Efficient Network Reconfiguration by Randomized Switching

Samuel Talkington **Dmitrii M. Ostrovskii** **Daniel K. Molzahn**
Georgia Institute of Technology

Abstract

We present an algorithm that efficiently computes nearly-optimal solutions to a class of combinatorial *reconfiguration* problems on weighted, undirected graphs. Inspired by societally relevant applications in networked infrastructure systems, these problems consist of *simultaneously* finding an *unreweighted* sparsified graph and nodal potentials that satisfy *fixed* demands, where the objective is to minimize some congestion criterion, e.g., a Laplacian quadratic form. These problems are mixed-integer nonlinear programming problems that are NP-hard, in general. To circumvent these challenges, instead of solving for a single best configuration, the proposed *randomized switching* algorithm seeks to design a distribution of configurations that, when sampled, ensures that congestion concentrates around its optimum. We show that the proposed congestion metric is a generalized self-concordant function in the space of switching probabilities, which enables the use of efficient conditional gradient methods. We implement our algorithm and show that it outperforms a state-of-the-art commercial mixed-integer second-order cone programming (MISOCP) solver by orders of magnitude over a large range of problem sizes.

1 INTRODUCTION

Consider an undirected weighted graph $G = (N, E, \boldsymbol{w})$ with $n = |N|$ nodes and $m = |E|$ *switchable edges* which can be opened or closed. The edge weights $\boldsymbol{w} \in \mathbb{R}^m_+$ are fixed. We are given a fixed vector of nodal demands $\boldsymbol{d} \in \mathbb{R}^n$ that satisfies $\boldsymbol{d}^\top \mathbf{1} = 0$. A *network reconfiguration problem* is to find edge *switching statuses* $\boldsymbol{s} \in S \subseteq \{0,1\}^m$ *and* nodal potentials $\boldsymbol{x} \in \mathbb{R}^n$ that minimize a congestion metric, subject to the constraints that the graph is connected and the nodal potentials satisfy network physics for the given demands over the switched graph. A network reconfiguration problem can be formulated as the following parametric mixed-integer non-linear program (MINLP):

$$
\begin{aligned}
\mathcal{D}(\boldsymbol{d}):\quad & \min_{\boldsymbol{s},\boldsymbol{x}} && \boldsymbol{x}^\top \boldsymbol{\Gamma}_{\boldsymbol{s}} \boldsymbol{x} && \text{(Congestion)}\\
& \text{s. t.} && \lambda_2(\boldsymbol{L}_{\boldsymbol{s}}) \geq \lambda_\star && \text{(Connected)}\\
& && \boldsymbol{L}_{\boldsymbol{s}} \boldsymbol{x} = \boldsymbol{d} && \text{(Physics)}\\
& && \boldsymbol{s} \in S && \text{(Switching)}\\
& && \boldsymbol{x} \perp \mathbf{1}, && \text{(Voltages)}
\end{aligned}
$$

where $\boldsymbol{L}_{\boldsymbol{s}} := \sum_{j>i} \boldsymbol{E}_{ij} w_{ij} s_{ij}$ is the *model Laplacian* of the switched graph $G_{\boldsymbol{s}} := (N, E, \boldsymbol{w} \circ \boldsymbol{s})$, and $\boldsymbol{\Gamma}_{\boldsymbol{s}} := \sum_{j>i} \boldsymbol{E}_{ij} \gamma_{ij} s_{ij}$ is a *target Laplacian* for a graph with the same connectivity, but potentially different edge weights. The matrix $\boldsymbol{E}_{ij} = (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top$ is an elementary Laplacian describing a two-node graph. The constraint on the second-smallest eigenvalue $\lambda_2(\boldsymbol{L}_{\boldsymbol{s}}) \geq \lambda_\star$ ensures the graph remains connected.

There are several particularly relevant cases for the objective (Congestion) and the target Laplacian $\boldsymbol{\Gamma}_{\boldsymbol{s}}$, with natural connections to engineering applications, such as in electric power systems:

1. the original *model Laplacian*, i.e., $\gamma_{ij} = w_{ij}$ for all $ij \sim 1, \ldots, m$. In this case, if w_{ij} represents the *conductance* of edge $(i,j) \in E$, the objective becomes the *total electric power dissipated* across all nodes in the network $\sum_{i=1}^n d_i x_i := \sum_{i=1}^n p_i$.
2. a *capacitated Laplacian*, e.g., $y_{ij} = w_{ij}^2 / c_{ij}^2$, for all $ij \sim 1, \ldots, m$. In this case, the objective function becomes the sum of the normalized line flows $\sum_{i<j} f_{ij}^2 / c_{ij}^2$.
3. a *risk*, *fairness*, or *robustness* functional, which measures the importance of an edge with respect to some metric. For instance, y_{ij} could represent the probability that a wildfire ignites at a line in an electrical network, or the criticality of the edge in socioeconomic factors.

These cases can be derived in a straightforward manner from the general framework of electrical networks and their corresponding graph representations.

The proposed model has direct applications to widely used approximate models for electric power flow and water flow in real-world infrastructure networks, both of which obey the conditions of the proposed theory.

1.1 Motivation

The program $\mathcal{D}$ is inspired by more complex network reconfiguration problems that are of great current interest to electric power system researchers; see [Babaeinejadsarookolaee et al., 2023, Haider et al., 2025, van der Sar et al., 2025] for recent examples. Such problems are known as *transmission switching*, *topology control*, or *network reconfiguration* in the relevant literature. The constraint (Physics) is identical to *DC power flow*, which is a linear approximation of the nonlinear AC power flow equations. Even our primitive problem $\mathcal{D}$ is NP-hard in general (see [Lehmann et al., 2014, Kooij and Achterberg, 2023] for a related discussion), and the objective function is nonconvex.

A natural first attempt to solve the problem is to relax the switching variables $\boldsymbol{s}$ to be continuous, i.e., $\boldsymbol{s} \in [0,1]^m$, and then solve the resulting optimization problem. However, this approach does not yield a feasible solution to the original problem, since the relaxed solution need not be integral. Moreover, the relaxed problem is still NP-hard, as (Physics) involves bilinear equality constraints.

Applications of interest. The present paper proposes a new primitive blending randomized graph design and mixed-integer quadratic optimization. This primitive is inspired by the electric power transmission network reconfiguration problem, where transmission lines are switched on and off to improve grid stability while keeping generator dispatches the same. The algorithm is designed to be usable for the *DC power flow approximation* of the non-linear AC power flow equations, a canonical linearization that appears frequently in practice; see [Stott et al., 2009] for a salient review. This approximation has the attractive problem of being *identical* to the Poisson problem. In essence, we are given a *net power injection vector* $\boldsymbol{p} \in \mathbb{R}^n$ such that $\boldsymbol{p}^\top \mathbf{1} = 0$ and we want to find a *voltage phase angle* vector $\boldsymbol{\theta} \in (-\pi, \pi]^n$ such that

$$\boldsymbol{p} = \boldsymbol{L}\boldsymbol{\theta}, \qquad \boldsymbol{L} := -\boldsymbol{B}, \qquad \boldsymbol{\theta}^\top \mathbf{1} = 0,$$

where $\boldsymbol{B} \preceq 0$ is a weighted graph Laplacian matrix corresponding to an undirected graph with negative edge weights $b : N \times N \to \mathbb{R}_-$, known as *susceptances.*

However, the proposed theory is a general graph algorithm and is distinct from this literature, which has thus far taken largely deterministic or mixed-integer based approaches. By targeting approximate preservation of Poisson solutions with controllable energy reduction through randomized edge activation, we contribute.

