Online Learning for Traffic Routing under Unknown Preferences

Devansh Jalota
Stanford University

Karthik Gopalakrishnan
Stanford University

Navid Azizan
Massachusetts Institute of Technology

Ramesh Johari
Stanford University

Marco Pavone
Stanford University

Abstract

In transportation networks, road tolling schemes are a method to cope with the efficiency losses due to selfish user routing, wherein users choose routes to minimize individual travel costs. However, the efficacy of tolling schemes often relies on access to complete information on users' trip attributes, such as their origin-destination (O-D) travel information and their values of time, which may not be available in practice. Motivated by this practical consideration, we propose an online learning approach to set tolls in a traffic network to drive heterogeneous users with different values of time toward a system-efficient traffic pattern. In particular, we develop a simple yet effective algorithm that adjusts tolls at each time period solely based on the observed aggregate flows on the roads of the network without relying on any additional trip attributes of users, thereby preserving user privacy. In the setting where the O-D pairs and values of time of users are drawn i.i.d. at each period, we show that our approach obtains an expected regret and road capacity violation of $O(\sqrt{T})$, where T is the number of periods over which tolls are updated. Our regret guarantee is relative to an offline oracle with complete information on users' trip attributes. We further establish a $\Omega(\sqrt{T})$ lower bound on the regret of any algorithm, which establishes that our algorithm is optimal up to constants. Finally, we demonstrate the superior performance of our approach relative to several benchmarks on a real-world traffic network, which highlights its practical applicability.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

1 INTRODUCTION

Many real-world systems are composed of self-interested users that interact non-cooperatively in a shared environment. However, in such systems, the lack of coordination between self-interested users often results in inefficient outcomes that are at odds with the goals of the system designer. For instance, in transportation networks, the *selfish routing* by users who choose routes to minimize their travel times (Roughgarden and Éva Tardos, 2004) typically results in a traffic pattern that is far from an efficient traffic assignment (Sheffi, 1985). As a result, there has been a growing interest in designing intervention and control schemes to cope with the selfishness of users (Roughgarden and Tardos, 2002) across a wide range of resource allocation applications, including energy management in smart grids (Palensky et al., 2011; Azizan et al., 2020) and traffic routing on road networks (Pigou, 1912; Sharon et al., 2018). One promising method that has emerged to mitigate the resulting efficiency losses is to set prices on the shared resources to influence and steer user behavior to align with system-efficient outcomes (Cole et al., 2003).

In the context of traffic routing, road tolls are commonly used to cope with the inefficiency loss due to the selfishness of users and enforce the system-optimum solution as a user equilibrium (Karakostas and Kolliopoulos, 2004). However, the computation of these tolls typically relies on solving a centralized optimization problem, which assumes complete information on users' trip attributes (Fleischer et al., 2004), such as their origin-destination (O-D) travel information and their values of time (Walters, 1961). In practice, this information is typically not available since this would violate user privacy and can thus confound the successful deployment of a tolling scheme to regulate road traffic. Further, users' trip attributes often vary with time, e.g., when users' values of time and O-D pairs are drawn i.i.d. from some unknown distribution. As a result, the central planner may need to periodically collect users' trip attributes and re-solve a large-scale optimization problem that may be computationally expensive.

As a result, in this work, we propose an online learning approach to set tolls in a traffic network to steer heterogeneous users with different values of time to a system optimum traffic pattern that minimizes the sum of the travel times of all users weighted by their values of time. In this setting, we assume that the value of time and O-D pair of each user are private information and cannot be used to design optimal tolling policies. Our method to set road tolls is different from prior centralized optimization approaches that require complete information on the values of time and O-D pairs of users. In this incomplete information setting, our algorithmic approach relies on adjusting road tolls based on the observed aggregate flows on the edges of the network. We mention that such aggregate flow data is readily available through modern sensing technologies, such as loop detectors, and helps maintain user privacy.

1.1 Our Contributions

In this work, we study the problem of setting optimal tolls that minimize the total system cost, i.e., the sum of the travel times of all users weighted by their values of time, in a capacity-constrained traffic network. We study this problem in the incomplete information setting when the O-D pairs and values of time of heterogeneous users are not known. As centralized optimization approaches are typically not conducive in this setting, we consider learning the tolls over time to minimize (i) regret, i.e., the optimality gap between the resulting allocation and that of an offline oracle with complete information on users' trip attributes, and (ii) constraint violation, i.e., the extent to which the road capacity constraints are violated.

To this end, we develop a simple yet effective approach to set tolls that preserves user privacy while achieving sublinear regret and constraint violation guarantees in the number of periods T over which the tolls are updated. Further, we establish a regret lower bound to show that our algorithm is asymptotically optimal, up to constants.

We then evaluate the performance of our approach on a real-world traffic network. The results of our experiments not only validate the theoretical regret and constraint violation guarantees but also highlight the superior performance of our algorithm relative to several benchmarks. Moreover, our approach achieves a total travel time close to the minimum achievable total travel time in the network.

Organization: Our paper is organized as follows. Section 2 reviews related literature. We then present a model of traffic flow and the regret and constraint violation performance measures to evaluate the efficacy of a tolling policy in Section 3. Then, in Section 4, we introduce our online learning algorithm and its associated regret and constraint violation guarantees. Next, we evaluate the performance of our approach on a real-world transportation network through numerical experiments in Section 5. Finally, we provide directions for future work in Section 6.

2 LITERATURE REVIEW

Traditional approaches to achieving an efficient allocation of resources have typically relied on complete information of users' preferences (Dafermos, 1973; Karakostas and Kolliopoulos, 2004). However, such approaches are, in general, practically infeasible since having complete information on users' preferences would violate user privacy.

As a result, there has been a growing interest in designing mechanisms that do not require complete information of users' preferences to achieve an efficient resource allocation. To this end, mechanism design has enabled the truthful elicitation of users' private information (Parkes et al., 2004). Furthermore, inverse game theory (Kuleshov and Schrijvers, 2015) has enabled the learning of users' preferences, e.g., O-D travel demands (Bertsimas et al., 2015). In contrast to these works, we do not directly learn or elicit user preferences to set optimal tolls. Instead, our approach bypasses the need to have information on users' preferences by using the total observed flow on each road as feedback to update tolls while retaining good performance.

Our toll update procedure is, in principle, similar to price update mechanisms that utilize past observations of user consumption, i.e., users' revealed preferences, to inform future pricing decisions. Such pricing mechanisms that use information from interactions with earlier buyers to inform pricing decisions for subsequent buyers have been widely studied in revenue management (Kleinberg and Leighton, 2003), online linear programming (Li et al., 2020), and Stackelberg games (Roth et al., 2016; Ji et al., 2018). In particular, as in (Li et al., 2020; Roth et al., 2016; Ji et al., 2018), our toll update procedure follows from performing gradient descent on the dual of the system optimization problem of the central planner. However, compared to prior works that typically consider the setting where users arrive sequentially and each user observes a different price, we study a multi-period setting wherein tolls are the same for all users but can vary across periods.

Dynamic pricing mechanisms have also been studied in the incomplete information setting for traffic routing applications. For instance, Yang et al. (2004, 2010) study trial-and-error approaches to set marginal cost tolls in the absence of users' demand functions. Furthermore, Melo (2011) develops dynamic variants of Pigou's solution to mitigate negative externalities by nudging users toward the system-optimum solution. In line with these approaches, we also learn and iteratively update road tolls through repeated interactions of users in the traffic network. However, in contrast to these approaches, we consider the setting of heterogeneous users with different values of time.

Our work focuses on jointly optimizing both regret and constraint violation as in the literature on constrained convex optimization with long-term constraints (Yu et al., 2017; Jenatton et al., 2016; Liakopoulos et al., 2019; Yi et al., 2021; Mahdavi et al., 2012). However, as opposed to the regret notion in these works defined based on the sub-optimality of an optimal static action in hindsight, we adopt a dynamic regret notion wherein the oracle can vary its actions across time steps. Our dynamic regret notion is similar to those considered in (Li et al., 2020; Chen et al., 2017; Cao and Liu, 2019); however, we consider a revealed preference setting wherein users' trip attributes are private information while, in these works, the central planner observes the cost functions after the arrival of each user.

3 MODEL AND PROBLEM FORMULATION

In this section, we introduce the basic definitions and user behavior model we consider in this work (Section 3.1), the system optimization objective of the central planner (Section 3.2), and the performance measures to evaluate the efficacy of a tolling policy (Section 3.3).

3.1 Preliminaries and User Optimization

We study the problem of routing heterogeneous users between their respective O-D pairs in a capacity-constrained road network. The road network is modeled as a directed graph $G = (V, E)$, with vertex and edge sets denoted by V and E, respectively. Each edge $e \in E$ has a fixed capacity c_e and a fixed latency (travel time) l_e, and we denote $\boldsymbol{c} = \{c_e\}_{e \in E}$ as the vector of edge capacities.

The set of all users is denoted by $\mathcal{U}$ and each user $u \in \mathcal{U}$ makes trips in the road network between an origin-destination (O-D) pair $w_u = (s_u, t_u)$, where s_u represents the origin and t_u represents the destination of the trip. Each O-D pair is connected by paths, which are finite sequences of directed edges between the origin and destination, and each user u selects one such path from the set of all paths $\mathcal{P}_u$ that connect the O-D pair w_u. An assignment of users to paths is denoted by the vector $\boldsymbol{f} = \{f_{P,u} : P \in \mathcal{P}_u, u \in \mathcal{U}\}$, where $f_{P,u} \in \{0, 1\}$ denotes whether user u is assigned to path P. Each path flow $\boldsymbol{f}$ corresponds to a unique edge flow vector $\boldsymbol{x} = \{x_e : e \in E\}$, where $x_e = \sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u : e \in P} f_{P,u}$. Each user u also has an *outside option* for which the user incurs a cost λ_u, which can be interpreted as the cost for not completing the trip, e.g., staying at home. The outside option is a commonly used modelling assumption in the transportation literature (Nikzad, 2017; Ostrovsky and Schwarz, 2019), and, in this traffic routing context, captures a decision making framework where each user u may not wish to incur a cost, including travel time and tolls, of more than λ_u for their trip. We denote whether users choose the outside option through the vector $\boldsymbol{f}_o = \{f_{o,u} : u \in \mathcal{U}\}$, where $f_{o,u} \in \{0, 1\}$ is a binary variable that takes the value one when user u chooses the outside option.

Users are assumed to be selfish and thus choose a path (or the outside option) that minimizes their travel cost, which we assume is a linear function of tolls and travel time. For each user u with a value-of-time $v_u > 0$ and a vector of edge prices (or tolls) $\boldsymbol{\tau} = \{\tau_e\}_{e \in E}$, the travel cost on a path P is given by $C_P(\boldsymbol{\tau}) = \sum_{e \in P} (v_u l_e + \tau_e)$. For ease of notation, we denote the total travel time on path P as $l_P = \sum_{e \in P} l_e$ and its toll as $\tau_P = \sum_{e \in P} \tau_e$. Then, given a vector of tolls $\boldsymbol{\tau}$, we define a path flow $\boldsymbol{f}$ and outside option flow $\boldsymbol{f}_o$ to be an *equilibrium* if each user chooses a path (or the outside option) that minimizes their travel cost.

