithm 1 and 2 as well.

Lemma 3 *Suppose Assumption 2 holds. Let $(\hat{v}^\tau, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, τ) , \hat{y}^τ be the output of Algorithm 2 with input $(\hat{\pi}^\tau, p^\tau, P, \tau)$. Then*

$$-h(u) = \sum_{t, s, a} \sum_{\tau \geq t} u_{tsa} \hat{y}_{tsa}^\tau.$$

Further, for any $u' \in \mathbb{R}^{T \times S \times A}$,

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

Proof See Appendix A.4.

Remark 4 *Function $g : \mathbb{R}^{T \times S \times A} \times \mathbb{R}^{T \times S} \rightarrow \mathbb{R}$ in Lemma 2 is the support function of a polyhedron, which is closed and convex [Rockafellar, 1970, p.28]. Further, Lemma 2 shows that the slope of a linear underestimator, or subgradient, of function $g(u, w)$ can be computed using Algorithm 1 and Algorithm 2. Similar observation is made in Lemma 3 for function $h : \mathbb{R}^{T \times S \times A} \rightarrow \mathbb{R}$.*

4.2 Iterative algorithms using linearization

We now develop iterative algorithms for optimization problems in Section 3 using the results from the previous subsection. We will use the following notion of ϵ -optimal solution.

Definition 1 *Given a constrained optimization where an objective function is optimized subject to constraints, 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 will also use the following additional assumptions on Game 1 and, respectively, Game 2.

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

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

Remark 5 Assumption 3 and Assumption 4 are mild assumptions on the differentiability of the corresponding functions. For example, if ϕ_{tsa} is continuously differentiable, then the mean value theorem states that for any $\alpha_1, \alpha_2 \in [0, \rho]$, there exists $\alpha_3 \in [0, \rho]$ such that

$$|\phi_{tsa}(\alpha_1) - \phi_{tsa}(\alpha_2)| \leq |\phi'_{tsa}(\alpha_3)| \cdot |\alpha_1 - \alpha_2|.$$

where ϕ'_{tsa} is the derivative of function ϕ_{tsa} . Hence Assumption 4 is satisfied by choosing

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

which takes a bounded value since ϕ'_{tsa} is continuous. However, the continuity of ϕ'_{tsa} is not necessary. For example, if ϕ_{tsa} is a piecewise linear function, i.e., a function that is affine over a collection of intervals, then it is still Lipschitz continuous even if its derivative is not continuous.

Based on Lemma 2 and Lemma 3, we propose to solve optimization (4) and (8) using Frank-Wolfe method [Frank and Wolfe, 1956], which repeatedly solve the linearized versions of (4) and (8). We summarize the Frank-Wolfe method for optimization (4) and (8) in Algorithm 3 and, respectively, Algorithm 4. The following theorem provides the overall arithmetic complexity analysis of Algorithm 3 and Algorithm 4.

Algorithm 3 Frank-Wolfe method

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

```

1: for  $k = 1, 2, \dots, K$  do
2:    $(\hat{v}, \hat{\pi}) \leftarrow \text{Alg. 1}(P, \phi(y), T)$ .
3:    $\hat{z}_{ts} = \begin{cases} p_{ts}, & \hat{v}_{ts} > \psi_{ts}(z_{ts}) \\ 0, & \hat{v}_{ts} \leq \psi_{ts}(z_{ts}) \end{cases}, \quad \forall t \in [T], s \in [S]$ 
4:    $\hat{y} \leftarrow \text{Alg. 2}(\hat{\pi}, p - \hat{z}, P, T)$ .
5:    $y \leftarrow y - \alpha^k(y - \hat{y})$ 
6:    $z \leftarrow z - \alpha^k(z - \hat{z})$ 
7: end for
```

The following theorem shows the convergence property of Algorithm 3 and Algorithm 4.

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

Algorithm 4 Multicommodity Frank-Wolfe method

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

```

1: for  $k = 1, 2, \dots, K$  do
2:    $\hat{y}_{tsa} = \sum_{\tau, \tau \geq t} y_{tsa}^\tau, \quad \forall t \in [T], s \in [S], a \in [A]$ 
3:    $(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow \text{Alg. 1}(P, \phi(\hat{y}), \tau), \quad \forall \tau \in \mathbb{T}$ 
4:    $\hat{y}^\tau \leftarrow \text{Alg. 2}(\hat{\pi}^\tau, p^\tau, P, \tau), \quad \forall \tau \in \mathbb{T}$ 
5:    $y^\tau \leftarrow y^\tau - \alpha^k(y^\tau - \hat{y}^\tau), \quad \forall \tau \in \mathbb{T}$ 
6: end for
```

tions. Similarly, if Assumption 2 and 4 hold, then Algorithm 4 with $\alpha^k = \frac{2}{k+1}$ gives an ϵ -optimal solution to (8) in $O(\frac{\sigma T^2 S^2 A}{\epsilon})$ arithmetic operations.