1.2 Related work

Our work contributes to a rich literature in spectral graph theory; in particular, the *effective resistance* of graphs, which has been studied extensively in recent decades. It has shown promise for improving the resilience of complex networks [Wang et al., 2014] and improving connectivity [Ghosh and Boyd, 2006]. A particularly meaningful metric is the *total effective resistance* of a graph, which is defined as the sum of the effective resistances of all edges in the graph. The total effective resistance is a convex function of the edge weights [Ghosh et al., 2008], and it has found numerous applications.

Graph algorithms. Researchers have made numerous recent breakthroughs in fast algorithms for problems involving massive graph structures; see [Teng, 2010] for a central historical review. Some particularly influential recent results are in solving Laplacian systems [Kyng and Sachdeva, 2016a], maximum flow, and minimum cost flow [Chen et al., 2022]. These algorithms are unified in spirit by incorporating a degree of randomness. Similar to parallel independent work in [Zhou et al., 2025b, Brown et al., 2024] and previous related work in [Ghosh and Boyd, 2006, Li et al., 2020], we consider the setting of a graph with m potential edges, where we wish to add $q \ll m$ such edges to design a favorable connected graph. This is related to the problem of maximization of algebraic connectivity, that is, maximization of the second-smallest eigenvalue of the Laplacian matrix [Brown et al., 2024, Ghosh and Boyd, 2006], and also equivalent to maximization of the smallest eigenvalue of a grounded Laplacian matrix [Zhou et al., 2025a]. This a classic question in spectral graph theory has been explored from various angles. The work of [Li et al., 2020] viewed this through the lens of maximizing the number of spanning trees, and [Ghosh and Boyd, 2006] through the lens of optimizing over edge weights lying on the simplex, and in [Zhou et al., 2025a] through the lens of adding and deleting rows of the Laplacian. This was addressed through the lens of edge addition in [Ru et al., 2025]. The problem of maximizing the algebraic connectivity λ_2 can be viewed as seeking to improve global connectivity and robustness properties, which has found numerous applications in engineering [Nagarajan et al., 2015, Somisetty et al., 2024a, Somisetty et al., 2024b]. Our work complements this rich literature by focusing on the context of *fixed* demands and edge weights; this problem setting introduces distinct challenges and opportunities.

Spectral graph theory Effective resistances are one of the many useful results in the field of spectral graph theory. Multiple results related to this concept have inspired the present paper. In particular, the seminal result of [Spielman and Srivastava, 2011], which produced a randomized procedure for spectral sparsification—first defined in [Spielman and Teng, 2004]—has heavily influenced our work. The extension of the spectral sparsification framework to the case where the graph edges cannot be augmented by the algorithm, i.e., "unweighted sparsifiers" like [Anderson et al., 2014], are highly related to our work.

Spielman and Teng explored algorithms for fast solutions to Laplacian systems of equations [Spielman and Teng, 2012]. More recent advances, including works by Kyng and collaborators, emphasize randomized algorithms for graph-based linear solvers and approximations [Kyng and Sachdeva, 2016b], which seems to yield advantages in theoretical simplicity and practical speed [Gao et al., 2023].

Deriving concentration inequalities for quantities in algebraic and spectral graph theory is one of the most promising applications of random matrix theory [Chen et al., 2021]. Oliveira's investigations of concentration inequalities for Laplacians [Oliveira, 2010] was an early result in this direction, which gave rise to the matrix Freedman inequality [Tropp, 2011a]. Similarly, interesting results have been observed for the concentration of the total effective resistance [Boumal and Cheng, 2014], also known as the Kirchhoff index. A key component of our analysis relates to the literature on concentration inequalities for the minimum eigenvalues of positive-semidefinite matrices, e.g., see [Tropp, 2011b, Oliveira, 2013]. Previously, Brown, Laddha, and Singh developed a framework to maximize the minimum eigenvalues of the sums of rank-one matrices [Brown et al., 2024], which also relied on matrix concentration.

Randomized algorithms. Our work relates broadly to randomized algorithms, an in particular to their intersection with spectral graph theory. The two fields have deep connections. Foundational graph sparsification techniques, such as those developed by Spielman and Srivastava [Spielman and Srivastava, 2011], seek to approximate a Laplacian operator by a sparse subgraph while preserving spectral properties globally. Graph sparsification has found many applications such as regularization in machine learning [Sadhanala et al., 2016], quantum computing [Moondra et al., 2024], and others. Numerous other approaches to this problem, such as approximate matrix multiplication [Charalambides and Hero, 2023] have been proposed. In addition, the authors of [Lau and Zhou, 2020] gave an iterative randomized rounding algorithm approach to spectral network design. This is related to the problem of randomized experimental design [Allen-Zhu et al., 2021]. We add to this literature by focusing on randomized network reconfiguration task aimed at controlling a Laplacian quadratic form under a specific solution profile induced by a fixed demand vector. This reveals previously unknown structural properties about the objective, and efficient algorithms for optimizing it.

Algorithmic tools. One of our analyses is based on a Frank-Wolfe type procedure; see [Pokutta, 2024] and [Braun et al., 2025] for recent surveys. This class of first-order iterative convex optimization algorithms are particularly useful in the setting where projection steps are inefficient. Frank-Wolfe methods have found numerous applications, ranging from traffic assignment problems in transportation networks [Fukushima, 1984], to recent SDP solvers [Yurtsever et al., 2021, Pham et al., 2023].

2 PRELIMINARIES

The following parameterized positive-semidefinite (PSD) Laplacian matrix functional is the central object of study.

Definition 2.1 (Switched Laplacian)**.** For a multigraph $G = (N, E, \boldsymbol{w})$, let $\boldsymbol{A} \in \{-1, 0, 1\}^{m \times n}$ be the node-to-edge incidence matrix for G. Let $\boldsymbol{s} \in \{0, 1\}^m$ be *switching variables* corresponding to a reconfiguration, and let $\boldsymbol{L}_{(\cdot)} : \{0,1\}^m \to \mathbb{S}^{n \times n}$ be a *switched Laplacian matrix* functional, where

$$\boldsymbol{L}_{\boldsymbol{s}} := \boldsymbol{A}^\top \boldsymbol{W}^{1/2} \boldsymbol{S} \boldsymbol{W}^{1/2} \boldsymbol{A},$$

corresponds to the switched graph $G_{\boldsymbol{s}} := (N, E, \boldsymbol{w} \circ \boldsymbol{s})$, with $\boldsymbol{S} := \mathsf{diag}(\boldsymbol{s})$.

