Following the Path of Least Resistance:
An $\tilde{O}(m\sqrt{n})$ Algorithm for the Minimum Cost Flow Problem

Yin Tat Lee  
MIT  
yintat@mit.edu

Aaron Sidford  
MIT  
sidford@mit.edu

Abstract

In this paper we present an $\tilde{O}(m\sqrt{n})$ time algorithm for solving the maximum flow problem on directed graphs with $m$ edges, $n$ vertices, and capacity ratio $U$. This improves upon the previous fastest running time of $O(m \min(n^{2/3}, m^{1/2}) \log(n^2/m) \log U)$ achieved over 15 years ago by Goldberg and Rao [8] and improves upon the previous best running times for solving dense directed unit capacity graphs of $O(\min(m^{3/2}, mn^{2/3}))$ achieved by Even and Tarjan [6] over 35 years ago and a running time of $O(m^{10/7})$ achieved recently by Madry [21].

We achieve these results through the development and application of a new general interior point method that we believe is of independent interest. The number of iterations required by this algorithm is better than that predicted by analyzing the best self-concordant barrier of the feasible region. By applying this method to the linear programming formulations of maximum flow, minimum cost flow, and lossy generalized minimum cost flow and applying analysis by Daitch and Spielman[5] we achieve running time of $\tilde{O}(m\sqrt{n})$ for these problems as well. Furthermore, our algorithm is parallelizable and using a recent nearly linear time work polylogarithmic depth Laplacian system solver of Spielman and Peng [25] we achieve a $\tilde{O}(\sqrt{n})$ depth algorithm and $\tilde{O}(m\sqrt{n})$ work algorithm for solving these problems.

1 Introduction

The maximum flow problem and its dual, the minimum $s$-$t$ cut problem are two of the most well studied problems in combinatorial optimization [27]. These problems are key algorithmic primitives used extensively throughout both the theory and practice of computer science [1]. Numerous problems in algorithm design efficiently reduce to solving maximum flow [2, 28] and techniques developed in the study of this problem have had far reaching implications [3, 2].

Study of the maximum flow problem dates back to 1954 when the problem was first posed by Harris [26]. After decades of work the current fastest running time for the maximum flow problem is due to a celebrated result of Goldberg and Rao in 1998 in which they produced a $O(\min(n^{2/3}, m^{1/2}) m \log(n^2/m) \log U)$ time algorithm for weighted directed graphs with $n$ vertices, $m$ edges and integer capacities of maximum capacity $U$. While there have been

\footnote{Here and throughout the rest of the paper when we refer to running times for solving maximum flow we are referring to “weakly” polynomial time algorithms. The current fastest running time for solving this problem in “strongly” polynomial is $O(nm)$ [24].}
numerous improvements in running times for generalizations and special cases of this problem (see Section 1.1), the running time for solving the maximum flow problem in full generality has not been improved since 1998.

In this paper, we provide an algorithm that solves the maximum flow problem in time \( \tilde{O}(m\sqrt{n} \log^2(U)) \), yielding the first improvement to the running time for maximum flow in 15 years and the running time for solving dense unit capacity directed graphs in 35 years.\(^2\) Furthermore, our algorithm is easily parallelizable and using [25], we obtain a \( \tilde{O}(m\sqrt{n} \log^2(U)) \) work \( \tilde{O}(\sqrt{n} \log(U)) \) depth algorithm. Using the same technique, we also solve the minimum cost flow problem in time \( \tilde{O}(m\sqrt{n} \log^2(U)) \) time and produce \( \epsilon \)-approximate solutions to the lossy generalized minimum cost flow problem in \( \tilde{O}(m\sqrt{n} \log^2(U/\epsilon)) \) time.

We achieve these running times through a novel extension of our previous work [18]. We show how to create a primal analog of the dual weighted path following algorithm in this paper and generalize this algorithm to work for a broad class of barrier functions. Ultimately this yields a path following algorithm that achieves a convergence rate better than that of the best possible self concordant barrier for feasible region, breaking a long-standing natural barrier to the convergence rate of general interior point methods [23]. By applying this algorithm to the linear program formulation in [5], using their error analysis, and employing nearly linear time algorithms for solving Laplacian systems [30, 14, 15, 13, 17, 19, 25], we achieve the desired running times.

While our analysis is technical and our approach is general, for the specific case of the maximum flow problem our algorithm has a slightly more straightforward interpretation. The algorithm simply alternates between re-weighting costs, solving electric flow problems to send more flow, and approximately computing the effective resistance of all edges in the graph to keep the effective resistance of all edges in the graph fairly small and uniform. Hence, by following the path of (almost) least (effective) resistance, we solve the maximum flow problem in \( \tilde{O}(m\sqrt{n} \log^2(U)) \).

1.1 Previous Work

While the worst case asymptotic running time for solving the maximum in the general case has remained unchanged over the past 15 years, there have been significant breakthroughs on specific instances, generalizations, and technical machinery for solving the maximum flow problem. Here we survey some of the key results that we leverage to achieve our running times.

Although the running time for solving general directed instances has remained relatively stagnant until recently [21], there have been significant improvements in the running time for computing maximum flows on undirected graphs. A beautiful line of work on faster algorithms for approximately solving the maximum flow problem on undirected graphs began with a result of Benzeur and Karger in which they showed how to reduce approximately computing minimum cuts in arbitrary undirected graphs to the same problem on sparse graphs, i.e. those with only a nearly linear number of vertices [3]. In later work Karger also showed how reduce computing approximate maximum flow on undirected graphs to computing approximate maximum flows on undirected graphs [10]. Pushing this idea even further, in a series of results Karger and Levine ultimately showed showed how to compute the exact maximum flow in an unweighted

\(^2\)Here and in the remainder of the paper we use \( \tilde{O}(\cdot) \) to hide \( \text{polylog}(n, m) \) factors.
undirected graph in time $\tilde{O}(m + nF)$ where $F$ is the maximum flow value of the graph [9].

Then in 2004 a breakthrough result of Spielman and Teng [30] showed that a particular class of linear systems, Laplacian, can be solved in nearly linear time and Christiano, Kelner, Mądry, and Spielman [4] showed how to use these fast Laplacian system solvers to approximately solve the maximum flow problem on undirected graphs in time $\tilde{O}(mn^{1/3}\epsilon^{-11/3})$. Later Lee, Rao and Srivastava [16] showed how to solve the problem in $\tilde{O}(mn^{1/3}\epsilon^{-2/3})$ for undirected unweighted graphs. This exciting line of work culminated in breakthrough results of Sherman [28] and Kelner, Lee, Orecchia and Sidford [12] who showed how to solve the problem in time almost linear in the number of edges in the graph, $\tilde{O}(m^{1+o(1)}\epsilon^{-2})$, using congestion-approximators, oblivious routings, efficient construction techniques developed by Mądry [20].

In the exact and directed setting, there has been significant progress made on the maximum flow problem and its generalizations using interior point methods, a powerful and general technique for convex optimization [11, 23]. In 2008, Daitch and Spielman [5] showed that, by careful application of interior point techniques, fast Laplacian system solvers [30], and a novel method for solving M-matrices, they could match (up to polylogarithmic factors) the running time of Goldberg Rao and achieve a running time of $\tilde{O}(m^{3/2})$ not just for maximum flow but also for minimum cost flow and lossy generalized minimum cost flow. Furthermore, very recent Mądry [21] in 2013 achieved an astounding running time of $\tilde{O}(m^{10/7})$ for solving the maximum flow problem on un-capacitated directed graphs by a novel application and modification of interior point methods. This shattered numerous barriers providing the first general improvement over the running time of $O(\min\{m^{3/2}, mn^{2/3}\})$ for solving unit capacity graphs proven over 35 years ago by Even and Tarjan [6] in 1975.

While our algorithm for solving the maximum flow problem is new, we make extensive use of these breakthroughs on the maximum flow problem. We use sampling techniques first discovered in the context of graph sparsification [29] but not to sparsify a graph but rather to re-weight the graph so that we make progress at a rate commensurate with the number of vertices and not the number of edges. We use fast Laplacian system solvers as in [4, 16], but we use it make the cost of interior point iterations cheap as in [5, 21]. We then use reductions and error analysis in Daitch and Spielman [5] as well as their solvers for M-matrices to apply our framework to flow problems. Furthermore, as in Mądry we use weights to change the central path we are following (albeit for a slightly different purpose). We believe this further emphasizes the power of these tools as general purpose techniques for algorithm design.

1.2 The Problem

Our approach to the maximum flow problem is motivated by our previous work [18]. In that paper, we provided a new method for solving a general linear program given in the following dual of standard form

$$\min_{\bar{x} \in \mathbb{R}^n} \; \bar{c}^T \bar{x} \quad (1.1)$$

where $A \in \mathbb{R}^{m \times n}$, $\bar{b} \in \mathbb{R}^m$, and $\bar{c} \in \mathbb{R}^n$. In that paper, we showed how to solve this linear program in $\tilde{O}(\sqrt{\text{rank}(A)L})$ iterations while solving only $\tilde{O}(1)$ linear systems in each iteration.\(^3\) Whereas the convergence rate of comparable linear program solvers depended on the square

\(^3\)Here and in the remainder of the paper we use $L$ to denote the standard bit complexity of the linear program. $L$ is at most the number of bits needed to represent the problem and often smaller for flow problems (see [31]).
root of the larger of the number of variables and number of constraints, in a fairly general
regime ours depended on the smaller of these two quantities.

Unfortunately this result was insufficient to produce faster algorithms for maximum flow
and its generalizations. For an arbitrary maximum flow instance there is a natural linear
program that one can use to express the problem:

\[
\min_{\bar{x} \in \mathbb{R}^m} \quad \bar{c}^T \bar{x}
\]
\[\forall i \in [m] : \quad l_i \leq x_i \leq u_i\] (1.2)

Where here the variables \(x_i\) denote the flow on edge, the \(l_i\) and \(u_i\) denote upper and lower
bounds on how much flow we can put on the edge, and \(A\) is the incidence matrix associated
with a graph[5]. In this formulation, \(\text{rank}(A)\) is less than the number of vertices in the
graph and using fast Laplacian system solvers we can solve linear systems involving \(A\) in time
nearly linear in the number of edges in the graph. Thus, if we could do similar error analysis
as in Daitch and Spielman [5] and solve (1.2) in time comparable to that we can achieve for
solving (1.1) this would immediately yield a \(\tilde{O}(m \sqrt{n} \log^2(U))\) algorithm for the maximum flow
problem. Unfortunately, it is not clear how to apply our result in this more general setting
and naive attempts to write problem (1.2) in the form of (1.1) without increasing \(\text{rank}(A)\)
fail.

