Online Learning for Traffic Navigation in Congested Networks

Sreenivas Gollapudi SGOLLAPU@GOOGLE.COM
Google Research

Kostas Kollias KOSTASKOLLIAS@GOOGLE.COM
Google Research

Chinmay Maheshwari CHINMAY_MAHESHWARI@BERKELEY.EDU
University of California, Berkeley, EECS

Manxi Wu MANXIWU@CORNELL.EDU
Cornell University, Operations Research and Information Engineering

Editors: Shipra Agrawal and Francesco Orabona

Abstract

We develop an online learning algorithm for a navigation platform to route travelers in a congested network with multiple origin-destination (o-d) pairs while simultaneously learning unknown cost functions of road segments (edges) using the crowd-sourced data. The number of travel requests is randomly realized, and the travel time of each edge is stochastically distributed with the mean being a linear function that increases with the edge load (the number of travelers who take the edge). In each step of our algorithm, the platform updates the estimates of cost function parameters using the collected travel time data, and maintains a rectangular confidence interval of each parameter. The platform then routes travelers in the next step using an optimistic strategy based on the lower bound of the up-to-date confidence interval. The key aspects of our setting include (i) the size and the spatial distribution of collected travel time data depend on travelers' routing strategies; (ii) we evaluate the regret of our algorithm for platforms with different objectives, ranging from minimizing the social cost to minimizing the individual cost of self-interested users. We prove that the regret upper bound of our algorithm is $O(\sqrt{T}\log(T)|E|)$, where T is the time horizon, and $|E|$ is the number of edges in the network. Furthermore, we show that the regret bound decreases as the number of travelers increases, which implies that the platform learns faster with a larger user population. Finally, we implement our algorithm on the network of New York City, and demonstrate the efficacy of the proposed algorithm.

Keywords: Online learning, Congestion games, Traffic networks, Regret analysis

1. Introduction

In recent years, travelers are increasingly relying on navigation platforms to learn the traffic conditions, and make routing decisions. The quality of the navigation service heavily depends on the platform's ability to collect travel time data from their users in order to train machine learning algorithms that accurately predict the travel time. However, the travel time data is only available in regions where the users are routed through, and thus the spatial distribution of the data is in turn governed by the routing strategy. *How can the platform efficiently learn the network cost functions from users' data, and effectively route a large number of travelers in a congested network?* To answer this question, we provide a repeated routing model, and an online learning algorithm.

We consider a congested network with multiple origin - destination (o-d) pairs. In every time step, a finite number of routing requests is randomly generated between each o-d pair. The platform

computes a routing strategy based on the received requests, and collects the realized travel time data from each user on their taken edges. The time cost of each edge equals to a linearly increasing function of the edge load – the number of travelers who take that edge – plus an independently and identically distributed random noise with zero mean. In each step, the number of data points collected by the platform of each edge equals to the number of travelers who take it. The platform does not have data on edges that are not taken by any traveler.

When deciding the routing strategy, the platform faces a trade-off between achieving efficiency for the society versus fairness among all their users (Kleinberg et al. (1999); Bertsimas et al. (2011); Jahn et al. (2005); Schulz and Stier-Moses (2006); Jalota et al. (2021)). On one hand, the platform can adopt a socially optimal routing in order to minimize the average cost of all travelers. On the other hand, the platform can implement the equilibrium routing that is fair for all travelers so that no one has an incentive to deviate. Due to the congestion effect, the equilibrium routing strategy is different from the socially optimal routing, and the efficiency gap is characterized by the notion of price of anarchy (Roughgarden (2002); Jahn et al. (2005); Roughgarden (2005)). In our setting, we consider the socially optimal routing, equilibrium routing, as well as a combined routing strategy for the platform. The combined routing strategy is characterized by a parameter that allows the platform to determine the relative weight between the goal of efficiency and fairness.

We propose an online learning algorithm for the platform to learn the estimates of edge cost parameters using the collected travel time data, and repeatedly update the routing strategy given the randomly realized travel demand (Algorithm 1). In each time step of the algorithm, the algorithm updates *(i)* a regularized ordinary least square estimate of the cost parameters (i.e. intercept and slope) of edge cost functions; *(ii)* a confidence interval that is independently constructed for each cost parameter of each edge; *(iii)* an optimistic routing strategy computed using the lower bound of each parameter in their confidence interval. We note that the parameter estimates and confidence intervals in *(i)* and *(ii)* are only updated for edges that are taken in each step. Moreover, the strategy in *(iii)* is computed as a socially optimal strategy, an equilibrium strategy, or a combined strategy depending on the goal of the platform.

The regret of our algorithm is defined as the accumulated gap between the value of the platform's objective function with the stage routing strategy and the optimal value if the platform were to know the true cost parameter of all edges. Since we allow for fluctuation of travel demand, and the platform's objective is in a continuous spectrum between efficiency and fairness, our regret in each stage is averaged over the number of travelers, and is parametrized by the efficiency-fairness trade-off parameter. We present an upper bound of the accumulated regret of our algorithm (Theorem 6). Our regret bound is $O(\sqrt{T}\log(T)|E|)$, where T is the time horizon, and $|E|$ is the number of edges in the network. This implies that the regret sub-linearly scales with the time steps, and linearly scales with the network size.

Interestingly, we find that the bound on the regret decreases with $\underline{N}$, where $\underline{N}$ is the minimum number of travelers per step increases. Therefore, the platform learns faster, and achieves lower regret for each user when the user population increases. This result is aligned with the practical observation that platforms with higher market share in a region often achieves high accuracy in travel time prediction thanks to their accessibility to a larger data set of travel time. Furthermore, we also notice that the advantage of user size decreases as the platform increases its relative weight on the goal of achieving fairness. This is because as the number of travelers increase, the congestion effect caused by selfish routing behavior becomes more prominent.

Our algorithm and regret analysis is inspired by the stochastic linear bandits literature (Dani et al. (2008); Rusmevichientong and Tsitsiklis (2010); Abbasi-Yadkori et al. (2011); Lattimore and Szepesvári (2020); Agrawal and Goyal (2013); Abeille and Lazaric (2017)). Both our algorithm and the classical learning algorithms for linear bandits maintains a confidence set of the unknown linear cost/reward function parameters and update strategies according to an optimistic decision rule. We emphasize that our setting, algorithm, and analysis approach are different from the classical linear bandit literature in the following three aspects:

(1) In our setting, the number of data on the realized travel time cost of each road segment equals to the edge load induced by the routing strategy (i.e. the covariate of the linear regression of each edge cost function). This is in contrast to the setting of stochastic linear bandit, where only one data point is collected every time step regardless of the selected strategy. As a result, our computation of the regularized OLS estimate and the bound of the probability of "clean event" is different from that in the linear bandit setting (Lemmas 2, 3, and 4).

(2) Our algorithm constructs a rectangular confidence set for each edge cost parameters that is the product of the independent confidence intervals of the intercept and slope parameters. This is inspired by Dani et al. (2008), and is in contrast to the approach of constructing an ellipse confidence set that is joint for all linear parameters. It is well known that computing the optimistic estimate and optimistic strategy with ellipse confidence set is computationally expensive (Lattimore and Szepesvári (2020)). Our approach of constructing rectangular confidence set allows us to directly compute the optimistic routing strategy with respect to the lower bound of each parameter (Lemma 5) This significantly simplifies the computation especially in large networks with many edges.

(3) In our problem, the objective function of regret minimization is not the same function that provides bandit feedback. Specifically, the objective function of the routing platform is quadratic average cost function, which depends on network topology, the cost estimates of every edge and the fluctuating total demand. On the other hand, the collected cost data is generated by the linear cost function of each edge that is taken. This is different compared with the majority of bandit literature, where the stochastic reward used in parameter estimate is generated by the same objective function of regret minimization.

Additionally, the online congestion routing problem studied in our work is complementary to the extensively studied online shortest path problem, which falls in the broad category of online linear optimization (Hannan (1957); Kalai and Vempala (2005); Lai et al. (1985); Littlestone and Warmuth (1994); Freund and Schapire (1997); Auer et al. (2002)) and combinatorial bandit problem (Cesa-Bianchi and Lugosi (2012); Gai et al. (2012); Kveton et al. (2015); Wen et al. (2015)) in networked settings. The online shortest part problem addresses the problem with finding the shortest path in a network with edge costs that have constant but unknown means. Specifically, the literature on online shortest path problem can be divided in two categories based on whether or not the randomness of edge costs is adversarial McMahan and Blum (2004); Dani and Hayes (2006); Awerbuch and Kleinberg (2008); Cesa-Bianchi and Lugosi (2012), or stochastic Gai et al. (2012); Combes et al. (2015); Talebi et al. (2017). Recently, the papers Levine et al. (2017); Pike-Burke and Grunewalder (2019); Awasthi et al. (2022) considers edges costs that are changing across time steps depending on the number of times being used. The problem considered in our paper is different from the online shortest path problem in that we consider a congested network with linearly increasing edge

cost functions, and simultaneously routing potentially a large number of travelers with fluctuating demand and different o-d pairs. Consequently, our technical approach is also different from that adopted in online shortest path problem.

More broadly, our online learning algorithm when adopting the equilibrium routing strategy in each stage is related to the literature on learning Nash equilibrium or Coarse Correlated equilibrium in routing games (Blum et al. (2006); Kleinberg et al. (2009); Krichene et al. (2014); Cominetti et al. (2010)). Our setup is different from these papers in that the traveler set in our problem changes from step to step, and thus the dynamics must be facilitated by a platform that collect travel time data to learn the parametrized cost functions. The papers Meigs et al. (2017) and Wu and Amin (2019) studied the similar setting of learning facilitated by navigation platforms for reaching equilibrium, but their focus is the asymptotic properties in non-atomic routing games. These papers do not provide finite-time regret analysis.