Definition 1 (Equilibrium)**.** For a given vector of tolls $\boldsymbol{\tau}$, a path flow $\boldsymbol{f}$ and outside option flow $\boldsymbol{f}_o$ is an equilibrium if for each user u, $\sum_{P \in \mathcal{P}_u} f_{P,u} + f_{o,u} = 1$, where $f_{P_u^*,u} = 1$ for some path P_u^* if $C_{P_u^*}(\boldsymbol{\tau}) \leq \min\big\{ \min_{Q \in \mathcal{P}_u} \{C_Q(\boldsymbol{\tau})\}, \lambda_u \big\}$, or $f_{o,u} = 1$ for the outside option if $\lambda_u \leq C_P(\boldsymbol{\tau})$, for all $P \in \mathcal{P}_u$.

A few comments about our modeling assumptions are in order. First, the travel time on any edge is independent of the number of users on that edge as long as the flow on that edge does not exceed its capacity. This approximation is largely consistent with observed travel times in real-world road networks, where the travel times on roads tend to stay relatively constant up until the road capacity, beyond which the travel time increases steeply (Li and Zhang, 2011). For a more detailed discussion on the validity of this assumption in modeling real-world traffic networks, we refer to Ostrovsky and Schwarz (2019). We do mention, however, that we also demonstrate how our proposed online learning approach can generalize to the context of congestion games, when the travel time on each edge is a function of its flow, in the extended version of our paper (Jalota et al., 2022) through numerical experiments. Furthermore, under our modeling assumptions, for an equilibrium flow, all users take the shortest "cost" path (or the outside option) given the set tolls $\boldsymbol{\tau}$ irrespective of whether this violates the road capacity constraints. Note that, in practice, setting the tolls too low will likely result in road capacity violations and thus congestion delays. However, as with general equilibrium models that do not model users' responses to shortages in the supplies of goods, for our purposes, we do not model traffic congestion since we seek to set tolls that keep capacity violations small. We note that in road networks small levels of capacity violation are acceptable since road capacities serve as a ball-park for the number of vehicles on the road such that congestion delays do not increase significantly.

3.2 System Optimization and Efficient Tolls

In this section, we present the system optimization problem of the central planner and the notion of market-clearing tolls that induce the optimal solution to the system optimization problem as an equilibrium flow.

System Optimization We now present the problem faced by the central planner, who seeks to minimize the total system cost while ensuring that the resulting traffic assignment is feasible.

$$U^* = \min_{\boldsymbol{f}, \boldsymbol{f}_o} \quad \sum_{u \in \mathcal{U}} \Big(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u} + \lambda_u f_{o,u} \Big), \tag{1a}$$

$$\text{s.t.} \quad \sum_{P \in \mathcal{P}_u} f_{P,u} + f_{o,u} = 1, \quad \forall u \in \mathcal{U}, \tag{1b}$$

$$\boldsymbol{f}_o \in \{0,1\}^{|U|}, f_{P,u} \in \{0,1\}, \forall P \in \mathcal{P}_u, u \in \mathcal{U}, \tag{1c}$$

$$\sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u : e \in P} f_{P,u} \leq c_e, \quad \forall e \in E, \tag{1d}$$

Here, (1a) is the total system cost objective, i.e., the sum of the travel times of all users using the network weighted by their values of time and the cost of the outside option for all users who do not use the network. Furthermore, (1b) are user allocation constraints as users will either use a feasible path or the outside option, (1c) are binary allocation constraints, and (1d) are the edge capacity constraints. Note that a feasible solution to Problem (1a)-(1d) exists since it is feasible for all users to choose the outside option o.

Efficient Tolls The central planner is tasked with setting tolls $\boldsymbol{\tau}$ that induce the system optimum solution, i.e., the solution to Problem (1a)-(1d), as an equilibrium flow. One such set of tolls that achieve this goal are *market-clearing* tolls that induce equilibrium flows $\boldsymbol{f}, \boldsymbol{f}_o$ satisfying the road capacity constraints, i.e., $\sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u : e \in P} f_{P,u} = x_e \leq c_e$ for all edges $e \in E$. Additionally, market-clearing tolls satisfy the property that $\tau_e = 0$ for all edges e such that the edge flow $x_e < c_e$, and $\tau_e \geq 0$ for all edges e such that $x_e = c_e$. Theorem 1 establishes that if the tolls are market clearing then the resulting equilibrium flows minimize the total system cost, i.e., the corresponding equilibrium flow is an optimal solution to Problem (1a)-(1d).

Theorem 1 (Efficiency of Market-Clearing Tolls)**.** *If the tolls $\boldsymbol{\tau}^*$ are market-clearing, then the corresponding equilibrium flows, given by $\boldsymbol{f}^*$, $\boldsymbol{f}_o^*$, are optimal solutions to Problem* (1a)-(1d).

Theorem 1 is akin to that in Ostrovsky and Schwarz (2019); however, we present its proof in Appendix 7 for completeness, as we additionally consider the setting when users have an outside option. While market-clearing tolls provide a method to induce equilibrium flows minimizing the total system cost, such tolls cannot typically be computed, e.g., through linear programming (see Section 4.1), since the values of time and O-D pairs of users are, in general, unknown to the central planner. Further, these user specific attributes tend to be time-varying, e.g., when users' values of time and O-D pairs are drawn i.i.d. from some distribution. Thus, a central planner would need to periodically collect these parameters and re-solve a large optimization problem to update the tolls, which may not be practically viable. Finally, we note that market-clearing tolls may not exist in general (Ostrovsky and Schwarz, 2019) though such tolls are guaranteed to exist if fractional flows are allowed (see Section 4.1).

As a result, in this work, we consider an online learning approach that only relies on past observations of aggregate flows on each road to set tolls and steer heterogeneous users toward a system-optimum traffic pattern over time. Our motivation for pursuing an online learning method is threefold. First, as opposed to centralized approaches, an online learning approach can bypass the need to have complete information on users' preferences. Next, mechanism design approaches that rely on truthfully eliciting users' preferences may not be practically viable and often be insufficient in inducing the system-optimum solution as an equilibrium flow (see the extended version of our paper (Jalota et al., 2022)). Finally, modern sensing technologies, e.g., loop detectors, are well equipped to collect aggregate road flow data and thus do not rely on periodically collecting information on user-specific attributes that may vary over time. Thus, an online learning approach that uses solely aggregate road flow data is practically viable.

3.3 Performance Measures for Online Learning

We now introduce the online learning setting and the performance measures used to gauge the efficacy of a tolling policy. We consider a setting wherein users make trips over multiple time periods $t = 1, \dots, T$, e.g., over multiple days, and users' O-D pairs and values of time at each period are drawn i.i.d. from some unknown probability distribution $\mathcal{D}$ that is fixed across the T periods. That is, the O-D pair and value of time vectors $(\boldsymbol{w}^t, \boldsymbol{v}^t) = ((w_u^t)_{u \in \mathcal{U}}, (v_u^t)_{u \in \mathcal{U}})$ are drawn i.i.d. from a distribution $\mathcal{D}$ with non-negative support for the value of time vector of users[1]. Note that a special case of this involves the setting where the O-D pair and value-of-time (w_u^t, v_u^t) of user u at each period t is drawn i.i.d. from some distribution $\mathcal{D}_u$, and the distribution $\mathcal{D} = \otimes_{u \in \mathcal{U}} \mathcal{D}_u$. Under this i.i.d. assumption on users' trip attributes, we focus on *privacy-preserving* tolling policies, wherein the central planner makes a tolling decision using only past observa-

[1] For the ease of exposition, we focus on the setting when the O-D pair and value of time vectors are drawn i.i.d. from a probability distribution. However, our proposed approach can also be extended to the setting when the cost of the outside option varies over time.

tions of the aggregate flows on the different roads of the traffic network. In other words, the tolling policy $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_T)$ that sets a sequence of tolls $\boldsymbol{\tau}^{(1)}, \ldots, \boldsymbol{\tau}^{(T)}$ is such that $\boldsymbol{\tau}^{(t)} = \pi_t(\{\boldsymbol{x}^{t'}\}_{t'=1}^{t-1})$, where $\boldsymbol{x}^t$ are the aggregate edge flows corresponding to an equilibrium solution $\boldsymbol{f}^t$ under a toll $\boldsymbol{\tau}^{(t-1)}$.

We evaluate the efficacy of an algorithm $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_T)$ using two metrics: (i) expected cumulative regret and (ii) expected cumulative constraint violation, where the expectation is with respect to the O-D pair and value-of-time distribution $\mathcal{D}$ of the users.

The regret of an algorithm $\boldsymbol{\pi}$ with a corresponding sequence of tolls $\boldsymbol{\tau}^{(1)}, \ldots, \boldsymbol{\tau}^{(T)}$ is evaluated through the expected difference between the optimal Objective (1a), given complete information on the values of time and O-D pairs of all users at each time period, and the cumulative objective of the algorithm $\boldsymbol{\pi}$ over the T periods. Let $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ denote the equilibrium flow at time t given the toll $\boldsymbol{\tau}^{(t)}$, and $\boldsymbol{f}^{t*}, \boldsymbol{f}_o^{t*}$ denote the system optimum flows at time t. Further, let $\mathcal{P}_u^t$ be the set of feasible paths corresponding to the O-D pair w_u^t. Then, the regret $R_T(\boldsymbol{\pi})$ of an algorithm $\boldsymbol{\pi}$ is

$$R_T(\boldsymbol{\pi}) = \mathbb{E}_{\mathcal{D}}\Big[\sum_{t=1}^{T}\Big(\sum_{u \in \mathcal{U}}\Big(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t\Big) - \sum_{u \in \mathcal{U}}\Big(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^{t*} + \lambda_u f_{o,u}^{t*}\Big)\Big)\Big],$$

where the expectation is taken with respect to the distribution $\mathcal{D}$. In the remainder of this work, with slight abuse of notation, we will assume that all expectations are with respect to $\mathcal{D}$ and thus we drop the subscript $\mathcal{D}$ in the expectation. We mention here that regret measures in online learning often define regret based on the sub-optimality with respect to an optimal static action in hindsight. In contrast, we adopt a more powerful offline oracle model, wherein the oracle can vary its actions across time steps as users' attributes are random.

We evaluate the constraint violation of an algorithm $\boldsymbol{\pi}$ through the norm of the expected cumulative excess flow beyond each edge's capacity. That is, given the edge flows $\boldsymbol{x}^t$ corresponding to the equilibrium flows induced by the tolls $\boldsymbol{\tau}^{(t)}$ for each period t, the cumulative constraint violation vector $\boldsymbol{v}$ of an algorithm $\boldsymbol{\pi}$ is $\boldsymbol{v}(\boldsymbol{\pi}) = \Big(\sum_{t=1}^{T}(\boldsymbol{x}^t - \boldsymbol{c})\Big)_+$, and its expected norm is $V_T(\boldsymbol{\pi}) = \mathbb{E}\left[\|\boldsymbol{v}(\boldsymbol{\pi})\|_2\right]$.

We focus on jointly optimizing regret and capacity violation as in several prior works (Yu et al., 2017; Jenatton et al., 2016; Liakopoulos et al., 2019; Yi et al., 2021; Mahdavi et al., 2012). Note that minimizing either one of these metrics is typically easy. For instance, the absence of tolls is likely to result in good regret guarantees since each user will solve a shortest path problem that disregards road capacity constraints. In contrast, setting large road tolls will likely reduce capacity violations but achieve a higher regret. Since there is a trade-off between regret and capacity violation, achieving good performance on both these measures under minimal assumptions is often challenging (Li et al., 2020). Thus, we focus on jointly optimizing performance across both metrics.

Furthermore, we focus on the stochastic setting as in prior literature on traffic routing problems (Bertsimas and Ryzin, 1993) since in the adversarial setting past observations of user consumption, i.e., aggregate road flows, may not be informative of users' future consumption behavior, as their values of time and O-D pairs can change dramatically between subsequent periods in an adversarial instance. Thus, obtaining sub-linear regret guarantees may not be possible in the adversarial setting, especially with our chosen strong regret benchmark.