Proof See Appendix A.5.

Theorem 1 provides arithmetical complexity of Algorithm 3 and 4, which not only depends on the problem size (i.e., T, S, A), but also the sparsity of the constraints (i.e., σ) in (4) and (8).

What about the dual problems? Observe that the optimization in (5) can be separated into two layers: an outer layer that optimizes over (u, w) , and an inner layer that optimizes over (v, λ) for a given value of (u, w) ; namely, (5) is equivalent to the following

$$\begin{aligned} \max_{u, w} \quad & -g(u, w) - \sum_{t, s, a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha \\ & - \sum_{t, s} \int_{\psi_{ts}(0)}^{w_{ts}} \psi_{ts}^{-1}(\alpha) d\alpha \end{aligned} \quad (15)$$

where

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

One can show that optimization (16) is exactly the dual problem of (13). Since the constraint sets in (13) and (16) are both nonempty, the optimal value of (13) and (16) are the same [Von Neumann and Morgenstern, 1953]. In other words, (5) can be equivalently written as follows

$$\begin{aligned} \max_{u, w} \quad & -g(u, w) - \sum_{t, s, a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha \\ & - \sum_{t, s} \int_{\psi_{ts}(0)}^{w_{ts}} \psi_{ts}^{-1}(\alpha) d\alpha \end{aligned} \quad (17)$$

s.t. $-g(u, w)$ is the optimal value of (13).

Using similar reasoning, we can rewrite (9) as follows

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

We already discussed in Remark 4 how the subgradients of function $g(u, w)$ and $h(u)$ can be computed efficiently using Algorithm 1 and Algorithm 2. In addition, all the other terms in the objective functions of problem (17) and (18) are continuously differentiable. This suggest that (17) and (18) are suited for the projected subgradient method, which optimizes a non-smooth function by repeatedly computing its subgradients and projections onto its domain. We summarize the projected subgradient method applied to (17) and (18) in Algorithm 5 and, respectively, Algorithm 6.

Algorithm 5 Subgradient method

Input: $P, p, \phi, \psi, T, \{\alpha^k\}$, initial value for u, w .
1: **for** $k = 1, 2, \dots, K$ **do**
2: $(\hat{v}, \hat{\pi}) \leftarrow \text{Alg. 1}(P, u, T)$
3: $\hat{z}_{ts} = \begin{cases} p_{ts}, & \hat{v}_{ts} > w_{ts} \\ 0, & \hat{v}_{ts} \leq w_{ts} \end{cases}, \forall t \in [T], s \in [S]$
4: $\hat{y} \leftarrow \text{Alg. 2}(\hat{\pi}, p - \hat{z}, P, T)$
5: $u \leftarrow \min\{\bar{u}, \max\{\underline{u}, u + \alpha^k(\hat{y} - \phi^{-1}(u))\}\}$
6: $w \leftarrow \min\{\bar{w}, \max\{\underline{w}, w + \alpha^k(\hat{z} - \psi^{-1}(w))\}\}$
7: **end for**

Algorithm 6 Multi-commodity subgradient method

Input: $p, P, \phi, \mathbb{T}, \{\alpha^k\}$, initial value of u .
1: **for** $k = 1, 2, \dots, K$ **do**
2: $(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow \text{Alg. 1}(P, u, \tau), \forall \tau \in \mathbb{T}$
3: $\hat{y}^\tau \leftarrow \text{Alg. 2}(\hat{\pi}^\tau, p^\tau, P, \tau), \forall \tau \in \mathbb{T}$
4: $\hat{y}_{tsa} = \sum_{\tau, \tau \geq t} y_{tsa}^\tau, \forall t \in [T], s \in [S], a \in [A]$
5: $u \leftarrow \min\{\bar{u}, \max\{\underline{u}, u + \alpha^k(\hat{y} - \phi^{-1}(u))\}\}$
6: **end for**

The following theorem shows the convergence property of Algorithm 5 and Algorithm 6.

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

Proof See Appendix A.6

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 algorithms in Section 4 by comparing them against commercial software Mosek (<https://www.mosek.com>) over extensive numerical experiments.

