Optimal network pricing with oblivious users: a new model and algorithm

Yixuan Li, Andersen Ang, Sebastian Stein
School of Electronics and Computer Science
University of Southampton
United Kingdom

October 10, 2025

Abstract

Traffic modeling is important in modern society. In this work we propose a new model on the optimal network pricing (Onp) with the assumption of oblivious users, in which the users remain oblivious to real-time traffic conditions and others' behavior. Inspired by works on transportation research and network pricing for selfish traffic, we mathematically derive and prove a new formulation of Onp with decision-dependent modeling that relax certain existing modeling constraints in the literature. Then, we express the Onp formulation as a constrained nonconvex stochastic quadratic program with uncertainty, and we propose an efficient algorithm to solve the problem, utilizing graph theory, sparse linear algebra and stochastic approximation. Lastly, we showcase the effectiveness of the proposed algorithm and the usefulness of the new Onp formulation. The proposed algorithm achieves a 5x speedup by exploiting the sparsity structure of the model.

Keywords: Traffic modeling, Optimal network pricing, Decision-dependent modeling, Stochastic optimization, Optimization algorithm, Sparsity, Graph theory

Contents

1 Introduction

Traffic modeling is important in modern society [22]. Motivated by the need to manage congestion and design efficient pricing policies, we propose a new model for optimal network pricing (Onp). As real-world users often lack timely access to traffic information or follow habitual, pre-planned decision rules, we adopt the *oblivious-user* assumption [8], where users plan their routes in advance, relying on fixed criteria like shortest distance and pricing signals, and remain oblivious to real-time traffic conditions and others' behavior. We mathematically derive a new formulation of Onp with decision-dependent modeling and prove its properties. After establishing the new formulation, we express it as a quadratic program (QP) with uncertainty, and propose an efficient algorithm to solve it. We showcase the effectiveness of the proposed algorithm and the usefulness of the new Onp model.

Behavioral modeling of network users. In classical models of network user behavior, travelers are assumed to be fully aware of others' actions and to interact strategically, yielding congestion games under a fixed total flow [24, 18]. Dynamic Traffic Assignment extends this picture by modeling the time-evolution of flows and travelers' adjustments of routes and departure times under dynamic equilibrium conditions [3]. However, because many users lack timely traffic information, several works instead consider oblivious users who do not react to instantaneous network states; e.g., the price of anarchy at the equilibrium in networks with oblivious users was studied in [8]. Overall, the literature emphasizes equilibrium characterization rather than the design of platform pricing or tolling mechanisms to influence user decisions.

Network Pricing. Prior research has studied pricing algorithm design to steer outcomes toward the social optimum. From an algorithmic game-theory viewpoint, works model user interactions as games, e.g. Stackelberg models [2, 16]. In [7], a congestion-pricing framework for networks is developed with variable traffic flow in which users' preferences are captured by a disutility function. It studied system-optimal outcomes from a social-welfare (non-profit planner) perspective and derives pricing characterizations for networks composed of parallel links, with the disutility representation to analyze tolling that minimizes aggregate social cost. Other research [6, 19] studied network pricing in specific settings.

Decision-dependent optimization. In performative prediction, the optimality conditions such as Performative Optimality (PO) [15, Eq.2.1] and Performative Stability (PS) [15, Def.2.3] are proposed:

$$\underbrace{\theta_{PO} := \operatorname*{argmin}_{\theta} \mathop{\mathbb{E}}_{Z\sim\mathcal{D}(\theta)} \ell(Z;\theta),}_{\text{PO}} \qquad \underbrace{\theta_{PS} := \operatorname*{argmin}_{\theta} \mathop{\mathbb{E}}_{Z\sim\mathcal{D}(\theta_{PS})} \ell(Z;\theta)\,.}_{\text{PS}} \tag{1}$$

The work [5] investigated the performance of stochastic algorithms in solving a problem (subsuming PS, PO) with decision-dependent distributions. The work [11] studied the greedy and lazy deployment scheme of stochastic optimization to solve PS with the consideration of a small *performative risk* (PO). The work [10] relaxed the assumption on the convexity of function ℓ used in [5] by defining an optimality condition δ-*stationary performative stable* solution and analyze the greedy deployment scheme. Based on the performative prediction framework, [13] introduced decision-dependent games where players account for the performative effect of users when making predictions. They applied this framework to competing ride-share markets and analyze pricing strategies when each platform aims to maximize its own revenue.

Contribution & paper organization. We propose a new model:

$$\begin{array}{ll}
\operatorname*{argmin}_{p} \mathop{\mathbb{E}}_{x\sim\mathcal{D}(p)} \left[\dfrac{\lambda}{2}\|p\|_2^2 + \dfrac{1}{2}\langle Qx, x\rangle - \langle s, x\rangle\right] & \text{quadratic cost} \\
\text{s.t.} \begin{cases} x \in [0, x_u^{\mathsf{pltf}}], p \in [p_l, p_u] \\ Kx \geq l \end{cases} & \begin{array}{l}\text{flow bound, price bound} \\ \text{commodity}\end{array} \\
\mathcal{D}(p) = \mathbf{\Pi}_{[x_l^{\mathsf{usr}}, x_u^{\mathsf{usr}}]}(Bp + \zeta) & \text{decision-dependent elasticity} \\
\zeta \sim \mathcal{N}(\mu, \Sigma) & \text{uncertainty}
\end{array} \tag{Onp}$$

Here is the overview of our model. (Onp) aims to decide the optimal price of each route the users have to pay, as well as to reduce the total traffic congestion. (Onp) considers a 2-player game between *platform* and *users*, where *platform* refers to a pricing system that determines the price p of each route of the traffic network (a graph), and *users* refers to the entities that use the network. The network congestion cost is described by a quadratic function on traffic flow variable x and p under parameters λ, Q, s. The price and the flow are related by a truncated model $\mathcal{D}(p) = \mathbf{\Pi}_{[x_l^{\text{usr}}, x_u^{\text{usr}}]}(Bp + \zeta) = x$, where $B \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$ is price elasticity in the network with $|\mathcal{R}|$ routes. The model $\mathcal{D}$ contains uncertainty, represented by the random variable $\zeta \in \mathbb{R}^{|\mathcal{R}|}$, which follows a base distribution, assumed to be Gaussian for simplicity. The symbol $\mathbf{\Pi}_{[x_l^{\text{usr}}, x_u^{\text{usr}}]}$ means project a vector onto the box $[x_l^{\text{usr}}, x_u^{\text{usr}}]$. Lastly $p_l, p_u, x_u^{\text{pltf}}, x_l^{\text{usr}}, x_u^{\text{usr}}, K, l$ are constants. **Notation.** $\langle a, b\rangle$ is inner product, $\|\cdot\|_2$ is ℓ_2-norm, $\mathbb{E}$ is expectation. $\max, \geq$ are applied element-wise, $[0, x_u]$ is box constraint. $[x]_j$ is the jth entry of x. See Table 1 for other symbols.

Remark. *Our proposed methodology naturally extends to models with more general $\mathcal{D}(p)$ (e.g., alternative distributions for ζ) and more complex projection operators $\mathbf{\Pi}$; such extensions require only problem-specific adaptations to the implementation and analysis.*

Assumption 1. We assume the solution set of (Onp) is nonempty and $x_l^{\text{usr}} = 0$, $x_u^{\text{usr}} = x_u^{\text{pltf}} = x_u$, i.e., users' lower and upper bounds coincide with the platform's capacity bounds [1].
The contribution of this work is 3-fold:

1. (Modeling & Theory)

- The new elasticity model $\mathcal{D}(p)$ is inspired by $Bp+\zeta = x$ in [13, Sec7.2], where we introduce $\mathbf{\Pi}_{[0,x_u]}$ to reflect practical considerations in traffic modeling.
- Previous works [24, 18] use a quadratic congestion cost without analyzing the structure of the quadratic coefficient matrix Q, we derive Q and (Onp), see section 2.
- Previous works [7, 8] assume *parallel link model*, meaning an edge in the network cannot be shared across different routes, which mathematically simplifies the matrix Q. In contrast, we relax the parallel link assumption by considering general Q, making our model more general and mathematically more challenging, see [8] and section 2 and section 5 for details.

