

Adaptive Constraint Satisfaction for Markov Decision Process Congestion Games: Application to Transportation Networks

Sarah H.Q. Li ^a, Yue Yu ^a, Nico Miguel ^c, Daniel Calderone ^b Lillian J. Ratliff ^b
Behçet Açıkmüşe ^a

^aDepartment of Aeronautics and Astronautics, University of Washington, Seattle, USA. (e-mail: {sarahli, yueyu, behcet}@uw.edu).

^bDepartment of Electrical Engineering, University of Washington, Seattle, Washington, USA. (e-mail: {djcal, ratliff}@uw.edu)

^cDepartment of Mechanical Engineering, University of Washington, Seattle, Washington, USA. (e-mail: nmiguel@uw.edu)

Abstract

Under the *Markov decision process (MDP) congestion game* framework, we study the problem of enforcing population distribution constraints on a population of players with stochastic dynamics and coupled congestion costs. Existing research demonstrates that the constraints on the players' population distribution can be satisfied by enforcing tolls. However, to compute the minimum toll value for constraint satisfaction requires accurate modeling of the player's congestion costs. Motivated by settings where an accurate congestion cost model is unavailable (e.g. transportation networks), we consider a MDP congestion game with *unknown* congestion costs. We assume that a constraint-enforcing authority can repeatedly enforce tolls on a population of players who converges to an ϵ -optimal population distribution for any given toll. We then construct a myopic update algorithm to compute the minimum toll value while ensuring that the constraints are satisfied on average. We analyze how the players' sub-optimal responses to tolls impact the rates of convergence towards the minimum toll value and constraint satisfaction. Finally, we apply our results to transportation by building a high-fidelity game model using data from the New York City's (NYC) Taxi and Limousine Commission (TLC), and illustrate how to efficiently reduce congestion while minimizing the impact on driver earnings.

Key words: Markov decision process, incentive design, congestion games, online optimization, transportation systems, stochastic games

1 Introduction

Congestion games play a fundamental role in engineering [26]. In large-scale networks such as urban traffic and electricity markets, congestion games capture how the competition among self-motivated decision-makers, known as the *players*, impacts network-level trends [23,10]. In particular, when all the players are equipped with identical congestion costs and transition probabilities in a finite state-action space, the game can be modeled as a *Markov decision process (MDP) congestion game* [6].

We study the feasibility of using tolls to enforce system-level constraints on a game in which both the players and the system operator do not know the true congestion costs. We are motivated by a plethora of system-level constraints encountered in large-scale networks, including satisfying safety constraints for decentralized autonomous swarms, meeting carbon-emission targets in urban transportation systems and minimizing voltage violations in competitive electricity markets [7,15,17]. Since tolling is a common and easily imple-

mentable mechanism in networked systems [31], we assume the system operator can freely impose tolls on the players.

We are interested in the *minimum toll value* that ensures constraint satisfaction, which can be solved as a function of the MDP congestion costs [14]. However, extracting the MDP congestion cost is difficult when the player objectives are complex and unknown to the constraint-enforcing entity. This is also true in simulation engines and higher complexity models, where the effects of a given toll can be computed, but not the minimum toll value. As such, we assume that there exists an oracle who can compute an ϵ -optimal solution for a game with a known toll. The inexact oracle is motivated by model-free, learning-based algorithms that can approximate the Nash equilibria for routing games with unknown link costs [33,12].

Contributions. We derive a gradient-based tolling algorithm that enforces linear population distribution constraints on a class of MDP congestion games with unknown but strictly increasing congestion costs. The algorithm requires access

to an inexact oracle that takes an input toll and returns a population distribution that is ϵ -optimal for the toll-augmented game. We show a direct relationship between the ϵ -optimal population distribution and an inexact gradient of the toll-augmented game with respect to the toll. We bound the following quantities as functions of the oracle's sub-optimality ϵ : convergence of 1) the average toll value towards the minimum toll value, 2) the average population distribution towards the optimal distribution under the minimum toll, 3) the average constraint violation towards zero. Finally, we construct a high-fidelity game model using real-world data from NYC TLC [21] to demonstrate our algorithm's effectiveness at reducing congestion in transportation networks.

The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, we review MDP congestion games. In Section 4, we introduce the toll-augmented game and the inexact oracle. In Section 5, we introduce the tolling algorithm and prove its convergence properties. In Section 6, the MDP congestion game model is used to reduce ride-share congestion levels in Manhattan, NYC.

2 Literature Overview

MDP congestion games [6] are related to non-atomic routing games [2,23], stochastic games [29], and mean field games [13], but differ in modeling assumptions. MDP congestion games extend non-atomic routing games by generalizing the player dynamics from deterministic to stochastic. Stochastic games assume that player costs differ and are functions of the joint policy. MDP congestion games assume that players costs are identical and are functions of the population distribution [32]. Finally, MDP congestion games are analogous to a discrete mean field game where the continuous stochastic processes are discretized in time, state, and action spaces. We show that these differences in assumptions enable MDP congestion games to more easily model large-scale networks such as transportation systems.

Tolling schemes for non-atomic routing games have been studied under capacitated traffic assignment literature [23, Sec. 2.8.2]. Adaptive incentive design for games has also been considered in deterministic and stochastic settings in [25] for players without MDP dynamics. Presently, we adopt a form of adaptive incentive design to guarantee *constraint satisfaction*. Tolling to satisfy external objectives is more generally interpreted as a Stackelberg game between a leader and its followers [30]. Techniques for updating the Stackelberg leader's actions to optimize the social cost of its followers are derived in [27]. Tolling non-atomic games where players have identical MDP dynamics and unknown congestion costs has not yet been analyzed.

Our minimum toll computation algorithm is an inexact gradient descent [8], and has been applied to settings such as distributed optimization and model predictive control [11,19]. In the game theory, the inexact gradient descent method has been applied to computing the Nash equilibria of a two

player min-max game [22]. However, it is not been previously connected to approximate Wardrop equilibria.

3 MDP Congestion Game

Notation. The notation $[K] = \{0, \dots, K-1\}$ denotes an index set of length K , $\mathbb{R}(\mathbb{R}_+)$ denotes a set of real (non-negative) numbers, $\mathbf{1}_N$ denotes a vector of ones of size N , and $[x]_+ = \max\{x, 0\}$ denotes a vector-valued function in which max is element-wise applied to vectors x and 0.

Consider a continuous population of players, each with identical MDP dynamics and congestion costs over a state-action set $[S] \times [A]$ for $(T+1)$ time steps. Under the non-atomic game assumption, the relationship between individual player distributions and the population distribution is described in [12, Sec.2]. Presently, we are only concerned with the resulting population distribution. We denote the set of feasible population distributions as $\mathcal{Y}(P, p)$, given by

$$\mathcal{Y}(P, p) = \{y \in \mathbb{R}_+^{(T+1)SA} \mid \sum_a y_{t+1,sa} = \sum_{s',a} P_{ts's'a} y_{ts's'a}, \sum_a y_{0sa} = p_s\}, \quad (1)$$

where y_{tsa} is the portion of the playing population who takes action a from state s at time t . We emphasize that y is a vector in $\mathbb{R}_+^{(T+1)SA}$ whose coordinates are ordered as

$$y = [y_{000} \dots y_{010} \dots y_{100} \dots y_{T(S-1)(A-1)}]^\top \quad (2)$$

The stochastic transition dynamics are given by $P \in \mathbb{R}^{T \times S \times S \times A}$, where $P_{ts's'a}$ denotes the transition probability from state s' to s under action a at time t . The transition dynamics satisfy $\sum_{s' \in [S]} P_{ts's'a} = 1 \forall (t, s, a) \in T \times S \times A$, and $P_{ts's'a} \geq 0, \forall (t, s', s, a) \in T \times S \times S \times A$. The initial population distribution is given by $p \in \mathbb{R}_+^S$, where p_s denotes the portion of the players' population in state s at $t = 0$.