4 ONLINE TRAFFIC ROUTING ALGORITHM

In this section, we develop an online learning algorithm that relies on only the aggregate road flows in the traffic network and achieves sub-linear regret and constraint violation guarantees. To perform the regret analysis, we first consider the linear programming relaxation of Problem (1a)-(1d) and its corresponding dual in Section 4.1. We then present our online learning algorithm in Section 4.2. Finally, in Section 4.3, we show that our algorithm achieves the optimal regret guarantee, up to constants, by establishing an upper bound on its regret and constraint violation and a lower bound on the regret of any online algorithm.

4.1 Linear Programming Relaxation and Dual

We first present the linear programming relaxation of Problem (1a)-(1d)

$$\begin{aligned} U^* = \min_{\boldsymbol{f}, \boldsymbol{f}_o} \quad & \sum_{u \in \mathcal{U}}\Big(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u} + \lambda_u f_{o,u}\Big), && \text{(2a)} \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_u} f_{P,u} + f_{o,u} = 1, \quad \forall u \in \mathcal{U}, && \text{(2b)} \\ & \boldsymbol{f}_o \geq \mathbf{0}, f_{P,u} \geq 0, \forall P \in \mathcal{P}_u, u \in \mathcal{U} && \text{(2c)} \\ & \sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u : e \in P} f_{P,u} \leq c_e, \quad \forall e \in E, && \text{(2d)} \end{aligned}$$

where the binary allocation Constraints (1c) are relaxed with non-negativity Constraints (2c). Denote μ_u as the dual variable for Constraint (2b) for each user u and the toll τ_e as the dual variable of the capacity Constraint (2d) for each edge e. Then, the vector of road tolls $\boldsymbol{\tau}$ can be computed through the following dual of the linear Program (2a)-(2d)

$$\max_{\tau_e, \mu_u} \quad \sum_{u \in \mathcal{U}} \mu_u - \sum_{e \in E} \tau_e c_e, \tag{3a}$$

$$\text{s.t.} \quad \tau_e \geq 0, \quad \forall e \in E, \tag{3b}$$

$$\mu_u \leq v_u l_P + \sum_{e \in P} \tau_e, \forall P \in \mathcal{P}_u, u \in \mathcal{U}, \tag{3c}$$

$$\mu_u \leq \lambda_u, \quad \forall u \in \mathcal{U}. \tag{3d}$$

A few comments about the dual Problem (3a)-(3d) are in order. First, dual linear programs analogous to that in Problem (3a)-(3d) have been used to set tolls in the complete information setting when a central planner has knowledge of users' values of time and O-D pairs (Fleischer et al., 2004). Next, the optimal tolls $\boldsymbol{\tau}^*$ of Problem (3a)-(3d) satisfy a market-clearing property that $\tau_e = 0$ on a given edge if the aggregate flow on edge e is strictly below its capacity and $\tau_e \geq 0$ otherwise. Furthermore, Constraint (3c) (and Constraint (3d)), together with the complementary slackness optimality conditions, imply that the flow of user u on path P (or the outside option) is strictly greater than zero, i.e., $f_{P,u} > 0$ (or $f_{o,u} > 0$), if the dual variable μ_u of the allocation Constraint (2b) is equal to the travel cost on that path (or the cost of the outside option). That is, Constraints (3c) and (3d) imply that the travel cost μ_u incurred by each user u is the minimum across all paths and the outside option under the tolls $\boldsymbol{\tau}^*$. As a result, the optimal solution $\boldsymbol{f}^*, \boldsymbol{f}_o^*$ to Problem (2a)-(2d) is a (non-atomic) equilibrium under the optimal tolls $\boldsymbol{\tau}^*$ of the dual Problem (3a)-(3d). Finally, since μ_u is the minimum travel cost across all feasible paths and the outside option for each user, the dual Problem (3a)-(3d) can be reformulated solely in terms of the tolls $\boldsymbol{\tau}$ as

$$\max_{\boldsymbol{\tau} \geq \mathbf{0}} \sum_{u \in \mathcal{U}} \min \Big\{ \min_{P \in \mathcal{P}_u} \Big\{ v_u l_P + \sum_{e \in P} \tau_e \Big\}, \lambda_u \Big\} - \sum_{e \in E} \tau_e c_e. \tag{4}$$

4.2 Online Learning Algorithm

In this section, we leverage the dual Problem (4) to derive an algorithm to dynamically set tolls for the setting when the O-D pair and values of time of users are drawn i.i.d. from some unknown distribution. In this setting, we develop an algorithm wherein the toll on each edge is adjusted at each time step based on the observed aggregate flows at the previous time step. In particular, we increase the toll on an edge if its flow is higher than its capacity and decrease the edge's toll if its flow is strictly lower than its capacity. To update the tolls, we use a step-size γ and ensure that the tolls are non-negative at each period $t \in [T]$. We reiterate that given the toll $\boldsymbol{\tau}^{(t)}$ at each period t, users choose paths (or the outside option) to minimize their travel costs. This process is presented formally in Algorithm 1, and the toll update procedure is depicted in Figure 3 in Appendix 10.

Algorithm 1: Efficient Routing via Privacy-Preserving Tolls

```
Input : Time Period $T$, Road Capacities $\boldsymbol{c}$
Set the Toll $\boldsymbol{\tau}^{(1)} \leftarrow \mathbf{0}$
for $t = 1, \dots, T$ do
    Phase I: User Equilibrium for Toll $\boldsymbol{\tau}^{(t)}$
    Initialize $\boldsymbol{f}^t, \boldsymbol{f}_o^t \leftarrow \mathbf{0}$
    /* Minimum cost Route */
    $Q_u^* \leftarrow \arg\min_{Q \in \mathcal{P}_u^t \cup \{o\}} \{ \min_{P \in \mathcal{P}_u^t} \{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \}, \lambda_u \}$ ;
    /* Users choose Paths */
    For each user $u \in \mathcal{U}$, set $f_{Q_u^*,u}^t = 1$ if $Q_u^* \in \mathcal{P}_u^t$, else $f_{o,u}^t = 1$ ;
    /* Observed Edge Flows */
    $x_e^t \leftarrow \sum_{u \in U} \sum_{P: e \in P} f_{P,u}^t$ ;
    Phase II: Toll Update
    $\boldsymbol{\tau}^{(t+1)} \leftarrow (\boldsymbol{\tau}^{(t)} - \gamma(\boldsymbol{c} - \boldsymbol{x}^t))_+$
end
```

A few comments about Algorithm 1 are in order. First, Algorithm 1 is privacy-preserving since the toll updates do not require any information on the O-D pair, values of time of users, or their traversed paths and only rely on the observed aggregate flows on each edge of the network. Observe that Phase I is a completely distributed step as users choose paths (or the outside option) to minimize their travel costs in response to the set tolls while the central planner adjusts tolls only in Phase II in response to the observed aggregate equilibrium edge flows. Next, the computational complexity of the toll updates (Phase II of Algorithm 1) at each period t is only $O(|E|)$, which makes Algorithm 1 practically viable. Furthermore, the computational complexity of Phase I of Algorithm 1 is $O(|\mathcal{U}|(|E|+|V|\log(|V|)))$, since at each period t each user solves a shortest path problem on the graph, which has a complexity of $O(|E|+|V|\log(|V|))$, in response to the set tolls. Furthermore, we note that the toll update procedure in Algorithm 1 follows as a direct consequence of applying gradient descent to the dual Problem (4). Despite the strong connection between Algorithm 1 and the dual Problem (4), we reiterate that Algorithm 1 results in binary allocations and thus corresponds to a solution to the original traffic routing problem presented in Section 3 rather than solely its relaxed linear programming variant presented in Section 4.1. Finally, we note that Algorithm 1 can also be generalized in the context of congestion games, when the travel time on each edge is a function of its flow (see the extended version of our paper (Jalota et al., 2022)).

4.3 Regret and Constraint Violation Guarantees

We now show that Algorithm 1 achieves the optimal regret guarantee, up to constants, by establishing matching upper and lower bounds on its regret.

We first present the main result of this work, which establishes that both the regret and constraint violation of Algorithm 1 are upper bounded by $O(\sqrt{T})$.

Theorem 2 (Square Root Regret and Constraint Violation)**.** *Suppose that the O-D pairs and values of time of users are drawn i.i.d. at each period $t \in [T]$ from some distribution $\mathcal{D}$ with non-negative support for users' values of time. Further, let $\boldsymbol{x}^t$ be the sequence of observed traffic flows under the equilibrium flows $\boldsymbol{f}^t$, $\boldsymbol{f}_o^t$ given the tolls $\boldsymbol{\tau}^{(t)}$ for each period t. Then, under Algorithm 1 with step size $\gamma = \frac{1}{\sqrt{T}}$, the regret $R_T(\boldsymbol{\pi}) \leq \frac{|E|(|\mathcal{U}|+\max_{e \in E} c_e)^2}{2}\sqrt{T}$, and constraint violation*[2] *$V_T(\boldsymbol{\pi}) \leq |E|(\max_{u \in \mathcal{U}} \lambda_u + \max_{e \in E} c_e + |\mathcal{U}|)\sqrt{T}$.*

Proof (Sketch) To prove this claim, we first derive a generic upper bound on the regret of any algorithm using linear programming duality, and then use the toll update steps to establish an upper bound on the regret of Algorithm 1. For the constraint violation bound, we first use the toll update steps to obtain a $O(\frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{\tau}^{(T+1)}\|_2])$ upper bound. Then, we show that the tolls remain bounded by a constant at each time step, since users will never travel on a path with a cost greater than that of their outside option, which establishes an $O(\frac{1}{\gamma}) = O(\sqrt{T})$ bound on the constraint violation. □

We refer to Appendix 8 for a complete proof of Theorem 2. We note that the proof of Theorem 2 is akin to that in Li et al. (2020); however, our work differs from Li et al. (2020) in several ways. First, Li et al. (2020) observes users' attributes at each iteration. In contrast, Algorithm 1 only relies on users' revealed preferences, i.e., past observations of user consumption, to perform the toll updates (Phase II of Algorithm 1), as we assume that each user's values of time and O-D pairs are private information. Thus, our algorithm requires less information to perform the toll updates than that in Li et al. (2020), as the optimization in Phase I of Algorithm 1 is done distributedly by the users. Next, the setting considered by Li et al. (2020) corresponds to one where no more than one decision variable is non-zero at each iteration of the algorithm. In contrast, our setting involves a more complex decision space with multiple non-zero decision variables at each iteration since all users are assigned to either one of the paths or the outside option at each step. Finally, our work considers equality constraints in user allocations compared to the inequality constraints in Li et al. (2020), as users in our setting must be allocated to either some path or the outside option at each step. This structural difference in the constraints between our formulation and that of Li et al. (2020) influences the analysis of bounding the tolls and hence the constraint violation of Algorithm 1.

We also reiterate that the step-size of the price updates of $\gamma = 1/\sqrt{T}$ is required to establish the bound in Theorem 2 since we adopt a more powerful oracle model, wherein the oracle can vary its actions across time steps. As a result, classical gradient descent approaches using the step-size of $\gamma_t = 1/\sqrt{t}$ for $t \in [T]$ (rather than $\gamma = 1/\sqrt{T}$ as our work) in the setting with an optimal static action in hindsight may not result in the desired regret and constraint violation guarantees due to the stronger regret notion adopted in this work. In particular, we note that numerical results of our problem setting with a time-varying step-size of $\gamma_t = 1/\sqrt{t}$ for $t \in [T]$ demonstrated a non-vanishing regret.