2. (Methodology) We design a highly efficient algorithm to solve nonconvex (Onp) efficiently using sparsity, statistics and optimization theory, see section 5. We achieve a speedup ranged from 5x to 100x, depending to problem setting. In terms of the optimality condition, our approach can approximately achieve PO, while previous works can only achieve an ineffective PS for our problem, see section 2. In terms of optimization complexity, our model is nonconvex, nonlinear and nonsmooth, thus being nontrivial to solve. For example, the introduction of $\mathbf{\Pi}_{[0,x_u]}$ lead to nonconvexity of the problem. See more in section 3, section 4.

3. (Application) We apply the model and algorithm on traffic modeling, see section 6. The proposed algorithm achieves a 5x speedup by exploiting the sparsity structure of the model on a real-world transportation network with 3000 routes.

2 Derivation and theory of Onp

Now we derive and analyze (Onp).

Graph. A traffic network is a *simple strongly-connected digraph*[2] $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{R})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E} = \{e_1, e_2, \ldots\}$. The *route set* $\mathcal{R} \subseteq \text{PowerSet}(\mathcal{E})$ contains *route* r. The assignment between $\mathcal{R}, \mathcal{E}$ is denoted by the matrix A

$$A = \Big[a_1, \ldots, a_{|\mathcal{R}|}\Big]^\top \in \{0,1\}^{|\mathcal{R}|\times|\mathcal{E}|}. \qquad (\mathcal{R} - \mathcal{E} \text{ assignment})$$

Flow. $x = [x_1, \ldots, x_{|\mathcal{R}|}]^\top \in \mathbb{R}_+^{|\mathcal{R}|}$ is the amount of traffic passing each route[3]. We have $x \in [0, x_u]$ with $x_u \in \mathbb{R}_+^{|\mathcal{R}|}$ is the maximum amount of flow allowed in each route.

[1]This is a standard practical assumption; any violation can be handled by adding constraints and it will not change the subsequent analysis.

[2]We mean no loop, no multi-edge, and between any pair of nodes there is a path.

[3]E.g., x_3 is the number of users in route r_3.

symbol	Definition / Meaning
A, a_r	$\mathcal{R}$-$\mathcal{E}$ assignment matrix, an assignment vector r
B	$\lvert\mathcal{R}\rvert \times \lvert\mathcal{R}\rvert$ price elasticity matrix
$c^{\text{os}}, c^{\text{coe}}$	offset and coefficient of cost function on each edge
C	helper matrix (Definition 1)
$\mathcal{D}(p)$	distribution under parameter p
$\mathcal{E}, e$	edge set of graph $\mathcal{G}$, an edge in $\mathcal{E}$
$\mathcal{G}$	simple connected digraph with vertex $\mathcal{V}$
$\mathcal{K}, k, K$	commodity: set, index in $[1, \lvert\mathcal{K}\rvert]$, assignment matrix
l	lower bound of traffic needed for commodity
p	price on each route, a vector in $\mathbb{R}^{\lvert\mathcal{R}\rvert}$
$\mathbf{\Pi}$	orthogonal projection
Q	$\lvert\mathcal{R}\rvert \times \lvert\mathcal{R}\rvert$ quadratic coefficient matrix in (Onp)
$\mathcal{R}, r$	Route set and a particular route (a set of edges)
s	linear coefficient in $\mathbb{R}^{\lvert\mathcal{R}\rvert}$ in (Onp)
x	traffic flow, a vector in $\mathbb{R}^{\lvert\mathcal{R}\rvert}$
ζ	random noise, we assume $\zeta \sim \mathcal{N}(\mu, \Sigma)$
λ	price regularization parameter, $\lambda \in \mathbb{R}_+$

Table 1: Notation

Price for traffic control. $p \in \mathbb{R}_+^{|\mathcal{R}|}$ is the price on each route. p is used to influence the congestion level. We have $p \in [p_l, p_u]$.

Multi-commodity. We let the commodity set $\mathcal{K} := \{(s_1, t_1), \ldots\}$ contain source-sink pairs (s_k, t_k) that each traffic flow has to transport between. For each commodity k, we let $\mathcal{R}_k \subset \mathcal{R}$ be the set of $r \in \mathcal{R}$ such that (s_k, t_k) is the source-sink of r. Set $\mathcal{K}$ is represented by a matrix $K \in \{0,1\}^{|\mathcal{K}|\times|\mathcal{R}|}$, and $Kx = \sum_{r\in\mathcal{R}_k} x_r$ is the sum of traffic in each route r with respect to commodity k. The constraint $Kx \geq l$ enforces element-wise minimum-flow requirements, where $l \in \mathbb{R}_+^{|\mathcal{K}|}$ is the lower bound of traffic required, which means that the total flow on routes connecting the source s_k to the sink t_k must be at least l_k.

Congestion cost on an edge. Let cost_e be the congestion cost on an edge $e \in \mathcal{E}$. To model cost_e, we use a linear cost model following the literature on congestion games [12, 9] and traffic flow control [23]. Here cost_e is defined by constants $c^{\text{os}}, c^{\text{coe}} \in \mathbb{R}_+^{|\mathcal{E}|}$, where c^{os} is the constant congestion costs on e and c^{coe} is the rate of cost on e. Scaled by the flow x_r on route r, we have

$$\text{cost}_e \;=\; c_e^{\text{coe}} \sum_{r\ni e} x_r + c_e^{\text{os}}, \tag{2}$$

where $\sum_{r\ni e}$ means we sum x_r across routes that include edge e.

Congestion cost on the graph. We define $\text{cost}_{\mathcal{G}}$ as the sum of cost_r (the costs across route r), where cost_r is the sum of its constituent cost_e.

$$\text{cost}_{\mathcal{G}} = \sum_{r\in\mathcal{R}} \text{cost}_r = \sum_{r\in\mathcal{R}} \sum_{e\in r} \text{cost}_e \overset{(2)}{=} \sum_{r\in\mathcal{R}} \sum_{e\in r} \left(c_e^{\text{coe}} \sum_{r\ni e} x_r + c_e^{\text{os}} \right).$$

The expression $\text{cost}_{\mathcal{G}}$ can be simplified using the route-edge assignment A (telling which e is in which r), and a helper matrix:

Definition 1. (Helper matrix) *Let $C \in \mathbb{R}_+^{|\mathcal{R}|\times|\mathcal{E}|}$ as $C_{r,e} = c_e^{coe}$ if $e \in r$ and $C_{r,e} = 0$ otherwise.*

Theorem 1. *Using assignment A, the vector c^{coe}, Hadamard product $\odot$ and tensor product $\otimes$, then*

$$CA^\top \;=\; \big([c^{coe} \otimes \mathbf{1}_{|\mathcal{R}|}]^\top \odot A\big)A^\top \;=\; A Diag(c^{coe}) A^\top. \tag{3}$$

Proof. By definition, C, A are $|\mathcal{R}| \times |\mathcal{E}|$ matrices that

$$C_{r,e} = \begin{cases} c_e^{\text{coe}} & e \in r, \\ 0 & \text{else.} \end{cases} \quad A_{r,e} = \begin{cases} 1 & e \in r, \\ 0 & \text{else.} \end{cases}$$

Next, $[c^{\text{coe}} \otimes \mathbf{1}_{|\mathcal{R}|}]^\top = \mathbf{1}_{|\mathcal{R}|}(c^{\text{coe}})^\top \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{E}|}$. By the structure of C and A. $\square$

Remark. *The matrix $CA^\top$ is symmetric by $ADiag(c^{coe})A^\top$ in* (3).

Total congestion cost from traffic flow. We define cost_r under the flow amount x as $\text{cost}_r(x) = \langle Ca_r, x\rangle + \langle a_r, c^{\text{os}}\rangle$, where a_r is the rth column of A. Then $x_r\text{cost}_r(x)$ is $\text{cost}_r(x)$ scaled by x_r, representing the total congestion cost on route r. Thus, the congestion cost on the whole graph is the sum over all $r \in \mathcal{R}$ as

$$\text{cost}_{\text{total}} = \sum_{r\in\mathcal{R}} x_r\text{cost}_r(x) = \langle AC^\top x + Ac^{\text{os}}, x\rangle = \frac{1}{2}\langle Qx, x\rangle - \langle s, x\rangle,$$

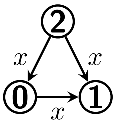
where $Q \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$ and $s \in \mathbb{R}^{|\mathcal{R}|}$ are defined as

$$Q = 2AC^\top \overset{\text{Theorem 1}}{=} 2A\text{Diag}(c^{\text{coe}})A^\top, \qquad s = -Ac^{\text{os}}. \tag{4}$$

Note that Q is only for analysis and is never computed, see section 3.