At time t , each player incurs a cost as a function of y , $\ell_{tsa} : \mathbb{R}^{(T+1)SA} \rightarrow \mathbb{R}$. We collect ℓ_{tsa} into a cost vector $\ell(y) \in \mathbb{R}^{(T+1)SA}$ under the same ordering as y in (2).

Similar to MDP literature, a player's expected cost-to-go at (t, s, a) is its Q -value function, recursively defined as

$$Q_{tsa}(y) = \begin{cases} \ell_{tsa}(y) & t = T \\ \ell_{tsa}(y) + \sum_{s'} P_{ts's'a} \min_{a' \in [A]} Q_{t+1,s'a'}(y) & t \in [T] \end{cases} \quad (3)$$

In an MDP congestion game, the Q -value functions depend on the players' collective action choices through y , the population distribution. Each player's objective is to minimize its own Q -value function by choosing individual actions. An *MDP Wardrop equilibrium* y^* is reached if no player can unilaterally decrease its Q -value function further by changing its actions.

Definition 1 (MDP Wardrop Equilibrium [6]) A population distribution y^* is a MDP Wardrop equilibrium if for every $(t, s, a) \in [T+1] \times [S] \times [A]$,

$$y_{tsa}^* > 0 \Rightarrow Q_{tsa}(y^*) \leq Q_{tsa'}(y^*), \quad \forall a' \in [A] \quad (4)$$

The set of $y^* \in \mathcal{Y}(P, p)$ that satisfies (4) is denoted by $\mathcal{W}(\ell)$.

At an MDP Wardrop equilibrium, every positive portion of the population distribution exclusively selects actions with the lowest Q -values.

Remark 1 The player costs vary with the population distribution in MDP congestion games and with the joint policy in stochastic games [29]. Furthermore, policy and population distribution are the primal and dual variables of MDP's linear program [24, Eqn 6.9.2], respectively. Therefore, the two games can be interpreted as game extensions of the MDP using either its primal or dual form.

If ℓ is a continuous vector-valued function and there exists an explicit potential function F satisfying $\nabla F(y) = \ell(y)$, then the MDP congestion game is a *potential game* [18].

Proposition 1 [6, Thm.1.3] For a MDP congestion game with cost vector ℓ , if a potential function F satisfies

$$\nabla F(y) = \ell(y), \quad F : \mathbb{R}^{(T+1) \times [S] \times [A]} \mapsto \mathbb{R}, \quad (5)$$

then the MDP Wardrop equilibrium is given by the optimal solution of

$$\begin{aligned} \min_y & F(y) \\ \text{s.t. } & y \in \mathcal{Y}(P, p). \end{aligned} \quad (6)$$

Games of form (6) can be solved by convex optimization techniques [14]. Using the corresponding potential function, we can characterize the degree of sub-optimality for any feasible population distribution within $\mathcal{Y}(P, p_0)$.

Definition 2 (ϵ -MDP Wardrop equilibrium) For a game with the cost vector ℓ , potential function F (5), and MDP Wardrop equilibrium y^* , if for some $\epsilon > 0$, $\hat{y}(\epsilon)$ satisfies

$$F(\hat{y}(\epsilon)) \leq F(y^*) + \epsilon, \quad \hat{y}(\epsilon) \in \mathcal{Y}(P, p_0),$$

then $\hat{y}(\epsilon)$ is a ϵ -MDP Wardrop equilibrium. The set of ϵ -MDP Wardrop equilibria is given by $\mathcal{W}(\ell, \epsilon)$.

Among cost vectors ℓ that have explicit potential functions, we focus on those that are strongly convex [3, Eqn B.6].

Assumption 1 The cost function ℓ has an explicit potential F (5) that is α -strongly convex for all $y \in \mathcal{Y}(P, p)$.

$$\nabla_y \ell(y) \in \mathbb{R}^{(T+1)SA \times (T+1)SA}, \quad \nabla_y \ell(y) \succeq \alpha I, \quad \alpha > 0.$$

Assumption 1 implies congestion in *all* state-action costs. To model games in which *some* state-action costs are constant, we can approximate the constant cost by a slowly growing congestion cost.

Remark 2 If $\ell_{tsa} : \mathbb{R}_+ \mapsto \mathbb{R} \forall (t, s, a) \in [T+1] \times [S] \times [A]$ are scalar functions, then Assumption 1 implies that each ℓ_{tsa} strictly increases and satisfies $\alpha|y_{tsa} - y'_{tsa}| \leq |\ell_{tsa}(y_{tsa}) - \ell_{tsa}(y'_{tsa})|$. Its potential is also given by

$$F_0(y) = \sum_{t,s,a} \int_0^{y_{tsa}} \ell_{tsa}(u) du. \quad (7)$$

For all ϵ -MDP Wardrop equilibria $\hat{y}(\epsilon)$, Assumption 1 implies $\|\hat{y}(\epsilon) - y^*\|_2^2 \leq \frac{2\epsilon}{\alpha}$.

4 Tolling for constraint satisfaction

In this section, we formulate system-level constraints using affine population distributions inequalities and relate the inexact oracle of the tolled MDP congestion game to an ϵ -MDP Wardrop equilibrium. Affine constraints cover many design requirements for large-scale networks. For example, the Department of Transportation may meet carbon-emission targets by tolling fossil fuel vehicles on major freeways [15]. This can be expressed as an affine inequality on the fossil fuel vehicle population. In competitive electricity markets, power auction are followed by a procedure to predict and eliminate power violations by initializing offline generators [17]. To minimize the incurred cost of these initialization, the system operator may use tolls during the auction to limit voltage demands. This can again be expressed as an affine constraint on local voltages.

Definition 3 (Affine Constraints) The set of constraints on the population distribution is given by

$$\mathcal{C} = \{y \in \mathbb{R}_+^{(T+1)SA} \mid Ay - b \leq 0\} \quad (8)$$

where $A \in \mathbb{R}^{C \times (T+1)SA}$, $b \in \mathbb{R}^C$, and $0 \leq C < \infty$ denotes the total number of constraints imposed.

Let $A_i \in \mathbb{R}^{(T+1)SA}$ be the i^{th} row of the matrix A . For each constraint i , instead of searching over all possible tolls, we only consider tolls of the form $\tau_i A_i \in \mathbb{R}^{(T+1)SA}$ for $\tau_i \in \mathbb{R}_+$. This formulation ensures that τ_i only affects the (t, s, a) component of ℓ when $A_{i,tsa}$ is non-zero, where the toll magnitude is controlled by τ_i . We denote the toll-augmented game cost as

$$\ell_\tau(y) := \ell(y) + A^\top \tau, \quad \tau \in \mathbb{R}_+^C. \quad (9)$$

When ℓ satisfies Assumption 1, we denote ℓ_τ 's potential as $L(\cdot, \tau)$, such that $\nabla_y L(y, \tau) = \ell_\tau$ and L augments F (5) as

$$L(y, \tau) = F(y) + \tau^\top (Ay - b). \quad (10)$$

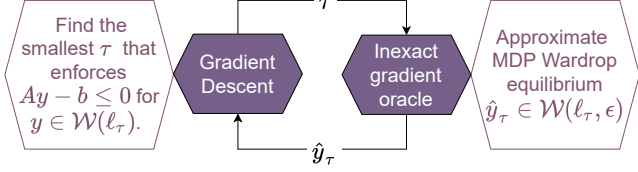


Fig. 1. Using approximate MDP Wardrop equilibrium, we perform inexact gradient descent on τ to find the minimum toll value.