Finally, we test our learning analysis using a real-world example of routing travelers in Manhattan, New York City. We implement learning with both equilibrium routing and socially optimal routing. We demonstrate that our algorithm is computationally feasible even when the network scale is large. The algorithm efficiently learns the true edge cost parameters, and the equilibrium/optimal routing strategy. The numerical experiment is consistent with our theoretical regret bound: We observed that the accumulated regret scales sub-linearly with the number of stages, and the accumulated regret with respect to socially optimal routing is smaller than that with equilibrium routing.

2. The Model

Consider a traffic network $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where $\mathcal{E}$ is the set of edges and $\mathcal{V}$ is the set of vertices. Each edge $e \in \mathcal{E}$ has a finite integer capacity $d_e \in \mathbb{N}_+$.[1] The network has a set of I origin-destination (o-d) pairs, where each o-d pair $i \in I$ is connected by route set $\mathcal{R}_i$. The set of all routes is $\mathcal{R} = \cup_{i \in I} \mathcal{R}_i$

A navigation platform repeatedly provide route recommendations to travelers in discrete time steps $t = 1, 2, \ldots, T$.[2] In each step t, a finite set of agents N_i^t request route recommendation to travel between each o-d pair i. The number of travelers N_i^t are randomly realized (and can be correlated) across steps and o-d pairs with a positive lower bound $\underline{N} \in \mathbb{R}_+$.[3] The platform route travelers according to a strategy $q^t = (q_r^t)_{r \in \mathcal{R}}$, where q_r^t is the number of travelers sent to take route r in step t. A routing strategy is feasible if it satisfies the following constraints:

$$\sum_{r \in \mathcal{R}_i} q_r^t = N_i^t, \quad \forall i \in I, \quad \sum_{r \ni e} q_r^t \leq d_e, \quad \forall e \in \mathcal{E}, \quad q_r^t \geq 0, \quad \forall r \in \mathcal{R}. \tag{1}$$

The constraints in (1) ensure that the demand of every origin-destination pair i and every time step t is routed; the load of each edge $e \in \mathcal{E}$ induced by q^t does not exceed the edge capacity; and the flow on each route is non-negative. We assume that the number of travelers requesting the trip is

1. The finite edge capacity ensures that the cost of each edge is bounded. This is a standard assumption in bandit literature – the cost/reward of edge arm is upper bounded. The setting of capacity is without loss of generality when the network capacity is higher than the number of travelers.
2. The time steps in our model can be viewed as service batches such that the travelers in the same batch impose congestion externality on each other, but there is no congestion effect across batches (e.g. travelers in different batches have distance in between).
3. When $\underline{N} = 1$, then the platform only routes one traveler at a time, and there is no congestion effect. In practice, $\underline{N}$ is typically a large number.

below the capacity of the network in the sense that there always exists a feasible routing strategy q^t that satisfies (1). We denote the set of all feasible strategies in stage t as Q^t.

Given a routing strategy $q^t \in Q^t$, the edge load on each edge $e \in \mathcal{E}$ is $w_e^t = \sum_{r \ni e} q_r^t$. The expected driving time cost of edge e is $\ell_e(w_e^t) = \boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} w_e^t$, where $\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1} > 0$ is the free flow travel time, and the slop of edge e, respectively.[4] We denote the cost parameter vector of each edge e as $\boldsymbol{\theta}_e = (\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1})$.[5,6] Each edge e is congestible in that the average travel time increases in the load on the edge. In each step t, the driving time experienced by traveler $k \in [w_e^t]$ on edge e is given by:

$$c_{e,k}^t = \ell_e(w_e^t) + \epsilon_{e,k}^t, \quad \forall k \in [w_e^t], \tag{2}$$

where $\epsilon_{e,k}^t$ is a $1-$sub-Gaussian random variable, and is independently and identically distributed across all w_e^t travelers.[7] The platform collects data of realized driving time of each traveler on all taken edges, denoted as $c^t = (c_{e,k}^t)_{k \in [w_e^t], e \in E^t}$, where $E^t = \{e \in \mathcal{E} | w_e^t > 0\}$ is the set of edges taken by positive number of travelers. The number of data points collected by the platform on an edge equals to the edge load, thus no data is available on edges that are not taken by any travelers. We consider the following three types of routing strategies for the platform:

1. Socially optimal routing. The platform aims at minimizing the average cost of all travelers. Given N^t, the *socially optimal routing* strategy q^{t*} is an optimal solution of the following convex optimization problem:

$$\min_{q^t} C(q^t) = \frac{1}{N^t} \sum_{e \in \mathcal{E}} \ell_e(w_e^t) w_e^t, \quad s.t. \quad w_e^t = \sum_{r \ni e} q_r^t, \quad q^t \in Q^t. \tag{3}$$

2. Equilibrium routing. The platform aims at inducing an equilibrium $q^{t\dagger}$ (the Beckmann user equilibrium defined in Correa et al. (2004)). That is, no o-d pair has an unsaturated route (i.e. a route is unsaturated if the number of travelers who take it is smaller than the capacity) with strictly smaller cost than any route used by travelers:

$$\forall i \in I, \ \forall r \in \mathcal{R}_i, \ q_r^{t\dagger} > 0, \ \Rightarrow \ \sum_{e \in r} \ell_e(w_e^{t\dagger}) \leq \sum_{e \in r'} \ell_e(w_e^{t\dagger}), \quad \forall r' \in \{\mathcal{R}_i | \sum_{r \ni e} q_r^t < d_e, \forall e \in r\},$$

where $w^{t\dagger} = (w_e^{t\dagger})_{e \in \mathcal{E}}$ is the equilibrium edge load vector induced by $q^{t\dagger}$. Thus, no traveler has an incentive to deviate from the taken route. Given N^t, the equilibrium routing strategy $q^{t\dagger}$ can be computed by solving the following convex optimization problem that minimizes the potential function of the routing game (Monderer and Shapley (1996); Roughgarden (2005)):

$$\min_{q^t} \Phi(q^t) = \sum_{e \in \mathcal{E}} \sum_{j=1}^{w_e^t} \ell_e(j), \quad s.t. \quad w_e^t = \sum_{r \ni e} q_r^t, \quad q^t \in Q^t.$$

4. Our algorithm and analysis can be extended to polynomial cost functions including the widely adopted Bureau of Public Roads (BPR) latency functions (Dafermos and Sparrow (1969)).
5. We use the bold text to distinguish the true cost parameter $\boldsymbol{\theta}$ from a generic cost parameter θ.
6. Our model accounts for the change of costs due to the fluctuation of demand, but assumes that the physical environment of road network captured by the average edge cost functions remain unchanged. We leave the nonstationary environment as an interesting direction for future works.
7. The mean latency function $\ell_e(\cdot)$ captures the interdependence of travel times and the congestion effect experienced by travelers who use the same road.

Remark 1 *For any feasible routing strategy q^t, the maximum reduction of any traveler's cost of deviating from the route that they take in q^t (i.e. the individual regret) is $\max_{\{r,r'\in\mathcal{R}|q_r^t>0\}} \sum_{e\in r} \ell_e(w_e^t) - \sum_{e\in r} \ell_e(w_e^{t'})$, where $w_e^{t'} = w_e^t$ for $e \in r' \setminus r$ and $w_e^{t'} = w_e^t$ for $e \in r \cap r'$. We can show that this individual regret is upper bounded by the gap between $\Phi(q^t)$ and $\phi(q^{t\dagger})$:*

$$\begin{aligned}\max_{\{r,r'\in\mathcal{R}|q_r^t>0\}} \sum_{e\in r} \ell_e(w_e^t) - \sum_{e\in r} \ell_e(w_e^{t'}) = \Phi(q^t) - \Phi(q^{t'}) \le \Phi(q^t) - \min_{q^t} \Phi(q^t) \\ = \Phi(q^t) - \Phi(q^{t\dagger}),\end{aligned} \tag{4}$$

where $q^{t'}$ is the routing strategy with one traveler deviating from route r to r' and the remaining travelers do not change their route choices. Thus, in (4), the first equality is derived from the fact that the routing game is a potential game with potential function Φ. The last equality arises from the fact that the equilibrium routing strategy is the minimizer of the potential function.

3. Combined routing. The platform aims at striking a balance between minimizing the average social cost (i.e. efficiency) and minimizing the equilibrium gap of individual travelers (i.e. fairness). The combined routing strategy $q^{t,\xi}$ is computed by solving the following convex optimization problem that minimizes the linear combination between the average social cost function and the potential function, where $\xi \in (0,1)$ governs the trade-off between efficiency and fairness:

$$\min_{q^t} \ \Psi(\xi, q^t) = \xi C(q^t) + (1-\xi)\Phi(q^t), \quad s.t. \quad q^t \in Q^t.$$

We note that when $\xi = 1$ (resp. $\xi = 0$), the combined routing strategy $q^{t,\xi}$ reduces to the socially optimal routing q^{t*} (resp. the equilibrium routing $q^{t\dagger}$).

