Stackelberg Control in Combinatorial Congestion Games without Differentiating Through Equilibria

Saeed Masiha[1,*], Sepehr Elahi[1,*], Negar Kiyavash[1], Patrick Thiran[1]
[1]EPFL, Switzerland
{mohammadsaeed.masiha, sepehr.elahi, negar.kiyavash, patrick.thiran}@epfl.ch

We study Stackelberg control in combinatorial congestion games, where a leader chooses parameters and a nonatomic population responds by a Wardrop equilibrium over an exponentially large strategy set. Even with smooth primitives, the resulting hyper-objective is typically Lipschitz but nonsmooth, since small parameter changes can switch which combinatorial strategies are active at equilibrium. We propose an oracle-based bilevel method that avoids differentiating through equilibria: an inner, projection-free Frank–Wolfe solver computes approximate equilibria using only gradient information and a linear minimization oracle, while an outer two-point zeroth-order scheme updates the leader using only objective evaluations at these approximate equilibria. For the inner loop, we analyze a subsampled oracle and prove an $O(1/(\kappa_m T))$ rate under a mild optimizer-hit condition, together with stratified sampling schemes that keep κ_m nontrivial in large, imbalanced strategy spaces. For the full method, we establish convergence to generalized Goldstein stationary points of the hyper-objective, with explicit dependence on the equilibrium approximation error. Overall, our results provide a convergence guarantee for Stackelberg control in CCGs while retaining scalability through combinatorial primitives such as shortest paths and decision-diagram-based oracles. Experiments on real-world transportation networks corroborate the theory and highlight the practical gains of our oracle-based approach across both polynomial-time and NP-hard strategy families.

1. Introduction

Modern transportation, communication, and logistics systems are often operated under *Stackelberg control*: a designer adjusts network parameters (tolls, capacities, incentives) to influence how a large population of selfish users routes its demand. In many applications, however, users do not choose smooth flow splits but rather *combinatorial* objects such as paths, trees, or schedules. This leads naturally to *combinatorial congestion games* (CCGs), where the follower's action space can be exponentially large and equilibrium behavior can change abruptly when the leader perturbs the parameters [1–3].

Problem and notation. Let $\theta \in \Theta \subseteq \mathbb{R}^k$ denote the leader-controlled parameter vector. Followers choose strategies S from a family $\mathcal{S} \subseteq 2^{[n]}$ (e.g., s–t paths); we represent each S by its incidence vector $\mathbf{1}_S \in \{0,1\}^n$ and write $y \in [0,1]^n$ for the induced aggregate load, where y_i is the fraction of demand using resource i. The feasible set of loads is the polytope $\mathcal{C} := \mathrm{conv}\{\mathbf{1}_S : S \in \mathcal{S}\}$. For a given θ, the Wardrop equilibrium load $y^\star(\theta)$ is the (unique) solution of a convex potential minimization problem

$$y^\star(\theta) \in \arg\min_{y \in \mathcal{C}} f(\theta, y),$$

[*]Equal contribution.

Preprint. Work in progress.

and the leader's Stackelberg objective evaluates an equilibrium through an upper-level loss $F(\theta, y)$. The resulting *reduced* objective is

$$\Phi(\theta) := F\big(\theta, y^\star(\theta)\big),$$

and the goal is to minimize Φ over Θ.

Φ is typically nonsmooth. Even when the primitives $(\theta, y) \mapsto f(\theta, y)$ and $(\theta, y) \mapsto F(\theta, y)$ are smooth, the reduced objective Φ need not be. The key difficulty is that $\mathcal{C}$ is a polytope: as θ varies, the optimizer $y^\star(\theta)$ moves along faces, and small parameter changes can alter which constraints are active, equivalently, which combinatorial strategies are used with positive mass. These active-set switches create *kinks* in the equilibrium map and hence in Φ (see Example 2.7). As a consequence, differentiating through equilibria is delicate in CCGs, and the leader's problem cannot be treated as a standard smooth bilevel program.

Prior work: differentiable equilibrium computation with decision diagrams. A complementary line of work makes the *equilibrium computation* differentiable and then applies gradient-based bilevel optimization by differentiating through the lower-level solver. For Stackelberg CCGs, Sakaue and Nakamura [4] use zero-suppressed binary decision diagrams (ZDDs) to compactly represent the exponentially large strategy set and implement a differentiable Frank–Wolfe-style equilibrium routine via a smooth (softmin) relaxation of the linear minimization oracle. This yields a practical hyper-gradient pipeline for optimizing the finite-iteration surrogate objective $F(\theta, y_T(\theta))$ obtained from the inner solver. However, since the true hyper-objective $\Phi(\theta) = F(\theta, y^\star(\theta))$ can be nonsmooth due to active-set changes, establishing end-to-end guarantees for Φ requires controlling both equilibrium approximation error and surrogate-to-true mismatch, as also noted in Sakaue and Nakamura [4]. An expanded discussion of related work and a more detailed comparison appear in Appendix A.

Our approach: zeroth-order optimization of the true (nonsmooth) objective. We take a complementary viewpoint: rather than enforcing differentiability of $y^\star(\theta)$, we treat equilibrium computation as a black box and optimize Φ using only *oracle access*.[2] Concretely, our method has two coupled components. (i) An *inner* Frank–Wolfe routine approximates $y^\star(\theta)$ over $\mathcal{C}$ using only gradients $\nabla_y f(\theta, \cdot)$ and a *linear minimization oracle* (LMO) over $\mathcal{C}$. (ii) An *outer* zeroth-order method updates θ using two-point function evaluations of $\widehat{\Phi}_T(\theta) := F(\theta, y_T(\theta))$, where $y_T(\theta)$ is the inner iterate after T steps. This design never differentiates through $y^\star(\theta)$ and remains applicable when Φ is Lipschitz but nonsmooth.

The inner Frank–Wolfe step reduces to solving a linear problem over $\mathcal{C}$, equivalently over $\mathcal{S}$: given weights $g \in \mathbb{R}^n$, compute $\arg\min_{S \in \mathcal{S}} \sum_{i \in S} g_i$. See, e.g., Jaggi [5] for background on Frank–Wolfe and linear minimization oracles. In routing games this is a shortest-path computation [6]; in other settings it can be implemented by specialized combinatorial optimization routines or decision-diagram methods [4, 7]. When an exact oracle is too expensive, we study a *subsampled* LMO that minimizes over m randomly drawn candidate strategies, following the subsampling-oracle viewpoint of Kerdreux et al. [8].

In our convergence analysis, we make two points precise. First, for subsampled Frank–Wolfe, we define a single optimizer-hit parameter κ_m, the probability that the sampled candidate set contains an exact LMO minimizer, and prove an $O(1/(\kappa_m T))$ convergence rate. This recovers the standard subsampling-oracle Frank–Wolfe guarantee under uniform-inclusion sampling [8, Theorem 2.1], while allowing general (non-uniform, structure-aware) sampling distributions captured by κ_m. Second, we prove that the full bilevel procedure converges to a generalized Goldstein stationary point of the hyper-objective Φ [9], with an explicit dependence on the inner equilibrium approximation error.

Contributions. The main contributions are:

- An oracle-based bilevel algorithm for Stackelberg CCGs that combines a Frank–Wolfe equilibrium solver with a zeroth-order outer loop, and does *not* require differentiating through equilibria.

[2] Here, oracle access means we do not need an explicit formula for the equilibrium map or its derivatives

- A subsampled Frank–Wolfe guarantee with an $O(1/(\kappa_m T))$ rate under a mild optimizer-hit assumption, where κ_m is the probability that the sampled candidate set contains an exact LMO minimizer. We also propose stratified (e.g., length-debiased) sampling schemes that keep κ_m nontrivial in large, imbalanced strategy spaces (Section 5.1).
- End-to-end convergence of the full Stackelberg method to generalized Goldstein stationary points (GGSPs) of the (Lipschitz, possibly nonsmooth) hyper-objective Φ, with explicit dependence on the inner equilibrium approximation error.
- Efficient Python implementations of our zeroth-order method with both exact and subsampled LMOs based on ZDDs, including for st-path, Hamiltonian path, and Steiner cycle strategy families. We also provide experiments on real-world transportation networks that highlight our method's flexibility and performance (Section 6).

Roadmap. Section 2 formalizes the Stackelberg model of CCGs and basic regularity properties. Section 3 presents the bilevel algorithm. Section 4 discusses implementations of the LMO. Section 5 provides the convergence analysis, and Section 6 reports empirical results.

2. Problem Setting

In this section we formalize the Stackelberg CCG model introduced in Section 1, recall the potential formulation of Wardrop equilibria, and state the regularity assumptions used in our analysis.

2.1. Combinatorial Congestion Games (CCGs)

We adopt the notation from Section 1. A nonatomic population of total mass 1 chooses a pure strategy $S \in \mathcal{S} \subseteq 2^{[n]}$ over a finite resource set $[n] := \{1, \dots, n\}$, where each S is represented by its incidence vector $\mathbf{1}_S \in \{0,1\}^n$. Let $d := |\mathcal{S}|$ and let $z = (z_S)_{S \in \mathcal{S}} \in \Delta^d$ denote the induced population distribution over strategies, where

$$\Delta^d := \Big\{ z \in \mathbb{R}^d : z_S \geq 0,\ \sum_{S \in \mathcal{S}} z_S = 1 \Big\}.$$

The corresponding resource-load vector is

$$y(z) := \sum_{S \in \mathcal{S}} z_S\, \mathbf{1}_S \in \mathcal{C},$$

where the feasible set of loads is the polytope $\mathcal{C} := \operatorname{conv}\{\mathbf{1}_S : S \in \mathcal{S}\} \subseteq [0,1]^n$.

Cost functions. Each resource $i \in [n]$ has a congestion-dependent cost function $c_i : [0,1] \to \mathbb{R}$, and a player using strategy S under load $y \in \mathcal{C}$ incurs cost $c_S(y) := \sum_{i \in S} c_i(y_i)$. Throughout this section we make the following assumption which expresses that using a more congested resource is never cheaper.

Assumption 2.1 (Resource costs)**.** *For each $i \in [n]$, the function c_i is continuous and strictly increasing on $[0,1]$.*

2.2. Wardrop equilibrium and potential minimization

In our nonatomic model, each individual player controls an infinitesimal amount of flow and therefore has a negligible impact on congestion. As a result, it is natural to require that an equilibrium only prevents *unilateral* cost improvements for a single player, taking aggregate loads as fixed. This leads to the notion of *Wardrop equilibrium*: in equilibrium, all strategies that are used with positive mass must have minimal cost, and unused strategies cannot be strictly cheaper. Wardrop equilibrium can be viewed as the large-population limit of Nash equilibria in finite congestion games and is the standard solution concept in transportation and routing models [10].

Definition 2.2 (Wardrop equilibrium)**.** *A population profile $z \in \Delta^d$ with induced load $y = y(z) \in \mathcal{C}$ is a* (Wardrop) equilibrium *if the following holds: for every strategy $S \in \mathcal{S}$ with $z_S > 0$,*

$$c_S(y) \;\le\; c_{S'}(y) \qquad \text{for all } S' \in \mathcal{S}.$$

We associate to the game the standard Beckmann-type potential function

$$f : \mathbb{R}^n \to \mathbb{R}, \qquad f(y) := \sum_{i\in[n]} \int_0^{y_i} c_i(u)\, du.$$

Under Assumption 2.1, f is continuously differentiable on $[0,1]^n$ with

$$\nabla f(y)_i = c_i(y_i), \qquad i \in [n],$$

and strictly convex, since each c_i is strictly increasing.

We consider the convex optimization problem

$$\min_{y\in\mathcal{C}} f(y). \tag{1}$$

The next proposition states the classical equivalence between Wardrop equilibria and minimizers of (1), and we provide a proof in Appendix B.1 for completeness.

Proposition 2.3 (Equilibrium $\iff$ potential minimizer)**.** *Let $z \in \Delta^d$ and $y = y(z)$. Under Assumption 2.1, z is a Wardrop equilibrium if and only if $y = \arg\min_{y'\in\mathcal{C}} f(y')$.*

In what follows we will therefore work primarily with the load vector y.