For a toll value τ , the toll-augmented game and the tolled MDP Wardrop equilibrium, $y_\tau \in \mathcal{W}(\ell_\tau)$, are given by

$$d(\tau) = \min_{y \in \mathcal{Y}(P, p)} L(y, \tau), \quad y_\tau \in \operatorname{argmin}_{y \in \mathcal{Y}(P, p_0)} L(y, \tau). \quad (11)$$

Using (9), any feasible affine constraint can be satisfied for arbitrarily large magnitudes of τ [14]. We specifically want to compute the minimum toll value that enforces \mathcal{C} (8) on the MDP Wardrop equilibrium of the toll-augmented game.

Definition 4 (Minimum toll value) For a constraint set \mathcal{C} (8), the minimum toll value, $\tau^* \in \mathbb{R}_+^C$, is the minimal non-negative toll which ensures that the toll-augmented cost vector ℓ_τ (9) generates a constraint-satisfying MDP Wardrop equilibrium—i.e.,

$$\tau^* = \min \{ \tau \in \mathbb{R}_+^C \mid \mathcal{W}(\ell_\tau) \subseteq \mathcal{C} \}. \quad (12)$$

We proved a sufficient condition for the existence of the minimum toll value in [14].

Proposition 2 [14]: When \mathcal{C} is convex, $\mathcal{C} \cap \mathcal{Y}(P, p_0)$ is non-empty, and the cost vector ℓ satisfies Assumption 1, a unique minimum toll value τ^* exists that uniquely maximizes $d(\tau)$.

$$\tau^* = \operatorname{argmax}_{\tau \in \mathbb{R}_+^C} \left[\min_{y \in \mathcal{Y}(P, p_0)} L(y, \tau) \right] = \operatorname{argmax}_{\tau \in \mathbb{R}_+^C} d(\tau). \quad (13)$$

When ℓ is known, (13) directly computes τ^* . When ℓ 's is unknown, we cannot explicitly solve for either τ^* or $d(\tau)$.

Problem 1 To find τ^* when ℓ satisfies Assumption 1 and is unknown, we apply gradient descent to $d(\tau)$ by querying the ϵ -MDP Wardrop equilibria of $d(\tau)$ and forming an inexact oracle. Our set-up is summarized in Figure 1.

To demonstrate how the ϵ -MDP Wardrop equilibrium of a tolled game induces an inexact oracle for $\nabla d(\tau)$, we first derive the analytical expression of $\nabla d(\tau)$.

Proposition 3 When the congestion costs ℓ satisfy Assumption 1 and \mathcal{C} satisfies Definition 3, d from (11) has the following properties.

- d is concave.

- d is $\bar{\alpha}$ -smooth with $\bar{\alpha} = \frac{\|A\|_2^2}{\alpha}$. I.e., for any $\sigma, \tau \in \mathbb{R}^C$,

$$d(\tau) + \nabla d(\tau)^\top (\sigma - \tau) - \frac{\bar{\alpha}}{2} \|\sigma - \tau\|_2^2 \leq d(\sigma). \quad (14)$$

- Let y_τ be defined as (11), then $\nabla d(\tau)$ is given by

$$\nabla d(\tau) = Ay_\tau - b. \quad (15)$$

See Appendix A.1 for proof. When the costs ℓ are unknown, we can still obtain ϵ -approximate MDP Wardrop equilibrium via learning algorithms [33, 12, 32]. When applied to games with tolls, these ϵ -approximate MDP Wardrop equilibria form ϵ -inexact oracles of ∇d and d .

Definition 5 (ϵ -inexact oracle) The ϵ -inexact oracles of $\nabla d(\tau)$ and $d(\tau)$ are given by

$$\hat{\nabla} d(\tau) = A\hat{y}_\tau(\epsilon) - b, \quad \hat{d}(\tau) = L(\hat{y}_\tau(\epsilon), \tau), \quad (16)$$

where $\hat{y}_\tau(\epsilon) \in \mathcal{W}(\ell_\tau, \epsilon)$ is an ϵ -MDP Wardrop equilibrium satisfying

$$L(\hat{y}_\tau(\epsilon), \tau) \leq L(y_\tau, \tau) + \epsilon. \quad (17)$$

When $\epsilon = 0$, the oracle is exact. When $\epsilon > 0$, the oracle's 'inexactness' can be characterized by how much it changes the concavity and smoothness of d .

Lemma 1 (Concavity) Under Assumption 1, all ϵ -MDP Wardrop equilibria $\hat{y}_\tau(\epsilon)$ given by (17) will generate ϵ -inexact oracles (16) that satisfy

$$d(\sigma) \leq \hat{d}(\tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau), \quad \forall \sigma, \tau \in \mathbb{R}_+^C.$$

See Appendix A.2 for proof. Next, we show that \hat{d} and $\hat{\nabla} d$ preserve d 's smoothness up to 2ϵ .

Lemma 2 (ϵ -approximate smoothness) Let $\bar{\alpha} = \frac{\|A\|_2^2}{\alpha}$. Under Assumption 1, all ϵ -MDP Wardrop equilibria $\hat{y}_\tau(\epsilon)$ given by (17) will generate inexact oracles (16) that satisfy

$$\hat{d}(\tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau) - \bar{\alpha} \|\sigma - \tau\|_2^2 \leq d(\sigma) + 2\epsilon, \quad \forall \tau, \sigma \in \mathbb{R}^C. \quad (18)$$

See Appendix A.3 for proof.

5 Minimum toll algorithm

Convergence of first-order gradient methods relies on the objective's convexity and smoothness. If an inexact gradient preserves the concavity and smoothness, it can also be shown to converge [8]. In this section, we apply the same concept

Algorithm 1 Iterative toll synthesis

Input: ℓ, P, p_s, τ_0 .**Output:** τ^N, y^N .

```
1: for  $k = 0, 1, \dots$  do
2:    $y^k \in \mathcal{W}(\ell + A^\top \tau^k, \epsilon^k)$ 
3:    $\tau^{k+1} = [\tau^k + \gamma^k (Ay^k - b)]_+$ 
4: end for
```

in the context of tolling for MDP congestion games and elaborate on how constraint violation is affected by the ϵ -MDP Wardrop equilibrium.

In Algorithm 1, we denote the k^{th} toll charged, the k^{th} ϵ -MDP Wardrop equilibrium, and the ϵ in the k^{th} ϵ -inexact oracle as τ^k , y^k , and ϵ^k , respectively. When $\epsilon^k = 0 \forall k \in \mathbb{N}$, Algorithm 1 corresponds a projected gradient ascent on $d(\tau)$, for which the convergence rate is sublinear [4]. We analyze Algorithm 1's convergence properties when $\epsilon^k > 0$ by utilizing the following quantities,

$$\bar{\tau}^k = \frac{1}{k} \sum_{s=1}^k \tau^s, \quad \bar{y}^k = \frac{1}{k} \sum_{s=0}^{k-1} y^s, \quad E^k = \sum_{s=0}^{k-1} \epsilon^s, \quad (19)$$

where $\bar{\tau}^k / \bar{y}^k / E^k$ is the average toll/average ϵ -MDP Wardrop equilibrium/accumulated ϵ up to iteration k , respectively.