Having established an upper bound on the regret of Algorithm 1, we now show that no algorithm can achieve a regret lower than $\Omega(\sqrt{T})$ to establish that Algorithm 1 is optimal up to constants.

Theorem 3 (Regret Lower Bound)**.** *There exists a distribution $\mathcal{D}$ such that the regret of any algorithm is $\Omega(\sqrt{T})$ for the traffic routing problem where the O-D pair and values of time of users are drawn i.i.d. from $\mathcal{D}$ at each period.*

For a proof of Theorem 3, we refer to Appendix 9.

5 NUMERICAL EXPERIMENTS

We now evaluate the performance of Algorithm 1 on a real-world traffic network. Our numerical results not only validate the theoretical guarantees obtained in Theorem 2 but also show that our algorithm achieves better performance on regret, constraint violation, and total travel time metrics as compared to several benchmarks. In this section, we introduce three benchmarks to which we compare Algorithm 1 (Section 5.1) and present the results that demonstrate the theoretical and practical efficacy of Algorithm 1 (Section 5.2). The implementation details and the Sioux Falls data set that we use to test Algorithm 1 and the benchmarks are presented in Appendix 11.1, and our code is publicly available at https://github.com/StanfordASL/online-tolls. We also present additional numerical results in the extended version of our paper (Jalota et al., 2022) to demonstrate how our online learning approach can generalize to the context of congestion games, when the traffic on each edge is a function of its flow.

5.1 Benchmarks

While we presented a lower bound (Theorem 3) to establish the asymptotic optimality of Algorithm 1, in our experiments, we compare Algorithm 1 to several benchmarks. In particular, the first two benchmarks assume some knowledge about the user attributes, i.e., the mean value of time of the entire population or that of each user, and set static tolls that do not vary over time as with many existing road tolling schemes. In contrast, the third benchmark is analogous to Algorithm 1 in that it does not require any information on users' values of time; however, in this benchmark, the tolls are updated by a fixed constant at each step.

[2] The constant derived for the constraint violation bound is for the L_2 norm, and by norm equivalence this upper bound on the constraint violation holds for any p-norm, e.g., the L_∞ norm.

Population Value-of-Time (Population Mean VoT) We assume the central planner has access to the mean value-of-time of users in the entire population. In this setting, we compute the tolls through the optimal dual variables of the capacity constraints of Problem (2a)-(2d) when the values-of-time v_u of each user u are set equal to the mean value of time of the entire population. Note that this benchmark does not account for the variability in users' values of time, as is the case for many congestion pricing policies that neglect users' values of time in their tolling decisions.

Mean User Value-of-time (User Mean VoT) We assume that the central planner has more fine-grained information through access to the mean value of time of each user. In this context, we compute the tolls through the optimal dual variables of the capacity constraints of Problem (2a)-(2d) when the values-of-time v_u of each user u are set equal to their mean value of time.

Reactive Toll Updates We consider a variant of Algorithm 1, wherein the toll on each edge is updated at each period by a constant (set to $\$0.1$ in our experiments) solely based on the observed aggregate flows on that edge. In particular, if the flow exceeds the edge's capacity, the toll is increased by a specified constant irrespective of the magnitude of the capacity violation. On the other hand, if the flow is below the edge's capacity, the toll is decreased by a constant (or set to zero). This toll update procedure could be a natural strategy in large-scale traffic networks, where tolls can only be adjusted in constant increments.

We note that our chosen benchmarks reflect natural tolling strategies in large-scale traffic networks and thus provide a reasonable point of comparison for our algorithm. In particular, the static tolling benchmarks are akin to existing congestion pricing schemes that tend to remain fixed over time and entail solving a stochastic program, which serves as a universal benchmark for algorithm design under i.i.d. data (Li and Ye, 2021). The Population Mean VoT benchmark is likely easy to implement as data on the mean value of time of all users can typically be easily obtained through surveys or income statistics. On the other hand, the User Mean VoT benchmark is harder to implement but corresponds to a stronger benchmark with access to the mean values of time of all users. Beyond the static tolling approaches, the reactive toll update benchmark is reflective of practical toll update mechanisms, as tolls can often only be adjusted in small increments between periods.

5.2 Results

Assessment of Theoretical Bounds Figure 1 depicts the regret (right) and a log-log plot of the capacity violation (left), wherein the O-D pairs and values of time of users are drawn i.i.d. from a distribution, specified in Appendix 11.1. As expected from our theoretical results, for the capacity violation, the black dots representing the empirically observed capacity violations of Algorithm 1 in Figure 1 all lie very close to the theoretical $O(\sqrt{T})$ bound represented by a line with a slope of 0.5 on a logarithmic scale. Furthermore, the regret also satisfies the $O(\sqrt{T})$ bound since it is negative for this data set due to capacity violations.

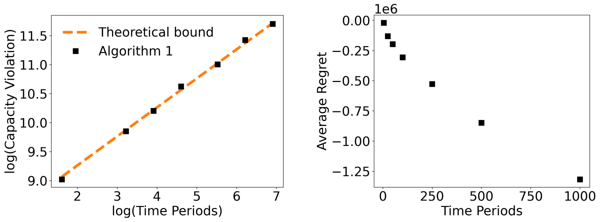


Figure 1: Validation of the theoretical upper bounds on regret and constraint violation obtained in Theorem 2 on the Sioux Falls data set. The infinity norm of the capacity violation is plotted on a log-log plot, and its empirical performance is very close, with a root mean square error of 0.037, to the theoretical $O(\sqrt{T})$ bound, represented by a line with a slope of 0.5. The regret is negative for this data set due to capacity violations and thus satisfies the $O(\sqrt{T})$ bound.

Regret and Capacity Violation Comparisons We now compare the normalized regret and capacity violation of Algorithm 1 to that of the benchmarks in Figure 2. Here, the normalized regret is the ratio between the regret and the optimal offline total system cost over the T time periods, and the normalized capacity violation is the ratio between the capacity violation and the cumulative capacity over the T periods. In Figure 2 (left and center), we observe that Algorithm 1 (i) outperforms all the benchmarks on both metrics for large values of T, (ii) obtains better performance in terms of regret as compared to the two static toll benchmarks for all values of T, and (iii) obtains a superior performance in terms of constraint violation as compared to the Reactive Toll Update benchmark for all values of T.

Between the two static tolling benchmarks, we observe from Figure 2 that the User Mean VoT benchmark performs better than the Population Mean VoT benchmark on both regret and constraint violation metrics. This result follows since the User Mean VoT benchmark has access to fine-grained information on the mean values of time of each user while the Population Mean VoT benchmark only has access to the mean value of time of the entire population. The performance of these two benchmarks, thus, points to the importance of considering the variability in users' values of time in designing congestion pricing schemes.

Compared to the static tolling benchmarks that assume knowledge of the mean values of time of each user (or the population), both dynamic tolling policies do not have access to any information on users' trip attributes. De-

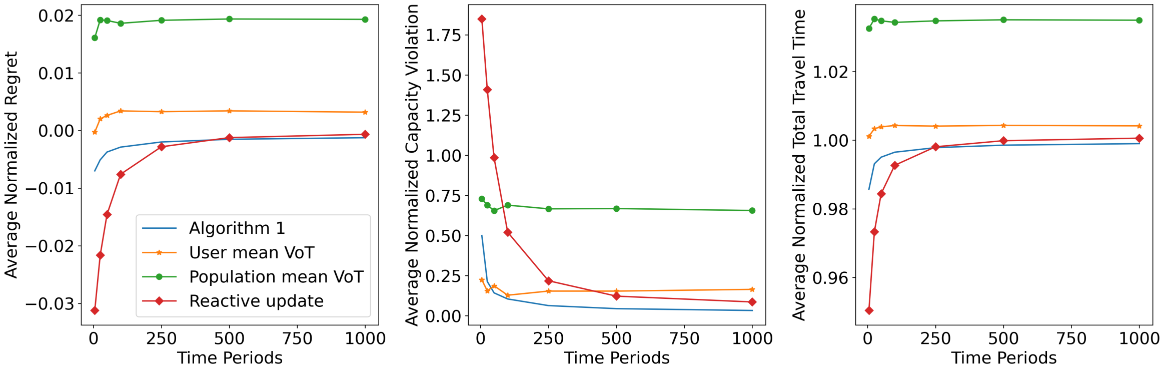


Figure 2: Comparison of the normalized regret, capacity violation, and the total travel time of Algorithm 1 and the benchmarks presented in Section 5.1. The normalized regret (left) represents the ratio between the regret and the optimal offline total system cost over the T periods. The normalized capacity violation (center) is the ratio between the L_∞ norm of the capacity violation and the cumulative road capacity over the T periods. The normalized total travel time (right) represents the ratio between the average total travel time of each algorithm over T periods and the optimal total travel time, i.e., the minimum total travel time of a traffic routing solution that satisfies the road capacity constraints.

spite this, Figure 2 shows that the dynamic tolling policies achieve better regret as compared to the static tolling benchmarks. The reason for the low regret of the Reactive Toll Update benchmark is that it has a higher capacity violation that enables users to take lower-cost routes. For larger values of T, Figure 2 indicates that Algorithm 1 eventually outperforms the static toll benchmarks on both regret and capacity violation. This observation suggests that setting fixed tolls, even using the mean values of time of each user, might be fraught with error since users' values of time may vary over time, even though the distribution from which values of time are drawn is stationary.

Between the two dynamic tolling policies, we observe that the Reactive Toll Update benchmark achieves a lower regret for small values of T because of a higher capacity violation. However, for larger T, Algorithm 1 outperforms this benchmark on both regret and capacity violation metrics. This result follows since Algorithm 1 updates tolls based on the exact discrepancy between the capacity and the edge flows. On the other hand, the Reactive Toll Update benchmark updates the tolls on each edge by a pre-specified increment depending solely on whether the capacity is violated. As a result, Algorithm 1, which can make infinitesimally small toll updates, achieves a vanishing normalized regret and capacity violation with large T. However, we do note that the Reactive Toll Update benchmark does come close to the performance of Algorithm 1 as it achieves only a slightly higher regret and only an 8% capacity violation for large values of T. Such a performance indicates that the Reactive Toll Update, in addition to Algorithm 1, can also be practically deployed in real-world traffic networks.

Total Travel Time We now demonstrate the practical efficacy of Algorithm 1 by comparing its total travel time, which may also be an important practical consideration for a central planner, to that of the benchmarks. Figure 2 (right) depicts the ratio of the average total travel time of each of the algorithms to the minimum achievable total travel time in the network, which corresponds to a solution satisfying the capacity constraints of all roads. In particular, both the dynamic tolling policies achieve lower total travel times than the static tolling benchmarks. Furthermore, while incurring small levels of capacity violation, Algorithm 1 achieves close to the minimum possible total travel time. Thus, even though Theorem 2 only provides guarantees for Algorithm 1 on regret and constraint violation metrics, it achieves good practical performance on even the total travel time metric, which may be of direct importance to central planners.

6 CONCLUSION AND FUTURE WORK

In this work, we proposed an online learning approach to set tolls in a traffic network to induce users with different values of time toward a system-optimum traffic pattern.

There are various directions for future research. First, it would be interesting to investigate whether the regret and capacity violation guarantees extend when the users' trip attributes are not drawn i.i.d., e.g., when they are drawn according to a random permutation model. Next, it would be worthwhile to generalize the obtained theoretical results to the context of congestion games, where the travel time on each edge is a function of its flow (see the extended version of our paper (Jalota et al., 2022)). Finally, it would also be valuable to explore the extension of the ideas and algorithm developed in this paper to objectives beyond system efficiency.