Def. 2.1 is also known as an *unweighted sparsifier* [Anderson et al., 2014]. From here, we can naturally define corresponding effective resistance and leverage score functionals.

Definition 2.2 (Effective resistance and leverage)**.** The effective resistance $\varrho_{ij} : \{0,1\}^m \to \mathbb{R}_+$ of a multiedge $ij \in E$ under switching strategy $\boldsymbol{s}$ is defined as

$$\varrho_{ij}(\boldsymbol{s}) := \boldsymbol{e}_{ij}^\top \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{e}_{ij}.$$

Moreover, let $w_{ij} \in \mathbb{R}_+$ be the weight of the edge ij. The *leverage* of the edge ij is defined as the effective resistance scaled by the weight:

$$\ell_{ij}(\boldsymbol{s}) := w_{ij} \cdot \varrho_{ij}(\boldsymbol{s}) \leq 1.$$

Note that one has $\sum_{ij \in E} \ell_{ij}(\boldsymbol{s}) = n - 1$. This is also known as Foster's theorem [Foster, 1949].

2.1 Relaxed problem formulation

Throughout, we work under the following simplifying assumptions.

Assumption 1. The target edge weights are the same as the model weights, $y_e = w_e$ for all $e \in E$; equivalently, $\boldsymbol{\Gamma}_{\boldsymbol{s}} = \boldsymbol{L}_{\boldsymbol{s}}$ for any $\boldsymbol{s} \in \{0,1\}^m$.

Assumption 2. There exists a connected template graph $S_0 \subseteq E$ such that $s_e = 1$ always for all $e \in S_0$.

Assumption 1 is without loss of generality, while Assumption 2 is a somewhat restrictive assumption that provides a clean way to handle connectivity constraints.

Relaxation to switching probabilities. The key step is to relax the binary variables to lie in the unit cube $\boldsymbol{s} \in [0,1]^m$. This choice has two benefits. From an optimization perspective, the relaxed reconfiguration program becomes much more tractable. Moreover, fractional solutions of the relaxed reconfiguration problem can be viewed as *switching probabilities*, reminiscent of randomized rounding [Raghavan and Tompson, 1987]. Under Assumptions 1 and 2, by primal feasibility, the reconfiguration problem can be written as

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \boldsymbol{d}^\top \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d} \;=: \varphi(\boldsymbol{s}) \\ \text{s.t.} \quad & s_e = 1 \quad \forall e \in S_0, \\ & \|\boldsymbol{s}\|_1 \leq q. \end{aligned}$$

The relaxed congestion objective $\varphi : [0,1]^m \to \mathbb{R}_+$ can be written as

$$\varphi(\boldsymbol{s}) = \boldsymbol{d}^\top \boldsymbol{L}_{\boldsymbol{s}} \boldsymbol{d}, \qquad \boldsymbol{L}_{\boldsymbol{s}} = \boldsymbol{A}^\top \mathsf{diag}(\boldsymbol{w} \circ \boldsymbol{s}) \boldsymbol{A}, \quad (1)$$

and it has notable structural properties, which we now investigate.

2.2 First-order information

We first give a corollary of [Ghosh et al., 2008].

Lemma 1. *Let $R(\boldsymbol{s}) = \sum_{i<j} \varrho_{ij}(\boldsymbol{s}) = \frac{1}{2}\mathbf{1}^\top \boldsymbol{R}_{\boldsymbol{s}} \mathbf{1}$ be the total effective resistance of the graph $G_{\boldsymbol{s}} = (N, E, \boldsymbol{w} \circ \boldsymbol{s})$. Then,*

$$\frac{\partial}{\partial s_{ij}} R(\boldsymbol{s}) = -n w_{ij} \boldsymbol{e}_{ij}^\top \boldsymbol{L}_{\boldsymbol{s}}^{2\dagger} \boldsymbol{e}_{ij}, \qquad ij \in E.$$

In particular, in our setting of fixed demands, we generalize the above result to the following:

Lemma 2 (Congestion gradient)**.** *For a fixed demand $\boldsymbol{d} \perp \mathbf{1}$, the gradient of the relaxed congestion (1) is given elementwise for each edge $e \in E$ as*

$$\frac{\partial}{\partial s_e} \varphi(\boldsymbol{s}) = -w_e \left| \left\langle \boldsymbol{a}_e, \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d} \right\rangle \right|^2 := -w_e \Delta_e^2 \leq 0, \quad (2)$$

where $\Delta_e = \hat{x}_i - \hat{x}_j$ is the voltage difference solution map across edge $e = (i,j) \in E$.

2.3 Second-order information and Hessian

To analyze the performance of our algorithms, we need to understand the spectral content of the Hessian of the congestion function $\varphi(\boldsymbol{s})$ with respect to the switching variables $\boldsymbol{s}$. We will first state and prove the following technical lemma.

Lemma 3 (Cauchy–Schwarz for voltage differences)**.** *Let $\boldsymbol{d} \in \mathbb{R}^n$ be a vector of demands, and let $\boldsymbol{x} = \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d}$ be the voltages induced by the demands under switching strategy $\boldsymbol{s} \in \{0,1\}^m$. Then, for any edge $ij \in E$ and switching strategy $\boldsymbol{s}$, we have*

$$\Delta_{ij}^2 = \left|\boldsymbol{a}_{ij}^\top \boldsymbol{x}\right|^2 \leq \varrho_{ij} \cdot \varphi(\boldsymbol{s}) \qquad \textit{for all } (i,j) \in E. \quad (3)$$

Lemma 3 says that the squared voltage differences in an electrical networks can never exceed the product of the effective resistance of the edge and the total congestion in the network. This is a consequence of the fact that $\boldsymbol{L}_{\boldsymbol{s}}, \boldsymbol{L}_{\boldsymbol{s}}^\dagger \succeq 0$, and the Cauchy-Schwarz inequality applied to the voltages induced by the demands $\boldsymbol{d}$ on the switched graph $G_{\boldsymbol{s}} = (N, E, \boldsymbol{w} \circ \boldsymbol{s})$.