Even more troubling, achieving a faster than \(\tilde{O}(\sqrt{mL})\) iteration interior point method
for solving general linear programs in this form would break a long-standing barrier for the
convergence rate of interior point methods. In a seminal result of Nesterov and Nemirovski [23],
they provided a unifying theory for interior point methods and showed that given the ability
to construct a \(v\)-self concordant barrier for a convex set one can minimize linear functions
over that convex set with a convergence rate of \(O(\sqrt{v})\). Furthermore, they showed how to
construct such barriers for a variety of convex sets and thereby achieve fast running times.

To the best of the authors knowledge, there is no general purpose interior point method
that achieves a convergence rate faster then than the self concordance of the best barrier of
the feasible region. Furthermore, using lower bounds results of Nesterov and Nemirovski, it
is not hard to see that any general barrier for (1.2) must have self concordance \(\Omega(m)\). To be
more precise, note the following result of Nesterov and Nemirovski.

**Theorem 1.** [23, Proposition 2.3.6] Let \(\Omega\) be a convex polytope in \(\mathbb{R}^m\). Suppose there is a
vertices of the polytope belongs exactly to \(k\) linearly independent \((n - 1)\)-dimensional facets.
Then, the self-concordance of any barrier on \(\Omega\) is at least \(k\).

Using this, even if our maximum flow instance just consisted of \(O(m)\) edges in parallel this result
implies that a barrier for the polytope must have self-concordance at least \(\Omega(m)\). As we
have mentioned, we could circumvent this problem in some instances, e.g. when there are no
upper bounds, by simply producing a barrier for the dual polytope. However, any formulation
of the dual as a linear program seems to increase the number of variables by at least \(\otimes(m)\)
and cause any barrier to have self-concordance \(\Omega(m)\).\(^4\)

\(^4\)Note that this does not rule out a different reduction of the problem as minimizing a linear function over
a convex body for which there is a \(O(n)\) self concordant barrier. This explanation is simply meant to explain
the difficulty in using standard interior point analysis.
1.3 Our Contribution

In this paper we provide an $O(\sqrt{\text{rank}(A)L})$ iteration algorithm for solving linear programs of the form (1.2). This is the first general interior point method we are aware of that converges at a faster rate than the best self-concordance of the feasible region. As with our previous result, each iteration of our algorithm involves solving of $O(1)$ linear systems of the form $A^T DA \vec{x} = \vec{d}$. By applying this method to the linear programs analyzed in Daitch and Spielman [5], we achieve a running time of $\tilde{O}(m\sqrt{n})$.

We achieve this running time by a novel extension of the ideas in [18] to work with the primal linear program formulation (1.2) directly. Using an idea from [7], we create a 1-self concordant barrier for each of the $l_i \leq x \leq u_i$ constraints and run a primal path following algorithm with the sum of these barriers. While this would naively yield a $O(\sqrt{mL})$ iteration method, we show how to use weights in a similar manner as in [18] to improve the convergence rate to $O(\sqrt{\text{rank}(A)L})$.

While there are similarities between this analysis and our previous work, we cannot use that result directly. Changing from weighted path following to the dual linear program formulation to this primal formulation changes the behavior of the algorithm and in a sense moves degeneracies in manipulating weights to degeneracies in manipulating the current feasible point. This makes some parts of the analysis simpler and some more complicated. On the positive side, the optimization problem we need to solve to compute the weights becomes better conditioned and hence we no longer need to regularize that formula to compute the weights. Furthermore, inverting the behavior of the weights obviates the need for $r$-steps that were key to our analysis in our previous work.

Instead, we have to regularize the weight computation so that weight changes do not undo Newton steps on the feasible point and we have to do further work to show that Newton steps on the current feasible point are stable. In the dual formulation it was easy to assert that small Newton steps on the current point do not change the point multiplicatively. However, for this primal analysis this is no longer the case and we need to explicitly bound the size of the Newton step in both the $\ell_\infty$ norm and a weighted $\ell_2$ norm. Hence, we measure the centrality of our points by the size of the Newton step in a mixed norm of the form $\|\cdot\| = \|\cdot\|_\infty + C_{\text{norm}}\|\cdot\|_W$ to keep track of these two quantities simultaneously.

This measuring of Newton step size both with respect to $\ell_\infty$ and $\ell_2$ helps to explain how our method outperforms the self-concordance of the best barrier for the space. Self-concordance is based on $\ell_2$ analysis and the lower bounds for self-concordance are precisely the failure of the sphere to approximate a box. While ideally we would just do optimization over the box, $\ell_\infty$ is ripe with degeneracies that make this analysis difficult. Nevertheless, unconstrained minimization over a box is quite simple and in effect by working with this mixed norm and choosing weights to improve the conditioning we are taking advantage of the simplicity of minimizing $\ell_\infty$ over much of the domain and only paying for the $n$-self concordance of a barrier for the smaller subspace induce by the requirement that $A^T \vec{x} = \vec{b}$. We hope that this analysis may find further applications.

1.4 Overview

The rest of our paper is structured as follows: in Section 2 we present the notation we use throughout the paper, in Section 3 we describe the general linear program we attempt to solve...
and the basic quantities we use in our analysis, in Section 4 we present the key Lemmas and analysis used to achieve our desired convergence rate for solving linear programs, in Section 5 we show how to use these Lemmas to produce a complete algorithm for linear programming, and in Section 5 we show how to use this algorithm to achieve our desired running times for the maximum flow problem and its generalizations.

We note that some of the analysis in this paper is similar to our previous work and when the analysis or the intuition is nearly the same we often omit details. We encourage the reader to look at [18] for more detailed analysis and longer expositions on the analysis. Also note that throughout this paper we make no attempt to reduce polylogarithmic factors in our running times.

2 Notation

Here we introduce various notation that we will use throughout the paper. The notation in this paper is almost same as [18]. This section should be used primarily for reference as we will reintroduce various notation as we need them later in the paper.

**Variables:** We use the vector symbol, e.g. \( \vec{x} \), to denote a vector and we omit the symbol when we denote the vectors entries, e.g. \( \vec{x} = (x_1, x_2, \ldots) \). We use bold, e.g. \( \mathbf{A} \), to denote a matrix.

**Vector Operations:** We frequently overload notation applying scalar operations to vectors with the interpretation that these operations should be applied coordinate wise. For example, for vectors \( \vec{x}, \vec{y} \in \mathbb{R}^n \) we let \( \vec{x}/\vec{y} \in \mathbb{R}^n \) with \( [\vec{x}/\vec{y}]_i = (x_i/y_i) \) for all \( i \) and \( \log(\vec{x}) \in \mathbb{R}^n \) with \( [\log(\vec{x})]_i = \log(x_i) \). We define the median function \( \text{med}(\vec{x}, \vec{y}, \vec{z}) \), which is equal to Median of \( x_i, y_i \) and \( z_i \).

**Matrix Operations:** We call a symmetric matrix \( \mathbf{A} \in \mathbb{R}^{n \times n} \) positive definite (PD) if \( \vec{x}^T \mathbf{A} \vec{x} > 0 \) for all \( \vec{x} \in \mathbb{R}^n \) and we call \( \mathbf{A} \) positive semidefinite if \( \vec{x}^T \mathbf{A} \vec{x} \geq 0 \) for all \( \vec{x} \in \mathbb{R}^n \). For symmetric matrices \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n} \) we write \( \mathbf{A} \prec \mathbf{B} \) to indicate that \( \mathbf{B} - \mathbf{A} \) is PD (i.e. \( \vec{x}^T \mathbf{A} \vec{x} < \vec{x}^T \mathbf{B} \vec{x} \) for all \( \vec{x} \in \mathbb{R}^n \)) and we write \( \mathbf{A} \preceq \mathbf{B} \) to indicate that \( \mathbf{B} - \mathbf{A} \) is PSD (i.e. that \( \vec{x}^T \mathbf{A} \vec{x} \leq \vec{x}^T \mathbf{B} \vec{x} \) for all \( \vec{x} \in \mathbb{R}^n \)). We define \( \succeq \) and \( \preceq \) analogously. For \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m} \) we let \( \mathbf{A} \odot \mathbf{B} \) denote their Shur product, i.e. \( \mathbf{A} \odot \mathbf{B} \) \( ij \) \( \mathbf{A} \) \( B \) and we let \( \mathbf{A}^{(2)} \defeq \mathbf{A} \odot \mathbf{A} \).

**Diagonal Matrices** For \( \mathbf{A} \in \mathbb{R}^{n \times n} \) we let \( \text{diag}(\mathbf{A}) \in \mathbb{R}^n \) denote the vector such that \( \text{diag}(\mathbf{A})_i = A_{ii} \). For \( \vec{x} \in \mathbb{R}^n \) we let \( \text{diag}(\vec{x}) \in \mathbb{R}^{n \times n} \) be the diagonal matrix such that \( \text{diag}(\text{diag}(\vec{x})) = \vec{x} \). For \( \mathbf{A} \in \mathbb{R}^{n \times n} \) we let \( \text{diag}(\mathbf{A}) \) be the diagonal matrix such that \( \text{diag}(\text{diag}(\mathbf{A})) = \text{diag}(\mathbf{A}) \). For a vector \( \vec{x} \in \mathbb{R}^n \) when the meaning is clear from context we use \( \mathbf{X} \in \mathbb{R}^{n \times n} \) to denote \( \mathbf{X} \defeq \text{diag}(\vec{x}) \).

**Multiplicative Approximations:** In numerous places throughout this paper we need to convey that two vectors \( \vec{x} \) and \( \vec{y} \) are close multiplicatively. We often write \( \|X^{-1}(\vec{y} - \vec{x})\|_\infty \leq \epsilon \) to convey the equivalent fact that \( (1 - \epsilon)X \preceq \vec{Y} \preceq (1 + \epsilon)X \) or \( y_i \in [\lfloor (1 - \epsilon)x_i, (1 + \epsilon)x_i \rfloor] \) for all \( i \in [n] \). At times we will find it more convenient to write \( \|\log(\vec{x}) - \log(\vec{y})\| \leq \epsilon \) which is approximately equivalent for small \( \epsilon \). In Appendix, we show how to translate between these expressions.