Quadratic objective. In (Onp), we propose the cost as the sum of $\text{cost}_{\text{total}}$ and the platform price $\frac{\lambda}{2}\|p\|_2^2$ with $\lambda \geq 0$.

Toy example. A 3-node graph with edge $\mathcal{E} = \big\{(0,1), (2,0), (2,1)\big\}$ and route $\mathcal{R} = \big\{[0,1], [2,0], [2,0,1], [2,1]\big\}$ is drawn.



$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; CA^\top = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \not\succeq \\ \not\preceq \end{matrix} \mathbf{0}.$$

Thus, the matrix Q is indefinite.

Oblivious user assumption. In the modeling above, we assume that users (e.g. drivers) do not know that a congestion game exists, so the traffic flow x on each route $r \in \mathcal{R}$ is *completely characterized* by the price signals p. This contrasts with classical congestion games [24, 18], where the total flow amount $\sum_r x_r$ is held constant and users interact within the network. Our oblivious-user assumption implies that alternative transportation options are available for model (Onp), allowing the traffic flow within the network to shift to other networks, and vice versa. E.g., in network pricing for freight transport,*alternative transportation* may include sea or air shipping as substitutes for road freight.

Optimization with decision-dependent distributions. In (Onp) both x, p are optimization variables and $\mathbb{E}_{x\sim\mathcal{D}(p)}$ is known as *decision-dependent*, meaning the optimality of solutions is affected by the relationship $x \sim \mathcal{D}(p)$. As our formulation lies within the framework of optimization with decision-dependent distributions, the optimality conditions of PO and PS seem to be relevant. However, the current research primarily aims to solve PS in (1), e.g. [5, Def.3.2]; and achieving this may result in a substantial gap from PO in (1). Note that the algorithms in [5] are not applicable to

$$\min_x \mathop{\mathbb{E}}_{z\sim\mathcal{D}(x)} \ell_1(z) + \lambda\ell_2(x).$$

Using the update $x_{t+1} = S_{x_t}(x_t)$ from [5, Sec4.1] reduces the problem to minimizing $\lambda\ell_2(x)$, which satisfies PS in (1) but can result in a large gap from PO in (1).

Distribution. $\mathcal{D}(p) = \mathbf{\Pi}_{[0,x_u]}(Bp + \zeta)$ is a *projected distribution*, the jth component of a vector $y = Bp + \zeta$ is

$$\mathbf{\Pi}_{[0,x_u]}(y_j) = \begin{cases} 0 & \text{if } y_j \leq 0, \\ y_j & \text{if } 0 < y_j < [x_u]_j, \\ [x_u]_j & \text{if } y_j \geq [x_u]_j. \end{cases} \tag{5}$$

Price elasticity. The matrix $B \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$ is called elasticity [13, Sect.7.2]. We adopt the case that B is negative semi-definite in section 6.

3 Monte Carlo, Gradient and Hessian

We propose an algorithm to solve (Onp). In this section, we first explain why we need efficient algorithm. Then we reformulate (Onp), eliminate the variable x, followed by Monte Carlo approximation. In the remaining part we focus entirely on deriving the expression of gradient and Hessian, and discuss sparsity-aware efficient computation. We present the algorithm in section 5.

Source of complexity. As $\mathcal{R} \subset \text{PowerSet}(\mathcal{E})$ and $|\mathcal{R}| \leq 2^{|\mathcal{E}|}$, the number of routes can grow exponentially, causing high dimensionality in vectors $x, p \in \mathbb{R}^{|\mathcal{R}|}$ and matrices $B, Q \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$.

Reformulation. We eliminate the variable x. Replacing x in (Onp) by $x(p,\zeta) = \mathbf{\Pi}_{[0,x_u]}(Bp+\zeta)$ gives the following stochastic problem

$$\begin{array}{cl} \underset{p\in[p_l,p_u]}{\operatorname{argmin}} & \dfrac{\lambda}{2}\|p\|_2^2 + \underset{\zeta\sim\mathcal{N}(\mu,\Sigma)}{\mathbb{E}} \Big\langle \dfrac{1}{2} Qx(p,\zeta) - s, \quad x(p,\zeta) \Big\rangle \\ \text{s.t.} & K\,\mathbb{E}_{\zeta\sim\mathcal{N}(\mu,\Sigma)}[x(p,\zeta)] \geq l. \end{array} \tag{Onp-x}$$

We note that, for the reformulation, $x(p,\zeta) = \mathbf{\Pi}_{[0,x_u]}(Bp+\zeta)$ eliminates the flow bound constraint $x \in [0, x_u]$ inside (Onp) in (Onp-x) by Assumption 1. Also $x(p,\zeta)$ is Lipschitz continuous in p.

Lemma 1. *$x(p,\zeta)$ is $\|B\|_2$−continuous in p, independent of ζ.*

Proof. We have that

$$\begin{array}{rcl} \|x(p_1,\zeta) - x(p_2,\zeta)\|_2 & = & \|\Pi(Bp_1+\zeta) - \Pi(Bp_2+\zeta)\|_2 \\ & \leq & \|(Bp_1+\zeta) - (Bp_2+\zeta)\|_2 \\ & \leq & \|B\|_2\|p_1-p_2\|_2, \end{array}$$

where the first inequality is by $\mathbf{\Pi}$ is 1-Lipschitz [1]. $\square$

3.1 Deriving the Monte Carlo approximation.

We solve (Onp-x) by Monte Carlo (MC) approximation. We draw N i.i.d. samples $\Xi = [\zeta^{(1)}, \ldots, \zeta^{(N)}]$ (called *scenarios* in stochastic program [20]) from $\mathcal{N}(\mu,\Sigma)$. Let $\mathbf{1}_N \in \mathbb{R}^N$ be vector-of-1, we define the vector $y^{(i)}$ and the matrix $Y = [y^{(1)},\ y^{(2)} \ldots]$ as

$$\begin{array}{rcl} y_p^{(i)} & = & Bp + \zeta^{(i)}, \\ Y_p & = & Bp\mathbf{1}_N^\top + \Xi. \end{array} \tag{6}$$

We let $x^{(i)}$ as projected $y^{(i)}$, and being contained in a matrix X

$$\begin{array}{rcl} x_p^{(i)} & = & \mathbf{\Pi}_{[0,x_u]}\big(y_p^{(i)}\big), \\ X_p & = & \mathbf{\Pi}_{[0,x_u\mathbf{1}_N^\top]}(Y_p). \end{array} \tag{7}$$

We emphasize that the relationship between X, Y (and $x^{(i)}, y^i$) is nonlinear. Now, the objective of the MC reformulation is

$$f_N(p) \;:=\; \frac{\lambda}{2}\|p\|_2^2 + \frac{1}{N}\sum_{i=1}^N \Big[\frac{1}{2}\big\langle Qx_p^{(i)},\, x_p^{(i)}\big\rangle - \big\langle s, x_p^{(i)}\big\rangle\Big], \tag{8}$$

and the constraint function in the MC reformulation is

$$c_N(p) \;=\; l - K\frac{1}{N}\sum_{i=1}^N x_p^{(i)}. \tag{9}$$

We approximate (Onp-x) by a constrained nonlinear least-squares:

$$\underset{p\in[p_l,p_u]}{\operatorname{argmin}}\ f_N(p) \qquad \text{s.t.} \qquad c_N(p) \leq 0. \tag{Onp-x-MC}$$

The convexity of f_N and c_N is a non-trivial issue, see section 4.

Consistency of large-N approximation. MC replaces the expectation in (Onp-x) with an empirical average in (Onp-x-MC). We now show this approach works, which is nontrivial due to the presence of $\Pi_{[0,x_u]}$ and $f_N(p)$ is nonconvex in p.