2.3. Bi-level model of CCG

Leader–follower formulation. As in Section 1, the leader controls a parameter vector $\theta \in \Theta \subseteq \mathbb{R}^k$, where Θ is closed and convex, which parametrizes the resource cost functions. For each resource $i \in [n]$, let $c_i(\cdot;\theta) : [0,1] \to \mathbb{R}$ be such that, for every fixed $\theta \in \Theta$, the function $c_i(\cdot;\theta)$ satisfies Assumption 2.1, and for every fixed $y_i \in [0,1]$ the map $\theta \mapsto c_i(y_i;\theta)$ is continuously differentiable. The associated (Beckmann) potential is

$$f(\theta, y) \;:=\; \sum_{i\in[n]} \int_0^{y_i} c_i(u;\theta)\, du, \qquad y \in \mathcal{C}.$$

For each $\theta \in \Theta$, let $y^\star(\theta) := \arg\min_{y\in\mathcal{C}} f(\theta, y)$ denote the (Wardrop) equilibrium load. By Proposition 2.3, $y^\star(\theta)$ coincides with the equilibrium of the congestion game with parameters θ.

The leader evaluates an equilibrium via an upper-level objective $F : \Theta \times \mathcal{C} \to \mathbb{R}$, e.g., the social cost $F(\theta, y) = \sum_{i\in[n]} c_i(y_i;\theta)\, y_i$, and solves the Stackelberg problem

$$\min_{\theta\in\Theta} F(\theta, y) \quad \text{s.t.} \quad y \in \arg\min_{u\in\mathcal{C}} f(\theta, u). \tag{2}$$

Since the lower-level minimizer is unique, (2) reduces to the single-variable problem

$$\min_{\theta\in\Theta} \Phi(\theta) \;:=\; F\big(\theta, y^\star(\theta)\big). \tag{3}$$

Lipschitz stability of the equilibrium map. As discussed in Section 1, Φ can be nonsmooth because $y^\star(\theta)$ may switch between faces of the polytope $\mathcal{C}$. Here we focus on a complementary property: under mild conditions, the equilibrium map $\theta \mapsto y^\star(\theta)$, and hence Φ, is globally Lipschitz. Specifically, we assume a local quadratic growth property around each equilibrium, together with standard Lipschitz regularity of $\nabla_y f$ and of F.

Assumption 2.4 (Local quadratic growth around equilibrium)**.** *There exist constants $\alpha > 0$ and $r > 0$ such that for all $\theta \in \Theta$ and all $y \in \mathcal{C}$ with $\|y - y^\star(\theta)\| \le r$, we have*

$$f(\theta, y) - f(\theta, y^\star(\theta)) \;\ge\; \frac{\alpha}{2}\, \|y - y^\star(\theta)\|^2. \tag{4}$$

Assumption 2.5 (Regularity of the lower- and upper-level problems)**.** *There exists $L_{f,2} \geq 0$ such that for all $\theta, \theta' \in \Theta$ and all $y, y' \in \mathcal{C}$,*

$$\big\|\nabla_y f(\theta, y) - \nabla_y f(\theta', y')\big\| \leq L_{f,2}\big(\|\theta - \theta'\| + \|y - y'\|\big). \tag{5}$$

Moreover, $F : \Theta \times \mathcal{C} \to \mathbb{R}$ is Lipschitz in each argument uniformly in the other: there exist $L_{F,1} \geq 0$ such that for all $\theta, \theta' \in \Theta$ and all $y, y' \in \mathcal{C}$,

$$|F(\theta, y) - F(\theta', y')| \leq L_{F,1}[\|\theta - \theta'\| + \|y - y'\|]. \tag{6}$$

Lemma 2.6 (Lipschitzness of the equilibrium map and hyper-objective)**.** *Let Assumptions 2.4 and 2.5 hold and assume Θ is convex. Then the equilibrium map $\theta \mapsto y^\star(\theta)$ is globally Lipschitz on Θ:*

$$\|y^\star(\theta) - y^\star(\theta')\| \leq \frac{L_{f,2}}{\alpha}\,\|\theta - \theta'\| \qquad \forall \theta, \theta' \in \Theta. \tag{7}$$

Moreover, the hyper-objective $\Phi(\theta) = F(\theta, y^\star(\theta))$ is globally Lipschitz on Θ:

$$|\Phi(\theta) - \Phi(\theta')| \leq \Big(L_{F,1} + \frac{L_{F,1}L_{f,2}}{\alpha}\Big)\|\theta - \theta'\| \quad \forall \theta, \theta' \in \Theta. \tag{8}$$

When does quadratic growth hold? The local quadratic growth condition is mild and is implied by standard second-order regularity at the equilibrium. Indeed, if for each θ the function $f(\theta, \cdot)$ is twice continuously differentiable and the minimizer is *nondegenerate* in the sense that $\nabla^2_{yy} f(\theta, y^\star(\theta)) \succ 0$, then by continuity of the Hessian there exist $\alpha > 0$ and $r > 0$ such that $\nabla^2_{yy} f(\theta, y) \succeq \alpha I$ for all $y \in B(y^\star(\theta), r) \cap \mathcal{C}$. Equivalently, $f(\theta, \cdot)$ is locally strongly convex around $y^\star(\theta)$, which yields the quadratic growth bound (4). Consequently, the local Lipschitz estimates in Lemma 2.6 apply.

Nonsmoothness via active-set changes. We illustrate the kink phenomenon described in Section 1 with a simple two-link example.

Example 2.7 (Kinks from active-set changes)**.** *Consider a parallel network with two resources $i = 1, 2$, strategies $\mathcal{S} = \{\{1\}, \{2\}\}$, and demand 1. Let $c_1(y_1; \theta) = y_1$ and $c_2(y_2; \theta) = \theta$ for $\theta \in \mathbb{R}$. Then*

$$f(\theta, y) = \tfrac{1}{2}y_1^2 + \theta y_2, \qquad y_1 + y_2 = 1, \;\; y_1, y_2 \in [0, 1].$$

Eliminating $y_2 = 1 - y_1$ yields $\min_{y_1 \in [0,1]} \frac{1}{2}y_1^2 + \theta(1 - y_1)$, whose solution is

$$y_1^\star(\theta) = \begin{cases} 0, & \theta \leq 0, \\ \theta, & 0 < \theta < 1, \\ 1, & \theta \geq 1, \end{cases} \qquad y_2^\star(\theta) = 1 - y_1^\star(\theta).$$

Thus $y^\star(\theta)$ is continuous and 1-Lipschitz but not differentiable at $\theta = 0$ and $\theta = 1$, where the active constraints (and the set of used resources) change.

This example motivates the zeroth-order outer loop in Section 3.

3. Zeroth-Order Algorithm for Stackelberg Control

We now describe our oracle-based procedure for approximately solving the reduced Stackelberg problem

$$\min_{\theta \in \Theta} \; \Phi(\theta) := F\big(\theta, y^\star(\theta)\big).$$

As discussed in Section 2, Φ is typically Lipschitz but nonsmooth, so we avoid differentiating through the equilibrium map. Instead, the algorithm couples: (i) an *inner* routine that computes an approximate equilibrium $y_T(\theta)$ for a given θ, and (ii) an *outer* zeroth-order method that updates θ using only value queries of Φ (approximated via y_T).

Algorithm 1 FW-EQUILIBRIUM(θ, T): Frank–Wolfe equilibrium solver

Input: parameter $\theta \in \Theta$, iterations $T \geq 1$ (and an oracle LMO or LMO_m)
initialize $y_0 \in \mathcal{C}$ (e.g., $y_0 = 1_{S_0}$ for some $S_0 \in \mathcal{S}$)
for $t = 0, 1, \ldots, T-1$ **do**
  compute $g_t := \nabla_y f(\theta, y_t)$
  oracle call $s_t := \mathrm{LMO}(g_t)$ (or $s_t := \mathrm{LMO}_m(g_t)$)
  choose step size $\gamma_t \in (0, 1]$ (e.g., exact line search: $\gamma_t \in \arg\min_{\gamma \in [0,1]} f(\theta, (1-\gamma)y_t + \gamma s_t)$)
  update $y_{t+1} := (1-\gamma_t)y_t + \gamma_t s_t$
end for
return $y_T(\theta) := y_T$

3.1. Inner loop: equilibrium approximation via Frank–Wolfe

The inner subproblem is the convex program, i.e., $\min_{y \in \mathcal{C}} f(\theta, y)$. To approximately solve it, we use the Frank–Wolfe method (a.k.a. conditional gradient), summarized in Algorithm 1. Each FW iteration requires only a gradient evaluation $\nabla_y f(\theta, y_t)$ and a call to a linear minimization oracle over $\mathcal{C} = \mathrm{conv}\{1_S : S \in \mathcal{S}\}$:

$$\mathrm{LMO}(g) \in \arg\min_{y \in \mathcal{C}} \langle g, y \rangle = \arg\min_{S \in \mathcal{S}} \sum_{i \in S} g_i.$$

Depending on the setting, Algorithm 1 uses either an exact oracle LMO or a subsampled variant LMO_m (see Section 4.3); the inner solver treats this choice as a plug-in.

In the inner loop, we use an exact or approximate line search along the FW segment $y_t + \gamma(s_t - y_t)$.

The output $y_T(\theta)$ is an approximate equilibrium and is used to form the approximate hyper-objective value

$$\widehat{\Phi}_T(\theta) := F\big(\theta, y_T(\theta)\big) \approx \Phi(\theta).$$

Section 4 discusses implementations of LMO and LMO_m for different strategy families, and Section 5 quantifies the accuracy of Algorithm 1 as a function of T.

3.2. Outer loop: zeroth-order updates on Θ

The outer loop minimizes the (possibly nonsmooth) hyper-objective Φ over the closed convex set Θ using a randomized two-point finite-difference estimator. Specifically, Algorithm 2 maintains iterates $\theta_t \in \Theta$ and, at each iteration, samples a random direction u_t uniformly from the unit sphere in $\mathbb{R}^k$. Fix a smoothing radius $\rho > 0$. At iteration t, we then estimate a descent direction from two function evaluations:

$$\widehat{g}_t := \frac{k}{2\rho}\Big(\widehat{\Phi}_T(\theta_t + \rho u_t) - \widehat{\Phi}_T(\theta_t - \rho u_t)\Big) u_t,$$

where $\widehat{\Phi}_T(\theta) := F(\theta, y_T(\theta))$. We then update θ_t by a projected step on Θ:

$$\theta_{t+1} := \Pi_\Theta\big(\theta_t - \eta\, \widehat{g}_t\big).$$

Each outer iteration makes two calls to the equilibrium solver (at $\theta_t \pm \rho u_t$), and each inner iteration makes one LMO call (exact or subsampled). Implementation details for the LMO are deferred to Section 4, while Section 5 establishes convergence guarantees for the full bilevel procedure.

4. Implementing the Linear Minimization Oracle

The efficiency of our procedure described in Section 3 hinges critically on the cost of each Frank–Wolfe iteration, which is dominated by the linear minimization oracle (LMO) over $\mathcal{C} = \mathrm{conv}\{\mathbf{1}_S : S \in \mathcal{S}\}$. Recall that for any weight vector $g \in \mathbb{R}^n$, $\mathrm{LMO}(g) \in \arg\min_{S \in \mathcal{S}} \sum_{i \in S} g_i$, so an LMO call is exactly the problem of finding a minimum-cost feasible strategy when each resource $i \in [n]$ has weight g_i.

Algorithm 2 ZO-STACKELBERG: zeroth-order Stackelberg optimization

Input: initial $\theta_0 \in \Theta$, outer iterations K, inner iterations T, smoothing $\rho > 0$, stepsize $\eta > 0$
for $t = 0, 1, \ldots, K-1$ **do**
  sample u_t uniformly from the unit sphere in $\mathbb{R}^k$
  $y^+ :=$ FW-EQUILIBRIUM$(\theta_t + \rho u_t, T)$, $\quad y^- :=$ FW-EQUILIBRIUM$(\theta_t - \rho u_t, T)$
  $\widehat{\Phi}^+ := F(\theta_t + \rho u_t, y^+)$, $\quad \widehat{\Phi}^- := F(\theta_t - \rho u_t, y^-)$
  $\widehat{g}_t := \frac{k}{2\rho}\left(\widehat{\Phi}^+ - \widehat{\Phi}^-\right) u_t$
  $\theta_{t+1} := \Pi_\Theta\big(\theta_t - \eta\, \widehat{g}_t\big)$
end for
return θ_K (for the theory in Section 5, we output a random iterate)

In our setting, the weights g arise as gradients of the potential. In particular, at FW iterate y_t and parameter θ, $g_t := \nabla_y f(\theta, y_t)$, so $[g_t]_i = c_i(y_{t,i}; \theta)$, for each resource $i \in [n]$ (i.e., the LMO selects a strategy minimizing the sum of current resource costs).

Many combinatorial congestion games are naturally defined on a graph, where resources correspond to edges (or nodes), and feasible strategies are discrete structures such as s–t paths, Steiner trees, or Hamiltonian cycles [2, 11, 12]. In this language, the LMO solves a (weighted) combinatorial optimization problem on the underlying graph: in some cases it is polynomial-time solvable (e.g., shortest paths), while in others it is NP-hard (e.g., Steiner tree, Hamiltonian cycle). Next we describe three LMO implementations: polynomial-time LMOs for tractable strategy families, exact LMOs for NP-hard families, and approximate LMOs via subsampling when exact methods are impractical.

