Iterative Refinement for ℓ_p-norm Regression *

Deeksha Adil† Rasmus Kyng‡ Richard Peng§ Sushant Sachdeva ¶

January 23, 2019

Abstract

We give improved algorithms for the ℓ_p-regression problem, $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_p$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, for all $p \in (1,2) \cup (2,\infty)$. Our algorithms obtain a high accuracy solution in $\widetilde{O}_p(m^{\frac{|p-2|}{2p+|p-2|}}) \leq \widetilde{O}_p(m^{1/3})$ iterations, where each iteration requires solving an $m \times m$ linear system, with m being the dimension of the ambient space.

Incorporating a procedure for maintaining an approximate inverse of the linear systems that we need to solve at each iteration, we give algorithms for solving ℓ_p-regression to $1/\text{poly}(n)$ accuracy that runs in time $\widetilde{O}_p(m^{\max\{\omega,7/3\}})$, where ω is the matrix multiplication constant. For the current best value of $\omega > 2.37$, this means that we can solve ℓ_p regression as fast as ℓ_2 regression, for all constant p bounded away from 1.

Our algorithms can be combined with nearly-linear time solvers for linear systems in graph Laplacians to give minimum ℓ_p-norm flow / voltage solutions to $1/\text{poly}(n)$ accuracy on an undirected graph with m edges in $\widetilde{O}_p(m^{1+\frac{|p-2|}{2p+|p-2|}}) \leq \widetilde{O}_p(m^{4/3})$ time.

For sparse graphs and for matrices with similar dimensions, our iteration counts and running times improve upon the p-norm regression algorithm by [Bubeck-Cohen-Lee-Li STOC'18], as well as general purpose convex optimization algorithms. At the core of our algorithms is an iterative refinement scheme for ℓ_p-norms, using the quadratically-smoothed ℓ_p-norms introduced in the work of Bubeck *et al.* Formally, given an initial solution, we construct a problem that seeks to minimize a quadratically-smoothed ℓ_p norm over a subspace, such that a crude solution to this problem allows us to improve the initial solution by a constant factor, leading to algorithms with fast convergence.

*This paper has been published at SODA 2019 [Adi+], and was initially submitted to SODA on July 12, 2018.

†University of Toronto. deeksha@cs.toronto.edu. Supported by an Ontario Graduate Scholarship, and by a Connaught New Researcher award to Sushant Sachdeva.

‡Harvard. rjkyng@gmail.com. Supported by ONR grant N00014-18-1-2562.

§Georgia Tech. richard.peng@gmail.com. Supported in part by the National Science Foundation under Grant No. 1718533.

¶University of Toronto. sachdeva@cs.toronto.edu. Research supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and a Connaught New Researcher award.

Contents

1 Introduction

Iterative methods that converge rapidly to a solution are of fundamental importance to numerical analysis, optimization, and more recently, graph algorithms. In the study of iterative methods, there are significant discrepancies between iterative methods geared towards linear problems, and ones that can handle more general convex objectives. For systems of linear equations, which corresponds to minimizing ℓ_2-norm objectives over a subspace, most iterative methods obtain ϵ-approximate solutions in iteration counts that scale as $\log(1/\epsilon)$. More generally, for appropriately defined notions of accuracy, a constant-accuracy linear system solver can be iterated to give a much higher accuracy solver using a few calls to the crude solver. Such phenomena are not limited to linear systems either: an algorithm that produces approximate maximum flows on directed graphs can be iterated on the residual graph to quickly obtain high-accuracy answers.

On the other hand, for the much wider space of non-linear optimization problems arising from optimization and machine learning, it's significantly more expensive to obtain high accuracy solutions. Many widely used methods such as (accelerated) gradient descent, obtain ϵ-approximate answers using iteration counts that scale as $\text{poly}(1/\epsilon)$. Such discrepancies also occur in the overall asymptotic running times. An important and canonical problem in this space is ℓ_p-norm regression:

$$\min_{\boldsymbol{x}\in\mathbb{R}^m:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{x}\|_p^p, \tag{*}$$

for some $\boldsymbol{A}\in\mathbb{R}^{n\times m}(m\geq n)$, and $\boldsymbol{b}\in\mathbb{R}^n$. For $p=2$, this corresponds exactly to solving a linear system, and hence is solvable by a matrix inversion in $O(m^\omega)$ time [1] For $p=1$ and $p=\infty$, this problem is inter-reducible to linear programming [Bub+18; Til13; Til15].

Interior point methods also allow us to solve ℓ_p-norm regression problems in $\sqrt{rank}$ iterations [LS14; NN94], where each iteration requires solving an $m\times m$ linear system for any $p\in[1,\infty]$. Bubeck *et al* [Bub+18] show that this iteration count is tight for the interior point method framework, and instead propose a different method which requires only $\widetilde{O}_p(m^{\left|\frac{1}{2}-\frac{1}{p}\right|})$ [2] iterations for $p\neq 1,\infty$, which for large constant p still tends to about $m^{1/2}$. On the other hand, ϵ-approximate solutions can be computed in about $m^{1/3}\text{poly}(1/\epsilon)$ iterations [Chi+13] [3].

Furthermore, this discrepancy also carries over to the graph theoretic case. If the matrix $\boldsymbol{A}$ is the vertex-edge incidence matrix of a graph, then this problem captures graph problems such as p-norm Lipschitz learning and finding ℓ_p-norm minimizing flows meeting demands given by $\boldsymbol{b}$. Here low accuracy approximate solutions can be obtained in nearly-linear time when $p=\infty$ [Pen16; She17a], and almost-linear time for all other values of p [She17b; Sid17]. However, the current best high accuracy solutions take at least $m^{\min\{10/7,1+\left|1/2-1/p\right|\}}$ time [Bub+18; Mad13].

1.1 Contributions

Iterative Refinement for ℓ_p-norms. In this paper, we propose a new iterative method for ℓ_p-norm regression problems (*) that achieves geometric convergence to the optimal solution. Our method only requires solving $O_p(\log 1/\varepsilon)$ *residual problems* to find an ε-approximate solution, or $O_p(\kappa\log 1/\varepsilon)$ residual problems, each solved to a κ-approximation factor. Such an iterative method

[1] ω is the matrix multiplication exponent. Currently we know $\omega\leq 2.3728639..$ [Le 14; Wil12].

[2] $O_p(\cdot)$ notation hides constant factors that depend on p, and its dual norm $\frac{p}{p-1}$. $\widetilde{O}_p(\cdot)$ notation also hides $\text{poly}(\log\frac{mn}{\varepsilon})$ factors in addition.

[3] this result only addressed the $p=\infty$ case, but its techniques generalize to all other p

was previously known only for $p = 2$ and ∞. Curiously, our residual problems look very similar to the original problem (*), with the ℓ_p norms replaced by their quadratically-smoothed versions introduced by Bubeck *et al* [Bub+18]. This result, Theorem 4.1, can be stated informally as:

Theorem 1.1. *There exists a class of residual problems for p-norm regression (which we will define in Definition 4.3) such that any p-norm regression problem can be solved to ϵ-relative accuracy by solving to relative error κ a sequence of $O_p(\kappa \log(\frac{m}{\varepsilon}))$ residual problems.*

Improved Iteration Count for ℓ_p-Regression. We then give an algorithm for quickly solving the residual problem motivated by the approximate maximum flow by electrical flows algorithm by Christiano et al. [Chr+11] and its generalizations to regression problems [Chi+13]. This is given as Theorem 5.1, and can be stated informally as:

Theorem 1.2. *For any $p > 2$, an instance of a residual problem for p-norm regression as defined in Definition 4.3 can be solved in $\widetilde{O}_p(m^{\frac{p-2}{3p-2}})$ iterations, each of which consist of solving a system of linear equations plus updates that take linear time.*

This improves on the work of Bubeck *et al* [Bub+18] for all $p > 2$, with the number of iterations equaling $\widetilde{O}_p(1)$ for $p = 2$ (essentially the same as Bubeck *et al*) and tending to $\widetilde{O}(m^{1/3})$ as p goes to ∞ (compared to $\widetilde{O}(m^{1/2})$ for Bubeck *et al*). However, our results don't give anything for $p = \infty$ due to the dependency in p in the $\widetilde{O}_p(\cdot)$ term. It's worth noting that even in the constant error regime, this improves by a factor of about $\min\{m^{\frac{(p-2)^2}{2p(3p-2)}}, m^{\frac{4}{3(3p-2)}}\}$ over the current state of the art, which for small p is due to Bubeck et al. [Bub+18], and for large p is based on unpublished modifications to Christiano et al. [Chr+11; Mad11].

A Duality Based Approach to ℓ_p-Regression. For the remaining case of $1 < p < 2$, we instead solve the dual problem, which is a $\frac{p}{p-1}$-norm regression problem, and utilize its solution to solve our original ℓ_p-Regression. This leads to iteration counts of the form $\widetilde{O}_p(m^{\frac{2-p}{p+2}} \log(1/\epsilon))$ for solving such problems. Note that this result also does not give anything when $p = 1$, as the constants related to its dual norm, $\frac{p}{1-p}$ become prohibitive. For all $p \in (1, \infty)$, our iteration count achieves the following exponent on m,

$$\frac{\left|\frac{1}{2} - \frac{1}{p}\right|}{1 + \left|\frac{1}{2} - \frac{1}{p}\right|},$$

while the exponent from the previous result [Bub+18] is $\left|\frac{1}{2} - \frac{1}{p}\right|$: our algorithm has better dependence on m on all constant p (albeit with larger constants depending on p).

For the case of $p = 4$, a manuscript by Bullins [Bul18] from December 2018 (after our paper was accepted to SODA 2019, but independently developed), gives the same iteration count as our algorithm of $n^{1/5} \log(1/\epsilon)$ up to polylogs. Bullins' approach requires a linear system solve per iteration, similar to our approach when implemented without inverse maintenance. Bullins' algorithm is based on higher-order acceleration, and the agreement between running times suggests there may be a strong connection between our "accelerated" multiplicative weight method and his accelerated gradient-based method.

Faster ℓ_p-Regression. Our improved iteration counts can be readily combined with methods for speeding up optimization algorithms that utilize linear system solvers, including inverse maintenance [LS15; Vai89]. This results in an $\widetilde{O}_p(m^{\max\{\omega,7/3\}})$ time algorithm for solving ℓ_p regression problems for all $p \in (1,\infty)$, which we formalize in Theorem 6.1.

This bound for p-norm regression with general matrices brings us to the somewhat surprising conclusion that for the current value of $\omega > 7/3$, p-norm regression problems (with constant p that's also constant-bounded away from 1) on square matrices can be solved as fast as solving the underlying linear systems, or equivalently, ℓ_2 regression problems.

This is based on maintaining an approximate inverse to the linear systems we need to solve in each step of the iterative method as pioneered by Vaidya [Vai89]. However, our modification interacts directly with the potential functions we use to control iteration counts in the inner loop of our iterative method. A similar approach for maintaining an approximate inverse was used by Cohen et al. [CLS18] to give an $\tilde{O}(m^{\omega})$ algorithm for Linear Programming, after our initial submission to SODA, but before our paper was publicly available. Both works build on ideas developed by Cohen, see [Lee17].

Faster p-Norm Flows. When solving p-norm flow problems, our algorithm can made faster by using Laplacian solvers for graph problems [Ten10; Vai90] to solve the linear equations that arise during our iterations. This gives algorithms for finding p-norm flows on undirected graphs to accuracy ϵ with running time $\widetilde{O}_p\left(m^{1+\frac{|p-2|}{2p+|p-2|}}\log(1/\epsilon)\right)$ for $p \in (1,\infty)$ via direct invocations of fast Laplacian solvers [ST14].

Our results thus give the first evidence that wide classes of graph optimization problems can be solved in time $m^{4/3}$ or faster. While such a bound (via. fast Laplacian solvers) is by now well-known in the approximate setting [Chr+11], the $m^{10/7}$ iteration bounds due to Madry [Mad13; Mad16] represent the only results to date in this direction for high accuracy answers on sparse graphs.

Generalizations and Extensions. While we focus on Problem (*), under mild assumptions about polynomially bounded objectives, we can solve the following more general problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{d}\|_p \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

The reduction is discussed in Section 7. The combination of an affine constraint on $\boldsymbol{x}$ with an affine transformation in the p-norm objective means we can solve most variants of p-norm optimization problems.

Similar ideas can be used to solve p-norm Lipschitz learning problems [Kyn+15] on graphs quickly.

1.2 Comparison to Previous Works

Numerical Methods and Preconditioning. Iterative methods and preconditioning are the most fundamental tools in numerical algorithms [Axe94; Saa03]. As studies of such methods often focus on linear problems, many existing analyses of iterative methods are restricted to linear systems. Generalizing such methods, as well as numerical methods, to broader settings is a major topic of study [Hen03; Kel99; KK04; NW06].

The study of more efficient algorithms for combinatorial flow problems has benefited enormously from ideas from linear and non-linear preconditioning. Recent advances in approximate maximum flow and transshipment algorithms [Gha+15; Kel+14; LRS13; Pen16; RST14; She13; She17a; She17b] build upon such ideas. However, these methods rely on the preconditioner being a linear operator, and give poly$(1/\epsilon)$ dependence.

Optimization Algorithms. Our techniques for solving the residual problems are directly motivated by approximating maximum flow using electrical flows [Chr+11]. While this algorithm has been extended to multicommodity flows and regression problems [Chi+13; KMP12], all these results have poly$(1/\epsilon)$ dependencies.

Several recent results for obtaining $\log(1/\epsilon)$ dependencies are all motivated by convex optimization techniques. In particular, the state of the art running times are by interior point methods. These include directly modifying the interior point method (IPM) [Kyn+15; LS14; LS15], combining techniques from the electrical flow algorithms with IPM update steps [Coh+17a; KRS15; Kyn+15; Mad13; Mad16], and increasing the 'confidence interval', and in turn step lengths, of the IPM update steps [All+17; Bub+18; Coh+17b]. Our result based on creating intermediate problems has the most in common with the last of these. However, our method differs in that our guarantees for this intermediate problem holds over the entire space.

Inverse Maintenance Our final running time of $\widetilde{O}_p(m^{\max\{\omega,7/3\}})$ for ℓ_p-regression incorporates inverse maintenance. This is a method introduced by Vaidya [Vai89] for speeding up optimization algorithms for solving minimum cost and multicommodity flows. It takes advantage of the controllable rate at which optimization algorithms modify the solution variables to reuse inverses of matrices constructed from such variables.

Previous studies of inverse maintenance [LS14; LS15; Vai89] have been geared towards the interior point method. Here the norm per update step can be controlled, and we believe this also holds for their applications in faster cutting plane methods [LSW15]. While such methods also give gains in the case of our algorithm, for the final bound of about m^ω, we instead bound the progress of the steps against a global potential function motivated by the electrical flow max-flow algorithm [Chr+11].

Speedups for Matrices with Uneven Dimensions Our algorithm on the other hand does not take into account sparsity of the input matrix, or possibly uneven dimensions (e.g. $m \approx n^2$)). In these settings, the methods based on accelerated stochastic gradient descent from [Bub+18] obtain better performances. On the other hand, we believe our methods have the potential of extending to such settings by combining the intermediate problems with ℓ_p row sampling [CP15]. However, analysis of such row sampling routines for our residual problems containing mixed ℓ_2 and ℓ_p norm functions is outside the scope of this paper.

2 Technical Overview

Iterative Refinement for ℓ_p-norms. To design their algorithm for ℓ_p-norm regression, Bubeck *et al* [Bub+18] construct a function $\gamma_p(t, x)$, which is C_1, [4], quadratic in the range $|x| \leq t$, and behaves

[4] A function is said to be C_1 if it's continuous, differentiable, and has a continuous derivative

as $|x|^p$ asymptotically (see Def. 3.1). Our key lemma states one can locally approximate $\|x+\Delta\|_p^p$ as a linear function plus a $\gamma_p(|x|,\Delta)$ "error" term [5] (Lemma 4.5):

$$|x+\Delta|^p = |x|^p + \Delta\frac{d}{dx}|x|^p + O_p(1)\gamma_p(|x|,\Delta).$$

Surprisingly, this approximation only has an $O_p(1)$ "condition number". Proceeding just as for gradient descent, or Newton's method, means that if at each step we solve the following local approximation problem to a factor κ,

$$\max_{\boldsymbol{A}\Delta=0} \boldsymbol{g}^\top\Delta - \gamma_p(\boldsymbol{x},\Delta),$$

where $\boldsymbol{g}$ is the gradient of our loss function, we can converge to an ε-approximate solution in roughly $\kappa\log{}^1\!/_\varepsilon$ iterations (Theorem 4.1).

Improved Algorithms for ℓ_p-regression for $p \geq 2$. The key advantage afforded by our iterative algorithm is that we now only need to design a algorithm for the residual problem that achieves a crude approximation factor (we achieve $O_p(1)$). As a first step, by a binary search and some rescaling, we show (Lemma 5.5) that it suffices to achieve a constant factor approximation to $O_p(\log{}^m\!/_\varepsilon)$ problems of the following form,

$$\min_{\boldsymbol{A}\boldsymbol{x}=0,\boldsymbol{g}^\top\boldsymbol{x}=c} \gamma_p(\boldsymbol{t},\boldsymbol{x}).$$

The technical heart of our proof is to give an algorithm (Gamma-Solver, Algorithm 4) inspired by the multiplicative weight update (MWU) method (see [AHK12] for a survey), combined with the width-reduction inspired by the faster flow algorithm of Christiano *et al.* [Chr+11], and its matrix version by Chin *et al.*[Chi+13]. At each iteration, we solve a weighted ℓ_2 minimization problem to find the next update step. If this update step has small ℓ_p norm, we add this to our current solution, and update the weights. Otherwise, we identify a set of coordinates that have small current weights, and yet are contributing most of the ℓ_p norm, and we penalize them by increasing their weights (and do not add our update step to the current solution). Setting the parameters carefully, we show that after $\widetilde{O}_p(m^{\frac{p-2}{3p-2}})$ iterations, the average of the update steps achieves an $O_p(1)$-approximation to our modified residual problem (Theorem 5.8). Combining this with our iterative refinement algorithm, we obtain our algorithms for ℓ_p-norm regression that require only $\widetilde{O}_p(m^{\frac{p-2}{3p-2}})$ iterations (or linear system solves).

Maintaining Inverses for Improved Algorithm. Our inverse maintenance procedure utilizes the same combination of low-rank updates and matrix multiplications as in previous results [LS14; LS15; Vai89]. However, the rate of convergence of our algorithm, and in turn the rate at which we adjust the weights from the MWU procedure, are governed by growths in the ℓ_2 minimization problem. This leads to the difficulty of uneven progress across the iterations.

We solve this issue by a simple yet subtle scheme motivated by lazy updates in data structures [Abr+16; GP13]. We bucket changes to the values of entries based on their magnitudes, and update entries that received too many updates of a certain magnitude separately. This differs with previous methods that update weights exceeding approximation thresholds as they happen, and enables a closer interaction with the overall potential function based convergence analysis.

[5] It is useful to compare the γ_p term to the second-order Hessian term in Taylor expansion

3 Preliminaries

We use the following family of functions, $\gamma_p(t,x)$ defined in [Bub+18].

Definition 3.1 (γ_p function)**.** *For $t \geq 0$ and $p \geq 1$, define*

$$\gamma_p(t,x) = \begin{cases} \frac{p}{2}t^{p-2}x^2 & \text{if } |x| \leq t, \\ |x|^p + (\frac{p}{2}-1)t^p & \text{otherwise .} \end{cases}$$

These functions can be thought of a quadratic approximation of $|x|^p$ in a small range around zero. The following properties follow directly from the definition.

1. $\gamma_p(0,x) = |x|^p$.
2. $\gamma_p(t,x)$ is quadratic in the range $-t \leq x \leq t$.
3. γ_p is C^1 in both x, t.

We show several other important properties of γ_p in the following lemmas. Their proofs are straightforward and deferred to Appendix A.1

Lemma 3.2. *Function γ_p is as defined above.*

1. *For any $p \geq 2$, $t \geq 0$, and $x \in \mathbb{R}$, we have $\gamma_p(t,x) \geq |x|^p$, and $\gamma_p(t,x) \geq \frac{p}{2}t^{p-2}x^2$.*
2. *It is homogeneous under rescaling of* both *t and x, i.e., for any $t, \lambda \geq 0, p \geq 1$, and any x we have $\gamma_p(\lambda t, \lambda x) = \lambda^p \gamma_p(t,x)$.*
3. *For any $t > 0, p \geq 1$ and any x, we have $\gamma_p'(t,x) = p \max\{t,|x|\}^{p-2}x$.*

The next lemma shows a bound on the value of γ_p when x is scaled up or down.

Lemma 3.3. *For any $p > 1, \Delta \in \mathbb{R}$ and $\lambda \geq 0$, we have,*

$$\min\{2,p\} \leq x\frac{\gamma_p'(t,x)}{\gamma_p(t,x)} \leq \max\{2,p\}.$$

This implies,

$$\min\{\lambda^2,\lambda^p\}\gamma_p(t,\Delta) \leq \gamma_p(t,\lambda\Delta) \leq \max\{\lambda^2,\lambda^p\}\gamma_p(t,\Delta).$$

The following lemma allows us to bound the second order change in $\gamma_p(\boldsymbol{x})$ as $\boldsymbol{x}$ changes to $\boldsymbol{x}+\Delta$.

Lemma 3.4. *For any $p \geq 2, t \geq 0$ and any x, Δ, we have*

$$\gamma_p(t, x+\Delta) \leq \gamma_p(t,x) + \left|\gamma_p'(t,x)\Delta\right| + p^2 2^{p-3}\max\{t,|x|,|\Delta|\}^{p-2}\Delta^2.$$

Notation. For a vector $\boldsymbol{x}$, let $|\boldsymbol{x}|$ denote the vector with its i^{th} coordinate as $|\boldsymbol{x}_i|$. For any two vectors $\boldsymbol{t}$ and $\boldsymbol{x}$, $\gamma_p(\boldsymbol{t},\boldsymbol{x})$ denotes the sum $\sum_i \gamma_p(\boldsymbol{t}_i,\boldsymbol{x}_i)$.

4 Main Iterative Algorithm

In this section we analyze procedure p-Norm, i.e., Algorithm 1. Our main result for this section is,

Theorem 4.1 (ℓ_p-norm Iterative Refinement)**.** *For any $p \in (1, \infty)$, and $\kappa \geq 1$. Given an initial feasible solution $\boldsymbol{x}^{(0)}$ (Definition 4.7) to our optimization problem (Equation (*)), Algorithm 1 finds an ϵ-approximate solution to (*) in $O_p\left(\kappa \log\left(\frac{m}{\varepsilon}\right)\right)$ calls to a κ-approximate solver for the residual problem (Equation (1)).*

Algorithm 1 Meta-Iterative Algorithm

procedure p-Norm($\boldsymbol{A}, \boldsymbol{b}, \varepsilon$)
  $\boldsymbol{x}^{(0)} \leftarrow \min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{x}\|_2^2$.
  $T \leftarrow O_p(\kappa \log\left(\frac{m}{\varepsilon}\right))$
  $\lambda \leftarrow \left(\frac{p-1}{p4^p}\right)^{\frac{1}{\min(1,p-1)}}$
  for $t = 0$ to T **do**
    $\Delta \leftarrow \kappa$-Approx($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}^{(t)}$), $\lambda, \boldsymbol{x}^{(0)}$ ▷ κ-approximate solution to (1)
    $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} - \lambda\Delta$
    if $\|\boldsymbol{x}^{(t+1)}\|_p^p \geq \|\boldsymbol{x}^{(t)}\|_p^p$ **then**
      return $\boldsymbol{x}^{(t)}$
  return $\boldsymbol{x}^{(T)}$

The theorem says that it is sufficient to solve an instance of the residual problem (1) crudely, and only a logarithmic number of times. Before we prove the theorem, we define the terms used in the statement and prove some results that would be needed for the proof. We begin by defining an ε-approximate solution to our main optimization problem.