Lemma 2. *$f_N(p)$ and $c_N(p)$ are continuous in p.*

Proof. $h^{(i)}$ is quadratic in $x(p,\zeta)$. By Lemma 1, $h^{(i)}$ are continuous in p, so f_N is continuous. The result for $c_N(p)$ is similar. □

Theorem 2. *For N sufficiently large, f_N approximates $f(p)$ well:*

$$\sup_{p\in[p_l,p_u]} \big|f_N(p) - f(p)\big| \xrightarrow{\text{a.s.}} 0, \qquad \sup_{p\in[p_l,p_u]} \big|c_N(p) - c(p)\big| \xrightarrow{\text{a.s.}} 0.$$

where a.s. denotes almost surely[4].

Proof. By the bounded constraint $[p_l, p_u]$, continuity lemma 2 and strong law of large number [20, Ch.7]. □

Why MC? We consider MC to solve the stochastic problem (Onp-x) due to **simplicity**: (Onp-x) involves multivariate CDFs and high-dimensional integrals, which are mathematically and computationally complicated. MC gives a simpler treatment. Also, recall section 1, **MC can potentially handle more complicated $\mathbf{\Pi}$** where a simple closed-form expression for $\mathbb{E}$ is unavailable, such as $\mathbf{\Pi}$ is not defined coordinate-wise: e.g., projection onto a linear inequality constraint as $\exists\, \mathcal{G} \subseteq \mathcal{R} : l \leq \sum_{r\in\mathcal{G}} x_r \leq u$.

3.2 Gradient derivation and computation

We now compute the gradient $\nabla_p f_N(p)$. By chain rule,

$$\nabla_p f_N(p) \overset{(8)}{=} \lambda p + \frac{1}{N}\sum_{i=1}^{N} \Big[\nabla_p x_p^{(i)}\Big]^\top \Big(Q x_p^{(i)} - s\Big). \tag{10}$$

Consider three Jacobian $\nabla_p x_p^{(i)}$, $\nabla_p y_p^{(i)}$, $\nabla_{y_p^{(i)}} x_p^{(i)}$ under chain rule:

$$\nabla_p x_p^{(i)} \overset{\text{chain rule}}{=} \nabla_{y_p^{(i)}} x_p^{(i)} \nabla_p y_p^{(i)} \overset{(6),(7)}{=} \Big\{\nabla_{y_p^{(i)}}\Big(\mathbf{\Pi}_{[0,x_u]}(y_p^{(i)})\Big)\Big\} B. \tag{11}$$

By (5), the Jacobian

$$\mathbf{J}^{(i)} \;:=\; \nabla_{y_p^{(i)}} x_p^{(i)} \overset{(7)}{=} \nabla_{y_p^{(i)}}\Big(\mathbf{\Pi}_{[0,x_u]}(y_p^{(i)})\Big) \tag{12}$$

has nonsmooth corners (at the boundary $\{0, x_u\}$). As $\mathbf{\Pi}$ is 1-Lipschitz, so the entry $[\mathbf{J}^{(i)}]_{j,j}$ in terms of Clarke subdifferential [4, 17] is

$$[\mathbf{J}^{(i)}]_{j,j} = \underbrace{\partial \mathbf{\Pi}_{[0,x_u]}(y_j^{(i)})}_{\text{subdifferential}} = \begin{cases} \{0\} & y_j^{(i)} \notin \big[0, [x_u]_j\big], \\ \{1\} & y_j^{(i)} \in \big(0, [x_u]_j\big), \\ [0,1] & y_j^{(i)} \in \big\{0, [x_u]_j\big\}. \end{cases} \tag{13}$$

Simplification by statistics. The event $[y_p^{(i)}]_j \in \{0, [x_u]_j\}$ in (13) for $\zeta \sim \mathcal{N}(\mu, \Sigma)$ has a probability zero thus we drop the boundary case of $\mathbf{J}^{(i)}$. Let $\omega^{(i)}_{0<y_p^{(i)}<x_u}$ be the characteristic vector for each i and the matrix $\mathbf{\Omega}$ storing $\omega^{(i)}$

$$\Big[\omega^{(i)}_{0<y_p^{(i)}<x_u}\Big]_j = \begin{cases} 1 & \text{if } 0 < \big[y_p^{(i)}\big]_j < [x_u]_j, \\ 0 & \text{else.} \end{cases} \tag{14a}$$

$$[\mathbf{\Omega}]_{j,i} = \begin{cases} 1 & \text{if } 0 < [Y_p]_{j,i} < [x_u]_j, \\ 0 & \text{else.} \end{cases} \tag{14b}$$

Using (14a), we have the following almost surely equivalence

$$\mathbf{J}^{(i)} \overset{\text{a.s.}}{=} \operatorname{Diag}\Big(\omega^{(i)}_{0<y_p^{(i)}<x_u}\Big) \overset{(13),(14b)}{=} \operatorname{Diag}\big([\mathbf{\Omega}]_{:,i}\big). \tag{15}$$

[4] It means we have probability 1.

Let $\odot$ be Hadamard product, put (11),(15) into (10) gives

$$\nabla_p f_N(p) \overset{\text{a.s.}}{=} \lambda p + B^\top \frac{1}{N}\sum_{i=1}^N \mathrm{Diag}\big([\mathbf{\Omega}]_{:,i}\big)\Big(Qx_p^{(i)} - s\Big) \overset{(14b),(7)}{=} \lambda p + B^\top \frac{1}{N}\Big(\mathbf{\Omega} \odot \big(QX_p - s\mathbf{1}_N^\top\big)\Big)\mathbf{1}_N. \quad (16)$$

The matrix form is cleaner mathematically and more efficient computationally, as it avoids the loop over samples and exploitation of the sparsity pattern with active sets.

Sparsity-aware computation. As N, the number of samples in (Onp-x-MC), can be huge for a tight approximation of (Onp-x), it is important to exploit sparsity for computational efficiency. The gradient computation in (16) requires evaluating the term QX_p. Computing QX_p naively costs $\mathcal{O}(N|\mathcal{R}|^2)$, which is prohibitive for large N and $|\mathcal{R}|$. However, we can exploit sparsity: by (4) and the fact that the route–edge assignment matrix A (Section 2) is sparse, the matrix $Q = 2AA^\top$ is sparse[5] and X_p is sparse by the projection. Below is a sparsity-aware implementation of the sum in (16):

- Given A, B, $s = -Ac^{\mathsf{os}}$ and the already computed $X_p, \mathbf{\Omega}$
- Initialize $M = \text{zeros}(|\mathcal{R}|, N)$ (to store $\mathbf{\Omega} \odot (QX_p - s\mathbf{1}_N^\top)$)
- For each sample $i = 1, \dots, N$ in parallel:

- $S_i = \{j : [\mathbf{\Omega}]_{j,i} = 1\}$ (active set for sample i)
- $A_{S_i} = A(S_i, :)$ (rows of A indexed by S_i)
- $J_i = \{j : \exists k \in S_i \text{ s.t. } A_{k,j} \neq 0\}$ (unique column indices in A_{S_i})
- $u_{J_i} = (A_{S_i}(:, J_i))^\top [X_p(S_i, i)]$
- $w_{S_i} = A_{S_i}(:, J_i)\, u_{J_i}$
- $r_{S_i} = w_{S_i} - s(S_i)$ (residual on active set)
- $M(S_i, i) = r_{S_i}$ (store result)

- Compute $\nabla_p f_N(p) = \lambda p + \frac{1}{N} B^\top (M\mathbf{1}_N)$

Computing $\mathbf{\Omega} \odot (QX_p - s\mathbf{1}_N^\top)$ now has a total cost of

$$\mathcal{O}\left(\sum_{i=1}^N [2 \cdot \mathrm{nnz}(A_{S_i}(:, J_i)) + |S_i|]\right) \ll \mathcal{O}(N|\mathcal{R}|^2),$$

where $\sum_{i=1}^N \mathrm{nnz}(A_{S_i}(:, J_i)) \leq N \cdot \min\big\{\mathrm{nnz}(A), \max_i |S_i| \cdot \max_i |J_i|\big\}$.

