Approximation Algorithms for ℓ_p-Shortest Path and ℓ_p-Group Steiner Tree

Yury Makarychev
Toyota Technological Institute at Chicago, IL, USA

Max Ovsiankin
Toyota Technological Institute at Chicago, IL, USA

Erasmo Tani
University of Chicago, IL, USA

Abstract

We present polylogarithmic approximation algorithms for variants of the Shortest Path, Group Steiner Tree, and Group ATSP problems with vector costs. In these problems, each edge e has a vector cost $c_e \in \mathbb{R}^{\ell}_{\geq 0}$. For a feasible solution – a path, subtree, or tour (respectively) – we find the total vector cost of all the edges in the solution and then compute the ℓ_p-norm of the obtained cost vector (we assume that $p \geq 1$ is an integer). Our algorithms for series-parallel graphs run in polynomial time and those for arbitrary graphs run in quasi-polynomial time.

To obtain our results, we introduce and use new flow-based Sum-of-Squares relaxations. We also obtain a number of hardness results.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Shortest paths; Theory of computation → Approximation algorithms analysis; Theory of computation → Routing and network design problems

Keywords and phrases Shortest Path, Asymmetric Group Steiner Tree, Sum-of-Squares

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.111

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `http://arxiv.org/abs/2404.17669` [24]

Funding *Yury Makarychev*: Partially supported by NSF Awards CCF-1955173, CCF-1934843, and ECCS-2216899.
Max Ovsiankin: Partially supported by the Institute for Data, Econometrics, Algorithms, and Learning (IDEAL) with NSF Grant ECCS-2216899.
Erasmo Tani: Partially supported by the Institute for Data, Econometrics, Algorithms, and Learning (IDEAL) with NSF Grant ECCS-2216912.

Acknowledgements We would like to thank the anonymous reviewers for providing references to relevant prior work and for comments that improved the presentation of this paper.

1 Introduction

In this work, we study robust versions of network design problems. In the ℓ_p-Shortest Path problem, we are given $p \geq 1$, a graph $G = (V, E)$ with vector-valued edge costs $c_e \in \mathbb{R}^{\ell}_{\geq 0}$, and two vertices s and t; the goal is to find a path P from s to t in G that minimizes the following cost:

$$\mathrm{cost}_{\ell_p}(P) = \left\| \sum_{e \in P} c_e \right\|_p .$$

This problem is a natural generalization of the classical shortest path problem, but surprisingly has not received much attention till recently. The problem has been studied for $p = \infty$ under the name Robust Shortest Path. Aissi, Bazgan and Vanderpooten [1] used dynamic programming to obtain a fully polynomial-time approximation scheme for the case when the number of coordinates ℓ is a constant and $p = \infty$ (this result generalizes to other p). Kasperski and Zieliński [18] proved that ℓ_∞-Shortest Path is hard to approximate within $\log^{1-\varepsilon} \ell$ for all $\varepsilon > 0$ unless NP $\subseteq$ DTIME($n^{\mathrm{polylog}\, n}$). More recently, the same authors [21] gave an $O(\sqrt{n \log \ell / \log\log \ell})$-approximation to the problem by rounding a flow-based linear programming relaxation and proved that their LP has the integrality gap of $\Omega(\sqrt{n})$. In a recent breakthrough, Li, Xu, and Zhang [23] gave an $O(\log n \log \ell)$-approximation algorithm for ℓ_∞-Shortest Path with running time quasi-polynomial in the size of the input instance. In particular, their algorithm is the first polylogarithmic approximation known to date. They also show that the same approximation guarantees can be obtained in polynomial time for graphs of bounded treewidth, and they give a polynomial time $O(d \log \ell)$ approximation algorithm for series-parallel graphs, where d is the depth (order) of the series-parallel decomposition of the input graph. No results were known for any other exponents $p \in (1, \infty)$. However, it is trivial to get an $\ell^{1-1/p}$-approximation by solving standard shortest path with edge costs $\|c_e\|_1$.

Introducing vector-valued costs to the graph's edges allows this model to capture a number of different applications. First, it allows us to describe a situation in which different parties, each corresponding to a different coordinate of the cost vectors, incur different cost when an edge is added to the solution. In this interpretation, as $p \to \infty$, the problem will increasingly favor paths in which every party simultaneously incurs small cost. Alternatively, each coordinate of the cost vectors may represent the cost incurred in terms of a different resource. This model would then allow one to balance minimizing the total amount of resources spent and ensuring that no single resource is depleted. Furthermore, one can think of this problem as providing an avenue for modeling robustness of a solution in the presence of uncertainty. Each coordinate would then represent the cost incurred by adding an edge in a distinct possible scenario, and the value of the ℓ_p-Shortest Path problem would amount to a trade-off between average and worst-case cost among all scenarios. Finally, this problem generalizes congestion minimization in directed graphs (a fact that we prove in Section 8 of the full version of the paper [24]).

Our results for the ℓ_p-Shortest Path problem. In this paper, we introduce a natural flow-based sum-of-squares (SoS) relaxation for ℓ_p-Shortest Path (Section 3) and present approximation algorithms for all integer $p \geq 1$.

First, we give an $O(pd^{1-1/p})$-approximation algorithm for the problem running in $n^{O(p)}$ time for series-parallel graphs of depth/order d (Section 5). We do this by considering a natural rounding algorithm for the SoS relaxation. We prove the following theorem:

▶ **Theorem 1.** *There exists an approximation algorithm for the ℓ_p-Shortest Path problem in series-parallel graphs that, given a series-parallel graph G of order/depth d and parameters $p \in \mathbb{Z}_{\geq 1}$ and $\varepsilon \in (0,1)$, finds a $(1+\varepsilon)\mathscr{B}_d(p)^{1/p} = O(pd^{1-1/p})$ approximation in time $m^{O(p)}/\varepsilon^{O(1)}$ (which is polynomial time when p and ε are fixed). Here, $\mathscr{B}_d(p)$ is the p^{th} d-dimensional Bell number.*[1]

[1] We provide a review of Bell numbers in the preliminaries.

For graphs of small series-parallel order/depth $d \leq \log^* p$, the approximation factor is $\mathscr{B}_d^{1/p} \leq O(p/\log^{(d)} p)$. Remarkably, in a complementary analysis given in Section 8 of the full version of the paper, we show that the approximation factor $\mathscr{B}_d(p)^{1/p}$ is tight for our rounding scheme. In all algorithms, we assume that ℓ is at most polynomial in n (if not, the running times will also depend on ℓ).

Then, we give a $O(p\log^{1-1/d} n)$-approximation for arbitrary graphs (Section 6 of the full version of the paper), obtaining the following theorem:

▶ **Theorem 2.** *There exists an approximation algorithm for the ℓ_p-Shortest Path problem in arbitrary graphs that, given a graph G and parameters $p \in \mathbb{Z}_{\geq 1}$ and $c \in (0, 1/2)$, finds a $cp\log^{1-1/p} n$ approximation in time $m^{ce^{O(1/c)}\log n} = m^{O_c(\log n)}$.*

Note that when $p = \log \ell$, this result yields an $O(c \log n \log \ell)$ approximation. This is very similar to the approximation guarantee of $O(\log n \log k)$ by Li, Xu, and Zhang, except that in our algorithm c can be an arbitrarily small constant.

▶ Remark 3. As previously discussed, an $\ell^{1-1/p}$-approximation is trivial to achieve in polynomial time by solving standard shortest path with edge costs $\|c_e\|_1$. On the other hand, for each fixed ℓ' there is a polynomial-time approximation scheme (PTAS) for ℓ_p-Shortest Path in ℓ' dimensions that runs in time $n^{O(\ell')}$. We now explain how to combine these two results. Fix $\delta \in (1/\ell, 1/2)$. Divide the coordinates of the cost vectors $c_e \in \mathbb{R}^\ell$ into $\ell' = \lceil 1/\delta \rceil$ groups, each of size at most $k = \lceil \delta\ell \rceil$ and then add up the coordinates in each group. For each cost vector $c_e \in \mathbb{R}^\ell$, we obtain a new vector $c'_e \in \mathbb{R}^{\ell'}$. Costs c'_e approximate costs c_e within a factor of $k^{1-1/p}$ in the following sense: for every path P,

$$\Big\|\sum_{e\in P} c_e\Big\|_p \leq \Big\|\sum_{e\in P} c'_e\Big\|_p \leq k^{1-1/p}\Big\|\sum_{e\in P} c_e\Big\|_p. \tag{1}$$

Using the PTAS, we solve the problem with costs c'_e and by (1) get a $(1+\varepsilon)k^{1-1/p}$ approximation to the original problem. We conclude that there exists an approximation algorithm that finds an $O((\delta\ell)^{1-1/p})$ approximation in time $n^{O(1/\delta)}$ (for every $\delta \in (1/\ell, 1/2)$).

In the course of analyzing our algorithms, we prove a new majorization inequality for pseudo-expectations (see Section 4) generalizing pseudo-expectation Lyapunov's and Hölder inequality (for the latter see [2, arXiv version]). We believe this result to be of independent interest.

Hardness results. We also complement the analysis above with several hardness results for ℓ_p-Shortest Path. First, in Section 8.1 of the full version of the paper, we give a reduction showing that the problem of congestion minimization can be reduced to the ℓ_∞-Shortest Path problem. This simultaneously speaks to the broad applicability of ℓ_p-Shortest Path and implies hardness for the ℓ_∞-version of the problem, following a result of Chuzhoy and Khanna [9].

▶ **Theorem 4.** *The ℓ_∞-Shortest Path problem is hard to approximate within an $\Omega(\log n/\log\log n)$-factor unless* $\mathrm{NP} \subseteq \mathrm{ZPTIME}(n^{\log\log n})$.

This theorem slightly strengthens the $\Omega(\log^{1-\varepsilon} \ell)$-hardness of approximation result by Kasperski and Zieliński [18]. Finally, in Section 8.2 of the full version of the paper, we show that allowing the entries of the cost vectors to be negative makes the problem substantially harder. We do this by giving a reduction from the Closest Vector problem in lattices to this (potentially negative costs) version of the ℓ_p-Shortest Path problem. Below, Ω_p hides a constant depending on p.