In our setting, the platform does not know the edge latency function parameters $(\theta_e)_{e\in E}$. Instead, the platform learns the network cost parameters by repeatedly routing travelers in each stage, and updating the parameter estimates based on the collected data of travelers' experience travel time c^t. In the this section, we define the notion of regret of each of the three types of routing strategies made by the platform compared to the case with complete information of the cost functions. We also present the online learning algorithm, and provide detailed discussions on the cost parameter estimates, and confidence intervals.

For any given $\xi \in [0,1]$, we define the total regret of all T stages as follows:

$$R(\xi) = \sum_{t=1}^{T} \left(\xi C(q^t) + (1-\xi)\Phi(q^t) - \min_{q\in Q^t} \left(\xi C(q) + (1-\xi)\Phi(q)\right) \right)$$

We note that for $\xi = 1$, the regret equals to the difference between the average social cost and the socially optimal average cost, and the regret is zero if and only if the routing strategy is socially optimal. On the other hand, for $\xi = 0$, the regret equals to the difference between the potential function value and the minimum value of potential function, which is an upper bound on individual traveler's regret of not choosing the shortest path. In this case, the regret is zero if the routing strategy is an equilibrium. Additionally, for $\xi \in (0,1)$, the regret is zero if and only if the routing strategy is the optimal combined strategy with weight ξ. Therefore, by minimizing the regret function $R(\xi)$ with a chosen weight ξ, the platform makes the trade-off between learning for minimizing the average social cost and learning for minimizing the individual's travel regret.[8]

8. Due to the fluctuating demand, our regret is defined for inidividual travelers.

3. Algorithms and Computation

Before presenting the learning algorithm, we first introduce the regularized ordinary least square (OLS) estimate of the cost parameters of each stage t, and construct a rectangular confidence interval of the cost estimates.

Lemma 2 *For any $t \in [T]$, edge $e \in \mathcal{E}$, the parameter estimate $\hat{\theta}_e^t = (\hat{\theta}_{r,0}^t, \hat{\theta}_{e,1}^t)$ is computed as the regularized least square estimate of θ_e given the regularizer λ_e:*

$$\hat{\theta}_e^t = \arg\min_{\theta_e} \left(\sum_{j \in \{[t-1] | w_e^j > 0\}} \sum_{k=1}^{w_e^j} (c_{e,k}^j - \theta_{e,0} - \theta_{e,1} w_e^j)^2 + \lambda_e \|\theta_e\|^2 \right) = (V_e^t)^{-1} U_e^t, \quad \forall e \in E,$$

where

$$V_e^t = \begin{bmatrix} \lambda_e & 0 \\ 0 & \lambda_e \end{bmatrix} + \sum_{j \in \{[t-1] | w_e^j > 0\}} \begin{bmatrix} w_e^j & (w_e^j)^2 \\ (w_e^j)^2 & (w_e^j)^3 \end{bmatrix}, \quad U_e^t = \sum_{j \in \{[t-1] | w_e^j > 0\}} \begin{bmatrix} \sum_{k=1}^{w_e^j} c_{e,k}^j \\ w_e^j \left(\sum_{k=1}^{w_e^j} c_{e,k}^j \right) \end{bmatrix}.$$

In Lemma 2, the OLS estimate of each edge e uses data collected by the platform in all stages j with positive number of travelers before t, i.e. $j \in \{[t-1] | w_e^j > 0\}$. Additionally, the data collected on each edge e in stage j includes the realized costs of all travelers who take that edge, i.e. $(c_{e,k}^j)_{k \in [w_e^j]}$.

We next construct the rectangular confidence intervals for the OLS estimate of edge. In each step $t \in [T]$, we denote the confidence interval as $\mathcal{RC}^t = \prod_{e \in \mathcal{E}} \mathcal{RC}_e^t$, where $\mathcal{RC}_e^t$ is given by:

$$\mathcal{RC}_e^t = \left\{ \theta_e = (\theta_{e,0}, \theta_{e,1}) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} \hat{\theta}_{e,0}^t - \gamma_{e,0}^t \le \theta_{e,0} \le \hat{\theta}_{e,0}^t + \gamma_{e,0}^t, \\ \hat{\theta}_{e,1}^t - \gamma_{e,1}^t \le \theta_{e,1} \le \hat{\theta}_{e,1}^t + \gamma_{e,1}^t \end{array} \right\}, \quad \forall e \in E. \tag{5}$$

In equation (5), $\gamma_{e,0}^t$ (resp. $\gamma_{e,1}^t$) represents half the width of the confidence interval of the estimate $\hat{\theta}_{e,0}^t$ (resp. $\hat{\theta}_{e,1}^t$), and is defined as follows:

$$\gamma_{e,0}^t = \sqrt{\frac{4\eta_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}}, \quad \gamma_{e,1}^t = \sqrt{\frac{4\nu_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}}, \tag{6}$$

where

$$\beta_e^t = \sqrt{\lambda_e} d_e + \sqrt{2\log(t) + 2\log\left(\frac{2\lambda_e + t(d_e)^2}{2\lambda_e} \right)}, \quad \nu_e^t = \lambda_e + \sum_{j \in \{[t-1] | w_e^j > 0\}} w_e^j, \tag{7a}$$

$$\kappa_e^t = 2 \sum_{j \in \{[t-1] | w_e^j > 0\}} (w_e^j)^2, \quad \eta_e^t = \lambda_e + \sum_{j \in \{[t-1] | w_e^j > 0\}} (w_e^j)^3. \tag{7b}$$

The following lemma shows that the unknown cost parameter vector $\boldsymbol{\theta}$ falls into the constructed rectangular confidence interval with high probability.

Lemma 3 *For any edge $e \in \mathcal{E}$, with probability at least $1 - 1/T$ the unknown edge latency parameter $\boldsymbol{\theta}_e \in \mathcal{RC}_e^t$ for all $t \in [T]$. Consequently, with probability higher than $\left(1 - \frac{|\mathcal{E}|}{T}\right)$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_e)_{e \in \mathcal{E}} \in \mathcal{RC}^t$ for all $t \in [T]$.*

We note that in (5), the rectangular confidence interval of each edge cost parameters is constructed as the product of the interval of the intercept and the interval of the slope, which are independently constructed. This is in contrast to the ellipse confidence set defined in linear bandits literature Abbasi-Yadkori et al. (2011); Lattimore and Szepesvári (2020), where the confidence intervals of the intercept and the slope are jointly determined by the bound of the $\|\cdot\|_{V_e^t}$-norm.

The proof of Lemma 3 addresses two aspects that are new in our setting: *(i)* the number of data points collected on each edge for each stage, and the lack of when not taken; *(ii)* rectangular instead of ellipse confidence intervals. In particular, our proof builds on the following lemma that bounds the rectangular confidence set between two ellipse confidence sets.

Lemma 4 *For any edge $e \in E$, the rectangle $\mathcal{RC}_e^t$ is a bounded between two ellipsoids as follows*

$$\{\theta \in \mathbb{R}^2 : \|\theta - \hat{\theta}_e^t\|_{V_e^t}^2 \le \beta_e^t\} =: \mathcal{C}_e^t \subset \mathcal{RC}_e^t \subset \tilde{\mathcal{C}}_e^t := \{\theta \in \mathbb{R}^2 : \|\theta - \hat{\theta}_e^t\|_{V_e^t}^2 \le \upsilon_e^t \beta_e^t\},$$

where $\upsilon_e^t = \frac{4\sqrt{\nu_e^t \eta_e^t}}{2\sqrt{\nu_e^t \eta_e^t} - \kappa_e^t} \le 6\sqrt{d_e}$ *for all $e \in E$ and all $t \in [T]$.*

We define the optimistic parameter estimate and the optimistic routing strategy $(\tilde{\theta}^t, \tilde{q}^t)$ as the solution of the following optimization problem that jointly minimizes the routing objective function over the parameter estimate in the confidence interval and the routing strategy:

$$\begin{aligned}(\tilde{\theta}^t, \tilde{q}^t) &= \mathop{\arg\min}_{\theta \in \mathcal{RC}^t, q^t \in Q^t} \Psi(q^t, \xi) \\ &= \mathop{\arg\min}_{\theta \in \mathcal{RC}^t, q^t \in Q^t} \xi \frac{1}{N^t} \left(\sum_{e \in \mathcal{E}} (\theta_{e,0} + \theta_{e,1} w_e^t) w_e^t \right) + (1 - \xi) \left(\sum_{e \in \mathcal{E}} \sum_{j=1}^{w_e^t} (\theta_{e,0} + \theta_{e,1} j) \right) \end{aligned} \tag{8}$$

We next show that given the rectangular confidence interval, the optimistic parameter estimate $\tilde{\theta}^t$ is the minimum parameter vector in the rectangle confidence interval, and the $\tilde{q}^t$ is the minimizer of the routing objective function associated with $\tilde{\theta}$. Therefore, the rectangular confidence set significantly simplifies the computation of the optimistic parameter estimates and routing strategies in (8) compared to that with an ellipse confidence set.