5.1 Multicommodity ride-sharing game

We consider the game played by ride-sharing drivers in Seattle, competing for customers. We first abstract the Seattle area as an undirected graph illustrated in Fig. 2, 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 a ride-sharing driver on a typical weekend night (7pm-1am) as an MDP defined as follows.

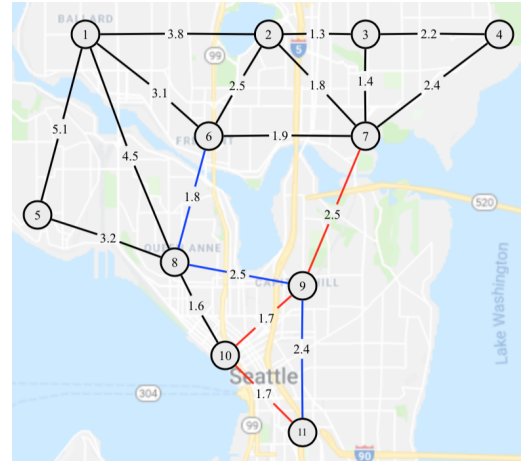


Fig. 2. Seattle transportation network and candidate LRT routes: 7-9-10-11(red) and 6-8-9-11(blue).

- Time steps: $t = 1, 2, \dots, 36$ denotes the (end of) 10-minute-intervals between 7pm and 1am.
- 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 $P_{sas'}$ is given by

$$P_{sas'} = \begin{cases} 1, & \text{if } a = a_{s'}, s' \in \mathcal{N}_s, \\ 1/(|\mathcal{N}_s| + 1), & \text{if } a = a_{\text{wait}}, s' \in \mathcal{N}_s \cup \{s\}. \end{cases}$$

All other entries of $P_{sas'}$ are zero. Here we use an uniform distribution over neighboring states to describe the uncertain destinations of future riders³.

- 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

$$f_{tss'} = \alpha + \beta \left(1 - \frac{y_{tsa_{s'}}}{\gamma_{tss'}}\right) \text{dist}_{ss'}, \quad (19)$$

³ Such distribution can be approximated more accurately using historical data in practical applications.

for all $t \in [T], s \in [S]$, where α and β is 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} defined as follows

$$\phi_{tsa}(y_{tsa}) = \begin{cases} -f_{tss'}, & \text{if } a = a_{s'}, s' \in \mathcal{N}_s. \\ -\sum_{s' \in \mathcal{N}_s} P_{sas'} f_{tss'}, & \text{if } a = a_{\text{wait}}. \end{cases}$$

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

Notice that the function ϕ_{tsa} defined above is linear with slope α , hence Assumption 2 is satisfied with $L = \alpha$. In general, as long as ϕ_{tsa} is modeled or approximated as a continuously differentiable function, Assumption 4 is always satisfied, as we discussed in Remark 5. 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.

Notice that since drivers can start the game at different times during $1 \leq t \leq 12$ and they will only plan for the next 24 time steps. In other words, drivers with heterogeneous planning time windows will coexist in the network for $t = 2, 3, \dots, 24$. Therefore the equilibrium of this 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. 2. We compute the driver number in the downtown area $\mathcal{D} = \{9, 10, 11\}$ by solving the optimization in (8) using commercial software Mosek (<https://www.mosek.com>). The results are demonstrate in Fig. 3, where we can see that the driver number increases during $1 \leq t \leq 12$, then decreases during $24 \leq t \leq 36$. There are also two sudden changes in the increasing/decreasing rate around $t = 7$ and $t = 31$, which is due to the corresponding changes in values of γ in Table 1. This example extends the single commodity case considered in [Calderone and Sastry, 2017b] and [Li et al., 2019] where all players enter and exit the game simultaneously.

A relevant application of the above simulation framework is transportation network design. For example, 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. 2), as a means to alleviate the congestion caused by the ride-sharing traffic in downtown

area, assuming that the LRT will reduce the demand of ride-sharing services (namely, value of $\gamma_{tss'}$) by 50% along its route. The simulated equilibrium with different LRT routes are also given in Fig. 3, which shows that route 6-8-9-11 is more effective than route 7-9-10-11 in terms of reducing amount of drivers in \mathcal{D} . These results clearly demonstrate the power of Markovian network equilibrium model in transportation system design.