▶ **Theorem 5.** *For every $p \in [1, \infty]$, it is* NP*-Hard to approximate the ℓ_p-Shortest Path problem allowing negative edge costs within a factor of $n^{\Omega_p(1/\log\log n)}$.*

▶ Remark 6. The requirement that all the coordinates of cost vectors c_e are non-negative can be slightly relaxed when $p = 2$. For our algorithms to work, it is sufficient that all pairwise inner products of cost vectors are non-negative. That is, instead of requiring that the Gram matrix of cost vectors is completely positive, we can only require that it is doubly non-negative.

From shortest path to network design. In network design problems, one is given a graph $G = (V, E)$ with non-negative edge costs $c_e \geq 0$, and wishes to find a subgraph $F = (V_F, E_F)$ that minimizes the total cost $\sum_{e \in E_F} c_e$ subject to some connectivity constraints. By varying the set of allowed subgraphs F, this paradigm encapsulates many central and well-studied network design problems, including the Survivable Network Design, Steiner Forest, Steiner Tree, and Minimum Spanning Tree. In this paper we explore two network design problems, Group Steiner Tree and Asymetric Traveling Salesperson (ATSP). We first recall the Group Steiner Tree problem.

▶ **Problem 7** (Group Steiner Tree)**.** *Given a weighted undirected graph $G = (V, E, c)$, as well as k subsets $R_1, \ldots, R_k$ of V, find a minimum-cost subtree T of G containing at least one vertex from each R_i.*

We then introduce an analogous group variant of ATSP:

▶ **Problem 8** (Group ATSP)**.** *Given a weighted directed graph $G = (V, E, c)$ and a collection of subsets $R_1, \ldots, R_k$ of V, find a minimum-cost tour that visits at least one vertex in each R_i.*

As in the case of Shortest Path, it is natural to ask whether we can approximately solve ℓ_p versions of other network design problems efficiently. Prior work has been done in this area. Hamacher and Ruhe [15] studied ℓ_∞-Minimum Spanning Tree, and proved that it is NP-complete. Following that, the complexity of the problem has been nearly completely settled: Chekuri, Vondrák, and Zenklusen [8] presented an $O(\log \ell / \log\log \ell)$-approximation algorithm, while Kasperski and Zielinski [19] (also see [20], Table 1) proved an $\Omega(\log^{1-\varepsilon} \ell)$-hardness of approximation for every $\varepsilon > 0$, unless all problems in NP can be solved in quasi-polynomial time. Laddha, Singh and Vempala [22] studied the ℓ_∞-version of a subclass of network design problems which encompasses the Generalized Steiner Network problem, and gave a polynomial-time ℓ-approximation algorithm for it.

Our results for ℓ_p-Group Steiner Tree and ℓ_p-Group ATSP. We consider the ℓ_p-version of the Group Steiner Tree and the Group ATSP problems.

In Section 7 of the full version of the paper, we refine the SoS relaxation for ℓ_p-Shortest Path to obtain approximation algorithms for ℓ_p-Group ATSP and ℓ_p-Group Steiner Tree, and thus obtain approximation algorithms for these problems as well. In particular, we prove the following two results:

▶ **Theorem 9.** *There exists an approximation algorithm for ℓ_p-Group ATSP that given graph G, groups R_i, and parameters $p \in \mathbb{Z}_{\geq 0}$ and $c \in (0, 1/2)$ finds a $c^2 p \log^{2-1/p} n \log k$ approximation in time $m^{O(p) + ce^{O(1/c)} \log n} = m^{O_c(\log n)}$. We assume that k is at most polynomial in n.*

▶ **Theorem 10.** *There exists an approximation algorithm for the ℓ_p-Group Steiner Tree problem in undirected graphs that given a graph G, groups R_i, and parameters $p \in \mathbb{Z}_{\geq 1}$ and $c \in (0, 1/2)$ finds a $c^2 p \log^{2-1/p} n \log k$ approximation in time $m^{ce^{O(1/c)} \log n} = m^{O_c(\log n)}$. We assume that k is at most polynomial in n.*

Note that for $p = \lceil \log \ell \rceil$, we get an approximation algorithm for the ℓ_∞ norm.

1.1 Related Works

Previous results on the scalar-cost group Steiner tree problem. Group Steiner Tree with scalar costs was introduced by Reich and Widmayer [29]. Garg, Konjevod, and Ravi [14] gave an $O(\log^2 n \log k)$-approximation to the problem, the first polylogarithmic approximation. Charikar, Chekuri, Goel, and Guha [6] gave the same $O(\log^2 n \log k)$-approximation with a deterministic algorithm. Then Charikar, Chekuri, Cheung, Dai, Goel, Guha, and Li [5] gave an $O(\log^3 k)$ approximation that works even with directed graphs; however, it required quasipolynomial time. Finally, Chekuri and Pál [7] gave an $O(\log^2 k)$ approximation for undirected graphs also in quasipolynomial time. The approximation guarantees of [14] and [6] are presented above with the improvement resulting from using metric embedding by Fakcharoenphol, Rao and Talwar [12]. In terms of the approximation guarantee and running time, out algorithm is most similar to that by Chekuri and Pál [7]: when $k = \Theta(n)$, the approximation guarantees match. In terms of techniques used, our algorithm uses some ideas from that by Garg, Konjevod, and Ravi [14].

Dijkstra-style Algorithm for ℓ_p-Shortest Path. The authors of [3] describe a Dijkstra-style algorithm for the ℓ_p-Shortest Path problem and claim that it achieves an $O(\min\{p, \log \ell\})$-approximation. However, we show that this claim is incorrect and, in fact, the approximation factor of their algorithm is at least $\Omega(n^{1-1/p})$. We discuss this algorithm in Appendix B of the full version of the paper.

Multi-objective combinatorial optimization for shortest path and network design. The work in this paper is closely related to multi-objective combinatorial optimization (MOCO). This area studies combinatorial optimization problems in the presence of multiple competing objective functions. Much of the literature on MOCO is concerned with finding all or some Pareto efficient solutions, that is, solutions that are not dominated in every objective by any other solution, a problem which is often intractable due to the exponential number of these points. In particular, there is prior MOCO work on both shortest path [16, 25, 4] and network design problems [28]. We refer the reader to the paper of Ruzika and Hamacher [30] for a survey on multi-objective spanning tree problems, and the book of Ehrgott [11] for an overview of multi-criteria optimization area as a whole.

1.2 Technical Overview

Let us first discuss the ℓ_p-Shortest Path problem. The most basic variant of this problem is when G is a series-parallel graph (see Section 2 for definitions) and $p = 2$. The first idea is to write an LP flow relaxation with a convex objective: minimize $\| \sum_{e \in E} c_e x_e \|^2$ subject to the constraint that $(x_e)_{e \in E}$ define a unit flow from s to t. However, this LP has an integrality gap of $\sqrt{\ell}$. To deal with this problem, Li, Xu, and Zhang [23] introduced a new constraint for ℓ_∞-Shortest Path: the constraint loosely speaking says that the LP cost of every subgraph/block B in the series-parallel decomposition of G is at most OPT times the probability (according to the LP) that the path visits B. Unfortunately, this new constraint does not help when $p = 2$.

Instead, we consider a sum-of-squares (SoS) strengthening of this LP.[2] The SoS relaxation gives valuations not only to individual edges but also to tuples of edges. Using the standard notation of pseudo-expectations (see Section 2), the SoS relaxation for $p = 2$ gives $\widetilde{\mathbb{E}}[x_e]$ for every edge e and $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}]$ for every pair of edges e_1 and e_2. The former could be interpreted as the probability that $e \in P$ and the latter as the probability as both $e_1, e_2 \in P$ according to the relaxation. To the best of our knowledge, this is the first flow-based SoS or SDP relaxation studied in the literature.

Our algorithm is very straightforward. We start at $u_0 = s$, then choose one of the edges outgoing from u_1 with probability of choosing e being equal to $\widetilde{\mathbb{E}}[x_e]$. We get to a vertex u_1 and then again sample one of the edges leaving u_1 with probability of choosing e proportional to $\widetilde{\mathbb{E}}[x_e]$. We repeat this step over and over until we reach t. It is clear that the algorithm finds an s-t path P.

Now we need to upper bound the cost of P. We do that recursively using the series-parallel decomposition of G.[3] Assume that G is composed of subgraphs/blocks $B_1, \dots, B_t$ and our algorithm achieves an α approximation for the squared ℓ_2-cost in each of them. For simplicity, assume that $t = 2$ for now. There are two cases: G is a (i) parallel and (ii) series composition of B_1 and B_2. Consider the first case. The SoS relaxation ensures that $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}] = 0$ for all $e_1 \in B_1$ and $e_2 \in B_2$; this means that the SoS solution is simply a convex combination of solutions for B_1 and B_2 with some weights p_1 and p_2. Also, with probability p_1, the first edge of P will be in B_1 and then the entire path will be in B_1; similarly, with probability p_2, the entire path will be in B_2. Thus running the algorithm reduces to randomly choosing a block B_i with probability p_i and then running the algorithm in B_i. Since the algorithm gets an α approximation in each B_i, it also gets an α approximation in the entire graph. Interestingly, this step would already fail if we used the basic LP relaxation; however, a Sherali–Adams or configuration LP would work in *this* case.

The second case – when G is a series composition of B_1 and B_2 is more challenging and requires the power of an SDP relaxation. Let $P_i = P \cap B_i$. Write the squared objective as follows:

$$\Big\|\sum_{e\in P} c_e\Big\|_2^2 = \underbrace{\Big\|\sum_{e\in P_1} c_e\Big\|_2^2 + \Big\|\sum_{e\in P_2} c_e\Big\|_2^2}_{\leq \alpha \mathsf{OPT}^2 \text{ (in expectation)}} + 2\sum_{\substack{e_1\in P_1\\ e_2\in P_2}} \langle c_{e_1}, c_{e_2}\rangle. \tag{2}$$

The first two terms are squared ℓ_2-costs of paths P_1 and P_2. As we assumed, they are at most α times their SoS costs, and thus their sum is at most $\alpha\mathsf{OPT}^2$ (in expectation). We now analyze the third term. It is not hard to see that our algorithm samples edges in P_1 and P_2 independently (because the last vertex of P_1 and the first vertex of P_2 are fixed). Therefore,

$$\mathbb{E}\Big[\sum_{\substack{e_1\in P_1\\ e_2\in P_2}} \langle c_{e_1}, c_{e_2}\rangle\Big] = \sum_{\substack{e_1\in B_1\\ e_2\in B_2}} \langle c_{e_1}, c_{e_2}\rangle \cdot \Pr(e_1, e_2 \in P) = \sum_{\substack{e_1\in B_1\\ e_2\in B_2}} \langle c_{e_1}, c_{e_2}\rangle \cdot \Pr(e_1 \in P) \cdot \Pr(e_2 \in P)\,. \tag{3}$$

[2] It is sufficient to use a vector-flow SDP with one vector variable per edge in order to approximate the ℓ_2-cost in series-parallel graphs. However, we need higher degree SoS relaxations when $p > 2$ and in general graphs.