Lemma 5 *For any $t \in [T]$, and any $\xi \in [0, 1]$,*

$$\tilde{\theta}_{e,0}^t = \hat{\theta}_{e,0}^t - \gamma_{e,0}^t, \quad \tilde{\theta}_{e,1}^t = \hat{\theta}_{e,1}^t - \gamma_{e,1}^t, \quad \forall e \in E,$$

$$\tilde{q}^t = \mathop{\arg\min}_{q^t \in Q^t} \tilde{\Psi}^t(\xi, q^t) = \mathop{\arg\min}_{q^t \in Q^t} \xi \frac{1}{N^t} \left(\sum_{e \in \mathcal{E}} (\tilde{\theta}_{e,0}^t + \tilde{\theta}_{e,1}^t w_e^t) w_e^t \right) + (1 - \xi) \left(\sum_{e \in \mathcal{E}} \sum_{j=1}^{w_e^t} (\tilde{\theta}_{e,0}^t + \tilde{\theta}_{e,1}^t j) \right).$$

We are now ready to present our online learning algorithm. In each step $t \in [T]$, given the total number of routing requests N^t, Algorithm 1 first computes the optimistic routing strategy $\tilde{q}^t$ using the optimistic parameter estimate $\tilde{\theta}^t = (\tilde{\theta}_e)_{e \in E}$ (line 3).[9] *Then, the platform routes travelers,*

9. To improve the algorithm efficient, we solve $\tilde{q}^t$ as a (fractional) solution of continuous optimization problem, and randomly round $\tilde{q}$. The difference in social cost and individual cost (and consequently the regret) is small since the impact of individual traveler on the congestion cost is almost negligible. This is also consistent with the widely adopted modeling framework of nonatomic routing games.

and collect data of edge load and realized costs on edges (line 4-6). The algorithm updates the regularized OLS estimates $\hat{\theta}_e^{t+1}$ (line 7-8), and the optimistic parameter estimate $\tilde{\theta}_e^{t+1}$ as the lower bound of the rectangular confidence interval $\mathcal{RC}_e^{t+1}$ (line 9) for all edges that are taken in stage t. The parameter estimates of the remaining edges are kept unchanged (line 10).

Algorithm 1 Online learning algorithm for exploration in routing

1 **Input:** For each $e \in E$, $(\lambda_e = 2d_e^2), \tilde{\theta}_e^1, (N_i^t)_{i\in I, t\in[T]}, (d_e)_{e\in\mathcal{E}}, V_e^1 = \begin{bmatrix}\lambda_e & 0\\ 0 & \lambda_e\end{bmatrix}, Q_e^1 = \begin{bmatrix}0\\0\end{bmatrix}$.

2 **for** *t=1,2,..* **do**

3 Compute $\tilde{q}^t \leftarrow \arg\min_{q^t\in Q^t} \tilde{\Psi}^t(\xi, q^t)$ with $\xi = 1$ for socially optimal routing, $\xi = 0$ for equilibrium routing, and $\xi \in (0,1)$ for combined routing.

4 Compute $\tilde{w}_e^t \leftarrow \sum_{r\ni e} \tilde{q}_r^t$ for all $e \in E$

5 **for** $e \in \mathcal{E}$ **do**

`// Data collection of realized costs`

6 **if** $\tilde{w}_e^t > 0$ *receive edge costs* $(c_{e,k}^t)_{k\in[\tilde{w}_e^t]}$ **then**

`// Calculate least square estimate`

7 $V_e^{t+1} = V_e^t + \begin{bmatrix}\tilde{w}_e^t & (\tilde{w}_e^t)^2\\ (\tilde{w}_e^t)^2 & (\tilde{w}_e^t)^3\end{bmatrix}, \quad Q_e^{t+1} = Q_e^t + \begin{bmatrix}\sum_{k=1}^{\tilde{w}_e^t} c_{e,k}^t\\ \tilde{w}_e^t \left(\sum_{k=1}^{\tilde{w}_e^t} c_{e,k}^t\right)\end{bmatrix}$

8 $\hat{\theta}_e^{t+1} \leftarrow (V_e^{t+1})^{-1} Q_e^{t+1}$

`// Calculate optimistic estimate`

9 $\tilde{\theta}_e^{t+1} \leftarrow \hat{\theta}_e^{t+1} - \begin{bmatrix}\gamma_{e,0}^{t+1}\\ \gamma_{e,1}^{t+1}\end{bmatrix}$, where $\gamma_{e,0}^{t+1}$ and $\gamma_{e,1}^{t+1}$ are given by (6)

else

10 $V_e^{t+1} \leftarrow V_e^t$, $Q_e^{t+1} \leftarrow Q_e^t$, $\hat{\theta}_e^{t+1} \leftarrow \hat{\theta}_e^t$, $\tilde{\theta}_e^{t+1} \leftarrow \tilde{\theta}_e^t$

end

end

end

4. Regret Analysis

The following theorem presents the regret bound of Algorithm 1.

Theorem 6 *With probability at least $1 - |E|/T$, the accumulated regret of individual travelers with combined routing with weight parameter $\xi \in [0,1]$ is given by:*

$$R(\xi) \le 10|E|K\left(1 - \xi + \frac{\xi}{\underline{N}}\right)(\bar{d})^{3/4}\sqrt{T}\left(2\ln(T\bar{d}) + \sqrt{2\bar{d}\|\theta\|\ln(T\bar{d})}\right),$$

where $K = \max_e(\alpha_e + \beta_e d_e)\, d_e$, $\underline{N} = \min_t N_t$ and $\bar{d} = \max_{e\in E} d_e$. In particular, the regret of socially optimal routing is $R^* = R(1)$, and the regret of equilibrium routing is $R^\dagger = R(0)$.

Our regret bound in Theorem 6 is $O(\sqrt{T})$, and scales linearly on the number of edges $|E|$ in the network. Furthermore, with $\xi = 1$, the regret of socially optimal routing $R(1)$ decreases in the minimum number of travelers $\underline{N}$ of every stage. The intuition is that as more travelers are routed by

the platform, more data is collected to learn the cost parameters faster, and thus leads to a smaller averaged regret for every traveler under the socially optimal routing. On the other hand, as $\xi = 0$, the equilibrium routing regret bound is independent of the number of travelers. This is because although more travelers leads to more number of data points, the inefficiency due to selfish routing also becomes more significant as the number of travelers increase. As a result, the two effect cancels out in the analysis of the equilibrium routing regret bound. Finally, the regret bound of combined routing ξ scales linearly between $R(1)$ and $R(0)$, and increases in ξ.

The proof of Theorem 6 builds on Lemmas 2 – 5

Proof of Theorem 6. Let's consider the event $\mathcal{Z} = \{\boldsymbol{\theta}_{e,0} \geq \tilde{\theta}^t_{e,0}, \boldsymbol{\theta}_{e,1} \geq \tilde{\theta}^t_{e,1} \; \forall \, e \in E, t \in [T]\}$. From Lemma 3 we know that $\Pr(\mathcal{E}) \geq 1 - |E|/T$. In what follows we assume $\mathcal{E}$ holds. Given any $\xi \in (0,1)$, we compute the value of the objective function with the optimal combined routing strategy $q^{t,\xi}$ and the induced edge load vector $w^{t,\xi}$:

$$
\begin{aligned}
\Psi(\xi, q^{t,\xi}) &= \sum_{e\in E} \left(\frac{\xi}{N^t}(\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} w_e^{t,\xi}) w_e^{t,\xi} + (1-\xi) \left(\sum_{j=1}^{w_e^{t,\xi}} \boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} j \right) \right) \\
&= \sum_{e\in E} \left(\frac{\xi}{N^t}(\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} w_e^{t,\xi}) w_e^{t,\xi} + (1-\xi)(\boldsymbol{\theta}_{e,0} w_e^{t,\xi} + \boldsymbol{\theta}_{e,1} \frac{(1+w_e^{t,\xi}) w_e^{t,\xi}}{2}) \right) \\
&= \sum_{e\in E} \left(\left(1 - \xi + \frac{\xi}{N^t} \right) \boldsymbol{\theta}_{e,0} + \frac{1-\xi}{2} \boldsymbol{\theta}_{e,1} \right) w_e^{t,\xi} + \left(\frac{1-\xi}{2} + \frac{\xi}{N^t} \right) \boldsymbol{\theta}_{e,1} (w_e^{t,\xi})^2 \\
&= \sum_{e\in E} [\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1}] \cdot M^t \cdot \begin{bmatrix} w_e^{t,\xi} \\ (w_e^{t,\xi})^2 \end{bmatrix},
\end{aligned}
$$

where $M^t = \begin{bmatrix} 1 - \xi + \frac{\xi}{N^t} & 0 \\ \frac{1-\xi}{2} & \frac{1-\xi}{2} + \frac{\xi}{N^t} \end{bmatrix}$. Under the event $\mathcal{Z}$, it holds that $[\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1}] \in \mathcal{RC}^t_e$. Consequently, it follows that

$$
\begin{aligned}
\Psi(\xi, q^{t,\xi}) &= \sum_{e\in E} [\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1}] \cdot M^t \cdot \begin{bmatrix} w_e^{t,\xi} \\ (w_e^{t,\xi})^2 \end{bmatrix} \geq \min_{\theta \in \mathcal{RC}^t} \sum_{e\in E} [\boldsymbol{\theta}_{e,0}, \boldsymbol{\theta}_{e,1}] \cdot M^t \cdot \begin{bmatrix} w_e^{t,\xi} \\ (w_e^{t,\xi})^2 \end{bmatrix} \\
&= \sum_{e\in E} [\tilde{\theta}^t_{e,0}, \tilde{\theta}^t_{e,1}] \cdot M^t \cdot \begin{bmatrix} w_e^{t,\xi} \\ (w_e^{t,\xi})^2 \end{bmatrix} \geq \sum_{e\in E} [\tilde{\theta}^t_{e,0}, \tilde{\theta}^t_{e,1}] \cdot M^t \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} = \tilde{\psi}^t(\xi, \tilde{q}^t),
\end{aligned}
$$

where the last inequality is due to the fact that $\tilde{q}^t$ minimizes $\tilde{\psi}^t(\xi, q^t)$.