Theorem 1 *If the cost vector ℓ satisfies Assumption 1, and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$ for each $k \in \mathbb{N}$, then $\bar{\tau}^k$ from (19) satisfies*

$$d(\tau^*) - d(\bar{\tau}^k) \leq \frac{1}{k} \left(\frac{1}{2\gamma} \|\tau^0 - \tau^*\|_2^2 + 2E^k \right), \quad (20)$$

where τ^* is the minimum toll value (12).

PROOF. Let $r^s = \|\tau^s - \tau^*\|_2^2$. From Lemma 3, when $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, we have

$$r^{s+1} \leq \quad (21)$$

$$r^s + 2\gamma \left(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s + \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^*) \right)$$

From Lemma 1, we have

$$\hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^*) \leq L(y^s, \tau^s) - d(\tau^*) \quad (22)$$

Summing up (21) and $2\gamma \times (22)$, we obtain

$$r^{s+1} - r^s \leq 2\gamma (d(\tau^{s+1}) - d(\tau^*) + 2\epsilon^s) \quad (23)$$

Summing over (23) for $s = 0 \dots k-1$, we obtain $0 \leq r^k \leq r^0 - 2\gamma \sum_{s=1}^k (d(\tau^s) - d(\tau^*)) + 4\gamma \sum_{s=0}^{k-1} \epsilon^s$. Finally, the concavity of d from Proposition 3 implies that $-kd(\bar{\tau}^k) = -kd(\sum_{s=1}^k \tau^s) \leq -\sum_{s=1}^k d(\tau^s)$. This completes the proof.

Remark 3 When $\epsilon^k = \epsilon$ for all $k \in \mathbb{N}$, (20) becomes

$$d(\tau^*) - d(\bar{\tau}^k) \leq \frac{1}{k} \left(\frac{1}{2\gamma} \|\tau^0 - \tau^*\|_2^2 \right) + 2\epsilon.$$

We note that this convergence rate is sublinear in k , which is comparable to gradient descent with exact oracles. Additionally, the convergence rate depends linearly on the degree of sub-optimality of the oracles as 2ϵ .

Our proof is inspired by [8,19]. Theorem 1 shows that E^k 's effect on the convergence rate is independent of the chosen step-size. Furthermore, the convergence of $\bar{\tau}^k$ towards τ^* depends on the magnitude of the minimum toll value required to enforce \mathcal{C} and the accumulated error made in approximating the MDP Wardrop equilibrium. The constraint violation of the average MDP Wardrop equilibrium approximation \bar{y}^k is similarly bounded.

Corollary 1 *If the cost vector ℓ satisfies Assumption 1 and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, then the constraint violation of the average population distribution \bar{y}^k from (19) satisfies*

$$\|[A\bar{y}^k - b]_+\|_2 \leq \frac{1}{\gamma k} \left(\|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k} \right). \quad (24)$$

PROOF. We first derive an upper bound for $\|\tau^k\|_2$ and then bound the left hand side of (24) by $\|\tau^k\|_2$. Recall (23), we use $d(\tau^*) - d(\tau^k) \geq 0$ to derive $r^{s+1} \leq r^s + 4\gamma \epsilon^s$. Summing over $s = 0, \dots, k-1$, we have

$$\|\tau^k - \tau^*\|_2^2 \leq \|\tau^0 - \tau^*\|_2^2 + 4\gamma E^k. \quad (25)$$

Taking the square root of both sides of (25) and noting the identity $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain

$$\|\tau^k - \tau^*\|_2 \leq \|\tau^0 - \tau^*\|_2 + \sqrt{4\gamma E^k}. \quad (26)$$

We add $\|\tau^*\|_2$ to both sides of (26) and use the triangle inequality $\|\tau^k\|_2 \leq \|\tau^k - \tau^*\|_2 + \|\tau^*\|_2$ to obtain

$$\|\tau^k\|_2 \leq \|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + \sqrt{4\gamma E^k}. \quad (27)$$

Next, we bound $\|[A\bar{y}^k - b]_+\|_2$ using $\|\tau^k\|_2$. From line 3 of Algorithm 1, $\tau^{s+1} \geq \tau^s + \gamma(Ay^s - b)$. We sum over $s = 0, \dots, k-1$ to obtain $\tau^k \geq \tau^0 + \gamma k(A\bar{y}^k - b)$. Noting $\tau^0 \in \mathbb{R}_+^C$ can be dropped, $\gamma k[A\bar{y}^k - b]_+ \leq \tau^k$ combined with (27) completes the proof.

Remark 4 *If the oracle incurs a constant error $\epsilon^k = \epsilon$ at each iteration k , Corollary 1 shows that the average constraint violation will still asymptotically reduce to zero.*

Remark 5 As Corollary 1 is concerned with the average constraint violation, Algorithm 1 is not appropriate for enforcing safety-critical system constraints.

Corollary 1 shows that larger step sizes in Algorithm 1 can reduce the amount of average constraint violation. Larger step-sizes can also directly reduce the effect of approximation error on constraint violation by a factor of $2\sqrt{E^k\gamma^{-1}}$.

We can further show that the average population distribution \bar{y}^k (19) converges to the same optimal population distribution under the minimum toll value.

Theorem 2 *If the cost vector ℓ satisfies Assumption 1 and $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, then the average player population distribution given by \bar{y}^k (19) satisfies*

$$\|\bar{y}^k - y^*\|_2^2 \leq \frac{\alpha}{2\gamma k} D(\tau^0, \tau^*, E^k), \quad (28)$$

where τ^* is the minimum toll value, y^* is the optimal population distribution for $d(\tau^*)$, and $D(\tau^0, \tau^*, E^k)$ is given by

$$D(\tau^0, \tau^*, E^k) = \max \left\{ \frac{1}{2} \|\tau^0\|_2^2 + 2E^k, \|\tau^*\|_2^2 + \|\tau^*\|_2 \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k} \right\}. \quad (29)$$

PROOF. We first upper bound and lower bound the term $F(\bar{y}^k) - F(y^*)$. First consider the upper bound. From Lemma 3, let $\tau = 0$,

$$\|\tau^{s+1}\|_2^2 \leq \|\tau^s\|_2^2 + 2\gamma \left(d(\tau^{s+1}) + 2\epsilon^s - L(y^s, \tau^s) + \hat{\nabla} d(\tau^s)^\top \tau^s \right). \quad (30)$$

Recall from (15) and (10), $\hat{\nabla} d(\tau^s) = Ay^s - b$ and $L(y^s, \tau^s) = F(y^s) + (\tau^s)^\top (Ay^s - b)$. Therefore $L(y^s, \tau^s) - \hat{\nabla} d(\tau^s)^\top \tau^s = F(y^s)$. Then (30) becomes

$$\|\tau^{s+1}\|_2^2 + 2\gamma(F(y^s) - d(\tau^{s+1})) \leq \|\tau^s\|_2^2 + 4\gamma\epsilon^s \quad (31)$$

Summing over $s = 0, \dots, k-1$, $\sum_{s=0}^{k-1} F(y^s) - d(\tau^{s+1}) \leq \frac{1}{2\gamma} \|\tau^0\|_2^2 + 2E^k$. Taking the average \bar{y}^k and noting that $d(\tau^k) \leq d(\tau^*) = F(y^*)$ for all $\tau^k \in \mathbb{R}_+^C$,

$$F(\bar{y}^k) - F(y^*) \leq \frac{1}{2\gamma k} \|\tau^0\|_2^2 + \frac{2E^k}{k}. \quad (32)$$