Lemma 4 (Operator norm bound on the Hessian)**.** *Given a switching strategy set $\mathcal{S} \subseteq [0,1]^m$ and demands $\boldsymbol{d}$, let $\boldsymbol{H}(\cdot) : \mathcal{S} \to \mathbb{S}_+^m$ be the Hessian of the congestion function $\varphi(\cdot)$. The operator norm of the Hessian $\boldsymbol{H}(\cdot)$ is bounded as follows:*

$$\|\boldsymbol{H}\|_{\mathsf{op}} \leq L := 2 \max_{s \in \mathcal{S}} \varphi(s, \boldsymbol{d}) = \frac{2\,\|\boldsymbol{d}\|_2^2}{\inf_{s \in \mathcal{S}} \lambda_2(\boldsymbol{L}_{\boldsymbol{s}})} \leq \frac{2\,\|\boldsymbol{d}\|_2^2}{\lambda_2\left(\boldsymbol{L}_{\boldsymbol{s}_0}\right)}.$$

Moreover, if $\|\boldsymbol{d}\|_2 \leq 1$ always, and the template is such that $\lambda_2(\boldsymbol{L}_{\boldsymbol{s}_0}) \geq 1$, then it holds that $\|\boldsymbol{H}\|_{\mathsf{op}} \leq 2$.

The proof appears in the supplementary material.

2.4 Structural properties of congestion

The goal of this section is to outline useful properties about the relaxed congestion (1).

Remark. Set $\mathbf{1}_T \in \{0,1\}^m$ to be the indicator of a spanning tree $T \subseteq E$ of G. In the feasible set

$$\mathcal{D}_T \;:=\; \{\, s \in [0,1]^m : s_e = 1 \text{ for all } e \in T\},$$

the graph remains connected for all $s \in \mathcal{D}_T$ because the tree edges have switching value one.

Furthermore, we have the following useful property, which is reminiscient of [Ghosh et al., 2008].

Lemma 5. *The relaxed congestion (1) is a convex, homogeneous function of degree -1, namely,*

$$\varphi(\boldsymbol{s}) = -\left\langle \nabla\varphi(\boldsymbol{s}), \boldsymbol{s} \right\rangle \qquad \forall \boldsymbol{s} \in \mathcal{D}_T.$$

Theorem 1 (Generalized self–concordance)**.** *Let $T \subseteq E$. For $\boldsymbol{s} \in \mathcal{D}_T$ define $\varphi(\boldsymbol{s}) = \boldsymbol{d}^\top \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d}$ as in (1). Let $\boldsymbol{\varrho}_T := \mathsf{diag}(\boldsymbol{A}\boldsymbol{L}_T^\dagger \boldsymbol{A}^\top) \in \mathbb{R}^m$ denote the vector of effective resistances in the tree T, i.e., $\boldsymbol{\varrho}_T(e) = \boldsymbol{a}_e^\top \boldsymbol{L}_T^\dagger \boldsymbol{a}_e$ where $\boldsymbol{L}_T = \boldsymbol{A}^\top \mathsf{diag}(\boldsymbol{w} \odot \mathbf{1}_T)\boldsymbol{A}$ is the tree Laplacian and $\boldsymbol{L}_T^\dagger$ its pseudoinverse. Then φ is (M,ν)–generalized self–concordant on $\mathcal{D}_T$ with*

$$\nu = 2 \quad \text{and} \quad M \;=\; 3\,||\boldsymbol{w} \odot \boldsymbol{\varrho}_T||_2\,. \tag{4}$$

That is, for all $s \in \mathcal{D}_T$ and all directions $u, w \in \mathbb{R}^m$ one has

$$\left|D^3\varphi(s)[w,u,u]\right| \;\le\; 3\left\|\boldsymbol{w} \odot \boldsymbol{\varrho}_T\right\|_2 \|w\|_2\, \|u\|^2_{\nabla^2\varphi(s)}. \tag{5}$$

In particular, the generalized self–concordance constant M is finite and independent of $\boldsymbol{s}$ on the entire feasible domain $\mathcal{D}_T$.

3 APPROXIMATION SCHEME

As shown in Thm. 1, φ is a generalized self-concordant function in the space of *switching probabilities* $\boldsymbol{s} \in \mathcal{D}_T$. This makes a Frank-Wolfe algorithm a good choice to design a distribution of configurations [Dvurechensky et al., 2023, Pham et al., 2023, Carderera et al., 2024]. In particular, we aim for our output random switching vector to be such that, when we sample edges without replacement (i.e., *draw a configuration*), it is likely to yield a connected subgraph with at most $O(q)$ edges, and to satisfy $d^\top L_s^\dagger d \le \alpha \cdot d^\top L_{\mathbf{1}_m}^\dagger d$, where we desire α to be a small as possible.

3.1 Fast Laplacian solver with switching

Below is a simple corollary of the result in [Kyng, 2017, Thm. 1.2.1.], which proves useful in our subsequent analysis.

Corollary 1.1 ([Kyng, 2017])**.** *Fix a scalar $\delta < 1/n^{100}$ and a switching strategy $\boldsymbol{s}$ with at most q non-zero entries, and let $\boldsymbol{L}_{\boldsymbol{s}} \succeq \mathbf{0}$ be as in Def. 2.1. There exists an algorithm that returns a random Cholesky factor $\boldsymbol{C}_{\boldsymbol{s}}$ such that, with probability at least $1 - O(\delta)$,*

$$\tfrac{1}{2}\boldsymbol{L}_{\boldsymbol{s}} \preceq \boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{C}_{\boldsymbol{s}}^\top \preceq \tfrac{3}{2}\boldsymbol{L}_{\boldsymbol{s}}.$$

This algorithm runs in time $O\left(q \log^2(1/\delta)\log(n)\right)$.

Consequently, given demands $\mathbf{d} \perp \mathbf{1}$ there is an algorithm $\mathsf{Solve}(\boldsymbol{L}_{\boldsymbol{s}}, \boldsymbol{d}, \delta)$ that returns an approximate voltage solution $\hat{\boldsymbol{x}}(\boldsymbol{s})$ such that with probability at least $1 - O(\delta)$,

$$\left|\left|\hat{\boldsymbol{x}}(\boldsymbol{s}) - \boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d}\right|\right|_{\boldsymbol{L}_{\boldsymbol{s}}} \le \epsilon \left|\left|\boldsymbol{L}_{\boldsymbol{s}}^\dagger \boldsymbol{d}\right|\right|_{\boldsymbol{L}_{\boldsymbol{s}}},$$

where we define the norm $||\boldsymbol{x}||_{\boldsymbol{L}} := \sqrt{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathbb{R}^n$. The algorithm runs in $O(q\log^2(1/\delta)\log(n)\log(1/\epsilon))$.

Access to a near-linear time Laplacian solver immediately provides a simple, near-linear time algorithm for *simultaneously* determining the gradient of the congestion criterion $\nabla\varphi \in \mathbb{R}^m$, and pairwise voltage differences across all edges, $\Delta \in \mathbb{R}^m$, which we present in Section 4.1.

3.2 Switching strategies and edge budget

A useful reconfiguration problem is to select a subset of edges $S \subseteq E$ such that the graph remains connected, i.e., the algebraic connectivity $\lambda_2(\boldsymbol{L}_S) > 0$, the number of edges in S is at most K, and the congestion (graph energy) does not grow too much. There are two such ways we can model this constraint set.