Gradient of $c_N(p)$. We have $\nabla_p c_N(p) \in \mathbb{R}^{|\mathcal{K}|\times|\mathcal{R}|}$ as

$$\nabla_p c_N(p) \overset{(9)}{=} -K\frac{1}{N}\sum_{i=1}^N \nabla_p x_p^{(i)} \overset{(11),(13)}{=} -K\Big(\frac{1}{N}\sum_{i=1}^N \mathbf{J}^{(i)}\Big)B. \quad (17)$$

Let $\boldsymbol{d} = \mathbf{\Omega}\mathbf{1}_N \in \mathbb{R}^{|\mathcal{R}|}$ be the vector counting active samples per route, then

$$\sum_{i=1}^N \mathbf{J}^{(i)} \overset{(15)}{=} \sum_{i=1}^N \mathrm{Diag}\big([\mathbf{\Omega}]_{:,i}\big) = \mathrm{Diag}\left(\mathbf{\Omega}\mathbf{1}_N\right) = \mathrm{Diag}(\boldsymbol{d}),$$

so in compact matrix notation, (17) becomes

$$\begin{aligned} \nabla_p c_N(p) &\overset{\text{a.s.}}{=} -K\frac{1}{N}\mathrm{Diag}\left(\boldsymbol{d}\right)B, \\ \boldsymbol{d} &:= \mathbf{\Omega}\mathbf{1}_N. \end{aligned} \quad (18)$$

Sparsity-aware computation. $\nabla_p c_N$ in (18) requires evaluating $K\mathrm{Diag}(\mathbf{\Omega}\mathbf{1}_N)B$. As $K \in \{0,1\}^{|\mathcal{K}|\times|\mathcal{R}|}$ is sparse and $\mathrm{Diag}(\mathbf{\Omega}\mathbf{1}_N)$ is diagonal, we exploit sparsity for efficient implementation:

[5] Where A has been pre-scaled as $A \leftarrow A\,\mathrm{Diag}(\sqrt{c^{\mathsf{coe}}})$.

- Store K, B, precompute $\boldsymbol{d} = \boldsymbol{\Omega}\mathbf{1}_N$ with cost $\mathcal{O}(N|\mathcal{R}|)$
- Compute $\nabla_p c_N(p) = -\frac{1}{N}K(B^\top \mathrm{Diag}(\boldsymbol{d}))^\top$
- Exploit sparsity of K: for each nonzero $K_{k,j}$:

– Multiply column $B_{:,j}$ by d_j and accumulate into row k of gradient

Computing $K\mathrm{Diag}(\boldsymbol{d})B$ has cost $\mathcal{O}(\mathrm{nnz}(K)|\mathcal{R}|)$, which is efficient for sparse K.

3.3 Hessian of f_N and c_N

$\nabla_p^2 f_N$ is obtained by applying derivative on (16): for each sample i, the linearity of ∇_p gives:

$$\nabla_p \left[\mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)\big(Qx_p^{(i)} - s\big)\right] \;=\; \nabla_p \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big) \cdot \big(Qx_p^{(i)} - s\big) + \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big) Q \nabla_p x_p^{(i)}.$$

For the first term, we can drop it by the following lemma:

Lemma 3. $\nabla_p Diag\big([\boldsymbol{\Omega}]_{:,i}\big) = 0$ *almost everywhere for* $p \in \mathbb{R}^{|\mathcal{R}|}$.

Proof. $\omega^{(i)}$ changes only when $y_p^{(i)}$ hits the boundary $\{0, x_u\}$, which occurs with probability zero. □

For the second term, $\nabla_p x_p^{(i)} = \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)B$ by (11) and (15). Thus

$$\nabla_p^2 f_N(p) \overset{\text{a.s.}}{=} \lambda I + \frac{1}{N} B^\top \left(\sum_{i=1}^{N} \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big) Q \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)\right) B. \tag{19}$$

While the gradient had a clean matrix form using $\boldsymbol{\Omega}$, the Hessian requires a sum over samples that cannot be directly expressed as a simple matrix operation with $\boldsymbol{\Omega}$.

Sparsity-aware computation for. Computing (19) requires evaluating the sum $\sum_{i=1}^N \mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)Q\mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)$. As $Q = 2AA^\top$ is sparse and $\boldsymbol{\Omega}$ indicates active routes, we can exploit sparsity. Note that $\mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)Q\mathrm{Diag}\big([\boldsymbol{\Omega}]_{:,i}\big)$ is simply the submatrix Q_{S_i,S_i}, where $S_i = \{j : [\boldsymbol{\Omega}]_{j,i} = 1\}$ is the active set for sample i.

- Store A, B, and given $\boldsymbol{\Omega}$ (already computed)
- Initialize $H_{\text{inner}} = \text{zeros}(|\mathcal{R}|, |\mathcal{R}|)$ (to accumulate the inner sum)
- For each sample $i = 1, \ldots, N$ in parallel:

– $S_i = \{j : [\boldsymbol{\Omega}]_{j,i} = 1\}$ (active set for sample i)
– $A_{S_i} = A(S_i, :)$ (rows of A indexed by S_i)
– Compute $Q_{S_i} = A_{S_i}A_{S_i}^\top$ (local Q on active set)
– Accumulate: $H_{\text{inner}}(S_i, S_i) \mathrel{+}= Q_{S_i}$

- Compute $\nabla_p^2 f_N(p) = \lambda I + \frac{1}{N}B^\top H_{\text{inner}} B$

Computing the Hessian now has a total cost of:

$$\mathcal{O}\left(\sum_{i=1}^{N} |S_i|^2 + |\mathcal{R}|^3\right) \;\le\; \mathcal{O}\big(N\big(\max_i |S_i|\big)^2 + |\mathcal{R}|^3\big) \;\ll\; \mathcal{O}\big(N|\mathcal{R}|^2 + |\mathcal{R}|^3\big).$$

Lastly, for c_N, by (3) we have

$$\nabla_p^2 c_N(p) \overset{a.s.}{=} 0.$$

4 Nonconvexity and Convexity

We now discuss the convexity of f_N, which affects our model optimality condition as well as algorithm design. First we give an example to show that f_N is nonconvex in general. Then we give a lemma on the definiteness of $\nabla_p^2 f_N$. And then, we use statistics to argue that we can ignore the points $p \in [p_l, p_u]$ where f_N is nonconvex, and thus treating $f_N(p)$ practically convex in the algorithm. Lastly, we discuss the nonconvexity of c_N.

4.1 f_N is nonconvex

Given the indefiniteness of Q (see the toy example in section 2), by (19), it is easy to see that $\nabla_p^2 f_N(p)$ can be indefinite, and $f_N : \mathbb{R}^{|\mathcal{R}|} \to \mathbb{R}$ in (Onp-x-MC) is nonconvex in p due to the presence of the piecewise projection $\mathbf{\Pi}$. Here is a scalar example, take $\zeta^{(i)} = Q = 0$, $p = s = 1$, $x_u = 0.5$, $B = -1$, then $x(p) = \mathbf{\Pi}_{[0,0.5]}(p) = \min\{0.5, \max\{0, p\}\}$ and

$$f_N(p) = \frac{\lambda}{2}p^2 - \min\Big\{0.5,\ \max\{0, -p\}\Big\}, \tag{20}$$

which is generally nonconvex for $\lambda > 0$, see Fig. 1.

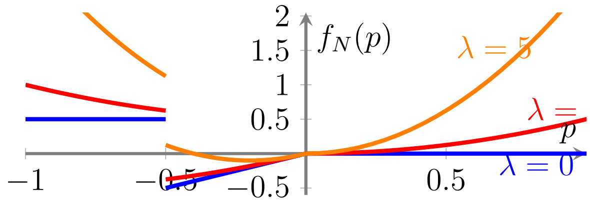


Figure 1: The plot of (20) for three different values of λ. The function is nonconvex globally because of the kink at $p = 0$.