Then, the regret at stage t is given by

$$
r^t = \Psi(\xi, \tilde{q}^t) - \Psi(\xi, q^{t,\xi}) \leq \Psi(\xi, \tilde{q}^t) - \tilde{\psi}^t(\xi, \tilde{q}^t) = \sum_{e\in E} \left(\boldsymbol{\theta}_e - \tilde{\theta}^t_e \right)^\top \cdot M^t \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} \tag{9}
$$

Moreover, since the capacity on each edge is finite, and costs are positive, we have:

$$
\begin{aligned}
r^t = \Psi(\xi, \tilde{q}^t) - \tilde{\psi}^t(\xi, \tilde{q}^t) &\leq \Psi(\xi, \tilde{q}^t) \leq \sum_{e\in E} \left(1 - \xi + \frac{\xi}{N^t} \right) \left(\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} \tilde{w}_e^t \right) \tilde{w}_e^t \\
&\leq \sum_{e\in E} (\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} \tilde{w}_e^t) \tilde{w}_e^t \leq \sum_{e\in E} (\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} d_e) d_e
\end{aligned} \tag{10}
$$

where the second to last inequality follows by noting that $N^t \geq 1$ otherwise $\tilde{w}_e^t = 0$ which will not contribute to the sum and the final inequality follows by the capacity constraint $\tilde{w}_e^t \leq d_e$. We define the constant $K_e = (\boldsymbol{\theta}_{e,0} + \boldsymbol{\theta}_{e,1} d_e)\, d_e$, then combining (9) and (10) we obtain

$$r^t \leq \sum_e \left\{ K_e \wedge \left(\boldsymbol{\theta}_e - \tilde{\theta}_e^t\right)^\top M^t \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} \right\}$$

We apply Cauchy-Schwartz inequality to bound the total regret as follows:

$$\begin{aligned} R = \sum_{t=1}^T r^t &= \sum_{e \in E} \sum_{t=1}^T \left\{ K_e \wedge \left(\boldsymbol{\theta}_e - \tilde{\theta}_e^t\right)^\top M^t \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} \right\} \\ &\leq |E| \sqrt{T \sum_{t=1}^T \left\{ K \wedge \left(\boldsymbol{\theta}_{\tilde{e}} - \tilde{\theta}_{\tilde{e}}^t\right)^\top M^t \cdot \begin{bmatrix} \tilde{w}_{\tilde{e}}^t \\ (\tilde{w}_{\tilde{e}}^t)^2 \end{bmatrix} \right\}^2} \end{aligned} \tag{11}$$

where $\tilde{e} = \arg\max_e \sum_{t=1}^T \left\{ K_e \wedge \left(\boldsymbol{\theta}_e - \tilde{\theta}_e^t\right)^\top M^t \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} \right\}$, $K = \max_{e \in E} K_e$. Next, we note that for every $e \in E$ it holds that

$$\begin{aligned} &\left(\boldsymbol{\theta}_e - \tilde{\theta}_e^t\right)^\top M \cdot \begin{bmatrix} \tilde{w}_e^t \\ (\tilde{w}_e^t)^2 \end{bmatrix} = \tilde{w}_e^t \begin{bmatrix} \boldsymbol{\theta}_{e,0} - \tilde{\theta}_{e,0}^t & \boldsymbol{\theta}_{e,1} - \tilde{\theta}_{e,1}^t \end{bmatrix} \begin{bmatrix} 1 - \xi + \frac{\xi}{N^t} & 0 \\ \frac{1-\xi}{2} & \frac{1-\xi}{2} + \frac{\xi}{N^t} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{w}_e^t \end{bmatrix} \\ &\leq \tilde{w}_e^t \left(1 - \xi + \frac{\xi}{N^t}\right) \left((\boldsymbol{\theta}_{e,0} - \tilde{\theta}_{e,0}^t) + (\boldsymbol{\theta}_{e,1} - \tilde{\theta}_{e,1}^t) \tilde{w}_e^t \right) \end{aligned} \tag{12}$$

where the inequality is obtained by noting that under the event $\mathcal{E}$, we have $\boldsymbol{\theta}_{e,0} \geq \tilde{\theta}_{e,0}^t$ and $\boldsymbol{\theta}_{e,1} \geq \tilde{\theta}_{e,1}^t$. Define $\Xi^t = \left(1 - \xi + \frac{\xi}{N^t}\right)$ and $y_{\tilde{e}}^t = \begin{bmatrix} 1 \\ \tilde{w}_{\tilde{e}}^t \end{bmatrix}$. Then, it follows that

$$\begin{aligned} &\left| \left(\boldsymbol{\theta}_{\tilde{e}} - \tilde{\theta}_{\tilde{e}}^t\right)^\top M^t \cdot y_{\tilde{e}}^t \right|^2 \leq (\tilde{w}_{\tilde{e}}^t \Xi^t)^2 \left| \left(\boldsymbol{\theta}_{\tilde{e}} - \tilde{\theta}_{\tilde{e}}^t\right)^\top y_{\tilde{e}}^t \right|^2 \leq (\tilde{w}_{\tilde{e}}^t \Xi^t)^2 \|\boldsymbol{\theta}_{\tilde{e}} - \tilde{\theta}_{\tilde{e}}^t\|_{V_e^t}^2 \|y_{\tilde{e}}^t\|_{(V_e^t)^{-1}}^2 \\ =&(\tilde{w}_{\tilde{e}}^t \Xi^t)^2 \left(\|\boldsymbol{\theta}_{\tilde{e}} - \hat{\theta}_{\tilde{e}}^t + \hat{\theta}_{\tilde{e}}^t - \tilde{\theta}_{\tilde{e}}^t\|_{V_{\tilde{e}}^t}^2 \right) \|y_{\tilde{e}}^t\|_{(V_{\tilde{e}}^t)^{-1}}^2 \\ \leq&2(\tilde{w}_{\tilde{e}}^t \Xi^t)^2 \left(\|\boldsymbol{\theta}_{\tilde{e}} - \hat{\theta}_{\tilde{e}}^t\|_{V_{\tilde{e}}^t}^2 + \|\hat{\theta}_e^t - \tilde{\theta}_{\tilde{e}}^t\|_{V_{\tilde{e}}^t}^2 \right) \|y_{\tilde{e}}^t\|_{(V_{\tilde{e}}^t)^{-1}}^2 \leq 4(\tilde{w}_{\tilde{e}}^t \Xi^t)^2 \upsilon_{\tilde{e}}^t \beta_{\tilde{e}}^T \|y_{\tilde{e}}^t\|_{(V_{\tilde{e}}^t)^{-1}}^2, \end{aligned} \tag{13}$$

where the first inequality is due to (12), the second inequality is due to Cauchy Schwartz inequality and the third inequality holds due to the fact that $\tilde{\theta}_e^t \in \mathcal{RC}_e^t$ along with Lemma 3 and Lemma 4. Thus, combining (11) and (13) we obtain

$$R \leq |E| K \Xi^{\max} \sqrt{4T \upsilon_{\tilde{e}}^{\max} \beta_{\tilde{e}}^T \sum_{t=1}^T \left(1 \wedge (\tilde{w}_{\tilde{e}}^t)^2 \|y_{\tilde{e}}^t\|_{(V_{\tilde{e}}^t)^{-1}}^2\right)},$$

where $\Xi^{\max} = 1 - \xi + \frac{\xi}{\underline{N}}$, $\underline{N} = 1 \wedge \min_t N^t$ and $\gamma_e^{\max} = \max_t \gamma_e^t$. Next, we note that for any $e \in E$

$$\begin{aligned}
&\sum_{t=1}^{T} \left(1 \wedge (\tilde{w}_e^t)^2 \|y_e^t\|^2_{(V_e^t)^{-1}}\right) \leq d_e \sum_{t=1}^{T} \left(1 \wedge \tilde{w}_e^t \|y_e^t\|^2_{(V_e^t)^{-1}}\right) \\
\leq & 2d_e \sum_{t=1}^{T} \ln\left(1 + \tilde{w}_e^t \|y_e^t\|^2_{(V_e^t)^{-1}}\right) = 2d_e \ln\left(\frac{\det\left(V_e^T\right)}{\det\left(V_e^0\right)}\right) \leq 2\bar{d} \ln\left(\frac{\det\left(V_e^T\right)}{\det\left(V_e^0\right)}\right),
\end{aligned}$$

where the second inequality follows by noting that $1 \wedge u \leq 2\ln(1+u)$ for any $u \geq 0$ and the equality follows due to Lemma 7 in the appendix. Next, from (11) we note that

$$R \leq |E| K \Xi^{\max} \sqrt{8T\bar{d}\upsilon_e^{\max} \beta_{\tilde{e}}^T \ln\left(\frac{\det(V_{\tilde{e}}^T)}{\det(V_{\tilde{e}}^0)}\right)} \leq |E| K \Xi^{\max} \sqrt{16T\bar{d}\upsilon_e^{\max} \beta_{\tilde{e}}^T \ln\left(\frac{\lambda_e + Td_e^3}{\lambda_e}\right)}$$

where the last inequality follows by Lemma 8. Next, by choosing $\lambda_e = 2d_e^2$ we have

$$R \leq |E| K \Xi^{\max} \sqrt{96T\bar{d}\sqrt{\bar{d}} \beta_{\tilde{e}}^T \ln\left(\frac{\lambda_e + Td_e^3}{\lambda_e}\right)} \leq |E| K \Xi^{\max} \sqrt{96T\bar{d}^{3/2} \beta_{\tilde{e}}^T \ln(Td_{\tilde{e}})}$$

where the first inequality is due to Lemma 4. Finally, noting that

$$\sqrt{\beta_e^T} \leq \sqrt{2\ln(T) + 2\ln\left(\frac{\lambda_e + Td_e^3}{\lambda_e}\right)} + \sqrt{\lambda_e}\|\boldsymbol{\theta}_e\| \leq \sqrt{4\ln(Td_e)} + 2d_e\|\theta\|_e$$

Thus, $R \leq 10|E|K\Xi^{\max}(\bar{d})^{3/4}\sqrt{T}\left(2\ln(T\bar{d}) + \sqrt{2\bar{d}\|\theta\|\ln(T\bar{d})}\right)$. □

5. Experimental Analysis in New York City

In this section we conduct an experimental analysis of our algorithm in the road network of New York City. We partition the road network of New York City into 8 regions and in each round we draw a random arrival rate for routing requests between each pair of regions. We model the requests as originating and terminating in the middle points of the regions. The arrival rates are uniformly drawn for each origin-destination pair, between 5 and 15 cars per time unit. We extract the road network from OpenStreetMap (2017). The total number of edges is 344, 524 (Fig. 1).