[3] Interestingly, neither the relaxation nor the algorithm uses the series-parallel decomposition of G.

It would be natural to upper bound this expression by the corresponding expression in the SoS objective (appropriately scaled):

$$\sum_{\substack{e_1 \in B_1 \\ e_2 \in B_2}} \langle c_{e_1}, c_{e_2} \rangle \cdot \widetilde{\mathbb{E}}[x_{e_1} x_{e_2}].$$

However, this is not possible, since it may happen that $\Pr(e_1 \in P) \cdot \Pr(e_2 \in P) > 0$ but $\widetilde{\mathbb{E}}[x_{e_1} x_{e_2}] = 0$. Instead, observing that for every edge e, $\Pr(e \in P) = \widetilde{\mathbb{E}}[x_e]$, we rewrite and upper bound (3):

$$\sum_{\substack{e_1 \in B_1 \\ e_2 \in B_2}} \langle c_{e_1}, c_{e_2} \rangle \, \widetilde{\mathbb{E}}[x_{e_1}] \, \widetilde{\mathbb{E}}[x_{e_2}] \leq \sum_{e_1, e_2 \in E} \langle c_{e_1}, c_{e_2} \rangle \cdot \widetilde{\mathbb{E}}[x_{e_1}] \cdot \widetilde{\mathbb{E}}[x_{e_2}]$$

$$= \Big\| \widetilde{\mathbb{E}}\Big[\sum_{e \in E} c_e x_e\Big] \Big\|_2^2 \leq \widetilde{\mathbb{E}}\Big[\Big\| \sum_{e \in E} c_e x_e \Big\|^2\Big] \leq \mathsf{OPT}^2.$$

Here, we first expanded the summation, then used the pseudo-expectation Lyapunov's inequality $\|\widetilde{\mathbb{E}}[f]\|_2^2 \leq \widetilde{\mathbb{E}}[\|f\|_2^2]$ (see Fact 18), and finally observed that the last pseudo-expectation is the SoS objective for G. We conclude that the expected squared cost of P is at most $(\alpha + 2)\mathsf{OPT}$. Applying this argument recursively, we get an $O(d)$-approximation for the squared cost and an $O(\sqrt{d})$-approximation for the cost itself in series-parallel graphs of order/depth d.

When $p > 2$ and blocks in the series-parallel composition of G are formed by $t > 2$ lower-order blocks, the proof becomes more technical. In particular, we need to use a new majorization inequality for pseudo-expectation, which we present in Section 4.

The SoS relaxation for arbitrary graphs is the same as that for series-parallel graphs (except that its degree is higher). However, the rounding algorithm is quite different. Very informally, the algorithm in its simplest form resembles Savitch's algorithm for s-t connectivity in $O(\log^2 n)$ space [31] (see also [7]): (i) we sample the middle edge $e = (u, v)$ of the path using probabilities provided by $\widetilde{\mathbb{E}}[\cdot]$, (ii) condition $\widetilde{\mathbb{E}}[\cdot]$ on e being the middle edge, (iii) then recursively find paths P_1 from s to u and (independently) P_2 from v to t, using the *conditional* pseudo-expectation. To upper bound the cost, as in the analysis of the algorithm for series-parallel graphs, we first use (2) , then bound the third term using a variant of (3), and finally use the majorization inequality for pseudo-expectations.

To solve Group ATSP, we loosely speaking add SoS constraints that require that the tour P visits every group (for technical reasons, we need to require that P visits each group exactly once). Then we run the rounding algorithm for ℓ_p-Shortest Path in arbitrary graphs. It is not guaranteed that P indeed visits every group; however, using the machinery we developed for bounding the cost of P, we show that P visits every group with probability at least $\Omega(1/\log n)$. By sampling sufficiently many tours and concatenating them, we obtain the desired solution with high probability. The Group Steiner Tree problem easily reduces to Group ATSP.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2 we define series-parallel graphs, relevant combinatorial quantities and notation used in the rest of the paper, and give some basic facts on Sum-of-Squares relaxations. In Section 3, we describe our Sum-of-Squares relaxation for ℓ_p-Shortest Path in directed acyclic graphs. In Section 4 we show a majorization inequality for pseudo-expectations used in the analysis of our algorithms. In Section 5 we describe and analyze our rounding algorithm for ℓ_p-Shortest Path in series-parallel graphs. In Section 6 we describe our approximation algorithm for ℓ_p-Shortest Path in arbitrary graphs.

The remaining sections appear in the full version of the paper [24]. Section 6 of the full version of the paper contains the proofs of the theorems that have been ommitted in this version. In Section 7 of the full version, we present our algorithms for ℓ_p-Group ATSP and ℓ_p-Group Steiner Tree. In Section 8 of the full version, we give our hardness results: hardness of approximation results for ℓ_p-Shortest Path with potentially negative edge costs and for ℓ_∞-Shortest Path. We also show that our analysis of the ℓ_p-Shortest Path algorithm in series-parallel graphs is tight. In Appendix A of the full version, we prove a recurrence formula and upper bound on multidimensional Bell numbers, which are used in the analyses of our algorithms for ℓ_p-Shortest Path.

2 Preliminaries and Notation

In this paper, all logs are base 2. In this paper, we consider Shortest Path and Group ATSP in directed graphs and Group Steiner Tree in undirected graphs. We assume that graphs may have parallel edges. Let $G = (V, E)$ be a directed graph. We denote $n = |V|$ and $m = |E|$. For $v \in V$, denote the sets of its outgoing and incoming edges by $\delta^+(v)$ and $\delta^-(v)$, respectively. Similarly, define $\delta^+(A)$ and $\delta^-(A)$ for subsets of vertices A. Finally, denote the set of edges from A to B by $\delta(A, B)$. We denote the i-th coordinate of vector edge cost c_e by $c_e(i)$.

2.1 Series-Parallel Graphs

We start with providing a recursive definition of directed series-parallel graphs with source s and sink t. A graph on two vertices s, t and one or more edges from s to t is a series-parallel graph of order (depth) 0. We denote the order of G as $\mathrm{ord}(G)$.

- **Parallel Composition.** Let $B_1, \ldots, B_t$ be series-parallel graphs that share only vertices s and t. Then their union G is a series-parallel graph. Define $\mathrm{ord}(G) = \max_j \mathrm{ord}(B_j)$.
- **Series Composition.** Let $B_1, \ldots, B_t$ be series-parallel graphs. Denote the source and sink of B_i by s_i and t_i (respectively). Assume that $t_i = s_{i+1}$ for all $i \in \{1, \ldots, t-1\}$ and that graphs B_i do not share any other vertices. Then the union G of graphs B_i is a series-parallel graph. Define $\mathrm{ord}(G) = \max_j \mathrm{ord}(B_j) + 1$.

In this definition, we only count *series* compositions when we compute the order of a series-parallel graph. We call vertices s and t *terminals*. We call intermediate graphs that we obtain while constructing G *blocks*. We denote the source and sink of a block B by s_B and t_B, respectively.

2.2 Combinatorics

Unlabeled Partitions. We say that a tuple of integers $\lambda = (\lambda_1, \ldots, \lambda_k)$ is an unlabeled partition of an integer $n \geq 1$ if $n = \sum_{i=1}^k \lambda_i$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 1$. We will denote this by $\lambda \vdash n$. We will denote the length of $\lambda = (\lambda_1, \ldots, \lambda_k)$ by $|\lambda| = k$.

Given an n and some tuple of non-negative integers α with $\alpha_1 + \ldots + \alpha_k = n$, we use standard notation for the multinomial coefficient

$$\binom{n}{\alpha} \stackrel{\text{def}}{=} \frac{n!}{\prod_{i=1}^k \alpha_i!}$$

Multidimensional Bell Numbers. Recall that the nth Bell number $\mathscr{B}_n$ equals the number of labeled partitions of a set of size n. In this paper, we will need a generalization of Bell numbers, known as multidimensional Bell numbers (see [32, Example 5.2.4] and [10]).

▶ **Definition 11.** *We say that a collection of subsets P is a partition of a set S if all subsets in P are disjoint and their union is S. Consider two partitions P and P' of S. We say that P' is a refinement of P if every $A \in P'$ is a subset of some $B \in P$.*

A d-dimensional partition of p is a tuple $(P_1, \ldots, P_d)$ where all P_i are partitions of $[p] \stackrel{\text{def}}{=} \{1, \ldots, p\}$ and each P_{i+1} is a refinement of P_i. The d-dimensional Bell number $\mathscr{B}_d(p)$ is the number of d-dimensional partitions of p. If $d = 0$ or $p = 0$, we let $\mathscr{B}_d(p) \stackrel{\text{def}}{=} 1$.

Note that 1-dimensional Bell numbers are simply the standard Bell numbers: $\mathscr{B}_1(i) = \mathscr{B}(i)$. We can also restate the definition of $\mathscr{B}_d(n)$ as follows. $\mathscr{B}_d(n)$ is the number of $(d+2)$-level rooted trees with n labeled leaves: the root must be in level 0, all leaves must be in level $d+1$, and all leaves are labeled with numbers from 1 to n with each number being used exactly once.

We will need the following recurrence formula for d-dimensional Bell numbers, which is proved in Appendix A of the full version of the paper.

