

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

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

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

^bOden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, USA. (e-mail: yueyu@utexas.edu).

^cDepartment of Aeronautics and Astronautics, Purdue University, USA. (e-mail: nmiguel@purdue.edu)

^dDepartment of Electrical Engineering, University of Washington, Seattle, USA. (e-mail: {djcal, ratliff}@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, computing 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 construct a congestion game model for Uber drivers in Manhattan, New York City (NYC) using data from the Taxi and Limousine Commission (TLC) to 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. This is motivated by a plethora of constrained large-scale network systems such as autonomous swarms [7], urban transportation [15] and competitive electricity markets [17]. For example, the Department of Transportation could meet carbon-emission targets by tolling fossil fuel vehicles on

freeways [15]. In electricity markets, power auctions are followed by a procedure to predict and eliminate power violations through initializing offline generators [17]. To minimize the incurred initialization cost, the system operator can use tolls during the auction to limit voltage demands. Since tolling is a common and easily implementable mechanism in networked systems [31], we assume the system operator can freely impose tolls on the players.

The *minimum toll value* for satisfying constraints can be computed as a function of the congestion costs [14]. However, extracting the congestion cost is difficult when the player objectives are complex and unknown. This is also true in simulation engines and higher complexity models, where the effect of a given toll can be computed, but not the minimum toll value itself. As such, we assume that an oracle exists who can compute the ϵ -optimal solution for a game with a known toll. The inexact oracle is motivated by model-free learning 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 an ϵ -optimal population distribution 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 an MDP congestion game model using real-world data from NYC TLC [21] to demonstrate how our algorithm reduces Uber driver congestion in Manhattan.

The rest of this paper is organized as follows. In Section 2, we review related works. In Section 3, we outline MDP congestion games. In Section 4, we formulate the toll-augmented game and the inexact oracle. In Section 5, we present the tolling algorithm and prove its convergence properties. In Section 6, we create a queuing network MDP 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 the player costs differ and are functions of the joint policy. MDP congestion games assume that the player 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 game incentive design 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 that guarantees *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 under MDP dynamics with unknown congestion costs is the topic of this paper.

Our minimum toll computation algorithm is an inexact gradi-

ent descent [8], and has been applied to settings such as distributed optimization and model predictive control [11,19]. In 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 has yet to be applied to constraint satisfaction with 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, an individual player’s probability distribution belongs to the zero measure subset of the population distribution (see [12, Sec.2] for details). Presently, we only deal with the population distribution. We denote the set of feasible population distributions as $\mathcal{Y}(P, p)$, given by

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

where $y_{t,sa}$ 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

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

The 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 $\sum_{s' \in [S]} P_{ts's'a} = 1$, 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 playing population in state s at $t = 0$.

At time t , each player incurs a cost as a function of y , $\ell_{t,sa} : \mathbb{R}^{(T+1)SA} \rightarrow \mathbb{R}$. We collect $\ell_{t,sa}$ 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

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

In an MDP congestion game, the Q -value functions depend on the players’ collective action choices through y , the population distribution. Each player minimizes 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 $(t, s, a) \in [T+1] \times [S] \times [A]$,

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

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

At y^* (4), all positive portions of the playing population distribution exclusively take actions with the lowest Q-values.

Remark 1 MDP congestion games and stochastic games are multi-player extensions of the MDP via its linear program formulation. The coupling quantities between players, the population distribution in MDP congestion games and the joint policy in stochastic games [29], are the primal and dual variables of the MDP [24, Eqn 6.9.2], respectively.

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] Given the MDP congestion game 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

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

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

Definition 2 (ϵ -MDP Wardrop equilibrium) For a game with the cost vector ℓ , potential function F (5), MDP Wardrop equilibrium y^* (4), and $\epsilon > 0$, the set of ϵ -MDP Wardrop equilibria is given by

$$\mathcal{W}(\ell, \epsilon) := \{\hat{y}(\epsilon) \in \mathcal{Y}(P, p) \mid F(\hat{y}(\epsilon)) \leq F(y^*) + \epsilon\}. \quad (7)$$

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

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

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 costs by increasing functions with infinitesimal growth rates.

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

$$F_0(y) = \sum_{t,s,a} \int_0^{y_{t_{sa}}} \ell_{t_{sa}}(u) du. \quad (8)$$

For an 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. As discussed in the introduction, meeting carbon emission goals in transportation and minimizing generator initialization costs in power grids are affine constraints on the fossil fuel vehicle population and local grid voltages, respectively.