Throughout, we treat all switching variables as the parameters of independent Bernoulli random variables, i.e., we assume that each edge $e \in E$ is switched with probability $s_e \in [s_{\mathsf{min}}, 1]$, with the lower bound to be defined later, and the switching decisions are independent. We want to show that $\boldsymbol{s}$ is a good *approximately optimal switching strategy*, in the following sense: if we sample a *random* configuration $\tilde{\boldsymbol{s}}$ with $\tilde{\boldsymbol{s}}_e \sim \mathsf{Ber}(\boldsymbol{s}_e)$, independently for each edge $e \in E$, then the congestion of the resulting graph does not grow too much. We will define this notion precisely in Theorem 2.

A natural constraint is to impose a budget on the number of edges that can be switched, that is, we require that $||\boldsymbol{s}||_1 \le q$, where q is the edge budget. This is a natural constraint in many applications, such as electrical networks, where we may not be able to switch all edges due to physical limitations or operational costs.

4 MAIN RESULTS

4.1 Fast gradients

We now give an algorithm that updates the congestion gradient and the voltage differences for a given switching strategy $\boldsymbol{s}$ and demands $\boldsymbol{d}$ in nearly-linear time.

Lemma 6 (Approximate gradient)**.** *Let $\boldsymbol{s} \in [0,1]^m$ be a switching strategy, and let $\boldsymbol{d} \perp \mathbf{1}$ be demands. Fix a $\delta \in (0,1)$, and let $\hat{\boldsymbol{x}}(\boldsymbol{s}) \leftarrow \mathsf{Solve}(\boldsymbol{L}_{\boldsymbol{s}}, \boldsymbol{d}, \delta)$ be an ϵ-approximate solution to $\boldsymbol{L}_{\boldsymbol{s}}\boldsymbol{x} = \boldsymbol{d}$ as in Corr. 1.1. Then, Algorithm 1 computes a random congestion gradient $\widehat{\nabla}\varphi(\boldsymbol{s})$ in time $O(m\log^c(n)\log(1/\epsilon))$, and, there exists a constant C such that, with probability at least $1 - O(\delta)$,*

$$\|\widehat{\nabla}\varphi(\boldsymbol{s}) - \nabla\varphi(\boldsymbol{s})\|_{\nabla^2\varphi(\boldsymbol{s})^\dagger} \le C\epsilon\,||\nabla\varphi(\boldsymbol{s})||_{\nabla^2\varphi(\boldsymbol{s})^\dagger}\,.$$

The proof appears in the supplementary material.

Input: $\boldsymbol{s}$, $\boldsymbol{d}$, E, $\boldsymbol{w}$, δ.
Output: Approximations $\hat{\nabla}\varphi(\boldsymbol{s})$ and $\hat{\Delta}(\boldsymbol{s})$.

```
function ApproxDiff(s, d, w, E, ε, δ)
    L_s ← Σ_{ij∈supp(s)} E_ij w_ij                    // O(m)
    ∇ ← 0_m, Δ ← 0_m
    // -- Solve the Laplacian system --
    x̂(s) ← Solve(L_s, d, δ)                          // Õ(m)
    // -- compute the gradient entries --
    for e = (i, j) ∈ E do
        // -- compute voltage diff. --
        Δ_e ← x̂_i(s) − x̂_j(s)                        // O(1)
        // -- compute gradient entry --
        ∇_e ← −w_ij Δ_e^2                            // O(1)
    end
    return ∇, Δ
end
```

Algorithm 1: Fast gradient and voltage difference computation for network reconfiguration.

Input: $\boldsymbol{s}_0$, $\boldsymbol{d} \perp \mathbf{1}$, E, $\boldsymbol{w}$, $q > n-1$, T, ϵ, δ.
Output: Random integral switching strategy $\tilde{\boldsymbol{s}}$.

```
function RandReconfig(T, K, E, w, d, ε, δ)
    L_{s_0} ← Σ_{ij∈supp(s_0)} E_ij w_ij              // O(m)
    for t = 0, 1, ..., to T − 1 do
        η_t ← 2/(t+2)                                // Set step size
        // -- compute the congestion gradient --
        ∇φ(s_t), Δ(s_t) ←
          ApproxDiff(s_t, d, w, E, ε, δ)
        // -- find a vertex --
        v_t ← arg min_{v∈[0,1]^m, ||v||_1 ≤ q} ⟨∇φ(s_t), v⟩
        // -- update convex combination --
        s_{t+1} ← (1 − η_t) s_t + η_t v_t
    end
    // -- round the fractional solution --
    s̃ ← Round(s_T, δ)                                // O(m)
    // -- solve for the voltages --
    x̃ ← Solve(L_s̃, d, δ)
    return s_⋆, x̃
end
```

Algorithm 2: Montonic Frank-Wolfe with randomized rounding for network reconfiguration.

Remark. In practice, Algorithm 1 should be supplied a persistent copy of $\nabla, \Delta \in \mathbb{R}^m$ for in-place storage of the congestion gradient and the voltage differences.

4.2 Randomized rounding

After the Frank–Wolfe phase of Alg. 2, we obtain a fractional switching vector $\boldsymbol{s}_t \in [0,1]^m$ satisfying $\|\boldsymbol{s}_t\|_1 \le q$ and $(\boldsymbol{s}_t)_e = 1$ for all backbone edges $e \in T$. To convert $\boldsymbol{s}_t$ into an integral solution while preserving the edge weights $\boldsymbol{w}$, we adopt the following procedure. Fix a failure probability $\delta \in (0,1)$ and choose a baseline probability

$$p_{\min} = \frac{C \log(n/\delta)}{n}$$

Input: $\boldsymbol{s} \in [0,1]^m$, $p_{\min}$, δ.
Output: Random integral $\tilde{\boldsymbol{s}}$.

```
function Round(s, p_min, δ)
    s̃ ← 0_m
    // -- round the fractional solution --
    for e = (i, j) ∈ E do
        s̄_e ← (6)
        (s̃)_e ~ Bern((s_T)_ij)                       // O(1)
    end
    return s̃
end
```

Algorithm 3: Randomized rounding scheme

with a universal constant $C > 0$. Define

$$\bar{s}_e = \begin{cases} \max\{(\boldsymbol{s}_t)_e,\, p_{\min}\} & e \in E \setminus T \\ 1 & e \in T. \end{cases} \tag{6}$$

Sample a random vector $\tilde{\boldsymbol{s}} \in \{0,1\}^m$ by including each edge e independently with probability $\bar{s}_e$. If $\|\tilde{\boldsymbol{s}}\|_1 > q$, remove edges in increasing order of $(\boldsymbol{s}_t)_e$ until at most q edges remain; if $\|\tilde{\boldsymbol{s}}\|_1 < q$, add edges in decreasing order of $(\boldsymbol{s}_t)_e$ until q edges are selected. Denote by $\boldsymbol{L}_{\tilde{\boldsymbol{s}}} = \sum_e \tilde{s}_e w_e \boldsymbol{a}_e \boldsymbol{a}_e^\top$ the Laplacian of the sampled configuration.