4.2 Condition for local convexity

If p is elementwise nonnegative ($p \geq 0$), by the nonnegativity of $\mathbf{J}^{(i)}$ and Q, we have that

$$\Big\langle [\nabla_p^2 f_N(p)]p, p \Big\rangle \ = \ \lambda \|p\|_2^2 + \bigg\langle \Big(\frac{1}{N} \sum_{i=1}^{N} \mathbf{J}^{(i)} Q \mathbf{J}^{(i)} \Big) Bp,\ Bp \bigg\rangle \geq 0,$$

thus the objective function of (Onp-x-MC) is locally convex wrt p when $p \geq 0$. This can also be observed in Fig. 1, where the parts of the curve f_N for $p \geq 0$ are convex for the three chosen λ.
Now for general p we discuss a related issue for the convexity of f_N, which is the definiteness of $\nabla_p^2 f_N$. Using the compact notation $\mathbf{J}^{(i)}$ in (15), now for all p, we require

$$\begin{aligned} \nabla_p^2 f_N(p) \succ 0 \quad & \overset{(19)}{\iff} \quad \bigg\langle \Big(\lambda I + B^\top \Big(\frac{1}{N} \sum_{i=1}^{N} \mathbf{J}^{(i)} Q \mathbf{J}^{(i)} \Big) B \Big) p,\ \ p \bigg\rangle > 0 \\ & \iff \ \lambda \|p\|_2^2 > - \bigg\langle B^\top \Big(\frac{1}{N} \sum_{i=1}^{N} \mathbf{J}^{(i)} Q \mathbf{J}^{(i)} \Big) Bp,\ \ p \bigg\rangle. \end{aligned}$$

As $\mathbf{J}^{(i)}$ changes with p, thus it is inefficient to compute λ for all the p, thus we consider the extreme case for λ that $\nabla_p^2 f_N(p) \succ 0$.

Lemma 4. *Let $\lambda_{\min}(Q)$ be the smallest eigenvalue of Q. The smallest eigenvalue of the matrix*

$$B^\top \Big(\frac{1}{N} \sum_{i=1}^{N} \mathbf{J}^{(i)} Q \mathbf{J}^{(i)} \Big) B$$

over all possible $\mathbf{J}^{(i)}$, is $\lambda_{\min}(Q)(\lambda_{\min}(B))^2$.

Proof. Let $\tilde{A} = \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \mathbf{J}^{(i)} Q \mathbf{J}^{(i)}$. First note that $Q, \mathbf{J}^{(i)}$ are symmetric (By section 2 for Q and $\mathbf{J}$ is diagonal by (15)), thus $B^\top \tilde{A} B$ is symmetric and its smallest eigenvalue is

$$\min_{x \neq 0} \frac{x^\top (B^\top \tilde{A} B) x}{\|x\|^2} \overset{y = Bx}{=} \min_{x \neq 0} \frac{y^\top \tilde{A} y}{\|x\|^2}.$$

Now we bound $y^\top \tilde{A} y$.

$$y^\top \tilde{A} y \;=\; \frac{1}{N}\sum_{i=1}^{N} y^\top \mathbf{J}^{(i)} Q \mathbf{J}^{(i)} y \;=\; \frac{1}{N}\sum_{i=1}^{N} (\mathbf{J}^{(i)} y)^\top Q (\mathbf{J}^{(i)} y).$$

As Q is symmetric, so $(\mathbf{J}^{(i)} y)^\top Q (\mathbf{J}^{(i)} y) \geq \lambda_{\min}(Q)\|\mathbf{J}^{(i)} y\|^2$, thus

$$y^\top \tilde{A} y \;\geq\; \lambda_{\min}(Q)\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{J}^{(i)} y\|^2.$$

As $\mathbf{J}^{(i)} \overset{\text{a.s.}}{=} \mathrm{Diag}\Big(\omega^{(i)}_{0<y_p^{(i)}<x_u}\Big)$ is 0-1 diagonal, by sparsity index $S_i = \Big\{j \,:\, 0 < [y^{(i)}(p)]_j < [x_u]_j\Big\}$ we have $\|\mathbf{J}^{(i)} y\|^2 = \sum_{j \in S_i} y_j^2$,

$$\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{J}^{(i)} y\|^2 \;=\; \sum_j \left(\frac{1}{N}\sum_{i:j\in S_i} 1\right) y_j^2 \;=\; \sum_j f_j y_j^2$$

where $f_j = \dfrac{1}{N}\sum_{i:j\in S_i} 1 \leq 1$. Thus:

$$y^\top \tilde{A} y \;\geq\; \lambda_{\min}(Q)\sum_j f_j y_j^2.$$

The lower bound is maximized when $f_j = 1$ for all j where $y_j \neq 0$, i.e. $\sum_j f_j y_j^2 = \|y\|^2$. This is achieved if $\mathbf{J}^{(i)} y = y$ for all i, i.e., $\mathbf{J}^{(i)}$ has 1s on the support of y. Now, at minimum,

$$\min_{x\neq 0} \frac{y^\top \tilde{A} y}{\|x\|^2} \;=\; \min_{x\neq 0} \frac{\lambda_{\min}(Q)\|y\|^2}{\|x\|^2} \overset{y=Bx}{=} \lambda_{\min}(Q) \min_{x\neq 0} \frac{x^\top B^\top B x}{\|x\|^2},$$

the minimum is achieved at $(\lambda_{\min}(B^\top B)) = \sigma^2_{\min}(B)$, where $\sigma_{\min}$ is the smallest singular value. □

The Lemma 4 tells the condition on λ that when will $\nabla^2 f_N$ be positive definite. λ is a problem input of (Onp), and as a result, in general λ may not satisfy the condition of Lemma 4.

- In case λ satisfies the condition, then $\nabla^2 f_N$ being positive definite, and then f_N is convex wrt p locally. These useful properties will be beneficial for the algorithm.
- In case λ does not satisfy the condition and leading to f_N being noncvonex, we will implement a method to deal with the nonconvexity, see section 5

4.3 f_N is convex almost surely for some λ

Now we apply the same statistical argument used when deriving (15) to argue that f_N is convex wrt p almost surely. Recall from section 3.2, the event $[y_p^{(i)}]_j \in \{0, [x_u]_j\}$ in (13) for the random variable $\zeta \sim \mathcal{N}(\mu, \Sigma)$ has a probability zero, meaning that we always have the strict inequality $0 < [y_p^{(i)}]_j < [x_u]_j$ almost surely. Thus, in terms of optimization algorithm, the iterate p_k such that the strict inequality $0 < [y_p^{(i)}]_j < [x_u]_j$ holds for for all i has a probability one. As a result, we can treat the function f_N is convex wrt p within the feasible region $[p_l, p_u]$. This result will lead to the fact that, at the global optimum, the PO optimality in (1) is achieved almost surely.

4.4 Convexity of the constraint set $c_N(p) \leq 0$

The function c_N in (Onp-x-MC) defines the feasible set $\{p : c_N(p) \leq 0\}$ of the problem. The feasible set is nonconvex in general, because $\sum_{i=1}^N \mathbf{\Pi}_{[0,x_u]}(Bp + \zeta^{(i)})$ is generally nonconvex in p. Although the projection $\mathbf{\Pi}_{[0,x_u]}(\cdot)$ is convex in its argument, the composition $p \mapsto \mathbf{\Pi}_{[0,x_u]}(Bp + \zeta^{(i)})$ is piecewise and nonconvex when B has mixed signs or the active set changes with p. See Fig. 2 for an example in $\mathbb{R}^2$.

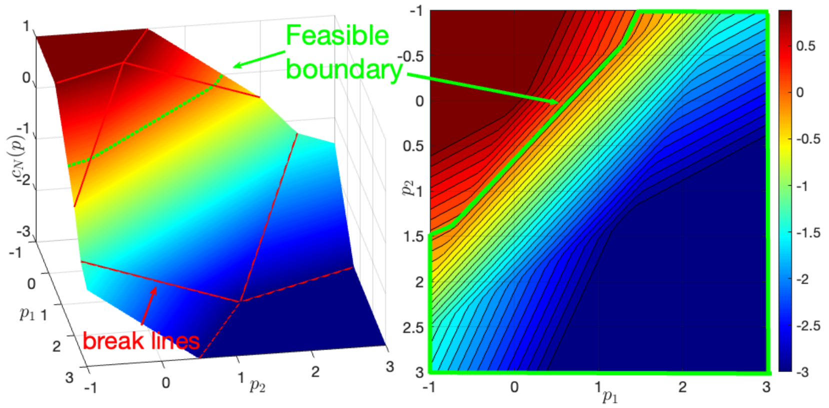


Figure 2: Left: the surface of a c_N in $\mathbb{R}^2$. Right: the contour plot of c_N. The feasible region is a nonconvex polygon with the green boundary.