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

$$\mathcal{C} = \{y \in \mathbb{R}_+^{(T+1)SA} \mid Ay - b \leq 0\} \quad (9)$$

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 A . 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 $A_{i,t_{sa}}$ is non-zero, where the toll magnitude is controlled by τ_i . We denote the toll-augmented game cost vector as

$$\ell_\tau(y) := \ell(y) + A^\top \tau, \quad \tau \in \mathbb{R}_+^C. \quad (10)$$

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

$$L(y, \tau) = F(y) + \tau^\top (Ay - b). \quad (11)$$

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

$$d(\tau) = \min_{y \in \mathcal{Y}(P, p)} L(y, \tau), \quad y_\tau \in \operatorname{argmin}_{y \in \mathcal{Y}(P, p)} L(y, \tau). \quad (12)$$

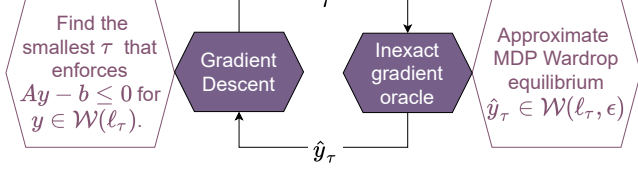


Fig. 1. Using approximate MDP Wardrop equilibrium, we perform inexact gradient descent on τ to find the minimum toll value.

Under cost vector (10), any feasible affine population constraint will hold for large values of τ [14]. We specifically want to compute the minimum toll value to enforce \mathcal{C} (9) on the MDP Wardrop equilibrium of the toll-augmented game.

Definition 4 (Minimum toll value) *Given a constraint set \mathcal{C} (9), the minimum toll value $\tau^* \in \mathbb{R}_+^C$ is the smallest non-negative toll which ensures that the MDP congestion game has constraint-satisfying MDP Wardrop equilibria—i.e.,*

$$\tau^* = \min \{ \tau \in \mathbb{R}_+^C \mid \mathcal{W}(\ell_\tau) \subseteq \mathcal{C} \}. \quad (13)$$

The minimum toll value exists under the following sufficient condition [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^* (13) maximizes $d(\tau)$.*

$$\tau^* = \operatorname{argmax}_{\tau \in \mathbb{R}_+^C} \left[\min_{y \in \mathcal{Y}(P, p)} L(y, \tau) \right] = \operatorname{argmax}_{\tau \in \mathbb{R}_+^C} d(\tau). \quad (14)$$

When ℓ is known, (14) directly computes τ^* . When ℓ is unknown, we cannot explicitly solve for either τ^* or $d(\tau)$.

Problem 1 *For MDP congestion games with unknown but strictly increasing congestion costs, find the minimum toll value τ^* (13) that ensures the resulting MDP Wardrop equilibrium y_{τ^*} (12) satisfies the desired affine constraints \mathcal{C} (9).*

As summarized in Figure 1, we compute τ^* by querying the ϵ -MDP Wardrop equilibria of $d(\tau)$ and forming an inexact oracle gradient descent. To see 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 *If the cost vector ℓ satisfies Assumption 1 and \mathcal{C} satisfies Definition 3, d (12) has the following properties.*

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

$$d(\tau) + \nabla d(\tau)^\top (\sigma - \tau) - \frac{\bar{\alpha}}{2} \|\sigma - \tau\|_2^2 \leq d(\sigma). \quad (15)$$

- Let y_τ be defined as (12), then $\nabla d(\tau)$ is given by

$$\nabla d(\tau) = Ay_\tau - b. \quad (16)$$

See App.1.1.1 for proof. When the costs ℓ are unknown, we can compute the ϵ -MDP Wardrop equilibrium via learning algorithms [33,12,32]. When applied to tolled games, these ϵ -MDP Wardrop equilibria form ϵ -inexact oracles of d .

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

$$\hat{\nabla} d(\tau) = A\hat{y}_\tau(\epsilon) - b, \quad \hat{d}(\tau) = L(\hat{y}_\tau(\epsilon), \tau), \quad (17)$$

where $\hat{y}_\tau(\epsilon) \in \mathcal{W}(\ell_\tau, \epsilon)$ (7) is an ϵ -MDP Wardrop equilibrium satisfying

$$L(\hat{y}_\tau(\epsilon), \tau) \leq L(y_\tau, \tau) + \epsilon. \quad (18)$$

When $\epsilon = 0$, the oracle is exact. When $\epsilon > 0$, the oracle's accuracy directly affects the concavity and smoothness of d (See App.1).

5 Minimum toll algorithm

Convergence of first-order gradient methods relies on the objective's convexity and smoothness. If an inexact gradient preserves concavity and smoothness, its gradient descent will also converge [8]. In this section, we apply the same concept to tolling in MDP congestion games and analyze on how constraint violation is affected by the ϵ -MDP Wardrop equilibrium.

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 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 is a projected gradient ascent on $d(\tau)$ with sub-linear convergence rates [4]. We analyze Algorithm 1's convergence when $\epsilon^k > 0$ by through 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 (13). and E^k (19) is the total approximation error.

See App.3 for proof. Our proof is inspired by [8,19].