Acknowledgements

This research was supported by the National Science Foundation under the CPS program, the Center for Automotive Research at Stanford University, and Docomo Inc. We thank Emma Brunskill, Michael Ostrovsky, Dorsa Sadigh, Edward Schmerling, Kiril Solovey, and Matthew Tsao for helpful discussions as well as anonymous reviewers for their valuable feedback and suggestions.

References

Sioux falls, sd, 2022. URL `https://datausa.io/profile/geo/sioux-falls-sd/`. Accessed Februuary 10, 2022.

Navid Azizan, Yu Su, Krishnamurthy Dvijotham, and Adam Wierman. Optimal pricing in markets with non-convex costs. *Operations Research*, 68(2):480–496, 2020.

Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch. Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Math. Program.*, 153(2): 595–633, nov 2015. ISSN 0025-5610. doi: 10.1007/s10107-014-0819-4. URL `https://doi.org/10.1007/s10107-014-0819-4`.

Dimitris J. Bertsimas and Garrett Van Ryzin. Stochastic and dynamic vehicle routing with general demand and interarrival time distributions. *Advances in Applied Probability*, 25(4):947–978, 1993. ISSN 00018678. URL `http://www.jstor.org/stable/1427801`.

Xuanyu Cao and K. J. Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on Automatic Control*, 64(7):2665–2680, 2019. doi: 10.1109/TAC.2018.2884653.

Tianyi Chen, Qing Ling, and Georgios B. Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017. doi: 10.1109/TSP.2017.2750109.

Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 521–530, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136749. doi: 10.1145/780542.780618. URL `https://doi.org/10.1145/780542.780618`.

Stella C. Dafermos. Toll patterns for multiclass-user transportation networks. *Transportation Science*, 7(3):211–223, 1973. doi: 10.1287/trsc.7.3.211. URL `https://doi.org/10.1287/trsc.7.3.211`.

L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 277–285, 2004. doi: 10.1109/FOCS.2004.69.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325, 2016.

Devansh Jalota, Karthik Gopalakrishnan, Navid Azizan, Ramesh Johari, and Marco Pavone. Online learning for traffic routing under unknown preferences. *CoRR*, abs/2203.17150, 2022.

Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411. PMLR, 2016.

Ziwei Ji, Ruta Mehta, and Matus Telgarsky. Social welfare and profit maximization from revealed preferences. In *International Conference on Web and Internet Economics*, pages 264–281. Springer, 2018.

G. Karakostas and S.G. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 268–276, 2004. doi: 10.1109/FOCS.2004.26.

R. Kleinberg and T. Leighton. The value of knowing a demand curve: bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 594–605, 2003. doi: 10.1109/SFCS.2003.1238232.

Volodymyr Kuleshov and Okke Schrijvers. Inverse game theory: Learning utilities in succinct games. In Evangelos Markakis and Guido Schäfer, editors, *Web and Internet Economics*, pages 413–427, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. ISBN 978-3-662-48995-6.

Jia Li and H Michael Zhang. Fundamental diagram of traffic flow: new identification scheme and further evidence from empirical data. *Transportation research record*, 2260(1):50–59, 2011.

Xiaocheng Li and Yinyu Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 0(0):null, 2021. doi: 10.1287/opre.2021.2164. URL `https://doi.org/10.1287/opre.2021.2164`.

Xiaocheng Li, Chunlin Sun, and Yinyu Ye. Simple and fast algorithm for binary integer and online linear programming. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In *International Conference on Machine Learning*, pages 3944–3952. PMLR, 2019.

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

Emerson Melo. Congestion pricing and learning in traffic network games. *Journal of Public Economic Theory*, 13(3):351–367, 2011. doi: https://doi.org/10.1111/j.1467-9779.2011.01503.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9779.2011.01503.x.

Afshin Nikzad. Thickness and competition in on-demand service platforms. Technical report, Working Paper, 2017.

Michael Ostrovsky and Michael Schwarz. Carpooling and the economics of self-driving cars. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 581–582, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367929. doi: 10.1145/3328526.3329625. URL https://doi.org/10.1145/3328526.3329625.

Peter Palensky, Senior Member, Dietmar Dietrich, and Senior Member. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, pages 1551–3203, 2011.

David C Parkes, Dimah Yanovsky, and Satinder Singh. Approximately efficient online mechanism design. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper/2004/file/fc03d48253286a798f5116ec00e99b2b-Paper.pdf.

Arthur Pigou. *Wealth and Welfare*. London, Macmillan and Co., 1 edition, 1912.

Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 949–962, 2016.

Tim Roughgarden and Éva Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47 (2):389 – 403, 2004. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2003.06.004. URL http://www.sciencedirect.com/science/article/pii/S089982560300188X.

Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, March 2002. ISSN 0004-5411. doi: 10.1145/506147.506153. URL https://doi.org/10.1145/506147.506153.

Guni Sharon, Michael Albert, Tarun Rambha, Stephen Boyles, and Peter Stone. Traffic optimization for a mixture of self-interested and compliant agents. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Yossi Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, 1 edition, 1985.

Transportation Networks for Research Core Team. Transportation networks for research. github.com/bstabler/TransportationNetworks, 2016. Accessed January 20, 2021.

A. A. Walters. The theory and measurement of private and social cost of highway congestion. *Econometrica*, 29 (4):676–699, 1961. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911814.

Hai Yang, Qiang Meng, and Der-Horng Lee. Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions. *Transportation Research Part B: Methodological*, 38(6):477–493, 2004. ISSN 0191-2615. doi: https://doi.org/10.1016/S0191-2615(03)00077-8. URL https://www.sciencedirect.com/science/article/pii/S0191261503000778.

Hai Yang, Wei Xu, Bing sheng He, and Qiang Meng. Road pricing for congestion control with unknown demand and cost functions. *Transportation Research Part C: Emerging Technologies*, 18(2):157–175, 2010. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2009.05.009. URL https://www.sciencedirect.com/science/article/pii/S0968090X09000679.

Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and Karl Johansson. Regret and cumulative constraint violation analysis for online convex optimization with long term constraints. In *International Conference on Machine Learning*, pages 11998–12008. PMLR, 2021.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. *Advances in Neural Information Processing Systems*, 30, 2017.

Supplementary Material

7 Proof of Theorem 1

To prove this claim, we show that the equilibrium flows $\boldsymbol{f}^*, \boldsymbol{f}_o^*$ corresponding to the toll $\boldsymbol{\tau}^*$ minimizes the total system cost, i.e., $\sum_{u \in \mathcal{U}} \left(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u}^* + \lambda_u f_{o,u}^* \right) \leq \sum_{u \in \mathcal{U}} \left(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u} + \lambda_u f_{o,u} \right)$ for any other feasible flows $\boldsymbol{f}, \boldsymbol{f}_o$ that satisfy Constraints (1b)-(1d). To this end, suppose, without loss of generality, that under the equilibrium flows corresponding to the toll $\boldsymbol{\tau}^*$, users $u \in \mathcal{U}_1$ choose paths $P_1^*, \ldots, P_k^*$, i.e., $f^*_{P^*_u,u} = 1$ for some path $P_u^* \in \mathcal{P}_u$ for all users $u \in \mathcal{U}_1$, while users $u \in \mathcal{U}_2$ choose the outside option, i.e., $f_{o,u}^* = 1$ for all users $u \in \mathcal{U}_2$, where $\mathcal{U}_1 \cup \mathcal{U}_2 = \mathcal{U}$ and $\mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset$. Then, for any other feasible flows $\boldsymbol{f}, \boldsymbol{f}_o$ that satisfy the Constraints (1b)-(1d), it holds by the definition of an equilibrium for all users $u \in \mathcal{U}_1$ that

$$v_u l_{P_u^*} + \sum_{e \in P_u^*} \tau_e^* \leq \sum_{P \in \mathcal{P}_u} \left(v_u l_P + \sum_{e \in P} \tau_e^* \right) f_{P,u} + \lambda_u f_{o,u},$$

and for users $u \in \mathcal{U}_2$ that

$$\lambda_u \leq \sum_{P \in \mathcal{P}_u} \left(v_u l_P + \sum_{e \in P} \tau_e^* \right) f_{P,u} + \lambda_u f_{o,u}.$$

Summing up the above inequalities for all users $u \in \mathcal{U}$ and rearranging the equation we get that

$$\sum_{u \in \mathcal{U}_1} v_u l_{P_u^*} + \sum_{u \in \mathcal{U}_2} \lambda_u \leq \sum_{P \in \mathcal{P}_u} \left(v_u l_P + \sum_{e \in P} \tau_e^* \right) f_{P,u} + \lambda_u f_{o,u} - \sum_{u \in \mathcal{U}_1} \sum_{e \in P_u^*} \tau_e^*.$$

Finally, since the tolls $\boldsymbol{\tau}^*$ are market-clearing, it holds that $\sum_{u \in \mathcal{U}_1} \sum_{e \in P_u^*} \tau_e^* = \sum_{e \in E} \tau_e^* c_e \geq \sum_{e \in E} \tau_e^* x_e$ for any feasible edge flow $\boldsymbol{x} \leq \boldsymbol{c}$. Thus, it follows that

$$\begin{aligned} \sum_{u \in \mathcal{U}} \left(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u}^* + \lambda_u f_{o,u}^* \right) = \sum_{u \in \mathcal{U}_1} v_u l_{P_u^*} + \sum_{u \in \mathcal{U}_2} \lambda_u &\leq \sum_{u \in \mathcal{U}} \left(\sum_{P \in \mathcal{P}_u} \left(v_u l_P + \sum_{e \in P} \tau_e^* \right) f_{P,u} + \lambda_u f_{o,u} \right) - \sum_{e \in E} \tau_e^* c_e, \\ &\leq \sum_{u \in \mathcal{U}} \left(v_u \sum_{P \in \mathcal{P}_u} l_P f_{P,u} + \lambda_u f_{o,u} \right), \end{aligned}$$

which proves our claim.

8 Proof of Theorem 2

In this section, we present the proof of Theorem 2. To this end, we first present a generic bound on the regret of any algorithm and then use this bound to derive an upper bound on the regret of Algorithm 1 in terms of the step size γ. We then derive an upper bound on the constraint violation of Algorithm 1 in terms of the step size γ as well. Finally, choosing $\gamma = O(\frac{1}{\sqrt{T}})$, we obtain that both these regret and constraint violation bounds are $O(\sqrt{T})$.

We reiterate that our regret measure differs from the classical regret notion used in online learning, wherein the regret is defined based on the sub-optimality with respect to an optimal static action in hindsight (Hazan et al., 2016). In contrast, our regret measure adopts a more powerful oracle model, wherein the oracle can vary its actions across time steps since the demand itself is random. As a result, we mention that methods to analyse regret in the classical setting with respect to an optimal static action in hindsight do not naturally transfer over to the setting considered in this work.

8.1 Generic Bound on Regret

We first present an upper bound on the expected regret of any algorithm that we will use to derive upper bounds on the regret of Algorithm 1. The key technique used in our analysis is linear programming duality, as in (Li et al., 2020), and, in this proof, we also use the equilibrium property, specified in Definition 1, to establish a relationship between the primal and dual objectives, i.e., Objectives (2a) and (3a), respectively, for sub-optimal tolls $\boldsymbol{\tau}^{(t)}$ at each time step $t \in [T]$.