**Standard Matrices:** We use \( \mathbb{R}_{\geq 0}^m \) to denote the vectors in \( \mathbb{R}^m \) where each coordinate is positive and for a matrix \( \mathbf{A} \in \mathbb{R}^{m \times n} \) and vector \( \vec{x} \in \mathbb{R}_{\geq 0}^n \) we define the following standard matrices and vectors

- Projection matrix \( \mathbf{P}_\mathbf{A}(\vec{x}) \in \mathbb{R}^{m \times n} \): \( \mathbf{P}_\mathbf{A}(\vec{x}) \defeq \mathbf{X}^{1/2} \mathbf{A}(\mathbf{A}^T \mathbf{X})^{-1} \mathbf{A} \mathbf{X}^{1/2} \)
• Leverage scores $\tilde{\sigma}_A(\bar{x}) \in \mathbb{R}^m$: $\tilde{\sigma}_A(\bar{x}) \equiv \text{diag}(P_A(\bar{x}))$

• Leverage matrix $\Sigma_A(\bar{x}) \in \mathbb{R}^{m \times m}$: $\Sigma_A(\bar{x}) \equiv \text{diag}(P_A(\bar{x}))$

• Projection Laplacian $\Lambda_A(\bar{x}) \in \mathbb{R}^{m \times m}$: $\Lambda_A(\bar{x}) \equiv \Sigma_A(\bar{x}) - P_A(\bar{x})^{(2)}$

**Convex Sets:** We call a set $U \subseteq \mathbb{R}^k$ convex if for all $\bar{x}, \bar{y} \in \mathbb{R}^k$ and all $t \in [0, 1]$ it holds that $t \cdot \bar{x} + (1 - t) \cdot \bar{y} \in U$. We call $U$ symmetric if $\bar{x} \in \mathbb{R}^k \iff -\bar{x} \in \mathbb{R}^k$. For any $p \in [1, \infty]$ we call the $\ell_p$ ball of radius $r > 0$ the set $\{\bar{x} \in \mathbb{R}^k \mid \|\bar{x}\|_p \leq r\}$. For any $\alpha > 0$ and convex set $U \subseteq \mathbb{R}^k$ we let $\alpha U \equiv \{\bar{x} \in \mathbb{R}^k \mid \alpha^{-1} \bar{x} \in U\}$.

## 3 Preliminaries

### 3.1 The Problem

Given $A \in \mathbb{R}^{n \times m}$, $\bar{b} \in \mathbb{R}^m$, and $\bar{c} \in \mathbb{R}^m$, the central goal in this paper is to efficiently solve the following problem

$$\min_{\bar{x} \in \mathbb{R}^m} \quad \bar{c}^T \bar{x} \quad \text{s.t.} \quad A^T \bar{x} = \bar{b} \quad \forall i \in [m] : l_i \leq x_i \leq u_i$$

where $l_i$ can be $-\infty$ and $u_i$ can be $+\infty$. We assume that that for all $i$ the domain of variable $x_i$, $\text{dom}(x_i) \equiv \{x : l_i \leq x \leq u_i\}$, is not degenerate, i.e. the interval is not the empty set ($l_i \leq u_i$), the interval is not a singleton ($l_i \neq u_i$), and the interval is not the entire real line ($l_i \neq -\infty$ or $u_i \neq +\infty$). Furthermore we make the standard assumptions that $A$ has full row rank, and therefore $m \geq n$, we assume that the interior of the polytope, $\Omega^0 \equiv \{\bar{x} \in \mathbb{R}^m : A^T \bar{x} = \bar{b}, l_i < x_i < u_i\}$, is non-empty.

If we set all the $l_i = 0$ and $u_i = +\infty$, i.e. $x_i \geq 0$ for all $i$ this is a linear program in standard form and it is well known that all linear programs could be written in this form. However, transformations to rewrite the linear program in standard form may increase the rank of the matrix and therefore slow down the running time linear programming algorithms. Hence we want to solve the problem in this form directly.

We solve this problem by performing weighted path following (See [18]). That is we maintain $\{\bar{x}, \bar{w}\} \in \Omega^0 \times \mathbb{R}^m_{>0}$ and we minimize the following penalized objective function

$$\min_{A^T \bar{x} = \bar{b}} f_t(\bar{x}, \bar{w}) \quad \text{where} \quad f_t(\bar{x}, \bar{w}) \equiv t \cdot \bar{c}^T \bar{x} + \sum_{i \in [m]} \bar{w}_i \phi_i(\bar{x}_i)$$

for increasing $t$ and small $\bar{w}$. Here each $\phi_i : \text{dom}(x_i) \to \mathbb{R}$ is a barrier function, i.e. $\lim_{x \to l_i} \phi_i(x) = \lim_{x \to u_i} \phi_i(x) = +\infty$. We assume that each $\phi_i$ is an 1-self-concordant barrier function defined as follows:

**Definition 2** (1-Self-Concordant Barrier Function). A thrice differentiable real valued barrier function $\phi$ on a convex subset of $\mathbb{R}$ is called a 1-self-concordant barrier function if

\[ \text{Typically 3.1 is written as } A \bar{x} = \bar{b} \text{ rather than } A^T \bar{x} = \bar{b} \text{ however we wrote it this way for consistency with the dual formulation in [18].}

\[ \text{The existence of a self concordant barrier for the domain is a standard assumption for path following methods. However typically higher dimensional barriers are allowed and there is greater flexibility for the constants in the inequalities [23].} \]
1. \( \phi \) is a self-concordant function, i.e.
\[
|\phi'''(x)| \leq 2(\phi''(x))^{3/2} \quad \text{for all } x \in \text{dom}(\phi).
\] (3.2)

2. One has
\[
|\phi'(x)| \leq \sqrt{\phi''(x)} \quad \text{for all } x \in \text{dom}(\phi).
\] (3.3)

The first condition bounds how quickly the second order approximation to the function can change and the second condition bounds how much force the barrier can exert in norm induced by the second order approximation. The existence of a self concordant barrier for the domain is a standard assumption for interior point methods [23]. However, for our case for each possible setting of the \( l_i \) and \( u_i \) there is an explicit 1-self-concordant barrier function we can use and we explain each case as follows.

- **Case (1):** \( l_i \) finite and \( u_i = \infty \): Here we use a log barrier defined as \( \phi_i(x)^\text{def} = -\log(x-l_i) \). For this barrier we have
\[
\phi'(x) = -\frac{1}{x-l_i}, \quad \phi''(x) = \frac{1}{(x-l_i)^2}, \quad \text{and} \quad \phi'''(x) = -\frac{2}{(x-l_i)^3}
\]
and therefore clearly \( |\phi'''(x)| = 2(\phi''(x))^{3/2} \), \( |\phi'(x)| = \sqrt{\phi''(x)} \), and \( \lim_{x \to l_i} \phi(x) = \infty \).

- **Case (2):** \( l_i = -\infty \) and \( u_i \) finite: Here we use a log barrier defined as \( \phi_i(x)^\text{def} = -\log(u_i-x) \). For this barrier we have
\[
\phi'(x) = \frac{1}{u_i-x}, \quad \phi''(x) = \frac{1}{(u_i-x)^2}, \quad \text{and} \quad \phi'''(x) = -\frac{2}{(u_i-x)^3}
\]
and therefore clearly \( |\phi'''(x)| = 2(\phi''(x))^{3/2} \), \( |\phi'(x)| = \sqrt{\phi''(x)} \), and \( \lim_{x \to u_i} \phi(x) = \infty \).

- **Case (3):** \( l_i \) finite and \( u_i \) finite: Here we use a trigonometric barrier\footnote{The authors are unaware of this barrier being used previously. In [7] they considered a similar setting of 0, 1, or 2 sided constraints in 3.1 however for the finite \( l_i \) and \( u_i \) case they considered either the the barrier \( -\log(u_i-x) - \log(x-l_j) \) which unfortunately does not meet condition 2 of Definition 2 (and rescaling forces the other condition to be violated) or the barrier \( -\log(\min(u_j-x, x-l_j)) + \min(u_j-x, x-l_j)/(u_j-l_j)/2 \) which unfortunately is not thrice differentiable and is not convenience for analysis.} defined as \( \phi(x)^\text{def} = -\log \cos(a_i x + b_i) \) for \( a_i = \frac{\pi}{u_i-l_i} \) and \( b_i = -\frac{\pi}{2} \frac{u_i+l_i}{u_i-l_i} \). Note for this choice as \( x \to u_i \) we have \( a_i x + b_i \to \frac{\pi}{2} \) and as \( x \to l_i \) we have \( a_i x + b_i \to -\frac{\pi}{2} \) and in both cases \( \phi(x) \to \infty \). Furthermore

\[
\phi'(x) = a_i \tan(a_i x + b_i), \quad \phi''(x) = \frac{a_i^2}{\cos^2(a_i x + b_i)}, \quad \text{and} \quad \phi'''(x) = 2a_i^3 \frac{\sin(a_i x + b_i)}{\cos^3(a_i x + b_i)}.
\]

Therefore, we have
\[
|\phi'''(x)| = \frac{2a_i^3 \sin(a_i x + b_i)}{\cos^3(a_i x + b_i)} \leq 2a_i^3 |\cos(a_i x + b_i)|^{-3} = 2(\phi''(x))^{3/2}
\]
and \( |\phi'(x)| \leq \frac{a_i}{|\cos x|} = \sqrt{\phi''(x)} \).
While there is much theory built around the theory of self concordant barrier functions, in the remainder of the paper we will only use two common properties about self concordant barriers functions. The first property is that the hessian of the barrier cannot change to quickly.