Definition 4.2 (ε-approximate solution)**.** *We say our solution $\boldsymbol{x}$ is an ε-approximate solution to (*) if $\boldsymbol{Ax} = \boldsymbol{b}$ and*

$$\|\boldsymbol{x}\|_p^p \leq (1+\varepsilon)\|\boldsymbol{x}^\star\|_p^p,$$

where $\boldsymbol{x}^\star$ is the OPT *of (*).*

We next define what we use as our residual problem and what we mean by a κ-approximate solution.

Definition 4.3 (Residual Problem)**.** *For any given $\boldsymbol{x}$ and $p > 1$, let*

$$\alpha(\Delta) \stackrel{\text{def}}{=} \langle \boldsymbol{g}, \Delta \rangle - \frac{p-1}{p2^p}\gamma_p(|\boldsymbol{x}|, \Delta),$$

where $\boldsymbol{g}$ is the gradient, $\boldsymbol{g} = p\,|\boldsymbol{x}|^{p-2}\,\boldsymbol{x}$. We call the following problem to be the residual problem *of (*) at $\boldsymbol{x}$.*

$$\max_{\boldsymbol{A}\Delta=0} \alpha(\Delta). \tag{1}$$

Definition 4.4 (κ-approximate solution)**.** *Let $\kappa \geq 1$. A κ-approximate solution for the residual problem is $\tilde{\Delta}$ such that $\boldsymbol{A}\tilde{\Delta} = 0$ and, $\alpha(\tilde{\Delta}) \geq \frac{1}{\kappa}\alpha(\Delta^\star)$. Here $\Delta^\star = \max_{\boldsymbol{A}\Delta=0} \alpha(\Delta)$.*

In order to see why we choose this problem as our residual problem we show that the objective of the residual problem bounds the change in p-norm of a vector $\boldsymbol{x}$ when perturbed by Δ (Lemma 4.6).

Lemma 4.5. *Let $p \in (1, \infty)$. Then for any $\boldsymbol{x}$ and any Δ,*

$$|x|^p + g\Delta + \frac{p-1}{p2^p}\gamma_p(|x|, \Delta) \leq |x + \Delta|^p \leq |x|^p + g\Delta + 2^p\gamma_p(|x|, \Delta),$$

where $g = p\,|x|^{p-2}\,x$ is the derivative of the function $|x|^p$.

The proof can be found in Appendix A.2.

Lemma 4.6. *Let $p \in (1, \infty)$ and λ be such that $\lambda^{\min\{1,p-1\}} \leq \frac{p-1}{p4^p}$. Then for any Δ we have,*

$$\|\boldsymbol{x}\|_p^p - \alpha(\lambda\Delta) \leq \|\boldsymbol{x} - \lambda\Delta\|_p^p \leq \|\boldsymbol{x}\|_p^p - \lambda\alpha(\Delta).$$

Proof. Applying lemma 4.5 to all the coordinates, we obtain,

$$\|\boldsymbol{x}\|_p^p - \langle g, \Delta\rangle + \frac{p-1}{p2^p}\gamma_p(|\boldsymbol{x}|, \Delta) \leq \|\boldsymbol{x} - \Delta\|_p^p \leq \|\boldsymbol{x}\|_p^p - \langle g, \Delta\rangle + 2^p\gamma_p(|\boldsymbol{x}|, \Delta).$$

Using definition 4.3, equation (4) directly implies, $\|\boldsymbol{x} - \Delta\|_p^p \geq \|\boldsymbol{x}\|_p^p - \alpha(\Delta)$ for all Δ. Now to prove the other side, note that for any $\lambda \in [0, 1]$, and any Δ, we have from Lemma 4.5 and Lemma 3.2

$$\|\boldsymbol{x} - \lambda\Delta\|_p^p \leq \|\boldsymbol{x}\|_p^p - \langle g, \lambda\Delta\rangle + 2^p\gamma_p(|\boldsymbol{x}|, \lambda\Delta) \leq \|\boldsymbol{x}\|_p^p - \lambda\left(\langle g, \Delta\rangle - \lambda^{\min\{1,p-1\}}2^p\gamma_p(|\boldsymbol{x}|, \Delta)\right).$$

Picking λ such that $\lambda^{\min\{1,p-1\}} \leq \frac{p-1}{p4^p}$, we obtain that for any Δ,

$$\|\boldsymbol{x} - \lambda\Delta\|_p^p \leq \|\boldsymbol{x}\|_p^p - \lambda\left(\langle g, \Delta\rangle - \frac{p-1}{p2^p}\gamma_p(|\boldsymbol{x}|, \Delta)\right) = \|\boldsymbol{x}\|_p^p - \lambda\alpha(\Delta),$$

thus concluding the proof of the lemma. □

For any iterative algorithm we need a starting feasible solution. We could potentially start with any feasible solution but we define the following starting solution which we claim is a *good* starting point. Lemma 4.8 shows us that our chosen starting point is only polynomially away from the optimum solution, and is thus a *good* choice. The proof of the lemma can be found in Appendix A.2.

Definition 4.7 (Initial Solution)**.** *We define $\boldsymbol{x}^{(0)}$ to be our initial feasible solution to be $\boldsymbol{x}^{(0)} = \min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \|\boldsymbol{x}\|_2^2$.*

Lemma 4.8. *For $\boldsymbol{x}^{(0)}$ as defined in Definition 4.7, $\|\boldsymbol{x}^{(0)}\|_p^p \leq m^{(p-2)/2}\mathrm{OPT}$.*

We are now ready to prove Theorem 4.1.

Proof. Let $\widetilde{\Delta}$ denote the solution returned by the κ-approximate solver. We know that $\alpha(\widetilde{\Delta}) \geq \frac{1}{k} \cdot \alpha(\Delta^\star)$. We have,

$$\alpha(\widetilde{\Delta}) \geq \frac{1}{\kappa} \cdot \alpha(\Delta^\star) \geq \frac{1}{\kappa}\alpha(\boldsymbol{x} - \boldsymbol{x}^\star) \geq \frac{1}{\kappa}\left(\|\boldsymbol{x}\|_p^p - \left\|\boldsymbol{x}^\star\right\|_p^p\right) = \frac{1}{\kappa}\left(\|\boldsymbol{x}\|_p^p - \mathrm{OPT}\right).$$

From Lemma 4.6, for $\lambda = \left(\frac{p-1}{p4^p}\right)^{\frac{1}{\min\{1,p-1\}}} = \Omega_p(1)$, we get,

$$\left\|\boldsymbol{x} - \lambda\widetilde{\Delta}\right\|_p^p \leq \|\boldsymbol{x}\|_p^p - \lambda\alpha(\widetilde{\Delta}).$$

Combining the above two equations and subtracting OPT from both sides gives us

$$\begin{aligned}\left\|\boldsymbol{x} - \lambda\widetilde{\Delta}\right\|_p^p - \mathrm{OPT} &\leq -\lambda\alpha(\widetilde{\Delta}) + \|\boldsymbol{x}\|_p^p - \mathrm{OPT} \\ &\leq -\frac{\lambda}{\kappa}\left(\|\boldsymbol{x}\|_p^p - \mathrm{OPT}\right) + \left(\|\boldsymbol{x}\|_p^p - \mathrm{OPT}\right) \\ &\leq \left(1 - \frac{\lambda}{\kappa}\right)\left(\|\boldsymbol{x}\|_p^p - \mathrm{OPT}\right).\end{aligned}$$

Using lemma 4.8 we get,

$$\boldsymbol{x}^{(t)} - \mathrm{OPT} \leq \left(1 - \frac{\lambda}{\kappa}\right)^t \left(\boldsymbol{x}^{(0)} - \mathrm{OPT}\right) \leq \left(1 - \frac{\lambda}{\kappa}\right)^t \left(m^{\frac{p-2}{2}} - 1\right)\mathrm{OPT}.$$

Setting $t = O_p\left(\kappa \log\left(\frac{m}{\varepsilon}\right)\right)$ gives us an ε-approximate solution. □

This concludes the discussion on the analysis of Algorithm 1. In the following sections we move on to analyzing how to solve the residual problem approximately.

5 Solving the Residual Problem

In this section, we give an algorithm that solves the residual problem to a constant approximation. Combined with the iterative refinement scheme from Theorem 4.1, we obtain the following result.

Theorem 5.1. *For $p \geq 2$, we can find an ε-approximate solution to (*) in time*

$$\widetilde{O}_p\left((m+n)^{\omega + \frac{p-2}{3p-2}} \log^2 1/\varepsilon\right).$$

Here ω is the matrix multiplication constant.

Recall that the residual problem

$$\max_{\boldsymbol{A}\Delta = 0} \boldsymbol{g}^\top \Delta - \frac{p-1}{p2^p}\gamma_p\left(\boldsymbol{t}, \Delta\right),$$

has a linear term followed by the γ_p function. Instead of directly optimizing this function, we guess an approximate value of the linear term, and for each such guess, we minimize the γ_p function under this additional constraint. We can scale the problem so that the optimum is at most 1. Finally,

we can perturb $\boldsymbol{t}$ so that each $\boldsymbol{t}_i$ lies in a polynomially bounded range without adding significant error. Our final problem looks as follows,

$$\begin{aligned} \min_{\Delta} \quad & \gamma_p(\boldsymbol{t}, \Delta) \\ & \boldsymbol{A}\Delta = 0, \\ & \boldsymbol{g}^\top \Delta = c, \end{aligned} \tag{2}$$

with $m^{-1/p} \leq \boldsymbol{t}_i \leq 1, \forall i$.

To sumarise, κ-Approx (Algorithm 2) formalizes this process and shows that we only need to solve a logarithmic number of instances of the above program, (2) and solving each to a κ-approximation gives a $\Omega_p\left(\kappa^{1/(\min\{2,p\}-1)}\right)$-approximate solution to (1). Gamma-Solver (Algorithm 4) solves problem (2) to an $O_p(1)$ approximation. Therefore, using Gamma-Solver as a subroutine for κ-Approx we get an $O_p(1)$ approximate solution to (1). Section 5.1 gives an analysis for κ-Approx. In Section 5.2, we give an oracle that is used in Gamma-Solver. We give an analysis of Gamma-Solver in Section 5.3. Finally in Section 5.4, we give a proof for Theorem 5.1.

5.1 Equivalent Problems

In this section we prove the following theorem.

Theorem 5.2. *Procedure κ-Approx (Algorithm 2) returns an $\Omega_p\left(\kappa^{1/(\min\{2,p\}-1)}\right)$-approximate solution to the residual problem given by (1), by solving $O_p\left(\log\left(\frac{m}{\varepsilon}\right)\right)$ instances of program (2) to a κ-approximation.*

The following lemmas will lead to the proof of the above theorem. The first lemma gives an upper and lower bound on the objective of (1).

Lemma 5.3. *Let $p \in (1, \infty)$ and assume that our current solution $\boldsymbol{x}$ is not an ε-approximate solution. Let λ be such that $\lambda^{\min\{1,p-1\}} = \frac{p-1}{p4^p}$. For some*

$$i \in \left[\log\left(\frac{\varepsilon\|\boldsymbol{x}^{(0)}\|_p^p}{m^{|p-2|/2}}\right), \log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right)\right],$$

$\alpha(\Delta^\star) \in [2^{i-1}, 2^i)$ where $\Delta^\star$ is the solution of (1).

We defer the proof to Appendix A.3. Lemma 5.3 suggests that we can divide the range of the objective of our residual problem, α into a logarithmic number of bins and solve a *decision problem* that asks if the optimum belongs to the bin. The lemma guarantees that at least one of the decision problems will be feasible. The following lemma defines the required *decision problems* and shows that solving these to a constant approximation is sufficient to get a constant approximate solution to (1).

Lemma 5.4. *Let $p \in (1,\infty)$. Suppose $\alpha(\Delta^\star) \in [2^{i-1}, 2^i)$ for some i where $\Delta^\star$ is the solution of (1). The following program is feasible:*

$$\begin{aligned} \gamma_p(\boldsymbol{t}, \Delta) &\le \frac{p}{p-1} 2^{i+p}, \\ \boldsymbol{g}^T \Delta &= 2^{i-1}, \\ \boldsymbol{A}\Delta &= 0. \end{aligned} \tag{3}$$

If $\Delta(i)$ is a β-approximate solution to program (3) for this choice of i, then, we can pick $\mu \le 1$ such that the vector $\mu\Delta(i)$ is an $\Omega_p\left(\beta^{\frac{1}{\min\{p,2\}-1}}\right)$-approximate solution to (1).

The proof can be found in Appendix A.3. We now scale down the objective of (3) so that it is at most 1. The next lemma shows what the scaled down problem looks like and how an approximate solution to the scaled down problem gives an approximate solution to (3). Again the proof of the lemma can be found in Appendix A.3.

Lemma 5.5. *Let $p \in (1,\infty)$. Let i be such that (3) is feasible. Let*

$$\hat{\boldsymbol{t}}_j = \begin{cases} m^{-1/p} & \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1} \boldsymbol{t}_j \le m^{-1/p}, \\ 1 & \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1} \boldsymbol{t}_j \ge 1, \\ \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1} \boldsymbol{t}_j & \text{otherwise.} \end{cases}$$

Note that $m^{-1/p} \le \hat{\boldsymbol{t}}_j \le 1$. Then program (2) with $\boldsymbol{t} = \hat{\boldsymbol{t}}$, and

$$\boldsymbol{c} = \left(\frac{2}{p}\right)^{1/2} \left(\frac{p-1}{p}\right)^{1/p} 2^{i\left(1-\frac{1}{p}\right)-2},$$

has OPT ≤ 1. *Let $\Delta^\star$ be a κ-approximate solution to (2). Then, $\Delta = \left(\frac{p}{2}\right)^{1/2} \left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p} \Delta^\star$ is a $\Omega_p(\kappa)$- approximate solution to (3).*

We now prove Theorem 5.2.

Proof. Lemma 5.3 suggests that there exists an index

$$j \in \left[\log\left(\frac{\varepsilon \|\boldsymbol{x}^{(0)}\|_p^p}{m^{(p-2)/2}}\right), \log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right) \right],$$

such that $OPT = \alpha(\Delta^\star) \in [2^{j-1}, 2^j)$. Lemma 5.4 implies that (3) is feasible for index j. Suppose $\Delta^{(j)}$ is a κ-approximate solution to the scaled down problem (2) for index j. Lemma 5.5 implies that $\tilde{\Delta}^{(j)} = \left(\frac{p}{2}\right)^{1/2} \left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p} \Delta^{(j)}$ is an $\Omega_p(\kappa)$ approximate solution to (3) for index j. Lemma 5.4 now implies that $\tilde{\Delta}^{(j)} = \mu\Delta^{(j)}$ is a $\Omega_p\left(\kappa^{\frac{1}{\min\{p,2\}-1}}\right)$-approximation to our residual problem

Algorithm 2 Approximate Solver

procedure κ-APPROX($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}, \lambda, \boldsymbol{x}^{(0)}$)

$\widehat{\boldsymbol{A}} \leftarrow \left[\boldsymbol{A}^\top, \boldsymbol{g}\right]^\top$

for $i \in \left[\log\left(\frac{\varepsilon\|\boldsymbol{x}^{(0)}\|_p^p}{m^{(p-2)/2}}\right), \log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right)\right]$ **do**

$c(i) \leftarrow \left(\frac{2}{p}\right)^{1/2}\left(\frac{p-1}{p}\right)^{1/p} 2^{i\left(1-\frac{1}{p}\right)-2}$

For every e,

$\boldsymbol{t}_e \leftarrow \max\{m^{-\frac{1}{p}}, (\frac{p-1}{p})^{1/p} 2^{-i/p-1}|\boldsymbol{x}_e^{(t)}|\}$

For every e, $\boldsymbol{t}_e \leftarrow \min\{1, \boldsymbol{t}_e\}$

$\boldsymbol{c} \leftarrow \left[\boldsymbol{0}^\top, c(i)\right]^\top$

$\Delta^\star \leftarrow$ GAMMA-SOLVER($\widehat{\boldsymbol{A}}, \boldsymbol{c}, \boldsymbol{t}$) ▷ κ-approximation to (2)

$\Delta^{(i)} \leftarrow \left(\frac{p}{2}\right)^{1/2}\left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p}\Delta^\star$

$\beta \leftarrow \left(\frac{p}{2}\right)^{p/2}\frac{p2^{p+1}}{p-1}2^i\kappa$

$\mu^{(i)} \leftarrow \begin{cases} \left(\frac{1}{2\beta p}\right)^{1/(p-1)} & \text{if } p \le 2 \\ \frac{1}{4\beta} & \text{otherwise.} \end{cases}$

$\Delta^{(i)} \leftarrow \mu^{(i)}\Delta^{(i)}$

return $\lambda \cdot \arg\min_{\Delta^{(i)}} \|\boldsymbol{x}^{(t)} - \lambda\Delta^{(i)}\|_p^p$

(1). Now, the algorithm solves the scaled down problem for every i and returns the $\tilde{\Delta}^{(i)}$ that when added to our current solution gives the minimum ℓ_p-norm. It either chooses $\tilde{\Delta}^{(j)}$ or some other solution $\tilde{\Delta}^{(i)}$. In case it returns $\tilde{\Delta}^{(i)}$,

$$\begin{aligned}
\|\boldsymbol{x}\|_p^p - \|\boldsymbol{x} - \lambda\tilde{\Delta}^{(i)}\|_p^p &\geq \|\boldsymbol{x}\|_p^p - \|\boldsymbol{x} - \lambda\tilde{\Delta}^{(j)}\|_p^p \\
&\geq \lambda \cdot \alpha(\tilde{\Delta}^{(j)}), \text{Lemma 4.6}, \\
&\geq \lambda \cdot \Omega_p\left(\kappa^{-\frac{1}{\min\{p,2\}-1}}\right)\alpha(\Delta^\star) \\
&= \Omega_p\left(\kappa^{-\frac{1}{\min\{p,2\}-1}}\right)\alpha(\Delta^\star).
\end{aligned}$$

From Lemma 4.6 we know,

$$\alpha(\lambda\tilde{\Delta}^{(i)}) \geq \|\boldsymbol{x}\|_p^p - \|\boldsymbol{x} - \lambda\tilde{\Delta}^{(i)}\|_p^p.$$

We thus have $\alpha(\lambda\tilde{\Delta}^{(i)}) \geq \Omega_p\left(\kappa^{-\frac{1}{\min\{p,2\}-1}}\right)\text{OPT}$, implying $\lambda \cdot \tilde{\Delta}^{(i)}$ is also a $\Omega_p\left(\kappa^{\frac{1}{\min\{p,2\}-1}}\right)$ approximate solution as required. □

It remains to solve problems of the form (2) up to a κ-approximation. Recall that these problems look like,

$$\begin{aligned}
\min_{\Delta} \quad & \gamma_p\left(\boldsymbol{t}, \Delta\right) \\
& \widehat{\boldsymbol{A}}\Delta = \boldsymbol{d},
\end{aligned}$$

and satisfy $OPT \leq 1$, and $m^{-1/p} \leq \boldsymbol{t}_j \leq 1, \forall j$.

5.2 Oracle

Our approach follows the format of the approximate max-flow algorithm by Christiano et al. [Chr+11]. Specifically, we use a variant of multiplicative weights update to converge to a solution with small $\gamma_p(t, \Delta)$. The multiplicative weights update scheme repeatedly updates a set of weights $\boldsymbol{w}$ using partial, local solutions computed based on these weights. The Christiano et al. algorithm can be viewed as picking these weights from the gradients of the soft-max function on flows. We will adapt this routine by showing that $\boldsymbol{w}$'s chosen from the gradient of $\gamma_p(t, \Delta)$ also suffices for approximately minimizing the problem stated in 2.

The subroutine that this algorithm passes the $\boldsymbol{w}$ onto is commonly referred to as an oracle. An oracle needs to compute a solution with both small dot-product against $\boldsymbol{w}$, and small *width*, which is defined as the maximum value of an entry. In such an oracle, the dot product condition is the hard constraint, in that the final approximation factor of the solution is directly related to the value of these dot products. The width, on the other hand, only affects the overall iteration count/ running time, and can even be manipulated/improved algorithmically. Therefore we first need to define and show a good upper bound on the objective of the optimization problem solved within the oracle.

Formally, our oracle subroutine Algorithm 3 takes as input some affine constraints and vector of weights $\boldsymbol{w}$. It first computes a vector of non-negative weights $\boldsymbol{r}$, and then returns a minimizer

to the following optimization problem

$$\begin{aligned} \Delta = \arg\min_{\Delta \in \mathbb{R}^m} \quad & \sum_e \boldsymbol{r}_e \Delta_e^2 \\ \text{s.t.} \quad & \widehat{\boldsymbol{A}} \Delta = \boldsymbol{d}. \end{aligned} \tag{4}$$

Appendix C contains an algorithm that solves such problems efficiently.

Algorithm 3 Oracle

1: **procedure** ORACLE($\widehat{\boldsymbol{A}}, \boldsymbol{d}, \boldsymbol{w}, \boldsymbol{t}$)
2: $\quad \boldsymbol{r}_e \leftarrow \left(m^{1/p} \boldsymbol{t}_e\right)^{p-2} + \boldsymbol{w}_e^{p-2}$
3: $\quad$ Compute the Δ using resistances $\boldsymbol{r}_e$ that satisfies solves the following optimization problem

$$\begin{aligned} \Delta = \arg\min_{\Delta'} \quad & \sum_e \boldsymbol{r}_e {\Delta'}_e^2 \\ \text{s.t.} \quad & \widehat{\boldsymbol{A}} \Delta' = \boldsymbol{d} \end{aligned}$$

4: $\quad$ **return** Δ

Let us now look at some properties of the solution returned by the oracle. Note that the objective of our problem (2) is at most 1. This implies that we have Δ such that

- $\sum_e (\Delta_e^*)^2 \boldsymbol{t}_e^{p-2} \leq 1$,
- $\sum_e |\Delta_e^*|^p \leq 1$, or $\|\Delta^*\|_p \leq 1$.

We next look at some relations on the weights and resistances. The following lemma is a simple application of Hölder's inequality. Its proof is given in Appendix A.3.

Lemma 5.6. *Let $p \geq 2$. For any set of weights $\boldsymbol{w}$ on the edges, $\sum_e \boldsymbol{w}_e^{p-2} (\Delta_e^*)^2 \leq \|\boldsymbol{w}\|_p^{p-2}$.*

Lemma 5.7. *Let $p \geq 2$. For any $\boldsymbol{w}$, let Δ be the electrical flow computed with respect to resistances*

$$\boldsymbol{r}_e \stackrel{\text{def}}{=} \left(m^{1/p} \boldsymbol{t}_e\right)^{p-2} + \boldsymbol{w}_e^{p-2},$$

and demand vector $\boldsymbol{d}$.