The average travel time cost of each edge takes the functional form of the Bureau of Public Roads (1964) with linear exponent. Specifically we set $\ell_e(w_e) = \alpha t_e^f w_e/d_e + t_e^f$, where t_e^f is the time needed to cross the edge when the road is empty, i.e., the free-flow travel time, and d_e is the capacity of the street, defined as the number of lanes multiplied by the free-flow speed. The value of t_e^f and d_e is directly known from using the data set from Open Street Map. We calibrate the parameter γ using the data of edge load and travel time from the online navigation systems. Thus, for each $e \in E$, we obtain the true cost parameter $\boldsymbol{\theta}_{e,0} = t_e^f$, and $\boldsymbol{\theta}_{e,1} = \alpha t_e^f/d_e$. The true parameter is not known by the learning algorithm.

We implement Algorithm 1 for socially optimal routing ($\xi = 1$) and equilibrium routing ($\xi = 0$). We initialize $\theta_{e,0}^1 = \theta_{e,1}^1 = \tilde{\theta}_{e,0}^1 = \tilde{\theta}_{e,1}^1 = 0$ for all edges e. In each round $t = 1, 2, \ldots$, the

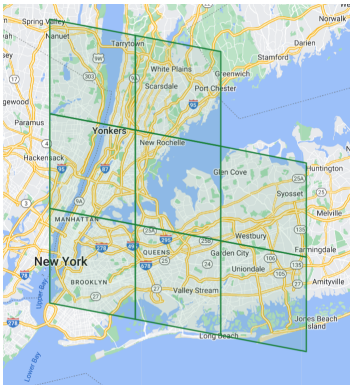

Figure 1: The regions of NYC that we use in our experiments.

algorithm observes the realized travel demand and their origin destination pairs, Based on the up-to-date optimistic parameter estimates $(\tilde{\theta}_e^t)_{e\in E}$, the algorithm computes either (a) socially optimal routing strategy or (b) the equilibrium routing strategy.[10] Then, the algorithm collects the realized cost of travelers on each taken edge, which equals to the value of the latency function $\ell_e(w_e^t)$ plus a random noise that is uniformly distributed in $[-0.05\ell_e(d_e)\ ,0.05\ell_e(d_e)]$. Given the collected cost data, the algorithm updates $\hat{\theta}_e^{t+1}$ and $\tilde{\theta}_e^{t+1}$ for edges that are taken.

By computing the equilibrium routing strategy and socially optimal routing strategy given the true cost parameters, we can compute the regret of each stage. In Fig. 2, we demonstrate that the regret in each stage decreases to zero for both the socially optimal routing and the equilibrium routing. In Fig. 3, we demonstrate that the accumulated regret increases sub-linearly with respect to t, and the regret of equilibrium routing is higher than that of socially optimal routing.

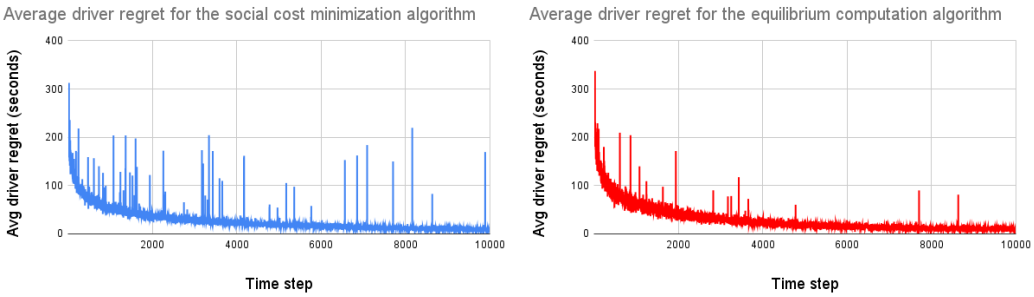

Figure 2: Stage regret of Algorithm 1 in NYC.

6. Concluding Remarks

In this article, we propose an online learning algorithm for a navigation platform to learn the unknown cost function of congested road segments and route travelers. Depending on the platform's

10. We compute a fractional routing strategy by solving the convex optimization program corresponding to the optimistic parameter estimates. Then, we do randomized routing if needed.

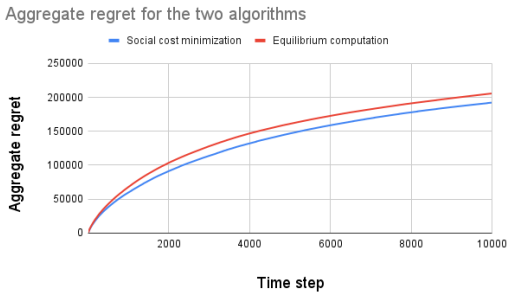

Figure 3: Accumulated regret of Algorithm 1 in NYC.

trade-off between efficiency and fairness, the routing strategy in the algorithm can be a socially optimal strategy, or an equilibrium strategy, or a combination of the two. Our regret analysis demonstrates that the accumulated regret upper bound of the proposed algorithm is sublinear in the number of time steps, and linear in the number of edges. We also show that the regret bound is linear in the inverse of the number of travelers, which implies that the platform with larger user size learns faster.

Acknowledgments

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Marc Abeille and Alessandro Lazaric. Linear thompson sampling revisited. In *Artificial Intelligence and Statistics*, pages 176–184. PMLR, 2017.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR, 2013.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Pranjal Awasthi, Kush Bhatia, Sreenivas Gollapudi, and Kostas Kollias. Congested bandits: Optimal routing via short-term resets. In *International Conference on Machine Learning*, pages 1078–1100. PMLR, 2022.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. The price of fairness. *Operations research*, 59(1):17–31, 2011.

Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: On convergence to nash equilibria of regret-minimizing algorithms in routing games. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 45–52, 2006.

Bureau of Public Roads. *Traffic assignment manual*. US Department of Commerce, 1964.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. *Advances in neural information processing systems*, 28, 2015.

Roberto Cominetti, Emerson Melo, and Sylvain Sorin. A payoff-based learning procedure and its application to traffic games. *Games and Economic Behavior*, 70(1):71–83, 2010.

José R Correa, Andreas S Schulz, and Nicolás E Stier-Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

Stella C Dafermos and Frederick T Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards B*, 73(2):91–118, 1969.

Varsha Dani and Thomas P Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *SODA*, volume 6, pages 937–943, 2006.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3(2):97–139, 1957.

Olaf Jahn, Rolf H Möhring, Andreas S Schulz, and Nicolás E Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4): 600–616, 2005.

Devansh Jalota, Kiril Solovey, Matthew Tsao, Stephen Zoepf, and Marco Pavone. Balancing fairness and efficiency in traffic routing via interpolated traffic assignment. *arXiv preprint arXiv:2104.00098*, 2021.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 568–578. IEEE, 1999.

Robert Kleinberg, Georgios Piliouras, and Éva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 533–542, 2009.

Walid Krichene, Benjamin Drighes, and Alexandre Bayen. On the convergence of no-regret learning in selfish routing. In *International Conference on Machine Learning*, pages 163–171. PMLR, 2014.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543. PMLR, 2015.

Tze Leung Lai, Herbert Robbins, et al. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. *Advances in neural information processing systems*, 30, 2017.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

H Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *International Conference on Computational Learning Theory*, pages 109–123. Springer, 2004.

Emily Meigs, Francesca Parise, and Asuman Ozdaglar. Learning dynamics in stochastic routing games. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 259–266. IEEE, 2017.

Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1): 124–143, 1996.

OpenStreetMap. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Tim Roughgarden. How unfair is optimal routing? In *Symposium on Discrete Algorithms: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, volume 6, pages 203–204, 2002.

Tim Roughgarden. *Selfish routing and the price of anarchy*. MIT press, 2005.

Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Andreas S Schulz and Nicolás E Stier-Moses. Efficiency and fairness of system-optimal routing with user constraints. *Networks: An International Journal*, 48(4):223–234, 2006.

Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 1113–1122. PMLR, 2015.

Manxi Wu and Saurabh Amin. Learning an unknown network state in routing games. *IFAC-PapersOnLine*, 52(20):345–350, 2019.