Next, consider the lower bound of $F(\bar{y}^k) - F(y^*)$. By definition, y^* solves $\min_{y \in \mathcal{Y}(P, p_0)} F(y) + (Ay - b)^\top \tau^*$ where $(Ay^* - b)^\top \tau^* = 0$. This implies that $F(y^*) \leq L(\bar{y}^k, \tau^*)$. We expand $L(\bar{y}^k, \tau^*)$ with (10) to obtain

$$F(y^*) - F(\bar{y}^k) \leq (A\bar{y}^k - b)^\top \tau^* \leq [A\bar{y}^k - b]_+^\top \tau^*.$$

We can then bound the difference $F(y^*) - F(\bar{y}^k)$ by $\|\tau^*\|_2 \|[A\bar{y}^k - b]_+\|_2$. From Corollary 1,

$$F(y^*) - F(\bar{y}^k) \leq \frac{\|\tau^*\|_2}{\gamma k} \left(\|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k} \right). \quad (33)$$

Together, (32) and (33) imply

$$|F(y^*) - F(\bar{y}^k)| \leq \frac{1}{\gamma k} D(\tau^*, \tau^0, E^k) \quad (34)$$

Strong convexity of F follows from Assumption 1, such that $\|\bar{y}^k - y^*\|_2^2 \leq \frac{\alpha}{2} |F(\bar{y}^k) - F(y^*)|$. This combined with (34) completes the proof.

Remark 6 Theorems 1 and 2 show that Algorithm 1 will converge to τ^* and y^* only if E^k 's growth rate is sublinear. If the oracle incurs a constant error ϵ every iteration, then $\lim_{k \rightarrow \infty} \|\bar{\tau}^k - \tau^*\|_2 = 2\epsilon$ and $\lim_{k \rightarrow \infty} \|\bar{y}^k - y^*\|_2 = \frac{\alpha}{\gamma} \epsilon$.

As in the case of average constraint violation, we observe that the step size influences the effects of ϵ -Wardrop equilibrium on the convergence of \bar{y}^k towards y^* . If we assume that at each step k , an approximate MDP Wardrop equilibrium y^k with the same constant ϵ is returned, then the error $\bar{y}^k - y^*$ is minimized for the largest step size, $\gamma = \frac{\alpha}{\|A\|_2^2}$.

Fast first-order gradient method. An alternative to Algorithm 1 is the fast gradient method [8]. When assuming $\epsilon^k = \epsilon$ for all $k \in \mathbb{N}$, the fast gradient method is equivalent to appending to Algorithm 1 the following after Step 3,

$$\tau^{k+1} = \frac{\|A\|_2^2}{\alpha(k+3)} \left[\sum_{i=1}^{k+1} \sqrt{i(i+1)} (A\hat{y}^i - b) \right]_+ + \frac{k+1}{k+3} \tau^{k+1}.$$

In large networked systems where the inexact oracle may have low accuracy, the fast gradient method theoretically and empirically diverges from the optimal objective, $d(\tau^*)$ [8]. It has also been observed that the fast gradient method's constraint violation is comparable to Algorithm 1 [19]. Thus, we focus on the classical first-order gradient method instead.

6 Congestion Reduction in Ride-share Networks

In this section, we model the competition among NYC's ride-share drivers as a MDP congestion game and apply Algorithm 1 to demonstrate how ride-share companies can implicitly enforce constraints by utilizing tolls. Since origin-destination-specific trip data for ride-share companies are not publicly available, we use the rider demand distribution provided by the NYC TLC as a proxy for Uber's rider demand distribution. In [28], the overall rider demand for TLC is estimated to be about 40% of the rider demand for Uber. We normalize the TLC rider demand distribution accordingly and assume that the TLC data is a proportionally accurate estimate of the rider demand for Uber.

6.1 Ride-sharing MDP

We consider a cohort of competitive ride-share drivers in Manhattan, NYC who repeatedly operate between 9 am and noon. Using over six hundred thousand data points from the yellow taxi data during January, 2019 [21], we model individual driver dynamics as a finite time horizon MDP.

Modelling assumptions. 1) In modelling the ride-share competition as a finite horizon MDP congestion game, we assume that all trips take identical time regardless of destination or congestion level. 2) The results we obtain are for an uniform initial driver distribution, i.e. $p_s = 10000/63$ for each $s \in [S]$. We found that varying the initial distribution did not significantly impact the time-averaged MDP Wardrop equilibrium or the approximate norm of the resulting tolls, as long the initial distribution does not violate constraints. 3) From [16], the Uber driver population in NYC is approximately 50000. We assume that 20% of the total population works in Manhattan between 9 am and noon.

States. As shown in Figure 2, Manhattan is discretized into 63 states that correspond to TLC's taxi zones (excluding the island zones). For state s , its neighbor states are those geographically adjacent (sharing one or more edges) to s in Figure 2. The set of neighbors is denoted by $\mathcal{N}(s)$.

Actions. Actions are state-specific. At $s \in [S]$, two types of actions are available to drivers. Action $a_{s'}$ for any $s' \in \mathcal{N}(s)$ transitions the driver from s to s' . Action a_s is taken when the driver stays in the current state s and attempts to pick up a rider. For all states, the total number of available actions is given by

$$A = 1 + \max_s |\mathcal{N}(s)|.$$

where we use $|\mathcal{N}(s)|$ to denote the number of neighbors that state s has. Each action $a_i \in [A]$ corresponds to different actions for different states. In states with less neighbors than $\max_s |\mathcal{N}(s)|$, the extra actions are repeats of previous actions. For example, for state 1 with neighbors $\{3, 4\}$, the available actions are $\{a_1, a_3, a_4, a_3, a_4\}$. For state 2 with neighbors $\{3, 4, 5, 6\}$, the available actions are $\{a_2, a_3, a_4, a_5, a_6\}$.

Time. The average trip time from the TLC data is 12.02 minutes. Therefore, we take 12 minutes time intervals between 9 am and noon for a total of $T = 15$ time steps.

Transition Dynamics. For action $a_{s'}$ from state s at time t , the transition probability to any state \bar{s} is given by

$$P(s, a_{s'}, \bar{s}, t) = \begin{cases} 1 - \delta, & \text{if } \bar{s} = s', \\ \frac{\delta}{|\mathcal{N}(s)|-1}, & \text{if } \bar{s} \neq s', \bar{s} \in \mathcal{N}(s), \\ 0, & \text{otherwise,} \end{cases}$$

where $\delta \in [0, 1)$ models the driver's probability of real-time deviation from a chosen strategy. We set $\delta = 0.1$.

For action a_s from state s at time t , the transition probabilities to any state \bar{s} is derived using the rider demand distribution at s . Let $N(s, \bar{s}, t)$ be the number of riders whose trip starts at s and ends at \bar{s} during time step t . The transition probability to state \bar{s} is given by

$$P(\bar{s}, s, a_s, t) = \frac{N(s, \bar{s}, t)}{\sum_{s'} N(s, s', t)}, \quad \forall \bar{s} \in [S], t \in [T].$$

Note that a_s may transition drivers to states outside of $\mathcal{N}(s)$.