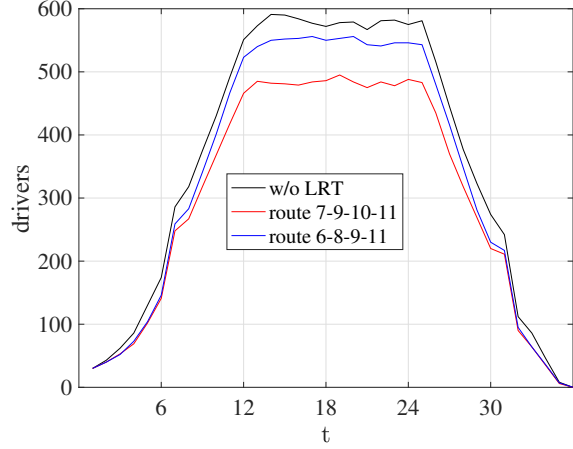


Fig. 3. Number of drivers in downtown area $\mathcal{D} = \{9, 10, 11\}$.

Table 1
Values of γ where $\mathcal{D} = \{9, 10, 11\}$.

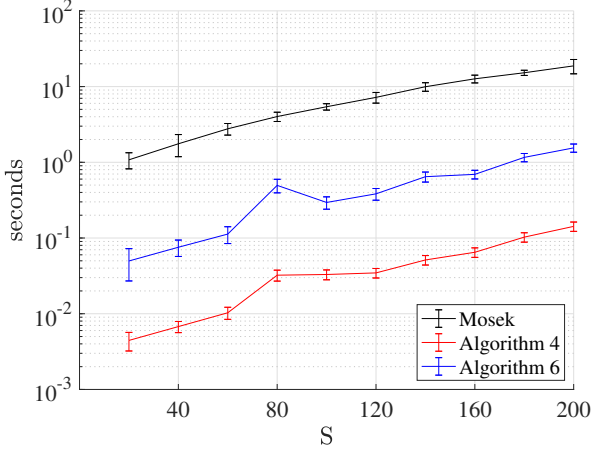
$\gamma_{tss'}$	$s \notin \mathcal{D}, s' \in \mathcal{D}$	$s \in \mathcal{D}, s' \in \mathcal{D}$	$s \in \mathcal{D}, s' \notin \mathcal{D}$	$s \notin \mathcal{D}, s' \notin \mathcal{D}$
$1 \leq t \leq 6$	600	200	60	60
$7 \leq t \leq 30$	200	400	200	60
$31 \leq t \leq 36$	60	100	600	60

5.2 Computation experiments

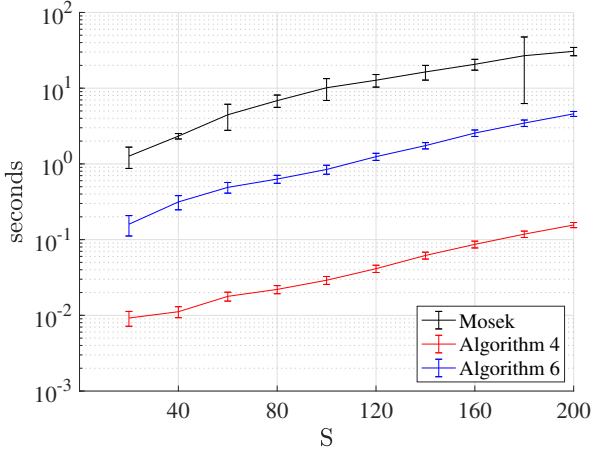
To demonstrate the efficiency of the algorithms developed in Section 4, we compare the computation time of our algorithms against commercial software Mosek, used in the previous section, over randomly generated examples. We use $\text{rand}(a, b)$ to denote a random number sampled from uniform distribution over interval $[a, b]$ where $a, b \in \mathbb{R}$ and $a \leq b$.

- $P_{sas'} = \text{rand}(0, 1)$ 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 $\mathbb{T} = \{5, 10\}$.



(a) Variable demand



(b) Multi-commodity flow

Fig. 4. Average computation time and 3-standard deviation intervals of 100 experiments with $T = A = 10$.

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. 4. All codes are in MATLAB and run on a 1.6GHz laptop. From results in Fig. 4 we can see that, over the randomly generated 2000 examples, subgradient method and Frank-Wolfe method reduces 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 stochastic user equilibrium 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.