▶ **Lemma 12.** *For every $d \geq 1$ and $p \geq 1$, we have*

$$\mathscr{B}_d(p) = \sum_{\lambda \vdash p} \binom{p}{\lambda} \prod_{i=1}^{|\lambda|} \mathscr{B}_{d-1}(\lambda_i) \bigg/ \prod_{j=1}^{p} \operatorname{count}(j, \lambda)!$$

where $\operatorname{count}(j, \lambda)$ *is the number of times j appears in λ.*

Now, we describe the exponential generating function for sequence $(\mathscr{B}_d(i))_i$ when d is fixed.

▶ **Fact 13** ([32, Example 5.2.4])**.** *Let $f_0(x) = \exp(x)$ and $f_{i+1}(x) = \exp(f_i(x) - 1)$. Then the exponential generating function for sequence $(\mathscr{B}_d(i))_{i=0}^{\infty}$ is given by:*

$$\sum_{i=0}^{\infty} \frac{\mathscr{B}_d(i) x^i}{i!} = f_d(x). \tag{4}$$

In this paper, we will present an approximation algorithm for ℓ_p-Shortest Path in depth-d series-parallel graphs with approximation factor $\mathscr{A}_d(p) \stackrel{\text{def}}{=} \mathscr{B}_d(p)^{1/p}$. From Fact 13, we obtain the following upper bound on $\mathscr{A}_d(p)$, proved in Appendix A of the full version of the paper.

▷ Claim 14. For all $p \geq 1$ and $d \geq 1$, we have

$$\mathscr{A}_d(p) = \mathscr{B}_d(p)^{1/p} = O(pd^{1-1/p}).$$

Let $\log^{(j)} p = \underbrace{\log \cdots \log}_{j \text{ times}} p$ and $\log^* p$ be the largest value of j such that $\log^{(j)} p \geq 1$. Then, the following upper bound on $\mathscr{A}_d(p)$ holds for $d \leq \log^* p$:

$$\mathscr{A}_d(p) = \mathscr{B}_d(p)^{1/p} \leq O\Big(\frac{p}{\log^{(d)} p}\Big).$$

2.3 Sum-of-Squares Relaxations

We recall some basics about the sum-of-squares relaxations. Sum-of-squares relaxations can be thought of in terms of moment matrices, pseudo-distributions, and pseudo-expectations. As is common, we will use the pseudo-expectation framework in this paper. We refer the reader to [13] for a detailed description of the sum-of-squares framework.

Consider a set of variables x_i where i belongs to some set of indices $\mathcal{I}$. We denote the entire collection of all variables $(x_i)_{i\in\mathcal{I}}$ by $\mathbf{x}$. Consider the set of multivariate polynomials $\mathbb{R}_{\leq d}[\mathbf{x}] \stackrel{\text{def}}{=} \mathbb{R}_{\leq d}[\{x_i : i \in \mathcal{I}\}]$ in variables x_i of degree at most d. We say that $f \in \mathbb{R}_{\leq d}[\mathbf{x}]$ is a sum of squares (SoS) if $f = \sum_{i=1}^m f_i^2$ for some polynomials $f_1, \dots, f_m$. Note that the product of SoS polynomials is a SoS, and so is any linear combination of SoS polynomials with positive coefficients.

In this paper, we consider SoS relaxations for the Boolean hypercube; that is, all variables x_i take values 0 and 1 in the intended solution. Therefore, we work with the quotient ring $\mathbb{R}_{\leq d}/\langle x_i^2 - x_i\rangle_i$, where $\langle x_i^2 - x_i\rangle_i$ is the ideal generated by polynomials $x_i^2 - x_i$. In other words, we identify monomials $x_{i_1}^{a_1}, \dots, x_{i_t}^{a_t}$ and $x_{i_1}, \dots, x_{i_t}$ for all $i_1, \dots, i_t$ and $a_1, \dots, a_t \geq 1$ such that $\sum_{i=1}^t a_i \leq d$. In particular, we will write $f = g$ if $f - g \in \langle x_i^2 - x_i\rangle_i$.

▶ **Definition 15.** *A linear map $\widetilde{\mathbb{E}} : \mathbb{R}_{\leq d}[\mathbf{x}]/\langle x_i^2 - x_i\rangle_i \to \mathbb{R}$ is a pseudo-expectation of degree d if it satisfies the following properties.*

- $\widetilde{\mathbb{E}}[1] = 1$,
- $\widetilde{\mathbb{E}}[f^2] \geq 0$ *for every polynomial f of degree at most $d/2$,*

We say that a pseudo-expectation $\widetilde{\mathbb{E}}$ satisfies an equality constraint $f = 0$ if $\widetilde{\mathbb{E}}[fg] = 0$ whenever $\deg fg \leq d$.

Given an objective function f and sets of equality and inequality constraints, we can find a pseudo-expectation $\widetilde{\mathbb{E}}$ that maximizes $\widetilde{\mathbb{E}}[f]$ and satisfies all the constraints in time polynomial in $N^{O(d)}$, where N is the number variables, as long as it satisfies certain regularity conditions [27]. When we prove any statements about pseudo-expectations $\widetilde{\mathbb{E}}[f]$ below, we will *always* implicitly assume that $d \geq \Omega(\deg f)$ so that all the inequalities appearing in the proofs have degree at most d.

▶ **Definition 16.** *Let $\widetilde{\mathbb{E}}$ be a pseudo-expectation of degree d. Assume that g is a sum of squares and $\widetilde{\mathbb{E}}[g] > 0$. Then the conditional pseudo-expectation $\widetilde{\mathbb{E}}[\cdot \mid g]$ operator is defined as follows: $\widetilde{\mathbb{E}}[f \mid g] \stackrel{\text{def}}{=} \widetilde{\mathbb{E}}[fg]/\widetilde{\mathbb{E}}[g]$.*

▶ **Fact 17.** *A conditional pseudo-expectation $\widetilde{\mathbb{E}}[\cdot \mid g]$ is a pseudo-expectation of degree $d' = d - \deg g$. If $\widetilde{\mathbb{E}}$ satisfies an equality or inequality constraint of degree at most d', then so does $\widetilde{\mathbb{E}}$.*

We will use Lyapunov's inequality for pseudo-expectations (which is also referred to as Jensen's inequality in the literature).

▷ Claim 18. Let g be a sum of squares and f be any polynomial. Assume that $\deg f^2 g \leq d$. Then,

$$\widetilde{\mathbb{E}}[fg]^2 \leq \widetilde{\mathbb{E}}[f^2 g]\,\widetilde{\mathbb{E}}[g]. \tag{5}$$

If $\widetilde{\mathbb{E}}[g] > 0$, the inequality can be restated as

$$\widetilde{\mathbb{E}}[f \mid g]^2 \leq \widetilde{\mathbb{E}}[f^2 \mid g]. \tag{6}$$

▷ Claim 19. Let $f_1, \dots, f_t$ be SoS polynomials. Then, $\widetilde{\mathbb{E}}[\left(\sum_{i=1}^t f_i\right)^p] \geq \sum_{i=1}^t \widetilde{\mathbb{E}}[f_i^p]$.

Proof. We expand $\left(\sum_{i=1}^t f_i\right)^p$ as $\sum_{\substack{\alpha_1,\dots,\alpha_t \geq 0 \\ \alpha_1+\cdots+\alpha_t = p}} \binom{p}{\alpha} f_1^{\alpha_1} \cdots f_t^{\alpha_t}$. All terms in the expansion are SoS polynomials and thus have non-negative pseudo-expectations. The claim follows from the observation that all terms f_i^p are present in the expansion. ◁

3 Sum of Squares Relaxation for ℓ_p-Shortest Path

In this section, we first present our SoS relaxation for ℓ_p-Shortest Path in directed acyclic graphs (DAGs). In Section 5, we will present a rounding algorithm for series-parallel graphs and then, in Section 6 of the full version of the paper, for layered graphs. The latter result will also yield an algorithm for arbitrary graphs. We will also describe a few basic properties that feasible solutions for this relaxation satisfy.

Relaxation. We use a degree $2p$ SoS relaxation with variables $\mathbf{x} = (x_e)_{e\in E}$ for ℓ_p-Shortest Path in series-parallel graphs.

$$\begin{aligned} \min \quad & \widetilde{\mathbb{E}}\Big[\sum_{i=1}^{\ell}\Big(\sum_{e} c_e(i)x_e\Big)^p\Big] \\ \text{subject to} \quad & (x_e)_{e\in E} \text{ is a unit flow from } s \text{ to } t \end{aligned}$$

The flow constraint says that $\sum_{e\in\delta^+(u)} x_e - \sum_{e\in\delta^-(u)} x_e = 0$ for all u other than s and t (flow conservation) and $\sum_{e\in\delta^+(s)} x_e - 1 = 0$ (x_e sends 1 unit of flow from s to t). It is clear that this is a relaxation for the ℓ_p-Shortest Path problem: $\widetilde{\mathbb{E}}[\sum_{i=1}^{\ell}(\sum_e c_e(i)x_e)^p\Big] \le \mathsf{OPT}^p$, where OPT is the ℓ_p-cost of the optimal s-t path.

Basic properties of the SoS relaxation. We say that two edges e_1 and e_2 are *compatible* if both of them belong to some s-t path; otherwise, we say that e_1 and e_2 are *incompatible*. In a series-parallel graph edges e_1 and e_2 are incompatible if and only if there exist two parallel blocks B_1 and B_2 such that e_1 lies in B_1 and e_2 lies in B_2. For any set of vertices A, let $\mathbf{x}_A^+ = \sum_{e\in\delta^+(A)} x_e$ and $\mathbf{x}_A^- = \sum_{e\in\delta^-(A)} x_e$.

▷ Claim 20. Assume that G is a DAG and $\widetilde{\mathbb{E}}$ is a feasible pseudo-expectation for the relaxation. Let h be a multivariate polynomial. Then

1. If $A \subseteq V$ contains neither of the terminals, then $\widetilde{\mathbb{E}}[(\mathbf{x}_A^+ - \mathbf{x}_A^-)h] = 0$. If A contains s but not t, $\widetilde{\mathbb{E}}[(\mathbf{x}_A^+ - \mathbf{x}_A^-)h] = \widetilde{\mathbb{E}}[h]$.
2. If e_1 and e_2 are not compatible, then $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}h] = 0$.
3. Assume further that G is a series-parallel graph. Let (L, R) be an s_B-t_B cut in a block B. Let $f_{LR} = \sum_{e\in\delta(L,R)} x_e$. Then
$$\widetilde{\mathbb{E}}\Big[f_{LR}h\Big] = \widetilde{\mathbb{E}}\Big[\Big(\sum_{e\in\delta^+(s_B)\cap B} x_e\Big)h\Big].$$
In particular, $\widetilde{\mathbb{E}}[f_{LR}h]$ does not depend on the cut (L, R) in B.