Theorem 2 (Rounding preserves congestion)**.** *Let $\boldsymbol{s}_t \in [0,1]^m$ satisfy $\|\boldsymbol{s}_t\|_1 \le q$ and $(\boldsymbol{s}_t)_e = 1$ for all $e \in T$, and let $\tilde{\boldsymbol{s}}$ be obtained by the independent Bernoulli rounding described above. Then there exist universal constants $C_1, C_2 > 0$ such that, for any $\delta \in (0,1)$ and $p_{\min} = C_1 \log(n/\delta)/n$, the following holds with probability at least $1-\delta$:*

$$(1-\epsilon)\, \boldsymbol{L}_{\boldsymbol{s}_t} \preccurlyeq \boldsymbol{L}_{\tilde{\boldsymbol{s}}} \preccurlyeq (1+\epsilon)\, \boldsymbol{L}_{\boldsymbol{s}_t}, \tag{7}$$

where $\epsilon = C_2 \sqrt{\frac{\log(n/\delta)}{n\, p_{\min}}}$. Consequently, for any demand vector $\boldsymbol{d} \perp \mathbf{1}$, one has

$$(1-\epsilon)\, \varphi(\boldsymbol{s}_t) \le \varphi(\tilde{\boldsymbol{s}}) \le (1+\epsilon)\, \varphi(\boldsymbol{s}_t).$$

Choosing $p_{\min}$ so that $\epsilon \le \alpha$ yields an integral configuration $\tilde{\boldsymbol{s}}$ with at most q switched edges and

$$\varphi(\tilde{\boldsymbol{s}}) \le (1+2\alpha)\, \varphi(\boldsymbol{s}_\star),$$

where $\boldsymbol{s}_\star$ is a minimizer of the relaxed problem.

Remark. The baseline probability $p_{\min}$ ensures that each edge has a non-negligible chance of being selected, which yields a concentration bound via a matrix Bernstein inequality. The truncation/augmentation step enforces the budget constraint $\|\tilde{\boldsymbol{s}}\|_1 \le q$ without changing the high-probability guarantee, since the edges with larger $(\boldsymbol{s}_t)_e$ are more likely to be retained. The proof of Theorem 2 is deferred to the supplementary material.

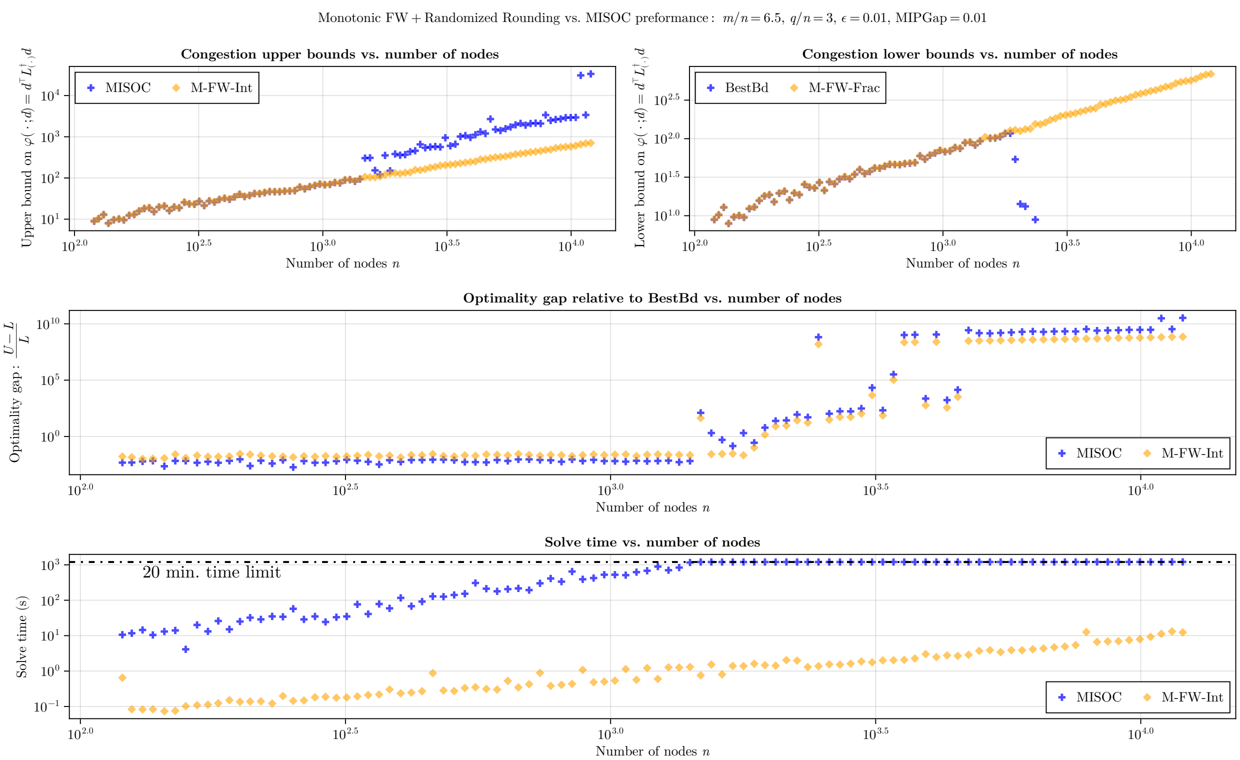

Figure 1: Comparison of Alg. 2 and (8) using Gurobi's MISOCP solver vs. number of nodes in a test graph. The bottom sub-figure depicts an orders-of-magnitude speedup.

4.3 Approximation certificate

In this section, we study an explicit, computable optimality certificate for a family of switch distributions.

Theorem 3. *Let $\boldsymbol{s}_t \in [0,1]^m$ be the t-th iterate of the switching probabilities in the Frank-Wolfe algorithm. Set*

$$\boldsymbol{v}_t \in \operatorname*{arg\,min}_{\boldsymbol{v} \in S} \langle \nabla\varphi(\boldsymbol{s}_t), \boldsymbol{v} \rangle ,$$

where $S \subseteq [0,1]^m$ is a convex set of almost surely connected switching distributions. If

$$\langle \nabla\varphi(s_t), s_t - v_t \rangle \le \tau \cdot \varphi(s_t),$$

then

$$\varphi(s_t) - \varphi(s_\star) \le \frac{\tau}{1-\tau}\varphi(s_\star).$$

In particular, for the top-q set with backbone, we have the following result.