A Appendix

A.1 Proof of Theorem 1

The objective function of problem (4) is convex (since ϕ_{tsa} and ψ_{ts} are strictly increasing), its constraints are affine, and the optimal value is obviously finite (since ϕ_{tsa} and ψ_{ts} are finitely valued). These imply that a solution pair to (4) and (5) necessarily satisfy the KKT conditions [Rockafellar, 1970, Thm. 28.3.1]. Let v_{ts} be the dual variable corresponding to the equality constraint containing p_{ts} , let $\mu_{tsa}, \theta_{ts}, \lambda_{ts} \geq 0$ be the dual variables corresponding to constraint $y_{tsa} \geq 0$, $z_{ts} \geq 0$ and, respectively, $z_{ts} \leq p_{ts}$. Then the Lagrangian of (4) and (5) is given by

$$\begin{aligned} L(y, z, v, \mu, \lambda, \theta) = & \sum_{t,s,a} \int_0^{y_{tsa}} \phi_{tsa}(\alpha) d\alpha - \sum_{t,s,a} \mu_{tsa} y_{tsa} \\ & + \sum_{t,s} \int_0^{z_{ts}} \psi_{ts}(\alpha) d\alpha + \sum_{t,s,a} v_{ts} (p_{ts} - z_{ts} - y_{tsa}) \\ & + \sum_{t,t < T} \sum_{s',a,s} v_{t+1,s} P_{s'as} y_{ts'a} - \sum_{t,s} ((\theta_{ts} - \lambda_{ts}) z_{ts} + \lambda_{ts} p_{ts}). \end{aligned}$$

The KKT conditions [Rockafellar, 1970, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L / \partial y_{tsa}, \partial L / \partial x_{ts}$ equal to zero)

$$\begin{aligned} v_{Ts} &= \phi_{Tsa}(y_{Tsa}) - \mu_{Tsa}, \\ v_{ts} &= \phi_{tsa}(y_{tsa}) + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa}, \quad t \in [T-1], \\ v_{ts} &= \psi_{ts}(z_{ts}) + \lambda_{ts} - \theta_{ts}, \quad t \in [T], \end{aligned} \tag{A.1}$$

for all $s \in [S], a \in [A]$, and the complementarity conditions

$$\begin{aligned} y_{tsa} \mu_{tsa} &= 0, \quad z_{ts} \theta_{ts} = 0, \quad \lambda_{ts} (z_{ts} - p_{ts}) = 0 \\ y_{tsa}, z_{ts}, \mu_{tsa}, \theta_{ts}, \lambda_{ts} &\geq 0, \quad \forall t \in [T], s \in [S], a \in [A]. \end{aligned} \tag{A.2}$$

Combining (A.1) and (A.2) yields (6) and (7). Note that same results can be derived from the dual problem (5).

A.2 Proof of Theorem 2

The objective function of problem (8) is convex (since ϕ_{tsa} is increasing), its constraints are affine, and the optimal value is obviously finite (since ϕ_{tsa} is finitely valued). These imply that a solution pair to (8) and (9) necessarily satisfies the KKT conditions [Rockafellar, 1970, Cor. 28.3.1]. Let v_{ts}^τ be the dual variable corresponding to the equality constraints containing $p_t^\tau s$, let $\mu_{tsa}^\tau \geq 0$ be the dual variables corresponding to constraint $y_{tsa}^\tau \geq 0$. Then the Lagrangian of (8) and (9) is given by

$$\begin{aligned} L(y, v, \mu) = & \sum_{t,s,a} \int_0^{\sum_{\tau, \tau \geq t} y_{tsa}^\tau} \phi_{tsa}(\alpha) d\alpha \\ & + \sum_{\tau} \sum_{t, t \leq \tau} \sum_{s,a} (v_{ts}^\tau (p_t^\tau - y_{tsa}^\tau) - \mu_{tsa}^\tau y_{tsa}^\tau) \\ & + \sum_{\tau} \sum_{t, t < \tau} \sum_{s', a, s} v_{t+1, s'}^\tau P_{s'as} y_{ts'a}^\tau. \end{aligned}$$

The KKT conditions [Rockafellar, 1970, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L / \partial y_{tsa}^\tau$ equal to zero)

$$\begin{aligned} v_{\tau s}^\tau &= \phi_{\tau sa} \left(\sum_{j, j \geq \tau} y_{\tau sa}^j \right) - \mu_{\tau sa}^\tau = 0, \\ v_{ts}^\tau &= \phi_{tsa} \left(\sum_{j, j \geq t} y_{tsa}^j \right) + \sum_{s'} P_{sas'} v_{t+1, s'}^\tau - \mu_{tsa}^\tau = 0, \end{aligned} \quad (\text{A.3})$$

for all $t \in [\tau - 1], \tau \in \mathbb{T}, s \in [S], a \in [A]$, and the complementarity conditions

$$y_{tsa}^\tau \mu_{tsa}^\tau = 0, \quad y_{tsa}^\tau, \mu_{tsa}^\tau \geq 0, \quad (\text{A.4})$$

for all $t \in [\tau], \tau \in \mathbb{T}, s \in [S], a \in [A]$. Combining (A.3) and (A.4) yields (10). Again, same results can be derived from the dual problem (9).