Driver costs. As in [14], the driver cost is the sum of expected earnings, fuel cost, and an artificial congestion cost.

$$\begin{aligned} \ell_{tsa}(y_{tsa}) &= \mathbb{E}_{s'} [c_{ts's}^{\text{trav}} - m_{ts's}] + c_t^{\text{wait}} \cdot y_{tsa} \\ &= \sum_{s'} P_{ts'sa} [c_{ts's}^{\text{trav}} - m_{ts's}] + c_t^{\text{wait}} \cdot y_{tsa} \end{aligned}$$

where $m_{ts's}$ is the monetary reward that the driver receives for transitioning from state s to s' (only available when $a = a_s$), $c_{ts's}^{\text{trav}}$ is the fuel cost of travelling from state s to s' , c_t^{wait} is the coefficient of congestion, scaled linearly by the portion of drivers who are waiting for a rider. For a trip time of Δt , the cost of UberX in NYC [1] is

$$m_{ts's} = \max \left(\$7, \$2.55 + \$0.35 \cdot \Delta t(\text{min}) + \$1.75 \cdot \Delta d(\text{mi}) \right), \quad (35)$$

where a base price of 7 is applied to each trip, augmented by charges proportional to the trip time and the trip distance. To compute $m_{ts's}$, we assign $\Delta t = 12$ minutes to be the average trip time, and $\Delta d = d_{ss'}$ to be the geographical distance between s and s' in miles. The other parameters in ℓ_{tsa} are computed as

$$c_{ts's}^{\text{trav}} = \underbrace{\mu}_{\text{mi}} \underbrace{d_{ss'}}_{\text{mi}} \underbrace{(\text{Vel})^{-1}}_{\text{hr/mi}} + \underbrace{\left(\frac{\text{Fuel}}{\text{Price}} \right)}_{\text{\$/gal}} \underbrace{\left(\frac{\text{Fuel}}{\text{Eff}} \right)^{-1}}_{\text{gal/mi}} \underbrace{d_{ss'}}_{\text{mi}} \quad (36a)$$

$$c_{tsa}^{\text{wait}} = \begin{cases} \mathbb{E}_{s'} [m_{ts's}] \cdot \left(\frac{\text{Customer Demand Rate}}{\text{rides}/\Delta t} \right)^{-1}, & \text{if } a = a_s \\ \omega_{tsa_{s'}}, & \text{if } a = a'_{s'} \end{cases} \quad (36b)$$

where μ is a time-money tradeoff parameter, $\omega_{tsa_{s'}} = 0.1$ models the minor congestion effects for Uber drivers who decide to traverse from s to s' .

The distance between states $d_{ss'}$ is the Haversine distance between s and s' . If $s' = s$, we average existing TLC data for trips that start and end in the same state to estimate $d_{ss'}$.

The customer demand rate is derived from TLC data per time interval per day. We assume that the TLC data is proportionally a reasonable representation of ride demands in Manhattan. From [5], the Yellow Taxi's ride demand is about 40% of the Uber's ride demand at the beginning of 2019. Therefore we scale the TLC ride demand data by 2.5.

Other parameter values are listed in Table 1.

μ	Velocity	Fuel Price	Fuel Eff
\$15 /mi	8 mph	\$2.5/gal	20 mi/gal

Table 1
Parameters for the driver cost function.

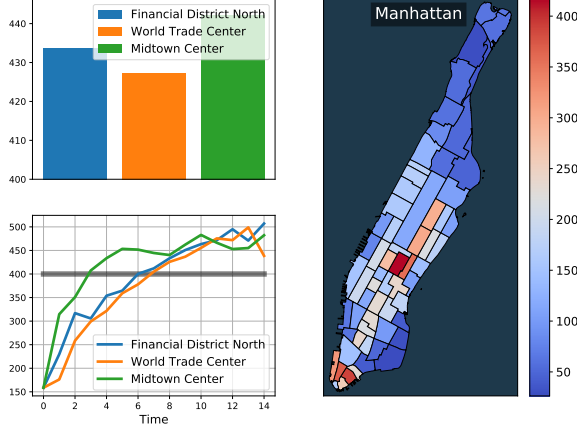


Fig. 2. Predicted ride-share traffic in Manhattan.

6.2 Online learning via conditional gradient descent

In the previous section, we built a MDP congestion game model for the purpose of simulating Manhattan’s ride-share game. We assume that the players cannot access the congestion costs model and transition dynamics. Instead, the drivers repeatedly receive costs for a chosen joint policy, and collectively learn the MDP Wardrop equilibrium.

We implement the learning method from [14, Alg.3]. Inspired by conditional gradient descent (Frank-Wolfe), this method implicitly enforces $y \in \mathcal{Y}(P, p_0)$ by solving the linearized objectives via dynamic programming. Although conditional gradient descent can be slow to converge to higher accuracy, it converges faster to lower accuracy solutions. Using the stopping criterion for conditional gradient descent, ϵ^k -MDP Wardrop equilibrium is achieved when the driver costs and population distribution satisfy

$$\epsilon^k = (\ell(y^k) + A^\top \tau^k)^\top (y^k - y^{k+1}). \quad (37)$$

We set $\epsilon^k = 2e^3$, which is approximately equivalent to a normalized error of 1% for the game potential F_0 (7). The corresponding ϵ -MDP Wardrop equilibrium as well as the time-averaged/varying driver densities of the most congested states are shown in the map/line/bars plot in Figure 2.

From Figure 2, we see that the three most congested zones are the red zones on the bottom-left and center of Manhattan. Interestingly, our simulation shows that they attract 13%

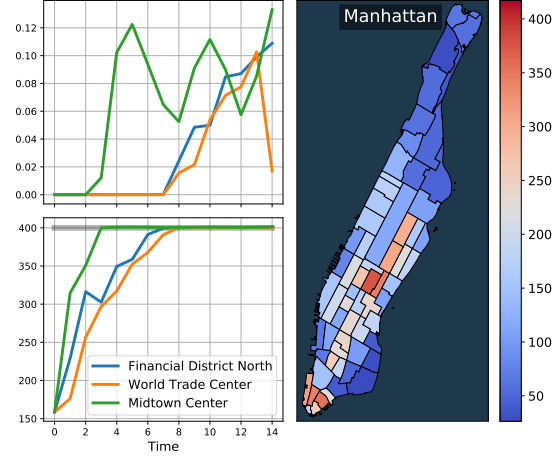


Fig. 3. Manhattan game with constraints (38) enforced. The plot labeled Manhattan shows the time-averaged driver distribution after tolling. The top line plot shows the time-varying toll as a function of time for each constraint-violating state, toll unit is \$. The bottom line plot shows the corresponding driver distribution after 2000 iterations.

of the working driver population. This further motivates the need to reduce congestion within ride-share drivers.

6.3 Reducing driver presence in congested taxi zones

If ride-share companies iteratively impose tolls according to Algorithm 1, they can reduce the driver presence in the congested taxi zones without constraining driver behaviour or significantly impacting driver earnings.

Suppose the ride-share company wishes to limit the driver count to no more than 400 per zone per time. This constraint can be formulated as

$$\sum_a y_{tsa} \leq 400, \quad \forall (t, s) \in [T+1] \times [S]. \quad (38)$$

For each (t, s) , the corresponding constraint is given by $A_i \in \mathbb{R}^{(T+1)SA}$, where the $(t, s, a)^{th}$ entry is 1 and all other entries are 0. Although only three states violate the capacity constraint in Figure 2, we toll all states at all times to ensure that no states will violate constraints in the resulting equilibria. Thus, we enforce a total of $63 \times 16 = 1008$ constraints of form (38). The constraint matrix is $A = [A_1, \dots, A_{(T+1)S}]^\top \in \mathbb{R}^{(T+1)S \times (T+1)SA}$.