Remark 3 When $\epsilon^k = \epsilon$ is constant, (20) becomes $d(\tau^*) - d(\bar{\tau}^k) \leq \frac{1}{k} (\frac{1}{2\gamma} \|\tau^0 - \tau^*\|_2^2) + 2\epsilon$. Similar to exact gradient descent, $\frac{1}{2\gamma} \|\tau^0 - \tau^*\|_2^2$ converges sublinearly in k . However, the term $2\epsilon > 0$ causes a constant convergence error.

Constraint violation of \bar{y}^k (19) 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

$$\|[A\bar{y}^k - b]_+\|_2 \leq \frac{1}{\gamma k} \left(\|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k} \right). \quad (21)$$

See App.4 for proof.

Remark 4 With a constant error oracle $\epsilon^k = \epsilon$, the average constraint violation will still asymptotically reduce to zero.

Unlike $\bar{\tau}^k$, E^k 's effect on the average constraint violation can be reduced with larger step sizes as $2\sqrt{E^k \gamma^{-1}}$. We note that Corollary 1 shows that Algorithm 1 is not appropriate for enforcing safety-critical system constraints.

Algorithm 1 also ensures that the average population distribution \bar{y}^k (19) converges to the optimal equilibrium for τ^* .

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

$$\|\bar{y}^k - y^*\|_2^2 \leq \frac{\alpha}{2\gamma k} D(\tau^0, \tau^*, E^k), \quad (22)$$

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

$$D(\tau^0, \tau^*, E^k) = \max \left\{ \frac{1}{2} \|\tau^0\|_2^2 + 2E^k, \|\tau^*\|_2^2 + \|\tau^*\|_2 \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k} \right\}. \quad (23)$$

See App.2 for proof.

Remark 5 Similar to $\bar{\tau}^k$, convergence of \bar{y}^k to y^* is sublinear in k and scales linearly with E^k . However, convergence error due to E^k can be minimized by taking larger stepsizes.

Fast first-order gradient method. When $\epsilon^k = \epsilon$ for all $k \in \mathbb{N}$, the fast gradient method [8] augments Algorithm 1

with the following update after Step 3,

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

In large networked systems with low-accuracy inexact oracles, the fast gradient method theoretically and empirically diverges from $d(\tau^*)$ [8]. Since its constraint violation results are comparable to Algorithm 1 [19], we focus on the standard first-order gradient descent instead.

6 Congestion Reduction in Ride-share Networks

In this section, we model 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.¹ 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.

6.1 Ride-sharing MDP Game With Queues

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

Modeling assumptions. 1) All trips take discretized times of $\{15, 20, 30, \dots\}$ based on the trip distance. 2) The initial driver distribution is uniform across all Manhattan zones. We find that varying the initial distribution does not significantly impact the time-averaged MDP Wardrop equilibrium or the toll norm, as long as the constraints are satisfied. 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. Each state is given by $s = (z, q)$, where $z \in [63]$ is one of 63 Manhattan zones, visualized the right plot in Figure 2 (islands excluded), and $q \in [Q]$, where $Q = 7$ is the queue level. If $q = 0$, the driver is in zone z without a rider. If $q > 0$, the driver is q time steps away from completing a ride to zone z . The set of geographically adjacent (sharing one or more edges) zones of z is given by $\mathcal{N}(z)$.

Actions. The action set of state (z, q) is q -dependent and is given by $\mathcal{A}(z, q)$. When $q > 0$, the driver is completing a ride. Therefore, the only action, a_z , is to finish the ride and $\mathcal{A}(z, q)$ is a singleton set $\mathcal{A}(z, q) = \{a_z\}, \forall z \in [63], q \geq 1$. When $q = 0$, the driver can either go to a neighboring zone

¹ Code for the Manhattan ride-share MDP congestion game is available at github.com/lisarah/manhattan.MDP.queue.game.

$(a_{z'})$ or pick up a rider in the current zone (a_z). The action set of $(z, 0)$ is $\mathcal{A}(z, 0) = \{a_{z'} \mid z' \in \mathcal{N}(z) \cup \{z\}\}$, $\forall z \in [63]$. We can extend the q -dependent action model to conform to the MDP model in Section 3 by defining $A = \max_{z,q} |\mathcal{A}(z, q)|$, and for all (z, q) where $|\mathcal{A}(z, q)| < A$, insert $A - |\mathcal{A}(z, q)|$ actions with infinite costs.

Time. The average trip time from the TLC data is 12.02 minutes. We add buffer time for drivers to locate and drop off riders, such that the MDP time interval is 15 minutes between 9 am and noon for a total of $T = 12$ time steps.