A.3 Proof of Lemma 2

Using a similar argument as in the proof of Theorem 1, we can show that the KKT conditions of (13) are given by the following

$$\begin{aligned} \sum_a y_{1sa} &= p_{1s} - z_{1s}, \\ \sum_a y_{t+1, sa} &= p_{t+1, s} - z_{t+1, s} + \sum_{s', a} P_{s'as} y_{ts'a}, \quad t \in [T - 1], \\ v_{Ts} &= u_{Ts} - \mu_{Ts}, \\ v_{ts} &= u_{ts} + \sum_{s'} P_{sas'} v_{t+1, s'} - \mu_{tsa}, \quad t \in [T - 1], \\ v_{ts} &= w_{ts} + \lambda_{ts} - \theta_{ts}, \\ y_{tsa} \mu_{tsa} &= 0, \quad z_{ts} \theta_{ts} = 0, \quad \lambda_{ts} (z_{ts} - p_{ts}) = 0, \\ y_{tsa}, z_{ts}, \mu_{tsa}, \theta_{ts}, \lambda_{ts} &\geq 0, \end{aligned} \quad (\text{A.5})$$

for all $t \in [T], s \in [S], a \in [A]$. Let $(\hat{v}, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, T) , $\hat{z} = p \odot (\hat{v} > w)$, \hat{y} be

the output of Algorithm 2 with input $(\hat{\pi}, p - \hat{z}, P, T)$, let

$$\begin{aligned} \hat{\mu}_{Ts} &= -\hat{v}_{Ts} + u_{Ts}, \\ \hat{\mu}_{tsa} &= -\hat{v}_{ts} + u_{tsa} + \sum_{s'} P_{sas'} \hat{v}_{t+1, s'}, \quad t \in [T - 1], \\ \hat{\lambda}_{ts} &= \max\{\hat{v}_{ts} - w_{ts}, 0\}, \quad \hat{\theta}_{ts} = \max\{w_{ts} - \hat{v}_{ts}, 0\}, \end{aligned} \quad (\text{A.6})$$

for all $t \in [T], s \in [S], a \in [A]$. Then it is straightforward to verify that $(\hat{y}, \hat{z}, \hat{v}, \hat{\mu}, \hat{\lambda}, \hat{\theta})$ satisfies all the KKT conditions in (A.5), hence (\hat{y}, \hat{z}) solves (13), which proves the equality. The inequality follows from the fact that, when (u, w) in (13) is perturbed to another value (u', w') , solution (\hat{y}, \hat{z}) is still feasible, but can be suboptimal.

A.4 Proof of Lemma 3

Notice that in optimization (14), both objective function and constraints are completely separable across y^τ with different value of τ . In other words, solving (14) is equivalent to solve the following optimization problem for each value of $\tau \in \mathbb{T}$ separately

$$\begin{aligned} \min_{y^\tau} \quad & \sum_{t \leq \tau, s, a} u_{tsa} y_{tsa}^\tau \\ \text{s.t.} \quad & \sum_a y_{1s}^\tau = p_{1s}^\tau \\ & \sum_a y_{t+1, sa}^\tau = p_{t+1, s}^\tau + \sum_{s', a} P_{s'as} y_{ts'a}^\tau, \quad t \in [\tau - 1], \\ & 0 \leq y_{tsa}^\tau, \quad \forall t \in [\tau], s \in [S], a \in [A]. \end{aligned} \quad (\text{A.7})$$

Since problem (A.7) is nothing but an instance of (1) with $T = \tau$, it can be solved by the output of Algorithm 2 with input $(\hat{\pi}^\tau, p^\tau, P, \tau)$, where $\hat{\pi}^\tau$ is the output of Algorithm 1 with input (P, u, τ) . This proves the equality; the inequality follows from the fact that, when u in (14) is perturbed to another value u' , solution $(\hat{y}^\tau, \tau \in \mathbb{T})$ is still feasible, but can be suboptimal.

A.5 Proof of Theorem 3

We start with Algorithm 3. The per-iteration computation of Algorithm 3 is clearly dominated by the execution of Algorithm 1 and Algorithm 2, which together cost $O(\sigma T S^2 A)$ arithmetical operations. Let $f(y, z)$ denote the objective function of problem (4). Then the gradients of f is given by

$$\partial_y f = \phi(y), \quad \partial_z f = \psi(z),$$

where $\phi(y)$ and $\psi(z)$ are defined as in (11). Then under Assumption 3, we know both $\frac{\partial f}{\partial y}$ and $\frac{\partial f}{\partial z}$ are Lipschitz. In addition, for any (y, z) satisfying the constraints of problem (4), one must have $y_{tsa} \in [0, \rho]$ and $z_{ts} \in [0, \rho]$, due to Assumption 1. In other words, the constraint set of problem (4) is a subset of $[0, \rho]^{T \times S \times A} \times [0, \rho]^{T \times S}$, which is bounded.