4.1. Polynomial-time LMO for tractable strategy families

While the strategy set $\mathcal{S}$ can be exponentially large, in many CCGs the LMO is solvable in time polynomial-in-n because it coincides with a classical combinatorial optimization problem.

A canonical example is s–t routing on a directed graph $G = (V, E)$ with resources as edges. Identifying $[n] \equiv E$, the LMO becomes $\min_{P \in \mathcal{P}_{s\to t}} \sum_{e \in P} g_e$, i.e., a shortest-path computation with edge weights g_e (where $g_e = c_e(y_e; \theta)$ at a FW step). This can be solved by any appropriate shortest-path routine (e.g., Dijkstra's algorithm when weights are nonnegative) in time polynomial in $|V| + |E|$ [6]. Thus, each FW iteration costs essentially one polynomial-in-n oracle call, and the inner loop can be run without ever enumerating $\mathcal{S}$.

Other tractable families can be handled analogously whenever the minimum-weight feasible structure under weights g admits a polynomial-time algorithm.

4.2. Exact LMO for NP-hard families via ZDD representations

For NP-hard strategy families, we implement an exact LMO by representing the strategy set $\mathcal{S}$ using a *zero-suppressed binary decision diagram* (ZDD) from [7]. A ZDD is a directed acyclic graph (DAG) whose root-to-$\top$ paths encode feasible strategies: each internal node is labeled by a resource $i \in [n]$ and has two outgoing edges, the *lo*-arc (exclude i) and the *hi*-arc (include i). We give a brief primer and notation in Appendix D. Crucially, the ZDD can represent $\mathcal{S}$ compactly: by sharing equivalent subgraphs, it often yields a diagram whose number of nodes is far smaller than $|\mathcal{S}|$ in practice [7]. This is exactly why we use ZDDs here: it lets us solve the LMO *exactly* by running dynamic programming on the ZDD DAG, without enumerating or even storing the exponentially many strategies in $\mathcal{S}$.

We build the ZDD once for the given combinatorial constraint and reuse it across all zeroth-order and FW iterations. Given weights $g \in \mathbb{R}^n$, the LMO $\min_{S \in \mathcal{S}} \sum_{i \in S} g_i$ can be solved by a shortest-path computation on the ZDD DAG: assign cost g_i to each *hi*-arc leaving a node labeled i (and cost 0 to *lo*-arcs), and compute a minimum-cost root-to-$\top$ path. This is correct because each root-to-$\top$ path corresponds to a unique strategy $S \in \mathcal{S}$, the path cost equals $\sum_{i \in S} g_i$, and therefore a minimum-cost

path realizes the desired $\arg\min$. The resulting strategy is the set of labels whose hi-arcs are taken along the minimum-cost path, yielding an exact minimizer in time polynomial in the ZDD size.

Constructing the ZDD (and the ZDD size itself) can be exponential in n in the worst case. However, as we demonstrate in Section 6, for several NP-hard families the ZDD is often practically manageable, so we pay a one-time preprocessing cost and then solve each LMO call by a fast shortest-path on a DAG that is vastly smaller than $|\mathcal{S}|$.

Conceptually, while our use of ZDDs is similar to the equilibrium computations in Sakaue and Nakamura [4], our role for them is different. We use ZDDs purely as a *combinatorial oracle* for linear minimization inside Frank–Wolfe; the inner loop remains the standard (non-smoothed) FW method, and we never require its iterations to be differentiable with respect to θ. This stands in contrast to differentiable approaches that modify the FW steps (e.g., via softmin relaxations) to enable backpropagation through the equilibrium map and rely on gradient-based updates for the leader's problem. Our algorithm instead keeps the equilibrium computation as a black box and handles the non-smooth dependence on θ entirely at the outer level via zeroth-order methods.

4.3. Approximate LMO via subsampling when ZDDs are impractical

In some instances the ZDD can still be very large, making an exact shortest-path computation on the diagram at *every* FW iteration too expensive. In this regime we use the ZDD only as a *sampler* to obtain a small candidate set of strategies and approximately solve the LMO over this subset.

Concretely, given weights $g \in \mathbb{R}^n$, we draw $m \ll |\mathcal{S}|$ strategies $S^{(1)}, \ldots, S^{(m)} \in \mathcal{S}$ by sampling root-to-$\top$ paths in the ZDD (without enumerating $\mathcal{S}$), and define the sampled candidate set $\mathcal{S}_m := \{S^{(1)}, \ldots, S^{(m)}\}$. We then use the *subsampled LMO*

$$\mathrm{LMO}_m(g) \in \operatorname*{arg\,min}_{S \in \mathcal{S}_m} \sum_{i \in S} g_i,$$

in place of the exact oracle. Sampling m strategies by traversing m root-to-$\top$ paths takes $\mathcal{O}(mn)$ time in our implementation, and the subsequent minimization over $\mathcal{S}_m$ can be done within the same order by accumulating the weights along each sampled strategy.

While LMO_m is not exact, Section 6 shows that it often matches the performance of the exact ZDD-based LMO closely while remaining practical in cases where the ZDD is too large to optimize over directly at every FW step.

5. Theoretical Analysis

In this section, we analyze Algorithm 2: the inner loop approximates the equilibrium $y^\star(\theta) = \arg\min_{y \in \mathcal{C}} f(\theta, y)$, while the outer loop performs zeroth-order updates on the hyper-objective $\Phi(\theta) := F(\theta, y^\star(\theta))$. We provide guarantees both for an exact LMO and for the subsampled LMO from Section 4.3.

Takeaways. The inner result (Theorem 5.2) shows that subsampling slows Frank–Wolfe only through the factor κ_m, yielding an $O(1/(\kappa_m T))$ rate for equilibrium approximation. The outer result (Theorem 5.3) establishes convergence of the full bilevel method to a generalized Goldstein stationary point of the (generally nonsmooth) reduced objective Φ, with an explicit dependence on the inner error ε_y.

5.1. Inner loop: equilibrium approximation via (subsampled) Frank–Wolfe

The inner loop (Algorithm 1) computes an approximate Wardrop equilibrium $y^\star(\theta) \in \arg\min_{y \in \mathcal{C}} f(\theta, y)$ using Frank–Wolfe (FW). Each FW step requires solving a linear problem over the polytope $\mathcal{C} = \mathrm{conv}\{\mathbf{1}_S : S \in \mathcal{S}\}$. Let $D := \max\{\|u - v\| : u, v \in \mathcal{C}\}$.

Exact and subsampled linear minimization. Given a vector $g \in \mathbb{R}^n$, the exact FW direction is obtained from the LMO

$$\mathrm{LMO}(g) \in \arg\min_{y \in \mathcal{C}} \langle g, y \rangle = \arg\min_{S \in \mathcal{S}} \sum_{i \in S} g_i.$$

When $\mathcal{S}$ is too large to optimize over directly, we use the subsampled oracle from Section 4.3: draw m i.i.d. strategies $S^{(1)}, \ldots, S^{(m)} \sim q$ and return the best one,

$$\mathrm{LMO}_m(g) \in \arg\min_{S \in \{S^{(1)}, \ldots, S^{(m)}\}} \sum_{i \in S} g_i.$$

The resulting inner algorithm (Algorithm 1) is identical to standard FW, except that LMO is replaced by LMO_m.

A single quantity that controls the effect of subsampling. For any $g \in \mathbb{R}^n$, let $\mathrm{Opt}(g) := \arg\min_{S \in \mathcal{S}} \sum_{i \in S} g_i$ denote the set of exact minimizers, and define

$$p(g) := q(\mathrm{Opt}(g)) = \Pr_{S \sim q}\big[S \in \mathrm{Opt}(g)\big],$$

i.e., the probability that a single draw from q is an exact minimizer of the linear objective with weights g. If we draw m candidates independently, then the probability that *at least one* of them is an exact minimizer is

$$\kappa_m(g) \;:=\; 1 - (1 - p(g))^m.$$

On the event that an optimizer is present in the sampled set, $\mathrm{LMO}_m(g)$ returns an exact LMO solution; otherwise it returns a (potentially much worse) approximate direction. For a clean FW rate, we assume that this optimizer-hit probability $p(g)$ is uniformly nontrivial along the FW run:

Assumption 5.1 (Uniform optimizer mass)**.** *There exists $\underline{p} \in (0, 1]$ such that for all FW gradients g encountered by the inner loop,*

$$p(g) = q(\mathrm{Opt}(g)) \;\geq\; \underline{p}.$$

Under Assumption 5.1, we can work with the uniform bound

$$\kappa_m \;:=\; 1 - (1 - \underline{p})^m,$$

which is the only place where the sampling distribution q enters the FW convergence rate.

Comparison to subsampling-oracle Frank–Wolfe. Kerdreux et al. [8] analyze randomized Frank–Wolfe where the LMO is solved over a random subset of atoms such that every atom is included with the same probability η, leading to an $O(1/(\eta T))$ rate. Our Assumption 5.1 is less prescriptive: it only requires a uniform lower bound on the probability of sampling an *exact* LMO minimizer (captured by κ_m), allowing non-uniform, structure-aware distributions q; the resulting rate has the same form with η replaced by κ_m.

Theorem 5.2 (Convergence of FW-Equilibrium with subsampled LMO)**.** *Let Assumptions 2.1, 2.5 and 5.1 hold. Fix $\theta \in \Theta$ and initialize $y_0 \in \mathcal{C}$. Run Algorithm 1 with the subsampled oracle LMO_m and exact line-search. Then for all $T \geq T_0$,*

$$\mathbb{E}[f(\theta, y_T)] - f(\theta, y^\star(\theta)) \;\leq\; \frac{2L_{f,2}D^2}{\kappa_m\,(T - T_0 + 1)}, \tag{9}$$

where

$$T_0 := \left\lceil \frac{1}{\kappa_m} \log\!\Big(1 + \frac{2[f(\theta, y_0) - f(\theta, y^\star(\theta))]}{L_{f,2}D^2}\Big) \right\rceil. \tag{10}$$

(*b*) *If Assumption 2.4 holds, then for every $\varepsilon \in (0, r]$, $\sup_{\theta \in \Theta} \mathbb{E}\|y_T(\theta) - y^\star(\theta)\| \leq \varepsilon$ provided*

$$T \;\geq\; \left\lceil \frac{2L_{f,2}D^2}{\alpha\,\kappa_m\,\varepsilon^2} - 2 \right\rceil + T_0.$$

The proof is given in Appendix C.2. Up to the logarithmic burn-in T_0, the bound (9) matches the standard $O(1/T)$ Frank–Wolfe rate, with the entire effect of subsampling captured by the single constant κ_m. Thus, designing q to make κ_m nontrivial is the key to making subsampled LMO calls effective; we discuss this next.

Why q matters: uniform sampling makes κ_m tiny. If q is uniform over $\mathcal{S}$, then $p(g) = |\mathrm{Opt}(g)|/|\mathcal{S}|$. When $|\mathcal{S}|$ is huge (often exponential), $p(g)$ is essentially zero unless the optimizer set is enormous, and for $m \ll |\mathcal{S}|$ we have

$$\kappa_m(g) = 1 - \left(1 - \tfrac{|\mathrm{Opt}(g)|}{|\mathcal{S}|}\right)^m \approx \frac{m\,|\mathrm{Opt}(g)|}{|\mathcal{S}|}.$$

Thus the FW rate in Theorem 5.2 becomes vacuous unless m is extremely large.

Our idea to improve $p(g)$: stratified sampling. Uniformly sampling from $\mathcal{S}$ is often ineffective because $\mathcal{S}$ is highly *imbalanced*: most strategies belong to a few large "combinatorial classes" (for instance, strategies with many selected resources, i.e., large cardinality $|S|$). When $|\mathcal{S}|$ is huge, uniform sampling therefore draws almost exclusively from these high-cardinality strategies, even if LMO minimizers typically lie in a much smaller structured subset[3].