5 TR-SQP algorithm

Following the discussion in section 4, (Onp-x-MC) is a constrained problem with nonconvex feasible set and nonconvex objective function. Such double nonconvexity limited our choice of optimization algorithms, leading us to use a specialized method. Among all the available methods, we use Trust-Region Sequential Quadratic Program (TR-SQP) to solve (Onp-x-MC). Here are reasons why we choose TR-SQP.

- The problem class (21) in TR-SQP subsumes our reformulation problem (Onp-x-MC).
- The SQP problem (22) for finding a direction d can utilize the sparsity structure of $\nabla f_p, \nabla c_p$, $\nabla^2 f_p$ and also $\nabla^2 c_N \overset{a.s.}{=} 0$, the computation is efficient.
- As discussed in section 4, the functions f_N, c_N in (Onp-x-MC) are nonconvex. The nonconvexity is handled by TR to ensure global optimality.
- TR-SQP is numerically stable, useful for our purpose.

We briefly review TR-SQP. It is a method for solving nonlinear constrained optimization problems of the form:

$$\begin{array}{cl} \underset{p \in \mathbb{R}^n}{\operatorname{argmin}} & f(p) \\ \text{s.t.} & c_j(p) \geq 0, \quad j \in \{1, 2, \ldots\}, \end{array} \tag{21}$$

where we remark that c_j here refers to general constraint, including $c_N(p) \leq 0$ and $p \in [p_l, p_u]$ in (Onp-x-MC). SQP iteratively solves a quadratic approximation of the Lagrangian:

$$\begin{array}{cl} \underset{d}{\operatorname{argmin}} & \dfrac{1}{2}\big\langle B_k d, d \big\rangle + \big\langle \nabla f(p_k), d \big\rangle \\ \text{s.t.} & \big\langle \nabla c(p_k), d \big\rangle + c(p_k) = 0 \end{array} \tag{22}$$

for obtaining the direction d. In (22), the term B_k approximates the Hessian of the Lagrangian:

$$B_k \approx \nabla^2 L(p_k, \gamma_k), \qquad \text{where} \qquad L(p_k, \gamma_k) = f(p) + \big\langle \gamma, c(p) \big\rangle.$$

TR-SQP augments SQP by imposing a *trust-region constraint* on the direction d as $\|d\| \leq \Delta_k$, where Δ_k is the trust-region radius. This ensures stability and global convergence, particularly when the quadratic model poorly approximates the true objective or constraints, and for a nonconvex objective in our case. The step is accepted or rejected based on the ratio of actual to predicted reduction in a merit function. TR-SQP thus combines the fast local convergence of SQP with the global reliability of trust-region strategies. We refer to [14] for the details. Algorithm 9 shows the pseudocode of TR-SQP.

Algorithm 1: TR-SQP

Input: Initial p_0, initial TR radius Δ_0, tolerance ϵ

for $k = 1, 2, \ldots$ *if* $\|\nabla L(p_k, \gamma_k)\| > \epsilon$ **do**

  Form the quadratic model of the Lagrangian:

$$m_k(d) = \frac{1}{2}\big\langle B_k d, d\big\rangle + \big\langle \nabla f(p_k), d\big\rangle.$$

  Solve TR subproblem:

$$\min_d m_k(d) \quad \text{s.t.} \quad \big\langle \nabla c(p_k), d\big\rangle + c(p_k) = 0, \quad \|d\| \le \Delta_k$$

  Compute the TR ratio $\rho_k = \dfrac{f(p_k) - f(p_k + d_k)}{m_k(0) - m_k(d_k)}$

  if ρ_k *is sufficiently large* **then**

    $p_{k+1} \leftarrow p_k + d_k$ accept the step;

  else

    $p_{k+1} \leftarrow p_k$ reject the step;

  Update Δ_{k+1} based on ρ_k;

  Update Lagrange multipliers γ_{k+1} and B_{k+1};

return p_k

6 Experiment

In this section we report the experimental results, focusing entirely on computational efficiency. First we verify the TR-SQP approach works for solving (Onp-x-MC) on a toy problem. Then we showcase the TR-SQP algorithm, exploiting sparsity to achieve speedup. Lastly we show the result on a real-world dataset. All the experiments were conducted in MATLAB R2024b[6].

6.1 Verification on a toy problem

We consider a simple strongly-connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{R})$ in Fig.3 with $|\mathcal{V}| = 4$ nodes, $|\mathcal{E}| = 6$ edges, $|\mathcal{R}| = 16$ routes as

Start 1	**Start 2**	**Start 3**	**Start 4**
$\{1,2\}$	$\{2,3\}$	$\{3,4\}$	$\{4,1\}$
$\{1,2,3\}$	$\{2,3,4\}$	$\{3,4,1\}$	$\{4,1,2\}$
$\{1,2,3,4\}$	$\{2,3,4,1\}$	$\{3,4,1,2\}$	$\{4,1,2,3\}$
$\{1,2,4\}$	$\{2,4\}$		$\{4,3\}$
	$\{2,4,1\}$		

Generating the commodity matrix K. Here we set $|\mathcal{K}| = 2$ commodity with and the matrix $K \in \{0,1\}^{2\times 16}$ has two nonzero, located at $K_{1,2}$ and $K_{2,5}$, i.e., K is 6% sparse. The source-sink pairs are $(1,3)$ and $(2,3)$ in $\mathcal{K}$, where ▲ denotes source and ▼ denotes sink in the graph. We take $N = 100$ scenarios for $\zeta^1, \ldots, \zeta^{100}$.

Generating he price elasticity matrix B. We generate the matrix $B = -I + \epsilon$zero-diagonal$(SS^\top)$, where $S \in \mathbb{S}^{|\mathcal{R}|\times|\mathcal{R}|}$ is randomly generated by with $S_{ij} \sim U([0,1])$, zero-diagonal means we replace the diagonal entries with zero, and $\epsilon < 1/\lambda_{\max}\big($zero-diagonal$(SS^\top)\big)$.

Algorithm. We run the algorithm with an initial price p_0 set to be $(p_l + p_u)/2$. Here the algorithm converges in two iterations for reaching tolerance 10^{-12}. We compare our algorithm with the standard Interior-Point Method in MATLAB[7]. All the methods produce a feasible solution for solving (Onp-x-MC). This result is to verify that the proposed TR-SQP is able to produce correct solution.

[6] On a MacBook Pro with M2 chip with 16GB memory.

[7] It is a solver that uses sparsity and is compatible with the problem's constraints. See `https://mathworks.com/help/optim/ug/choosing-the-algorithm.html`

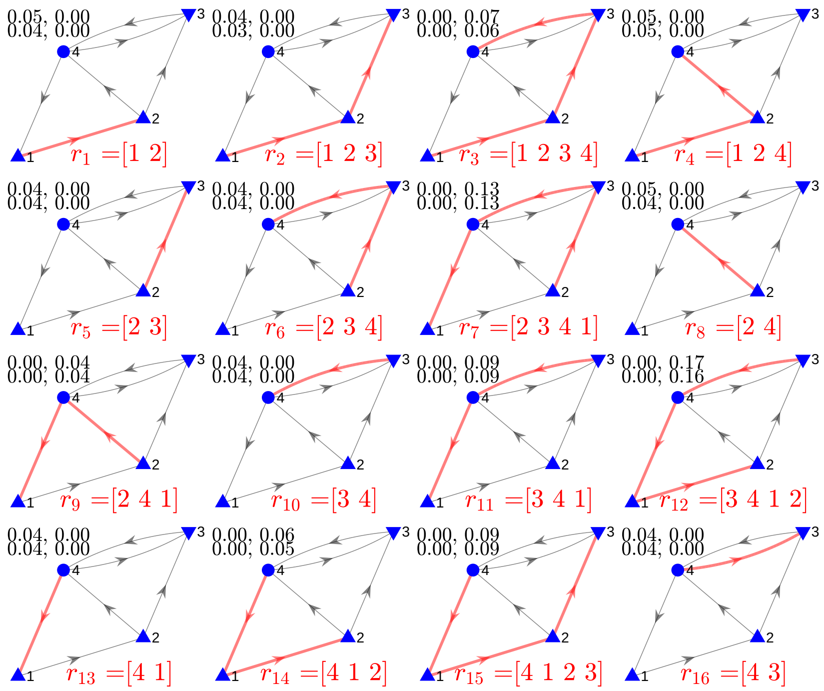


Figure 3: A graph. The top left corner shows the (x, p) value obtained by IPM and TR-SQP of the route highlighted in red.