Corollary 3.1 (α-competitive approximation certificate)**.** *Let $s_0 \in \{0,1\}^m$ be a connected* backbone*, and consider the feasible region of connected configurations*

$$S_q(s_0) := \{\boldsymbol{s} \in [0,1]^m : ||\boldsymbol{s}||_1 \le q, \\ s_e = 1 \ \forall e \in \mathsf{supp}(\boldsymbol{s}_0)\}.$$

Fix a parameter $\alpha \in (0,1)$. For any connected configuration $s \in S_q(s_0)$, define the linear minimization oracle $v_\star(s) := \mathsf{arg\ min}_{v \in S_q(s_0)} \langle \nabla\varphi(s), v \rangle$, where, for each $e \in E$,

$$(v_\star(s))_e := \begin{cases} 1 & \text{if } e \in \mathsf{supp}(s_0) \\ 1 & \text{if } e \in \mathsf{Top}_{q-n+1}\{(\mathbf{I}_m - \boldsymbol{S}_0)\nabla\varphi(s)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{S}_0 := \mathsf{diag}(\boldsymbol{s}_0)$. If, for some $\tau > 0$, we have

$$\langle \nabla\varphi(s), s - v_\star(s) \rangle \le \tau\varphi(s), \quad \tau := \frac{\alpha}{1+\alpha}, \quad \alpha \in (0,1),$$

then s is an α-approximate solution to the network reconfiguration problem. That is, for any potentially non-unique minimizer $s_\star \in \mathsf{arg\ min}_{s \in S_q(s_0)} \varphi(s)$, we have

$$\varphi(s) \le (1+\alpha)\,\varphi(s_\star), \qquad \alpha = \frac{\tau}{1-\tau} < 1.$$

Remark. When computing the convergence certificate with the approximated gradient of Lemma 6, the Frank–Wolfe gap estimated by the approximate Laplacian solver differs from the true gap by at most a factor $1 + O(\epsilon)$.

5 NUMERICAL RESULTS

5.1 Baseline formulation

The network reconfiguration studied in the present paper can be formulated as a mixed-integer second-order cone program (MISOCP). This structure is compatible with some commercial solvers, like Gurobi, to ostensibly solve the problem efficiently. Introducing a change of variable, we obtain the program

$$\min_{\boldsymbol{u},\boldsymbol{s},\boldsymbol{f}} \quad 2\sum_{e\in E} w_e^{-1} u_e \tag{8a}$$
$$\text{s.t.} \quad \boldsymbol{u} \geq \boldsymbol{0},\ \boldsymbol{s} \in \{0,1\}^m,\ \boldsymbol{f} \in \mathbb{R}^m, \tag{8b}$$
$$u_e s_e \geq \frac{1}{2} f_e^2, \quad \forall e \in E, \tag{8c}$$
$$s_e = 1 \quad \forall e \in S_0, \tag{8d}$$
$$||\boldsymbol{s}||_1 \leq q, \tag{8e}$$
$$\boldsymbol{A}^\top \boldsymbol{f} = \boldsymbol{d}. \tag{8f}$$

The program (8) is a mixed-integer second-order cone program (MISOCP). The objective function is a linearization of the congestion objective $\varphi(s)$, where u_e is an upper bound on the congestion of edge e. The constraints (8c) are second-order cone constraints (conditional on s_e) that enforce $u_e \geq \frac{1}{2}\frac{f_e^2}{s_e}$ when $s_e = 1$ and $u_e \geq 0$ when $s_e = 0$. The constraint (8e) enforces the power constraint. Finally, the constraint $A^\top f = d$ enforces flow conservation.

5.2 Numerical implementation of the approximation scheme

Algorithm 1 and Algorithm 2 were implemented in the Julia programming language, leveraging the fast Laplacian solver available in `Laplacians.jl`, presented in [Gao et al., 2023]. We generated random graphs over a large range of network sizes, using a different seed for each size. The experiments were run on a conventional laptop with a Ryzen 4750U chipset.

Similarly, as a baseline comparison, we implement the MISOCP formulation (8) in the JuMP algebraic modeling language [Dunning et al., 2017], which was then fed into Gurobi's MISOCP solver. As shown in Fig. 1, our implementation of Algorithm 1, based on the monotonic Frank-Wolfe algorithm with randomized rounding, yields up to a 1000× improvement in the total solve time from Gurobi. In particular, the upper and lower bounds on the congestion are close across both methods. Moreover, when Gurobi fails to declare convergence within a stipulated 20 minute time limit, the FW procedure achieves a superior optimality gap relative to the best incumbent lower bound determined by Gurobi.

6 CONCLUSION

We presented an approximation scheme to efficiently solve network reconfiguration problems. We proposed a particular instance of such an algorithm, RandReconfig, based on a conditional gradient procedure with randomized rounding. We demonstrated our method outperforming a commercial MINLP solver by orders of magnitude on a relevant problem.

Future work Inclusion of line flow limits (edge capacities) is an essential component for future work, in addition to allowing some entries of the demand vector to be continuous decision variables. The MISOCP formulation of the network reconfiguration problem (8) can serve an additional purpose. In addition to providing the benchmark, the random configurations generated by the proposed algorithm can be used as a warm start for the MISOCP.

References

[Allen-Zhu et al., 2021] Allen-Zhu, Z., Li, Y., Singh, A., and Wang, Y. (2021). Near-optimal discrete optimization for experimental design: a regret minimization approach. *Mathematical Programming*, 186(1):439–478.

[Anderson et al., 2014] Anderson, D. G., Gu, M., and Melgaard, C. (2014). An efficient algorithm for unweighted spectral graph sparsification.

[Babaeinejadsarookolaee et al., 2023] Babaeinejadsarookolaee, S., Park, B., Lesieutre, B., and DeMarco, C. L. (2023). Transmission congestion management via node-breaker topology control. *IEEE Systems Journal*, 17(3):3413–3424.

[Boumal and Cheng, 2014] Boumal, N. and Cheng, X. (2014). Concentration of the Kirchhoff index for ErdHos–Rényi graphs. *Systems & Control Letters*, 74:74–80.

[Braun et al., 2025] Braun, G., Carderera, A., Combettes, C. W., Hassani, H., Karbasi, A., Mokhtari, A., and Pokutta, S. (2025). Conditional Gradient Methods. arXiv:2211.14103 [math].

[Brown et al., 2024] Brown, A., Laddha, A., and Singh, M. (2024). Fast algorithms for maximizing the minimum eigenvalue in fixed dimension. *Operations Research Letters*, 57:107186.

[Carderera et al., 2024] Carderera, A., Besancon, M., and Pokutta, S. (2024). Scalable Frank-Wolfe on generalized self-concordant functions via simple steps. *SIAM Journal on Optimization*, 34(3):2231–2258.

[Charalambides and Hero, 2023] Charalambides, N. and Hero, A. O. (2023). Graph sparsification by approximate matrix multiplication. In *2023 IEEE Statistical Signal Processing Workshop (SSP)*, pages 180–184.

[Chen et al., 2022] Chen, L., Kyng, R., Liu, Y. P., Peng, R., Gutenberg, M. P., and Sachdeva, S. (2022). Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623.

[Chen et al., 2021] Chen, Y., Chi, Y., Fan, J., and Ma, C. (2021). Spectral methods for data science: A statistical perspective. *Foundations and Trends in Machine Learning*, 14(5):566–806. Publisher: Now Publishers, Inc.