Transition Dynamics. Transition dynamics are q -dependent. When $q > 0$, the driver is completing a ride (a_z). Then, for all $z \in [63]$ and $t \in [T]$, the transition dynamics of (z, q, a_z) is given by

$$P(t, s', a_z, z, q) = \begin{cases} 1 & s' = (z, q - 1) \\ 0 & \text{otherwise} \end{cases}, \forall q \geq 1. \quad (24)$$

When $q = 0$, the driver may go to an adjacent zone ($\{a_{z'} \mid z' \in \mathcal{N}(z)\}$) or pick up a rider (a_z). For $a_{z'}$, the transition dynamics is given by

$$P(t, s', a_{z'}, z, q) = \begin{cases} 1 - \delta, & \text{if } s' = (z', 0), \\ \frac{\delta}{|\mathcal{N}(z)| - 1}, & \text{if } s' = (\bar{z}, 0), \bar{z} \in \mathcal{N}(z) / \{z\}, \\ 0, & \text{otherwise,} \end{cases} \quad (25)$$

where $\delta \in [0, 1)$ models the driver's probability of real-time deviation from a chosen strategy. We set $\delta = 0.01$.

For action a_z from state $(z, 0)$, drivers will find a ride and transition to the appropriate queue in the destination zone. The transition dynamics for (z, q, a_z, t) is derived using the TLC ride demand distribution at $(z, 0)$ [21]. Let $N(z, z', q, t)$ be the number of trips with origin-destination (z, z') at time step t that took between $15q$ and $15(q + 1)$ minutes. For all $z' \in [63]$ and $q \in [Q]$ at time $t \in [T]$, the transition probability to state (z', q) is given by

$$P((z', q), (z, 0), a_z, t) = \frac{N(z, z', q, t)}{\sum_{\bar{q} \in [Q]} \sum_{\bar{z} \in [63]} N(z, \bar{z}, \bar{q}, t)}.$$

Note that z' need not be an adjacent zone to z .

Driver costs. Driver costs are q -dependent at each state $s = (z, q)$. When $q > 0$, the driver cost is given by

$$\ell_{tsa}(y_{tsa}) = \beta y_{tsa}, \forall t \in [T], a \in \mathcal{A}(z, q),$$

where β models the minor congestion effect of drivers entering zone z at queue level q . We set $\beta = 0.001$.

When $q = 0$, we follow the cost model in [14], given by

$$\begin{aligned} \ell_{tsa}(y_{tsa}) &= \mathbb{E}_{s'} [c_{ts's}^{\text{trav}} - m_{ts's}] + c_t^{\text{wait}} \cdot y_{tsa} \\ &= \sum_{s'} P_{ts'sa} [c_{ts's}^{\text{trav}} - m_{ts's}] + c_t^{\text{wait}} \cdot y_{tsa} \end{aligned} \quad (26)$$

The parameters in (26) are action-dependent: $m_{ts's}$ is the monetary reward, c_t^{wait} is the congestion scaling coefficient, and $c_{ts's}^{\text{trav}}$ is the fuel cost.

- (1) For $a_{z'}$ and $s' = (z', 0)$, $m_{ts's} = 0$ and $c_t^{\text{wait}} = 0.01$. The term $c_{ts's}^{\text{trav}}$ is the fuel cost of reaching z' , given by

$$c_{ts's}^{\text{trav}} = \mu \underbrace{d_{zz'}}_{\text{mi}} \underbrace{(\text{Vel})^{-1}}_{\text{hr/mi}} + \underbrace{(\text{Fuel Price})}_{\$/\text{gal}} \underbrace{(\text{Fuel Eff})^{-1}}_{\text{gal/mi}} \underbrace{d_{zz'}}_{\text{mi}}. \quad (27)$$

The parameter $d_{zz'}$ is the estimated trip distance between z and z' . When $z' = z$, d_{zz} is the average distance (mi) for all (z, z) trips from the TLC data. When $z' \neq z$, $d_{zz'}$ is the Haversine distance (mi) between z and z' . The parameter μ is a time-money tradeoff parameter, given in Table 1 along with other parameters.

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

Table 1
Parameters for the driver cost function.

- (2) For $a = a_z$ and $s' = (z', 0)$, $m_{ts's}$ is the monetary reward, defined using Uber's NYC pay rate [1] as

$$m_{ts's} = \max \left(\$7, \$2.55 + \$0.35 \cdot \Delta t + \$1.75 \cdot \Delta d \right), \quad (28)$$

where Δt is the trip time (min) and Δd is the trip distance (mi). We set $\Delta t = 12$ as the average trip time from the TLC data and Δd as the estimated Haversine distance between (z, z') . The parameter c_t^{wait} is the coefficient of congestion, scaled linearly by the portion of drivers who are waiting for a rider, and is given by

$$c_{tsa}^{\text{wait}} = \mathbb{E}_{s'} [m_{ts's}] \cdot \left(\underbrace{\text{Customer Demand Rate}}_{\text{rides}/\Delta t} \right)^{-1}, \quad (29)$$

where $m_{ts's}$ is given by (28) and the customer demand rate is derived from TLC data per time interval per day. We estimate the Uber ride demand to be 2.5 times more than the Yellow Taxi's ride demand in January 2019 [5] and scale the TLC data accordingly.