To correct this imbalance, we partition $\mathcal{S}$ into strata using a statistic $\tau : \mathcal{S} \to \mathcal{T}$ (e.g., $\tau(S) = |S|$), and sample in two stages: (i) pick a stratum $t \in \mathcal{T}$ according to a user-chosen distribution $w = (w_t)_{t\in\mathcal{T}}$ (with $w_t \geq 0$ and $\sum_t w_t = 1$), and (ii) sample S uniformly from $\mathcal{S}_t := \{S \in \mathcal{S} : \tau(S) = t\}$. This induces the distribution

$$q(S) = \frac{w_{\tau(S)}}{|\mathcal{S}_{\tau(S)}|}.$$

The key point is that the probability of sampling an exact LMO minimizer,

$$p(g) = q(\mathrm{Opt}(g)) = \sum_{t\in\mathcal{T}} w_t \cdot \frac{|\mathrm{Opt}(g) \cap \mathcal{S}_t|}{|\mathcal{S}_t|},$$

depends only on *within-stratum* optimizer fractions $|\mathrm{Opt}(g) \cap \mathcal{S}_t|/|\mathcal{S}_t|$ and on the chosen weights w_t, rather than on the global ratio $|\mathrm{Opt}(g)|/|\mathcal{S}|$. As a result, if minimizers are concentrated in a few strata where these within-stratum fractions are non-negligible and we assign those strata nontrivial weights, then $p(g)$ can be bounded away from zero even when $|\mathcal{S}|$ is enormous.

Two length-debiased choices (used in Section 4.3). A particularly natural stratification is by *strategy length* $\tau(S) = |S|$ (cardinality), since the LMO objective $\sum_{i\in S} g_i$ is additive and $g_i \geq 0$ in our setting. In many instances, this makes LMO minimizers favor low-cardinality strategies, whereas the number of feasible strategies typically grows rapidly with $|S|$; uniform sampling is then dominated by large $|S|$ and rarely draws short candidates.

Let $\mathcal{L} := \{|S| : S \in \mathcal{S}\}$ and $N_x := |\{S \in \mathcal{S} : |S| = x\}| = |\mathcal{S}_x|$. Any choice of length-weights $(w_x)_{x\in\mathcal{L}}$ yields $q(S) = w_{|S|}/N_{|S|}$. Two simple choices are:

- **Uniform-over-length.** Set $w_x = \frac{1}{|\mathcal{L}|}$ for all $x \in \mathcal{L}$, i.e., we first pick a length x uniformly from $\mathcal{L}$ and then pick a strategy uniformly from $\mathcal{S}_x$. Equivalently,

  $$q(S) = \frac{1}{|\mathcal{L}|\,N_{|S|}}.$$

  In particular, every length class $\mathcal{S}_x$ receives the same total sampling probability $1/|\mathcal{L}|$, so we do not oversample long strategies simply because there are many more of them.

- **Harmonic length bias.** Set $w_x = \frac{\frac{1}{x}}{\sum_{\ell\in\mathcal{L}} \frac{1}{\ell}}$. Then shorter lengths are sampled more frequently, while still sampling uniformly within each length class:

  $$q(S) = \frac{w_{|S|}}{N_{|S|}} = \frac{1}{\sum_{\ell\in\mathcal{L}} \frac{1}{\ell}} \cdot \frac{1}{|S|\,N_{|S|}} \;\propto\; \frac{1}{|S|\,N_{|S|}}.$$

  Compared to uniform-over-length, this further increases the probability of drawing short strategies, which empirically contain LMO minimizers more often, thereby increasing $p(g)$ and improving κ_m for a given m.

[3]Since the LMO minimizes an additive cost $\sum_{i\in S} g_i$ with nonnegative weights $g_i \geq 0$, adding extra resources can only increase the objective, so minimizers often favor low-cardinality strategies unless additional resources provide access to much cheaper costs.

5.2. Outer loop: Goldstein stationarity and convergence of Algorithm 2

The reduced Stackelberg objective $\Phi(\theta) = F(\theta, y^\star(\theta))$ is typically Lipschitz but nonsmooth because the equilibrium map can switch combinatorially with θ. Accordingly, we use Goldstein stationarity (based on Clarke subgradients) and the projected gradient mapping as our notion of approximate optimality, and we quantify how the outer-loop accuracy depends on the inner-loop equilibrium error. This yields a convergence guarantee for the hyper-objective in a weak first-order sense even though we never differentiate through $y^\star(\theta)$.

Stationarity measure. Let $f : \mathbb{R}^k \to \mathbb{R}$ be locally Lipschitz and $\Theta \subseteq \mathbb{R}^k$ be closed and convex. For $\rho > 0$, define the Goldstein ρ-subdifferential

$$\partial_\rho f(\theta) := \operatorname{conv}\Big(\bigcup_{\|v-\theta\| \le \rho} \partial f(v) \Big),$$

where $\partial f(\cdot)$ is the Clarke subdifferential [13]; this notion of (Goldstein) approximate stationarity dates back to Goldstein [9]. For a stepsize $\eta > 0$, define the projected gradient mapping

$$G_\Theta(\theta, g; \eta) := \frac{1}{\eta}\big(\theta - \Pi_\Theta(\theta - \eta g)\big).$$

We say that $\theta \in \Theta$ is an (ϵ, ρ, η)-generalized Goldstein stationary point (GGSP) of f if

$$\min_{g \in \partial_\rho f(\theta)} \|G_\Theta(\theta, g; \eta)\| \le \epsilon.$$

Inexact objective evaluations. The outer loop does not access Φ directly; instead it queries the approximate value $\widehat{\Phi}_T(\theta) := F(\theta, y_T(\theta))$ produced by running the inner solver for T steps. The only inner-loop quantity that enters the outer-loop bound is the uniform equilibrium error (in expectation),

$$\varepsilon_y := \sup_{\theta \in \Theta} \mathbb{E}\|y_T(\theta) - y^\star(\theta)\|.$$

When $F(\theta, \cdot)$ is $L_{F,y}$-Lipschitz, this implies an $O(L_{F,y}\varepsilon_y)$ error on function values, and hence a controlled bias in the two-point estimator used by Algorithm 2.

Theorem 5.3 (Convergence of Algorithm 2 to a GGSP of Φ (with subsampled FW))**.** *Let Assumptions 2.1, 2.4 and 2.5 hold. Let Φ be L_Φ-Lipschitz on Θ (Lemma 2.6). Fix $\rho > 0$ and run Algorithm 2 for K iterations with constant stepsize $\eta \le c_0\, \rho/(L_\Phi\sqrt{k})$ for a numerical constant $c_0 \in (0,1)$. Output $\widehat{\theta}$ uniformly at random from the iterates $\{\theta_0, \dots, \theta_{K-1}\}$. Then there exist universal constants $C_1, C_2 > 0$ such that*

$$\mathbb{E}\left[\min_{g \in \partial_\rho \Phi(\widehat{\theta})} \|G_\Theta(\widehat{\theta}, g; \eta)\|\right] \le C_1 \sqrt{\frac{\Phi(\theta_0) - \inf_\Theta \Phi + 2\rho L_\Phi}{\eta K}} \;+\; C_2 \frac{k}{\rho}\, L_{F,y}\, \varepsilon_y. \tag{11}$$

In particular, choosing $\varepsilon_y \le \frac{\rho}{2C_2 k L_{F,y}}\,\epsilon$ and $K \ge \frac{4C_1^2}{\eta\,\epsilon^2}\big(\Phi(\theta_0) - \inf_\Theta \Phi + 2\rho L_\Phi\big)$ guarantees that $\widehat{\theta}$ is an (ϵ, ρ, η)-GGSP of Φ in expectation.

Total oracle complexity. Each outer iteration evaluates $\widehat{\Phi}_T$ twice (at $\theta_t \pm \rho u_t$), hence makes two calls to the inner solver. Therefore, #(inner gradients) $= 2KT$, #(subsampled LMO calls) $= 2KT$, and #(sampled strategies) $= 2KTm$. Combining Theorem 5.3 with the inner-loop guarantee $\varepsilon_y \lesssim \sqrt{L_{f,2} D^2/(\alpha \kappa_m T)}$ from Theorem 5.2 (b) yields the sufficient inner-loop length

$$T \;\ge\; \left\lceil \frac{16 C_2^2\, L_{f,2} D^2\, k^2 L_{F,y}^2}{\alpha\, \kappa_m\, \rho^2 \epsilon^2} - 2 \right\rceil + T_0,$$

and thus total inner-gradient complexity $2KT$ and total sampled-strategy complexity $2KTm$.

Table 1: TNTP instances used in the paper. ZDD nodes and $|\mathcal{S}|$ denote, respectively, the size of the ZDD encoding and the number of feasible combinatorial strategies.

Scenario	Problem family	$\lvert V\rvert$	$\lvert E\rvert$	ZDD nodes	$\lvert\mathcal{S}\rvert$
1	s–t paths	49	82	1.36×10^4	3.12×10^7
2	Hamiltonian s–t paths	63	118	2.88×10^4	2.53×10^6
3	Steiner cycles	110	176	1.06×10^6	2.21×10^{14}

6. Experiments

We report experiments on three real-world transportation networks derived from the Transportation Networks for Research Core Team (TNTP) benchmark suite [14]. Our goal is to evaluate the end-to-end performance of ZO-Stackelberg (ZOS) and, in particular, to isolate the computational role of the lower-level linear minimization oracle (LMO) by moving from a setting with a polynomial-time oracle to NP-hard settings where exact oracles are only practical via an exact ZDD-based oracle or via subsampling.

6.1. Experimental setting

Networks and strategy families. We construct three instances from TNTP subgraphs, each inducing a different combinatorial strategy family. Scenario 1 is an s–t path game, where the LMO reduces to a standard shortest-path computation and can be implemented in polynomial time. Scenario 2 uses Hamiltonian s–t paths, an NP-hard strategy family, but in this instance the corresponding ZDD remains small enough that an exact ZDD-based LMO is still fast. Scenario 3 uses Steiner cycles through a fixed terminal set (NP-hard), where the ZDD becomes extremely large and motivates subsampling-based LMOs. Table 1 summarizes the resulting graphs and ZDD statistics.

Cost model and Stackelberg objective. We use the fractional edge-cost family

$$c_i(y_i;\theta_i) = d_i\Big(1 + C\,\frac{y_i}{\theta_i + 1}\Big),$$
$$\Theta := \{\theta \in \mathbb{R}_+^m : \mathbf{1}^\top\theta = m\},$$

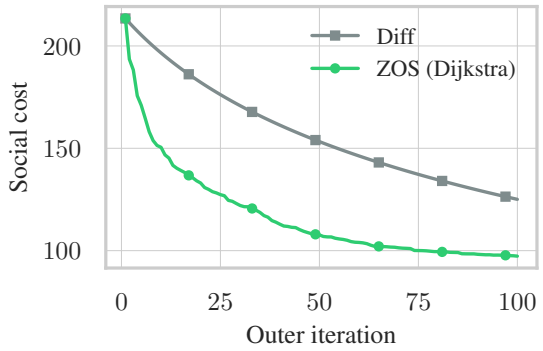
where d_i is the (normalized) free-flow length and C is a network-dependent scaling parameter (Scenario 1: C=500, Scenario 2: C=20, Scenario 3: C=50). The leader objective is the social cost $F(\theta, y) = \sum_i c_i(y_i;\theta_i)\, y_i$. All methods are run for 100 outer iterations, and each outer objective evaluation solves the lower-level equilibrium via T=1000 Frank–Wolfe steps.

6.2. Algorithms

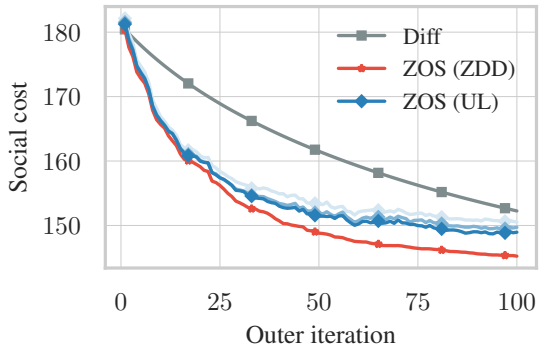
We compare our oracle-based ZO-Stackelberg (ZOS) procedure to the differentiable baseline of Sakaue and Nakamura [4].

Baseline (Diff). The method of Sakaue and Nakamura [4] replaces the lower-level equilibrium computation with a differentiable softmin oracle over the ZDD and applies gradient-based optimization to a smoothed surrogate objective.

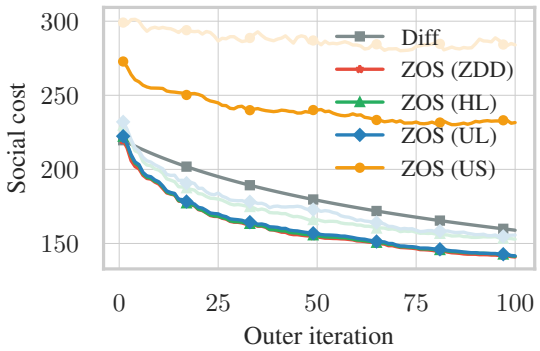
ZO-Stackelberg (ZOS; ours). We run ZO-Stackelberg from Algorithm 2, coupled with the Frank–Wolfe inner loop from Algorithm 1. The only setting-dependent component is the LMO used by the inner loop: (i) exact shortest-path (Dijkstra) for Scenario 1, (ii) exact ZDD argmin for Scenario 2, and (iii) a subsampling LMO for Scenario 3 that draws m feasible strategies and returns the best sample. We report three subsampling rules: uniform-strategy (US), uniform-over-length (UL), and harmonic-length (HL).