Lemma 3. [22, Thm 4.1.6] Suppose $\phi$ satisfies 3.2. For all $x, y \in \text{dom}(\phi)$ if $|s - t| \leq \frac{r}{\sqrt{\phi''(t)}}$ then

$$(1 - r)\sqrt{\phi''(s)} \leq \sqrt{\phi''(t)} \leq \frac{\sqrt{\phi''(s)}}{1 - r}$$

The second property allows us about how the force exerted by the barrier changes over the domain

Lemma 4. [22, Thm 4.2.4] Suppose $\phi$ satisfies 3.2. For all $x, y \in \text{dom}(\phi)$, we have

$$\phi'(x) \cdot (y - x) \leq 1$$

### 3.2 Measuring Centrality

Our weighted path following algorithm follows a simple iterative scheme. We assume we have some $\{\bar{x}, \bar{w}\} \in \{\Omega^0 \times \mathbb{R}^m_{\geq 0}\}$, and some function weight function $g(\bar{x}) : \Omega^0 \to \mathbb{R}_{\geq 0}$ and repeat the following

1. If $\bar{x}$ close to $\arg\min_{\bar{y} \in \Omega^0} f_t(\bar{y}, \bar{w})$, then increase $t$.
2. Otherwise, use projected Newton step to update $\bar{x}$ and move $\bar{w}$ closer to $g(\bar{x})$.
3. Repeat.

In this subsection, we explain is how to measure the distance from $\bar{x}$ to the minimum point $\arg\min_{\bar{x} \in \Omega} f_t(\bar{x}, \bar{w})$, i.e. how close $\bar{x}$ is to the weighted central path or centrality of $\bar{x}$. Whereas in [18] we could simply measure centrality by the size of the Newton step in the hessian norm here we need to do additional work to ensure that we can reason about multiplicative changes in the hessian.

To motivate our centrality measure we first need to compute the projected Newton step for $\bar{x}$. For all $\bar{x} \in \Omega$, we let $\bar{\phi}(\bar{x}) \in \mathbb{R}^m$ defined by $\bar{\phi}(\bar{x})_i = \phi_i(\bar{x}_i)$ for all $i \in [m]$. We define $\bar{\phi}'(\bar{x})$, $\bar{\phi}''(\bar{x})$, and $\bar{\phi}'''(\bar{x})$ similarly and we let $\Phi, \Phi', \Phi'', \Phi'''$ denote the diagonal of these matrices. Using this, we have that

$$\nabla_x f_t(\bar{x}, \bar{w}) = t \cdot \bar{c} + \bar{w} \bar{\phi}'(\bar{x}) \quad \text{and} \quad \nabla_{xx} f_t(\bar{x}, \bar{w}) = W\Phi''(\bar{x})$$

Therefore, we have that a Newton step for $\bar{x}$ is given by

$$\tilde{h}_t(\bar{x}, \bar{w}) = -(W\Phi''(\bar{x}))^{-1/2} P_{A^T(W\Phi''(\bar{x}))^{-1/2}} (W\Phi''(\bar{x}))^{-1/2} \nabla_x f_t(\bar{x}, \bar{w})$$

$$= -\Phi''(\bar{x})^{-1/2} P_{\bar{x}, \bar{w}} W^{-1} \Phi''(\bar{x})^{-1/2} \nabla_x f_t(\bar{x}, \bar{w})$$

where $P_{A^T(W\Phi''(\bar{x}))^{-1/2}}$ is the orthogonal projection onto the kernel of $A^T(W\Phi''(\bar{x}))^{-1/2}$ and $P_{\bar{x}, \bar{w}}$ is the orthogonal projection onto the kernel of $A^T(\Phi''(\bar{x}))^{-1/2}$ with respect to the norm $\| \cdot \|_W$ given by

$$P_{\bar{x}, \bar{w}} \overset{\text{def}}{=} I - W^{-1} A_x (A_x^T W^{-1} A_x)^{-1} A_x^T \quad \text{for} \quad A_x \overset{\text{def}}{=} \Phi''(\bar{x})^{-1/2} A.$$
As with standard convergence analysis of Newton’s method, we wish to keep the Newton step size in the hessian norm, i.e. \( \| \tilde{h}_t(\bar{x}, \bar{w}) \|_{W^{\phi}_t(\bar{x})} = \| \sqrt{\phi''(\bar{x})} \tilde{h}_t(\bar{x}, \bar{w}) \|_W \) small and the multiplicative change in the hessian, \( \| \sqrt{\phi''(\bar{x})} \tilde{h}_t(\bar{x}, \bar{w}) \|_{\infty} \), small (See Lemma 4). While in the unweighted case we can bound the multiplicative change by the change in the hessian norm (since \( \| \cdot \|_{\infty} \leq \| \cdot \|_2 \)), here we would like to use small weights and cannot do this comparison.

To track both these quantities simultaneously, we define the mixed norm for all \( \bar{y} \in \mathbb{R}^m \) by

\[
\| \bar{y} \|_{\bar{w}+\infty} \overset{\text{def}}{=} \| \bar{y} \|_{\infty} + C_{\text{norm}} \| \bar{y} \|_W
\]

for some \( C_{\text{norm}} > 0 \) that we define later. Note that \( \| \cdot \|_{\bar{w}+\infty} \) is indeed a norm for \( \bar{w} \in \mathbb{R}^m_+ \) as in this case both \( \| \cdot \|_{\infty} \) and \( \| \cdot \|_W \) are norms. However, rather than measuring centrality by the quantity \( \| \sqrt{\phi''(\bar{x})} \tilde{h}_t(\bar{x}, \bar{w}) \|_{\bar{w}+\infty} = \| P_{\bar{x},\bar{w}} \left( \nabla_x f_t(\bar{x}, \bar{w}) \right) \|_{\bar{w}+\infty} \), we instead will find it more convenient to use the following idealized form

\[
\delta_t(\bar{x}, \bar{w}) \overset{\text{def}}{=} \min_{\bar{y} \in \mathbb{R}^n} \left\| \frac{\nabla_x f_t(\bar{x}, \bar{w}) - A_{\bar{y}}}{\bar{w} \sqrt{\phi''(\bar{x})}} \right\|_{\bar{w}+\infty}.
\]

We justify this definition by showing these two quantities differ by at most a multiplicative factor of \( \| P_{\bar{x},\bar{w}} \|_{\bar{w}+\infty} \) as follows

\[
\delta_t(\bar{x}, \bar{w}) \leq \left\| \sqrt{\phi''(\bar{x})} \tilde{h}_t(\bar{x}, \bar{w}) \right\|_{\bar{w}+\infty} \leq \| P_{\bar{x},\bar{w}} \|_{\bar{w}+\infty} \cdot \delta_t(\bar{x}, \bar{w}). \tag{3.4}
\]

This a direct consequence of the more general Lemma 26 that we prove in the appendix.

We summarize the results of this section as follows

**Definition 5** (Centrality Measure). For \( \{ \bar{x}, \bar{w} \} \in \{ \Omega^0 \times \mathbb{R}^{m_0}_+ \} \) and \( t \geq 0 \), we let \( \tilde{h}_t(\bar{x}, \bar{w}) \) denote the projected Newton step for \( \bar{x} \) on the penalized objective \( f_t \) given by

\[
\tilde{h}_t(\bar{x}, \bar{w}) = -\frac{1}{\sqrt{\phi''(\bar{x})}} P_{\bar{x},\bar{w}} \left( \nabla_x f_t(\bar{x}, \bar{w}) \right)
\]

where \( P_{\bar{x},\bar{w}} \) is the orthogonal projection onto the kernel of \( A^T(\Phi'')^{-1/2} \) with respect to the norm \( \| \cdot \|_W \). We measure the centrality of \( \{ \bar{x}, \bar{w} \} \) by

\[
\delta_t(\bar{x}, \bar{w}) \overset{\text{def}}{=} \min_{\bar{y} \in \mathbb{R}^n} \left\| \frac{\nabla_x f_t(\bar{x}, \bar{w}) - A_{\bar{y}}}{\bar{w} \sqrt{\phi''(\bar{x})}} \right\|_{\bar{w}+\infty}
\]

where \( \| \cdot \|_{\bar{w}+\infty} \) is defined for all \( \bar{y} \in \mathbb{R}^m \) by \( \| \bar{y} \|_{\bar{w}+\infty} \overset{\text{def}}{=} \| \bar{y} \|_{\infty} + C_{\text{norm}} \| \bar{y} \|_W \) for some \( C_{\text{norm}} > 0 \) that we defined earlier.

\(^9\)For any norm \( \| \cdot \| \), the operator norm of \( M \) is defined by \( \| M \| = \sup_{\| x \|_1} \| Mx \| \).
3.3 The Weight Function

With the Newton step and centrality conditions defined, the specification of our algorithm becomes more clear. Our algorithm is as follows

1. If \( \delta_t(\bar{x}, \bar{w}) \) is small, then increase \( t \).
2. Set \( \bar{x}(\text{new}) \leftarrow \bar{x} + \bar{h}_t(\bar{x}, \bar{w}) \) and move \( \bar{w}(\text{new}) \) towards \( \bar{g}(\bar{x}(\text{new})) \).
3. Repeat.

To prove this algorithm converges, we need show what happens to \( \delta_t(\bar{x}, \bar{w}) \) when we change \( t, \bar{x}, \bar{w} \).

At the heart of this paper is understanding what conditions we need to impose on the weight function \( \bar{g}(\bar{x}) : \Omega^0 \rightarrow \mathbb{R}^m_0 \) so that we can bound this change in \( \delta_t(\bar{x}, \bar{w}) \) and hence achieve fast converge rates. In Lemma 7 we show that the effect of changing \( t \) on \( \delta_t \) is bounded by \( C \) norm and \( \| \bar{g}(\bar{x}) \|_1 \), in Lemma 8 we show that the effect of a \( \bar{x} \) Newton step on \( \delta_t \) is bounded by \( \| \bar{P}_{\bar{x},\bar{g}(\bar{x})} \| \bar{g}(\bar{x})+\infty \), and in Lemma 9 and 10 we show how changing \( \bar{w} \) as \( \bar{g}(\bar{x}) \) changes is bounded by \( \| \bar{G}(\bar{x})^{-1}\bar{G}'(\bar{x})(\bar{g}(\bar{x}))^{-1/2} \| \bar{g}(\bar{x})+\infty \).

Hence for the remainder of the paper we assume we have a weight function \( \bar{g}(\bar{x}) : \Omega^0 \rightarrow \mathbb{R}^m_0 \) and make the following assumptions regarding our weight function.