Then the following hold,

1. $\sum_e \Delta_e^2 \leq \sum_e \boldsymbol{r}_e \Delta_e^2 \leq m^{\frac{p-2}{p}} + \|\boldsymbol{w}\|_p^{p-2}$,
2. $\sum_e |\Delta_e| \left|\gamma'(m^{1/p} \boldsymbol{t}_e, \boldsymbol{w}_e)\right| \leq p \left(\sum_e \gamma_p(m^{1/p} \boldsymbol{t}_e, \boldsymbol{w}_e)\right)^{\frac{p-1}{p}} + p m^{\frac{p-2}{2p}} \left(\sum_e \gamma_p(m^{1/p} \boldsymbol{t}_e, \boldsymbol{w}_e)\right)^{\frac{1}{2}}$.

Proof. Since Δ is the electrical flow,

$$\sum_e \boldsymbol{r}_e \Delta_e^2 \leq \sum_e \boldsymbol{r}_e (\Delta_e^*)^2.$$

We have,

$$\sum_e \boldsymbol{r}_e \Delta_e^2 \le \sum_e \boldsymbol{r}_e (\Delta_e^*)^2 = \sum_e (m^{1/p}\boldsymbol{t}_e)^{p-2}(\Delta_e^*)^2 + \sum_e \boldsymbol{w}_e^{p-2}(\Delta_e^*)^2$$
$$\le m^{\frac{p-2}{p}} + \|\boldsymbol{w}\|_p^{p-2}, \text{follows from lemma 5.6 and the fact } \sum_e (\Delta_e^*)^2 \boldsymbol{t}_e^{p-2} \le 1.$$

Finally, using $\boldsymbol{r}_e \ge \left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} \ge 1$, we have $\sum_e \Delta_e^2 \le \sum_e \boldsymbol{r}_e \Delta_e^2$, completing part 1.

Now we know that,

$$\left|\gamma'(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)\right| = \begin{cases} p(m^{1/p}\boldsymbol{t}_e)^{p-2}\boldsymbol{w}_e & \text{if } |\boldsymbol{w}_e| \le m^{1/p}\boldsymbol{t}_e, \\ p|\boldsymbol{w}_e|^{p-2}\boldsymbol{w}_e & \text{otherwise .} \end{cases}$$

Using Cauchy Schwarz's inequality,

$$\left(\sum_e |\Delta_e| \left|\gamma'(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)\right|\right)^2 = \left(\sum_e p\,|\Delta_e| \left|\max(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)\right|^{p-2} \boldsymbol{w}_e^2\right)^2$$
$$\le p^2 \left(\sum_e \max(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)^{p-2}\boldsymbol{w}_e^2\right)\left(\sum_e \max(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)^{p-2}\Delta_e^2\right)$$
$$\le p^2 \gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w}) \sum_e \max(m^{1/p}\boldsymbol{t}_e, |\boldsymbol{w}_e|)^{p-2}\Delta_e^2$$

Combining the two cases we have,

$$\sum_e |\Delta_e| \left|\gamma'(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e)\right| \le p\sqrt{\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w}) \sum_e \max(m^{1/p}\boldsymbol{t}_e, |\boldsymbol{w}_e|)^{p-2}\Delta_e^2}$$
$$\le p\sqrt{\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w}) \sum_e \boldsymbol{r}_e \Delta_e^2}$$
$$\le p\sqrt{\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})}\sqrt{m^{\frac{p-2}{p}} + \|\boldsymbol{w}\|_p^{p-2}}$$
$$\le pm^{\frac{p-2}{2p}}\sqrt{\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})} + p\sqrt{\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})}\,\|\boldsymbol{w}\|_p^{\frac{p-2}{2}}$$
$$= pm^{\frac{p-2}{2p}}\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})^{\frac{1}{2}} + p\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})^{\frac{p-1}{p}},$$

where the last line uses $\|x\|_p^p \le \gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})$ for any $\boldsymbol{t}$. □

5.3 The Algorithm

Next, we integrate this oracle into the overall algorithm that repeatedly adjusts the weights. As with the use of electrical flow oracles for approximate max-flow [Chr+11], the convergence of such a scheme depends on the maximum values in the Δ returned by the oracle. However, because the overall objective is now a p-norm, the exact term of importance is actually the p-norm of Δ. Up to this discrepancy, we follow the algorithmic template from [Chr+11] by making an update when $\|\Delta\|_p^p$ is small, and make progress via another potential function otherwise.

In the cases where we do not take the step due to entries with large values, we show significant increases in an additional potential function, namely the objective of the quadratic minimization problem inside the oracle (Algorithm 3). However, the less graduate update schemes related to p-norms makes it no longer sufficient to update only the weight corresponding to the entry with maximum value. Furthermore, there may be entries with large values, whose corresponding resistances are too large for us to afford increasing. We address this by a scheme where we update an entry only if its value is larger than some threshold ρ, and that its resistance is at most another threshold β. Specifically, we show that for an appropriate choice of β and ρ, such updates both do not change the primary potential function (related to $\gamma_p(t, \boldsymbol{x})$) by too much (in Lemma 5.10), and increases the secondary potential function (the objective of the quadratic minimization problem) significantly whenever $\|\Delta\|_p^p$ is large (in Lemma 5.13). Pseudocode of this scheme is in Algorithm 4.

Algorithm 4 Algorithm for the Scaled down Problem

procedure GAMMA-SOLVER($\boldsymbol{A}', \boldsymbol{c}, \boldsymbol{t}$)
  $\boldsymbol{w}_e^{(0,0)} \leftarrow 0$
  $\boldsymbol{x} \leftarrow 0$
  $\rho \leftarrow \tilde{\Theta}_p\left(m^{\frac{(p^2-4p+2)}{p(3p-2)}}\right)$ ▷ width parameter
  $\beta \leftarrow \tilde{\Theta}_p\left(m^{\frac{p-2}{3p-2}}\right)$ ▷ resistance threshold
  $\alpha \leftarrow \tilde{\Theta}_p\left(m^{-\frac{p^2-5p+2}{p(3p-2)}}\right)$ ▷ step size
  $\tau \leftarrow \tilde{\Theta}_p\left(m^{\frac{(p-1)(p-2)}{(3p-2)}}\right)$ ▷ ℓ_p energy threshold
  $T \leftarrow \alpha^{-1}m^{1/p} = \tilde{\Theta}_p\left(m^{\frac{p-2}{3p-2}}\right)$
  $i \leftarrow 0, k \leftarrow 0$
  while $i < T$ **do**
    $\Delta = \text{ORACLE}(\boldsymbol{A}', \boldsymbol{c}, \boldsymbol{w}^{(i,k)}, \boldsymbol{t})$
    if $\|\Delta\|_p^p \leq \tau$ **then** ▷ flow step
      $\boldsymbol{w}^{(i+1,k)} \leftarrow \boldsymbol{w}^{(i,k)} + \alpha\,|\Delta|$
      $\boldsymbol{x} \leftarrow \boldsymbol{x} + \alpha\Delta$
      $i \leftarrow i + 1$
    else ▷ width reduction step
      For all edges e with $|\Delta_e| \geq \rho$ and $\boldsymbol{r}_e \leq \beta$
        $\boldsymbol{w}_e^{(i,k+1)} \leftarrow 4^{\frac{1}{p-2}} \max(m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e^{(i,k)})$
      $k \leftarrow k + 1$
  return $m^{-\frac{1}{p}}\boldsymbol{x}$

Theorem 5.8. *Let $p \geq 2$. Given a matrix $\widehat{\boldsymbol{A}}$ and vectors $\boldsymbol{x}$ and $\boldsymbol{t}$ such that $\forall e, m^{-1/p} \leq \boldsymbol{t}_e \leq 1$, Algorithm 4 uses $O_p\left(m^{\frac{p-2}{(3p-2)}}\left(\log\left(\frac{m\|\boldsymbol{d}\|_2^2}{\|\widehat{\boldsymbol{A}}\|^2}\right)\right)^{\frac{p}{3p-2}}\right)$ calls to the oracle and returns a vector $\boldsymbol{x}$ such*

that $\widehat{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{d}$*, and* $\gamma_p(\boldsymbol{t}, \boldsymbol{x}) = O_p(1)$*.*

Analysis of Potentials.

We define the following potential function for the analysis of our algorithm.

Definition 5.9. *Let* Φ *be the potential function defined as*

$$\Phi\left(\boldsymbol{w}^{(i)}\right) \stackrel{\text{def}}{=} \gamma_p\left(m^{1/p}\boldsymbol{t}, \boldsymbol{w}^{(i)}\right).$$

Initially, since we start with $\boldsymbol{w}^{(0)} = 0$, we have $\Phi(\boldsymbol{w}^{(0)}) = 0$. Observe that in the algorithm, we update the potentials in both the flow step and the width reduction step whereas we update the solution only in the flow step. It is easy to see that we always have $\boldsymbol{w}^{(i,k)} \geq \left|\boldsymbol{x}^{(i,k)}\right|$.

We next bound the potential. In addition, we track the energy of the electrical flow in the network with resistances $\boldsymbol{r}$. Let $\Psi(\boldsymbol{r})$ denote the minimum of routing $\boldsymbol{d}$ with resistances $\boldsymbol{r}$:

$$\Psi(\boldsymbol{r}) \stackrel{\text{def}}{=} \min_{\Delta:\boldsymbol{A}'\Delta=\boldsymbol{d}} \sum_e \boldsymbol{r}_e\Delta_e^2. \tag{5}$$

Note that this energy is equal to the energy calculated using the Δ obtained in the solution of (4).

Notation. We overload notation for $\Psi(i,k)$ to denote $\Psi\left(\boldsymbol{r}^{(i,k)}\right)$.

Our proof of Theorem 5.8 will be based two main parts:

1. Provided the total number of width reduction steps, K, is not too big, then $\Phi(T,K)$ is small. This in turn upper bounds cost of the approximate solution $m^{-1/p}\boldsymbol{x}$.
2. Showing that K cannot be too big, because each width reduction step cause large growth in $\Psi(\cdot)$, while we can bound the total growth in $\Psi(\cdot)$ by relating it to $\Phi(\cdot)$.

We start by observing that when we when increase the weight $\boldsymbol{w}_e$ of an edge during a width reduction step, this has the effect of at least doubling the resistance $\boldsymbol{r}_e$. Recall,

$$\boldsymbol{r}_e^{(i,k)} \stackrel{\text{def}}{=} \left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i,k)}\right)^{p-2}.$$

Now,

$$\frac{\boldsymbol{r}_e^{(i,k+1)}}{\boldsymbol{r}_e^{(i,k)}} = \frac{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i,k+1)}\right)^{p-2}}{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i,k)}\right)^{p-2}} = \frac{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + 4\max\left\{m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e^{(i,k)}\right\}^{p-2}}{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i,k)}\right)^{p-2}} \geq 2. \tag{6}$$

Meanwhile, the resistance does not grow by a factor larger than 4:

$$\frac{\boldsymbol{r}_e^{(i,k+1)}}{\boldsymbol{r}_e^{(i,k)}} = \frac{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + 4\max\left\{m^{1/p}\boldsymbol{t}_e, \boldsymbol{w}_e^{(i,k)}\right\}^{p-2}}{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i,k)}\right)^{p-2}} \leq 4. \tag{7}$$

We next show through the following lemma that the Φ potential does not increase too rapidly. The proof is through induction and can be found in Appendix B .

Lemma 5.10. *After i* flow *steps, and k width-reduction steps, provided*

1. $\alpha^p \tau \leq \alpha m^{\frac{p-1}{p}}$, *(controls Φ growth in flow-steps)*
2. $k \leq \rho^2 m^{2/p} \beta^{-\frac{2}{p-2}}$, *(acceptable number of width-reduction steps)*

the potential Φ is bounded as follows:

$$\Phi(i,k) \leq \left(p^2 2^p \alpha i + m^{1/p}\right)^p \exp\left(O_p(1)\frac{k}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right).$$

We next wish to prove that in each width-reduction step, the electrical energy $\Psi(\cdot)$ goes up significantly. For this, we will use the following Lemma which is proven in Appendix D. It generalizes Lemma 2.6 of [Chr+11] to arbitrary weighted ℓ_2 regression problems, and directly measures the change in terms of the electrical energy of the entries modified.

Lemma 5.11. *Assuming the program* (4) *is feasible, let Δ be an be a solution to the optimization problem* (4) *with weights $\boldsymbol{r}$. Suppose we increase the resistance on each entry to get $\boldsymbol{r}'$ Then,*

$$\Psi\left(\boldsymbol{r}'\right) \geq \exp\left(\frac{\sum_e \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\} \boldsymbol{r}_e \Delta_e^2}{2\Psi\left(\boldsymbol{r}\right)}\right) \Psi\left(\boldsymbol{r}\right).$$

This statement also implies the form of the lemma that concerns increasing the resistances on a set of entries uniformly [Chr+11, Lemma 2.6].

The next lemma gives a lower bound on the energy in iteration 0, i.e., when we start, and an upper bound on the energy at each step.

Lemma 5.12. *Initially, we have,*

$$\Psi\left(\boldsymbol{r}^{(0,0)}\right) \geq \frac{\|\boldsymbol{d}\|_2^2}{\|\boldsymbol{A}\|^2},$$

where $\|\boldsymbol{A}\|$ is the operator norm, or maximum singular value of $\boldsymbol{A}$. Let us call this ratio L. Moreover, at any step i, we have,

$$\Psi\left(\boldsymbol{r}^{(i,k)}\right) \leq m^{\frac{p-2}{p}} + \Phi(i,k)^{\frac{p-2}{p}}.$$

Proof. For the lower bound in the initial state, recall that we scale the problem such that $\mathrm{OPT} = 1$, and $\boldsymbol{t}_e \geq m^{1/p}$. Initially we have, $\boldsymbol{r}_e^{(0,0)} = (m^{1/p}\boldsymbol{t}_e)^{p-2} \geq 1$. This means for any solution Δ, we have

$$\sum_e \boldsymbol{r}^{(0,0)} \Delta_e^2 \geq \|\Delta\|_2^2.$$

On the other hand, because

$$\|\boldsymbol{A}\Delta\|_2 \leq \|\boldsymbol{A}\|_2 \|\Delta\|_2,$$

we get

$$\|\Delta\|_2 \geq \frac{\|\boldsymbol{d}\|_2}{\|\boldsymbol{A}\|_2},$$

upon which squaring gives the lower bound on $\Psi(\boldsymbol{r}^{(0,0)})$.
For the upper bound, Lemma 5.7 implies that

$$\Psi\left(\boldsymbol{r}^{(i,k)}\right) \leq m^{\frac{p-2}{p}} + \|\boldsymbol{w}\|_p^{p-2} \leq m^{\frac{p-2}{p}} + \Phi(i,k)^{\frac{p-2}{p}}.$$

□

The next Lemma says that the three assumptions (stated in the statement of the Lemma) can be used to ensure that the potential $\Psi(\cdot)$ grows quickly with each width reduction step, and that flow steps do not cause the potential to shrink.

Lemma 5.13. *Suppose at step (i,k) we have $\|\Delta\|_p > \tau$ so that we perform a width reduction step (line 18). If*

1. $\Phi(i,k) \leq O_p(1)m$,
2. $\tau^{2/p} \geq 2\Omega_p(1)\frac{m^{\frac{p-2}{p}}}{\beta}$, *and*
3. $\frac{\tau}{10} \geq \rho^{p-2}m^{\frac{p-2}{p}}$.

Then

$$\Psi(i,k+1) \geq \Psi(i,k)\left(1 + \Omega_p(1)\frac{\tau^{2/p}}{m^{\frac{p-2}{p}}}\right).$$

Furthermore, if at (i,k) we have $\|\Delta\|_p \leq \tau$ so that we perform a flow step, then

$$\Psi(i+1,k) \geq \Psi(i,k).$$

Proof. It will be helpful for our analysis to split the index set into three disjoint parts:

- $S = \{e : |\Delta_e| \leq \rho\}$
- $H = \{e : |\Delta_e| > \rho \text{ and } \boldsymbol{r}_e \leq \beta\}$
- $B = \{e : |\Delta_e| > \rho \text{ and } \boldsymbol{r}_e > \beta\}$.

Firstly, we note

$$\sum_{e\in S}|\Delta_e|^p \leq \rho^{p-2}\sum_{e\in S}|\Delta_e|^2 \leq \rho^{p-2}\sum_{e\in S}\boldsymbol{r}_e|\Delta_e|^p \leq \rho^{p-2}O_p(1)m^{(p-2)/p}.$$

hence, using Assumption 3

$$\sum_{e\in H\cup B}|\Delta_e|^p \geq \sum_e \Delta_e^p - \sum_{e\in S}|\Delta_e|^p \geq \tau - \rho^{p-2}m^{\frac{p-2}{p}} \geq 9\tau.$$

This means,

$$\sum_{e\in H\cup B}\Delta_e^2 \geq \left(\sum_{e\in H\cup B}\Delta_e^p\right)^{p/2} \geq \Omega_p(1)\tau^{2/p}.$$

Secondly we note that, using Assumption (1) and Lemma 5.7, we have

$$\sum_{e\in B}\Delta_e^2 \le \beta^{-1}\sum_{e\in B}\boldsymbol{r}_e\Delta_e^2 \le \beta^{-1}O_p(1)m^{\frac{p-2}{p}}.$$

So then, using Assumption 2,

$$\sum_{e\in H}\Delta_e^2 = \sum_{e\in H\cup B}\Delta_e^2 - \sum_{e\in B}\Delta_e^2 \ge \Omega_p(1)\tau^{2/p} - \beta^{-1}O_p(1)m^{\frac{p-2}{p}} \ge \Omega_p(1)\tau^{2/p}.$$

As $\boldsymbol{r}_e \ge 1$, this implies $\sum_{e\in H}\boldsymbol{r}_e\Delta_e^2 \ge \Omega_p(1)\tau^{2/p}$.

From Lemma 5.12 and Assumption 1 we have

$$\Psi(i,k) \le O_p(1)m^{(p-2)/p}.$$

So then, combining our last two observations, and applying Lemma 5.11, we get

$$\Psi(i,k+1) \ge \Psi(i,k)\left(1+\Omega_p(1)\frac{\tau^{2/p}}{m^{\frac{p-2}{p}}}\right).$$

Finally, for the "flow step" case, by applying Lemma 5.11 with H as the whole set of indices, $\delta = 1$ and $\gamma = 1$, we get that as the resistances only increase,

$$\Psi(i+1,k) \ge \Psi(i,k).$$

□

We are now ready to prove Theorem 5.8.

Proof of Theorem 5.8

Proof. We first observe that our parameter choices in the Algorithm 4 satisfy Assumption 1 of Lemma 5.10, namely, we can choose the parameters α and τ s.t.

- $\alpha \leftarrow \Theta_p\left(m^{-\frac{p^2-5p+2}{p(3p-2)}}\left(\log\left(\frac{m}{L}\right)\right)^{\frac{-p}{3p-2}}\right)$,
- $\tau \leftarrow \Theta_p\left(m^{\frac{(p-1)(p-2)}{(3p-2)}}\left(\log\left(\frac{m}{L}\right)\right)^{\frac{p(p-1)}{3p-2}}\right)$,

while ensuring $\alpha^p\tau \le \alpha m^{\frac{p-2}{p}}$. This means by Lemma 5.10, that if the Algorithm completes after taking $T = \alpha^{-1}m^{1/p}$ flow steps and $K \le \Omega_p(1)\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}$, when it returns, we have

$$\Phi(T,K) \le m\left(p^2 2^p + 1\right)^p e^1 \le O_p(1)m,$$

This means that the algorithm returns $m^{-\frac{1}{p}}\boldsymbol{x}$ with

$$\gamma(\boldsymbol{t}, m^{-\frac{1}{p}}\boldsymbol{x}) = \frac{1}{m}\gamma(m^{\frac{1}{p}}\boldsymbol{t},\boldsymbol{x}) \le \frac{1}{m}\gamma(m^{\frac{1}{p}}\boldsymbol{t},\boldsymbol{w}^{(T,K)}) = \frac{1}{m}\Phi(T,K) \le O_p(1).$$

Note the only alternative is that the algorithm takes more than $\Omega_p(1)\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}$ width reduction steps (and possibly infinitely many such steps, hence never terminating).

We will now show this cannot happen, by deriving a contradiction from the assumption that the algorithm takes a width reduction step starting from step (i,k) where $i < T$ and $k = \rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}$.

Since the conditions for Lemma 5.10 hold for all preceding steps, we must have $\Phi(i,k) \leq O_p(1)m$.

Additionally, we note that our parameter choice of $\beta = \Theta_p\left(m^{\frac{p-2}{3p-2}}\left(\log\left(\frac{m}{L}\right)\right)^{-\frac{2(p-1)}{3p-2}}\right)$ and $\rho = \Theta_p\left(m^{\frac{(p^2-4p+2)}{p(3p-2)}}\left(\log\left(\frac{m}{L}\right)\right)^{\frac{p(p-1)}{(p-2)(3p-2)}}\right)$ along with our choice of τ (see above), ensures that

$$\tau^{2/p} \geq 2\Omega_p(1)\frac{m^{\frac{p-2}{p}}}{\beta} \text{ and } \frac{\tau}{10} \geq \rho^{p-2}m^{\frac{p-2}{p}}.$$

This means that at every step (j,l) preceding the current step, the conditions of Lemma 5.13 are satisfied, so we can prove by a simple induction that

$$\Psi(i,k) \geq \Psi(0,0)\left(1+\Omega_p(1)\frac{\tau^{2/p}}{m^{\frac{p-2}{p}}}\right)^k > \Psi(0,0)\exp\left(\Omega_p(1)\frac{\tau^{2/p}}{m^{\frac{p-2}{p}}}k\right).$$

Since our parameter choices ensure $\Omega_p(1)\frac{\tau^{2/p}}{m^{\frac{p-2}{p}}}k > \Theta_p\left(\frac{m}{L}\right)$ this means

$$\Psi(i,k) > \Psi(0,0)\cdot\Theta_p\left(\frac{m}{L}\right).$$

But this contradicts Lemma 5.12, since this Lemma, combined with $\Phi(i,k) \leq O_p(1)m$ gives

$$\frac{\Psi(i,k)}{\Psi(0,0)} \leq O_p\left(m^{\frac{p-2}{p}}\right).$$

From this contradiction, we conclude that we never have more than $K = \Omega_p(1)\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}$ width reduction steps.

Now we observe that the total number of oracle calls in the algorithm is bounded by

$$T + K \leq \Theta_p\left(m^{\frac{p-2}{3p-2}}\left(\log\left(\frac{m}{L}\right)\right)^{\frac{p}{3p-2}}\right).$$

□

This concludes the analysis of our algorithm.