Lemma 1 (Generic Regret Bound)**.** *Consider an algorithm $\boldsymbol{\pi}$ that sets a sequence of tolls $\boldsymbol{\tau}^{(t)}$ and let $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ be the resulting equilibrium flows for each time period $t \in [T]$. Then, denoting $\boldsymbol{x}^t$ as the sequence of observed traffic flows corresponding to the equilibrium flows $\boldsymbol{f}^t$ for each time period $t \in [T]$, the regret*

$$R_T(\boldsymbol{\pi}) \leq \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{\tau}^{(t)} \cdot (\boldsymbol{c} - \boldsymbol{x}^t)\right].$$

Proof. To prove this claim, we first present a lower bound on the expected optimal objective $\mathbb{E}\left[U_t^*\right]$ at each time $t \in [T]$. Then, we present an upper bound on the expected regret accrued at each time $t \in [T]$ to obtain the desired regret upper bound.

To this end, first note that the optimal objective of the linear Program (2a)-(2d) is a lower bound on the optimal objective of Problem (1a)-(1d). Next, for each $t \in [T]$, let $\boldsymbol{f}^{t*}, \boldsymbol{f}_o^{t*}$ be the optimal solution of Problem (2a)-(2d) and $\boldsymbol{\tau}^{(t)*}$ be the optimal tolls. Then, observe for each $t \in [T]$ that

$$\begin{aligned}
\mathbb{E}\left[U_t^*\right] &= \mathbb{E}\left[\sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^{t*} + \lambda_u f_{o,u}^{t*} \right)\right], \\
&\overset{(a)}{=} \mathbb{E}\left[\sum_{u \in \mathcal{U}} \min\left\{ \min_{P \in \mathcal{P}_u^t} \left\{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)*} \right\}, \lambda_u \right\} - \sum_{e \in E} \tau_e^{(t)*} c_e \right], \\
&\overset{(b)}{\geq} \mathbb{E}\left[\sum_{u \in \mathcal{U}} \min\left\{ \min_{P \in \mathcal{P}_u^t} \left\{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \right\}, \lambda_u \right\} - \sum_{e \in E} \tau_e^{(t)} c_e \right],
\end{aligned}$$

where (a) follows by strong duality and (b) follows by the optimality of $\boldsymbol{\tau}^{(t)*}$ for the dual Problem (4a).

Next, we let $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ denote the vectors that encode the equilibrium flows under the tolls $\boldsymbol{\tau}^{(t)}$, i.e., the flow f_{Pu}^t ($f_{o,u}^t$) denotes whether user u was routed on P (the outside option) at time period t. Then, letting the objective $U_t = \sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right)$ be the total system cost incurred under the toll $\boldsymbol{\tau}^{(t)}$, we have from the above lower bound on the expected optimal objective $\mathbb{E}\left[U_t^*\right]$ that

$$\begin{aligned}
\mathbb{E}\left[U_t - U_t^*\right] &\leq \mathbb{E}\left[\sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right) - \sum_{u \in \mathcal{U}} \min\left\{ \min_{P \in \mathcal{P}_u^t} \left\{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \right\}, \lambda_u \right\} + \sum_{e \in E} \tau_e^{(t)} c_e \right], \\
&\overset{(a)}{=} \mathbb{E}\left[\sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right) + \sum_{e \in E} \tau_e^t x_e^t \right] \qquad (5) \\
&\quad + \mathbb{E}\left[- \sum_{u \in \mathcal{U}} \min\left\{ \min_{P \in \mathcal{P}_u^t} \left\{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \right\}, \lambda_u \right\} + \sum_{e \in E} \tau_e^{(t)} (c_e - x_e^t) \right], \\
&\overset{(b)}{=} \mathbb{E}\left[\sum_{e \in E} \tau_e^{(t)} (c_e - x_e^t)\right], \qquad (6)
\end{aligned}$$

where (a) follows by adding and subtracting the term $\sum_{e \in E} \tau_e^{(t)} x_e^t$, and (b) follows as

$$\sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right) + \sum_{e \in E} \tau_e^{(t)} x_e^t = \sum_{u \in \mathcal{U}} \min\{ \min_{P \in \mathcal{P}_u^t} \{ v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \}, \lambda_u \}, \qquad (7)$$

which we prove below. In particular, to see that the above equality is true, we begin by recalling that $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ are the equilibrium flows under the toll $\boldsymbol{\tau}^{(t)}$, i.e., $f_{o,u}^t > 0$ if and only if $\min\{\min_{P \in \mathcal{P}_u^t}\{v_u^t l_P + \sum_{e \in P} \tau_e^{(t)}\}, \lambda_u\} = \lambda_u$ and $f_{P'u}^t > 0$ for some path P' if and only if $\min\{\min_{P \in \mathcal{P}_u^t}\{v_u^t l_P + \sum_{e \in P} \tau_e^{(t)}\}, \lambda_u\} = v_u^t l_{P'} + \sum_{e \in P'} \tau_e^{(t)}$. Then, noting that $\sum_{P \in \mathcal{P}_u^t} f_{Pu}^t + f_{o,u}^t = 1$, it follows that

$$\min\{\min_{P \in \mathcal{P}_u^t}\{v_u^t l_P + \sum_{e \in P} \tau_e^{(t)}\}, \lambda_u\} = \sum_{P \in \mathcal{P}_u^t} \left(v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \right) f_{Pu}^t + \lambda_u f_{o,u}^t$$

for all users $u \in \mathcal{U}$. Next, summing the above relation over all users it follows that

$$\begin{aligned}
\sum_{u \in \mathcal{U}} \min\{\min_{P \in \mathcal{P}_u^t}\{v_u^t l_P + \sum_{e \in P} \tau_e^{(t)}\}, \lambda_u\} &= \sum_{u \in \mathcal{U}} \left(\sum_{P \in \mathcal{P}_u^t} \left(v_u^t l_P + \sum_{e \in P} \tau_e^{(t)} \right) f_{Pu}^t + \lambda_u f_{o,u}^t \right), \\
&= \sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right) + \sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u^t} \sum_{e \in P} \tau_e^{(t)} f_{P,u}^t, \\
&= \sum_{u \in \mathcal{U}} \left(v_u^t \sum_{P \in \mathcal{P}_u^t} l_P f_{Pu}^t + \lambda_u f_{o,u}^t \right) + \sum_{e \in E} \tau_e^{(t)} x_e^t,
\end{aligned}$$

where the last equality follows by noting that $\sum_{u \in \mathcal{U}} \sum_{P \in \mathcal{P}_u^t : e \in P} f_{Pu}^t = x_e^t$ by the edge decomposition of path flows. This proves the equality in Equation (7).

Finally, summing the inequality obtained in Equation (6) over all time periods $t \in [T]$, we get that

$$R_T(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=1}^T (U_t - U_t^*)\right] \leq \mathbb{E}\left[\sum_{t=1}^T \boldsymbol{\tau}^{(t)} \cdot (\boldsymbol{c} - \boldsymbol{x}^t)\right],$$

which proves our claim. □

We note that the above obtained bound in Lemma 1 presents an upper bound on the regret of any online algorithm and not the corresponding bound on the constraint violation, which we derive for Algorithm 1 in the next section. As a result, we reiterate that there are trivial algorithms, e.g., setting zero tolls, that can achieve a non-positive regret, as implied by the regret bound in Lemma 1; however, such algorithms will typically result in a large amount of constraint violation, which will scale as $O(T)$, i.e., linearly in the number of periods T. The key feature of our proposed algorithmic approach (Algorithm 1) is its ability to balance *both* regret and constraint violation to be of the order $O(\sqrt{T})$ in the setting of traffic routing.

8.2 Upper Bound on Regret of Algorithm 1

We now use the generic upper bound on the expected regret to show that the regret of Algorithm 1 is upper bounded by $O(\gamma T)$.

Lemma 2 (Upper Bound on Regret). *Let $\boldsymbol{x}^t$ be the sequence of observed traffic flows under the equilibrium flows $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ given the tolls $\boldsymbol{\tau}^{(t)}$ for each $t \in [T]$ under Algorithm 1. Then for Algorithm 1, the regret*

$$R_T(\boldsymbol{\pi}) \leq \gamma T \frac{|E|(\max_{e \in E} c_e + |\mathcal{U}|)^2}{2}.$$

Proof. To prove this claim, we present an upper bound on $\mathbb{E}\left[\sum_{t=1}^T \boldsymbol{\tau}^{(t)} \cdot (\boldsymbol{c} - \boldsymbol{x}^t)\right]$ and then use Lemma 1 to obtain a bound on the expected regret of Algorithm 1.

First observe from the toll update process that

$$\begin{aligned}
\left\|\boldsymbol{\tau}^{(t+1)}\right\|^2 &\leq \left\|\boldsymbol{\tau}^{(t)} - \gamma(\boldsymbol{c} - \boldsymbol{x}^t)\right\|^2, \\
&= \left\|\boldsymbol{\tau}^{(t)}\right\|^2 + \gamma^2 \left\|\boldsymbol{c} - \boldsymbol{x}^t\right\|^2 - 2\gamma \boldsymbol{\tau}^{(t)} \cdot (\boldsymbol{c} - \boldsymbol{x}^t).
\end{aligned}$$

Rearranging this equation, summing over $t \in [T]$ and taking expectations, we get that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{\tau}^{(t)} \cdot (\boldsymbol{c} - \boldsymbol{x}^t)\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{2\gamma}(\left\|\boldsymbol{\tau}^{(t)}\right\|^2 - \left\|\boldsymbol{\tau}^{(t+1)}\right\|^2) + \sum_{t=1}^{T} \frac{\gamma}{2}\left\|\boldsymbol{c} - \boldsymbol{x}^t\right\|^2\right], \\ &\leq \mathbb{E}\left[\frac{1}{2\gamma}\left\|\boldsymbol{\tau}^{(1)}\right\|^2 + \sum_{t=1}^{T} \frac{\gamma}{2}\left\|\boldsymbol{c} - \boldsymbol{x}^t\right\|^2\right], \\ &\overset{(a)}{=} \mathbb{E}\left[\sum_{t=1}^{T} \frac{\gamma}{2}\left\|\boldsymbol{c} - \boldsymbol{x}^t\right\|^2\right], \\ &\overset{(b)}{\leq} \gamma T \frac{|E|(\max_{e \in E} c_e + |\mathcal{U}|)^2}{2},\end{aligned}$$

where (a) follows since the initial toll $\boldsymbol{\tau}^{(1)} = \mathbf{0}$, and (b) follows as $|c_e - x_e^t| \leq \max_{e \in E} c_e + |\mathcal{U}|$. Finally, applying Lemma 1, we have proved the desired bound on the regret of Algorithm 1. □

8.3 Upper Bound on Constraint Violation of Algorithm 1

We now establish an upper bound on the constraint violation of Algorithm 1 in terms of the step size γ. To this end, we first show that the expected constraint violation of Algorithm 1 is upper bounded by $O\left(\frac{1}{\gamma}\mathbb{E}\left[\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2\right]\right)$ in Lemma 3. Then, in Lemma 4, we show that the tolls remain bounded at each time step to establish that the constraint violation of Algorithm 1 is bounded by $O(\frac{1}{\gamma})$.

Upper Bound on Expected Constraint Violation in terms of Step Size γ: We first show that the expected constraint violation of Algorithm 1 is upper bounded by $O\left(\frac{1}{\gamma}\mathbb{E}\left[\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2\right]\right)$.