(a) Scenario 1: s–t paths (no subsampling).



(b) Scenario 2: Hamiltonian s–t paths. Shades (light→dark) correspond to subsampling budgets $m \in \{10, 50, 100\}$.



(c) Scenario 3: Steiner cycles. Shades (light→dark) correspond to subsampling budgets $m \in \{10, 100\}$.

Figure 1: Social cost along outer iterations on the TNTP scenarios. We compare Diff to ZO-Stackelberg (ZOS) coupled with different LMOs in the lower-level Frank–Wolfe loop. In the subsampling LMO, a batch of m feasible strategies is sampled at each FW step and the best sample is returned; we report three sampling rules: uniform-strategy (US), i.e., uniform over feasible strategies; uniform-length (UL), i.e., uniform over distinct feasible lengths; and harmonic-length (HL), i.e., sampling lengths proportionally to the inverse of the length. When multiple subsampling budgets are run, we plot one curve per m using the same base color, where darker shades correspond to larger m.

6.3. Results

Social-cost convergence. Figure 1 reports the social cost along the outer iterations. In Scenario 1, the polynomial-time Dijkstra LMO enables ZO-Stackelberg to reduce social cost substantially faster than Diff. In Scenario 2, the exact ZDD LMO remains practical and yields the best final social cost, while subsampling trades oracle cost for slower convergence. In Scenario 3, the ZDD becomes very large; here, subsampling provides a scalable alternative that remains competitive in terms of social-cost reduction.

Runtime. Table 2 reports wall-clock runtime (including ZDD construction when applicable). Scenario 1 highlights the low overhead of a polynomial-time LMO (Dijkstra). Scenario 2 demonstrates that, when the ZDD remains small, an exact ZDD LMO is still practical. In Scenario 3, ZDD-based exact methods become substantially more expensive, and subsampling provides a scalable alternative.

Table 2: Wall-clock runtime in seconds (including ZDD construction when applicable). For subsampled variants, HL/UL/US denote harmonic-length, uniform-length, and uniform-strategy sampling, and m is the number of sampled strategies used by the subsampling LMO. Best (smallest) runtime within each scenario is highlighted in bold.

	Algorithm	Time (s)
Scenario 1	Diff	3301
	ZOS (Dijkstra)	**164**
Scenario 2	Diff	6160
	ZOS (ZDD)	170
	ZOS (UL, $m = 10$)	**58**
	ZOS (UL, $m = 50$)	124
	ZOS (UL, $m = 100$)	209
Scenario 3	Diff	358173
	ZOS (ZDD)	3155
	ZOS (US, $m = 10$)	**310**
	ZOS (HL, $m = 10$)	322
	ZOS (UL, $m = 10$)	323
	ZOS (US, $m = 100$)	548
	ZOS (HL, $m = 100$)	684
	ZOS (UL, $m = 100$)	709

References

[1] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952. doi: 10.1680/ipeds.1952.11259.

[2] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2): 236–259, 2002. doi: 10.1145/506147.506153.

[3] George Karakostas and Stavros G. Kolliopoulos. Stackelberg strategies for selfish routing in general multicommodity networks. *Algorithmica*, 53(1):132–153, 2009. doi: 10.1007/s00453-007-9018-5.

[4] Shinsaku Sakaue and Kengo Nakamura. Differentiable equilibrium computation with decision diagrams for stackelberg models of combinatorial congestion games. In *Neural Information Processing Systems*, 2021.

[5] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435. PMLR, 2013.

[6] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1): 269–271, 1959. doi: 10.1007/BF01386390.

[7] Shin ichi Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E76-A (2):141–147, 1993.

[8] Thomas Kerdreux, Fabian Pedregosa, and Alexandre d'Aspremont. Frank-Wolfe with subsampling oracle. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International*

Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 2571–2580. PMLR, 2018. URL `https://proceedings.mlr.press/v80/kerdreux18a.html`.

[9] Allen A. Goldstein. Optimization of lipschitz continuous functions. *Mathematical Programming*, 13(1):14–22, 1977.

[10] William H. Sandholm. *Population Games and Evolutionary Dynamics*. Economic Learning and Social Evolution. MIT Press, Cambridge, MA, 2010. ISBN 978-0-262-19587-8.

[11] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *45th Annual IEEE Symposium on Foundations of Computer Science* (*FOCS*), pages 295–304, 2004. doi: 10.1109/FOCS.2004.68.

[12] M. R. Garey, D. S. Johnson, and R. Endre Tarjan. The planar hamiltonian circuit problem is np-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi: 10.1137/0205049.

[13] Frank H. Clarke. *Optimization and Nonsmooth Analysis*, volume 5 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1990.

[14] Benjamin Stabler and Transportation Networks for Research Core Team. Transportation networks for research core team (tntp). `https://github.com/bstabler/TransportationNetworks`, 2026. Accessed: 2026-01-15.

[15] Martin J. Beckmann, C. B. McGuire, and Christopher B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956. Published for the Cowles Commission for Research in Economics.

[16] Larry J. LeBlanc, Edward K. Morlok, and William P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975. doi: 10.1016/0041-1647(75)90030-1.

[17] Yosef Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1984. ISBN 0139397299.

[18] M. Patriksson. *Traffic Assignment Problems: Models and Methods*. V.S.P. Intl Science, 1994. ISBN 9789067641814.

[19] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986. doi: 10.1109/TC.1986.1676819.

[20] David Bergman, André A. Cire, Willem-Jan van Hoeve, and John Hooker. *Decision Diagrams for Optimization*. Springer, 2018. ISBN 9783319826790.

[21] Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 136–145. PMLR, 2017. URL `https://proceedings.mlr.press/v70/amos17a.html`.

[22] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Zico Kolter. Differentiable convex optimization layers. *arXiv preprint arXiv:1910.12430*, 2019. URL `https://arxiv.org/abs/1910.12430`.

[23] James V. Burke, Adrian S. Lewis, and Michael L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005. doi: 10.1137/030601296.

[24] Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, 20(1):119–154, 2019. doi: 10.1007/s10208-018-09409-5.

[25] Zhuanghua Liu, Cheng Chen, Luo Luo, and Bryan Kian Hsiang Low. Zeroth-order methods for constrained nonconvex nonsmooth stochastic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30842–30872. PMLR, 2024.

[26] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394. SIAM, 2005.

[27] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[28] John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015. doi: 10.1109/TIT.2015.2409256.

[29] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. doi: 10.1109/9.119632.

[30] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009. ISBN 9780898716689.

[31] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. *arXiv preprint arXiv:1602.02101*, 2016. URL `https://arxiv.org/abs/1602.02101`.

[32] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018. URL `https://arxiv.org/abs/1804.09554`.

[33] J. Frédéric Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2000. ISBN 978-0-387-98705-7.

[34] Stephen M. Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1):43–62, 1980.

[35] J. Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

[36] Stephan Dempe. *Foundations of Bilevel Programming*, volume 61 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Boston, MA, 2002. ISBN 978-0-306-48045-4.

[37] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

A. Related Work

Stackelberg control and congestion games. Congestion games and selfish routing are classical models for large-scale transportation and communication systems, with Wardrop equilibria as the standard notion of flow-level equilibrium [1]. The potential formulation of Wardrop equilibria and its connection to convex optimization dates back to the Beckmann transformation [15] and has since been widely used in algorithmic and economic analyses of routing [2, 10]. Stackelberg control in these models includes leader decisions such as tolling, pricing, or capacity design that influence equilibrium behavior; see, e.g., Karakostas and Kolliopoulos [3] and references therein. For large networks, equilibrium computation is typically performed using methods that exploit shortest-path computations and other combinatorial subroutines. Classic traffic-assignment algorithms include Frank–Wolfe-type schemes specialized to the Beckmann potential [16]. Since Wardrop equilibria

coincide with minimizers of the (convex) Beckmann potential, each Frank–Wolfe step solves the linearization of the potential over the feasible flow polytope; in network models this reduces to an all-or-nothing assignment subproblem (typically implemented via shortest-path computations). This yields a projection-free method whose per-iteration cost is dominated by combinatorial primitives rather than Euclidean projections, which is one reason Frank–Wolfe is widely used in practice. Comprehensive discussions of step-size rules (e.g., exact line search vs. method-of-successive-averages heuristics), convergence, and practical variants can be found in standard references such as Sheffi [17], Patriksson [18].

Compact representations of combinatorial strategy sets. When the strategy family $\mathcal{S}$ is exponentially large, one approach is to represent $\mathcal{S}$ implicitly via compact data structures that share substructure across many feasible strategies. Binary decision diagrams (BDDs) encode a Boolean function as a reduced, ordered decision DAG [19]; in our setting, the function is simply the indicator of feasibility over subsets of $[n]$. Zero-suppressed BDDs (ZDDs) modify the reduction rules to better match *families of sparse sets*, roughly, instances where most sets omit most items, which often leads to dramatically smaller diagrams in practice [7]. Once constructed, the ZDD supports efficient dynamic programming on the DAG to perform common set-family operations (e.g., union/intersection), to count feasible strategies, to sample feasible strategies, and to optimize additive costs over $\mathcal{S}$ without enumerating it [7, 20]. Although ZDD size is problem- and ordering-dependent (and can be exponential in the worst case), decision-diagram techniques have become a practical tool for discrete optimization and for implementing combinatorial oracles over large strategy spaces [20].

Differentiating through equilibria and decision diagrams. A contrasting line of work makes the equilibrium computation itself *differentiable* and then applies gradient-based bilevel optimization by differentiating through the lower-level computation. For Stackelberg CCGs, Sakaue and Nakamura [4] leverage ZDDs to represent $\mathcal{S}$ and develop a differentiable Frank–Wolfe-style equilibrium solver via a softmin relaxation of the LMO, enabling automatic differentiation of the surrogate map $\theta \mapsto y_T(\theta)$ and the corresponding surrogate objective value $F(\theta, y_T(\theta))$. More broadly, differentiable optimization layers enable backpropagation through optimization problems [21, 22]. This yields a practical hyper-gradient pipeline and strong empirical performance, but the optimized objective is a smoothed surrogate rather than the true hyper-objective $\Phi(\theta) = F(\theta, y^\star(\theta))$. Establishing end-to-end guarantees for Φ requires quantifying both the equilibrium approximation error $y_T(\theta) \approx y^\star(\theta)$ and the mismatch between the gradient of surrogate ($\nabla F(\theta, y_T(\theta))$) and suitable (generalized) derivatives of Φ; this is explicitly noted as challenging in Sakaue and Nakamura [4]. In contrast, our method avoids differentiating through equilibria and provides convergence to generalized Goldstein stationary points of the (typically Lipschitz but nonsmooth) hyper-objective, with an explicit dependence on the inner equilibrium error.

Zeroth-order (derivative-free) optimization. Because the hyper-objective $\Phi(\theta) = F(\theta, y^\star(\theta))$ is typically Lipschitz but nonsmooth due to active-set changes, we adopt a weak first-order stationarity notion based on Clarke/Goldstein subgradients [9, 13]. These generalized derivatives and stationarity measures also underpin classic algorithms for nonsmooth nonconvex optimization such as gradient sampling [23], as well as modern convergence analyses for stochastic methods on nonsmooth landscapes [24, 25]. Our outer loop uses randomized two-point finite-difference estimators, a standard tool in derivative-free optimization and bandit/zeroth-order learning. Foundational analyses include Flaxman et al. [26] and later refinements such as Nesterov and Spokoiny [27], Duchi et al. [28]. More classical derivative-free schemes include stochastic approximation methods like SPSA [29] and general-purpose derivative-free trust-region methods surveyed in Conn et al. [30].

Randomized and approximate Frank–Wolfe oracles. To reduce the cost of LMO calls when $\mathcal{S}$ is very large, several works study conditional-gradient methods with randomized or approximate oracles. In particular, Kerdreux et al. [8] analyze a subsampling-oracle Frank–Wolfe method under a uniform-inclusion model, obtaining an $O(1/(\eta T))$ rate. Related variants of stochastic/online Frank–Wolfe have been studied in different settings, including projection-free stochastic optimization [31, 32]. Our analysis isolates a single optimizer-hit quantity κ_m and allows non-uniform, structure-aware sampling distributions tailored to imbalanced strategy spaces.