6.2 Speedup via exploiting sparsity

We conduct experiments to evaluate the impact of sparsity on the algorithm speedup gained by exploiting sparsity. We fix $|\mathcal{K}| = 2$ and take $|\mathcal{R}| \in \{50, 75, \ldots, 225\}$ and let $|\mathcal{E}| = |\mathcal{R}|/5$. We repeat the experiment 10 times for each setup and plot the median over the 10 random problem data. Fig. 4 shows the median computational time taken for the algorithm to converge, with error bar ($\pm$ one standard deviation). We observed a similar pattern for the case $|\mathcal{K}| = 8$.

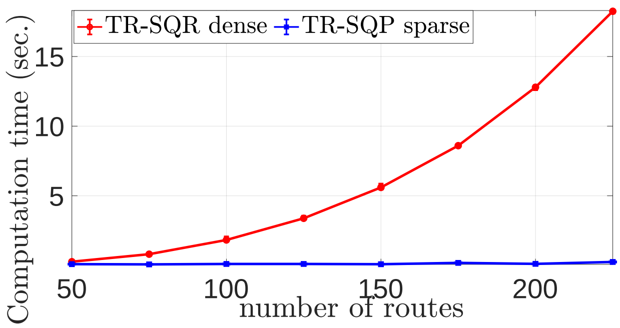


Figure 4: Experimental result on varying $|\mathcal{R}|$: the median curve over 50 random runs with error bar ($\pm$1 std). The result shows that exploiting the sparsity greatly reduces the computational time, with a speedup factor between 5x to 100x.

Real-world dataset. We use the data from *Transportation Networks for Research* [21]. In particular, we use the transportation network dataset from Sioux Falls, South Dakota, US, which consists of 3298 routes, see Fig.5. In an experiment with $|\mathcal{K}| = 2$, TR-SQP with sparsity takes 35 seconds to converge, while TR-SQP without exploiting sparsity takes 172 seconds to converge to the same objective function value. If we allow TR-SQP with sparsity to run 172 seconds, then it will run 5x more iterations than TR-SQP without exploiting sparsity.

7 Conclusion and future direction

In this work we propose a new model on the optimal network pricing (Onp) with the assumption of oblivious users, in which the users remain oblivious to real-time traffic conditions and others' behavior. Inspired by works

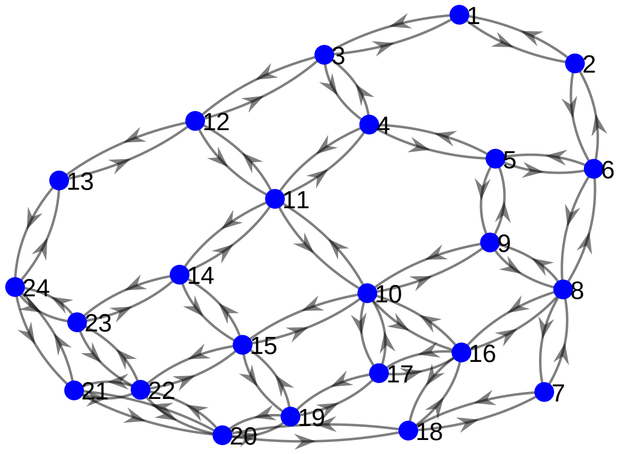


Figure 5: Transportation network of Sioux Falls, South Dakota, US. The graph has $|\mathcal{V}| = 24$, $|\mathcal{E}| = 76$ and $|\mathcal{R}| = 3298$.

on transportation research and network pricing for selfish traffic, we mathematically derive and prove a new formulation of Onp with decision-dependent modeling that relaxes certain existing modeling constraints in the literature. Then, we express the Onp formulation as a constrained nonconvex stochastic quadratic program with uncertainty, and we propose an efficient algorithm to solve the problem, utilizing graph theory, sparse linear algebra, and stochastic approximation. We showcase the effectiveness of the proposed algorithm and the usefulness of the new Onp formulation. We list some possible future works below.

- Relaxing the almost surely argument. In the derivation of the theory, we rely heavy on the statistical argument that the event $[y_p^{(i)}]_j \in \{0, [x_u]_j\}$ in (13) for $\zeta \sim \mathcal{N}(\mu, \Sigma)$ has a probability zero. Practically this is not the case as a computer has finite precision. For example, if the numerical precision of the computer is up to 16 decimal places, then the probability of the event $[y_p^{(i)}]_j \in \{0, [x_u]_j\}$ is now 10^{-16}, which is not zero. Relaxing this argument, such as using Majorization-Minimization, will be a meaningful future work. Resampling MC and subgradient for the gradient computation on the boundary can also be explored.

- Design algorithms dealing with multivariate CDFs and high-dimensional integrals in (Onp-x) directly without MC, and explore a more accurate MC-based methodology for the approximation of the objective function.

Acknowledgments The authors acknowledge the financial support received from the Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Acceleration Fellowship on Citizen-Centric AI Systems (EP/V022067/1), and the Future Electric Vehicle Energy Networks supporting Renewables (FEVER) Grant (EP/W005883/1).

References

[1] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[2] Patrick Briest, Martin Hoefer, and Piotr Krysta. Stackelberg network pricing games. *Algorithmica*, 62(3):733–753, 2012.

[3] Yi-Chang Chiu, Jon Bottom, Michael Mahut, Alexander Paz, Ramachandran Balakrishna, Steven Waller, and Jim Hicks. Dynamic traffic assignment: A primer (transportation research circular e-c153). 2011.

[4] Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

[5] Dmitriy Drusvyatskiy and Lin Xiao. Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research*, 48(2):954–998, 2023.

[6] Matthias Falkner, Michael Devetsikiotis, and Ioannis Lambadaris. An overview of pricing concepts for broadband ip networks. *IEEE Communications Surveys & Tutorials*, 3(2):2–13, 2000.

[7] Ara Hayrapetyan, Eva Tardos, and Tom Wexler. A network pricing game for selfish traffic. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 284–291, 2005.

[8] George Karakostas, Taeyon Kim, Anastasios Viglas, and Hao Xia. On the degradation of performance for traffic networks with oblivious users. *Transportation Research Part B: Methodological*, 45(2):364–371, 2011.

[9] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pages 404–413. Springer, 1999.

[10] Qiang Li and Hoi-To Wai. Stochastic optimization schemes for performative prediction with nonconvex loss. *arXiv preprint arXiv:2405.17922*, 2024.

[11] Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33:4929–4939, 2020.

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

[13] Adhyyan Narang, Evan Faulkner, Dmitriy Drusvyatskiy, Maryam Fazel, and Lillian J Ratliff. Multiplayer performative prediction: Learning in decision-dependent games. *Journal of Machine Learning Research*, 24(202):1–56, 2023.

[14] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[15] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020.

[16] Sébastien Roch, Gilles Savard, and Patrice Marcotte. An approximation algorithm for stackelberg network pricing. *Networks: An International Journal*, 46(1):57–67, 2005.

[17] R Tyrrell Rockafellar and Roger JB Wets. *Variational analysis*. Springer, 1998.

[18] Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[19] Srinivas Shakkottai and Rayadurgam Srikant. Economics of network pricing with multiple isps. *IEEE/ACM Transactions On Networking*, 14(6):1233–1245, 2006.

[20] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.

[21] Transportation Networks for Research Core Team. Transportation networks for research. `https://github.com/bstabler/TransportationNetworks`. Accessed: 2025-10-05.

[22] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 227:228, 2013.

[23] William S Vickrey. Congestion theory and transport investment. *The American economic review*, 59(2):251–260, 1969.

[24] John Glen Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362, 1952.