**Definition 6. [Weight Function]** A weight function is any differentiable function from \( \bar{g} : \Omega^0 \rightarrow \mathbb{R}^m_0 \). We associate the following constants with a weight function:

- **Size** \( c_1 : 1 \leq c_1 \) and \( \| \bar{g}(\bar{x}) \|_1 \leq c_1(\bar{g}) \).
- **Condition Number** \( c_\gamma : 1 \leq c_\gamma \leq 2 \) and \( \| \bar{P}_{\bar{x},\bar{w}} \|_{\bar{w}+\infty} \leq c_\gamma(\bar{g}) \) for any \( \bar{w} \) such that \( \frac{4}{5}\bar{g}(\bar{x}) \leq \bar{w} \leq \frac{5}{4}\bar{g}(\bar{x}) \).
- **Consistency** \( c_\delta : \| \bar{G}(\bar{x})^{-1}\bar{G}'(\bar{x})(\bar{g}(\bar{x}))^{-1/2} \| \bar{g}(\bar{x})+\infty \leq c_\delta \leq 1 \)

where these inequalities hold for all \( \bar{x} \in \Omega^0 \). Furthermore, we require that \( c_\delta \cdot c_\gamma < 1 \) and \( \| \bar{g}(\bar{x}) \|_\infty \leq 2 \).

4 The Weighted Path

In this section, we provide the main lemmas we need to provide a \( O(\sqrt{\text{rank}(A)L}) \) iteration weighted path following algorithm for 3.1. In Section 4.1, 4.2, and 4.3 we show how centrality, \( \delta_t(\bar{x}, \bar{w}) \), is affected by changing \( t, \bar{x} \in \Omega^0, \) and \( \bar{w} \in \mathbb{R}^m_0 \) respectively. In Section 4.4 we then show how to use these Lemmas to improve centrality using approximate computations of the weight function, \( \bar{g} : \Omega^0 \rightarrow \mathbb{R}^m_0 \), and in Section 4.5 we present our weight function and show how to compute it approximately.

In the remaining sections we show how to put these lemmas together to actually solve 3.1 in \( O(\sqrt{\text{rank}(A)L}) \) iterations of \( \bar{O}(1) \) linear system solves (Section 5) and ultimately solve a variety of flow problems in \( \bar{O}(m\sqrt{n}... \) time (Section 6).
4.1 The effect of $t$ change on $\delta_t$

Here we analyze the effect of a Newton step on the centrality. We show that for a centered point $\{\bar{x}, \bar{w}\} \in \{\Omega^0 \times \mathbb{R}_+^m\}$ the rate of increase is governed by $C_{\text{norm}}$ and $c_1$.

**Lemma 7.** For all $\{\bar{x}, \bar{w}\} \in \{\Omega^0 \times \mathbb{R}_+^m\}$ and $t > 0$, $\alpha \geq 0$ we have

$$\delta_{(1+\alpha)t}(\bar{x}, \bar{w}) \leq (1 + \alpha)\delta_t(\bar{x}, \bar{w}) + \alpha (1 + C_{\text{norm}}\sqrt{c_1}) \ .$$

**Proof.** Let $\eta_t \in \mathbb{R}^n$ be such that

$$\delta_t(\bar{x}, \bar{w}) = \left\| \nabla_{\bar{x}} f_t(\bar{x}, \bar{w}) + A \eta_t \right\|_{\bar{w}+\infty} = \left\| t \cdot \bar{c} + \bar{w} \bar{g}(\bar{x}) + A \eta_t \right\|_{\bar{w}+\infty} \ .$$

Applying this to the definition of $\delta_{(1+\alpha)t}$ and using that $\|\cdot\|_{\bar{w}+\infty}$ is a norm then yields

$$\delta_{(1+\alpha)t}(\bar{x}, \bar{w}) = \min_{\eta_t \in \mathbb{R}^n} \left\| (1 + \alpha)t \cdot \bar{c} + \bar{w} \bar{g}(\bar{x}) + A \eta_t \right\|_{\bar{w}+\infty} \quad \text{(Definition of centrality)}$$

$$\leq \left\| (1 + \alpha)t \cdot \bar{c} + \bar{w} \bar{g}(\bar{x}) + A (1 + \alpha) \eta_t \right\|_{\bar{w}+\infty}$$

$$\leq (1 + \alpha) \left\| \frac{t \cdot \bar{c} + \bar{w} \bar{g}(\bar{x}) + A \eta_t}{\bar{w} \sqrt{\bar{g}'(\bar{x})}} \right\|_{\bar{w}+\infty} + \alpha \left\| \frac{\bar{w} \bar{g}(\bar{x})}{\bar{w} \sqrt{\bar{g}'(\bar{x})}} \right\|_{\bar{w}+\infty}$$

$$= (1 + \alpha)\delta_t(\bar{x}, \bar{w}) + \alpha \left( \left\| \frac{\bar{g}'(\bar{x})}{\sqrt{\bar{g}'(\bar{x})}} \right\|_\infty + C_{\text{norm}} \left\| \frac{\bar{g}(\bar{x})}{\sqrt{\bar{g}'(\bar{x})}} \right\|_W \right)$$

Using that $|\bar{g}'(\bar{x})| \leq \sqrt{\bar{g}'(\bar{x})}$ for all $i \in [n]$ and $\bar{x} \in \mathbb{R}^n$ by Definition 2, we complete the result. \hfill \Box

4.2 The effect of $\bar{x}$ change on $\delta_t$

Here we analyze the effect of a Newton step on the centrality. We show for sufficiently central $\{\bar{x}, \bar{w}\} \in \{\Omega^0 \times \mathbb{R}_+^m\}$ and $\bar{w}$ sufficiently close to $\bar{g}(\bar{x})$ a Newton steps $\bar{x}$ has quadratic convergence properties.

**Lemma 8.** Let $\{\bar{x}_0, \bar{w}\} \in \{\Omega^0 \times \mathbb{R}_+^m\}$ such that $\delta_t(\bar{x}_0, \bar{w}) \leq \frac{1}{10}$ and $\frac{\bar{g}}{2} \leq \bar{w} \leq \frac{3}{2} \bar{g}(\bar{x})$ and consider a Newton step $\bar{x}_1 = \bar{x}_0 + \eta_1(\bar{x}, \bar{w})$. Then $\bar{x}_1 \in \Omega^0$ and $\delta_t(\bar{x}_1, \bar{w}) \leq 4 (\delta_t(\bar{x}_0, \bar{w}))^2$.

**Proof.** Let $\bar{\phi}_0 \overset{\text{def}}{=} \bar{\phi}(\bar{x}_0)$ and let $\bar{\phi}_1 \overset{\text{def}}{=} \bar{\phi}(\bar{x}_1)$. By the definition of $\eta_t(\bar{x}_0, \bar{w})$ and the formula of $P_{\bar{x}_0, \bar{w}}$ we know that there is some $\eta_0 \in \mathbb{R}^n$ such that

$$-\sqrt{\bar{\phi}_0^* \eta_1(\bar{x}_0, \bar{w})} = \frac{\bar{c} + \bar{w} \bar{\phi}_0 - A \bar{\eta}_0}{\bar{w} \sqrt{\bar{\phi}_0^*}} \ .$$
Therefore, $A\tilde{n}_0 = \tilde{c} + \tilde{w}\tilde{\phi}_0 + \tilde{w}\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})$. Recalling the definition of $\delta_t$ this implies that

$$\delta_t(\tilde{x}_1, \tilde{w}) = \min_{\tilde{n} \in \mathbb{R}^n} \left\| \tilde{c} + \tilde{w}\tilde{\phi}'_1 - A\tilde{n} \right\|_{\tilde{w}+\infty} \leq \left\| \tilde{c} + \tilde{w}\tilde{\phi}'_1 - A\tilde{n}_0 \right\|_{\tilde{w}+\infty} \leq \left\| \tilde{w}(\tilde{\phi}'_1 - \tilde{\phi}_0) - \tilde{w}\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w}) \right\|_{\tilde{w}+\infty} = \left\| (\tilde{\phi}'_1 - \tilde{\phi}_0) - \tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w}) \right\|_{\tilde{w}+\infty}.$$

By the mean value theorem, we have $\tilde{\phi}'_1 - \tilde{\phi}_0 = \tilde{\phi}''(\tilde{\theta})\tilde{h}_t(\tilde{x}_0, \tilde{w})$ for some $\tilde{\theta}$ between $\tilde{x}_0$ and $\tilde{x}_1$ coordinate-wise. Hence,

$$\delta_t(\tilde{x}_1, \tilde{w}) \leq \left\| \frac{\tilde{\phi}''(\tilde{\theta})\tilde{h}_t(\tilde{x}_0, \tilde{w}) - \tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})}{\sqrt{\tilde{\phi}'_1}} \right\|_{\tilde{w}+\infty} = \left\| \frac{(\tilde{\phi}''(\tilde{\theta}) - \tilde{\phi}'_0)}{\sqrt{\tilde{\phi}'_1}}(\sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})}) \right\|_{\tilde{w}+\infty}.$$

To bound the first term, we use Lemma 3 as follows

$$\left\| \frac{\tilde{\phi}''(\tilde{\theta}) - \tilde{\phi}'_0}{\sqrt{\tilde{\phi}'_1}} \right\|_{\tilde{w}+\infty} \leq \left\| \frac{\tilde{\phi}''(\tilde{\theta}) - \tilde{\phi}'_0}{\tilde{\phi}'_0} \right\|_{\tilde{w}+\infty} \cdot \left\| \frac{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})}{\sqrt{\tilde{\phi}'_1}} \right\|_{\tilde{w}+\infty} \leq \left( 1 - \left\| \sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})} \right\|_{\tilde{w}+\infty} \right)^{-2} - 1 \cdot \left( 1 - \left\| \sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})} \right\|_{\tilde{w}+\infty} \right)^{-1}. $$

Using (3.4), i.e. Lemma 26, the bound $c, \leq 2$, and the assumption on $\delta_t(\tilde{x}_0, \tilde{w})$, we have

$$\left\| \sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})} \right\|_{\tilde{w}+\infty} \leq \left\| \sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})} \right\|_{\tilde{w}+\infty} \leq c, \delta_t(\tilde{x}_0, \tilde{w}) \leq \frac{1}{5}. $$