5.4 Proof of Theorem 5.1

Proof. Theorem 5.8 implies that we can solve Program (2) using Algorithm 4 to get an $O_p(1)$-approximate solution in $\widetilde{O}_p\left(m^{\frac{p-2}{3p-2}}\right)$ calls to the Oracle. Implementing the Oracle requires solving

a linear system, and hence can be implemented in in $O(m+n)^\omega$ time where ω is the matrix multiplication constant (see the Appendix for a proof). Thus, we can find an $O_p(1)$-approximate solution to (2) in total time

$$\widetilde{O}_p\left((m+n)^{\omega+\frac{p-2}{3p-2}}\right).$$

Now, Theorem 5.2 implies that we can find an $O_p(1)$-approximate solution to the residual problem (1) in total time,

$$\widetilde{O}_p\left((m+n)^{\omega+\frac{p-2}{3p-2}}\log 1/\varepsilon\right).$$

Finally using Theorem 4.1 we can conclude that we have an ε-approximate solution to (*) in $\widetilde{O}_p\left(\log\frac{1}{\varepsilon}\right)$ calls to a $O_p(1)$-approximate solver to the residual problem (1). This gives us a total running time of,

$$\widetilde{O}_p\left((m+n)^{\omega+\frac{p-2}{3p-2}}\log^2 1/\varepsilon\right).$$

□

We now have a complete algorithm for the p-norm regression problem that gives an ε-approximate solution.

6 Speedups for General Matrices via. Inverse Maintenance

If $\boldsymbol{A}$ is an explicitly given, $m \times n$, matrix, we need to solve the quadratic minimization problem at each step. This can be solved via a linear systems solve in the matrix

$$\boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1}\boldsymbol{A}.$$

which takes $O((m+n)^\omega)$, where ω is the matrix multiplication constant. This directly gives a total running time cost of $\widetilde{O}_p(m^{\frac{p-2}{(3p-2)}}(m+n)^\omega\log(1/\epsilon))$, which for large values of p, along with the assumption of $\omega > 2.37$, exceeds 2.70.

This is more than the running time of about $O(mn^{1.5})$ of algorithms based on inverse maintenance [LS14; LS15; Vai90]. In this section we show that the MWU routine from Section 5 can also benefit from fast inverse maintenance. Our main result is:

Theorem 6.1. *If $\boldsymbol{A}$ is an explicitly given, m-by-n matrix with polynomially bounded condition numbers, and $p \geq 2$ Algorithm 4 as given in Section 5.3 can be implemented to run in total time*

$$\widetilde{O}_p\left((m+n)^{\max\left\{\omega,2+\frac{p-(10-4\omega)}{3p-2}\right\}}\right).$$

A few remarks about this running time: the term that dominates depends on the comparison between $2/3$ and $10-4\omega$, or after manipulation, the comparison between ω and $7/3$:

1. For the current best value of $\omega > 7/3$, the second term is at most ω, so the total running time is about $(m+n)^\omega$.

2. If $\omega = 2$, then this running time is simply $(m+n)^{\frac{p-2}{3p-2}}$: same as resolving the linear system at each step.

3. If $\omega \leq 7/3$, then the overhead in the exponent on the second term is at most

$$\frac{p - 2/3}{3p - 2} = 1/3,$$

and this value approaches $\frac{p-2}{3p-2}$ as $\omega \to 2$.

Our algorithm is based on gradually updating the $\boldsymbol{r}$ vector. First, note that $\boldsymbol{w}_e^{(i)}$'s, and thus $\boldsymbol{r}_e^{(i)}$'s are monotonically increasing. Secondly, for the $\boldsymbol{r}^{(i)}$ that do not double, we can replace with the original version while forming a factor 2 preconditioner. Thus, we only need to update the $\boldsymbol{r}^{(i)}$ entries that have significant increases. This update can be encapsulated by the following result on computing low rank perturbations to a matrix, which is a direct consequence of rectangular matrix multiplication and Woodbury matrix formula.

Lemma 6.2. *Given an m-by-n matrix $\boldsymbol{A}$, along with vectors $\widehat{\boldsymbol{r}}$ and $\widetilde{\boldsymbol{r}}$ that differ in k entries, as well as the matrix $\widehat{\boldsymbol{Z}} = (\boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\,\boldsymbol{A})^{-1}$, we can construct $(\boldsymbol{A}^\top \mathbf{Diag}\,(\widetilde{\boldsymbol{r}})^{-1}\,\boldsymbol{A})^{-1}$ in $O(k^{\omega-2}(m+n)^2)$ time.*

Proof. Let S denote the entries that differ in $\widehat{\boldsymbol{r}}$ and $\widetilde{\boldsymbol{r}}$. Then we have

$$\boldsymbol{A}^\top \mathbf{Diag}\,(\widetilde{\boldsymbol{r}})^{-1}\,\boldsymbol{A} = \boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\,\boldsymbol{A} + \boldsymbol{A}_{:,S}^\top \left(\mathbf{Diag}\,(\widetilde{\boldsymbol{r}}_S)^{-1} - \mathbf{Diag}\,(\widehat{\boldsymbol{r}}_S)^{-1}\right) \boldsymbol{A}_{S,:}.$$

This is a low rank perturbation, so by Woodbury matrix identity we get:

$$\left(\boldsymbol{A}^\top \mathbf{Diag}\,(\widetilde{\boldsymbol{r}})^{-1}\,\boldsymbol{A}\right)^{-1} = \widehat{\boldsymbol{Z}} - \widehat{\boldsymbol{Z}}\boldsymbol{A}_{:,S}^\top \left(\left(\mathbf{Diag}\,(\widetilde{\boldsymbol{r}}_S)^{-1} - \mathbf{Diag}\,(\widehat{\boldsymbol{r}}_S)^{-1}\right)^{-1} + \boldsymbol{A}_{S,:}\widehat{\boldsymbol{Z}}\boldsymbol{A}_{:,S}^\top\right)^{-1} \boldsymbol{A}_{S,:}\widehat{\boldsymbol{Z}},$$

where we use $\widehat{\boldsymbol{Z}}^\top = \widehat{\boldsymbol{Z}}$ because $\boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\,\boldsymbol{A}$ is a symmetric matrix. To explicitly compute this matrix, we need to:

1. compute the matrix $\boldsymbol{A}_{S,:}\widehat{\boldsymbol{Z}}$,
2. compute $\boldsymbol{A}_{:,S}\widehat{\boldsymbol{Z}}\boldsymbol{A}_{:,S}^\top$
3. invert the middle term.

This cost is dominated by the first term, which can be viewed as multiplying $\lceil n/k \rceil$ pairs of $k \times n$ and $n \times k$ matrices. Each such multiplication takes time $k^{\omega-1}n$, for a total cost of $O(k^{\omega-2}n^2)$. The other terms all involve matrices with dimension at most $k \times n$, and are thus lower order terms. □

Note that the running time of Lemma 6.2 favours 'batching' a large number of modified edges to insert. To this end, we show that it suffices to have an inverse that only approximates some entries of $\boldsymbol{r}^{(i)}$. To do so, we first need to introduce our notions of approximations:

Definition 6.3. *We use $a \approx_c b$ for positive numbers a and b iff $c^{-1}a \leq b \leq c \cdot b$, and for vectors and for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ we use $\boldsymbol{a} \approx_c \boldsymbol{b}$ to denote $\boldsymbol{a}_i \approx_c \boldsymbol{b}_i$ entry-wise.*

Since we are only updating k resistances that have a constant factor increase and using a constant factor preconditioning for the others, we need the following result on preconditioned iterative methods for solving systems of linear equations.

Lemma 6.4. *If $\boldsymbol{r}$ and $\widehat{\boldsymbol{r}}$ are vectors such that $\boldsymbol{r} \approx_{\widetilde{O}(1)} \widehat{\boldsymbol{r}}$, and we're given the matrix $\widehat{\boldsymbol{Z}} = (\boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\boldsymbol{A})^{-1}$ explicitly, then we can solve a system of linear equations involving $\boldsymbol{A}^\top \mathbf{Diag}\,(\boldsymbol{r})^{-1}\boldsymbol{A}$ to $1/poly(n)$ accuracy in $\widetilde{O}(n^2)$ time.*

As the resistances we provide to ORACLE are in the range $[1, O_p(m)]$, we get that each $\widehat{\boldsymbol{r}}_e$ only needs to be updated $O(\log m)$ times, instead of after each iteration. However, it's insufficient to use this bound in the worst-case manner: if there are $m^{1/3}$ iterations, each of which doubles the resistances on $m^{2/3}$ edges, then the total cost as given by Lemma 6.2 becomes

$$(m+n)^{2+\frac{1}{3}+\frac{2}{3}\cdot(\omega-2)},$$

which is about $(m+n)^{2.58}$.

We get an even better bound by using our analysis from Section 5 to show that for iteration/edge combinations i and e, the (relative) update to $\boldsymbol{r}_e^{(i)}$ is small. Such small changes also imply that we can wait on such updates. For simplicity, suppose we only increment the resistances by factors of $\frac{1}{L}$, then it takes $\Theta(L)$ such increments until the edge's resistance has deviated by a constant factor. Furthermore, we can wait for another $\Theta(L)$ iterations before having to reflect this change in $\widehat{\boldsymbol{r}}$: the total relative increases in these iterations is also at most $O(1)$. Formalizing this process leads to a lazy-update routine that tracks the increments of different sizes separately. Its Pseudocode is in Algorithms 5 and 6.

We will call the initialization routine INVERSEINIT at the first iteration, and subsequently call UPDATEINVERSE upon generating a new set of resistances in the call to Algorithm 3, ORACLE. This is in turn called from Line 11 of Algorithm 4. As a result, we will assume access to all variables of these routines. Furthermore, our routines keeps the following global variables:

1. $\widehat{\boldsymbol{r}}$: resistances from the last time we updated each entry.
2. $counter(\eta)_e$: for each entry, track the number of times that it changed (relative to $\widehat{\boldsymbol{r}}$) by a factor of about $2^{-\eta}$ since the previous update.
3. $\widehat{\boldsymbol{Z}}$, an inverse of the matrix given by $\boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\boldsymbol{A}$.

Algorithm 5 Inverse Maintenance Initialization

1: **procedure** INVERSEINIT
2: Set $\widehat{\boldsymbol{r}} \leftarrow \boldsymbol{r}^{(0)}$.
3: Set $counter(\eta)_e \leftarrow 0$ for all $0 \le \eta \le \log(m)$ and e.
4: Set $\boldsymbol{Z} \leftarrow (\boldsymbol{A}^\top \mathbf{Diag}\,(\widehat{\boldsymbol{r}})^{-1}\boldsymbol{A})^{-1}$ by explicitly inverting the matrix.

Algorithm 6 Inverse Maintenance Procedure

procedure UPDATEINVERSE
  for all entries e **do**
    Find the least non-negative integer η such that

$$\frac{1}{2^\eta} \leq \frac{\boldsymbol{r}_e^{(i)} - \boldsymbol{r}_e^{(i-1)}}{\widehat{\boldsymbol{r}}_e}.$$

    Increment $counter(\eta)_e$.
  $E_{changed} \leftarrow \cup_{\eta : i \pmod{2^\eta} \equiv 0} \{e : counter(\eta)_e \geq 2^\eta\}$
  $\widetilde{\boldsymbol{r}} \leftarrow \widehat{\boldsymbol{r}}$
  for all $e \in E_{changed}$ **do**
    $\widetilde{\boldsymbol{r}}_e \leftarrow \boldsymbol{r}_e^{(i)}$.
    Set $counter(\eta)_e \leftarrow 0$ for all η.
  $\widehat{\boldsymbol{Z}} \leftarrow$ LOWRANKUPDATE$(\boldsymbol{A}, \boldsymbol{Z}, \widehat{\boldsymbol{r}}, \widetilde{\boldsymbol{r}})$.
  $\widehat{\boldsymbol{r}} \leftarrow \widetilde{\boldsymbol{r}}$.

We first verify that the maintained inverse is always a good preconditioner to the actual matrix, $\boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}^{(i)}\right) \boldsymbol{A}$.

Lemma 6.5. *After each call to* UPDATEINVERSE*, the vector $\widehat{\boldsymbol{r}}$ satisfies*

$$\widehat{\boldsymbol{r}} \approx_{\widetilde{O}(1)} \boldsymbol{r}^{(i)}.$$

Proof. First, observe that any change in resistance exceeding 1 is reflected immediately Otherwise, every time we update $counter(j)_e$, $\boldsymbol{r}_e$ can only increase additively by at most

$$2^{-j+1}\widehat{\boldsymbol{r}}_e.$$

Once $counter(j)_e$ exceeds 2^j, e will be added to $E_{changed}$ after at most 2^j steps. So when we start from $\widehat{\boldsymbol{r}}_e$, e is added to $E_{changed}$ after $counter(j)_e \leq 2^j + 2^j = 2^{j+1}$ iterations. The maximum possible increase in resistance due to the bucket j is,

$$2^{-j+1}\widehat{\boldsymbol{r}}_e \cdot 2^{j+1} = 4\widehat{\boldsymbol{r}}_e.$$

Since there are only at most $m^{1/3}$ iterations, the contributions of buckets with $j > \log m$ are negligible. Now the change in resistance is influenced by all buckets j, each contributing at most $4\widehat{\boldsymbol{r}}_e$ increase. The total change is at most $4\widehat{\boldsymbol{r}}_e \log m$ since there are at most $\log m$ buckets. We therefore have

$$\widehat{\boldsymbol{r}}_e \leq \boldsymbol{r}_e^{(i)} \leq 5\widehat{\boldsymbol{r}}_e \log m.$$

for every i. □

It remains to bound the number and sizes of calls made to Lemma 6.2. For this we define variables

$$k\,(\eta)^{(i)}$$

to denote the number of edges added to $E_{changed}$ at iteration i due to the value of $counter(\eta)_e$. Note that $k(\eta)^{(i)}$ is non-zero only if $i \equiv 0 \pmod{2^\eta}$, and

$$\left|E^{(i)}_{changed}\right| \leq \sum_{\eta} k\left(\eta\right)^{(i)}.$$

The following lemma gives a lower bound on the relative change of energy across one update of resistances.

Lemma 6.6. *Assuming the program* (4) *is feasible, let* Δ *be be a solution to the optimization problem* (4) *with weights* $\boldsymbol{r}$*. Suppose we increase the resistance on each entry to* $\boldsymbol{r}'$ *Then,*

$$\frac{\Psi\left(\boldsymbol{r}'\right) - \Psi\left(\boldsymbol{r}\right)}{\Psi\left(\boldsymbol{r}\right)} \geq \Omega_p\left(m^{-(p-2)/p}\sum_e \left(\Delta_e\right)^2 \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\}\right).$$

Proof. Lemma 5.11 gives us that,

$$\begin{aligned}\frac{\Psi\left(\boldsymbol{r}'\right)}{\Psi\left(\boldsymbol{r}\right)} - 1 &\geq \exp\left(\frac{\sum_e \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\}\boldsymbol{r}_e\Delta_e^2}{2\Psi\left(\boldsymbol{r}\right)}\right) - 1\\ &\geq \left(\frac{\sum_e \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\}\boldsymbol{r}_e\Delta_e^2}{2\Psi\left(\boldsymbol{r}\right)}\right)\end{aligned}$$

Since $\boldsymbol{r}_e \geq 1$ and $\Psi\left(\boldsymbol{r}\right) \leq O_p(m^{\frac{p-2}{p}})$,

$$\frac{\Psi\left(\boldsymbol{r}'\right) - \Psi\left(\boldsymbol{r}\right)}{\Psi\left(\boldsymbol{r}\right)} \geq \Omega_p\left(m^{-(p-2)/p}\sum_e \left(\Delta_e\right)^2 \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\}\right)$$

which gives our result. □

We divide our analysis into 2 cases, when the relative change in resistance is at least 1 and when the relative change in resistance is at most 1. To begin with, let us first look at the following lemma that relates the change in weights to the relative change in resistance. The proof is in the Appendix.

Lemma 6.7. *Consider a flow step from Line 13 of Algorithm 4. We have*

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \left(1 + \alpha\left|\Delta_e\right|\right)^{p-2} - 1$$

where Δ *is the* ℓ_2 *minimizer solution produced by the oracle.*

Let us now see what happens when the relative change in resistance is at least 1.

Lemma 6.8. *Throughout the course of a run of Algorithm 4, the number of edges added to* $E_{changed}$ *due to relative resistance increase of at least* 1*,*

$$\sum_{1\leq i\leq T} k\left(0\right)^{(i)} \leq \widetilde{O}_P\left(m^{\frac{p+2}{3p-2}}\right).$$

Proof. From Lemma 6.6, we know that the relative change in energy over one iteration is at least,

$$\Omega_p\left(m^{-(p-2)/p}\sum_e(\Delta_e)^2\min\left\{1,\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}}\right\}\right).$$

Over all iterations, the relative change in energy is at least,

$$\Omega_p\left(m^{-(p-2)/p}\sum_i\sum_e(\Delta_e)^2\min\left\{1,\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}}\right\}\right)$$

which is upper bounded by $O(\log m)$. When iteration i is a width reduction step, the relative resistance change is always at least 1. In this case $|\Delta_e|\geq\rho$. When we have a flow step, Lemma 6.7 implies that when the relative change in resistance is at least 1 then,

$$\alpha\,|\Delta_e|\geq\Omega_p(1).$$

This gives, $|\Delta_e|\geq\Omega_p(\alpha^{-1})$. Using this bound on $|\Delta_e|$ is sufficient since $\rho>\Omega_p(\alpha^{-1})$ and both kinds of iterations are accounted for. The total relative change in energy can now be bounded.

$$\begin{aligned}
&\Omega_p\left(m^{-(p-2)/p}\alpha^{-2}\sum_i\sum_e\mathbf{1}_{\left[\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}}\geq 1\right]}\right)\leq\widetilde{O}_p(1)\\
\Leftrightarrow\;&\Omega_p\left(m^{-(p-2)/p}\alpha^{-2}\sum_i k\,(0)^{(i)}\right)\leq\widetilde{O}_p(1)\\
\Leftrightarrow\;&\sum_i k\,(0)^{(i)}\leq\widetilde{O}_p(m^{(p-2)/p}\alpha^2).
\end{aligned}$$

The Lemma follows by substituting $\alpha=\tilde{\Theta}_p\left(m^{-\frac{p^2-5p+2}{p(3p-2)}}\right)$ in the above equation. □

Lemma 6.9. *Throughout the course of a run of Algorithm 4, the number of edges added to $E_{changed}$ due to relative resistance increase between $2^{-\eta}$ and $2^{-\eta+1}$,*

$$\sum_{1\leq i\leq T}k\,(\eta)^{(i)}\leq\begin{cases}0 & \text{if } 2^\eta\geq T,\\ \widetilde{O}_p\left(m^{\frac{p+2}{3p-2}}2^{2\eta}\right) & \text{otherwise.}\end{cases}$$

Proof. From Lemma 6.6, the total relative change in energy is at least,

$$\Omega_p\left(m^{-(p-2)/p}\sum_i\sum_e(\Delta_e)^2\left(\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}}\right)\right).$$

We know that $\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \geq 2^{-\eta}$. Using Lemma 6.7, we have,

$$(1+\alpha\left|\Delta_e\right|)^{p-2}-1 \geq 2^{-\eta}.$$

We can bound $(1+\alpha\left|\Delta_e\right|)^{p-2}-1$ as,

$$(1+\alpha\left|\Delta_e\right|)^{p-2}-1 \leq \begin{cases} \alpha\left|\Delta_e\right| & \text{when } \alpha\left|\Delta_e\right| \leq 1 \text{ or } p-2 \leq 1 \\ O_p\left(\left(\alpha\left|\Delta_e\right|\right)^{p-2}\right) & \text{otherwise.} \end{cases}$$

Now, in the second case, when $\alpha\left|\Delta_e\right| \geq 1$ and $p-2>1$,

$$\left(\alpha\left|\Delta_e\right|\right)^{p-2} \geq 2^{-\eta} \Rightarrow \alpha\left|\Delta_e\right| \geq \left(\frac{1}{2^{\eta}}\right)^{1/(p-2)} \geq 2^{-\eta}$$

For both cases we get,

$$\alpha\left|\Delta_e\right| \geq \Omega_p\left(2^{-\eta}\right).$$

Using the above bound and the fact that the total relative change in energy is at most $\widetilde{O}_p(1)$, gives,

$$\begin{aligned}
&\Omega_p\left(m^{-(p-2)/p}\sum_i\sum_e\left(\Delta_e\right)^2\left(\frac{\boldsymbol{r}_e^{(i+1)}-\boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}}\right)\right) \leq \widetilde{O}_p(1) \\
\Rightarrow &\Omega_p\left(m^{-(p-2)/p}\sum_i\sum_e\left(\alpha^{-1}2^{-\eta}\right)^2\cdot\left(2^{-\eta}\mathbf{1}_{2^{-\eta+1}\geq\frac{r_e^{(i+1)}-r_e^{(i)}}{r_e^{(i)}}\geq 2^{-\eta}}\right)\right) \leq \widetilde{O}_p(1) \\
\Rightarrow &\Omega_p\left(m^{-(p-2)/p}\alpha^{-2}2^{-3\eta}\sum_i 2^{\eta}k\left(\eta\right)^{(i)}\right) \leq \widetilde{O}_p(1) \\
\Rightarrow &\sum_i k\left(\eta\right)^{(i)} \leq \widetilde{O}_p\left(m^{(p-2)/p}\alpha^2 2^{2\eta}\right)
\end{aligned}$$

The Lemma follows substituting $\alpha=\tilde{\Theta}_p\left(m^{-\frac{p^2-5p+2}{p(3p-2)}}\right)$ in the above equation. □

We can now use the concavity of $f(z)=z^{\omega-2}$ to upper bound the contribution of these terms.

Corollary 6.10. *Let $k(\eta)^{(i)}$ be as defined. Over all iterations we have,*

$$\sum_i\left(k\left(0\right)^{(i)}\right)^{\omega-2} \leq \widetilde{O}_p\left(m^{\frac{p-(10-4\omega)}{3p-2}}\right)$$

and for every η,

$$\sum_i^T\left(k\left(\eta\right)^{(i)}\right)^{\omega-2} \leq \begin{cases} 0 & \text{if } 2^{\eta} \geq T, \\ \widetilde{O}_p\left(m^{\frac{p-2+4(\omega-2)}{3p-2}}\cdot 2^{\eta(3\omega-7)}\right) & \text{otherwise.} \end{cases}$$

Proof. Due to the concavity of the $\omega - 2 \approx 0.3727 < 1$ power, this total is maximized when it's equally distributed over all iterations. In the first sum, the number of terms is equal to the number of iterations, i.e., $\widetilde{O}_p(m^{\frac{p-2}{3p-2}})$. In the second sum the number of terms is $\widetilde{O}_p(m^{\frac{p-2}{3p-2}})2^{-\eta}$. Distributing the sum equally over the above numbers give,

$$\sum_i^T \left(k\left(0\right)^{(i)}\right)^{\omega-2} \leq \left(\widetilde{O}_p\left(m^{\frac{p+2}{3p-2}-\frac{p-2}{3p-2}}\right)\right)^{\omega-2} \cdot \widetilde{O}_p\left(m^{\frac{p-2}{3p-2}}\right) = \widetilde{O}_p\left(m^{\frac{p-2+4(\omega-2)}{3p-2}}\right) \leq \widetilde{O}_p\left(m^{\frac{p-(10-4\omega)}{3p-2}}\right)$$

and

$$\begin{aligned}
\sum_i^T \left(k\left(\eta\right)^{(i)}\right)^{\omega-2} &\leq \widetilde{O}\left(m^{\frac{p-2}{3p-2}}2^{-\eta}\right) \cdot \widetilde{O}_p\left(\frac{m^{\frac{p+2}{3p-2}}2^{2\eta}}{m^{\frac{p-2}{3p-2}}2^{-\eta}}\right)^{\omega-2} \\
&= \widetilde{O}_p\left(m^{\frac{p-2+4(\omega-2)}{3p-2}}2^{-\eta} \cdot 2^{3\eta(\omega-2)}\right) \\
&= \widetilde{O}_p\left(m^{\frac{p-2+4(\omega-2)}{3p-2}}2^{\eta(3\omega-7)}\right).
\end{aligned}$$

□

Proof. (of Theorem 6.1) By Lemma 6.5, the $\widehat{\boldsymbol{r}}$ that the inverse being maintained corresponds to always satisfy $\widehat{\boldsymbol{r}} \approx_{\widetilde{O}(1)} \boldsymbol{r}^{(i)}$. So by the iterative linear systems solver method outlined in Lemma 6.4, we can implement each call to ORACLE (Section 5.2)in time $O((n+m)^2)$ in addition to the cost of performing inverse maintenance. This leads to a total cost of

$$\widetilde{O}_p\left((n+m)^{2+\frac{p-2}{3p-2}}\right).$$

across the $T = \Theta_p(m^{\frac{p-2}{3p-2}})$ iterations.