B. Proofs of Section 2

B.1. Proof of Proposition 2.3

Proof. We first recall the first-order optimality condition for (1). Since f is differentiable and convex, a point $\bar{y} \in \mathcal{C}$ minimizes f over $\mathcal{C}$ if and only if

$$\langle \nabla f(\bar{y}),\, y - \bar{y} \rangle \;\geq\; 0 \qquad \text{for all } y \in \mathcal{C}. \tag{12}$$

(Equilibrium ⇒ minimizer). Let $z \in \Delta^d$ be a Wardrop equilibrium and $y = y(z)$. By definition of f we have $\nabla f(y)_i = c_i(y_i)$, so for any $y' \in \mathcal{C}$,

$$\langle \nabla f(y),\, y' - y \rangle = \sum_{i \in [n]} c_i(y_i)\,(y'_i - y_i).$$

Every $y' \in \mathcal{C}$ can be written as $y' = \sum_{S \in \mathcal{S}} \xi_S\, \mathbf{1}_S$ for some $\xi \in \Delta^d$. Using $y = \sum_S z_S \mathbf{1}_S$ we obtain

$$\begin{aligned}
\langle \nabla f(y),\, y' - y \rangle &= \sum_{i \in [n]} c_i(y_i) \Big(\sum_{S \in \mathcal{S}} \xi_S [\mathbf{1}_S]_i - \sum_{S \in \mathcal{S}} z_S [\mathbf{1}_S]_i \Big) \\
&= \sum_{S \in \mathcal{S}} \xi_S \sum_{i \in S} c_i(y_i) \;-\; \sum_{S \in \mathcal{S}} z_S \sum_{i \in S} c_i(y_i) \\
&= \sum_{S \in \mathcal{S}} \xi_S\, c_S(y) \;-\; \sum_{S \in \mathcal{S}} z_S\, c_S(y).
\end{aligned}$$

Let $C_{\min} := \min_{S \in \mathcal{S}} c_S(y)$. By the Wardrop condition, $c_S(y) = C_{\min}$ whenever $z_S > 0$, and therefore $\sum_S z_S c_S(y) = C_{\min}$. On the other hand, $\sum_S \xi_S c_S(y) \geq C_{\min}$ for any $\xi \in \Delta^d$. Hence

$$\langle \nabla f(y),\, y' - y \rangle \;=\; \sum_S \xi_S c_S(y) - C_{\min} \;\geq\; 0 \qquad \text{for all } y' \in \mathcal{C}.$$

Thus y satisfies (12) and so y minimizes f over $\mathcal{C}$. By uniqueness of the minimizer, $y = y^\star$.

(Minimizer ⇒ equilibrium). Conversely, let $y \in \mathcal{C}$ minimize f over $\mathcal{C}$, and let $z \in \Delta^d$ be any population profile with $y = y(z)$. Assume for contradiction that z is not a Wardrop equilibrium. Then there exist strategies $S, S' \in \mathcal{S}$ such that $z_S > 0$ and $c_{S'}(y) < c_S(y)$.

For sufficiently small $\varepsilon > 0$ we can define a perturbed profile $z^\varepsilon \in \Delta^d$ by moving mass ε from S to S':

$$z^\varepsilon_S = z_S - \varepsilon, \qquad z^\varepsilon_{S'} = z_{S'} + \varepsilon, \qquad z^\varepsilon_T = z_T \text{ for all } T \notin \{S, S'\},$$

and let $y^\varepsilon := y(z^\varepsilon) \in \mathcal{C}$ be the induced load. Then

$$y^\varepsilon - y = \varepsilon(\mathbf{1}_{S'} - \mathbf{1}_S).$$

Using differentiability of f we compute the directional derivative of f at y in the direction $y^\varepsilon - y$:

$$\begin{aligned}
\frac{d}{d\varepsilon} f(y^\varepsilon)\Big|_{\varepsilon=0} &= \langle \nabla f(y),\, \mathbf{1}_{S'} - \mathbf{1}_S \rangle \\
&= \sum_{i \in S'} c_i(y_i) - \sum_{i \in S} c_i(y_i) \;=\; c_{S'}(y) - c_S(y) \;<\; 0.
\end{aligned}$$

Therefore, for all sufficiently small $\varepsilon > 0$ we have $f(y^\varepsilon) < f(y)$, contradicting the fact that y minimizes f over $\mathcal{C}$. Hence no such pair (S, S') can exist, and z must be a Wardrop equilibrium.

Combining the two directions, we obtain the claimed equivalence. □

B.2. Proof of Lemma 2.6

Proof. **Step 1 (a uniform local Lipschitz step).** Let $\delta := \alpha r / L_{f,2}$. We first show that whenever $\|\theta - \theta'\| \leq \delta$,

$$\|y^\star(\theta) - y^\star(\theta')\| \leq \frac{L_{f,2}}{\alpha}\, \|\theta - \theta'\|. \tag{13}$$

Fix such θ, θ' and denote $y := y^\star(\theta)$, $y' := y^\star(\theta')$, $d := y' - y$. By the first-order optimality (variational inequality) of y and y' over the convex set C,

$$\langle \nabla_y f(\theta, y),\, y' - y\rangle \geq 0, \qquad \langle \nabla_y f(\theta', y'),\, y - y'\rangle \geq 0.$$

Adding yields

$$\big\langle \nabla_y f(\theta, y) - \nabla_y f(\theta', y'),\, d\big\rangle \geq 0. \tag{14}$$

Add and subtract $\nabla_y f(\theta, y')$:

$$\big\langle \nabla_y f(\theta, y) - \nabla_y f(\theta, y'),\, d\big\rangle \geq \big\langle \nabla_y f(\theta', y') - \nabla_y f(\theta, y'),\, d\big\rangle.$$

By Assumption 2.4 (equivalently, local strong convexity / strong monotonicity around $y^\star(\theta)$ on radius r), whenever $\|d\| \leq r$ we have

$$\big\langle \nabla_y f(\theta, y') - \nabla_y f(\theta, y),\, d\big\rangle \geq \alpha \|d\|^2.$$

Using Cauchy–Schwarz and Assumption 2.5 with $y = y'$ gives

$$\big\langle \nabla_y f(\theta', y') - \nabla_y f(\theta, y'),\, d\big\rangle \leq \|\nabla_y f(\theta', y') - \nabla_y f(\theta, y')\|\, \|d\| \leq L_{f,2}\|\theta - \theta'\|\, \|d\|.$$

Combining yields $\alpha\|d\|^2 \leq L_{f,2}\|\theta - \theta'\|\, \|d\|$, hence $\|d\| \leq (L_{f,2}/\alpha)\|\theta - \theta'\|$. Under $\|\theta - \theta'\| \leq \delta = \alpha r / L_{f,2}$, this also implies $\|d\| \leq r$, so the local strong-convexity inequality applies self-consistently. This proves (13).

Step 2 (global Lipschitzness by chaining). Fix arbitrary $\theta, \theta' \in \Theta$. If $\theta = \theta'$ the claim is trivial. Otherwise set

$$m := \left\lceil \frac{\|\theta - \theta'\|}{\delta} \right\rceil \quad \text{and} \quad \theta_i := \theta + \frac{i}{m}(\theta' - \theta), \;\; i = 0, 1, \ldots, m.$$

Since Θ is convex, $\theta_i \in \Theta$ for all i, and $\|\theta_{i+1} - \theta_i\| = \|\theta' - \theta\|/m \leq \delta$. Applying the local step (13) to each consecutive pair and summing gives

$$\|y^\star(\theta) - y^\star(\theta')\| \leq \sum_{i=0}^{m-1} \|y^\star(\theta_{i+1}) - y^\star(\theta_i)\| \leq \frac{L_{f,2}}{\alpha} \sum_{i=0}^{m-1} \|\theta_{i+1} - \theta_i\| = \frac{L_{f,2}}{\alpha}\|\theta - \theta'\|,$$

which is (7).

Step 3 (Lipschitzness of Φ). Using Assumption 2.5 for F,

$$|\Phi(\theta) - \Phi(\theta')| = |F(\theta, y^\star(\theta)) - F(\theta', y^\star(\theta'))| \leq L_{F,1}\|\theta - \theta'\| + L_{F,1}\|y^\star(\theta) - y^\star(\theta')\|.$$

Plug (7) into the last term to obtain (8). □

Recall the lower-level problem from Section 2: for each parameter $\theta \in \Theta \subseteq \mathbb{R}^k$ the Wardrop equilibrium load $y^\star(\theta)$ is defined as the unique minimizer of the potential

$$\min_{y \in \mathcal{C}} f(\theta, y),$$

where $\mathcal{C} \subset \mathbb{R}^n$ is a fixed convex polytope (the set of feasible load vectors) and $f(\cdot; \theta)$ is the Beckmann-type potential associated with the congestion costs. In the main text we only assume that the equilibrium map $\theta \mapsto y^\star(\theta)$ is locally Lipschitz and allow it to be non-differentiable, which is the natural regime for combinatorial congestion games.

For completeness, we briefly summarize standard sufficient conditions under which $y^\star(\theta)$ is in fact *locally differentiable* with respect to θ. These conditions come from classical parametric optimization and bilevel optimization; we do not use them in our analysis, but they clarify in what sense our non-smooth viewpoint is more general.

Parametric convex program and KKT system. We write the lower-level problem in the form

$$\min_{y \in \mathbb{R}^n} f(\theta, y) \quad \text{subject to} \quad Ay = b,\ Gy \leq h, \tag{15}$$

where $\mathcal{C} = \{y \in \mathbb{R}^n : Ay = b,\ Gy \leq h\}$ and $\theta \in \Theta$ appears only in the objective $f(\cdot; \theta)$. Let $y^\star(\theta)$ denote the (assumed unique) solution of (15) and let $(\lambda^\star(\theta), \mu^\star(\theta))$ be the associated Lagrange multipliers for equality and inequality constraints. The KKT conditions for (15) at a given θ are

$$\begin{aligned} \nabla_y f(\theta, y) + A^\top \lambda + G^\top \mu &= 0, \\ Ay - b &= 0, \\ Gy - h \leq 0, \quad \mu \geq 0, \quad \mu^\top (Gy - h) &= 0. \end{aligned}$$

Under appropriate regularity assumptions, this system defines $(y^\star(\theta), \lambda^\star(\theta), \mu^\star(\theta))$ as the unique solution of a smooth (generalized) equation depending on θ.

A standard sufficient condition. The following hypothesis is a typical set of assumptions used in perturbation/sensitivity analysis of nonlinear programs (see, e.g., [33, Ch. 4–5], [34]).

Assumption B.1 (Local strong regularity at θ_0)**.** *Fix $\theta_0 \in \Theta$. Suppose that:*

1. *$f(\cdot; \cdot)$ is twice continuously differentiable in (y, θ) in a neighborhood of $(y^\star(\theta_0), \theta_0)$, and $f(\cdot; \theta_0)$ is strongly convex in y;*
2. *The feasible set $\mathcal{C}$ can be written as in (15) and admits a Slater point (a strictly feasible point) independent of θ;*
3. *At $y^\star(\theta_0)$, the active inequality constraints satisfy a linear independence constraint qualification (LICQ), and there exists a unique multiplier pair $(\lambda^\star(\theta_0), \mu^\star(\theta_0))$ solving the KKT system;*
4. *A second-order sufficient condition holds at $(y^\star(\theta_0), \lambda^\star(\theta_0), \mu^\star(\theta_0))$, e.g., the Hessian $\nabla^2_{yy} f(y^\star(\theta_0); \theta_0)$ is positive definite on the critical cone defined by the active constraints.*

Assumption B.1 is a specialization of Robinson's *strong regularity* for the KKT system viewed as a generalized equation [34], and matches the standard setting in the perturbation analysis of nonlinear programs [33, 35].

Proposition B.2 (Local differentiability of the equilibrium map)**.** *Let $\theta_0 \in \Theta$ and suppose that Assumption B.1 holds at θ_0. Then there exists a neighborhood U of θ_0 and a unique mapping*

$$(\theta \mapsto y^\star(\theta), \lambda^\star(\theta), \mu^\star(\theta))$$

defined on U such that, for every $\theta \in U$, the triple $(y^\star(\theta), \lambda^\star(\theta), \mu^\star(\theta))$ satisfies the KKT conditions of (15). Moreover, this mapping is continuously differentiable on U. In particular, the equilibrium map $\theta \mapsto y^\star(\theta)$ is C^1 on U, and the hyper-objective $\Phi(\theta) = F(\theta, y^\star(\theta))$ is differentiable on U whenever F is differentiable.