Using $(1 - t)^{-2} - 1 \cdot (1 - t)^{-1} \leq 4t$ for $t \leq 1/5$, we have

$$\left\| \frac{\tilde{\phi}''(\tilde{\theta}) - \tilde{\phi}'_0}{\sqrt{\tilde{\phi}'_1}} \right\|_{\tilde{w}+\infty} \leq 4 \left\| \sqrt{\tilde{\phi}'_0 h_t(\tilde{x}_0, \tilde{w})} \right\|_{\tilde{w}+\infty}. $$

Combining the above formulas yields that $\delta_t(\tilde{x}_1, \tilde{w}) \leq 4 (\delta_t(\tilde{x}_0, \tilde{w}))^2$ as desired. \hfill \Box

### 4.3 The effect of $\tilde{w}$ change on $\delta_t$

In the previous subsection we used the assumption that the weights, $\tilde{w}$, was multiplicatively close to the output of the weight function, $\tilde{\theta}(\tilde{x})$, for the current point $\tilde{x} \in \Omega^0$. Therefore, when we change $\tilde{x}$ we will need to change $\tilde{w}$ to move $\tilde{w}$ close to $\tilde{\theta}(\tilde{x})$. Here we bound how much $\tilde{\theta}(\tilde{x})$ can move as we move $\tilde{x}$ (Lemma 9) and we bound how much changing $\tilde{w}$ can hurt centrality (Lemma 10). Together these lemmas will allow us to show that we can keep $\tilde{w}$ close to $\tilde{\theta}(\tilde{x})$ while still improving centrality (Lemma 4.4).
Lemma 9. For all $t \in [0, 1]$, let $\tilde{x}_t \overset{\text{def}}{=} \tilde{x}_0 + t\Delta x$ for some $\Delta x \in \mathbb{R}^m$ such that $\tilde{x}_t \in \Omega^0$, $g_t = g(\tilde{x}_t)$ and $\epsilon = \left\| \sqrt{\phi_0''(\Delta x)} \right\|_{\bar{g}_0 + \infty} \leq \frac{1}{10}$. Then

$$\| \log(g_t) - \log(g_0) \|_{\bar{g}_0 + \infty} \leq c_\delta \epsilon (1 + 4\epsilon) \leq 1/5$$

and for all $s, t \in [0, 1]$ and for all $g \in \mathbb{R}^m$ we have

$$\| \bar{g} \|_{\bar{g}_s + \infty} \leq (1 + 2\epsilon) \| \bar{g} \|_{\bar{g}_t + \infty}.$$

Proof. Let $q : [0, 1] \to \mathbb{R}^m$ be given by $q(t) = \log(g_t)$ for all $t \in [0, 1]$. Then, we have

$$q(t) = G_t^{-1}G_t'(\tilde{x}_t).$$

Let $Q(t) = \| q(t) - q(0) \|_{\bar{g}_0 + \infty}$. Using Jensen’s inequality we have that for all $u \in [0, 1]$

$$Q(u) \leq \frac{\epsilon}{1 - \epsilon} \int_0^u \left\| G_t^{-1} \left( \frac{\phi''(u)}{\phi'(u)} \right)^{-1/2} \left\| \sqrt{\phi''(\Delta x)} \right\|_{\bar{g}_0 + \infty} dt. \quad (4.1)$$

Using Lemma 3 and $\epsilon \leq \frac{1}{10}$, we have for all $t \in [0, 1]$

$$\left\| \sqrt{\phi''(\Delta x)} \right\|_{\bar{g}_0 + \infty} \leq \left\| \sqrt{\phi''(\Delta x)} \right\|_{\bar{g}_0 + \infty} \left\| \sqrt{\phi''(\Delta x)} \right\|_{\bar{g}_0 + \infty} \leq \left(1 - \frac{\epsilon}{1 - \epsilon}\right)^{-1} \left\| \sqrt{\phi''(\Delta x)} \right\|_{\bar{g}_0 + \infty} \leq \frac{\epsilon}{1 - \epsilon}.$$

Thus, we have

$$Q(u) \leq \frac{\epsilon}{1 - \epsilon} \int_0^u \left\| G_t^{-1} \left( \frac{\phi''(u)}{\phi'(u)} \right)^{-1/2} \right\|_{\bar{g}_0 + \infty} dt. \quad (4.2)$$

Note that $Q$ is monotonically increasing. Let $\theta = \sup_{u \in [0, 1]} Q(u) \leq c_\delta \epsilon (1 + 4\epsilon) \leq 1/5$. By Lemma 27 we know that for all $s, t \in [0, \theta]$, we have

$$\left\| \frac{g(s) - g(t)}{g(s)} \right\|_{\infty} \leq \left\| q(s) - q(t) \right\|_{\infty} + \left\| q(s) - q(t) \right\|_{\infty}^2$$

and therefore

$$G_s \leq \left(1 + \left\| q(s) - q(t) \right\|_{\infty} + \left\| q(s) - q(t) \right\|_{\infty}^2 \right)^2 \leq (1 + c_\delta \epsilon (1 + 4\epsilon))^2$$

Consequently

$$\| \bar{g} \|_{\bar{g}_s + \infty} \leq (1 + c_\delta \epsilon (1 + 4\epsilon)) \| \bar{g} \|_{\bar{g}_s + \infty} \leq (1 + 2\epsilon) \| \bar{g} \|_{\bar{g}_s + \infty}.$$
Using (4.2), we have for all $u \in [0, \theta]$

\[
Q(u) \leq Q(u) \leq \epsilon - \epsilon \hat{u}_0 \cdot \left\| G_{t}^{-1} G'_t \left( \frac{\phi'_t}{\phi''_t} \right)^{-1/2} \right\|_{\tilde{g}_{t+\infty}} dt
\]

Consequently, we have that $\theta = 1$ and we have the desired result. \hfill \square

**Lemma 10.** Let $\vec{v}, \vec{w} \in \mathbb{R}^n_{>0}$ such that $\epsilon = \| \log(\vec{w}) - \log(\vec{v}) \|_{\vec{w}+\infty} \leq 1/10$. Then for $\vec{x} \in \Omega^0$ we have

\[
\delta_t(\vec{x}, \vec{v}) \leq (1 + 4\epsilon)(\delta_t(\vec{x}, \vec{w}) + \epsilon).
\]

**Proof.** Let $\vec{\eta}_{\vec{w}}$ be such that

\[
\delta_t(\vec{x}, \vec{w}) = \left\| \vec{c} + \vec{w} \vec{\phi}'(\vec{x}) - \vec{A} \vec{\eta}_{\vec{w}} \right\|_{\vec{w}+\infty}
\]

Furthermore, note that by Lemma 27 we have

\[
V \preceq (1 + \epsilon + \epsilon^2)W \preceq (1 + \epsilon)^2W \quad \text{and} \quad W \preceq (1 + \epsilon)^2V
\]

Using these, we bound the energy with the new weights as follows

\[
\delta_t(\vec{x}, \vec{v}) = \min_{\vec{\eta}_{\vec{w}}} \left\| \vec{c} + \vec{v} \vec{\phi}'(\vec{x}) - \vec{A} \vec{\eta}_{\vec{w}} \right\|_{\vec{v}+\infty} \leq \left\| \vec{c} + \vec{v} \vec{\phi}'(\vec{x}) - \vec{A} \vec{\eta}_{\vec{w}} \right\|_{\vec{v}+\infty} \leq (1 + \epsilon) \left\| \vec{c} + \vec{v} \vec{\phi}'(\vec{x}) - \vec{A} \vec{\eta}_{\vec{w}} \right\|_{\vec{v}+\infty}
\]

\[
\leq (1 + \epsilon) \cdot \sqrt{\epsilon} \langle \vec{v} - \vec{w}, \vec{\phi}'(\vec{x}) \rangle \leq (1 + \epsilon) \cdot \left\| \vec{\phi}'(\vec{x}) \right\|_{\vec{v}+\infty} \cdot \left\| \frac{\vec{v} - \vec{w}}{\vec{v}} \right\|_{\vec{v}+\infty}
\]

Using that $|\phi'_i(\vec{x})| \leq \sqrt{\phi''_i(\vec{x})}$ for all $i \in [m]$ by Definition 2 and using Lemma 27 we have that

\[
\delta_t(\vec{x}, \vec{v}) \leq (1 + \epsilon)^3 \delta_t(\vec{x}, \vec{w}) + (1 + \epsilon)^2 \epsilon \\
\leq (1 + 4\epsilon)(\delta_t(\vec{x}, \vec{w}) + \epsilon).
\]

\hfill \square
4.4 Centering Step

In the previous paper [18], we defined the chasing \( \vec{0} \) game. The player has a point \( \vec{x} \in \mathbb{R}^m \) and wants to keep this point close to \( \vec{0} \) in \( \ell_\infty \) norm, whereas the adversary wants to move \( \vec{x} \) away from \( \vec{0} \). We found the following strategy to play the game.

**Theorem 11.** [18] For \( \vec{x}_0 \in \mathbb{R}^m \) and \( 0 < \epsilon < \frac{1}{5} \), consider the two play game consisting of repeating the following for \( k = 1, 2, \ldots \)

1. The adversary chooses \( U^{(k)} \subseteq \mathbb{R}^k \), \( \vec{w}^{(k)} \in U^{(k)} \), and sets \( \vec{y}^{(k)} = \vec{x}^{(k)} + \vec{w}^{(k)} \).
2. The adversary chooses \( \vec{z}^{(k)} \) such that \( \| \vec{z}^{(k)} - \vec{y}^{(k)} \|_\infty \leq R \) and reveals \( \vec{z}^{(k)} \) and \( U^{(k)} \) to the player.
3. The player choose \( \Delta^{(k)} \in (1 + \epsilon)U^{(k)} \) and sets \( \vec{x}^{(k+1)} = \vec{y}^{(k)} + \Delta^{(k)} \).

Suppose that each \( U^{(k)} \) is a symmetric convex set that contains an \( \ell_\infty \) ball of radius \( r_k \) and is contained in a \( \ell_\infty \) ball of radius \( R_k \leq R \) and consider the strategy

\[
\Delta^{(k)} = (1 + \epsilon) \arg\min_{\Delta \in U^{(k)}} \left\langle \nabla \Phi_{\alpha}(\vec{z}), \Delta \rightangle
\]

where \( \alpha = \frac{\epsilon}{12R} \) and \( \Phi_{\alpha}(\vec{x}) = \sum_i (e^{\alpha x_i} + e^{-\alpha x_i}) \). Let \( \tau = \max_k \frac{R_k}{r_k} \) and suppose \( \Phi_{\alpha}(\vec{x}^{(0)}) \leq \frac{8n\tau}{\epsilon} \).