The costs of inverse maintenance is dominated by the calls to the low-rank update procedure outlined in Lemma 6.2. Its total cost is bounded by

$$O\left(\sum_i \left|E^{(i)}_{changed}\right|^{\omega-2}(m+n)^2\right) = O\left((m+n)^2 \sum_i \left(\sum_\eta k\left(\eta\right)^{(i)}\right)^{\omega-2}\right).$$

Because there are only $O(\log m)$ values of η, and each $k(\eta)^{(i)}$ is non-negative, we can bound the total cost by:

$$\widetilde{O}\left((m+n)^2 \sum_i \sum_\eta \left(k\left(\eta\right)^{(i)}\right)^{\omega-2}\right) \leq \widetilde{O}_p\left((m+n)^2 \sum_{\eta: 2^\eta \leq T} m^{\frac{p-2+4(\omega-2)}{3p-2}} \cdot 2^{\eta(3\omega-7)}\right),$$

where the inequality follows from substituting in the result of Lemma 6.10. Depending on the sign of $3\omega - 7$, this sum is dominated either at $\eta = 0$ or $\eta = \log T$. Including both terms then gives

$$\widetilde{O}_p\left((m+n)^{2+\frac{p-2+4(\omega-2)}{3p-2}} + (m+n)^{2+\frac{p-2+4(\omega-2)+(p-2)(3\omega-7)}{3p-2}}\right),$$

with the exponent on the trailing term simplifying to $\omega - 2$ to give,

$$\widetilde{O}_p\left((m+n)^{2+\frac{p-(10-4\omega)}{3p-2}} + (m+n)^{\omega}\right).$$

□

7 Other Regression Formulations

In this section we discuss how various variants of ℓ_p-norm regression can be translated into our setting of

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_p \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \end{aligned} \tag{*}$$

As we will address a multitude of problems, we will make generic numerical assumptions to simplify the derivations.

1. $m \leq \text{poly}(n)$.
2. All entries in $\boldsymbol{A}$ and $\boldsymbol{b}$ are at most $\text{poly}(n)$. Note that this implies that the maximum singular value of $\boldsymbol{A}$, $\sigma_{\max}(\boldsymbol{A})$ and $\|\boldsymbol{b}\|_2$ are at most $\text{poly}(n)$.
3. $\|\boldsymbol{b}\|_2 \geq 1$.
4. The minimum non-zero singular value of $\boldsymbol{A}$, $\sigma_{\min}(\boldsymbol{A})$ is at least $1/\text{poly}(n)$.

Note that these conditions also imply bounds on the optimum value and the optimum solution $\boldsymbol{x}^*$: Specifically,

$$OPT \leq \left\|\boldsymbol{x}^*\right\|_2 \sqrt{m} \leq \|\boldsymbol{b}\|_2 \sigma_{\min}(\boldsymbol{A})^{-1} \leq \text{poly}\,(n)\,,$$

and

$$\begin{aligned} OPT \geq \left\|\boldsymbol{x}^*\right\|_2 \sqrt{m}^{-1} &\geq \|\boldsymbol{b}\|_2 \sigma_{\max}\,(\boldsymbol{A})^{-1} \cdot m^{-1/2} \\ &\geq \frac{1}{\text{poly}\,(n)}. \end{aligned}$$

7.1 Affine transformations within the norm

Let $\boldsymbol{C}$ be a matrix with the same assumptions as $\boldsymbol{A}$ and $\boldsymbol{d}$ have assumptions similar to $\boldsymbol{b}$. Suppose we are minimizing $\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{d}\|_p$ instead of $\|\boldsymbol{x}\|_p$, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{d}\|_p \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \end{aligned}$$

Note that this can be reduced to the following unconstrained problem,

$$\min_{\boldsymbol{x}} \|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{d}\|_p\,.$$

To see this, first find the null space of $\boldsymbol{A}$, as well as a particular solution $\boldsymbol{x}_0$ that satisfies $\boldsymbol{A}\boldsymbol{x}_0 = \boldsymbol{b}$. Let the null space of $\boldsymbol{A}$ be generated by the matrix $\boldsymbol{V}$. Then the space of solutions can be parameterized as

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{V}\boldsymbol{y},$$

for some vector $\boldsymbol{y}$. Now our objective becomes,

$$\boldsymbol{C}\boldsymbol{V}\boldsymbol{y} + (\boldsymbol{C}\boldsymbol{x}_0 - \boldsymbol{d}),$$

which can be written as,

$$\min_{\boldsymbol{y}} \left\| \widehat{\boldsymbol{C}}\boldsymbol{y} - \widehat{\boldsymbol{d}} \right\|_p.$$

where $\widehat{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{V}$ and $\widehat{\boldsymbol{d}} = \boldsymbol{C}\boldsymbol{x}_0 - \boldsymbol{d}$. Observe that $\widehat{\boldsymbol{C}}y$ spans the column space of $\widehat{\boldsymbol{C}}$. Decomposing $\boldsymbol{d}$ into a linear combination of an orthonormal basis we could combine the part which is in the span of $\widehat{\boldsymbol{C}}$ with $\widehat{\boldsymbol{C}}\boldsymbol{y}$. We can thus replace $\widehat{\boldsymbol{C}}$ in the objective with an orthonormal basis $\boldsymbol{U}$ of its column space and replace $\widehat{\boldsymbol{d}}$ by $\boldsymbol{g}$, a vector orthogonal to all columns of $\boldsymbol{U}$. Then any vector

$$\boldsymbol{z} = \boldsymbol{U}\boldsymbol{y} - \boldsymbol{g},$$

can equivalently be described by the conditions

$$\begin{aligned} \boldsymbol{z}^\top \widehat{\boldsymbol{d}}_\perp &= \|\boldsymbol{g}\|_2^2, \\ \boldsymbol{z}^\top \boldsymbol{v} &= 0, \qquad \forall \boldsymbol{v}\ s.t.\ \boldsymbol{U}\boldsymbol{v} = 0, \boldsymbol{g}^\top \boldsymbol{v} = 0. \end{aligned}$$

For the last condition, it suffices to generate an orthonormal basis of the null space of $\boldsymbol{U}$. So the problem can be written as a linear constraint on $\boldsymbol{z}$ instead.

7.2 $1 < p < 2$

In case $1 < p < 2$, we instead solve the dual problem:

$$\begin{aligned} \max_{\boldsymbol{y}} \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ & \left\| \boldsymbol{A}^\top \boldsymbol{y} \right\|_q \leq 1, \end{aligned}$$

for $q = \frac{p}{p-1} > 2$. We can rescale the above problem to the equivalent q-norm ball-constrained projection problem,

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \left\| \boldsymbol{A}^\top \boldsymbol{y} \right\|_q \\ & \boldsymbol{b}^\top \boldsymbol{y} = 1, \end{aligned}$$

where the goal is to check whether the optimum is less than 1. This problem is covered by the problem introduced in Section 7.1 and can thus be solved to high accuracy in the desired time.

It remains is to transform a nearly-optimal solution $\boldsymbol{y}$ of this q-norm ball-constrained projection problem to a nearly-optimal solution $\boldsymbol{x}$ of the original subspace p-norm minimization problem. Since both of these problems' solutions are invariant under scalings to $\boldsymbol{A}$ or $\boldsymbol{b}$, we may also assume that the optimum is at most 1.

Lemma 7.1. *If the optimum of*

$$\begin{aligned} \max \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ & \left\| \boldsymbol{A}^\top \boldsymbol{y} \right\|_q \leq 1, \end{aligned}$$

is at most 1*, and we have some* $\boldsymbol{y}$ *such that*

$$\begin{aligned} \boldsymbol{b}^\top \boldsymbol{y} &\geq 1 - \delta \\ \left\| \boldsymbol{A}^\top \boldsymbol{y} \right\|_q &= 1, \end{aligned}$$

then the gradient of $\left\| \boldsymbol{A}^\top \boldsymbol{y} \right\|_q$*,*

$$\nabla = \boldsymbol{A}\text{sgn}(\boldsymbol{A}^\top \boldsymbol{y}) \left(\boldsymbol{A}^\top \boldsymbol{y} \right)^{q-1},$$

satisfies

$$\|\nabla - \boldsymbol{b}\|_2 \leq \delta poly(n).$$

Proof. Let $\Delta = \nabla - \boldsymbol{b}$ and $p(n)$ be a polynomial such that $\gamma_q \left(\left| \boldsymbol{A}^\top \boldsymbol{y} \right|, \boldsymbol{A}^\top (\nabla - b) \right) \leq p(n)$. By the assumption of $\boldsymbol{A}$ and $\boldsymbol{b}$ being poly(n) bounded, the above γ function is polynomially bounded. Let $q(n)$ be a polynomial in n such that, $q(n) \geq \sqrt{\frac{4p(n)}{\delta}}$. Suppose,

$$\|\nabla - \boldsymbol{b}\|_2^2 \geq \varepsilon > \delta q(n).$$

This gives us,

$$\Delta^\top \nabla \geq \Delta^\top \boldsymbol{b} + \varepsilon.$$

Now consider the solution

$$\widehat{\boldsymbol{y}}\,(\theta) \leftarrow \boldsymbol{y} - \theta\Delta,$$

for step size $\theta = \varepsilon / 2p(n)$. Lemma 4.5 and Lemma 3.3 gives

$$\begin{aligned} \left\| \boldsymbol{A}^\top \widehat{\boldsymbol{y}} \right\|_q^q &\leq 1 - \theta\Delta^\top \nabla + \gamma_q \left(\left| \boldsymbol{A}^\top \boldsymbol{y} \right|, \theta \boldsymbol{A}^\top \Delta \right) \\ &\leq 1 - \theta\Delta^\top \nabla + \theta^2 \gamma_q \left(\left| \boldsymbol{A}^\top \boldsymbol{y} \right|, \boldsymbol{A}^\top \Delta \right) \\ &\leq 1 - \theta\Delta^\top \boldsymbol{b} - \theta\epsilon + \theta^2 p(n) \\ &= 1 - \theta\Delta^\top \boldsymbol{b} - \frac{\theta\epsilon}{2}. \end{aligned}$$

We can scale the solution $\widehat{\boldsymbol{y}}$ up by a factor of $1/(1 - \theta\Delta^\top \boldsymbol{b} - \theta\epsilon/2)$ to get a solution with objective value

$$\frac{1 - \delta - \theta\Delta^\top \boldsymbol{b}}{1 - \theta\Delta^\top \boldsymbol{b} - \theta\epsilon/2}.$$

But by the assumption of 1 being the optimum, this cannot exceed 1, so we get

$$1 - \delta - \theta\Delta^\top \boldsymbol{b} \leq 1 - \theta\Delta^\top \boldsymbol{b} - \theta\epsilon/2,$$

or

$$\theta\epsilon \leq 2\delta,$$

which combined with the choice of θ gives $\varepsilon < \delta q(n)$ which is a contradiction. So we much have, $\|\nabla - \boldsymbol{b}\|_2 \leq \delta q(n) \leq \delta \text{poly}(n)$ □

This means once $\delta \leq 1/\text{poly}(n)$, the solution created from the gradient

$$\widehat{\boldsymbol{x}} \leftarrow \text{sgn}(\boldsymbol{A}^\top \boldsymbol{y}) \left(\boldsymbol{A}^\top \boldsymbol{y}\right)^{q-1},$$

satisfies

$$\|\boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 \leq \delta \text{poly}\left(n\right).$$

Also, because $\sigma_{\min}(\boldsymbol{A}) \geq 1/\text{poly}(n)$, we can create a solution $\widetilde{\boldsymbol{x}}$ from $\widehat{\boldsymbol{x}}$ by doing a least squares projection on this difference. This gives:

$$\boldsymbol{A}\widetilde{\boldsymbol{x}} = \boldsymbol{b},$$

and

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\| \leq \delta \text{poly}\left(n\right) \sigma_{\min}\left(\boldsymbol{A}\right) \leq \delta \text{poly}\left(n\right).$$

Furthermore, note that because

$$\left(z^{q-1}\right)^p = z^{(q-1)(p-1)+q-1} = z^q,$$

we have

$$\|\nabla\|_p^p = \left\|\boldsymbol{A}\text{sgn}(\boldsymbol{A}^\top \boldsymbol{y}) \left(\boldsymbol{A}^\top \boldsymbol{y}\right)^{q-1}\right\|_p^p = \left\|\boldsymbol{A}^\top \boldsymbol{y}\right\|_q^q = 1,$$

so

$$\|\boldsymbol{x}\|_p \leq \|\nabla\|_p + \|\boldsymbol{x} - \nabla\|_p \leq 1 + \delta \text{poly}\left(n\right).$$

Thus, for sufficiently small δ, we can get high accuracy answer to the p-norm problem as well.

8 p-Norm Optimization on Graphs

In this section we discuss the performance of our algorithms on graphs. Here instead of invoking general linear algebraic routines, we instead invoke Laplacian solvers, which provide $1/\text{poly}(n)$ accuracy solutions to Laplacian linear equations in nearly-linear ($\widetilde{O}(m)$) time [Coh+14; Kel+13; KMP11; KMP14; KS16; Kyn+16; PS14; ST14], and the current best running time is $O(m \log^{1/2} n \log 1/\varepsilon)$ (up to polyloglog n factors) [Coh+14].

Such matrices can be succinctly described as

$$\boldsymbol{A}^\top \textbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A}$$

where $\boldsymbol{r}$ is the vector of resistances just as provided in the Oracle from Algorithm 3, but $\boldsymbol{A}$ is the edge-vertex incidence matrix: with each row corresponding to an edge, each column corresponding to a vertex, and entries given by:

$$\boldsymbol{A}_{e,v} = \begin{cases} 1 & \text{if } v \text{ is the head of } e, \\ -1 & \text{if } v \text{ is the tail of } e, \\ 0 & \text{otherwise.} \end{cases}$$

Throughout this entire section, we will use $\boldsymbol{A}$ to refer to the edge-vertex incidence matrix of a graph.

The main difficult of reducing to Laplacian solvers is that we can no longer manipulate general matrices. Specifically, instead of directly working with the normal matrices as in Section 7.1, we need to implicitly track the subspaces, and optimize quadratics on them. As a result, we need to tailor such reductions towards the specific problems.

8.1 p-Norm Flows

This is closest to the general regression problem that we study:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_p \\ & \boldsymbol{A}^\top \boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

except with $\boldsymbol{A}$ as an edge vertex incidence matrix.

When $p \geq 2$, the residual problem then has an extra condition of

$$\boldsymbol{g}^\top \boldsymbol{f} = \alpha,$$

which means we need to solve the problem of

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \sum_e \boldsymbol{r}_e \boldsymbol{x}_e^2 \\ & \boldsymbol{A}^\top \boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{g}^\top \boldsymbol{x} = \alpha \end{aligned}$$

which becomes a solve in the system of linear equations

$$[\boldsymbol{A}; \boldsymbol{x}]^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} [\boldsymbol{A}; \boldsymbol{x}] = \begin{bmatrix} \boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A} & \boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{x} \\ \boldsymbol{x}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A} & \boldsymbol{x}^\top \boldsymbol{r}^{-1} \boldsymbol{x}^\top \end{bmatrix}.$$

This matrix is a rank 3 perturbation to the graph Laplacian $\boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A}$, and can thus be solved in $\widetilde{O}(m)$ time. A more detailed analysis of a generalization of this case can be found in Appendix B of [DS08].

When $1 < p < 2$, we invoke the dualization from Section 7.2 to obtain

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \|\boldsymbol{A}\boldsymbol{y}\|_q \\ & \boldsymbol{b}^\top \boldsymbol{y} = \alpha \end{aligned}$$

and if we retain the form of $\boldsymbol{A}^\top \boldsymbol{y}$, but transfer the gradient over to $\boldsymbol{y}$, the problem that we get is:

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \boldsymbol{y}^\top \boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A}\boldsymbol{y} \\ & \boldsymbol{b}^\top \boldsymbol{y} = \alpha \\ & \boldsymbol{g}^\top \boldsymbol{y} = \beta \end{aligned}$$

The two additional linear constraints can removed by writing a variable of $\boldsymbol{y}$ as a linear combination of the rest (as well as α). This then gives an unconstrained minimization problem on a subset of entries S,

$$\min_{\boldsymbol{y}_S} \boldsymbol{y}_S^\top \boldsymbol{L}_{[S,S]} \boldsymbol{y}_S + \boldsymbol{c}^\top \boldsymbol{y}_S$$

where $\boldsymbol{L}_{[S,S]}$ is a minor of the Laplacian above, and this solution is obtained by solving for

$$\boldsymbol{y}_S \leftarrow \frac{1}{2} \boldsymbol{L}_{[S,S]}^\dagger \boldsymbol{c}.$$

8.2 Lipschitz Learning and Graph Labelling

This problem asks to label the vertices of a graph, with a set T fixed to the vector $\boldsymbol{s}$, while minimizing the p-norm difference between neighbours. It can be written as

$$\min_{\boldsymbol{x}:\boldsymbol{x}|T=\boldsymbol{s}} \quad \|\boldsymbol{A}\boldsymbol{x}\|_p^p$$

where $\boldsymbol{A}$ is the edge-vertex incidence matrix.

In the case of $p \geq 2$, the residue problem becomes

$$\begin{aligned} \min_{\boldsymbol{x}:\boldsymbol{x}|T=\boldsymbol{s}} \quad & \boldsymbol{x}^\top \boldsymbol{A}^\top \mathbf{Diag}\left(\boldsymbol{r}\right)^{-1} \boldsymbol{A}\boldsymbol{x} \\ & \boldsymbol{g}^\top \boldsymbol{x} = \beta \end{aligned}$$

Here the gradient condition can be handled in the same way as with the voltage problem above: by fixing one additional entry of $V \setminus T$, and then solving an unconstrained quadratic minimization problem on the rest of the variables.

In the case of $1 < p < 2$, we first write down the problem as an unconstrained minimization problem on $V \setminus T$:

$$\min_{\boldsymbol{x}_{V\setminus T}} \quad \left\| \boldsymbol{A}_{:,V\setminus T} \boldsymbol{x}_{V\setminus T} - \boldsymbol{A}_{:,T} \boldsymbol{s} \right\|_p.$$

Let $\boldsymbol{b} = \boldsymbol{A}_{:,T}\boldsymbol{s}$ and taking the dual gives:

$$\begin{aligned} \max \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ & \|\boldsymbol{y}\|_q \leq 1 \\ & \left(\boldsymbol{A}^\top\right)_{V\setminus T,:} \boldsymbol{y} = 0 \end{aligned}$$

That is, solving for a small q-norm flow that maximizes the cost against $\boldsymbol{b}$, while also having 0 residues at the vertices not in T.

As $q > 2$, we can now invoke our main algorithm on $\boldsymbol{y}$. Upon binary search, and taking residual problems, we get ℓ_2 problems of the form

$$\begin{aligned} \min \quad & \sum_e \boldsymbol{r}_e \boldsymbol{y}_e^2 \\ & \left(\boldsymbol{A}^\top\right)_{V\setminus T,:} \boldsymbol{y} = 0 \\ & \boldsymbol{b}^\top \boldsymbol{y} = \alpha \\ & \boldsymbol{g}^\top \boldsymbol{y} = \beta, \end{aligned}$$

which is solved by another low rank perturbation on a minor of the graph Laplacian.

References

[Abr+16] I. Abraham, D. Durfee, I. Koutis, S. Krinninger, and R. Peng. "On Fully Dynamic Graph Sparsifiers". In: *Symposium on Foundations of Computer Science (FOCS)*. Available at: http://arxiv.org/abs/1604.02094. 2016, pp. 335–344 (cit. on p. 5).

[Adi+] D. Adil, R. Kyng, R. Peng, and S. Sachdeva. "Iterative Refinement for ℓ_p-norm Regression". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1405–1424. eprint: `https://epubs.siam.org/doi/pdf/10.1137/1.9781611975482.86` (cit. on p. 1).

[AHK12] S. Arora, E. Hazan, and S. Kale. "The Multiplicative Weights Update Method: a Meta-Algorithm and Applications." In: *Theory of Computing* 8.1 (2012), pp. 121–164 (cit. on p. 5).

[All+17] Z. Allen-Zhu, Y. Li, R. M. de Oliveira, and A. Wigderson. "Much Faster Algorithms for Matrix Scaling". In: *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. Available at: https://arxiv.org/abs/1704.02315. 2017, pp. 890–901 (cit. on p. 4).

[Axe94] O. Axelsson. *Iterative Solution Methods*. New York, NY: Cambridge University Press, 1994. ISBN: 0-521-44524-8 (cit. on p. 3).

[Bub+18] S. Bubeck, M. B. Cohen, Y. T. Lee, and Y. Li. "An Homotopy Method for Lp Regression Provably Beyond Self-concordance and in Input-sparsity Time". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2018. Los Angeles, CA, USA: ACM, 2018, pp. 1130–1137. ISBN: 978-1-4503-5559-9 (cit. on pp. 1, 2, 4, 6, 48).

[Bul18] B. Bullins. "Fast minimization of structured convex quartics". In: *CoRR* abs/11812.10349 (2018). https://arxiv.org/abs/1812.10349 (cit. on p. 2).

[Chi+13] H. H. Chin, A. Madry, G. L. Miller, and R. Peng. "Runtime guarantees for regression problems". In: *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. ITCS '13. Available at http://arxiv.org/abs/1110.1358. Berkeley, California, USA: ACM, 2013, pp. 269–282. ISBN: 978-1-4503-1859-4 (cit. on pp. 1, 2, 4, 5).

[Chr+11] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng. "Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs". In: *Proceedings of the 43rd annual ACM symposium on Theory of computing*. STOC '11. Available at http://arxiv.org/abs/1010.2921. San Jose, California, USA: ACM, 2011, pp. 273–282. ISBN: 978-1-4503-0691-1 (cit. on pp. 2, 3, 4, 5, 13, 15, 18).

[CLS18] M. B. Cohen, Y. T. Lee, and Z. Song. "Solving Linear Programs in the Current Matrix Multiplication Time". In: *CoRR* abs/1810.07896 (2018). Available at: http://arxiv.org/abs/1810.07896 (cit. on p. 3).