The proof follows from treating the KKT conditions as a generalized equation and applying Robinson's implicit function theorem for strongly regular generalized equations [34], or, equivalently, from the sensitivity analysis of parametric nonlinear programs as developed by Bonnans and Shapiro [33, Ch. 4–5]. We refer also to Dempe's monograph on bilevel programming [36, Ch. 2] for a discussion of similar assumptions in the context of bilevel problems.

Interpretation for CCGs. In the combinatorial congestion game setting, the strong regularity conditions above have a clear but restrictive interpretation. Strong convexity of $f(\cdot; \theta_0)$ and uniqueness of the multipliers encode uniqueness and stability of the Wardrop equilibrium at θ_0. LICQ and the second-order condition essentially require that the active constraints of $\mathcal{C}$ (and hence the set of used strategies and binding load constraints) remain unchanged under small perturbations of θ.

In realistic networks, however, small parameter changes often cause new strategies to become attractive or previously used strategies to drop out of equilibrium, so active sets change frequently. As a result, the differentiable regime described by Proposition B.2 typically holds only on relatively

small regions of the parameter space, separated by kinks where the equilibrium map is merely Lipschitz and non-differentiable. This is the main reason why, in the body of the paper, we adopt a non-smooth viewpoint and work with zeroth-order methods and generalized stationarity notions instead of relying on differentiability of $y^\star(\theta)$.

C. Proofs for Section 5

C.1. Auxiliary lemmas

Throughout this appendix we fix a parameter $\theta \in \Theta$ and write $f(y) := f(\theta, y)$, $y^\star := y^\star(\theta) \in \arg\min_{y\in\mathcal{C}} f(y)$. Let $D := \max\{\|u - v\| : u, v \in \mathcal{C}\}$.

Lemma C.1 (From local quadratic growth to a global error bound)**.** *Assume Assumption 2.4: there exist $\alpha > 0$ and $r > 0$ such that for all $y \in \mathcal{C}$ with $\|y - y^\star\| \le r$,*

$$f(y) - f(y^\star) \ge \frac{\alpha}{2}\|y - y^\star\|^2.$$

If f is convex on $\mathcal{C}$, then for every $y \in \mathcal{C}$,

$$f(y) - f(y^\star) \;\ge\; \frac{\alpha}{2}\min\{\|y - y^\star\|^2,\; r\|y - y^\star\|\}.$$

In particular, if $f(y) - f(y^\star) \le \frac{\alpha}{2}r^2$, then $\|y - y^\star\| \le r$ and $\|y - y^\star\|^2 \le \frac{2}{\alpha}(f(y) - f(y^\star))$.

Proof. If $\|y - y^\star\| \le r$ the claim is exactly Assumption 2.4. Otherwise let $d := \|y - y^\star\| > r$ and define the point on the segment $y_r := y^\star + \frac{r}{d}(y - y^\star)$ so that $\|y_r - y^\star\| = r$. By convexity,

$$f(y_r) \le \Big(1 - \frac{r}{d}\Big)f(y^\star) + \frac{r}{d}f(y) \quad\Rightarrow\quad f(y) - f(y^\star) \ge \frac{d}{r}\big(f(y_r) - f(y^\star)\big).$$

Applying Assumption 2.4 to y_r yields $f(y_r) - f(y^\star) \ge \frac{\alpha}{2}r^2$, hence $f(y) - f(y^\star) \ge \frac{d}{r}\cdot\frac{\alpha}{2}r^2 = \frac{\alpha r}{2}d$. This gives the stated $\min\{\cdot,\cdot\}$ lower bound and the corollaries. □

C.2. Proof of Theorem 5.2

Proof. Fix θ and abbreviate $f(y) := f(\theta, y)$ and $y^\star := y^\star(\theta)$. Let $h_t := f(y_t) - f(y^\star)$. By $L_{f,2}$-smoothness and for all $y, s \in \mathcal{C}$ with $\|s - y\| \le D$ and any $\gamma \in [0, 1]$,

$$f\big((1-\gamma)y + \gamma s\big) \le f(y) + \gamma\langle\nabla f(y), s - y\rangle + \frac{C_f}{2}\gamma^2, \qquad \text{where } C_f := L_{f,2}D^2. \tag{16}$$

Let $s_t^\star \in \arg\min_{s\in\mathcal{C}}\langle\nabla f(y_t), s\rangle$ and define the (exact) FW gap

$$g_t := \langle\nabla f(y_t), y_t - s_t^\star\rangle.$$

By convexity, $h_t \le g_t$ for all t.

Step 1: the hit event. Let $\mathcal{H}_t$ be the event that the m i.i.d. samples used by LMO_m contain an exact minimizer of $s \mapsto \langle\nabla f(y_t), s\rangle$. On $\mathcal{H}_t$, the best-of-m rule returns an exact minimizer, hence $s_t = s_t^\star$. By Assumption 5.1, conditionally on y_t,

$$\Pr(\mathcal{H}_t \mid y_t) \ge \kappa_m. \tag{17}$$

Step 2: line-search decrease on the hit event and definition of ψ. Since $\gamma = 0$ is feasible in the line-search, we always have $f(y_{t+1}) \le f(y_t)$, hence $h_{t+1} \le h_t$. On the hit event $\mathcal{H}_t$ we have $s_t = s_t^\star$, and therefore

$$f(y_{t+1}) = \min_{\gamma\in[0,1]} f\big((1-\gamma)y_t + \gamma s_t^\star\big).$$

Combining this with (16) (with $y = y_t$, $s = s_t^\star$) yields, for every $\gamma \in [0, 1]$,

$$f(y_{t+1}) \le f(y_t) - \gamma g_t + \frac{C_f}{2}\gamma^2.$$

Define

$$\psi(g) := \max_{\gamma \in [0,1]} \Big\{ \gamma g - \frac{C_f}{2}\gamma^2 \Big\}. \tag{18}$$

Then on $\mathcal{H}_t$,

$$h_{t+1} \le h_t - \psi(g_t). \tag{19}$$

Moreover, (18) has the closed form

$$\psi(g) = \begin{cases} \frac{g^2}{2C_f}, & 0 \le g \le C_f, \\ g - \frac{C_f}{2}, & g \ge C_f, \end{cases} \tag{20}$$

since the maximizer is $\gamma^\star(g) = \min\{g/C_f,\, 1\}$. In particular, $\psi(g) \ge 0$ for all $g \ge 0$.

Step 3: conditional recursion. From (18), for each fixed $\gamma \in [0,1]$ the map $g \mapsto \gamma g - \frac{C_f}{2}\gamma^2$ is affine and nondecreasing in g. Hence ψ, being the pointwise supremum over $\gamma \in [0,1]$ of such functions, is (i) *nondecreasing* and (ii) *convex* on $[0,\infty)$.

On $\mathcal{H}_t^c$ we have $h_{t+1} \le h_t$. Using (19) on $\mathcal{H}_t$ and conditioning on y_t,

$$\begin{aligned} \mathbb{E}[h_{t+1} \mid y_t] &\le \Pr(\mathcal{H}_t \mid y_t)\,(h_t - \psi(g_t)) + \Pr(\mathcal{H}_t^c \mid y_t)\, h_t \\ &= h_t - \Pr(\mathcal{H}_t \mid y_t)\,\psi(g_t) \;\le\; h_t - \kappa_m\,\psi(g_t), \end{aligned} \tag{21}$$

where we used (17). Since $g_t \ge h_t$ and ψ is nondecreasing, $\psi(g_t) \ge \psi(h_t)$, so

$$\mathbb{E}[h_{t+1} \mid y_t] \le h_t - \kappa_m\,\psi(h_t). \tag{22}$$

Take total expectation in (22):

$$\mathbb{E}[h_{t+1}] \le \mathbb{E}[h_t] - \kappa_m\,\mathbb{E}[\psi(h_t)].$$

Since ψ is convex, Jensen gives $\mathbb{E}[\psi(h_t)] \ge \psi(\mathbb{E}[h_t])$. Let $a_t := \mathbb{E}[h_t]$. Then

$$a_{t+1} \le a_t - \kappa_m\,\psi(a_t). \tag{23}$$

Step 4: solve the recursion. Using the closed form (20):

Regime 1: $a_t \ge C_f$. Then $\psi(a_t) = a_t - \frac{C_f}{2}$ and (23) yields

$$a_{t+1} \le (1-\kappa_m)a_t + \kappa_m \frac{C_f}{2}.$$

Iterating gives

$$a_t \le (1-\kappa_m)^t a_0 + \frac{C_f}{2} \qquad \forall t \ge 0. \tag{24}$$

Regime 2: $a_t \le C_f$. Then $\psi(a_t) = \frac{a_t^2}{2C_f}$ and (23) becomes

$$a_{t+1} \le a_t - \frac{\kappa_m}{2C_f} a_t^2. \tag{25}$$

Let $c := \kappa_m/(2C_f)$. Since $a_{t+1} \le a_t$, we may assume $a_t > 0$. From (25) we have $a_{t+1} \le a_t(1 - ca_t)$, hence

$$\frac{1}{a_{t+1}} \ge \frac{1}{a_t(1-ca_t)}.$$

Using $a_t \le C_f$ gives $ca_t \le \kappa_m/2 < 1$, and thus $\frac{1}{1-x} \ge 1 + x$ for $x \in [0,1)$ implies

$$\frac{1}{a_{t+1}} \ge \frac{1}{a_t}\Big(1 + ca_t\Big) = \frac{1}{a_t} + c.$$

Summing from $t = 0$ to $T-1$ yields $\frac{1}{a_T} \ge \frac{1}{a_0} + cT \ge cT$, hence

$$a_T \le \frac{1}{cT} = \frac{2C_f}{\kappa_m T}. \tag{26}$$

From Regime 1 we have for all $t \geq 0$ the geometric bound

$$a_t \leq (1-\kappa_m)^t a_0 + \frac{C_f}{2}. \tag{24}$$

Choose T_0 as in (10). Using $1 - x \leq e^{-x}$ for $x \in [0,1]$ gives

$$(1-\kappa_m)^{T_0} a_0 \leq e^{-\kappa_m T_0} a_0 \leq \frac{C_f}{2},$$

and therefore

$$a_{T_0} \leq (1-\kappa_m)^{T_0} a_0 + \frac{C_f}{2} \leq C_f.$$

Hence, for all $t \geq T_0$ the iterates satisfy $a_t \leq C_f$, so Regime 2 applies from time T_0 onward. Applying the sublinear estimate (26) to the shifted sequence $\{a_{T_0}, a_{T_0+1}, \dots\}$ yields, for all $T \geq T_0$,

$$a_T \leq \frac{2C_f}{\kappa_m (T - T_0 + 1)}.$$

Recalling $a_T = \mathbb{E}[h_T] = \mathbb{E}[f(\theta, y_T) - f(\theta, y^\star(\theta))]$ proves (9).

□

C.3. Proof of Theorem 5.3

This proof is a standard "randomized smoothing + projected (stochastic) gradient mapping" argument; see, e.g., [25–27, 37]. We include it for completeness and identify the only place where the inner-loop error enters.

Step 1: smoothing and Goldstein subdifferentials. Define the ball-smoothed function

$$\Phi_\rho(\theta) := \mathbb{E}_{v \sim \mathrm{Unif}(\mathbb{B})}[\Phi(\theta + \rho v)].$$

This is the standard uniform-ball smoothing used in gradient-free methods [27, 37]. The following facts are standard for Lipschitz Φ:

1. (Approximation) $|\Phi_\rho(\theta) - \Phi(\theta)| \leq \rho L_\Phi$ for all θ [27].
2. (Differentiability) Φ_ρ is continuously differentiable, and $\nabla \Phi_\rho$ is Lipschitz with constant $L_\rho \leq c\, \frac{L_\Phi \sqrt{k}}{\rho}$ for a universal constant $c > 0$ [27].
3. (Goldstein inclusion) $\nabla \Phi_\rho(\theta) \in \partial_\rho \Phi(\theta)$ for all θ [9].

We use these as black-box smoothing properties.

Step 2: the two-point estimator and the inexact value oracle. Let $u_t \sim \mathrm{Unif}(\mathbb{S}^{k-1})$ and consider the symmetric estimator (as in Algorithm 2)

$$\widehat{g}_t := \frac{k}{2\rho}\Big(\widehat{\Phi}_T(\bar{\theta}_t + \rho u_t) - \widehat{\Phi}_T(\bar{\theta}_t - \rho u_t)\Big) u_t, \qquad \widehat{\Phi}_T(\theta) = F(\theta, y_T(\theta)).$$

Define also the "ideal" estimator using the exact hyper-objective:

$$g_t := \frac{k}{2\rho}\Big(\Phi(\bar{\theta}_t + \rho u_t) - \Phi(\bar{\theta}_t - \rho u_t)\Big) u_t.$$

Then $\mathbb{E}[g_t \mid \bar{\theta}_t] = \nabla \Phi_\rho(\bar{\theta}_t)$ [26, 27, 37], and $\mathbb{E}\|g_t\|^2 \leq c' k L_\Phi^2$ for a universal constant $c' > 0$.