6.2 Online learning via conditional gradient descent

When drivers optimize their strategies for the ride-share game in Section 6.1, we assume that they cannot directly access the congestion costs model and transition dynamics. Instead, they collectively receive costs for a chosen joint policy, and iterate to find the equilibrium policy.

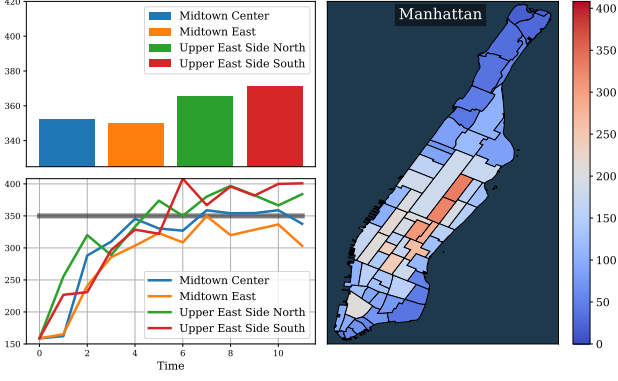


Fig. 2. Predicted ride-share traffic in Manhattan.

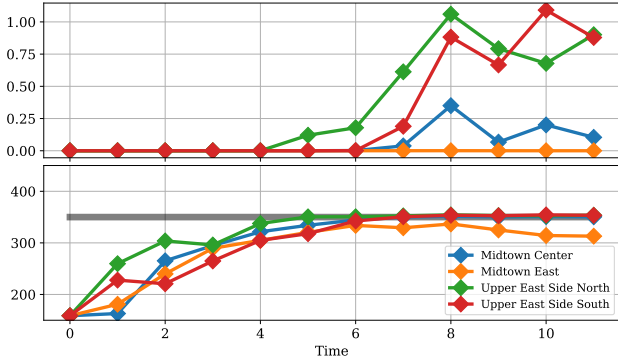


Fig. 3. Manhattan game under constraint (31). The top line plot shows congestion tolls (\$/15min). The bottom line plot shows the congested driver distributions (drivers/15min).

We implement the learning method from [14, Alg.3]. Inspired by conditional gradient descent (Frank-Wolfe), [14, Alg.3] implicitly enforces $y \in \mathcal{Y}(P, p_0)$ by solving linearized game potentials (8) via dynamic programming. Frank-Wolfe converges rapidly to low-accuracy solutions. Based on Frank-Wolfe’s stopping criterion, the ϵ in the ϵ -MDP Wardrop equilibrium is given by

$$\epsilon^k = (\ell(y^k) + A^\top \tau^k)^\top (y^k - y^{k+1}). \quad (30)$$

We set $\epsilon^k = 1e^3$, which is approximately equal to a normalized error of 0.5% for the unconstrained game potential. The corresponding ϵ -MDP Wardrop equilibrium and the driver densities of the most congested zones are shown in Figure 2.

6.3 Reducing driver presence in congested taxi zones

Suppose the ride-share company wishes to reduce the driver density in congested zones to below 350 per zone per time step via Algorithm 1. This constraint can be formulated as

$$\sum_a y_{tsa} \leq 350, \quad s = (z, 0), \quad \forall (t, z) \in [T+1] \times [63]. \quad (31)$$

Each $(t, z) \in [T+1] \times [63]$ corresponds to a constraint $A_i \in \mathbb{R}^{(T+1)SA}$, where for all $a \in \mathcal{N}(z, 0)$, the $(t, (z, 0), a)^{th}$

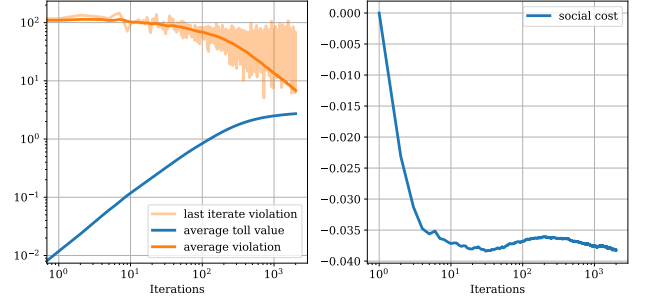


Fig. 4. Left: average toll value and constraint violation during toll synthesis. Right: average driver earnings during toll synthesis.

entry is 1 and all other entries are 0. Thus, we enforce a total of $63 \times 12 = 1008$ constraints of form (31). 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 2000 iterations at $\epsilon^k = 0.5\%$ of the unconstrained potential value. The results are shown in Figure 3. The resulting constraint violation has 2 norm 10.24 for the whole time horizon. The tolls 2 norm is 2.94.