Proof.

1. The SoS relaxation satisfies the flow conservation constraints and the constraint that the amount of flow being routed equals 1. Therefore, it satisfies any linear combination of them. In particular, it satisfies degree-1 polynomial equations $\mathbf{x}_A^+ - \mathbf{x}_A^- = 0$ when $s, t \notin A$ and $\mathbf{x}_A^+ - \mathbf{x}_A^- = 1$ when $s \in A$ but $t \notin A$. The first item follows.
2. It is sufficient to prove the statement for all monomials (the claim then follows by the linearity of $\widetilde{\mathbb{E}}$). Thus, we will assume that h is a monomial. Recall that an s-t cut is monotone if it cuts exactly one edge on every s-t path. Since e_1 and e_2 are incompatible, there is a monotone s-t cut $(A, \bar{A})$ that cuts both of them. We apply item 1 to polynomial

$x_{e_1}h$ and get $\widetilde{\mathbb{E}}[x_{e_1}h] = \widetilde{\mathbb{E}}[x_{e_1}\mathbf{x}_A^+ h] = \widetilde{\mathbb{E}}[x_{e_1}(x_{e_1} + x_{e_2} + \dots)h]$. Here, $\dots$ is a sum of some x_e (the coefficient of each of them is 1). Note that $x_e h$ is a monomial, thus $x_e h = (x_e h)^2$ and therefore $\widetilde{\mathbb{E}}[x_e h] \geq 0$. We conclude that

$$\widetilde{\mathbb{E}}[x_{e_1}h] \geq \widetilde{\mathbb{E}}[x_{e_1}(x_{e_1} + x_{e_2})h] = \widetilde{\mathbb{E}}[(x_{e_1} + x_{e_1}x_{e_2})h]$$

and simplifying, we get $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}h] \leq 0$. Since $x_{e_1}x_{e_2}h$ is a monomial, $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}h] \geq 0$, and thus $\widetilde{\mathbb{E}}[x_{e_1}x_{e_2}h] = 0$.

3. Item 3 follows immediately from item 1. ◁

4 Majorization Inequalities for Pseudo-expectations

In this section, we will prove a majorization inequality for pseudo-expectations. This inequality generalizes already known pseudo-expectation Lyapunov's (see Claim 18) and Hölder's (see [2]) inequalities.

▶ **Definition 21.** *Consider two integer sequences $a_1 \geq a_2 \geq \dots \geq a_k \geq 0$ and $b_1 \geq \dots \geq b_k \geq 0$. We say a majorizes b and write $a \succeq b$, if $\sum_{i=1}^k a_i = \sum_{i=1}^k b_i$ and for all $1 \leq i \leq k$ we have*

$$a_1 + \dots + a_i \geq b_1 + \dots + b_i.$$

Sequence majorization is a powerful tool for proving inequalities; it appears in widely used Muirhead's [26] and Karamata's [17] majorization inequalities. We now present a majorization inequality for pseudo-expectations. An analogous inequality for true expectations easily follows both from Karamata's and from Muirhead's majorization inequality.

▶ **Lemma 22.** *Consider a degree d pseudo-expectation $\widetilde{\mathbb{E}}$. Let $a \succeq b$ and f be an SoS polynomial of degree $\deg(f^{a_1}) \leq d$. Then*

$$\prod_{i=1}^k \widetilde{\mathbb{E}}[f^{a_i}] \geq \prod_{i=1}^k \widetilde{\mathbb{E}}[f^{b_i}]. \tag{7}$$

Proof. First, observe that this inequality for sequences $(r+1, r-1) \succeq (r, r)$ follows from Lyapunov's inequality for pseudo-expectations (Claim 18). Indeed, let $g = f^{r-1}$. Then, Lyapunov's inequality states that

$$\widetilde{\mathbb{E}}[f^r]^2 = \widetilde{\mathbb{E}}[fg]^2 \leq \widetilde{\mathbb{E}}[f^2 g]\,\widetilde{\mathbb{E}}[g] = \widetilde{\mathbb{E}}[f^{r+1}]\,\widetilde{\mathbb{E}}[f^{r-1}], \tag{8}$$

as required. Now we use this inequality to show the desired inequality (7) for a more general case $(p+1, q-1) \succeq (p, q)$.

▷ Claim 23. Let $p \geq q \geq 1$ be integers. Then

$$\widetilde{\mathbb{E}}[f^{p+1}]\,\widetilde{\mathbb{E}}[f^{q-1}] \geq \widetilde{\mathbb{E}}[f^p]\,\widetilde{\mathbb{E}}[f^q].$$

Proof. We just proved the inequality when $p = q$. So we will assume below that $p > q$. Since f is an SoS polynomial, $\widetilde{\mathbb{E}}[f^r] \geq 0$ for all integers $0 \leq r \leq p$. Let us assume first that the inequality is strict for all r: $\widetilde{\mathbb{E}}[f^r] > 0$. Then, by dividing and multiplying $\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^p]$ by $\prod_{r=q}^{p-1} \widetilde{\mathbb{E}}[f^r]$, we obtain the following identity.

$$\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^p] = \frac{\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^{q+1}]\cdots\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^p]}{\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^{q+1}]\cdots\widetilde{\mathbb{E}}[f^{p-1}]}$$

Now we upper bound the numerator of this fraction by iteratively applying (8).

$$\left(\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^q]\right)\widetilde{\mathbb{E}}[f^{q+1}]\cdots\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^p] \le \left(\widetilde{\mathbb{E}}[f^{q-1}]\,\widetilde{\mathbb{E}}[f^{q+1}]\right)\widetilde{\mathbb{E}}[f^{q+1}]\cdots\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^p]$$

$$\widetilde{\mathbb{E}}[f^{q-1}]\left(\widetilde{\mathbb{E}}[f^{q+1}]\,\widetilde{\mathbb{E}}[f^{q+1}]\right)\cdots\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^p] \le \widetilde{\mathbb{E}}[f^{q-1}]\left(\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^{q+2}]\right)\cdots\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^p]$$

$$\vdots$$

$$\widetilde{\mathbb{E}}[f^{q-1}]\,\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^{q+1}]\cdots\left(\widetilde{\mathbb{E}}[f^p]\,\widetilde{\mathbb{E}}[f^p]\right) \le \widetilde{\mathbb{E}}[f^{q-1}]\,\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^{q+1}]\cdots\left(\widetilde{\mathbb{E}}[f^{p-1}]\,\widetilde{\mathbb{E}}[f^{p+1}]\right)$$

We conclude that

$$\widetilde{\mathbb{E}}[f^q]\,\widetilde{\mathbb{E}}[f^p] \le \widetilde{\mathbb{E}}[f^{q-1}]\,\widetilde{\mathbb{E}}[f^{p+1}],$$

as desired. If $\widetilde{\mathbb{E}}[f^r] = 0$ for some r, we apply the inequality to $\hat{f} = f + \varepsilon$ (where $\varepsilon > 0$). Now $\widetilde{\mathbb{E}}[\bar{f}^r] \ge \varepsilon^r > 0$, since f is a SoS polynomial. Therefore, $\widetilde{\mathbb{E}}[\hat{f}^q]\,\widetilde{\mathbb{E}}[\hat{f}^p] \le \widetilde{\mathbb{E}}[\hat{f}^{q-1}]\,\widetilde{\mathbb{E}}[\hat{f}^{p+1}]$. Letting $\varepsilon \to 0$, we obtain the desired inequality in the limit. ◁

Consider an integer sequence $a_1 \ge \dots a_k \ge 0$ and two indices $1 \le i^* < j^* \le k$ such that $a_{i^*} - a_{j^*} \ge 2$. Define a *transfer* or *T-transform* as follows: we decrease a_{i^*} by 1, increase a_{j^*} by 1, and then sort the obtained sequence in descending order. Claim 23 implies that Lemma 22 holds for sequence a and sequence b obtained from a by a T-transform.

Finally, we use that if $a \succeq b$ then b can be obtained by a sequence of T-transforms [26]: $a = a^{(0)} \mapsto a^{(1)} \mapsto a^{(2)} \mapsto \cdots \mapsto a^{(T)} = b$. As we proved, the value of the product $\prod_{i=1}^k \widetilde{\mathbb{E}}[f^{a_i^{(t)}}]$ may only decrease each time we apply a T-transform. This concludes the proof of the lemma. ◀

Importantly, conditional pseudo-expectations are pseudo-expectations (see Fact 17) and thus Lemma 22 holds for conditional pseudo-expectations as well.

5 Sum-of-Squares Relaxation Rounding

In this section, we describe and analyze a rounding algorithm for series-parallel graphs. It gives an $(1+\varepsilon)A_d(p) = O(pd)$ approximation for ℓ_p-Shortest Path in series-parallel graphs of order d. Later we use a different algorithm with a similar analysis to solve the problem in layered and arbitrary graphs.

5.1 Algorithm

Let us denote $p_e = \widetilde{\mathbb{E}}[x_e]$ and $p_u = \sum_{e\in\delta^+(u)} \widetilde{\mathbb{E}}[x_e]$. Note that the SoS relaxation constraints ensure that p_e is an s-t flow; p_u equals the amount of flow that leaves vertex u. The total amount of flow is 1.