6.4 Discussion

We run Algorithm 1 for 500 iterations. The results are shown in Figure 3. When ϵ^k is about 1% of the unconstrained optimal potential value, the resulting constraint violation is quite low, with less than 5 driver violations per time step. As shown in the upper left plot of Figure 3, the tolls required to enforce these constraints do not extend beyond \$0.15 per time step for all three states.

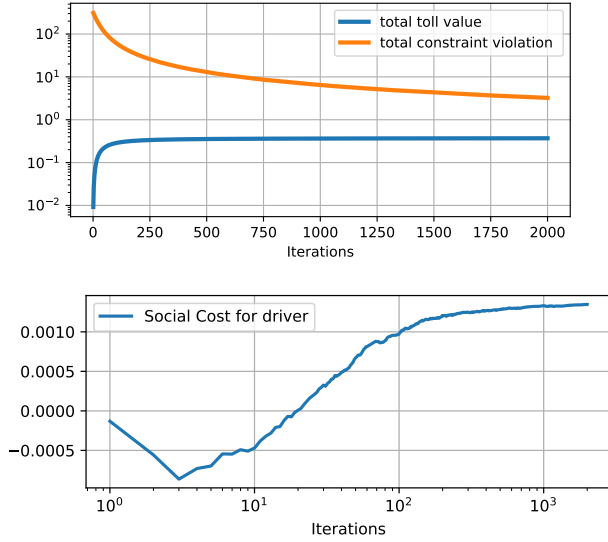


Fig. 4. Top: the total toll enforced and the total constraint violation per iteration of toll synthesis. Bottom: the average driver cost per iteration of toll synthesis, scaled by the nominal average driver cost without toll.

From Figure. 4, we conclude that for a total of less than \$1 toll over all states per time step, we can decrease the total constraint violation from over 200 drivers to less than 5 drivers over the entire time horizon. Furthermore, our tolls amounts to \$3.25 for spending 4 hours in the most congested state; this is comparable in value to the current toll proposed for lower Manhattan (\$2.25 per entry) [9].

A major concern for the ride-share company is the effect of tolls on the average driver earning. If the average driver earning decreases significantly under tolls, then drivers may choose to quit, leading to less workforce for the ride-share company. In general, tolling does not necessarily cause average player earning to decrease [14]. This is because the average player earning can be viewed as the *socially optimal* population distribution, which minimizes a different optimization objective than the game’s potential (6). Here, we can demonstrate empirically that the tolls do not significantly detract from driver earnings. In Figure 4, we plotted the average driver cost over the tolling iteration process, normalized by the unconstrained average cost. We observe that the average cost of playing the game changes by less than 0.1% with the changing tolls. Therefore tolling will not promote quitting among the profit-driven drivers under this constraint satisfaction scenario.

Next, we investigate the effects of ϵ^k on both the tolling value and average constraint violation. In Figure 5, we compare the total average tolling value $\mathbf{1}^\top \bar{\tau}^k$ after $k = 500$ iterations for ϵ inexact oracles values $\epsilon = [1e3, 2e3, 2e4, 1e5]$ and the average constraint violation $\mathbf{1}^\top [A\bar{y}^k - b]_+$ after $k = 500$ iterations. We see that increased accuracy in the ϵ oracle decreases both the tolling value and the constraint violation level during the toll iteration process, thus providing more

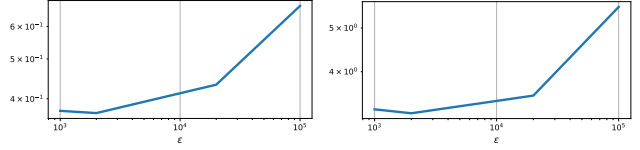


Fig. 5. Left: total average toll. Right: total constraint violation of average driver distribution. Both figures are obtained for $k = 500$ iterations as a function of ϵ inexact oracles.

incentive to find the minimum toll value to higher accuracy.

7 Conclusion

We presented an iterative tolling method that allows system-level operators to enforce constraints on a MDP congestion game with unknown congestion costs. We showed that an ϵ -MDP Wardrop equilibrium corresponds to an inexact gradient oracle of the tolled game, and derived conditions for convergence of the inexact gradient descent problem. We applied our results the ride-share system in Manhattan, NYC. Future extensions to this work include extending the toll synthesis method to work for general convex constraints.

References

- [1] Alvia. Uber new york. <http://www.alvia.com/uber-city/uber-new-york/>, 2021. Accessed: 2021-02-14.
- [2] Martin Beckmann. A continuous model of transportation. *Econometrica*, pages 643–660, 1952.
- [3] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific Belmont, 1999.
- [4] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.
- [5] Nicu Calcea. Nycdot’s experience with big data and use in transportation projects. <https://citymonitor.ai/transport/uber-lyft-rides-during-coronavirus-pandemic-taxi-data-5232>, 2017. Accessed: 2021-02-14.
- [6] Dan Calderone and S Shankar Sastry. Markov decision process routing games. In *Proc. Int. Conf. Cyber-Physical Syst.*, pages 273–279. ACM, 2017.
- [7] Nazlı Demir, Utku Eren, and Behçet Açıkmeşe. Decentralized probabilistic density control of autonomous swarms with safety constraints. *Autonomous Robots*, 39(4):537–554, 2015.
- [8] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.
- [9] Lauren Aratani Erin Durkin. New york becomes first city in us to approve congestion pricing. <https://www.theguardian.com/us-news/2019/apr/01/new-york-congestion-pricing-manhattan>. Accessed: 2021-02-14.
- [10] S Rasoul Etesami, Walid Saad, Narayan B Mandayam, and H Vincent Poor. Smart routing of electric vehicles for load balancing in smart grids. *Automatica*, 120:109148, 2020.
- [11] Mahyar Fazlyab, Santiago Paternain, Alejandro Ribeiro, and Victor M Preciado. Distributed smooth and strongly convex optimization with inexact dual methods. In *2018 Annual American Control Conference (ACC)*, pages 3768–3773. IEEE, 2018.