In Figure 3(top), we see that for tolls around \$1 per time step per state, we can decrease the average constraint violation from over 200 drivers to less than 10 drivers for whole time horizon. This is comparable to the proposed toll value for lower Manhattan (\$2.25 per entry) [9]. In Figure 4 left, we evaluate Algorithm 1 by its toll value convergence and the constraint violation during tolling. Note that the average constraint violation $\|[A\bar{y}^k - b]_+\|_2$ differs from the last-iterate constraint violation $\|[Ay^k - b]_+\|_2$, and the last iterate constraint violation does not converge in part due to drivers’ imprecision in finding equilibrium strategy. .

Social cost. A major concern is the effects of tolling on the average drivers’ earnings, measured by the *social cost* [14]. When the social cost increases significantly, drivers may quit, thus reducing the ride-share workforce. We show empirically in Figure 4 (right) that tolling does not significantly impact driver earnings: the average driver earnings during the tolling process are normalized against the untolled average earnings. During tolling, the social cost in fact decreased, implying that the average driver earnings increased. Therefore, congestion-based tolling is unlikely to cause quitting among the driver population.

Equilibrium accuracy. The effect of ϵ^k on the toll value $\|\bar{\tau}\|_2$ and the average constraint violation $\|[A\bar{y}^k - b]_+\|_2$ is shown in Figure 5 after 1000 iterations for $\epsilon = [100, 1000, 5000, 10000, 50000]$. The increased accuracy in ϵ decreases both the toll value and the constraint violation during the tolling process, thus providing more incentive to accurately compute the minimum toll value.

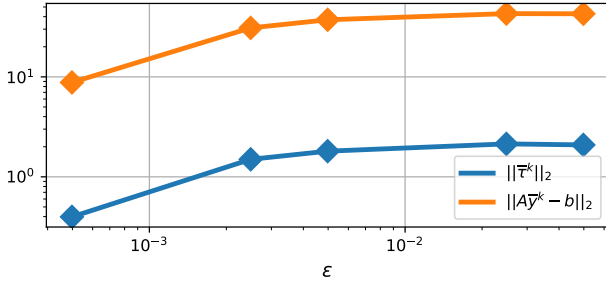


Fig. 5. ϵ vs average toll and constraint violation at $k = 1000$.

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. *Auton. Robots*, 39(4):537–554, 2015.
- [8] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 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 *Amer. Control Conf.*, 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. *Jpn. 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. In *2019 American Control Conference (ACC)*, pages 1238–1243. IEEE, 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 Trans. Power Syst.*, 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 J. Optim.*, 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, 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. Variable demand and multi-commodity flow in markovian network equilibrium. *Automatica*, 140:110224, 2022.
- [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 Properties of the tolled game

To supplement our main results, we provide concavity and smoothness properties of the tolled game $\hat{d}(\tau)$ with respect to ϵ -MDP Wardrop equilibria $\hat{y}_\tau(\epsilon)$.

Lemma 1 (Concavity) *Under Assumption 1, all ϵ -MDP Wardrop equilibria $\hat{y}_\tau(\epsilon)$ given by (18) will generate ϵ -inexact oracles (17) that satisfy*

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

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 (12) we can show that

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

Combining (A.25) 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.

Lemma 2 (ϵ -approximate smoothness) *Under Assumption 1, all ϵ -MDP Wardrop equilibria $\hat{y}_\tau(\epsilon)$ (18) will generate inexact oracles (17) that satisfy*

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

PROOF. We denote $\hat{y}_\tau(\epsilon)$ by \hat{y}_τ for simplicity and recall y_τ from (12). From Proposition 3, we know that

$$\nabla d(\tau) = \hat{\nabla}d(\tau) + A(y_\tau - \hat{y}_\tau). \quad (\text{A.3})$$

Substituting (A.26) into (15), we obtain the following

$$0 \leq d(\sigma) - d(\tau) - \hat{\nabla}d(\tau)^\top(\sigma - \tau) + \frac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2 - (A(y_\tau - \hat{y}_\tau))^\top(\sigma - \tau). \quad (\text{A.4})$$

Furthermore, we can show

$$\begin{aligned} \left| (A(y_\tau - \hat{y}_\tau))^\top(\sigma - \tau) \right| &\leq \|\hat{y}_\tau - y_\tau\|_2 \cdot \|A\|_2 \cdot \|\sigma - \tau\|_2 \\ &\leq \frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 + \frac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2, \end{aligned} \quad (\text{A.5})$$

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 $L(y, \tau)$ (11) is strongly convex under Assumption 1. We combine this with the fact that $L(y_\tau, \tau) = d(\tau)$ from (12) to obtain

$$\frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 \leq L(\hat{y}_\tau, \tau) - d(\tau). \quad (\text{A.6})$$

From (18), \hat{y}_τ satisfies

$$L(\hat{y}_\tau, \tau) - d(\tau) \leq \epsilon. \quad (\text{A.7})$$

Summing up (18), (A.27), (A.28), (A.29), and $2 \times$ (A.30), we obtain (A.2), which completes the proof.