[Coh+14] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu. "Solving SDD Linear Systems in Nearly Mlog1/2N Time". In: *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*. STOC '14. New York, New York: ACM, 2014, pp. 343–352. ISBN: 978-1-4503-2710-7 (cit. on p. 33).

[Coh+17a] M. B. Cohen, A. Madry, P. Sankowski, and A. Vladu. "Negative-Weight Shortest Paths and Unit Capacity Minimum Cost Flow in $\tilde{O}(m^{10/7} \log W)$ Time (Extended Abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. Available at: https://arxiv.org/abs/1605.01717. 2017, pp. 752–771 (cit. on p. 4).

[Coh+17b] M. B. Cohen, A. Madry, D. Tsipras, and A. Vladu. "Matrix Scaling and Balancing via Box Constrained Newton's Method and Interior Point Methods". In: *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. Available at: https://arxiv.org/abs/1704.02310. 2017, pp. 902–913 (cit. on p. 4).

[CP15] M. B. Cohen and R. Peng. "ℓ_p Row Sampling by Lewis Weights". In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. STOC '15. Available at http://arxiv.org/abs/1412.0588. Portland, Oregon, USA: ACM, 2015, pp. 183–192. ISBN: 978-1-4503-3536-2 (cit. on p. 4).

[DS08] S. I. Daitch and D. A. Spielman. "Faster approximate lossy generalized flow via interior point algorithms". In: *Proceedings of the 40th annual ACM symposium on Theory of computing*. STOC '08. Available at http://arxiv.org/abs/0803.0988. Victoria, British Columbia, Canada: ACM, 2008, pp. 451–460. ISBN: 978-1-60558-047-0 (cit. on p. 34).

[Gha+15] M. Ghaffari, A. Karrenbauer, F. Kuhn, C. Lenzen, and B. Patt-Shamir. "Near-Optimal Distributed Maximum Flow: Extended Abstract". In: *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastin, Spain, July 21 - 23, 2015*. Available at: https://arxiv.org/abs/1508.04747. 2015, pp. 81–90 (cit. on p. 4).

[GP13] M. Gupta and R. Peng. "Fully Dynamic $1+\epsilon$-Approximate Matchings". In: *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. Available at http://arxiv.org/abs/1304.0378. 2013, pp. 548–557 (cit. on p. 5).

[Hen03] V. E. Henson. "Multigrid methods nonlinear problems: an overview". In: *Computational Imaging, Santa Clara, CA, USA, January 20, 2003*. 2003, pp. 36–48 (cit. on p. 3).

[Kel+13] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. "A simple, combinatorial algorithm for solving sdd systems in nearly-linear time". In: *STOC*. 2013 (cit. on p. 33).

[Kel+14] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. "An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*. Available at http://arxiv.org/abs/1304.2338. 2014, pp. 217–226 (cit. on p. 4).

[Kel99] C. T. Kelley. *Iterative methods for optimization*. Vol. 18. Siam, 1999 (cit. on p. 3).

[KK04] D. A. Knoll and D. E. Keyes. "Jacobian-free Newton–Krylov methods: a survey of approaches and applications". In: *Journal of Computational Physics* 193.2 (2004), pp. 357–397 (cit. on p. 3).

[KMP11] I. Koutis, G. L. Miller, and R. Peng. "A Nearly-m log n Time Solver for SDD Linear Systems". In: *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. FOCS '11. Available at http://arxiv.org/abs/1102.4842. Washington, DC, USA: IEEE Computer Society, 2011, pp. 590–598. ISBN: 978-0-7695-4571-4 (cit. on p. 33).

[KMP12] J. A. Kelner, G. L. Miller, and R. Peng. "Faster approximate multicommodity flow using quadratically coupled flows". In: *Proceedings of the 44th symposium on Theory of Computing*. STOC '12. Available at http://arxiv.org/abs/1202.3367. New York, New York, USA: ACM, 2012, pp. 1–18. ISBN: 978-1-4503-1245-5 (cit. on p. 4).

[KMP14] I. Koutis, G. Miller, and R. Peng. "Approaching Optimality for Solving SDD Linear Systems". In: *SIAM Journal on Computing* 43.1 (2014). Available at http://arxiv.org/abs/1003.2958, pp. 337–354 (cit. on p. 33).

[KRS15] R. Kyng, A. B. Rao, and S. Sachdeva. "Fast, Provable Algorithms for Isotonic Regression in all ℓ_p-norms". In: *NIPS*. 2015 (cit. on p. 4).

[KS16] R. Kyng and S. Sachdeva. "Approximate Gaussian Elimination for Laplacians - Fast, Sparse, and Simple". In: *FOCS*. Available at http://arxiv.org/abs/1605.02353. IEEE Computer Society, 2016, pp. 573–582 (cit. on p. 33).

[Kyn+15] R. Kyng, A. B. Rao, S. Sachdeva, and D. A. Spielman. "Algorithms for Lipschitz learning on graphs". In: *COLT*. 2015 (cit. on pp. 3, 4).

[Kyn+16] R. Kyng, Y. T. Lee, R. Peng, S. Sachdeva, and D. A. Spielman. "Sparsified Cholesky and multigrid solvers for connection laplacians". In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. Available at http://arxiv.org/abs/1512.01892. ACM. 2016, pp. 842–850 (cit. on p. 33).

[Le 14] F. Le Gall. "Powers of tensors and fast matrix multiplication". In: *Proceedings of the 39th international symposium on symbolic and algebraic computation*. Available at: https://arxiv.org/abs/1401.7714. ACM. 2014, pp. 296–303 (cit. on p. 1).

[Lee17] Y. T. Lee. "Uniform Sampling and Inverse Maintenance". Talk at Michael Cohen Memorial Symposium, Available at: https://simons.berkeley.edu/talks/welcome-and-birds-eye-view-michaels-work. 2017 (cit. on p. 3).

[LRS13] Y. T. Lee, S. Rao, and N. Srivastava. "A new approach to computing maximum flows using electrical flows". In: *STOC*. 2013 (cit. on p. 4).

[LS14] Y. T. Lee and A. Sidford. "Path Finding Methods for Linear Programming: Solving Linear Programs in $\tilde{O}(\sqrt{rank})$ Iterations and Faster Algorithms for Maximum Flow". In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. Available at http://arxiv.org/abs/1312.6677 and http://arxiv.org/abs/1312.6713. IEEE. 2014, pp. 424–433 (cit. on pp. 1, 4, 5, 22).

[LS15] Y. T. Lee and A. Sidford. "Efficient Inverse Maintenance and Faster Algorithms for Linear Programming". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. Available at: https://arxiv.org/abs/1503.01752. 2015, pp. 230–249 (cit. on pp. 3, 4, 5, 22).

[LSW15] Y. T. Lee, A. Sidford, and S. C. Wong. "A Faster Cutting Plane Method and its Implications for Combinatorial and Convex Optimization". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. Available at: https://arxiv.org/abs/1508.04874. 2015, pp. 1049–1065 (cit. on p. 4).

[Mad11] A. Madry. Private Communication. 2011 (cit. on p. 2).

[Mad13] A. Madry. "Navigating Central Path with Electrical Flows: From Flows to Matchings, and Back". In: *FOCS*. 2013 (cit. on pp. 1, 3, 4).

[Mad16] A. Madry. "Computing Maximum Flow with Augmenting Electrical Flows". In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. Available at: https://arxiv.org/abs/1608. 2016, pp. 593–602 (cit. on pp. 3, 4).

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994. eprint: `https://epubs.siam.org/d` (cit. on p. 1).

[NW06] J. Nocedal and S. J. Wright. *Nonlinear Equations*. Springer, 2006 (cit. on p. 3).

[Pen16] R. Peng. "Approximate undirected maximum flows in $O(m \text{ polylog}(n))$ time". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Available at http://arxiv.org/abs/1411.7631. SIAM. 2016, pp. 1862–1867 (cit. on pp. 1, 4).

[PS14] R. Peng and D. A. Spielman. "An Efficient Parallel Solver for SDD Linear Systems". In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. STOC '14. Available at http://arxiv.org/abs/1311.3286. New York, New York: ACM, 2014, pp. 333–342. ISBN: 978-1-4503-2710-7 (cit. on p. 33).

[RST14] H. Racke, C. Shah, and H. Taubig. "Computing Cut-Based Hierarchical Decompositions in Almost Linear Time". In: *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '14. 2014, pp. 227–238 (cit. on p. 4).

[Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. 2nd. Available at http://www-users.cs.umn.edu/~saad/toc.pdf. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003. ISBN: 0898715342 (cit. on p. 3).

[She13] J. Sherman. "Nearly Maximum Flows in Nearly Linear Time". In: *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. Available at http://arxiv.org/abs/1304.2077. 2013, pp. 263–269 (cit. on p. 4).

[She17a] J. Sherman. "Area-convexity, l_∞ regularization, and undirected multicommodity flow". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. 2017, pp. 452–460 (cit. on pp. 1, 4).

[She17b] J. Sherman. "Generalized Preconditioning and Undirected Minimum-cost Flow". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '17. Available at: https://arxiv.org/abs/1606.07425. Barcelona, Spain, 2017, pp. 772–780 (cit. on pp. 1, 4).

[Sid17] A. Sidford. "Michael Cohen and Oblivious Routing". Talk at Michael Cohen Memorial Symposium, Available at: https://simons.berkeley.edu/talks/tba-4. 2017 (cit. on p. 1).

[ST14] D. Spielman and S. Teng. "Nearly Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems". In: *SIAM Journal on Matrix Analysis and Applications* 35.3 (2014). Available at http://arxiv.org/abs/cs/0607105, pp. 835–885 (cit. on pp. 3, 33).

[Ten10] S.-H. Teng. "The Laplacian Paradigm: Emerging Algorithms for Massive Graphs". In: *Theory and Applications of Models of Computation*. 2010, pp. 2–14 (cit. on p. 3).

[Til13] A. Tillmann. *Computational aspects of compressed sensing*. Verlag Dr. Hut, 2013 (cit. on p. 1).

[Til15] A. M. Tillmann. "Equivalence of Linear Programming and Basis Pursuit". In: *PAMM* 15.1 (2015), pp. 735–738 (cit. on p. 1).

[Vai89] P. M. Vaidya. "Speeding-up linear programming using fast matrix multiplication". In: *30th Annual Symposium on Foundations of Computer Science*. Oct. 1989, pp. 332–337 (cit. on pp. 3, 4, 5).

[Vai90] P. M. Vaidya. "Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners." Unpublished manuscript UIUC 1990. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991, Minneapolis. 1990 (cit. on pp. 3, 22).

[Wil12] V. V. Williams. "Multiplying Matrices Faster Than Coppersmith-winograd". In: *STOC*. 2012 (cit. on p. 1).

A Missing Proofs

A.1 Proofs from Section 3

Lemma 3.2. *Function γ_p is as defined above.*

1. *For any $p \geq 2$, $t \geq 0$, and $x \in \mathbb{R}$, we have $\gamma_p(t,x) \geq |x|^p$, and $\gamma_p(t,x) \geq \frac{p}{2}t^{p-2}x^2$.*
2. *It is homogeneous under rescaling of* both *t and x, i.e., for any $t, \lambda \geq 0, p \geq 1$, and any x we have $\gamma_p(\lambda t, \lambda x) = \lambda^p \gamma_p(t,x)$.*
3. *For any $t > 0, p \geq 1$ and any x, we have $\gamma_p'(t,x) = p\max\{t,|x|\}^{p-2}x$.*

Proof. 1. We have $p \geq 2$. When $|x| \leq t$,

$$\gamma_p(t,x) = \frac{p}{2}t^{p-2}x^2 \geq \frac{p}{2}|x|^p \geq |x|^p .$$

Otherwise,

$$\gamma_p(t,x) = |x|^p + \left(\frac{p}{2} - 1\right)t^p \geq |x|^p .$$

Let $s(x) = \frac{p}{2}t^{p-2}x^2$. At $|x| = t$ we have $\gamma_p(t,t) = s(t)$. Now, $\gamma_p'(t,x) = p|x|^{p-2}x \geq pt^{p-2}x = s'(x)$. This means that for x negative, γ_p decreases faster than s and for x positive, γ_p increases faster than s. The two functions are equal in the range $-t \leq x \leq t$. Therefore, $\gamma_p(t,x) \geq s(x)$ for all x.

2.

$$\begin{aligned}\gamma_p(\lambda t, \lambda x) &= \begin{cases} \frac{p}{2}(\lambda t)^{p-2}(\lambda x)^2 & \text{if } |x| \le t, \\ |\lambda x|^p + (\frac{p}{2}-1)(\lambda t)^p & \text{otherwise .} \end{cases} \\ &= \begin{cases} \lambda^p \frac{p}{2} t^{p-2} x^2 & \text{if } |x| \le t, \\ \lambda^p |x|^p + \lambda^p (\frac{p}{2}-1) t^p & \text{otherwise .} \end{cases} \\ &= \lambda^p \gamma_p(t,x)\end{aligned}$$

3. Taking the derivative of $\gamma_p(t,x)$ with respect to x gives,

$$\frac{d}{dx}\gamma_p = \begin{cases} pt^{p-2}x & \text{if } |x| \le t, \\ p\,|x|^{p-1} \cdot sign(x) & \text{otherwise .} \end{cases}$$

The statement clearly follows. □

Lemma 3.3. *For any $p > 1, \Delta \in \mathbb{R}$ and $\lambda \ge 0$, we have,*

$$\min\{2,p\} \le x\frac{\gamma_p'(t,x)}{\gamma_p(t,x)} \le \max\{2,p\}.$$

This implies,

$$\min\{\lambda^2, \lambda^p\}\gamma_p(t,\Delta) \le \gamma_p(t,\lambda\Delta) \le \max\{\lambda^2,\lambda^p\}\gamma_p(t,\Delta).$$

Proof.

$$x\frac{\gamma_p'(t,x)}{\gamma_p(t,x)} = \begin{cases} 2 & \text{if } |x| \le t, \\ \frac{p|x|^p}{|x|^p + (\frac{p}{2}-1)t^p} & \text{otherwise .} \end{cases}$$

Now, when $|x| \ge t$, we have the following. When $p \le 2$,

$$p = \frac{p\,|x|^p}{|x|^p} \le \frac{p\,|x|^p}{|x|^p + \left(\frac{p}{2}-1\right)t^p} \le \frac{p\,|x|^p}{\frac{p}{2}\,|x|^p} = 2$$

and when $p \ge 2$,

$$2 = \frac{p\,|x|^p}{\frac{p}{2}\,|x|^p} \le \frac{p\,|x|^p}{|x|^p + \left(\frac{p}{2}-1\right)t^p} \le \frac{p\,|x|^p}{|x|^p} = \frac{p}{x}$$

The above computations imply that,

$$\min\{2,p\} \le x\frac{\gamma_p'(t,x)}{\gamma_p(t,x)} \le \max\{2,p\}.$$

Let $\lambda \ge 1$ and $x \ge 0$. Integrating both sides of the right inequality gives,

$$\int_x^{\lambda x} \frac{\gamma_p'(t,x)}{\gamma_p(t,x)}dx \le \int_x^{\lambda x} \frac{\max\{2,p\}}{x}dx$$

$$\Leftrightarrow \log\left(\frac{\gamma_p(t,\lambda x)}{\gamma_p(t,x)}\right) \leq \log(\lambda)^{\max\{2,p\}}$$
$$\Leftrightarrow \gamma_p(t,\lambda x) \leq \lambda^{\max\{2,p\}}\gamma_p(t,x).$$

Integrating both sides of the left inequality from x to λx gives the required left inequality. Now, let $\lambda \leq 1$. Integrating both sides of the left inequality gives,

$$\int_{\lambda x}^{x} \frac{\gamma_p'(t,x)}{\gamma_p(t,x)}dx \geq \int_{\lambda x}^{x} \frac{\min\{2,p\}}{x}dx$$
$$\Leftrightarrow \log\left(\frac{\gamma_p(t,x)}{\gamma_p(t,\lambda x)}\right) \geq \log\left(\frac{1}{\lambda}\right)^{\min\{2,p\}}$$
$$\Leftrightarrow \gamma_p(t,\lambda x) \leq \lambda^{\min\{2,p\}}\gamma_p(t,x).$$

Similar to the previous case, integrating both sides of the right inequality from λx to x gives the required left inequality. When $x \leq 0$, the direction of the inequality changes but it gets reversed again after putting limits, since we integrate from λx to x when $\lambda \geq 1$ and x to λx when $\lambda \leq 1$. We thus have,

$$\min\{\lambda^2, \lambda^p\}\gamma_p(t,\Delta) \leq \gamma_p(t,\lambda x) \leq \max\{\lambda^2, \lambda^p\}\gamma_p(t,x)$$

□

Lemma 3.4. *For any $p \geq 2, t \geq 0$ and any x, Δ, we have*

$$\gamma_p(t, x+\Delta) \leq \gamma_p(t,x) + \left|\gamma_p'(t,x)\Delta\right| + p^2 2^{p-3}\max\{t, |x|, |\Delta|\}^{p-2}\Delta^2.$$

Proof. Since $\gamma_p(t,x) = \gamma_p(t,|x|)$, and $\gamma_p(t,x)$ is increasing in x, it suffices to prove the claim for $x, \Delta \geq 0$. We have,

$$\begin{aligned}
\gamma_p'(t,x+z) - \gamma_p'(t,x) &= p\max\{t,\ x+z\}^{p-2}(x+z) - p\max\{t,\ |x|\}^{p-2}x \\
&= p\max\{t^{p-2}(x+z) - \max\{t,\ |x|\}^{p-2}x, \\
&\qquad (x+z)^{p-1} - \max\{t,\ |x|\}^{p-2}x\} \\
&\leq p\max\{t^{p-2}(x+z) - t^{p-2}x,\ (x+z)^{p-1} - x^{p-1}\} && \text{(Since } p \geq 2\text{)} \\
&\leq p\max\{t^{p-2}z,\ (p-1)(x+z)^{p-2}z\} && \text{(Using Rolle's theorem)} \\
&\leq p\max\{t^{p-2}z,\ p(2x)^{p-2}z,\ p(2z)^{p-2}z\} \\
&\leq p^2 2^{p-2}\max\{t,x,\Delta\}^{p-2}z && \text{(Since } z \leq \Delta\text{)}
\end{aligned}$$

Integrating over $z \in [0,\Delta]$, we get,

$$\gamma_p(x+\Delta) - \gamma_p(x) - \Delta\gamma_p'(x) \leq p^2 2^{p-3}\max\{t, |x|, |\Delta|\}^{p-2}\Delta^2.$$

□

A.2 Proofs from Section 4

Lemma 4.5. *Let $p \in (1, \infty)$. Then for any $\boldsymbol{x}$ and any Δ,*

$$|x|^p + g\Delta + \frac{p-1}{p2^p}\gamma_p(|x|, \Delta) \leq |x + \Delta|^p \leq |x|^p + g\Delta + 2^p\gamma_p(|x|, \Delta),$$

where $g = p\,|x|^{p-2}\,x$ is the derivative of the function $|x|^p$.

Proof. We first show the following two lemmas.

Lemma A.1. *For $|\alpha| \leq 1$ and $p \geq 1$,*

$$1 + \alpha p + \frac{(p-1)}{4}\alpha^2 \leq (1+\alpha)^p \leq 1 + \alpha p + p2^{p-1}\alpha^2.$$

Proof. Let us first show the left inequality, i.e. $1 + \alpha p + \frac{p-1}{4}\alpha^2 \leq (1+\alpha)^p$. Define the following function,

$$h(\alpha) = (1+\alpha)^p - 1 - \alpha p - \frac{p-1}{4}\alpha^2.$$

When $\alpha = 1, -1$, $h(\alpha) \geq 0$. The derivative of h with respect to α is, $h'(\alpha) = p(1+\alpha)^{p-1} - p - \frac{(p-1)}{2}\alpha$. When $p \geq 2$ and $-1 < \alpha < 1$,

$$\begin{aligned}
&\left((1+\alpha)^{p-2} - 1\right) sign(\alpha) \geq 0 \\
&\Rightarrow \left((1+\alpha)^{p-1} - (1+\alpha)\right) sign(\alpha) \geq 0 \\
&\Rightarrow \left(p(1+\alpha)^{p-1} - p - p\alpha\right) sign(\alpha) \geq 0 \\
&\Rightarrow \left(p(1+\alpha)^{p-1} - p - \frac{(p-1)}{2}\alpha\right) sign(\alpha) \geq 0
\end{aligned}$$

For the last inequality, note that when the product is positive, either both terms are positive or both terms are negative. When both terms are positive, subtracting $(p-1)/2$ instead of p gives a larger positive quantity. When both terms are negative then subtracting $(p-1)/2$ instead of p gives only a smaller quantity, so the inequality holds. This shows that $h'(\alpha)sign(\alpha) \geq 0$, which means minimum of h is at $h(0) = 0$. Next let us see what happens when $p \leq 2$ and $|\alpha| < 1$.

$$h''(\alpha) = p(p-1)(1+\alpha)^{p-2} - \frac{p-1}{2} = (p-1)\left(\frac{p}{(1+\alpha)^{2-p}} - \frac{1}{2}\right) \geq 0$$

This implies that $h'(\alpha)$ is an increasing function of α and α_0 for which $h'(\alpha_0) = 0$ is where h attains its minimum value. The only point where h' is 0 is $\alpha_0 = 0$. This implies $h(\alpha) \geq h(0) = 0$. This concludes the proof of the left inequality. For the right inequality, define:

$$s(\alpha) = 1 + \alpha p + p2^{p-1}\alpha^2 - (1+\alpha)^p.$$

Note that $s(0) = 0$ and $s(1), s(-1) \geq 0$. We have,

$$s'(\alpha) = p + p2^p\alpha - p(1+\alpha)^{p-1}$$

Using the mean value theorem for $p \geq 2$ and $\alpha < 0$,

$$\begin{aligned}(1+\alpha)^{p-1} - 1 &= (p-1)\alpha \cdot (1+z)^{p-2}, z \in (\alpha, 0) \\ &\geq \alpha 2^p.\end{aligned}$$

This implies that $s'(\alpha) \leq 0$ for negative alpha. When $1 > \alpha > 0$, using the convexity of $f(x) = (1+x)^{p-1}$ for $p > 2$, we get,

$$f(\alpha \cdot 1 + (1-\alpha) \cdot 0) \leq \alpha f(1) + (1-\alpha) f(0)$$

which gives us

$$(1+\alpha)^{p-1} \leq \alpha 2^{p-1} + 1.$$

This implies, $s'(\alpha) \geq 0$ for positive α. The function s is thus increasing for positive α and decreasing for negative α, so it attains the minimum at 0 which is $s(0) = 0$ giving us $s(\alpha) \geq 0$. We now look at the case $p \leq 2$. We have

$$(1+\alpha)^{p-1} sign(\alpha) \leq (1+\alpha) sign(\alpha).$$

Using this, we get, $s'(\alpha) sign(\alpha) \geq p|\alpha|(2^p - 1) \geq 0$ which says $s'(\alpha)$ is positive for α positive and negative for α negative. Thus the minima of s is at 0 which is 0. So $s(\alpha) \geq 0$ in this range too. □

Lemma A.2. *For $\beta \geq 1$ and $p \geq 1$, $(\beta - 1)^{p-1} + 1 \geq \frac{1}{2^p}\beta^{p-1}$.*

Proof. $(\beta - 1) \geq \frac{\beta}{2}$ for $\beta \geq 2$. So the claim clearly holds for $\beta \geq 2$ since $(\beta-1)^{p-1} \geq \left(\frac{\beta}{2}\right)^{p-1}$. When $1 \leq \beta \leq 2$, $1 \geq \frac{\beta}{2}$, so the claim holds since, $1 \geq \left(\frac{\beta}{2}\right)^{p-1}$ □

We now prove the theorem.