Therefore problem (4) is minimizing a function with Lipschitz gradients over a bounded set. Hence the Frank-Wolfe method given by Algorithm 3 converges to ϵ -optimal solution in $O(\frac{1}{\epsilon})$ iterations [Bubeck et al., 2015, Thm.3.8]. The proof for Algorithm 4 is similar.

A.6 Proof of Theorem 4

We start with Algorithm 5. First, the per-iteration computation of Algorithm 5 is clearly dominated by the execution of Algorithm 1 and Algorithm 1, which together cost $O(\sigma TS^2A)$ arithmetical operations. From Assumption 1 and 3 we have, for any $u_{tsa}, u'_{tsa}, w_{ts}, w'_{ts} \in [0, \rho]$

$$\begin{aligned} |u_{tsa} - u'_{tsa}| &= |\phi_{tsa}(\phi_{tsa}^{-1}(u_{tsa}) - \phi_{tsa}(\phi_{tsa}^{-1}(u'_{tsa}))| \\ &\leq L|\phi_{tsa}^{-1}(u_{tsa}) - \phi_{tsa}^{-1}(u'_{tsa})|, \\ |w_{ts} - w'_{ts}| &= |\psi_{ts}(\psi_{ts}^{-1}(w_{ts}) - \psi_{ts}(\psi_{ts}^{-1}(w'_{ts}))| \\ &\leq L|\psi_{ts}^{-1}(w_{ts}) - \psi_{ts}^{-1}(w'_{ts})|, \end{aligned}$$

for all $t \in [T], s \in [S], a \in [A]$, which implies the objective function of problem (17) (in particular, the integral terms) is $\frac{1}{L}$ -strongly convex [Nesterov, 2013, Thm.2.1.10]. Let $-f(u, w)$ denote the objective function of problem (17). Then from Lemma 2 we know that the subgradients of f is given by

$$\partial_u f = -\hat{y} + \phi^{-1}(u), \quad \partial_w f = -\hat{z} + \psi^{-1}(w),$$

where $\phi^{-1}(u), \psi^{-1}(w)$ are defined as in (11), (\hat{y}, \hat{z}) is a solution to problem (13). From Assumption 1 we know that $\phi^{-1}(u) \in [0, \rho]^{T \times S \times A}$ and $\psi^{-1}(w) \in [0, \rho]^{T \times S}$. Further, (\hat{y}, \hat{z}) must satisfy the constraints in problem (13), which implies that $\hat{y} \in [0, \rho]^{T \times S \times A}$ and $\hat{z} \in [0, \rho]^{T \times S}$. Hence the elements in $\partial_u f$ and $\partial_w f$ are bounded.

Therefore, problem (17) is minimizing a strongly convex function whose subgradients have bounded elements. hence the projected subgradient method given by Algorithm 5 converges to an ϵ -optimal solution in $O(\frac{1}{\epsilon})$ iterations [Bubeck et al., 2015, Th.3.9]. The proof for Algorithm 6 is similar.