Now define the pointwise oracle errors $e_t^\pm := \widehat{\Phi}_T(\bar{\theta}_t \pm \rho u_t) - \Phi(\bar{\theta}_t \pm \rho u_t)$. By $L_{F,y}$-Lipschitzness of $F(\theta, \cdot)$ and the definition of ε_y,

$$\mathbb{E}|e_t^\pm| \leq L_{F,y}\, \mathbb{E}\|y_T(\bar{\theta}_t \pm \rho u_t) - y^\star(\bar{\theta}_t \pm \rho u_t)\| \leq L_{F,y} \varepsilon_y.$$

Consequently,

$$\mathbb{E}\|\widehat{g}_t - g_t\| \leq \frac{k}{2\rho}\, \mathbb{E}\big(|e_t^+| + |e_t^-|\big) \leq \frac{k}{\rho} L_{F,y} \varepsilon_y. \tag{27}$$

Step 3: projected descent on the smoothed objective. Let $\theta_{t+1} = \Pi_\Theta(\theta_t - \eta \widehat{g}_t)$. Using the smoothness of Φ_ρ and the standard projection inequality (see, e.g., [25]), one obtains the one-step bound (for $\eta \le 1/L_\rho$)

$$\Phi_\rho(\theta_{t+1}) \le \Phi_\rho(\theta_t) - \frac{\eta}{2}\|G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta)\|^2 + \eta\big\langle G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta),\, \widehat{g}_t - \nabla\Phi_\rho(\theta_t)\big\rangle.$$

Taking expectation and using Cauchy–Schwarz yields

$$\frac{\eta}{2}\,\mathbb{E}\|G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta)\|^2 \le \mathbb{E}[\Phi_\rho(\theta_t) - \Phi_\rho(\theta_{t+1})] + \eta\,\mathbb{E}\|G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta)\| \cdot \mathbb{E}\|\widehat{g}_t - \nabla\Phi_\rho(\theta_t)\|.$$

Summing over $t = 0, \dots, K-1$ and bounding $\Phi_\rho(\theta_K) \ge \inf_\Theta \Phi_\rho$ gives

$$\frac{\eta}{2}\sum_{t=0}^{K-1}\mathbb{E}\|G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta)\|^2 \le \Phi_\rho(\theta_0) - \inf_\Theta \Phi_\rho + \eta\sum_{t=0}^{K-1}\mathbb{E}\|G_\Theta(\theta_t, \nabla\Phi_\rho(\theta_t); \eta)\| \cdot \mathbb{E}\|\widehat{g}_t - \nabla\Phi_\rho(\theta_t)\|.$$

Using the smoothing approximation $\Phi_\rho(\theta_0) - \inf \Phi_\rho \le \Phi(\bar{\theta}_0) - \inf_\Theta\Phi + 2\rho L_\Phi$, and the bias control (27) (absorbing the intrinsic smoothing-estimation terms into constants by the stepsize restriction $\eta \le c_0\rho/(L_\Phi\sqrt{k})$), one obtains universal constants $C_1, C_2 > 0$ such that

$$\mathbb{E}\Big[\min_{0\le t\le K-1}\|G_\Theta(\bar{\theta}_t, \nabla\Phi_\rho(\bar{\theta}_t); \eta)\|\Big] \le C_1\sqrt{\frac{\Phi(\bar{\theta}_0) - \inf_\Theta \Phi + 2\rho L_\Phi}{\eta K}} \;+\; C_2\frac{k}{\rho}\,L_{F,y}\,\varepsilon_y.$$

Finally, since $\nabla\Phi_\rho(\bar{\theta}_t) \in \partial_\rho\Phi(\bar{\theta}_t)$, we may take $g = \nabla\Phi_\rho(\bar{\theta}_t)$ in the definition of GGSP, and choosing $\widehat{\theta}$ uniformly at random from $\{\bar{\theta}_0, \dots, \bar{\theta}_{K-1}\}$ yields (11).

D. Zero-supressed Decision Diagrams (ZDD)

Zero-suppressed decision diagrams (ZDDs) are a canonical data structure for representing a family of subsets of a finite universe in a compact form. They can be viewed as a specialization of reduced ordered binary decision diagrams (BDDs) [19] tailored to sparse set systems [7]. In our setting, the universe typically consists of graph resources (e.g., edges), and a feasible strategy (path, tree, cycle) is encoded by its incidence vector, i.e., a subset of universe items.

D.1. Definition and construction

ZDDs as compressed set representations. Fix an ordered universe $U = (e_1, \dots, e_n)$ and a family $\mathcal{S} \subseteq 2^U$. A ZDD for $\mathcal{S}$ is a rooted directed acyclic graph with two terminals $\top$ and $\bot$. Each internal node v is labeled by an item index $\ell(v) \in \{1, \dots, n\}$ and has two outgoing arcs: a *lo*-arc corresponding to excluding $e_{\ell(v)}$ and a *hi*-arc corresponding to including $e_{\ell(v)}$. A root-to-$\top$ path encodes a unique subset $S \subseteq U$ given by the labels whose hi-arcs are taken; the family represented by the diagram is exactly the collection of subsets obtained from all root-to-$\top$ paths. Figure 2 illustrates this correspondence on a toy network.

The key advantage of ZDDs is that they admit aggressive reductions. As in BDDs, isomorphic subgraphs are merged so that identical subproblems are shared [19]. In addition, the *zero-suppression* rule removes any node whose hi-child is $\bot$, redirecting the incoming arc(s) to its lo-child [7]. Intuitively, if including an item can never lead to a feasible set, then that item is irrelevant for the family and should be skipped. After applying these reductions, the resulting diagram is canonical for a fixed variable ordering of U.

Construction and size. ZDDs can be built by incremental, constraint-driven enumeration procedures that traverse items in the chosen order and maintain a compact representation of partial feasibility. For graph families (paths, Steiner structures, cycles), such constructions often proceed by a frontier-style dynamic program that tracks only local connectivity information at the boundary between processed and unprocessed edges. The resulting preprocessing time is roughly linear in the number of ZDD nodes produced.

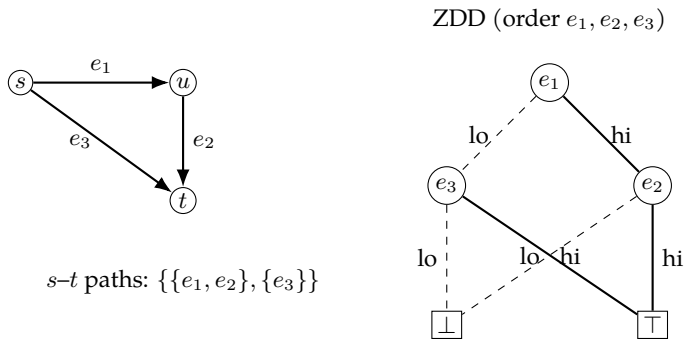


Figure 2: Left: a three-edge network with two s–t paths. Right: a ZDD encoding the corresponding strategy family. Root-to-$\top$ paths correspond to feasible strategies, with hi-arcs indicating selected edges.

The practical challenge is that ZDD size is highly instance dependent. In the worst case, the number of nodes can grow exponentially with n (and it depends sensitively on the chosen ordering), which makes ZDD-based approaches unsuitable as a universal replacement for combinatorial optimization. However, when the underlying constraints have exploitable structure, ZDDs can be orders of magnitude smaller than the explicit family $\mathcal{S}$ and enable exact computations over exponentially large strategy sets [7].

D.2. Exact optimization over a ZDD

A ZDD enables exact minimization of any additive weight function over the underlying family. Given weights $w \in \mathbb{R}^n$ (e.g., edge costs), consider

$$\min_{S \in \mathcal{S}} \sum_{e_i \in S} w_i.$$

Because each feasible strategy corresponds to a root-to-$\top$ path, this problem is equivalent to finding a minimum-cost root-to-$\top$ path in the ZDD DAG, where taking the hi-arc at a node labeled i incurs cost w_i and taking the lo-arc incurs cost 0. Since the ZDD is acyclic, the optimum can be computed by a single bottom-up dynamic program. Writing $\mathrm{lo}(v)$ and $\mathrm{hi}(v)$ for the children of a node v and letting $\mathrm{cost}(\top) = 0$, $\mathrm{cost}(\bot) = +\infty$, we compute

$$\mathrm{cost}(v) = \min\{\mathrm{cost}(\mathrm{lo}(v)),\ w_{\ell(v)} + \mathrm{cost}(\mathrm{hi}(v))\},$$

in reverse topological order. Recovering an optimal strategy amounts to following the minimizing branch from the root. The runtime is linear in the number of ZDD nodes (and requires only additions and minima), so the diagram acts as an *exact linear minimization oracle* over an exponentially large family.

D.3. Sampling strategies from a ZDD

Beyond exact optimization, ZDDs support efficient sampling of feasible strategies without enumerating $\mathcal{S}$. This is useful when exact optimization becomes costly (e.g., for very large diagrams) and one wishes to form a subsampled candidate set. Sampling is again implemented by dynamic programming followed by a randomized root-to-$\top$ traversal.

Uniform sampling over strategies. Let $N(v)$ denote the number of root-to-$\top$ completions from a node v. These counts satisfy $N(\top) = 1$, $N(\bot) = 0$, and

$$N(v) = N(\mathrm{lo}(v)) + N(\mathrm{hi}(v)).$$

After computing $N(v)$ for all nodes bottom-up, a uniformly random strategy is obtained by starting at the root and, at each node v, choosing the lo-arc with probability $N(\mathrm{lo}(v))/N(v)$ and the hi-arc with probability $N(\mathrm{hi}(v))/N(v)$. Whenever the hi-arc is chosen, we include the corresponding item $e_{\ell(v)}$. This produces an exactly uniform draw from $\mathcal{S}$.

Length-stratified sampling. In many graph families, the distribution of strategy sizes $|S|$ is highly skewed, so uniform sampling over $\mathcal{S}$ can overwhelmingly produce long, dense strategies. To control the sampled lengths, it is convenient to first choose a target length and then sample uniformly conditional on that length.

Let $N_r(v)$ denote the number of completions from v that select exactly r additional items. These length-refined counts satisfy $N_0(\top) = 1$, $N_r(\top) = 0$ for $r \geq 1$, $N_r(\bot) = 0$, and

$$N_r(v) = N_r(\mathrm{lo}(v)) + N_{r-1}(\mathrm{hi}(v)), \qquad r \geq 0,$$

with the convention $N_{-1}(\cdot) = 0$. Given a desired remaining length r, a uniformly random strategy of that length is obtained by traversing from the root and selecting the lo-arc with probability $N_r(\mathrm{lo}(v))/N_r(v)$ (keeping r unchanged) and selecting the hi-arc with probability $N_{r-1}(\mathrm{hi}(v))/N_r(v)$ (including $e_{\ell(v)}$ and decrementing r).

Choosing the length distribution. Let $\mathcal{L} := \{r \geq 0 : N_r(\mathrm{root}) > 0\}$ denote the set of feasible lengths. A general length-stratified sampler is defined by choosing a distribution π on $\mathcal{L}$ and then sampling uniformly conditional on the selected length. We use three choices:

- **Uniform strategy (US).** Choose r with probability $\pi(r) = N_r(\mathrm{root})/\sum_{\ell \in \mathcal{L}} N_\ell(\mathrm{root})$. This recovers uniform sampling over all strategies.
- **Uniform length (UL).** Choose r uniformly over feasible lengths, i.e., $\pi(r) = 1/|\mathcal{L}|$ for $r \in \mathcal{L}$. This equalizes the mass assigned to each length class.
- **Harmonic length (HL).** Choose r with probability proportional to $1/r$ over $\mathcal{L} \cap \{1, 2, \dots\}$. This further biases sampling toward shorter strategies.

Complexity and numerical stability. Once the diagram is built, uniform-strategy sampling requires only the node counts $N(v)$ and then one root-to-terminal traversal per sample. Length-stratified sampling requires computing the table $(N_r(v))_{v,r}$ up to the maximum feasible length, after which each sample is generated by selecting a length and performing a single traversal with the appropriate conditional probabilities. In practice the counts can be astronomically large, so it is common to store them in log-scale and to form the branching probabilities via log-sum-exp arithmetic.