Let $\Delta = \alpha x$. The term $g\Delta = p|x|^{p-1} sign(x) \cdot \alpha x = \alpha p |x|^{p-1}|x| = \alpha p |x|^p$. Let us first look at the case when $|\alpha| \leq 1$. We want to show,

$$\begin{aligned}&|x|^p + \alpha p |x|^p + c\frac{p}{2}|x|^{p-2}|\alpha x|^2 \leq |x + \alpha x|^p \leq |x|^p + \alpha p |x|^p + C\frac{p}{2}|x|^{p-2}|\alpha x|^2 \\ &\Leftrightarrow (1 + \alpha p) + c\frac{p}{2}\alpha^2 \leq (1+\alpha)^p \leq (1+\alpha p) + C\frac{p}{2}\alpha^2.\end{aligned}$$

This follows from Lemma A.1 and the facts $\frac{cp}{2} \leq \frac{p-1}{4}$ and $\frac{Cp}{2} \geq p2^{p-1}$. We next look at the case when $|\alpha| \geq 1$. Now, $\gamma^p_{|f|}(\Delta) = |\Delta|^p + (\frac{p}{2} - 1)|f|^p$. We need to show

$$|x|^p(1+\alpha p) + \frac{|x|^p(p-1)}{p2^p}(|\alpha|^p + \frac{p}{2} - 1) \leq |x|^p|1+\alpha|^p \leq |x|^p(1+\alpha p) + 2^p|x|^p(|\alpha|^p + \frac{p}{2} - 1).$$

When $|x| = 0$ it is trivially true. When $|x| \neq 0$, let

$$h(\alpha) = |1+\alpha|^p - (1+\alpha p) - \frac{(p-1)}{p2^p}(|\alpha|^p + \frac{p}{2} - 1).$$

Now, taking the derivative with respect to α we get,

$$h'(\alpha) = p\left(|1+\alpha|^{p-1}sign(\alpha) - 1 - \frac{(p-1)}{p2^p}|\alpha|^{p-1}sign(\alpha)\right).$$

When $\alpha \geq 1$ and $p \geq 2$,

$$h''(\alpha) \geq p(p-1)(1+\alpha)^{p-2} - p\frac{(p-1)}{2^p}\alpha^{p-2} \geq 0.$$

So we have $h'(\alpha) \geq h'(1) \geq 0$. When $p < 2$, we use the mean value theorem to get,

$$\begin{aligned}(1+\alpha)^{p-1} - 1 &= (p-1)\alpha(1+z)^{p-2}, z \in (0,\alpha) \\ &\geq (p-1)\alpha(2\alpha)^{p-2} \\ &\geq \frac{p-1}{2}\alpha^{p-1}\end{aligned}$$

which implies $h'(\alpha) \geq 0$ in this range as well. When $\alpha \leq -1$ it follows from Lemma A.2 that $h'(\alpha) \leq 0$. So the function h is increasing for $\alpha \geq 1$ and decreasing for $\alpha \leq -1$. The minimum value of h is $min\{h(1), h(-1)\} \geq 0$. It follows that $h(\alpha) \geq 0$ which gives us the left inequality. The other side requires proving,

$$|1+\alpha|^p \leq 1 + \alpha p + 2^p(|\alpha|^p + \frac{p}{2} - 1).$$

Define:

$$s(\alpha) = 1 + \alpha p + 2^p(|\alpha|^p + \frac{p}{2} - 1) - |1+\alpha|^p.$$

The derivative $s'(\alpha) = p + \left(p2^p|\alpha|^{p-1} - p|1+\alpha|^{p-1}\right) sign(\alpha)$ is non negative for $\alpha \geq 1$ and non positive for $\alpha \leq -1$. The minimum value taken by s is $\min\{s(1), s(-1)\}$ which is non negative. This gives us the right inequality.

□

Lemma 4.8. *For $\boldsymbol{x}^{(0)}$ as defined in Definition 4.7, $\|\boldsymbol{x}^{(0)}\|_p^p \leq m^{(p-2)/2}\mathrm{OPT}$.*

Proof. Let $\boldsymbol{x}^\star$ give the OPT. We know that, for any $\boldsymbol{x}$,

$$\|\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_2 \leq m^{(p-2)/2p}\|\boldsymbol{x}\|_p.$$

This along with the fact $\|\boldsymbol{x}^{(0)}\|_2 \leq \|\boldsymbol{x}^\star\|_2$ gives us,

$$\|\boldsymbol{x}^{(0)}\|_p^p \leq \|\boldsymbol{x}^{(0)}\|_2^p \leq m^{(p-2)/2}\|\boldsymbol{x}^\star\|_p^p = m^{(p-2)/2}\mathrm{OPT}.$$

□

A.3 Proofs from Section 5

Lemma 5.3. *Let $p \in (1,\infty)$ and assume that our current solution $\boldsymbol{x}$ is not an ε-approximate solution. Let λ be such that $\lambda^{\min\{1,p-1\}} = \frac{p-1}{p4^p}$. For some*

$$i \in \left[\log\left(\frac{\varepsilon\|\boldsymbol{x}^{(0)}\|_p^p}{m^{|p-2|/2}}\right), \log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right)\right],$$

$\alpha(\Delta^\star) \in [2^{i-1}, 2^i)$ where $\Delta^\star$ is the solution of (1).

Proof. Let $\boldsymbol{x}^\star$ denote the optimum solution of (*) and $\boldsymbol{x}^{(0)}$ be as defined in Definition 4.7. We know that for any $\boldsymbol{x}$,

$$\|\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_2 \leq m^{(p-2)/2p} \|\boldsymbol{x}\|_p .$$

This along with the fact $\|\boldsymbol{x}^{(0)}\|_2 \leq \|\boldsymbol{x}^\star\|_2$ gives us,

$$\|\boldsymbol{x}^{(0)}\|_p^p \leq \|\boldsymbol{x}^{(0)}\|_2^p \leq m^{(p-2)/2} \|\boldsymbol{x}^\star\|_p^p.$$

Now from Lemma 4.6 we have,

$$\alpha(\Delta) \leq \frac{1}{\lambda}(\|\boldsymbol{x}^{(0)}\|_p^p - \|\boldsymbol{x}^\star\|_p^p) \leq \frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}(1 - m^{-(p-2)/2}).$$

Let us assume $\alpha(\Delta) \geq \varepsilon \|\boldsymbol{x}^\star\|_p^p \geq \varepsilon m^{-(p-2)/2} \|\boldsymbol{x}^{(0)}\|_p^p$. If this is not true we already have an ε approximate solution to our problem. We thus have the following bound on α,

$$\varepsilon m^{-(p-2)/2} \|\boldsymbol{x}^{(0)}\|_p^p \leq \alpha(\Delta) \leq \frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}.$$

This gives us that,

$$2^{\log\left(\frac{\varepsilon \|\boldsymbol{x}^{(0)}\|_p^p}{m^{(p-2)/2}}\right)} \leq \alpha(\Delta) \leq 2^{\log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right)}.$$

When $p \leq 2$, following a similar proof and using,

$$\|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_p \leq m^{(2-p)/2p} \|\boldsymbol{x}\|_2 ,$$

we get,

$$2^{\log\left(\frac{\varepsilon \|\boldsymbol{x}^{(0)}\|_p^p}{m^{(2-p)/2}}\right)} \leq \alpha(\Delta) \leq 2^{\log\left(\frac{\|\boldsymbol{x}^{(0)}\|_p^p}{\lambda}\right)},$$

thus concluding the proof of the lemma. thus concluding the proof of the lemma. □

Lemma 5.4. *Let $p \in (1, \infty)$. Suppose $\alpha(\Delta^\star) \in [2^{i-1}, 2^i)$ for some i where $\Delta^\star$ is the solution of (1). The following program is feasible:*

$$\begin{aligned} \gamma_p(\boldsymbol{t}, \Delta) &\leq \frac{p}{p-1} 2^{i+p}, \\ \boldsymbol{g}^T \Delta &= 2^{i-1}, \\ \boldsymbol{A}\Delta &= 0. \end{aligned} \tag{3}$$

If $\Delta(i)$ is a β-approximate solution to program (3) for this choice of i, then, we can pick $\mu \leq 1$ such that the vector $\mu\Delta(i)$ is an $\Omega_p\left(\beta^{\frac{1}{\min\{p,2\}-1}}\right)$-approximate solution to (1).

Proof. Assume that the optimum solution to (1), $\Delta^\star$ satisfies

$$\alpha(\Delta^\star) = \boldsymbol{g}^\top \Delta^\star - \frac{p-1}{p2^p} \gamma_p\left(\boldsymbol{t}, \Delta^\star\right) \in \left[2^{i-1}, 2^i\right),$$

in addition to $\boldsymbol{A}\Delta^\star = 0$. Note that we know that the objective is strictly positive (as 0 is a feasible solution). Since $\gamma_p \geq 0$, we must have,

$$\boldsymbol{g}^T\Delta^\star \geq 2^{i-1},$$

Consider scaling $\Delta^\star$ by a factor $\lambda > 0$. Since $\Delta^\star$ is optimal, we must have

$$\frac{d}{d\lambda}\left(\lambda \boldsymbol{g}^\top\Delta^\star - \frac{p-1}{p2^p}\gamma_p(\boldsymbol{t}, \lambda\Delta^\star)\right)\Big|_{\lambda=1} = 0.$$

Now, from Lemma 3.3, we know that

$$\frac{\min\{2,p\}}{\lambda}\gamma_p(\boldsymbol{t},\lambda\Delta^\star) \leq \frac{d}{d\lambda}\gamma_p(\boldsymbol{t},\lambda\Delta^\star) \leq \frac{\max\{2,p\}}{\lambda}\gamma_p(\boldsymbol{t},\lambda\Delta^\star).$$

Thus, we get,

$$\frac{p-1}{p2^p}\min\{2,p\}\gamma_p(\boldsymbol{t},\Delta^\star) \leq \boldsymbol{g}^\top\Delta^\star \leq \frac{p-1}{p2^p}\max\{2,p\}\gamma_p(\boldsymbol{t},\Delta^\star),$$

$$\frac{p-1}{p2^p}(\min\{2,p\}-1)\gamma_p(\boldsymbol{t},\Delta^\star) \leq \boldsymbol{g}^\top\Delta^\star - \frac{p-1}{p2^p}\gamma_p(\boldsymbol{t},\Delta^\star) \leq \frac{p-1}{p2^p}(\max\{2,p\}-1)\gamma_p(\boldsymbol{t},\Delta^\star).$$

Thus, $\gamma_p(\boldsymbol{t},\Delta^\star) \leq \frac{1}{\frac{p-1}{p2^p}\left(\min\{2,p\}-1\right)}2^i$, and hence $\boldsymbol{g}^\top\Delta^\star \leq \frac{\max\{2,p\}}{\min\{2,p\}-1}2^i = \max\left\{p, 2/p-1\right\}2^i$.

Now consider the vector $\Delta = \lambda\Delta^\star$, where $\lambda = \frac{2^{i-1}}{\boldsymbol{g}^\top\Delta^\star}$. Note that $\lambda \in \left[\min\left\{1/2p, (p-1)/4\right\}, 1\right]$. We have

$$\boldsymbol{g}^\top\Delta = \boldsymbol{g}^\top\left(\frac{2^{i-1}}{\boldsymbol{g}^\top\Delta^\star}\Delta^\star\right) = 2^{i-1}$$

$$\gamma_p(\boldsymbol{t},\Delta) \leq \gamma_p(\boldsymbol{t},\lambda\Delta^\star) \leq \max\{\lambda^2,\lambda^p\}\gamma_p(\boldsymbol{t},\Delta^\star) \leq \frac{p2^p}{p-1}2^i.$$

Thus, Δ is a feasible solution to Program (3). A β-approximate solution $\Delta(i)$ must be such that,

$$\begin{aligned}\boldsymbol{A}\Delta(i) &= 0,\\ \boldsymbol{g}^T\Delta(i) &= 2^{i-1},\\ \gamma_p\left(\boldsymbol{t},\Delta(i)\right) &\leq \beta\frac{p2^p}{p-1}2^i.\end{aligned}$$

Now, we consider $\Delta = \mu\Delta(i)$ for some $\mu \leq 1$. We have, $\boldsymbol{A}\Delta = 0$, and,

$$\begin{aligned}\boldsymbol{g}^\top\Delta - \frac{p-1}{p2^p}\gamma_p\left(\boldsymbol{t},\Delta\right) &= \mu\boldsymbol{g}^\top\Delta(i) - \frac{p-1}{p2^p}\gamma_p\left(\boldsymbol{t},\mu\Delta(i)\right)\\ &\geq \mu 2^{i-1} - \max\{\mu^2,\mu^p\}\beta 2^i \qquad \text{(Using Lemma 3.3)}\end{aligned}$$

We can pick,

$$\mu \leq \begin{cases}\left(\frac{1}{2\beta p}\right)^{\frac{1}{p-1}} & \text{if } p \leq 2\\ \frac{1}{4\beta} & \text{if } p \geq 2.\end{cases}$$

In either case, we get,

$$\boldsymbol{g}^\top \Delta - \frac{p-1}{p2^p}\gamma_p(\boldsymbol{t},\Delta) \geq \mu\left(1 - \frac{1}{\min\{2,p\}}\right)2^{i-1}$$

Since we assumed that the optimum of Program (1) is at most 2^i, this implies that $\mu\Delta(i)$ achieves an objective value for Program (1) that is within an $\Omega_p\left(\beta^{-\frac{1}{\min\{p,2\}-1}}\right)$ fraction of the optimal. □

Lemma 5.5. *Let $p \in (1,\infty)$. Let i be such that (3) is feasible. Let*

$$\hat{\boldsymbol{t}}_j = \begin{cases} m^{-1/p} & \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1}\boldsymbol{t}_j \leq m^{-1/p}, \\ 1 & \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1}\boldsymbol{t}_j \geq 1, \\ \left(\frac{p-1}{p}\right)^{1/p} 2^{-i/p-1}\boldsymbol{t}_j & \textit{otherwise.} \end{cases}$$

Note that $m^{-1/p} \leq \hat{\boldsymbol{t}}_j \leq 1$. Then program (2) with $\boldsymbol{t} = \hat{\boldsymbol{t}}$, and

$$\boldsymbol{c} = \left(\frac{2}{p}\right)^{1/2}\left(\frac{p-1}{p}\right)^{1/p} 2^{i\left(1-\frac{1}{p}\right)-2},$$

has OPT ≤ 1. *Let $\Delta^\star$ be a κ-approximate solution to (2). Then, $\Delta = \left(\frac{p}{2}\right)^{1/2}\left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p}\Delta^\star$ is a $\Omega_p(\kappa)$- approximate solution to (3).*

Proof. We choose i such that (3) is feasible, i.e., there exists Δ such that,

$$\begin{aligned} \gamma_p\left(\boldsymbol{t},\Delta\right) &\leq \frac{p2^p}{p-1}2^i, \\ \boldsymbol{g}^T\Delta &= 2^{i-1}, \\ \boldsymbol{A}\Delta &= 0. \end{aligned}$$

Scaling both $\boldsymbol{t}$ and Δ to $\tilde{\boldsymbol{t}} = \left(\frac{p-1}{p}\right)^{1/p} 2^{-1-i/p}\boldsymbol{t}$ and $\tilde{\Delta} = \left(\frac{p-1}{p}\right)^{1/p} 2^{-1-i/p}\Delta$ gives us the following.

$$\begin{aligned} \gamma_p\left(\tilde{\boldsymbol{t}},\tilde{\Delta}\right) &\leq 1, \\ \boldsymbol{g}^T\tilde{\Delta} &= \left(\frac{p-1}{p}\right)^{1/p} 2^{i(1-1/p)-2}, \\ \boldsymbol{A}\tilde{\Delta} &= 0. \end{aligned}$$

Now, let $\boldsymbol{t}' = \max\{m^{-1/p}, \tilde{\boldsymbol{t}_e}\}$. We claim that when $p \geq 2$, $\gamma_p\left(\boldsymbol{t}',\tilde{\Delta}\right) - \gamma_p\left(\tilde{\boldsymbol{t}},\tilde{\Delta}\right) \leq \frac{p}{2} - 1$. To see this, for a single j, let us look at the difference $\gamma_p\left(\boldsymbol{t}'_j,\tilde{\Delta}_j\right) - \gamma_p\left(\tilde{\boldsymbol{t}}_j,\tilde{\Delta}_j\right)$. If $\tilde{\boldsymbol{t}}_j \geq m^{-1/p}$ the difference is 0. Otherwise from the proof of Lemma 5 of [Bub+18],

$$\gamma_p\left(\boldsymbol{t}'_j,\tilde{\Delta}_j\right) - \gamma_p\left(\tilde{\boldsymbol{t}}_j,\tilde{\Delta}_j\right) \leq \gamma_p\left(\boldsymbol{t}'_j,\tilde{\Delta}_j\right) - \left|\tilde{\Delta}_j\right|^p \leq \left(\frac{p}{2}-1\right)\left(m^{-1/p}\right)^p.$$

When $p \leq 2$, we claim that $\gamma_p\left(\tilde{\boldsymbol{t}}, \tilde{\Delta}\right) - \gamma_p\left(\boldsymbol{t}', \tilde{\Delta}\right) \leq 1 - \frac{p}{2}$. Again if $\tilde{\boldsymbol{t}}_j \geq m^{-1/p}$ the difference is 0. Otherwise,

$$\gamma_p\left(\tilde{\boldsymbol{t}}_j, \tilde{\Delta}_j\right) - \gamma_p\left(\boldsymbol{t}'_j, \tilde{\Delta}_j\right) \leq \left|\tilde{\Delta}_j\right|^p - \gamma_p\left(\boldsymbol{t}'_j, \tilde{\Delta}_j\right) \leq \left(1 - \frac{p}{2}\right)\left(m^{-1/p}\right)^p.$$

To see the last inequality, when $\left|\Delta_j\right| \leq \boldsymbol{t}'_j$, we require, $\left|\Delta_j\right|^p - \frac{p}{2}\boldsymbol{t}_j^{p-2}\Delta_j^2 \leq \left(1 - \frac{p}{2}\right)\boldsymbol{t}_j^p$ which is true. When $\left|\Delta_j\right| \geq \boldsymbol{t}_j$, it directly follows. Summing over all j gives us our claims. We know that $\gamma_p\left(\tilde{\boldsymbol{t}}, \tilde{\Delta}\right) \leq 1$. Thus, $\gamma_p\left(\boldsymbol{t}', \tilde{\Delta}\right) \leq \frac{p}{2}$. Next we set $\hat{\Delta} = \left(\frac{2}{p}\right)^{1/2}\tilde{\Delta}$. Note that $\max\{\left(\frac{2}{p}\right)^2, \left(\frac{2}{p}\right)^p\} = \left(\frac{2}{p}\right)^2$ for all p. Lemma 3.3 thus implies,

$$\gamma_p\left(\boldsymbol{t}', \hat{\Delta}\right) \leq \left(\frac{2}{p}\right)\gamma_p\left(\boldsymbol{t}', \tilde{\Delta}\right) \leq 1.$$

Define $\hat{t}_j = \min\{1, \boldsymbol{t}'_j\}$. Note that $\gamma_p\left(\hat{\boldsymbol{t}}, \hat{\Delta}\right) = \gamma_p\left(\boldsymbol{t}', \hat{\Delta}\right)$ since $\gamma_p\left(\boldsymbol{t}', \hat{\Delta}\right) \leq 1$ and as a result we have $\gamma_p\left(\hat{\boldsymbol{t}}, \hat{\Delta}\right) \leq 1$. Observe that $\hat{\Delta}$ is a feasible solution of (2) thus suggesting that for problem (2) OPT ≤ 1. Let $\Delta^\star$ be a κ - approximate solution to (2), i.e.,

$$\gamma_p\left(\hat{\boldsymbol{t}}, \Delta^\star\right) \leq \kappa \cdot \text{OPT} \leq \kappa.$$

When $p \geq 2$, γ_p is an increasing function of $\boldsymbol{t}$ giving us,

$$\gamma_p\left(\tilde{\boldsymbol{t}}, \Delta^\star\right) \leq \gamma_p\left(\boldsymbol{t}', \Delta^\star\right) = \gamma_p\left(\hat{\boldsymbol{t}}, \Delta^\star\right) \leq \kappa.$$

When $p \leq 2$,

$$\gamma_p\left(\tilde{\boldsymbol{t}}, \Delta^\star\right) \leq \gamma_p\left(\boldsymbol{t}', \Delta^\star\right) + 1 - \frac{p}{2} \leq \kappa + 1$$

This gives,

$$\gamma_p\left(\boldsymbol{t}, \left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p}\Delta^\star\right) \leq \frac{p2^p}{p-1}2^i(\kappa+1)$$

and Lemma 3.3 then implies,

$$\gamma_p\left(\boldsymbol{t}, \left(\frac{p}{2}\right)^{1/2}\left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p}\Delta^\star\right) \leq \left(\frac{p}{2}\right)^{p/2}\frac{p2^p}{p-1}2^i(\kappa+1).$$

Finally, $\Delta = \left(\frac{p}{2}\right)^{1/2}\left(\frac{p}{p-1}\right)^{1/p} 2^{1+i/p}\Delta^\star$ satisfies the constraints of (3) and is a $\Omega_p(\kappa)$ approximate solution. □

Lemma 5.6. *Let $p \geq 2$. For any set of weights $\boldsymbol{w}$ on the edges, $\sum_e \boldsymbol{w}_e^{p-2}(\Delta_e^*)^2 \leq \|\boldsymbol{w}\|_p^{p-2}$.*

Proof. Using Hölder's inequality, we have,

$$\sum_e \boldsymbol{w}_e^{p-2}(\Delta_e^*)^2 \leq \left(\sum_e ((\Delta_e^*)^2)^{p/2}\right)^{2/p}\left(\sum_e (|\boldsymbol{w}_e|^{p-2})^{p/(p-2)}\right)^{(p-2)/p}$$

$$= \left(\sum_e \left|\Delta_e^*\right|^p\right)^{2/p} \left(\sum_e |\boldsymbol{w}_e|^p\right)^{(p-2)/p}$$
$$\leq \left(\sum_e |\boldsymbol{w}_e|^p\right)^{\frac{(p-2)}{p}}, \text{ since } \sum_e (\Delta_e^*)^p \leq 1.$$

□

A.4 Proofs from Section 6

Lemma 6.7. *Consider a flow step from Line 13 of Algorithm 4. We have*

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \left(1 + \alpha \left|\Delta_e\right|\right)^{p-2} - 1$$

where Δ is the ℓ_2 minimizer solution produced by the oracle.