Lemma 3 *Under Assumption 1, if $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, τ^s from Algorithm 1 satisfies*

$$\begin{aligned} \|\tau^{s+1} - \tau\|_2^2 &\leq \|\tau^s - \tau\|_2^2 + 2\gamma(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s \\ &\quad + \hat{\nabla}d(\tau^s)^\top(\tau^s - \tau)), \quad \forall \tau \in \mathbb{R}_+^C, k \geq 0. \end{aligned} \quad (\text{A.8})$$

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

$$r^{s+1} - r^s = 2(\tau^{s+1} - \tau^s)^\top(\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2. \quad (\text{A.9})$$

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

$$0 \leq (\tau^s + \gamma\hat{\nabla}d(\tau^s) - \tau^{s+1})^\top(\tau^{s+1} - \tau) \quad (\text{A.10})$$

From (A.33), we can upper bound $(\tau^{s+1} - \tau^s)^\top(\tau^{s+1} - \tau)$ and combine with (A.32) to obtain

$$r^{s+1} - r^s \leq 2\gamma\hat{\nabla}d(\tau^s)^\top(\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2. \quad (\text{A.11})$$

From Lemma 2, we recall

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

We can then combine (A.34) and $2\gamma \times$ (A.35) to derive

$$\begin{aligned} r^{s+1} - r^s + 2\gamma(L(y^s, \tau^s) - d(\tau^{s+1}) - 2\epsilon^s) \\ \leq 2\gamma\hat{\nabla}d(\tau^s)^\top(\tau^s - \tau) + \left(\frac{2\|A\|_2^2}{\alpha}\gamma - 1\right) \|\tau^{s+1} - \tau^s\|_2^2, \end{aligned} \quad (\text{A.13})$$

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

A.1.1 Proof of Proposition 3

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

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

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 Theorem 2

Let $r^s = \|\tau^s - \tau^*\|_2^2$. From Lemma 4, when $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, we have

$$r^{s+1} \leq r^s + 2\gamma(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s + \hat{\nabla}d(\tau^s)^\top(\tau^s - \tau^*)) \quad (\text{A.14})$$

From Lemma 1, we have

$$\hat{\nabla}d(\tau^s)^\top(\tau^s - \tau^*) \leq L(y^s, \tau^s) - d(\tau^*) \quad (\text{A.15})$$

Summing up (A.14) and $2\gamma \times (\text{A.15})$, we obtain

$$r^{s+1} - r^s \leq 2\gamma(d(\tau^{s+1}) - d(\tau^*) + 2\epsilon^s) \quad (\text{A.16})$$

Summing over (A.16) 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.

A.3 Proof of Theorem 1

We bound the term $F(\bar{y}^k) - F(y^*)$. First consider the upper bound. From Lemma 4, let $\tau = 0$,

$$\|\tau^{s+1}\|_2^2 \leq \|\tau^s\|_2^2 + 2\gamma(d(\tau^{s+1}) + 2\epsilon^s - L(y^s, \tau^s) + \hat{\nabla}d(\tau^s)^\top \tau^s). \quad (\text{A.17})$$

Recall from (16) and (11), $\hat{\nabla}d(\tau^s) = Ay^s - b$ 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 (A.17) becomes

$$\|\tau^{s+1}\|_2^2 + 2\gamma(F(y^s) - d(\tau^{s+1})) \leq \|\tau^s\|_2^2 + 4\gamma\epsilon^s \quad (\text{A.18})$$

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 $\tau^k \in \mathbb{R}_+^C$,

$$F(\bar{y}^k) - F(y^*) \leq \frac{1}{2\gamma k} \|\tau^0\|_2^2 + \frac{2E^k}{k}. \quad (\text{A.19})$$

Next, consider the lower bound of $F(\bar{y}^k) - F(y^*)$. By definition, y^* solves $\min_{y \in \mathcal{Y}(P, p)} F(y) + (Ay - b)^\top \tau^*$ 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 (11) 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,

$$F(y^*) - F(\bar{y}^k) \leq \frac{\|\tau^*\|_2}{\gamma k} (\|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + 2\sqrt{\gamma E^k}). \quad (\text{A.20})$$

Together, (A.19) and (A.20) imply

$$|F(y^*) - F(\bar{y}^k)| \leq \frac{1}{\gamma k} D(\tau^*, \tau^0, E^k) \quad (\text{A.21})$$

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 (A.21) completes the proof.

A.4 Proof of Corollary 1

We first derive an upper bound for $\|\tau^k\|_2$ and then bound the left hand side of (21) by $\|\tau^k\|_2$. Recall (A.16), 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

$$\|\tau^k - \tau^*\|_2^2 \leq \|\tau^0 - \tau^*\|_2^2 + 4\gamma E^k. \quad (\text{A.22})$$

Taking the square root of both sides of (A.22) and noting the identity $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain

$$\|\tau^k - \tau^*\|_2 \leq \|\tau^0 - \tau^*\|_2 + \sqrt{4\gamma E^k}. \quad (\text{A.23})$$

We add $\|\tau^*\|_2$ to both sides of (A.23) and use the triangle inequality $\|\tau^k\|_2 \leq \|\tau^k - \tau^*\|_2 + \|\tau^*\|_2$ to obtain

$$\|\tau^k\|_2 \leq \|\tau^*\|_2 + \|\tau^0 - \tau^*\|_2 + \sqrt{4\gamma E^k}. \quad (\text{A.24})$$

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 (A.24) completes the proof.