Lemma 3 (Constraint Violation Bound). *Let $\boldsymbol{x}^t$ be the sequence of observed traffic flows under the equilibrium flows $\boldsymbol{f}^t, \boldsymbol{f}_o^t$ given the tolls $\boldsymbol{\tau}^{(t)}$ for each $t \in [T]$ under Algorithm 1. Then for Algorithm 1, the constraint violation*

$$V_T(\boldsymbol{\pi}) \leq \frac{1}{\gamma}\mathbb{E}\left[\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2\right].$$

Proof. By the toll update process, we know that

$$\boldsymbol{\tau}^{(t+1)} \geq \boldsymbol{\tau}^{(t)} - \gamma(\boldsymbol{c} - \boldsymbol{x}^t).$$

Rearranging the above equation and summing over $t \in [T]$, we get that

$$\sum_{t=1}^{T}(\boldsymbol{x}^t - \boldsymbol{c}) \leq \frac{1}{\gamma}\sum_{t=1}^{T}(\boldsymbol{\tau}^{(t+1)} - \boldsymbol{\tau}^{(t)}) \leq \frac{1}{\gamma}\boldsymbol{\tau}^{(T+1)}.$$

From this, we obtain that the expected constraint violation

$$V_T(\boldsymbol{\pi}) = \mathbb{E}\left[\left\|\left(\sum_{t=1}^{T}(\boldsymbol{x}^t - \boldsymbol{c})\right)_+\right\|_2\right] \leq \frac{1}{\gamma}\mathbb{E}\left[\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2\right],$$

which proves our claim. □

Boundedness of Tolls: Since the constraint violation is bounded by $\frac{1}{\gamma}\mathbb{E}\left[\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2\right]$, we seek an upper bound on the toll at time $T+1$ to obtain an upper bound for the constraint violation. In particular, we show that the tolls are bounded by a constant and thus the toll on any edge does not increase with the number of time periods T.

Lemma 4 (Boundedness of Tolls). *Under Algorithm 1, the toll on any given edge at each time step t is upper bounded by $\max_{u \in \mathcal{U}}\{\lambda_u\} + \max_{e \in E}\{c_e\} + |\mathcal{U}|$ for any step-size $\gamma \leq 1$.*

Proof. To prove this claim, first note that the toll $\tau_e^{(1)} = 0$ for all edges and suppose that $\tau_e^{(t)} > \max_{u \in \mathcal{U}}\{\lambda_u\}$. Then, it is clear that $x_e^t = 0$ and the toll on this edge must reduce in the next time step as users can use the outside option and incur a cost of λ_u instead. Next, if $\tau_e^{(t)} \leq \max_{u \in \mathcal{U}}\{\lambda_u\}$, then it must hold that $\tau_e^{(t+1)} \leq \tau_e^{(t)} + \gamma(c_e + |U|) \leq \max_{u \in \mathcal{U}}\{\lambda_u\} + \max_{e \in E}\{c_e\} + |\mathcal{U}|$, which proves our claim. □

8.4 Square Root Bound on Regret and Constraint Violation

From Lemma 2 we observed that the regret is upper bounded by $\gamma T \frac{|E|(\max_{e \in E} c_e + |\mathcal{U}|)^2}{2}$, $i.e.$, $O(\gamma T)$, and from Lemmas 3 and 4 we have that the expected constraint violation is upper bounded by $|E|(\max_{u \in \mathcal{U}} \lambda_u + \max_{e \in E} c_e + |\mathcal{U}|)\frac{1}{\gamma}$, $i.e.$, $O(\frac{1}{\gamma})$, since $\left\|\boldsymbol{\tau}^{(T+1)}\right\|_2$ is bounded by a constant. Setting $\gamma T = \frac{1}{\gamma}$, we have that both the upper bounds on regret and constraint violation are minimized when $\gamma = \frac{1}{\sqrt{T}}$ as in the statement of Theorem 2. Finally, taking $\gamma = \frac{1}{\sqrt{T}}$, it is clear that the expected regret $R_T(\boldsymbol{\pi}) \leq \frac{|E|(\max_{e \in E} c_e + |\mathcal{U}|)^2}{2}\sqrt{T}$ and that the expected constraint violation $V_T(\boldsymbol{\pi}) \leq |E|(\max_{u \in \mathcal{U}} \lambda_u + \max_{e \in E} c_e + |\mathcal{U}|)\sqrt{T}$, which proves Theorem 2.

9 Proof of Theorem 3

Consider a two edge network, where the edge $e_1 = (s_1, t_1)$ is between the origin s_1 and destination t_1, respectively, and the edge e_2 is between the origin s_2 and destination t_2, respectively. Furthermore, both the edges have a capacity of one, and edge e_1 has a travel time of $l_1 = 1$ while edge e_2 has a travel time of $l_2 = 0$. Next, consider a population of one user, i.e., $|\mathcal{U}| = 2$, and a distribution $\mathcal{D}$ such that users have a fixed value of time of $v_1 = 1$ (and the cost of outside option of $\lambda_1 = 2$) but traverse the O-D pair (s_1, t_1) with probability 0.5 and the O-D pair (s_2, t_2) with probability 0.5. We term the first group of users as type I and the users with parameters $v_2 = 0$ and $c_2 = 0$ as type II users.

Given the above defined instance, first observe that over a time horizon of T days, the expected optimal total system cost is $\frac{T}{2}$, as users will traverse each O-D pair with equal probability in expectation. To derive the $\Omega(\sqrt{T})$ regret lower bound, we suppose that S_1 users of type I arrive over the time horizon T. Then, it holds that the expected regret is given by

$$\text{Regret} = \mathbb{E}\left[\max\left\{S_1, \frac{T}{2}\right\} - T\right] = \mathbb{E}\left[(S_1 - \frac{T}{2})_+\right].$$

Finally, from the central limit theorem, it is clear that the regret is $\Omega(\sqrt{T})$.

10 Depiction of Toll Update Procedure in Algorithm 1

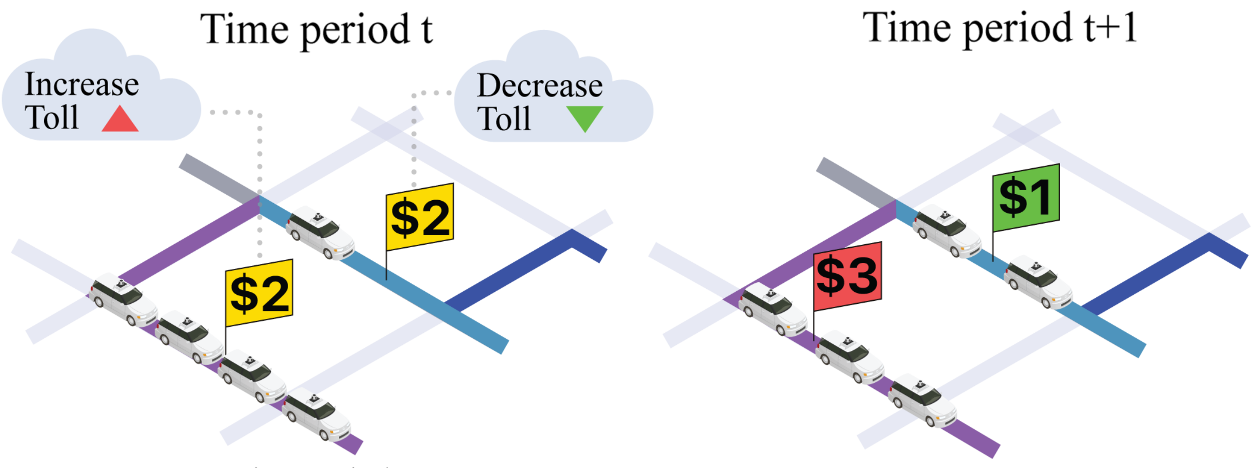


Figure 3: Toll update step in Algorithm 1. Between subsequent time periods, the toll on each edge is increased (decreased) if there is more (less) flow than the capacity of that edge.

11 Numerical Experiments

11.1 Numerical Implementation Details

In this section, we describe the implementation details of Algorithm 1 and the benchmarks introduced in Section 5.1, which we test on the Sioux Falls network (one of the most commonly used data-sets in the traffic routing literature), depicted in Figure 4, obtained from (Transportation Networks for Research Core Team). The data set contains both origin-destination travel information for all users and information on the capacity, length, and maximum speed of every road in the traffic network. To obtain the travel time of every edge, we assume that vehicles travel at the maximum speed for that edge. Furthermore, we scale the total user demand by a factor of 0.5 to ensure feasibility of the linear Program (1a)-(1d).

Since the computational complexity of solving the linear Program (1a)-(1d) scales with the number of users, we group users with the same origin-destination pairs to have the same values-of-time to improve the computational tractability of the resulting optimization problems. For the experiments, we further assume that the value-of-time for users scales proportionally to their incomes. In Sioux Falls, the range of people's incomes ranges from below $\$10,000$ to over $\$200,000$ (SFI, 2022), which amounts to a value-of-time range from about $\$5/\text{hr}$ to $\$100/\text{hr}$, assuming 40 hours of work a week for 50 working weeks. Then, for each user group g, we draw their mean value-of-time, denoted by μ_g, uniformly at random between the range $\$5/\text{hr}$ and $\$100/\text{hr}$. We further assume that at each period, all users from a group have a value-of-time drawn from a uniform distribution over the interval $[0.8\mu_g, 1.2\mu_g]$. The O-D pair for every user is drawn from a distribution defined as follows: with probability 0.8, the user travels on their default O-D pair, as described in the data set, and with probability 0.2, the user chooses an O-D pair uniformly at random from the space of all possible O-D pairs. In Algorithm 1, we set the step-size $\gamma = \frac{5\times10^{-4}}{\sqrt{T}}$. We reiterate that the square root regret guarantee for Algorithm 1 would hold by Theorem 2 for any choice of $\gamma = \frac{D}{\sqrt{T}}$ for any constant $D \in (0, 1]$. For our particular problem instance, the choice of $\gamma = \frac{5\times10^{-4}}{\sqrt{T}}$ demonstrated fast convergence and thus we used this for our numerical experiments, though we mention that choosing any constant $D \in (0, 1]$ would have led to similar results. Finally, to break ties among equivalent minimum cost routes for users, we add a noise to the Population Mean VoT and User Mean VoT tolls every time step drawn from the uniform distribution in $[-5 \times 10^{-4}, 5 \times 10^{-4}]$.

To efficiently implement the outside option we consider a modified network with additional edges between the corresponding O-D pairs for each user group. We set the capacity of these edges to be strictly higher than the total demand between the corresponding O-D pair. Furthermore, we set the travel time of these edges to be 1.5 times the cost, including travel times and tolls, of the shortest path under the optimal tolls. We mention that to improve computational tractability, we club the outside options for all users belonging to the same group (i.e., having the same origin-destination pair) into one edge. In this modified network, each user must traverse one path, which may be a path in the original graph or on the added edge representing the outside option for that user. Finally, note that the tolls on the added edges must be zero for all tested algorithms since the maximum possible flow on any of the added edges will be lower than the edge capacities by construction.

All our experiments are carried out on a 2019 MacBook Pro, with 32 GB RAM and a 2.4 GHz 8-Core Intel Core i9 processor. The program is written in Python 3.7, and we use Gurobi (free academic license) for solving the optimization problems. All the basemaps for the Sioux Falls region are obtained from Open Street Maps using the `contextily` package. The versions of all packages used in the codes are detailed in the *requirements.txt* file in the repository and our code is publicly available at https://github.com/StanfordASL/online-tolls.

We note that all our experiments are designed to evaluate the asymptotic behaviour of our toll update procedure. Hence, due to the large number of samples at the asymptotic limits, the results are not sensitive to random seeds that determine the realizations of the user values of time. Nevertheless for technical correctness and reproducibility, we have set a random seed of five for generating user values of time.