Algorithm 1 Rounding algorithm for SoS.

```
1: Input: series-parallel graph G with source s and sink t, a pseudo-expectation Ẽ
2: Output: an s-t path in G
3: Let u = s and P be an empty path.
4: while u ≠ t do
5:     Sample e ∈ δ+(u) with probability p_e/p_u = Ẽ[x_e] / Σ_{e∈δ+(u)} Ẽ[x_e]
6:     Append e to path P
7: end while
8: return P
```

▶ **Lemma 24.** *Let P be the path returned by Algorithm 1. Then* $\Pr(e \in P) = p_e$ *for every edge e and* $\Pr(u \in P) = p_u$ *for every vertex u.*

Proof. We consider all vertices in topological order and prove the desired formulas for $\Pr(u \in P)$ and $\Pr((u,v) \in P)$ by induction. For $u = s$, we have $\Pr(s \in P) = 1 = p_s$. Then, $\Pr((s,v) \in P) = p_{(s,v)}/p_s = p_{(s,v)}$, as required. Now assume that we proved the formulas for vertices u' proceeding u in the topological order. We have,

$$\Pr(u \in P) = \sum_{e=(u',u)\in\delta^-(u)} \Pr(e \in P) = \sum_{e=(u,u')\in\delta^-(u)} p_e = p_u.$$

Here, we used the induction hypothesis and the flow conservation condition at vertex u. Now let $e = (u,v)$.

$$\Pr(e \in P) = \Pr(e \in P \mid u \in P)\Pr(u \in P) = (p_e/p_u) \cdot p_u. \qquad ◀$$

Now we will prove an upper bound on the ℓ_p-cost of path P. The proof will be by induction on the series-parallel decomposition of G, going from lower to higher order blocks B. To analyze different blocks B, we first introduce some relevant notation.

Let us say that a path P visits block B if it contains at least one edge from B. Let $P_B = P \cap B$ be the restriction of P to B. If P does not visit B, let $P_B = \varnothing$. Note that if P visits B then it must go through the source s_B and sink t_B of B. However, if B has a parallel block B', a path may go through s_B and t_B but visit B' rather than B itself. It follows from Lemma 24 that the probability that P visits B equals $p_B \stackrel{\text{def}}{=} \sum_{e\in\delta^+(s_B)\cap B} p_e$.

Now, we define conditional expectations and pseudo-expectations restricted to B (that is, conditioned on the event that P visits B). Let $h_B = \sum_{e\in\delta^+(s_B)\cap B} x_e = \sum_{e\in\delta^+(s_B)\cap B} x_e^2$ be a SoS indicator of the event that P visits B. We let

$$\mathbb{E}_B[\cdot] \stackrel{\text{def}}{=} \mathbb{E}[\cdot \mid P \text{ visits } B] \qquad \text{and} \qquad \widetilde{\mathbb{E}}_B[\cdot] \stackrel{\text{def}}{=} \widetilde{\mathbb{E}}[\cdot \mid h_B].$$

In the sequel, we shall bound the costs of P coordinate-by-coordinate. Thus, we consider a set of scalar non-negative edge weights $a_e \geq 0$. Define $f_B \stackrel{\text{def}}{=} \sum_{e\in B} a_e \cdot x_e$. For a path P', let $\text{cost}(P') = \sum_{e\in P'} a_e$. Note that $f_B = \sum_{e\in B} a_e \cdot x_e^2$ and thus is a sum of squares.

▷ Claim 25. Let B' be a block inside B (possibly $B' = B$). Assume $p_B > 0$. Then, we have $\mathbb{E}_B[\text{cost}(P_{B'})^r] = \frac{\mathbb{E}[\text{cost}(P_{B'})^r]}{p_B}$ and $\widetilde{\mathbb{E}}_B[f_{B'}^r] = \frac{\widetilde{\mathbb{E}}[f_{B'}^r]}{p_B}$ when $r \in \{1,\ldots,p\}$.

Proof. We have

$$\begin{aligned}\mathbb{E}[\text{cost}(P_{B'})^r] &= \mathbb{E}[\text{cost}(P_{B'})^r \mid P_B \neq \varnothing] \cdot \Pr(P_B \neq \varnothing) + \mathbb{E}[\text{cost}(P_{B'})^r \mid P_B = \varnothing] \cdot \Pr(P_B = \varnothing)\\ &= \mathbb{E}_B[\text{cost}(P_{B'})^r] \cdot p_B + 0 \cdot (1 - p_B) = p_B \cdot \mathbb{E}_B[\text{cost}(P_{B'})^r],\end{aligned}$$

as required. To prove the second identity, consider a monomial g in the expansion of $f_{B'}^r$. We now prove that $\widetilde{\mathbb{E}}[gh_B] = \widetilde{\mathbb{E}}[g]$ and thus $\widetilde{\mathbb{E}}_B[g] \stackrel{\text{def}}{=} \widetilde{\mathbb{E}}[gh_B]/p_B = \widetilde{\mathbb{E}}[g]/p_B$. Note that only x_e with $e \in B'$ appear in g, and $\deg g = r \geq 1$. Choose an arbitrary x_e in g, say $e = (u,v)$ and let g' be such that $g = g'x_e$. Let (L,R) be a monotone s_B-t_B cut in B that cuts e. By Claim 20, item 3,

$$\widetilde{\mathbb{E}}[gh_B] = \widetilde{\mathbb{E}}\Big[g \sum_{e'\in\delta(L,R)} x_{e'}\Big] = \widetilde{\mathbb{E}}\Big[g' \sum_{e'\in\delta(L,R)} x_e x_{e'}\Big].$$

Now all edges $e' \in \delta(L,R)$ other than e are incompatible with e; for them, $\widetilde{\mathbb{E}}[g'x_e x_{e'}] = 0$ by Claim 20, item 2. Also, $\widetilde{\mathbb{E}}[g'x_e x_{e'}] = \widetilde{\mathbb{E}}[g'x_e]$ for $e' = e$. Therefore, $\widetilde{\mathbb{E}}[gh_B] = \widetilde{\mathbb{E}}[g'x_e] = \widetilde{\mathbb{E}}[g]$, as required. ◁

▶ **Lemma 26.** *Let $\widetilde{\mathbb{E}}$ be a feasible solution for the SoS relaxation of ℓ_p-Shortest Path. Let B be a block of order $\hbar = \operatorname{ord}(B)$ with $p_B > 0$. Then for every $r \le p$, we have*

$$\mathbb{E}_B\left[cost(P_B)^r\right] \le \mathscr{B}_\hbar(r)\,\widetilde{\mathbb{E}}_B[f_B^r].$$

Here, $\mathscr{B}_\hbar(r)$ is an $\hbar$-dimensional Bell number (see Section 2.2 for details).

Proof. We will prove the upper bound on $\mathbb{E}\left[\operatorname{cost}(P_B)^r\right]$ by induction on r and on the series-parallel decomposition of B. If $r = 0$ or $\hbar = 0$, the claim trivially holds. We consider two cases: when a block B is a parallel composition and when it is a series composition of lower-level blocks. We start with the former, much simpler case when B is a parallel composition of blocks $B_1, \ldots, B_t$ sharing the same source s_B and sink t_B. If P visits B, then it visits exactly one of the blocks $B_1, \ldots, B_t$. Therefore,

$$\mathbb{E}_B\left[\operatorname{cost}(P_B)^r\right] = \sum_{i=1}^{t} \mathbb{E}_{B_i}\left[\operatorname{cost}(P_{B_i})^r\right] \Pr\left(P \text{ visits } B_i | P \text{ visits } B\right) = \sum_{i=1}^{t} \frac{p_{B_i}}{p_B} \mathbb{E}_{B_i}\left[\operatorname{cost}(P_{B_i})^r\right]. \quad (9)$$

Note that $f_B = \sum_{i=1}^{t} f_{B_i}$. Applying Claim 19 and then Claim 25 twice, we get

$$\widetilde{\mathbb{E}}_B[f_B^r] \ge \sum_{i=1}^{t} \widetilde{\mathbb{E}}_B[f_{B_i}^r] = \sum_{i=1}^{t} \frac{1}{p_B}\,\widetilde{\mathbb{E}}[f_{B_i}^r] = \sum_{i=1}^{t} \frac{p_{B_i}}{p_B}\,\widetilde{\mathbb{E}}_{B_i}[f_{B_i}^r]. \quad (10)$$

In fact the inequality above is an equality by Claim 20, part 2, but we do not need that here. Comparing (9) and (10) term-by-term, and using the induction hypothesis, we get

$$\mathbb{E}_B\left[\operatorname{cost}(P_B)^r\right] \le \mathscr{B}_\hbar(r)\,\widetilde{\mathbb{E}}_B[f_B^r].$$

This concludes the analysis of this case. Now we assume that B is a series composition of blocks $B_1, \ldots, B_t$. In this case, if P visits B then it visits all B_i; if it visits B_i, it visits B and all other B_j. Thus, $p_B = p_{B_1} = \cdots = p_{B_t}$. Also, P_B is the concatenation of $P_{B_1}, \ldots, P_{B_t}$. Using the multinomial theorem, we get

$$\begin{aligned}\mathbb{E}_B\left[\operatorname{cost}(P_B)^r\right] &= \mathbb{E}_B\left[\left(\sum_{i=1}^{t} \operatorname{cost}(P_{B_i})\right)^r\right] = \mathbb{E}_B\left[\sum_{\substack{\alpha_1,\ldots,\alpha_t \ge 0\\ \alpha_1+\cdots+\alpha_t = r}} \binom{r}{\alpha} \prod_{i=1}^{t} \operatorname{cost}(P_{B_i})^{\alpha_i}\right] \quad (11)\\ &= \sum_{\substack{\alpha_1,\ldots,\alpha_t \ge 0\\ \alpha_1+\cdots+\alpha_t = r}} \binom{r}{\alpha} \mathbb{E}_B\left[\prod_{i=1}^{t} \operatorname{cost}(P \cap B_i)^{\alpha_i}\right].\end{aligned}$$

Observe that if P enters B, it necessarily visits $s_{B_1}, \ldots, s_{B_t}$ and thus paths $P_{B_1}, \ldots, P_{B_t}$ are mutually independent. We then have

$$\mathbb{E}_B\left[\operatorname{cost}(P_B)^r\right] = \sum_{\substack{\alpha_1,\ldots,\alpha_t \ge 0\\ \alpha_1+\cdots+\alpha_t = r}} \binom{r}{\alpha} \prod_{i=1}^{t} \mathbb{E}_B\left[\operatorname{cost}(P_{B_i})^{\alpha_i}\right].$$