This strategy guarantees that for all \( k \) we have

\[
\Phi_{\alpha}(\vec{x}^{(k)}) \leq \frac{8n\tau}{\epsilon} \text{ and } \|\vec{x}^{(k)}\|_\infty \leq \frac{12R}{\epsilon} \log \left( \frac{8n\tau}{\epsilon} \right).
\]

We can think updating weight is playing this game. We are only given oracle to compute the weight function approximately and we want to make sure \( \vec{w} \) close to \( \vec{g}(\vec{x}) \).

Formally, we will measure its distance from the optimal weights in log scale by \( \Psi(\vec{x}, \vec{w}) \overset{def}{=} \log(\vec{g}(\vec{x})) - \log(\vec{w}) \). Our goal will be to keep \( \left\| \Psi(\vec{x}, \vec{w}) \right\|_{\|w\|_\infty} \leq K \) for some error \( K \) that is just small enough to not impair our ability to decrease \( \delta_t \) linearly and not to impair our ability to approximate \( \vec{g} \). We will attempt to do this without moving \( \vec{w} \) too much in \( \|\cdot\|_{\|w\|_\infty} \).

\[
(\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) = \text{centeringInexact}(\vec{x}, \vec{w}, K)
\]

| Step | Description |
|------|-------------|
| 1.   | \( c_k = \frac{1}{1 - \epsilon c_k}, R = \frac{K}{48c_k \log(400m)}, \delta_t = \delta_t(\vec{x}, \vec{w}) \) and \( \epsilon = \frac{1}{2c_k} \).
| 2.   | \( \vec{x}^{(\text{new})} = \vec{x} - \frac{1}{\sqrt{\phi^{(t)}(\vec{x})}} P_{\vec{x},\vec{w}} \left( \frac{t\vec{z} - \vec{w} \vec{g}(\vec{x})}{\vec{w} \sqrt{\phi^{(t)}(\vec{x})}} \right) \).
| 3.   | Let \( U = \{ \vec{x} \in \mathbb{R}^m \mid \|\vec{x}\|_{\|w\|_\infty} \leq \left( 1 - \frac{7}{8c_k} \right) \delta_t \} \).
| 4.   | Find \( \vec{z} \) such that \( \|\vec{z} - \log(\vec{g}(\vec{x}^{(\text{new})}))\|_{\|w\|_\infty} \leq R. \)
| 5.   | \( \vec{w}^{(\text{new})} = \exp \left( \log(\vec{w}) + (1 + \epsilon) \arg\min_{\vec{w} \in U} \left\langle \nabla \Phi_{\frac{12R}{\epsilon}}(\vec{z} - \log(\vec{w})), \vec{w} \right\rangle \right) \).
| 6.   | \( \vec{x}^{(\text{new})} = \arg\min_{\vec{x} \in U} \|\vec{x} - \vec{x}^{(\text{new})}\|^2. \)

The problem \( \arg\min_{\vec{w} \in U} (\cdot, \vec{w}) \) in step 5 is simply a projection onto the convex set \( U \) and it can be done in \( O(1) \) depth and \( O(m) \) work. See section A.1 for details. The last step of the algorithm is to make the constraint \( A^T \vec{x} = \vec{b} \) will be enforced for inexact computation. In the following theorem, we will assume exact computation for simplicity and the last step is not necessary.
Theorem 12. For simplicity, we assume that \(24m^{1/4} \geq c_k \overset{\text{def}}{=} \frac{1}{1-c_k c_y} \geq 4\), \(1 \leq C_{\text{norm}} \leq 2c_k\) and \(K \leq \frac{1}{2c_k}\). Let \(\bar{\Psi}(\bar{x}, \bar{w}) \overset{\text{def}}{=} \log(\bar{g}(\bar{x})) - \log(\bar{w})\). Suppose that

\[
\delta \overset{\text{def}}{=} \delta_t(\bar{x}, \bar{w}) \leq \frac{K}{48c_k \log(400m)} \quad \text{and} \quad \Phi_\alpha(\bar{\Psi}(\bar{x}, \bar{w})) \leq 10^5 m^2.
\]

Let \((\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})}) = \text{centeringInexact}(\bar{x}, \bar{w}, K)\), then

\[
\delta_t(\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})}) \leq \left(1 - \frac{1}{4c_k}\right) \delta \quad \text{and} \quad \Phi_\alpha(\bar{\Psi}(\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})})) \leq 10^5 m^2.
\]

Also, we have \(\|\log(\bar{g}(\bar{x}^{(\text{new})})) - \log(\bar{w})\|_\infty \leq K\).

Proof. By Lemma 9, (3.4), the assumption on \(\delta\) and the assumption on \(c_k\), we have

\[
\left\| \log \left(\frac{\bar{g}(\bar{x}^{(\text{new})})}{\bar{g}(\bar{x})}ight) \right\|_{\infty} \leq \left\| \log \left(\frac{\bar{w}}{\bar{g}(\bar{x})}\right) \right\|_{\infty} + \left\| \log (\bar{w}) - \log (\bar{g}(\bar{x})) \right\|_{\infty} \leq \frac{21}{20} K \leq \frac{1}{16c_k}.
\]

Hence, we have

\[
\left\| \log \left(\frac{\bar{g}(\bar{x}^{(\text{new})})}{\bar{g}(\bar{x})}\right) \right\|_{\infty} \leq \left(1 + \frac{1}{16c_k}\right) \left(1 - \frac{15}{16c_k}\right) \delta
\]
\[
\leq \left(1 - \frac{7}{8c_k}\right) \delta.
\]

Therefore, we know that for the Newton step, we have \(\bar{\Psi}(\bar{x}^{(\text{new})}, \bar{w}) - \bar{\Psi}(\bar{x}, \bar{w}) \in U\) where \(U\) is the symmetric convex set given by

\[
U \overset{\text{def}}{=} \{\bar{x} \in \mathbb{R}^n \mid \|\bar{x}\|_{\bar{w}+\infty} \leq C\}
\]

where \(C = \left(1 - \frac{7}{8c_k}\right) \delta\). Note that from our assumption on \(\delta\), we have

\[
C \leq \delta \leq \frac{K}{48c_k \log(400m)} = R.
\]

It ensures that \(U\) are contained in some \(\ell_\infty\) ball of radius \(R\). Therefore, we can play the chasing 0 game on \(\bar{\Psi}(\bar{x}, \bar{w})\) attempting to maintain the invariant that \(\|\bar{\Psi}(\bar{x}, \bar{w})\|_\infty \leq K \leq \frac{1}{2c_k}\) without taking steps that are more than \(1 + \epsilon\) times the size of \(U\) where we pick \(\epsilon = \frac{1}{2c_k}\) so to not interfere with our ability to decrease \(\delta_t\) linearly.
However, to do this with the chasing 0 game, we need to ensure that $R$ satisfying the following

\[
\frac{12R}{\varepsilon} \log \left( \frac{8m\tau}{\varepsilon} \right) \leq K
\]

where here $\tau$ is as defined in Theorem 11.

To bound $\tau$, we need to lower bound the radius of $\ell_\infty$ ball it contains. Since by assumption $\|\bar{g}(\bar{x})\|_\infty \leq 2$ and $\|\bar{\Psi}(\bar{x}, \bar{w})\|_\infty \leq \frac{1}{8}$, we have that $\|\bar{w}\|_\infty \leq 3$. Hence, we have

\[
\forall u \in \mathbb{R}^m : \|u\|_\infty \geq 1.3 \|u\|_2.
\]

Consequently, if $\|\bar{w}\|_\infty \leq \frac{\delta}{5c_{\text{norm}}\sqrt{m}}$, then $\bar{u} \in U$. So, we have that $U$ contains a box of radius $\frac{\delta}{5c_{\text{norm}}\sqrt{m}}$ and since $U$ is contained in a box of radius $\delta$, we have that

\[
\tau \leq 5C_{\text{norm}}\sqrt{m}
\]

\[
\leq 10c_k\sqrt{m}.
\]

Using $c_k \leq 24m^{1/4}$, we have

\[
\frac{12R}{\varepsilon} \log \left( \frac{8m\tau}{\varepsilon} \right) \leq 24c_k R \log \left( 160m^{3/2}c_k^2 \right)
\]

\[
\leq 48c_k R \log (400m) = K.
\]

and

\[
\frac{8m\tau}{\varepsilon} \leq 160m^{3/2}c_k^2 \leq 10^5 m^2.
\]

This proves that we meet the conditions of Theorem 11. Consequently, $\|\bar{\Psi}(\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})})\|_\infty \leq K$ and $\Phi_{\text{af}}(\bar{\Psi}(\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})})) \leq 10^5 m^2$.

Since $K \leq \frac{1}{4}$, Lemma 8 shows that

\[
\delta_t(\bar{x}^{(\text{new})}, \bar{w}) \leq 4 (\delta_t(\bar{x}, \bar{w}))^2.
\]

The step 5 shows that

\[
\left\| \log(\bar{w}) - \log(\bar{w}^{(\text{new})}) \right\|_{\bar{w}+\infty} \leq \left( 1 + \frac{1}{2c_k} \right) \left( 1 - \frac{7}{8c_k} \right) \delta
\]

\[
\leq \left( 1 - \frac{3}{8c_k} \right) \delta.
\]

Using $\delta \leq \frac{1}{80c_k}$, the Lemma 10 shows that

\[
\delta_t(\bar{x}^{(\text{new})}, \bar{w}^{(\text{new})}) \leq \left( 1 + 4 \left( 1 - \frac{3}{8c_k} \right) \delta \right) \left( \delta_t(\bar{x}^{(\text{new})}, \bar{w}) + \left( 1 - \frac{3}{8c_k} \right) \delta \right)
\]

\[
\leq (1 + 4\delta) \left( 4\delta^2 + \left( 1 - \frac{3}{8c_k} \right) \delta \right)
\]

\[
\leq \left( 1 - \frac{3}{8c_k} \right) \delta + 4\delta^2 + 16\delta^3 + 4\delta^2
\]

\[
\leq \left( 1 - \frac{1}{4c_k} \right) \delta.
\]
4.5 Weight Function

In this subsection we present the weight function that we use to achieve our $\tilde{O}(\sqrt{\text{rank}(A)L})$ linear program solver. The weight function is very similar to the one we used in [18] however there are subtle differences throughout the analysis so for completeness we repeat some of the analysis. For further intuition on the weight function or details in the proof we refer the reader to [18].