11.2 Properties of Computed Tolls

In this section, we demonstrate the practical efficacy of Algorithm 1 by investigating the properties of the tolls set in the traffic network using Algorithm 1 and the benchmarks. To this end, Figure 5 depicts the tolls set by Algorithm 1 and the three benchmarks at $T = 1000$. From this figure, we observe that the tolls are typically placed either on roads in the dense urban areas on the center-right of the Sioux Falls network or on roads on the left and bottom of the network that have smaller road capacities. We further note that the set tolls were about $\$0.75$ on average for the two dynamic algorithms and the User Mean VoT benchmark, with the maximum toll for the algorithms ranging between \$2.4-\$2.7, which is in

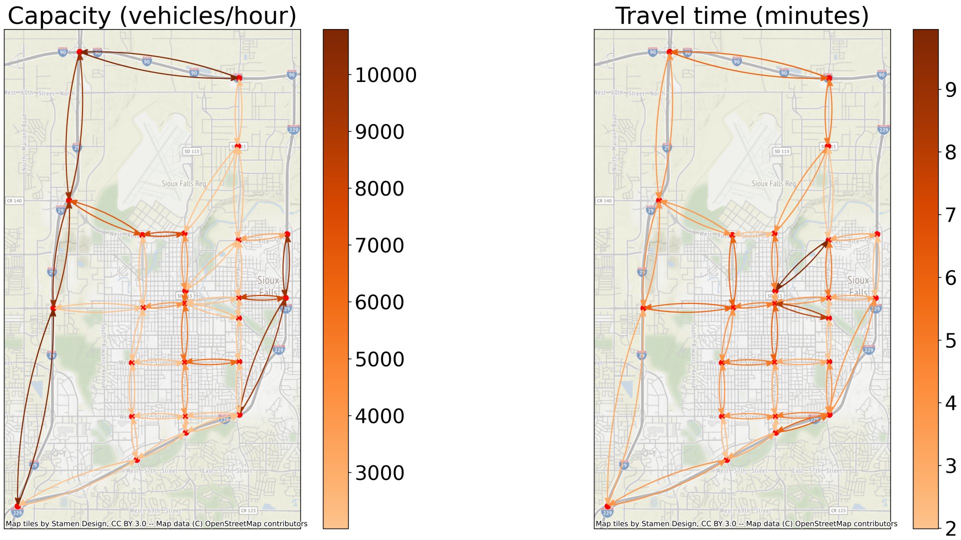

Capacity (vehicles/hour)
10000
9000
8000
7000
6000
5000
4000
3000
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors
Travel time (minutes)
9
8
7
6
5
4
3
2
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

Figure 4: Depiction of the Sioux Falls road network. The capacities on each edge of the network are depicted on the left, while the travel times are depicted on the right.

alignment with the order of magnitude of typical tolls in real-world transportation networks.

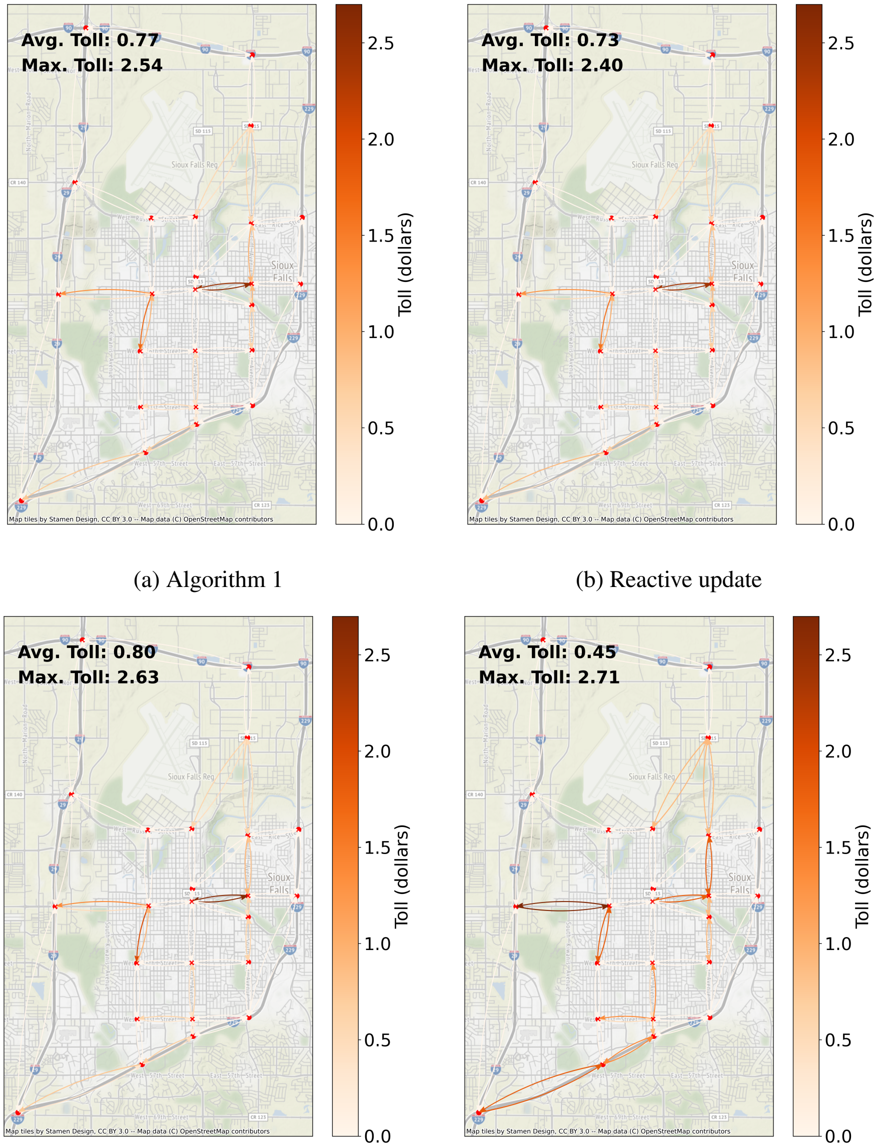
In Figure 6, we present a histogram representing the edge tolls corresponding to the Algorithm 1 at $T = 1000$. This histogram indicates that most road tolls are zero, which corresponds to traditional congestion pricing schemes that operate on only certain roads or regions of the traffic network. Furthermore, we observe that most non-zero tolls are in the range of \$0.25-\$1.00, while only about five percent of the edges have road tolls that exceed \$1.

Finally, in Figure 7, we depict the evolution of the cumulative tolls, i.e., the sum of the tolls on all edges, over the $T = 1000$ time horizon. This figure illustrates that both the dynamic tolling algorithms, i.e., Algorithm 1 and the Reactive Toll Update benchmark, approach a set of road tolls in a small number of periods, after which the tolls oscillate to achieve a good balance between constraint violation and regret. Note that the Reactive Toll Update benchmark stabilizes to a small range of toll values earlier with its constant toll updates as compared to Algorithm 1 since the step size of the updates of Algorithm 1 is of the order $O(\frac{1}{\sqrt{1000}})$ for $T = 1000$. However, due to the constant updates in the tolls at each time step, the Reactive Toll Update benchmark also has a greater degree of variability in its tolls after arriving at a stable value for the cumulative tolls.

12 Societal Impact

In this work, we proposed an online learning approach to set tolls in a traffic network to induce users with different values of time toward a system-optimum traffic pattern. We believe that our proposed algorithmic approach is practically viable for several reasons. First, it only relies on access to the aggregate flows on the roads of the traffic network and does not rely on any information on the values of time, O-D pairs, or the path taken by users in the system. In particular, our approach achieves good performance on both regret and constraint violation metrics without compromising user privacy or seeking potentially sensitive data. Next, our toll update mechanism is very computationally inexpensive as the computational complexity of Phase II of Algorithm 1 is only $O(|E|)$, which further motivates the practical viability of our online learning method. Furthermore, Algorithm 1 is very intuitive from the perspective of a central planner seeking to deploy road tolling mechanisms, as it involves a very intuitive toll update step. In particular, the toll is increased on the roads if the flow on those roads exceeds capacity and is decreased otherwise. Finally, we note that beyond Algorithm 1, even the reactive toll updates benchmark achieved good numerical performance. We reiterate that reactive toll updates can be a natural strategy in large-scale traffic networks, wherein tolls can only be adjusted between periods in constant increments.

Beyond the applicability of our proposed learning algorithm, we note that our theoretical results apply in the setting of capacity-constrained road networks, which, as mentioned in Section 3.1, are largely consistent with the operation of real-



(a) Algorithm 1

(b) Reactive update

(c) User mean VoT

(d) Population mean VoT

Figure 5: Tolls corresponding to Algorithm 1 and the three benchmark approaches in the Sioux Falls traffic network at $T = 1000$.

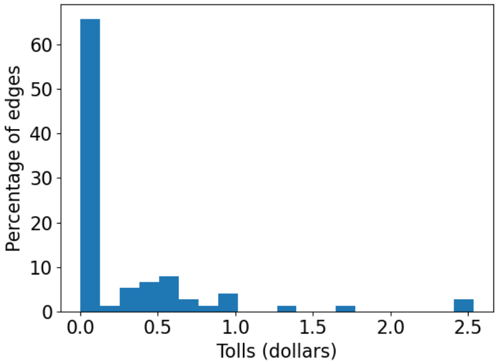


Figure 6: Histogram of edge tolls in the Sioux Falls traffic network for Algorithm 1 at T=1000.

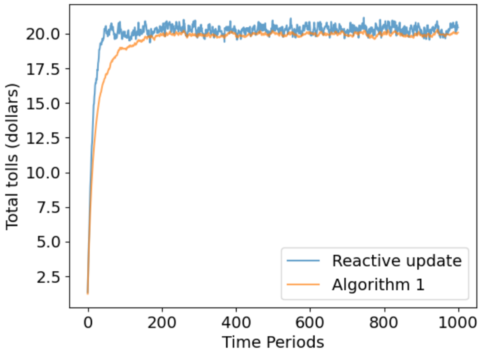


Figure 7: Time evolution of cumulative tolls on all edges of the Sioux Falls traffic network for Algorithm 1 and the Reactive Toll Update benchmark for $T = 1000$.

world road networks. However, we also demonstrate how our proposed online learning approach can generalize to the context of congestion games, when the travel time on each edge is a function of its flow, in Appendix **??** through numerical experiments.

While our proposed approach has several benefits in terms of practical viability, several considerations must be taken into account for the real-world implementation of our mechanism. In particular, our focus in this work is on optimizing for system efficiency. However, there may be other metrics beyond efficiency, e.g., revenue maximization or fairness, that a central planner may want to optimize. Furthermore, our theoretical results apply in the setting when the values of time and O-D pairs of users are drawn i.i.d. from some distribution. While this assumption may reflect real-world traffic settings during rush hour periods as users commute to and from work, such an assumption may not hold in other contexts, e.g., during a sporting event with sudden surges in demand to particular locations. In addition, to obtain our theoretical guarantees, we do not impose any lower or upper bounds on the tolls that can be set on different roads on the traffic network. Such lower or upper bounds on the tolls may be a practical consideration in the real-world deployment of tolling schemes, as some roads often cannot be tolled while there is often an upper bound on the tolls that can be placed on other roads. As a result, a central planner must give detailed thought regarding the extent to which the modeling assumptions used for the theoretical analysis apply in practice, as this will have a bearing on the efficacy of Algorithm 1.