Proof. Recall from the setting of resistances from Line 2 of ORACLE (Algorithm 3) that

$$\boldsymbol{r}_e^{(i)} = \left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}.$$

By Line 13 of Algorithm 4, we have

$$\boldsymbol{w}_e^{(i+1)} - \boldsymbol{w}_e^{(i)} = \alpha \left|\Delta_e\right|.$$

Substituting this in gives

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} = \frac{\left(\boldsymbol{w}_e^{(i)} + \alpha \left|\Delta_e\right|\right)^{p-2} - \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}}{\left(m^{1/p}\boldsymbol{t}_e\right)^{p-2} + \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}}.$$

There are two cases to consider:

1. $\boldsymbol{w}_e^{(i)} \geq m^{1/p}\boldsymbol{t}_e$.

   $$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \frac{\left(\boldsymbol{w}_e^{(i)} + \alpha \left|\Delta_e\right|\right)^{p-2} - \left(\boldsymbol{w}_e^{(i)}\right)^{p-2}}{\left(\boldsymbol{w}_e^{(i)}\right)^{p-2}} \leq \left(1 + \frac{\alpha \left|\Delta_e\right|}{\boldsymbol{w}_e^{(i)}}\right)^{p-2} - 1 \leq \left(1 + \alpha \left|\Delta_e\right|\right)^{p-2} - 1$$

   where the last inequality utilizes $\boldsymbol{w}_e^{(i)} \geq 1$, which is due to the assumption and $m^{1/p}\boldsymbol{t}_e \geq 1$.

2. $\boldsymbol{w}_e^{(i)} \leq m^{1/p}\boldsymbol{t}_e$, then replacing the denominator with the $(m^{1/p}\boldsymbol{t}_e)^{p-2}$ term and simplifying gives

   $$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \left(\frac{\boldsymbol{w}_e^{(i)}}{m^{1/p}\boldsymbol{t}_e} + \frac{\alpha \left|\Delta_e\right|}{m^{1/p}\boldsymbol{t}_e}\right)^{p-2} - \left(\frac{\boldsymbol{w}_e^{(i)}}{m^{1/p}\boldsymbol{t}_e}\right)^{p-2}.$$

As the function $(z+\theta)^{p-2} - z^{p-2}$ is monotonically increasing when $\theta, p-2 \geq 0$, we may replace the $\frac{\boldsymbol{w}_e^{(i)}}{m^{1/p}\boldsymbol{t}_e}$ by its upper of 1 (given by the assumption) to get

$$\frac{\boldsymbol{r}_e^{(i+1)} - \boldsymbol{r}_e^{(i)}}{\boldsymbol{r}_e^{(i)}} \leq \left(1 + \frac{\alpha\left|\Delta_e\right|}{m^{1/p}\boldsymbol{t}_e}\right)^{p-2} - 1 \leq \left(1 + \alpha\left|\Delta_e\right|\right)^{p-2} - 1,$$

where the last inequality follows from $m^{1/p}\boldsymbol{t}_e \geq 1$.

□

B Controlling Φ

Lemma 5.10. *After i* flow *steps, and k width-reduction steps, provided*

1. $\alpha^p \tau \leq \alpha m^{\frac{p-1}{p}}$, *(controls Φ growth in flow-steps)*
2. $k \leq \rho^2 m^{2/p} \beta^{-\frac{2}{p-2}}$, *(acceptable number of width-reduction steps)*

the potential Φ is bounded as follows:

$$\Phi(i,k) \leq \left(p^2 2^p \alpha i + m^{1/p}\right)^p \exp\left(O_p(1)\frac{k}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right).$$

Proof. We prove this claim by induction. Initially, $i = k = 0$, and $\Phi(0,0) = 0$, and thus, the claim holds trivially. Assume that the claim holds for some $i, k \geq 0$. We will use Φ as an abbreviated notation for $\Phi(i,k)$ below.

Flow Step. For brevity, we let $\gamma_p(\boldsymbol{w})$ denote $\gamma_p(m^{1/p}\boldsymbol{t}, \boldsymbol{w})$, and use $\boldsymbol{w}$ to denote $\boldsymbol{w}^{(i,k)}$.

If the next step is a *flow* step,

$$\begin{aligned}
\Phi(i+1,k) =& \gamma_p\left(\boldsymbol{w}^{(i,k)} + \alpha\left|\Delta\right|\right) \\
\leq& \gamma_p(\boldsymbol{w}) + \alpha\left|\Delta^\top \gamma'(\boldsymbol{w})\right| + p^2 2^{p-3}\alpha^2 \sum_e \max\{m^{1/p}\boldsymbol{t}, \left|\boldsymbol{w}_e\right|, \alpha\Delta_e\}^{p-2}\Delta_e^2, \quad \text{by Lemma 3.4} \\
\leq& \gamma_p(\boldsymbol{w}) + p\alpha\gamma_p(\boldsymbol{w})^{\frac{p-1}{p}} + p\alpha m^{\frac{p-2}{2p}}\gamma_p(\boldsymbol{w})^{1/2} \\
&+ p^2 2^{p-3}\alpha^2 \sum_e \left(\boldsymbol{r}_e\Delta_e^2 + \alpha^{p-2}\Delta_e^p\right), \quad \text{by Lemma 5.7} \\
\leq& \gamma_p(\boldsymbol{w}) + p\alpha\gamma_p(\boldsymbol{w})^{\frac{p-1}{p}} + p\alpha m^{\frac{p-2}{2p}}\gamma_p(\boldsymbol{w})^{1/2} \\
&+ p^2 2^{p-3}\left(\alpha^2 m^{\frac{p-2}{p}} + \alpha^2\left\|\boldsymbol{w}\right\|_p^{p-2} + \sum_e \alpha^p \Delta_e^p\right), \quad \text{by Lemma 5.7} \\
\leq& \gamma_p(\boldsymbol{w}) + p\alpha\gamma_p(\boldsymbol{w})^{\frac{p-1}{p}} + p\alpha m^{\frac{p-2}{2p}}\gamma_p(\boldsymbol{w})^{1/2} \\
&+ p^2 2^{p-3}\left(\alpha^2 m^{\frac{p-2}{p}} + \alpha^2\left\|\boldsymbol{w}\right\|_p^{p-2} + \alpha m^{\frac{p-1}{p}}\right), \quad \text{by Assumption 1 of this Lemma}
\end{aligned}$$

Using $\|\boldsymbol{w}\|_p \le \gamma_p(\boldsymbol{w})$

$$\begin{aligned}
&\le \gamma_p(\boldsymbol{w}) + p\alpha\gamma_p(\boldsymbol{w})^{\frac{p-1}{p}} + p\alpha m^{\frac{p-2}{2p}}\gamma_p(\boldsymbol{w})^{1/2} \\
&\quad + p^2 2^{p-3}\left(\alpha^2 m^{\frac{p-2}{p}} + \alpha^2\gamma_p(\boldsymbol{w})^{\frac{p-2}{p}} + \alpha m^{\frac{p-1}{p}}\right)
\end{aligned}$$

Recall $\gamma_p(\boldsymbol{w}) = \Phi(\boldsymbol{w})$. Letting z denote $\max\{\Phi(\boldsymbol{w}), m\}^{1/p}$, we have,

$$\begin{aligned}
&\le z^p + p\alpha z^{(p-1)} + p\alpha z^{(p-1)} \\
&\quad + p^2 2^{p-3}\left(\alpha^2 z^{(p-2)} + \alpha^2 z^{p-2} + \alpha z^{(p-1)}\right) \\
&\le z^p + p^2 2^{p-2}\alpha z^{(p-1)} \\
&\quad + p^2 2^{p-2}\alpha^2 z^{(p-2)} \\
&\le (z + p^2 2^p \alpha)^p.
\end{aligned}$$

From the inductive assumption, we have

$$\begin{aligned}
z = \max\{\Phi, m\}^{1/p} &\le \max\left\{\left(p^2 2^p \alpha i + m^{1/p}\right)^p \exp\left(O_p(1)\frac{k}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right), m\right\}^{1/p} \\
&= \left(p^2 2^p \alpha k_1 + m^{1/p}\right)\left(\exp\left(O_p(1)\frac{k}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right)\right)^{1/p}.
\end{aligned}$$

Thus,

$$\Phi(i+1,k) \le (z + p^2 2^p \alpha)^p \le \left(p^2 2^p \alpha(i+1) + m^{1/p}\right)^p \exp\left(O_p(1)\frac{k}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right)$$

proving the inductive claim.

Width Reduction Step. To analyze a width-reduction step, we first observe that, by Lemma 5.7 and the induction hypothesis, which ensures $\left\|\boldsymbol{w}^{(i,k)}\right\|_p^p \le \Phi \le O_p(1)m$, and hence $\sum_e \boldsymbol{r}_e f_e^2 \le O_p(1)m^{(p-2)/p}$ so we have

$$\sum_{e\in H} \boldsymbol{r}_e \le \rho^{-2}\sum_{e\in H} \boldsymbol{r}_e f_e^2 \le \rho^{-2}\sum_{e} \boldsymbol{r}_e f_e^2 \le \rho^{-2} O_p(1) m^{(p-2)/p}.$$

Thus, when the next step is a width-reduction step, we have,

$$\begin{aligned}
\Phi(i,k+1) &\le \Phi + O_p(1)\sum_{e\in H} \boldsymbol{r}_e^{\frac{p}{p-2}} \\
&\le \Phi + O_p(1)\left(\sum_{e\in H} \boldsymbol{r}_e\right)\left(\max_{e\in H} \boldsymbol{r}_e\right)^{\frac{p}{p-2}-1}
\end{aligned}$$

$$\leq \Phi + O_p(1)\left(\rho^{-2}m^{\frac{p-2}{p}}\right)\beta^{\frac{2}{p-2}}$$

Letting z denote $\max\{\Phi(\boldsymbol{w}), m\}^{1/p}$, we have,

$$\leq z\left(1 + O_p(1)\left(\rho^{-2}m^{-\frac{2}{p}}\right)\beta^{\frac{2}{p-2}}\right).$$

Thus,

$$\Phi(i, k+1) \leq z\left(1 + O_p(1)\left(\rho^{-2}m^{-\frac{2}{p}}\right)\beta^{\frac{2}{p-2}}\right) \leq \left(p^2 2^p \alpha i + m^{1/p}\right)^p \exp\left(O_p(1)\frac{k+1}{\rho^2 m^{2/p}\beta^{-\frac{2}{p-2}}}\right)$$

proving the inductive claim.

□

C Solving L2 problems

Lemma C.1. *Given an algorithm* Solver *for solving* $\boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{x} = \boldsymbol{d}$, *for a* $m \times n$*-fixed matrix* $\boldsymbol{B}$, *a fixed positive diagonal matrix* $\boldsymbol{R} > 0$ *and an arbitrary vector* $\boldsymbol{d}$, *there is an algorithm* EnhancedSolver *that can solve*

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & \frac{1}{2}\boldsymbol{f}^\top\boldsymbol{R}\boldsymbol{f} \\ s.t. \quad & \boldsymbol{B}^\top\boldsymbol{f} = 0 \\ & \boldsymbol{g}^\top\boldsymbol{f} = z \end{aligned} \tag{8}$$

with one call to Solver, *two multiplications of* $\boldsymbol{B}$ *with a vector, and an additional* $O(m+n)$ *time, if we assume*

$$\boldsymbol{g}^\top\boldsymbol{R}^{-1}\boldsymbol{B}\left(\boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{g} < \boldsymbol{g}^\top\boldsymbol{R}^{-1}\boldsymbol{g}.$$

Proof. Introducing the Lagrangian multipliers $\boldsymbol{v}, a$ respectively for the constraint $\boldsymbol{B}^\top\boldsymbol{f} = 0$, and $\boldsymbol{g}^\top\boldsymbol{f} = z$, we can write the Lagrangian as

$$\frac{1}{2}\boldsymbol{f}^\top\boldsymbol{R}\boldsymbol{f} - \boldsymbol{v}^\top\boldsymbol{B}^\top\boldsymbol{f} - a\left(\boldsymbol{g}^\top\boldsymbol{f} - z\right).$$

Now, optimizing the Lagrangian with respect to an unconstrained $\boldsymbol{f}$, allows us to write

$$\boldsymbol{f} = \boldsymbol{R}^{-1}\left(\boldsymbol{B}\boldsymbol{v} + a\boldsymbol{g}\right).$$

Plugging this back, we can simplify our Lagrangian as

$$-\frac{1}{2}\left(\boldsymbol{B}\boldsymbol{v} + a\boldsymbol{g}\right)^\top\boldsymbol{R}^{-1}\left(\boldsymbol{B}\boldsymbol{v} + a\boldsymbol{g}\right) + az.$$

Optimizing with respect to a, gives us,

$$a = \frac{z - \boldsymbol{g}^\top\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{v}}{\boldsymbol{g}^\top\boldsymbol{R}^{-1}\boldsymbol{g}}.$$

Plugging this back, gives the Lagrangian as

$$-\frac{1}{2}\boldsymbol{v}^\top \boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{v} + \frac{\left(z - \boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{v}\right)^2}{2\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g}}.$$

We let $\widetilde{\boldsymbol{g}}$ denote the vector $\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{g}$ and $\boldsymbol{M}$ denote the matrix $\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B}$. Thus, the Lagrangian can be written as,

$$-\frac{1}{2}\boldsymbol{v}^\top \boldsymbol{M}\boldsymbol{v} + \frac{\left(z - \widetilde{\boldsymbol{g}}^\top \boldsymbol{v}\right)^2}{2\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g}}.$$

This implies that the optimal $\boldsymbol{v}$ is given by the equation

$$\left(\boldsymbol{M} - \frac{1}{\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g}}\widetilde{\boldsymbol{g}}\widetilde{\boldsymbol{g}}^\top\right)\boldsymbol{v} = -\frac{z\widetilde{\boldsymbol{g}}}{\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g}}.$$

From the condition assumed on $\boldsymbol{g}$, we have $\widetilde{\boldsymbol{g}}\boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}} < \boldsymbol{g}\boldsymbol{R}^{-1}\boldsymbol{g}$. Thus, we can solve this system using the Sherman-Morrisson formula as follows,

$$\begin{aligned}\boldsymbol{v} &= -\left(\boldsymbol{M}^{-1} + \frac{\boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}}\widetilde{\boldsymbol{g}}^\top \boldsymbol{M}^{-1}}{\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g} - \widetilde{\boldsymbol{g}}^\top \boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}}}\right)\frac{z\widetilde{\boldsymbol{g}}}{\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g}} \\ &= -\frac{z\boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}}}{\boldsymbol{g}^\top \boldsymbol{R}^{-1}\boldsymbol{g} - \widetilde{\boldsymbol{g}}^\top \boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}}}\end{aligned}$$

The algorithm EnhancedSolver computes $\widetilde{\boldsymbol{g}}$, and then invokes Solver to compute $\boldsymbol{M}^{-1}\widetilde{\boldsymbol{g}}$. This allows us to compute $\boldsymbol{v}$ is an additional $O(m+n)$ time. Finally, we can compute $\boldsymbol{f} = \boldsymbol{R}^{-1}(\boldsymbol{B}\boldsymbol{v} + a\boldsymbol{g})$ using another multiplication with $\boldsymbol{B}$ and an additional $O(m+n)$ time. □

D General ℓ_2 Resistance Monotonicity

Lemma 5.11. *Assuming the program* (4) *is feasible, let* Δ *be an be a solution to the optimization problem* (4) *with weights* $\boldsymbol{r}$*. Suppose we increase the resistance on each entry to get* $\boldsymbol{r}'$ *Then,*

$$\Psi\left(\boldsymbol{r}'\right) \geq \exp\left(\frac{\sum_e \min\left\{1, \frac{r'_e - r_e}{r_e}\right\} \boldsymbol{r}_e \Delta_e^2}{2\Psi\left(\boldsymbol{r}\right)}\right)\Psi\left(\boldsymbol{r}\right).$$

Proof. Recall

$$\begin{aligned}\Psi\left(\boldsymbol{r}\right) = \min_{\Delta} \quad & \sum_e \boldsymbol{r}_e \Delta_e^2 \\ \text{s.t.} \quad & \boldsymbol{A}'\Delta = \boldsymbol{c}\end{aligned}$$

Letting $\boldsymbol{R}$ denote the diagonal matrix with $\boldsymbol{r}$ on its diagonal, we can write the above as

$$\Psi\left(\boldsymbol{r}\right) = \min_{\Delta} \quad \Delta^\top \boldsymbol{R}\Delta \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{A}'\Delta = \boldsymbol{c}$$

Using Lagrangian duality, and noting that strong duality holds, we can write this as

$$\begin{aligned}\Psi(\boldsymbol{r}) &= \min_{\Delta}\max_{\boldsymbol{y}} \quad \Delta^\top \boldsymbol{R}\Delta + 2\boldsymbol{y}^\top(\boldsymbol{c} - \boldsymbol{A}'\Delta) \\ &= \max_{\boldsymbol{y}}\min_{\Delta} \quad 2\boldsymbol{y}^\top\boldsymbol{c} + \Delta^\top\boldsymbol{R}\Delta - 2\boldsymbol{y}^\top\boldsymbol{A}'\Delta\end{aligned}$$

The minimizing Δ can be found by setting the gradient w.r.t. to this variable to zero. This gives $2\boldsymbol{R}\Delta - 2\boldsymbol{y}^\top\boldsymbol{A}' = 0$, so that $\Delta = \boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y}$. Plugging in this choice of Δ, we arrive at the dual program

$$\Psi(\boldsymbol{r}) = \max_{\boldsymbol{y}} \quad 2\boldsymbol{c}^\top\boldsymbol{y} - \boldsymbol{y}^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y} \tag{10}$$

Crucially, strong duality also implies that if Δ^* is an optimal solution of the primal program (9), and $\boldsymbol{y}^*$ is an optimal solution to the dual then

$$\min_{\Delta} \quad 2(\boldsymbol{y}^*)^\top\boldsymbol{c} + \Delta^\top\boldsymbol{R}\Delta - 2(\boldsymbol{y}^*)^\top\boldsymbol{A}'\Delta$$

is optimized at $\Delta = \Delta^*$. This in turn implies the gradient w.r.t. Δ at $\Delta = \Delta^*$ is zero, so that $\Delta^* = \boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y}^*$. Let $\boldsymbol{a}_e$ be the eth row of $\boldsymbol{A}'$. Then the previous equation tells us that $\Delta^*_e = \frac{1}{\boldsymbol{r}_e}\boldsymbol{a}_e^\top\boldsymbol{y}^*$. This implies that

$$\forall e.\boldsymbol{r}_e(\Delta^*_e)^2 = \frac{1}{\boldsymbol{r}_e}(\boldsymbol{a}_e^\top\boldsymbol{y}^*)^2. \tag{11}$$

Consider another program, essentially the same as (10), but with additional scalar valued variable $\theta \in \mathbb{R}$ introduced.

$$\Psi(\boldsymbol{r}) = \max_{\boldsymbol{z},\theta} \quad \theta \cdot 2\boldsymbol{c}^\top\boldsymbol{z} - \theta^2 \cdot \boldsymbol{z}^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{z} \tag{12}$$

The two programs (10) and (12) have the same value, since for any $\boldsymbol{y}$, the assignment $(\boldsymbol{z},\theta) = (\boldsymbol{y},1)$ ensures both objectives take the same value, and conversely for any $(\boldsymbol{z},\theta)$, the assignment $\boldsymbol{y} = \theta\boldsymbol{z}$ ensures both programs take the same value.

We see that $\boldsymbol{z} = \boldsymbol{y}^*$, and $\theta = 1$ is an optimal solution to (12). Hence

$$\left[\frac{d}{d\theta}\left(\theta \cdot 2\boldsymbol{c}^\top\boldsymbol{y}^* - \theta^2 \cdot (\boldsymbol{y}^*)^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y}^*\right)\right]_{\theta=1} = 0$$

Consequently, $\boldsymbol{c}^\top\boldsymbol{y}^* = (\boldsymbol{y}^*)^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y}^*$. Hence $\boldsymbol{c}^\top\boldsymbol{y}^* = \Psi(\boldsymbol{r})$. Again, by a scaling argument, this implies that

$$\begin{aligned}2 - \frac{1}{\Psi(\boldsymbol{r})} = \max_{\boldsymbol{y}} \quad & 2\boldsymbol{c}^\top\boldsymbol{y} - \boldsymbol{y}^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{c}^\top\boldsymbol{y} = 1\end{aligned}$$

So that

$$\frac{1}{\Psi(\boldsymbol{r})} = \min_{\boldsymbol{y}} \quad \boldsymbol{y}^\top\boldsymbol{A}'\boldsymbol{R}^{-1}(\boldsymbol{A}')^\top\boldsymbol{y} \tag{13}$$

$$\text{s.t.} \qquad \boldsymbol{c}^\top \boldsymbol{y} = 1$$

Note that one optimal assignment for the program (13) is $\widetilde{\boldsymbol{y}} = \frac{\boldsymbol{y}^*}{\Psi(\boldsymbol{r})}$. Also observe that if we consider the program (13) with $\boldsymbol{r}'$ instead of $\boldsymbol{r}$ as the resistances, then $\widetilde{\boldsymbol{y}} = \frac{\boldsymbol{y}^*}{\Psi(\boldsymbol{r})}$ is still a feasible solution. Hence, using the observation (11), we get

$$\frac{1}{\Psi\left(\boldsymbol{r}'\right)} \leq \frac{1}{\Psi\left(\boldsymbol{r}\right)^2} \left(\boldsymbol{y}^*\right)^\top \boldsymbol{A}' \left(\boldsymbol{R}'\right)^{-1} \left(\boldsymbol{A}'\right)^\top \boldsymbol{y}^* = \frac{1}{\Psi\left(\boldsymbol{r}\right)^2} \sum_e \frac{\boldsymbol{r}_e}{\boldsymbol{r}'_e} \boldsymbol{r}_e (\Delta_e^*)^2$$

Factoring out the $\Psi(\boldsymbol{r})$ term gives

$$\frac{1}{\Psi\left(\boldsymbol{r}'\right)} \leq \frac{1}{\Psi\left(\boldsymbol{r}\right)} \left(1 - \frac{\sum_e \left(1 - \frac{\boldsymbol{r}_e}{\boldsymbol{r}'_e}\right) \boldsymbol{r}_e (\Delta_e^*)^2}{\Psi\left(\boldsymbol{r}\right)}\right) \leq \exp\left(-\frac{\sum_e \left(1 - \frac{\boldsymbol{r}_e}{\boldsymbol{r}'_e}\right) \boldsymbol{r}_e (\Delta_e^*)^2}{\Psi\left(\boldsymbol{r}\right)}\right).$$

Now consider the term $1 - \frac{\boldsymbol{r}_e}{\boldsymbol{r}'_e}$: if $\boldsymbol{r}'_e \geq 2\boldsymbol{r}_e$, then it is at least $1/2$. Otherwise, it can be rearranged to

$$\frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}'_e} \geq \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{2\boldsymbol{r}_e}.$$

So in either case, we have

$$\frac{1}{\Psi\left(\boldsymbol{r}'\right)} \leq \frac{1}{\Psi\left(\boldsymbol{r}\right)} \exp\left(-\frac{\sum_e \min\left\{1, \frac{\boldsymbol{r}'_e - \boldsymbol{r}_e}{\boldsymbol{r}_e}\right\} \boldsymbol{r}_e (\Delta_e^*)^2}{2\Psi\left(\boldsymbol{r}\right)}\right),$$

which upon rearranging gives the desired result. □