By Claim 25, $\mathbb{E}_B\left[\operatorname{cost}(P_{B_i})^{\alpha_i}\right] = \mathbb{E}\left[\operatorname{cost}(P_{B_i})^{\alpha_i}\right]/p_B = \mathbb{E}\left[\operatorname{cost}(P_{B_i})^{\alpha_i}\right]/p_{B_i} = \mathbb{E}_{B_i}\left[\operatorname{cost}(P_{B_i})^{\alpha_i}\right]$ and $\widetilde{\mathbb{E}}_B[f_{B_i}^{\alpha_i}] = \widetilde{\mathbb{E}}[f_{B_i}^{\alpha_i}]/p_B = \widetilde{\mathbb{E}}[f_{B_i}^{\alpha_i}]/p_{B_i} = \widetilde{\mathbb{E}}_{B_i}[f_{B_i}^{\alpha_i}]$. Using that $\operatorname{ord}(B_i) \le h - 1$ and applying the induction hypothesis, we get

$$\mathbb{E}_B\left[\text{cost}(P_B)^r\right] = \sum_{\substack{\alpha_1,\dots,\alpha_t\geq 0\\ \alpha_1+\dots+\alpha_t=r}} \binom{r}{\alpha} \prod_{i=1}^{t} \mathbb{E}_B\left[\text{cost}(P_{B_i})^{\alpha_i}\right] = \sum_{\substack{\alpha_1,\dots,\alpha_t\geq 0\\ \alpha_1+\dots+\alpha_t=r}} \binom{r}{\alpha} \prod_{i=1}^{t} \mathbb{E}_{B_i}\left[\text{cost}(P_{B_i})^{\alpha_i}\right]$$

$$\leq \sum_{\substack{\alpha_1,\dots,\alpha_t\geq 0\\ \alpha_1+\dots+\alpha_t=r}} \binom{r}{\alpha} \prod_{i=1}^{t} \mathscr{B}_{\hbar-1}(\alpha_i)\cdot \underset{B_i}{\widetilde{\mathbb{E}}}\, f_{B_i}^{\alpha_i} = \sum_{\substack{\alpha_1,\dots,\alpha_t\geq 0\\ \alpha_1+\dots+\alpha_t=r}} \binom{r}{\alpha} \prod_{i=1}^{t} \mathscr{B}_{\hbar-1}(\alpha_i)\cdot \underset{B}{\widetilde{\mathbb{E}}}\, f_{B_i}^{\alpha_i}.$$

Now, every α in the summation defines an unlabeled partition λ of n: λ is obtained by sorting all non-zero entries α_i of α. Let us denote this $\alpha \to \sigma$. For example, $\alpha = (4,1,0,2,4,0,2,2) \to \lambda = (4,4,2,2,2,1)$. Thus, to go over all α, it is sufficient to go over all $\lambda \vdash r$ and then all α such that $\alpha \to \lambda$. To do the latter, we go over all choices of distinct indices $j_1,\dots,j_{|\lambda|}$ and define α as follows: $\alpha_{j_i} = \lambda_i$ for $i \in [|\lambda|]$ and $\alpha_j = 0$ for all other j. However, if $\lambda_i = \lambda_{i'}$, then indices $j_1,\dots,j_{|\lambda|}$ and those with j_i and $j_{i'}$ swapped define the same α. It is easy to see that the above procedure defines every α exactly $\prod_{j=1}^{r}\text{count}(j,\lambda)$ times. In the example above with $\lambda = (4,4,2,2,2,1)$, this procedure defines every α exactly $2!\cdot 3!$ times. Finally note that $\binom{r}{\alpha} = \binom{r}{\lambda}$. Keeping this discussion in mind, we rewrite the upper bound on $\mathbb{E}_B\left[\text{cost}(P_B)^r\right]$ as follows.

$$\mathbb{E}_B\left[\text{cost}(P_B)^r\right] \leq \sum_{\lambda\vdash r,\ |\lambda|\leq t} \binom{r}{\lambda} \frac{\sum_{\substack{j_1,\dots,j_{|\lambda|}\in[t]\\ \text{distinct}}} \prod_{i=1}^{|\lambda|} \mathscr{B}_{\hbar-1}(\lambda_i)\cdot \widetilde{\mathbb{E}}_B[f_{B_{j_i}}^{\lambda_i}]}{\prod_{j=1}^{r}\text{count}(j,\lambda)}$$

$$\leq \sum_{\lambda\vdash r,\ |\lambda|\leq t} \binom{r}{\lambda}\cdot \frac{\prod_{i=1}^{|\lambda|}\mathscr{B}_{\hbar-1}(\lambda_i)}{\prod_{j=1}^{r}\text{count}(j,\lambda)} \sum_{j_1,\dots,j_{|\lambda|}\in[t]} \prod_{i=1}^{|\lambda|} \underset{B}{\widetilde{\mathbb{E}}}[f_{B_{j_i}}^{\lambda_i}].$$

Note that we removed the requirement that all j_i are distinct in the last inequality (this is valid, since all terms are non-negative). Now we use Claim 19 and then the majorization inequality (see Lemma 22) to upper bound each term in the inner sum.

$$\sum_{j_1,\dots,j_{|\lambda|}\in[t]} \prod_{i=1}^{|\lambda|} \underset{B}{\widetilde{\mathbb{E}}}[f_{B_{j_i}}^{\lambda_i}] = \prod_{i=1}^{|\lambda|}\sum_{j\in[t]} \underset{B}{\widetilde{\mathbb{E}}}\Big[f_{B_j}^{\lambda_i}\Big] = \prod_{i=1}^{|\lambda|} \underset{B}{\widetilde{\mathbb{E}}}\Big[\sum_{j\in[t]} f_{B_j}^{\lambda_i}\Big] \leq \prod_{i=1}^{|\lambda|} \underset{B}{\widetilde{\mathbb{E}}}[f_B^{\lambda_i}] \leq \widetilde{\mathbb{E}}[f_B^r].$$

Using the recurrence relation for multidimensional Bell numbers from Lemma 12, we conclude that

$$\mathbb{E}_B\left[\text{cost}(P_B)^r\right] \leq \sum_{\lambda\vdash r} \frac{\binom{r}{\lambda}\prod_{i=1}^{|\lambda|}\mathscr{B}_{\hbar-1}(\lambda_i)}{\prod_{j=1}^{r}\text{count}(j,\lambda)}\,\widetilde{\mathbb{E}}[f_B^r] = \mathscr{B}_{\hbar}(r)\,\widetilde{\mathbb{E}}[f_B^r]. \qquad \blacktriangleleft$$

▶ **Theorem 27.** *Algorithm 1 gives an $(1+\varepsilon)\mathscr{A}_d(p) \stackrel{\text{def}}{=} (1+\varepsilon)\mathscr{B}_d(p)^{1/p}$ approximation for the problem in series-parallel graphs of order d in time polynomial in $n^{O(p)}$ and $1/\varepsilon$.*

Proof. We apply Lemma 26 with $a_e = c_e(i)$ to every coordinate $i \in [\ell]$ and add up the obtained upper bounds on $\mathbb{E}\left[\left(\sum_{e\in P} c_e(i)\right)^p\right]$. We get that

$$\mathbb{E}\left[\Big\|\sum_{e\in P} c_e\Big\|_p^p\right] \leq \mathscr{B}_{\hbar}(p)\,\widetilde{\mathbb{E}}\Big[\sum_{i=1}^{\ell}\Big(\sum_{e} c_e(i)x_e\Big)^p\Big] \leq \mathscr{B}_{\hbar}(p)\cdot \mathsf{OPT}^p.$$

By Markov's inequality, $\left\|\sum_{e\in P} c_e\right\|_p^p \leq (1+\varepsilon)\mathscr{B}_{\hbar}(p)\cdot\mathsf{OPT}^p$ with probability at least $\varepsilon/(1+\varepsilon)$. By running the algorithm $1/\varepsilon$ times, we find a solution of cost at most $\left\|\sum_{e\in P} c_e\right\|_p \leq (1+\varepsilon)^{1/p}\mathscr{B}_{\hbar}(p)^{1/p}\cdot\mathsf{OPT} \leq (1+\varepsilon)\mathscr{A}_{\hbar}(p)\cdot\mathsf{OPT}$ with constant probability. (As is standard, we can run this procedure many times and make the failure probability exponentially small.) ◀

6 Algorithms for ℓ_p-Shortest Path in Arbitrary Graphs

In this section, we describe an approximation algorithm for ℓ_p-Shortest Path in arbitrary graphs. We note that there is a black-box reduction from the problem in arbitrary graphs to that in series-parallel graphs, which is implicitly used by Li, Xu, and Zhang [23] in their algorithm for ℓ_∞-Shortest Path (which they call Robust s-t Path). This reduction outputs a series-parallel graph with $O(n^{\log n})$ vertices, where n is the number of vertices in the original graph. By using this reduction, we immediately get an $O(p\log^{1-1/p} n)$-approximation algorithm for general graphs with running time $n^{O(p\log n)}$. We describe here how to get an approximation algorithm for general graphs with an improved running time and slightly improved approximation factor; namely we describe how to get a $O(cp\log^{1-1/p} n)$-approximation in time $m^{O(ce^{1/c}\log n)}$ for every $c \in (0, 1/2)$. We assume below that ℓ is at most polynomial in n; thus, we may assume $p \le \lceil \log_2 \ell \rceil = O(\log n)$ (since all norms $\|\cdot\|_r$ with $r \ge \log_2 \ell$ are equivalent within a factor of 2).

Layered graphs and reduction from general graphs to layered graphs. We say that a directed acyclic graph (DAG) $G = (V, E)$ is an s-t layered graph with Δ edge layers if V is the disjoint union of vertex layers $V_0, V_1, \dots, V_\Delta$, E is the disjoint union of layers $E_1, \dots, E_\Delta$, and each edge in E_i goes from V_{i-1} to V_i. Further, we require that $V_0 = \{s\}$ and $V_\Delta = \{t\}$.