- [12] W Krichene, B Drighès, and A Bayen. Online Learning of Nash Equilibria in Congestion Games. *SIAM J. Control Optim.*, 53(2):1056–1081, 2015.
- [13] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japan J. Math.*, 2(1):229–260, 2007.
- [14] Sarah HQ Li, Yue Yu, Daniel Calderone, Lillian Ratliff, and Behçet Açıkmeşe. Tolling for constraint satisfaction in markov decision process congestion games. *arXiv preprint arXiv:1903.00747*, 2019.
- [15] X Lin. Environmental constraints in urban traffic management: Traffic impacts and an optimal control framework. 2018.
- [16] Aarian Marshall. New york city flexes again, extending cap on uber and lyft. <https://www.wired.com/story/new-york-city-flexes-extending-cap-uber-lyft/>. Accessed: 2021-02-14.
- [17] Enrique Lobato Miguélez, Luis Rouco Rodríguez, TGS Roman, FM Echavarren Cerezo, Ma Isabel Navarrete Fernández, Rosa Casanova Lafarga, and Gerardo López Camino. A practical approach to solve power system constraints with application to the spanish electricity market. *IEEE Transactions on Power Systems*, 19(4):2029–2037, 2004.
- [18] Dov Monderer and Lloyd S Shapley. Potential games. *Games Economic Behavior*, 14(1):124–143, 1996.
- [19] Ion Necoara and Valentin Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2013.
- [20] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.
- [21] City of New York. Tlc trip record data. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. Accessed: 2021-02-14.
- [22] Dmitrii M Ostrovskii, Andrew Lowy, and Meisam Razaviyayn. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems. *SIAM Journal on Optimization*, 31(4):2508–2538, 2021.
- [23] Michael Patriksson. *The Traffic Assignment Problem: Models and Methods*. Courier Dover Publications, 2015.
- [24] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
- [25] Lillian J Ratliff and Tanner Fiez. Adaptive incentive design. *arXiv preprint arXiv:1806.05749 [cs.GT]*, 2018.
- [26] Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
- [27] Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *Proc. Ann. ACM Symp. Theory Comput.*, pages 949–962. ACM, 2016.
- [28] Todd Schneider. Taxi and ridehailing usage in new york city. <https://toddschneider.com/dashboards/nyc-taxi-ridehailing-uber-lyft-data/>, 2021. Accessed: 2021-02-14.
- [29] Lloyd S Shapley. Stochastic games. *Proc. Nat. Acad. Sci.*, 39(10):1095–1100, 1953.
- [30] Chaitanya Swamy. The effectiveness of stackelberg strategies and tolls for network congestion games. In *Proc. ACM-SIAM Symp. Discrete Algorithms*, pages 1133–1142. Society for industrial and applied mathematics, 2007.
- [31] Uber. How surge pricing works. <https://www.uber.com/us/en/drive/driver-app/how-surge-works/>, 2021. Accessed: 2021-04-27.
- [32] Yue Yu, Dan Calderone, Sarah HQ Li, Lillian J Ratliff, and Behçet Açıkmeşe. A primal-dual approach to markovian network optimization. *arXiv preprint arXiv:1901.08731[math.OC]*, 2019.
- [33] Bo Zhou, Qiankun Song, Zhenjiang Zhao, and Tangzhi Liu. A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. *Applied Mathematics and Computation*, 371:124895, 2020.

A Appendix

A.1 Proof of Proposition 3

PROOF. $d(\tau)$ is the dual function of the optimization problem

$$\min_{y \in \mathcal{Y}(P, p_0)} F(x) \text{ s.t. } Ay \leq b.$$

As the dual function of a convex optimization with linear constraints, it is concave [3, Prop 5.1.2]. The smoothness constant of $d(\tau)$ follows from [20, Thm 1], where α is the strong convexity factor of F_0 . Finally, the computation of $\nabla d(\tau)$ follows directly from [3, Prop.B.25].

A.2 Proof of Lemma 1

PROOF. We denote $\hat{y}_\tau(\epsilon)$ by \hat{y}_τ for simplicity. Since $\hat{y}_\tau \in \mathcal{W}(\ell_\tau, \epsilon) \subset \mathcal{Y}(P, p_0)$, using (11) we can show that

$$d(\sigma) \leq L(\hat{y}_\tau, \sigma). \quad (\text{A.1})$$

Combining (A.1) with the fact that $L(\hat{y}_\tau, \sigma) = L(\hat{y}_\tau, \tau) + \hat{\nabla} d(\tau)^\top (\sigma - \tau)$, we obtain Lemma 1.

A.3 Proof of Lemma 2

PROOF. We denote $\hat{y}_\tau(\epsilon)$ by \hat{y}_τ for simplicity and recall y_τ from (11). From Proposition 3, we know that

$$\nabla d(\tau) = \hat{\nabla} d(\tau) + A(y_\tau - \hat{y}_\tau). \quad (\text{A.2})$$

Substituting (A.2) into (14), we obtain the following

$$0 \leq d(\sigma) - d(\tau) - \hat{\nabla} d(\tau)^\top (\sigma - \tau) + \frac{\bar{\alpha}}{2} \|\sigma - \tau\|_2^2 - (A(y_\tau - \hat{y}_\tau))^\top (\sigma - \tau). \quad (\text{A.3})$$

Furthermore, we can show

$$\begin{aligned} \left| (A(y_\tau - \hat{y}_\tau))^\top (\sigma - \tau) \right| &\leq \|\hat{y}_\tau - y_\tau\|_2 \cdot \|A\|_2 \cdot \|\sigma - \tau\|_2 \\ &\leq \frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 + \frac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2, \end{aligned} \quad (\text{A.4})$$

where the first step is due to the Cauchy–Schwarz inequality, and the second step is due to the inequality of arithmetic and geometric inequalities.

Next, we note that F (5) and subsequently $L(y, \tau)$ (10) is strongly convex under Assumption 1. We combine this with the fact that $L(y_\tau, \tau) = d(\tau)$ from (11) to obtain

$$\frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 \leq L(\hat{y}_\tau, \tau) - d(\tau). \quad (\text{A.5})$$

From (17), \hat{y}_τ satisfies

$$L(\hat{y}_\tau, \tau) - d(\tau) \leq \epsilon. \quad (\text{A.6})$$

Summing up (17), (A.3), (A.4), (A.5), and $2 \times (\text{A.6})$, we obtain (18), which completes the proof.

A.4 Lemma 3

Lemma 3 *Under Assumption 1, if $\gamma \leq \frac{1}{2\bar{\alpha}}$, τ^s from Algorithm 1 satisfy*

$$\begin{aligned} \|\tau^{s+1} - \tau\|_2^2 &\leq \|\tau^s - \tau\|_2^2 + 2\gamma \left(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s \right. \\ &\quad \left. + \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau) \right), \quad \forall \tau \in \mathbb{R}_+^C, k \geq 0. \end{aligned} \quad (\text{A.7})$$

PROOF. Given $\tau \in \mathbb{R}_+^C$, let $r^s = \|\tau^s - \tau\|_2^2$. We compute $r^{s+1} - r^s$ using the law of cosine as

$$r^{s+1} - r^s = 2(\tau^{s+1} - \tau^s)^\top (\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2. \quad (\text{A.8})$$

From line 3 of Algorithm 1, $\tau^{s+1} = [\tau^s + \gamma(Ay^s - b)]_+$. Using [4, Lem 3.1], the projection onto \mathbb{R}_+^C implies that

$$0 \leq (\tau^s + \gamma \hat{\nabla} d(\tau^s) - \tau^{s+1})^\top (\tau^{s+1} - \tau) \quad (\text{A.9})$$

From (A.9), we can upper bound $(\tau^{s+1} - \tau^s)^\top (\tau^{s+1} - \tau)$ and combine with (A.8) to obtain

$$r^{s+1} - r^s \leq 2\gamma \hat{\nabla} d(\tau^s)^\top (\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2 \quad (\text{A.10})$$

From Lemma 2, we recall

$$\begin{aligned} &L(y^s, \tau^s) - d(\tau^{s+1}) - 2\epsilon^s \\ &\leq \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^{s+1}) + \bar{\alpha} \|\tau^{s+1} - \tau^s\|_2^2. \end{aligned} \quad (\text{A.11})$$

We can then combine (A.10) and $2\gamma \times (\text{A.11})$ to derive

$$\begin{aligned} &r^{s+1} - r^s + 2\gamma(L(y^s, \tau^s) - d(\tau^{s+1}) - 2\epsilon^s) \\ &\leq 2\gamma \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau) + (2\gamma\bar{\alpha} - 1) \|\tau^{s+1} - \tau^s\|_2^2, \end{aligned} \quad (\text{A.12})$$

and use the fact that $2\gamma\bar{\alpha} \leq 1$ to eliminate the $\|\tau^{s+1} - \tau^s\|_2^2$ term and complete the proof.