Appendix A. Proof of Lemmas

Proof of Lemma 2. We note that the optimization problem for computing the regularized least square estimate is strongly convex in θ_e. Therefore, the regularized OLS estimate $\hat{\theta}_e^t$ satisfies the following first order condition:

$$\sum_{j\in\{[t-1]|w_e^j>0\}} \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ w_e^j & w_e^j & \cdots & w_e^j \end{bmatrix}}_{w_e^j \text{ columns}} \cdot \begin{bmatrix} 1 & w_e^j \\ 1 & w_e^j \\ \vdots & \vdots \\ 1 & w_e^j \end{bmatrix} \cdot \begin{bmatrix} \hat{\theta}_{e,0}^t \\ \hat{\theta}_{e,1}^t \end{bmatrix} = \sum_{j\in\{[t-1]|w_e^j>0\}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ w_e^j & w_e^j & \cdots & w_e^j \end{bmatrix} \cdot \begin{bmatrix} c_{e,1}^j \\ c_{e,2}^j \\ \vdots \\ c_{e,w_e^j}^j \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \sum_{j\in\{[t-1]|w_e^j>0\}} w_e^j & \sum_{j\in\{[t-1]|w_e^j>0\}} (w_e^j)^2 \\ \sum_{j\in\{[t-1]|w_e^j>0\}} (w_e^j)^2 & \sum_{j\in\{[t-1]|w_e^j>0\}} (w_e^j)^3 \end{bmatrix} \begin{bmatrix} \hat{\theta}_{e,0}^t \\ \hat{\theta}_{e,1}^t \end{bmatrix} = \begin{bmatrix} \sum_{j\in\{[t-1]|w_e^j>0\}} \sum_{k=1}^{w_e^j} c_{e,k}^j \\ \sum_{j\in\{[t-1]|w_e^j>0\}} w_e^j (\sum_{k=1}^{w_e^j} c_{e,k}^j) \end{bmatrix}$$

$$\Rightarrow V_e^t \cdot \begin{bmatrix} \hat{\theta}_{e,0}^t \\ \hat{\theta}_{e,1}^t \end{bmatrix} = Q_e^t$$

$$\Rightarrow \begin{bmatrix} \hat{\theta}_{e,0}^t \\ \hat{\theta}_{e,1}^t \end{bmatrix} = (V_e^t)^{-1} Q_e^t$$

That is, the parameter estimate $\hat{\theta}^t$ computed in Algorithm 1 is the regularized OLS estimate. □

Since Lemma 3 builds on Lemma 4, we first proof Lemma 4.

Proof of Lemma 4. We first define the following half spaces:

$$L_1 = \left\{ (\theta_{e,0}, \theta_{e,1}) \in \mathbb{R}^2 \,\middle|\, \theta_{e,1} \le \hat{\theta}_{e,1}^t + \sqrt{\frac{4\nu_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}} \right\},$$

$$L_2 = \left\{ (\theta_{e,0}, \theta_{e,1}) \in \mathbb{R}^2 \,\middle|\, \theta_{e,1} \ge \hat{\theta}_{e,1}^t - \sqrt{\frac{4\nu_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}} \right\},$$

$$L_3 = \left\{ (\theta_{e,0}, \theta_{e,1}) \in \mathbb{R}^2 \,\middle|\, \theta_{e,0} \le \hat{\theta}_{e,0}^t + \sqrt{\frac{4\eta_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}} \right\},$$

$$L_4 = \left\{ (\theta_{e,0}, \theta_{e,1}) \in \mathbb{R}^2 \,\middle|\, \theta_{e,0} \ge \hat{\theta}_{e,0}^t - \sqrt{\frac{4\eta_e^t \beta_e^t}{4\nu_e^t \eta_e^t - (\kappa_e^t)^2}} \right\}$$

where $\nu_e^t, \kappa_e^t, \eta_e^t$ and β_e^t are given by (7). Thus, $\mathcal{RC}_e^t = L_1 \cap L_2 \cap L_3 \cap L_4$. We next re-write the characterization of set $\mathcal{C}_t$ defined in the lemma.

$$\mathcal{C}_e^t = \{(\theta_{e,0}, \theta_{e,1}) \,\big|\, \nu_e^t(\hat{\theta}_{e,0}^t - \theta_{e,0})^2 + \kappa_e^t(\hat{\theta}_{e,0}^t - \theta_{e,0})(\hat{\theta}_{e,1}^t - \theta_{e,1}) + \eta_e^t(\hat{\theta}_{e,1}^t - \theta_{e,1})^2 \le \beta_e^t\} \quad (14)$$

To show $\mathcal{C}_e^t \subset \mathcal{RC}_e^t$, it is sufficient to show that $\mathcal{C}_e^t \subset L_i$ for all $i \in \{1,2,3,4\}$. For the sake of concise presentation, we only check the condition $\mathcal{C}_e^t \subset L_1$. To see this, we show that for any $\theta_e \notin L_1$ it holds that $\theta_e \notin \mathcal{C}_e^t$. Note that for any $\theta_e \notin L_1$ it holds that $\theta_{e,1} > \hat{\theta}_{e,1}^t + \sqrt{\frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}}$. Consequently, it holds that

$$(\theta_{e,1} - \hat{\theta}_{e,1})^2 > \frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}. \tag{15}$$

Define a quadratic function $f(z) = \nu_e^t z^2 + \kappa_e^t(\hat{\theta}_{e,1} - \theta_{e,1})z + \eta_e^t(\hat{\theta}_{e,1} - \theta_{e,1})^2 - \beta_e^t$. To show that $\theta_e \notin \mathcal{C}_e^t$, it is enough to show that $f(z) > 0$ for all z. To establish this we first note that $f(0) > 0$. Second, we see that the discriminant of $f(\cdot)$, denoted as Δ, is negative. That is,

$$\begin{aligned}\Delta &= (\kappa_e^t)^2(\hat{\theta}_{e,1} - \theta_{e,1})^2 - 4\nu_e^t\left(\eta_e^t(\hat{\theta}_{e,1} - \theta_{e,1})^2 - \beta_e^t\right)\\ &= 4\eta_e^t\beta_e^t - (4\eta_e^t\nu_e^t - (\kappa_e^t)^2)(\hat{\theta}_{e,1} - \theta_{e,1})^2 < 0,\end{aligned}$$

where the last inequality holds due to (15). Therefore, $f(z) > 0$ for all z which implies that $\theta_e \notin \mathcal{C}_e^t$.

Next we show that $\mathcal{RC}_e^t \subset \tilde{\mathcal{C}}_e^t$. Since the set $\tilde{\mathcal{C}}_e^t$ is an ellipsoid and therefore a convex set, it suffices to check that the corners of the rectangle $\mathcal{RC}_e^t$ lie inside $\tilde{\mathcal{C}}_e^t$. The corners of the rectange $\mathcal{RC}_e^t$ are

$$\begin{aligned}P_1 &= \left(\hat{\theta}_{e,0}^t + \sqrt{\frac{4\eta_e^t\beta_e^t}{(4\nu_e^t\eta_e^t - (\kappa_e^t)^2)}},\ \hat{\theta}_{e,1}^t + \sqrt{\frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}}\right),\\ P_2 &= \left(\hat{\theta}_{e,0}^t + \sqrt{\frac{4\eta_e^t\beta_e^t}{(4\nu_e^t\eta_e^t - (\kappa_e^t)^2)}},\ \hat{\theta}_{e,1}^t - \sqrt{\frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}}\right),\\ P_3 &= \left(\hat{\theta}_{e,0}^t - \sqrt{\frac{4\eta_e^t\beta_e^t}{(4\nu_e^t\eta_e^t - (\kappa_e^t)^2)}},\ \hat{\theta}_{e,1}^t + \sqrt{\frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}}\right),\\ P_4 &= \left(\hat{\theta}_{e,0}^t - \sqrt{\frac{4\eta_e^t\beta_e^t}{(4\nu_e^t\eta_e^t - (\kappa_e^t)^2)}},\ \hat{\theta}_{e,1}^t - \sqrt{\frac{4\nu_e^t\beta_e^t}{4\nu_e^t\eta_e^t - (\kappa_e^t)^2}}\right).\end{aligned}$$

One can check that P_1, P_4 lie on the boundary of $\tilde{\mathcal{C}}_e^t$ while P_2, P_3 lie strictly inside $\tilde{\mathcal{C}}_e^t$.

Given any $e \in E$ and $t \in [T]$, since $0 \le w_e^t \le d_e$, we have

$$\lambda_e t \le \nu_e^t \le \lambda_e t + d_e t, \quad 0 \le \kappa_e^t \le 2(d_e)^2 t, \quad \lambda_e t \le \eta_e^t \le \lambda_e t + t(d_e)^3$$

Using the preceding inequalities, we have

$$\lambda_e t \le \sqrt{\nu_e^t\eta_e^t} \le t\sqrt{(\lambda_e + d_e)(\lambda + (d_e)^3)}, \quad -2t(d_e)^2 \le -(\kappa_e^t)$$

Thus, we have

$$\upsilon_e^t = \frac{4\sqrt{\nu_e^t\eta_e^t}}{2\sqrt{\nu_e^t\eta_e^t} - \kappa_e^t} \le \frac{2\sqrt{(\lambda_e + d_e)(\lambda_e + (d_e)^3)}}{\lambda_e - (d_e)^2}$$

Since $d_e \geq 1$, by choosing $\lambda_e = 2d_e^2$, we have

$$v_e^t \leq 2\frac{\sqrt{(3(d_e)^2)(3(d_e)^3)}}{(d_e)^2} \leq 6\sqrt{d_e}.$$

□

We are now ready to prove Lemma 3.