References

- [Ahipařaoglu et al., 2019] Ahipařaoglu, S. D., Arıkan, U., and Natarajan, K. (2019). Distributionally robust markovian traffic equilibrium. *Transp. Sci.*, 53(6):1546–1562.
- [Baillon and Cominetti, 2008] Baillon, J.-B. and Cominetti, R. (2008). Markovian traffic equilibrium. *Math. Prog.*, 111(1-2):33–56.
- [Beckmann et al., 1956] Beckmann, M., McGuire, C. B., and Winsten, C. B. (1956). *Studies in the Economics of Transportation*. Yale University Press.
- [Bertsekas, 1996] Bertsekas, D. P. (1996). *Neuro-Dynamic Programming*. Athena scientific Belmont.
- [Bertsekas, 1998] Bertsekas, D. P. (1998). *Network Optimization: Continuous and Discrete Models*. Citeseer.
- [Bubeck et al., 2015] Bubeck, S. et al. (2015). Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357.
- [Bürger et al., 2014] Bürger, M., Zelazo, D., and Allgöwer, F. (2014). Duality and network theory in passivity-based cooperative control. *Automatica*, 50(8):2051–2061.
- [Calderone and Sastry, 2017a] Calderone, D. and Sastry, S. (2017a). Infinite-horizon average-cost markov decision process routing games. In *Proc. Int. Conf. Intell. Transp. Syst.*, pages 1–6. IEEE.
- [Calderone and Sastry, 2017b] Calderone, D. and Sastry, S. (2017b). Markov decision process routing games. In *Proc. Int. Conf. Cyber-Phys. Syst.*, pages 273–279. ACM.
- [Calderone, 2017] Calderone, D. J. (2017). *Models of Competition for Intelligent Transportation Infrastructure: Parking, Ridesharing, and External Factors in Routing Decisions*. PhD Thesis.
- [Cominetti et al., 2012] Cominetti, R., Facchinei, F., and Lasserre, J. B. (2012). Wardrop and stochastic user equilibrium. In *Modern Optimization Modelling Techniques*, pages 213–220. Springer.
- [Correa and Stier-Moses, 2010] Correa, J. R. and Stier-Moses, N. E. (2010). Wardrop equilibria. *Wiley Encyclopedia of Operations Research and Management Science*.
- [Daganzo and Sheffi, 1977] Daganzo, C. F. and Sheffi, Y. (1977). On stochastic models of traffic assignment. *Transp. Sci.*, 11(3):253–274.
- [Dial, 1971] Dial, R. B. (1971). A probabilistic multipath traffic assignment model which obviates path enumeration. *Transp. Res.*, 5(2):83–111.
- [Fisk, 1980] Fisk, C. (1980). Some developments in equilibrium traffic assignment. *Transp. Res. B*, 14(3):243–255.
- [Frank and Wolfe, 1956] Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 3(1-2):95–110.
- [Gartner, 1980a] Gartner, N. H. (1980a). Optimal traffic assignment with elastic demands: A review, Part I. Analysis framework. *Transp. Sci.*, 14(2):174–191.
- [Gartner, 1980b] Gartner, N. H. (1980b). Optimal traffic assignment with elastic demands: A review, Part II. Algorithmic approaches. *Transp. Sci.*, 14(2):192–208.
- [Gomes et al., 2009] Gomes, D. A., Mohr, J., and Souza, R. R. (2009). Discrete mean field games. *arXiv preprint arXiv:0903.1620*.
- [Gomes et al., 2010] Gomes, D. A., Mohr, J., and Souza, R. R. (2010). Discrete time, finite state space mean field games. *J. Math. Pures Appl.*, 93(3):308–328.
- [Guéant, 2011] 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, 2015] Guéant, O. (2015). Existence and uniqueness result for mean field games with congestion effect on graphs. *Appl. Math. Optim.*, 72(2):291–303.
- [Li et al., 2019] Li, S. H., Yu, Y., Calderone, D., Ratliff, L., and Aćrkmeře, B. (2019). Tolling for constraint satisfaction in markov decision process congestion games. In *Proc. Amer. Control Conf.*, pages 1238–1243. IEEE.
- [Liu et al., 2009] Liu, H. X., He, X., and 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.

- [Nesterov, 2013] Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.
- [Nilim and El Ghaoui, 2005] Nilim, A. and El Ghaoui, L. (2005). Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798.
- [Patriksson, 1994] Patriksson, M. (1994). *The Traffic Assignment Problem: Models and Methods*. Courier Dover Publications.
- [Puterman, 1994] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- [Rockafellar, 1970] Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.
- [Rockafellar, 1984] Rockafellar, R. T. (1984). *Network Flows and Monotropic Optimization*. John Wiley & Sons.
- [Sheffi and Powell, 1982] Sheffi, Y. and Powell, W. B. (1982). An algorithm for the equilibrium assignment problem with random link times. *Networks*, 12(2):191–207.
- [Tamar et al., 2016] Tamar, A., Wu, Y., Thomas, G., Levine, S., and Abbeel, P. (2016). Value iteration networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 2154–2162.
- [Tanaka et al., 2020] Tanaka, T., Nekouei, E., Pedram, A. R., and Johansson, K. H. (2020). Linearly solvable mean-field traffic routing games. *IEEE Trans. Autom. Control*.
- [Von Neumann and Morgenstern, 1953] Von Neumann, J. and Morgenstern, O. (1953). *Theory of games and economic behavior*. Princeton University Press.
- [Wardrop and Whitehead, 1952] Wardrop, J. G. and Whitehead, J. I. (1952). Correspondence. some theoretical aspects of road traffic research. *Proc. Inst. Civil Eng.*, 1(5):767–768.
- [Xiao et al., 2004] Xiao, L., Johansson, M., and Boyd, S. P. (2004). Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun.*, 52(7):1136–1144.