[Dunning et al., 2017] Dunning, I., Huchette, J., and Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.

[Dvurechensky et al., 2023] Dvurechensky, P., Safin, K., Shtern, S., and Staudigl, M. (2023). Generalized self-concordant analysis of Frank–Wolfe algorithms. *Mathematical Programming*, 198(1):255–323.

[Foster, 1949] Foster, R. M. (1949). The average impedance of an electrical network. *Contributions to Applied Mechanics (Reissner Anniversary Volume)*, 333.

[Fukushima, 1984] Fukushima, M. (1984). A modified Frank-Wolfe algorithm for solving the traffic assignment problem. *Transportation Research Part B: Methodological*, 18(2):169–177.

[Gao et al., 2023] Gao, Y., Kyng, R., and Spielman, D. A. (2023). Robust and practical solution of Laplacian equations by approximate elimination. arXiv:2303.00709 [math].

[Ghosh and Boyd, 2006] Ghosh, A. and Boyd, S. (2006). Growing well-connected graphs. In *45th IEEE Conference on Decision and Control*, pages 6605–6611.

[Ghosh et al., 2008] Ghosh, A., Boyd, S., and Saberi, A. (2008). Minimizing effective resistance of a graph. *SIAM Review*, 50(1):37–66.

[Haider et al., 2025] Haider, R., Annaswamy, A., Dey, B., and Chakraborty, A. (2025). SiPhyR: An end-to-end learning-based optimization framework for dynamic grid reconfiguration. *IEEE Transactions on Smart Grid*, 16(2):1248–1260.

[Kooij and Achterberg, 2023] Kooij, R. E. and Achterberg, M. A. (2023). Minimizing the effective graph resistance by adding links is NP-hard. *Operations Research Letters*, 51(6):601–604.

[Kyng, 2017] Kyng, R. (2017). Approximate Gaussian elimination. Ph.D. dissertation, Yale University.

[Kyng and Sachdeva, 2016a] Kyng, R. and Sachdeva, S. (2016a). Approximate Gaussian elimination for Laplacians - Fast, sparse, and simple. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 573–582.

[Kyng and Sachdeva, 2016b] Kyng, R. and Sachdeva, S. (2016b). Approximate Gaussian elimination for Laplacians: Fast, sparse, and simple.

[Lau and Zhou, 2020] Lau, L. C. and Zhou, H. (2020). A spectral approach to network design. In *52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 826–839.

[Lehmann et al., 2014] Lehmann, K., Grastien, A., and Hentenryck, P. V. (2014). The complexity of DC-switching problems.

[Li et al., 2020] Li, H., Patterson, S., Yi, Y., and Zhang, Z. (2020). Maximizing the number of spanning trees in a connected graph. *IEEE Transactions on Information Theory*, 66(2):1248–1260.

[Moondra et al., 2024] Moondra, J., Lotshaw, P. C., Mohler, G., and Gupta, S. (2024). Promise of graph sparsification and decomposition for noise reduction in qaoa: Analysis for trapped-ion compilations.

[Nagarajan et al., 2015] Nagarajan, H., Rathinam, S., and Darbha, S. (2015). On maximizing algebraic connectivity of networks for various engineering applications. In *European Control Conference (ECC)*, pages 1626–1632.

[Oliveira, 2010] Oliveira, R. I. (2010). Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges. arXiv:0911.0600 [math].

[Oliveira, 2013] Oliveira, R. I. (2013). The lower tail of random quadratic forms, with applications to ordinary least squares and restricted eigenvalue properties.

[Pham et al., 2023] Pham, C. B., Griggs, W., and Saunderson, J. (2023). A scalable Frank-Wolfe-based algorithm for the max-cut SDP. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27822–27839. PMLR.

[Pokutta, 2024] Pokutta, S. (2024). The Frank-Wolfe Algorithm: A Short Introduction. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 126(1):3–35.

[Raghavan and Tompson, 1987] Raghavan, P. and Tompson, C. D. (1987). Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374.

[Ru et al., 2025] Ru, X., Xia, W., and Cao, M. (2025). Maximizing the smallest eigenvalue of grounded Laplacian matrices via edge addition. *Automatica*, 176:112238.

[Sadhanala et al., 2016] Sadhanala, V., Wang, Y.-X., and Tibshirani, R. (2016). Graph sparsification approaches for Laplacian smoothing. In Gretton, A. and Robert, C. C., editors, *19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1250–1259, Cadiz, Spain. PMLR.

[Somisetty et al., 2024a] Somisetty, N., Nagarajan, H., and Darbha, S. (2024a). Optimal robust network design: Formulations and algorithms for maximizing algebraic connectivity. *IEEE Transactions on Control of Network Systems*, 12(1):918–929.

[Somisetty et al., 2024b] Somisetty, N., Nagarajan, H., and Darbha, S. (2024b). Spectral graph theoretic methods for enhancing network robustness in robot localization. In *63rd IEEE Conference on Decision and Control (CDC)*, pages 8811–8818.

[Spielman and Srivastava, 2011] Spielman, D. A. and Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926.

[Spielman and Teng, 2004] Spielman, D. A. and Teng, S.-H. (2004). Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *36th Annual ACM Symposium on Theory of Computing*, STOC '04, page 81–90.

[Spielman and Teng, 2012] Spielman, D. A. and Teng, S.-H. (2012). Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. arXiv:cs/0607105.

[Stott et al., 2009] Stott, B., Jardim, J., and Alsac, O. (2009). Dc power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300.

[Teng, 2010] Teng, S.-H. (2010). The Laplacian paradigm: Emerging algorithms for massive graphs. In *Theory and Applications of Models of Computation*, pages 2–14, Berlin, Heidelberg. Springer.

[Tropp, 2011a] Tropp, J. A. (2011a). Freedman's inequality for matrix martingales.

[Tropp, 2011b] Tropp, J. A. (2011b). User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434.

[van der Sar et al., 2025] van der Sar, E., Zocca, A., and Bhulai, S. (2025). Optimizing power grid topologies with reinforcement learning: A survey of methods and challenges.

[Wang et al., 2014] Wang, X., Pournaras, E., Kooij, R. E., and Van Mieghem, P. (2014). Improving robustness of complex networks via the effective graph resistance. *The European Physical Journal B*, 87(9):221.

[Yurtsever et al., 2021] Yurtsever, A., Tropp, J. A., Fercoq, O., Udell, M., and Cevher, V. (2021). Scalable Semidefinite Programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200.

[Zhou et al., 2025a] Zhou, X., Wang, R., Li, W., and Zhang, Z. (2025a). Maximizing the smallest eigenvalue of grounded Laplacian matrix. *Journal of Global Optimization*, 91(4):807–828.

[Zhou et al., 2025b] Zhou, X., Zehmakan, A. N., and Zhang, Z. (2025b). Efficient algorithms for minimizing the Kirchhoff index via adding edges. *IEEE Transactions on Knowledge and Data Engineering*, 37(6):3342–3355.