Proof of Lemma 3. The proof of this result is based on establishing the following two claims:

(a) For every $e \in \mathcal{E}, t \in [T]$ the following inclusion holds

$$\{\theta \in \mathbb{R}^2 : \|\theta - \hat{\theta}_e^t\|_{V_e^t}^2 \leq \beta_e^t\} =: \mathcal{C}_e^t \subset \mathcal{RC}_e^t$$

(b) For every edge $e \in \mathcal{E}$, with probability $1 - 1/T$ the unknown edge latency parameter $\boldsymbol{\theta}_e \in \mathcal{C}_e^t$ for all $t \in [T]$.

Note that (a) is satisfied due to Lemma 4. To establish (b), we note that at every time $t \in [T]$, edge $e \in E$, traveler $k \in [w_e^t]$ in the network communicates actual delay experienced to the platform which is $c_{e,k}^t = c_e(w_e^t) + \epsilon_{e,k}^t$ where $\epsilon_{e,k}^t$ is a $1-$sub-Gaussian random variable. We denote $S_e^t = \sum_{j=1}^t y_e^j \sum_{k=1}^{w_e^j} \epsilon_{e,k}^j$, where $y_e^j = \begin{bmatrix} 1 \\ w_e^j \end{bmatrix}$. We also denote that $W_e^t = V_e^t - \begin{bmatrix} \lambda_e & 0 \\ 0 & \lambda_e \end{bmatrix} = \sum_{j \in \{[t-1]|w_e^j > 0\}} \begin{bmatrix} w_e^j & (w_e^j)^2 \\ (w_e^j)^2 & (w_e^j)^3 \end{bmatrix}$. Next, we note that

$$\begin{aligned}
\|\hat{\theta}_e^t - \theta_e\|_{V_e^t} &= \|(V_e^t)^{-1} S_e^t + ((V_e^t)^{-1} W_e^t - I)\theta_e\|_{V_e^t} \\
&\leq \|(V_e^t)^{-1} S_e^t\|_{V_e^t} + \|((V_e^t)^{-1} W_e^t - I)\theta_e\|_{V_e^t} \\
&= \|(V_e^t)^{-1} S_e^t\|_{V_e^t} + \sqrt{\theta_e^\top ((V_e^t)^{-1} W_e^t - I)^\top V_e^t ((V_e^t)^{-1} W_e^t - I)\theta_e} \\
&= \|(V_e^t)^{-1} S_e^t\|_{V_e^t} + \sqrt{\lambda_e \|\theta_e\|^2 - \lambda_e \theta_e^\top W_e^{t\top} (V_e^t)^{-1} \theta_e} \\
&\leq \|S_e^t\|_{(V_e^t)^{-1}} + \sqrt{\lambda_e}\|\theta_e\| \qquad (16)
\end{aligned}$$

where the first equality is due to Lemma 2, the second inequality is by noting that $\theta_e^\top W_e^{t\top} (V_e^t)^{-1} \theta_e \geq 0$. Next we claim that with probability atleast $1 - 1/T$ the following holds

$$\|S_e^t\|_{(V_e^t)^{-1}} \leq \sqrt{2\log(T) + \log\left(\frac{\det(V_t)}{\lambda_e^2}\right)} + \sqrt{\lambda_e}\|\theta_e\| \qquad (17)$$

To see this we define $M_e^t(z) = \exp(\langle z, S_e^t \rangle - \frac{1}{2}\|z\|_{V_e^t}^2)$. Note that for $z \sim \mathcal{N}(0, \lambda_e I)$, $\bar{M}_e^t = \mathbb{E}_z[M_e^t(z)]$ is a supermartingale adapted to filtration $\mathcal{F}_e^t = \sigma(w_e^1, \tilde{c}_e^1, ..., \tilde{c}_e^{t-1}, w_e^t)$. Indeed, note that for any fixed z

$$\begin{aligned}
\mathbb{E}[M_e^t(z)|\mathcal{F}_e^t] &= M_e^{t-1}(z)\mathbb{E}\left[\exp\left(\left\langle z, y_e^t \sum_{k=1}^{w_e^t} \epsilon_{e,k}^t \right\rangle - \frac{1}{2}\|z\|_{w_e^t y_e^t y_e^{t\top}}^2\right) \middle| \mathcal{F}_t\right] \\
&= M_e^{t-1}(z) \prod_{k=1}^{w_e^t} \mathbb{E}[\exp(\epsilon_{e,k}^t \langle z, y_e^t \rangle - \frac{1}{2}\|z\|_{y_e^t y_e^{t\top}}^2)] \\
&\leq M_e^{t-1}(z)
\end{aligned}$$

where second equality is due to independence of noise. Meanwhile, the last inequality is due to the 1-sub Gaussian nature of noise. Thus it follows that $\bar{M}_e^t$ is a super martingale as well where $z \sim \mathcal{N}(0, \lambda_e I)$. Since $\bar{M}_e^t$ is a supermartingale, it follows from (Lattimore and Szepesvári, 2020, Theorem 20.4) that (17) holds with probability atleast $1 - 1/T$. We conclude the lemma by combining (16) and (17). □

Proof of Lemma 5. We first check that given any routing strategy q^t, $\tilde{\theta}$ minimizes the value of the objective function because $w_e^t \geq 0$, i.e.

$$\begin{aligned}
&\xi \frac{1}{N^t}\left(\sum_{e\in\mathcal{E}}(\tilde{\theta}_{e,0}^t + \tilde{\theta}_{e,1}^t w_e^t)w_e^t\right) + (1-\xi)\left(\sum_{e\in\mathcal{E}}\sum_{j=1}^{w_e^t}(\tilde{\theta}_{e,0}^t + \tilde{\theta}_{e,1}^t j)\right)\\
\leq &\xi \frac{1}{N^t}\left(\sum_{e\in\mathcal{E}}(\theta_{e,0}^t + \theta_{e,1}^t w_e^t)w_e^t\right) + (1-\xi)\left(\sum_{e\in\mathcal{E}}\sum_{j=1}^{w_e^t}(\theta_{e,0}^t + \theta_{e,1}^t j)\right), \quad \forall \theta \in \mathcal{RC}^t.
\end{aligned}$$

Additionally, since $\tilde{q}^t$ is the minimizer of the function $\tilde{\Psi}^t(\xi, q^t)$ with respect to $\tilde{\theta}^t$, we conclude that $(\tilde{\theta}^t, \tilde{q}^t)$ is the optimal solution of (8). □

Lemma 7 *For any edge $e \in E$, it holds that*

$$\sum_{t=1}^{T} \ln\left(1 + \tilde{w}_e^t \|y_e^t\|^2_{(V_e^t)^{-1}}\right) = \ln\left(\frac{\det V_e^T}{\det V_e^0}\right),$$

where $y_e^t = \begin{bmatrix} 1 \\ \tilde{w}_e^t \end{bmatrix}$.

Proof of Lemma 7. We note that

$$\begin{aligned}
V_e^t &= V_e^{t-1} + \tilde{w}_e^t y_e^t y_e^{t\top}\\
&= \left(V_e^{t-1}\right)^{1/2}\left(I + \tilde{w}_e^t \left(V_e^{t-1}\right)^{-1/2} y_e^t y_e^{t\top} \left(V_e^{t-1}\right)^{-1/2}\right)\left(V_e^{t-1}\right)^{1/2}.
\end{aligned}$$

Taking determinant of matrices on both sides of previous equation we obtain

$$\begin{aligned}
\det\left(V_e^t\right) &= \det\left(V_e^{t-1}\right)\det\left(I + \tilde{w}_e^t \left(V_e^{t-1}\right)^{-1/2} y_e^t y_e^{t\top} \left(V_e^{t-1}\right)^{-1/2}\right)\\
&= \det\left(V_e^{t-1}\right)\left(1 + \tilde{w}_e^t \|y_e^t\|^2_{(V_e^{t-1})^{-1}}\right).
\end{aligned}$$

Finally, $\det\left(V_e^T\right) = \det\left(V_e^0\right)\prod_{t=1}^T \left(1 + \tilde{w}_e^t \|y_e^t\|^2_{(V_e^t)^{-1}}\right)$. □

Lemma 8 *For any edge $e \in E$, $\det(V_e^T) \leq (\lambda_e + T d_e^3)^2$ and $\det(V_e^0) = \lambda_e^2$.*

Proof of Lemma 8. Note that $V_e^0 = \lambda_e I$ where I is the identity matrix in $\mathbb{R}^2$. Thus, $\det(V_e^0) = \lambda_e^2$. Next,

$$\begin{aligned}
\det(V_e^T) &\leq \left(\frac{1}{2}\text{trace}(V_e^T)\right)^2 = \left(\frac{1}{2}(2\lambda_e + \sum_{t=1}^{T} \tilde{w}_e^t y_e^{t\top} y_e^t)\right)^2\\
&\leq \frac{1}{4}\left(2\lambda_e + T d_e(1 + d_e^2)\right)^2 \leq (\lambda_e + T d_e^3)^2,
\end{aligned}$$

where the first inequality follows by noting that determinant of a matrix is product of eigenvalues and then bounding the product using the inequality of arithmetic and geometric means. The first equality follows by noting that $V_e^T = \lambda_e I + \sum_{t=1}^{T} \tilde{w}_e^t y_e^t y_e^{t\top}$. The second inequality follows by noting that $\tilde{w}_e^t \leq d_e$. □