We transform an arbitrary graph $G = (V_G, E_G)$ with terminals s and t into a layered graph $\hat{G} = (V_{\hat{G}}, E_{\hat{G}})$ with $\Delta = n - 1$ edge layers. We create vertex layers $V_0, V_1, \dots, V_\Delta$: $V_0 = \{s\}$, $V_\Delta = \{t\}$, and each of the other V_is is a disjoint copy of V_G. We connect $\hat{u} \in V_i$ with $\hat{v} \in V_{i+1}$ if there is an edge $(u, v) \in E_G$ between the corresponding vertices in G. The vector cost of $(\hat{u}, \hat{v})$ equals that of (u, v). Additionally, we add *padding* edges between copies of t in adjacent layers and assign these edges cost 0.

For every s-t path P with at most Δ edges in G there is a corresponding path $\hat{P}$ in $\hat{G}$ and vice versa (P might not be a simple path); if path P has $k < \Delta = n - 1$ edges, then $\hat{P}$ contains k non-padding edges and ends with $\Delta - k$ padding edges. Paths P and $\hat{P}$ have the same vector costs. Note that the ℓ_p-Shortest Path P^* between s and t is a simple path and thus contains at most $\Delta = n - 1$ edges. Therefore, there is a path $\hat{P}^*$ in $\hat{G}$ corresponding to P^*. An α-approximation for $\hat{P}^*$ in $\hat{G}$ gives an α-approximation for P^* in G. This reduction shows that it is sufficient to consider layered graphs.

An algorithm for layered graphs. Assume that G is a layered graph with Δ edge layers, source s, and sink t. We use the SoS relaxation from Section 3 for G. Let $a = \lceil e^{1/c} \rceil$. We require that $\widetilde{\mathbb{E}}$ be a pseudo-expectation of degree $2d = 2(p + (a+1)\lceil \log_{a+1} \Delta \rceil) = \Theta(p + ce^{1/c}\log \Delta)$. For a set of edges A of size at most d, we define polynomial $h_A = \sum_{e\in A} x_a$ and conditional pseudo-expectation $\widetilde{\mathbb{E}}_A[\cdot] \stackrel{\text{def}}{=} \widetilde{\mathbb{E}}[\cdot \,|\, h_A]$. Given A and a set of layer indices $I \subseteq [\Delta]$ so that $|A| + |I| \le d$, we define a sampling procedure that samples an edge from each layer E_i with $i \in I$ using pseudo-expectation $\widetilde{\mathbb{E}}_A$. We assume that the two algorithms below have access to graph G and pseudo-expectation $\widetilde{\mathbb{E}}$.

Algorithm 2 Edge sampling procedure.

Input: a subset of layer indices $I \subseteq [\Delta]$ and a subset A of edges.
Output: one edge from every layer E_i with $i \in I$.
$R = \varnothing$
for all $i \in I$ **do**
    Sample $e \in E_i$ with probability of choosing e equal to $\widetilde{\mathbb{E}}_{A\cup R}[x_e]$
    $R = R \cup \{e\}$
end for
return R

We say that we sample edges $e_1, \ldots, e_k$ in layers $i_1, \ldots, i_k$ conditioning on set A to mean that we run Algorithm 2 with parameters $I = \{i_1, \ldots, i_k\}$ and A.

Algorithm 3 Rounding algorithm for layered graphs.

Input: indices y and z of two edge layers ($1 \le y \le z \le \Delta$) and a subset of edges A.
Output: a path in $\hat{G}$ traversing layers E_y to E_z.
function FINDPATH(y, z, A)
  if $z - y + 1 \le a$ **then**
    Sample edges $e_0, \ldots, e_{z-y}$ in layers $E_y, E_{y+1}, \ldots, E_z$ conditioning on A.
    return the path formed by $e_0, \ldots, e_{z-y}$.
  end if
  Let $m_i = y + \lceil \frac{z-y}{a+1} \cdot i \rceil$ for $i \in [a]$.
  Sample edges $e_1, \ldots, e_a$ in layers $m_1, \ldots, m_a$ conditioning on A.
  Let $A' = A \cup \{e_i : i \in [a]\}$.
  Let $m_0 = z - 1$ and $m_{a+1} = y + 1$.
  for $i = 0$ **to** a **do**
    $P_i = \text{FindPath}(m_i + 1, m_{i+1} - 1, A')$ unless $m_i + 1 > m_{i+1} - 1$ then $P_i = \varnothing$
  end for
  Let P be the path formed by $P_0, e_1, P_1, e_2, \ldots, e_a, P_a$.
  return P
end function

To solve the problem, we run Algorithm 3 with $y = 1$, $z = \Delta$, and $A = \varnothing$. The analysis of this algorithm is quite similar to that of Algorithm 1 for series-parallel graphs, with the main difference that instead of the series-parallel decomposition of G, we consider the recursion tree whose nodes correspond to invocations of FindPath. The full analysis of this algorithm can be found in the full version of this paper [24], where the following theorems are shown.

▶ **Theorem 28.** *Algorithm 1 gives an $O(cp \log^{1-1/p} \Delta)$ approximation for the ℓ_p-Shortest Path problem in layered graphs with Δ layers in time $m^{O(p+ce^{1/c} \log \Delta)}$ for $c \in (0, 1/2)$.*

As a corollary, we get the following result for arbitrary graphs.

▶ **Theorem 29.** *There is an $O(cp \log^{1-1/p} n)$ approximation algorithm for the ℓ_p-Shortest Path problem in arbitrary graphs that runs in time $m^{O(p+ce^{1/c} \log n)}$ for $c \in (0, 1/2)$.*

References

1. Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Approximation of min–max and min–max regret versions of some combinatorial optimization problems. *European Journal of Operational Research*, 179(2):281–290, 2007. doi:10.1016/j.ejor.2006.03.023.
2. Boaz Barak, Jonathan A. Kelner, and David Steurer. Rounding sum-of-squares relaxations. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 31–40. ACM, 2014. doi:10.1145/2591796.2591886.
3. Vittorio Bilò, Ioannis Caragiannis, Angelo Fanelli, Michele Flammini, and Gianpiero Monaco. Simple greedy algorithms for fundamental multidimensional graph problems. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 125:1–125:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.125.

4 Thomas Breugem, Twan Dollevoet, and Wilco van den Heuvel. Analysis of fptases for the multi-objective shortest path problem. *Comput. Oper. Res.*, 78:44–58, 2017. doi:10.1016/j.cor.2016.06.022.

5 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *Journal of Algorithms*, 33(1):73–91, 1999. doi:10.1006/jagm.1999.1042.

6 Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median. In *Proceedings of the Symposium on Theory of Computing*, pages 114–123, 1998. doi:10.1145/276698.276719.

7 Chandra Chekuri and Martin Pál. A recursive greedy algorithm for walks in directed graphs. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 245–253, 2005. doi:10.1109/SFCS.2005.9.

8 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 575–584, 2010. doi:10.1109/FOCS.2010.60.

9 Julia Chuzhoy and Sanjeev Khanna. Hardness of directed routing with congestion. *Electron. Colloquium Comput. Complex.*, TR06-109(109), 2006. arXiv:TR06-109.

10 Jesús de la Cal and Javier Cárcamo. Set partitions and moments of random variables. *Journal of mathematical analysis and applications*, 378(1):16–22, 2011.

11 Matthias Ehrgott. *Multicriteria Optimization (2. ed.)*, volume 491. Springer, 2005. doi:10.1007/3-540-27659-9.

12 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Symposium on Theory of Computing*, pages 448–455, 2003. doi:10.1145/780542.780608.

13 Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1-2):1–221, 2019. doi:10.1561/0400000086.

14 Naveen Garg, Goran Konjevod, and Ramamoorthi Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000. doi:10.1006/jagm.2000.1096.

15 Horst W Hamacher and Günter Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52(4):209–230, 1994. doi:10.1007/BF02032304.

16 Pierre Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979*, pages 109–127. Springer, 1980.

17 Jovan Karamata. Sur une inégalité relative aux fonctions convexes. *Publications de l'Institut mathematique*, 1(1):145–147, 1932.

18 Adam Kasperski and Paweł Zieliński. On the approximability of minmax (regret) network optimization problems. *Information Processing Letters*, 109(5):262–266, 2009. doi:10.1016/j.ipl.2008.10.008.

19 Adam Kasperski and Paweł Zieliński. On the approximability of robust spanning tree problems. *Theoretical Computer Science*, 412(4-5):365–374, 2011. doi:10.1016/j.tcs.2010.10.006.

20 Adam Kasperski and Paweł Zieliński. Robust discrete optimization under discrete and interval uncertainty: A survey. *Robustness analysis in decision aiding, optimization, and analytics*, pages 113–143, 2016.

21 Adam Kasperski and Pawel Zielinski. Approximating some network problems with scenarios. *CoRR*, abs/1806.08936, 2018. doi:10.48550/arXiv.1806.08936.

22 Aditi Laddha, Mohit Singh, and Santosh S Vempala. Socially fair network design via iterative rounding. *Operations Research Letters*, 50(5):536–540, 2022. doi:10.1016/j.orl.2022.07.011.

23 Shi Li, Chenyang Xu, and Ruilong Zhang. Polylogarithmic approximation for robust s-t path. *CoRR*, abs/2305.16439, 2023. doi:10.48550/arXiv.2305.16439.

24 Yury Makarychev, Max Ovsiankin, and Erasmo Tani. Approximation algorithms for ℓ_p-shortest path and ℓ_p-group steiner tree, 2024. arXiv:2404.17669.

25 Ernesto Queiros Vieira Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245, 1984.

26 Robert Franklin Muirhead. Some methods applicable to identities and inequalities of symmetric algebraic functions of n letters. *Proceedings of the Edinburgh Mathematical Society*, 21:144–162, 1902.

27 Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 80:1–80:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.80.

28 R. Ravi, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Harry B. Hunt III. Many birds with one stone: multi-objective approximation algorithms. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 438–447. ACM, 1993. doi:10.1145/167088.167209.

29 Gabriele Reich and Peter Widmayer. Beyond steiner's problem: A vlsi oriented generalization. In *International Workshop on Graph-theoretic Concepts in Computer Science*, pages 196–210. Springer, 1989. doi:10.1007/3-540-52292-1_14.

30 Stefan Ruzika and Horst W Hamacher. A survey on multiple objective minimum spanning tree problems. In *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*, pages 104–116. Springer, 2009. doi:10.1007/978-3-642-02094-0_6.

31 Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.

32 Richard Stanley. *Enumerative Combinatorics: Volume 2*. Cambridge University Press, 2023.