A.5 Proof of Proposition 3

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

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

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.6 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 (12) we can show that

$$d(\sigma) \leq L(\hat{y}_\tau, \sigma). \quad (\text{A.25})$$

Combining (A.25) 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.7 Proof of Lemma 2

PROOF. We denote $\hat{y}_\tau(\epsilon)$ by \hat{y}_τ for simplicity and recall y_τ from (12). From Proposition 3, we know that

$$\nabla d(\tau) = \hat{\nabla} d(\tau) + A(y_\tau - \hat{y}_\tau). \quad (\text{A.26})$$

Substituting (A.26) into (15), we obtain the following

$$\begin{aligned} 0 \leq & d(\sigma) - d(\tau) - \hat{\nabla} d(\tau)^\top (\sigma - \tau) \frac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2 \\ & - (A(y_\tau - \hat{y}_\tau))^\top (\sigma - \tau). \end{aligned} \quad (\text{A.27})$$

Furthermore, we can show

$$\begin{aligned} \left| (A(y_\tau - \hat{y}_\tau))^\top (\sigma - \tau) \right| & \leq \|\hat{y}_\tau - y_\tau\|_2 \cdot \|A\|_2 \cdot \|\sigma - \tau\|_2 \\ & \leq \frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 + \frac{\|A\|_2^2}{2\alpha} \|\sigma - \tau\|_2^2, \end{aligned} \quad (\text{A.28})$$

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 $L(y, \tau)$ (11) is strongly convex under Assumption 1. We combine this with the fact that $L(y_\tau, \tau) = d(\tau)$ from (12) to obtain

$$\frac{\alpha}{2} \|\hat{y}_\tau - y_\tau\|_2^2 \leq L(\hat{y}_\tau, \tau) - d(\tau). \quad (\text{A.29})$$

From (18), \hat{y}_τ satisfies

$$L(\hat{y}_\tau, \tau) - d(\tau) \leq \epsilon. \quad (\text{A.30})$$

Summing up (18), (A.27), (A.28), (A.29), and $2 \times (\text{A.30})$, we obtain (A.2), which completes the proof.

A.8 Lemma 4

Lemma 4 Under Assumption 1, if $\gamma \leq \frac{\alpha}{2\|A\|_2^2}$, τ^s from Algorithm 1 satisfy

$$\begin{aligned} \|\tau^{s+1} - \tau\|_2^2 & \leq \|\tau^s - \tau\|_2^2 + 2\gamma \left(d(\tau^{s+1}) - L(y^s, \tau^s) + 2\epsilon^s \right. \\ & \quad \left. + \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau) \right), \quad \forall \tau \in \mathbb{R}_+^C, k \geq 0. \end{aligned} \quad (\text{A.31})$$

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

$$r^{s+1} - r^s = 2(\tau^{s+1} - \tau^s)^\top (\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2. \quad (\text{A.32})$$

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

$$0 \leq (\tau^s + \gamma \hat{\nabla} d(\tau^s) - \tau^{s+1})^\top (\tau^{s+1} - \tau) \quad (\text{A.33})$$

From (A.33), we can upper bound $(\tau^{s+1} - \tau^s)^\top (\tau^{s+1} - \tau)$ and combine with (A.32) to obtain

$$r^{s+1} - r^s \leq 2\gamma \hat{\nabla} d(\tau^s)^\top (\tau^{s+1} - \tau) - \|\tau^{s+1} - \tau^s\|_2^2 \quad (\text{A.34})$$

From Lemma 2, we recall

$$\begin{aligned} L(y^s, \tau^s) - d(\tau^{s+1}) - 2\epsilon^s \\ \leq \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau^{s+1}) + \frac{\|A\|_2^2}{\alpha} \|\tau^{s+1} - \tau^s\|_2^2. \end{aligned} \quad (\text{A.35})$$

We can then combine (A.34) and $2\gamma \times (\text{A.35})$ to derive

$$\begin{aligned} r^{s+1} - r^s + 2\gamma(L(y^s, \tau^s) - d(\tau^{s+1}) - 2\epsilon^s) \\ \leq 2\gamma \hat{\nabla} d(\tau^s)^\top (\tau^s - \tau) + (2\gamma \frac{\|A\|_2^2}{\alpha} - 1) \|\tau^{s+1} - \tau^s\|_2^2, \end{aligned} \quad (\text{A.36})$$

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