We define the weight function $\tilde{g} : \Omega^0 \to \mathbb{R}_{\geq 0}^m$ for all $\bar{x} \in \mathbb{R}_{\geq 0}^m$ as follows

$$
\tilde{g}(\bar{x}) \overset{\text{def}}{=} \arg \min_{\bar{w} \in \mathbb{R}_{\geq 0}^m} \hat{f}(\bar{x}, \bar{w}) \quad \text{where} \quad \hat{f}(\bar{x}, \bar{w}) \overset{\text{def}}{=} \bar{w}^T \bar{w} + \frac{1}{\alpha} \log \det (A_x^T \tilde{A}^{-\alpha} A_x) - \beta \sum_i \log w_i.
$$

where here and in the remainder of the subsection we let $A_x \overset{\text{def}}{=} (\Phi''(\bar{x}))^{-1/2} A$ and the parameters $\alpha, \beta$ are chosen later such that the following hold

$$
\alpha \in [1, 2) \quad , \quad \beta \in (0, 1) \quad , \text{and} \quad \beta^1 - \alpha \leq 2 \quad .
$$

(4.6)

Here we choose $\beta$ small and $\alpha$ just slightly larger than 1. Note that this formula is indeed different than the formula we used in [18], the sign of alpha has flipped and this causes minor changes throughout the analysis.

We start by computing the gradient and hessian of $\hat{f}(\bar{x}, \bar{w})$ with respect to $\bar{w}$.

**Lemma 13.** For all $\bar{x} \in \Omega^0$ and $\bar{w} \in \mathbb{R}_{\geq 0}^m$, we have

$$
\nabla_{\bar{w}} \hat{f}(\bar{x}, \bar{w}) = (I - \Sigma W^{-1} - \beta W^{-1}) \bar{I} \quad \text{and} \quad \nabla_{\bar{w} \bar{w}}^2 \hat{f}(\bar{x}, \bar{w}) = W^{-1} (\Sigma + \beta I + \alpha \Lambda) W^{-1}
$$

where $\Sigma \overset{\text{def}}{=} \Sigma_{A_x} (\bar{w}^{-\alpha})$ and $\Lambda \overset{\text{def}}{=} \Lambda_{A_x} (\bar{w}^{-\alpha})$.

**Proof.** Using Lemma 28 and the chain rule we compute the gradient of $\nabla_{\bar{w}} \hat{f}(\bar{x}, \bar{w})$ as follows

$$
\nabla_{\bar{w}} \hat{f}(\bar{x}, \bar{w}) = \bar{I} + \frac{1}{\alpha} \Sigma W^\alpha (-\alpha W^{-\alpha - 1}) - \beta W^{-1} \bar{I}
$$

$$
= (I - \Sigma W^{-1} - \beta W^{-1}) \bar{I}.
$$

Next, using Lemma 28 and chain rule, we compute the following for all $i, j \in [m]$

$$
\frac{\partial (\nabla_{\bar{w}} \hat{f}(\bar{x}, \bar{w}))}{\partial \bar{w}_j} = \frac{-\bar{w}_i \Lambda_{ij} \bar{w}_j}{\bar{w}_i^2} \left( -\alpha \bar{w}_j^{-\alpha - 1} \right) - \Sigma_{ij} 1_{i=j} - \beta 1_{i=j} \left\{ \bar{w}_i^{-2} \right\}
$$

$$
= \frac{\Sigma_{ij}}{\bar{w}_i \bar{w}_j} + \alpha \frac{\Lambda_{ij}}{\bar{w}_i \bar{w}_j} + \frac{\beta 1_{i=j}}{\bar{w}_i^2} \cdot (\text{Using that $\Sigma$ is diagonal}) \quad .
$$

Consequently, $\nabla_{\bar{w} \bar{w}}^2 \hat{f}(\bar{x}, \bar{w}) = W^{-1} (\Sigma + \beta I + \alpha \Lambda) W^{-1}$ as desired.  

**Lemma 14.** For all $\bar{x} \in \Omega^0$, the weight function $\tilde{g}(\bar{x})$ is a well defined weight function with

$$
\beta I \preceq G(\bar{x}) \preceq (1 + \beta) I \quad \text{and} \quad \|\tilde{g}(\bar{x})\|_1 = \text{rank}(A) + \beta \cdot m
$$

Furthermore, for all $\bar{x} \in \Omega^0$, the weight function obeys the following recurrences

$$
G(\bar{x}) = (\Sigma + \beta I) \bar{I} \quad , \quad G'(\bar{x}) = -G(\bar{x}) (G(\bar{x}) + \alpha \Lambda)^{-1} \Lambda \left( \Phi''(\bar{x}) \right)^{-1} \Phi'''(\bar{x})
$$

where $\Sigma \overset{\text{def}}{=} \Sigma_{A_x} (\tilde{g}^{-\alpha}(\bar{x}))$ and $\Lambda \overset{\text{def}}{=} \Lambda_{A_x} (\tilde{g}^{-\alpha}(\bar{x}))$. 

19
Proof. By Lemma 28 we have that $\Sigma \succeq A \succeq 0$. Therefore, by Lemma 13, we have that $\nabla^2_{w_1} \hat{f}(\vec{x}, \vec{w}) \succeq \beta I$ and $\hat{f}(\vec{x}, \vec{w})$ is strictly convex. Using the formula for the gradient in Lemma 13, we see that for all $i \in [m]$ it is the case that

$$\left[ \nabla_{w_i} \hat{f}(\vec{x}, \vec{w}) \right]_i = \frac{1}{\vec{w}_i} (\vec{w}_i - \Sigma iv - \beta).$$

Using that $0 \preceq \Sigma \preceq I$ (Lemma 28) and $\beta \in (0, 1)$ in (4.6), we see that if $\vec{w}_i \in (0, \beta)$ then $\left[ \nabla_{w_i} \hat{f}(\vec{x}, \vec{w}) \right]_i$ is strictly negative and if $\vec{w}_i \in (1 + \beta, \infty)$ then $\left[ \nabla_{w_i} \hat{f}(\vec{x}, \vec{w}) \right]_i$ is strictly positive. Therefore, for any $\vec{x} \in \Omega^0$, the minimizer of this strongly convex function $\hat{f}(\vec{x}, \vec{w})$ uniquely lies between the box between 0 and $\beta$ to 1 + $\beta$.

The formula for $G(\vec{x})$ follows by setting $\nabla_{w_i} \hat{f}(\vec{x}, \vec{w}) = 0$ and the size of $g(\vec{x})$ follows from the fact that $\|\Sigma \vec{x}\|_1 = \text{tr} (P_{A_x} (\vec{g}^{-\alpha}(\vec{x})))$. Since $P_{A_x} (\vec{g}^{-\alpha}(\vec{x}))$ is a projection onto the image of $G(\vec{x})^{-\alpha/2} A_x$ and since $\vec{g}(\vec{x}) > 0$ and $\vec{g}'(\vec{x}) > 0$, we have that the dimension of the image of $G(\vec{x})^{-\alpha/2} A_x$ is the rank of $A$. Hence, we have that $\|g(\vec{x})\|_1 = \text{rank}(A) + \beta \cdot m$.

By Lemma 28 and chain rule, we get the following for all $i, j \in [m]$

$$\frac{\partial (\nabla_{w_i} \hat{f}(\vec{x}, \vec{w}))}{\partial \vec{x}} = -\vec{w}_i^{-1} \Lambda \vec{w}_j \vec{g}'(\vec{x}) \left( (-\vec{g}'(\vec{x}))^{-2} \vec{g}''(\vec{x}) \right) = -\vec{w}_i^{-1} \Lambda \vec{w}_j \vec{g}'(\vec{x})^{-1} \vec{g}''(\vec{x})$$

Consequently, $J_{\vec{x}}(\nabla \hat{f}(\vec{x}, \vec{w})) = W^{-1} \Lambda (\Phi''(\vec{x}))^{-1} \Phi''(\vec{x})$. Since we have already know that $J_{\vec{w}}(\nabla \hat{f}(\vec{x}, \vec{w})) = \nabla^2 \hat{f}(\vec{x}, \vec{w}) = W^{-1} (\Sigma + \beta I + \alpha \Lambda) W^{-1}$ is positive definite (and hence invertible), by applying the implicit function theorem to the specification of $\vec{g}(\vec{x})$ as the solution to $\nabla_{w_i} \hat{f}(\vec{x}, \vec{w}) = 0$, we have

$$G'(\vec{x}) = -\left( J_{\vec{g}}(\nabla \hat{f}(\vec{x}, \vec{w})) \right)^{-1} \left( J_{\vec{x}}(\nabla \hat{f}(\vec{x}, \vec{w})) \right) = -G(\vec{x}) (G(\vec{x}) + \alpha \Lambda)^{-1} \Lambda (\Phi''(\vec{x}))^{-1} \Phi''(\vec{x}) .$$

Now we show the consistency

Lemma 15. [Consistency] For all $\vec{x} \in \Omega^0$ and $\vec{g} \in \mathbb{R}^m$, and

$$B \eqdef G(\vec{x})^{-1} G'(\vec{x})(\vec{g}(\vec{x}))^{-1/2},$$

we have

$$\|B \vec{g}\|_{G(\vec{x})} \leq \frac{2}{1 + \alpha} \|\vec{g}\|_{G(\vec{x})} \quad \text{and} \quad \|B \vec{g}\|_{\infty} \leq \frac{2}{1 + \alpha} \left( \|\vec{g}\|_{\infty} + \frac{2 + 2\alpha}{1 + \alpha} \|\vec{g}\|_{G(\vec{x})} \right) .$$

Therefore

$$\|B\|_{\vec{g} + \infty} \leq \frac{2}{1 + \alpha} \left( 1 + \frac{2}{C_{\text{norm}}} \right)$$

Proof. Fix an arbitrary $\vec{x} \in \Omega^0$ and let $\vec{g} \eqdef \vec{g}(\vec{x})$, $\vec{g} \eqdef \vec{g}(\vec{x})$, $\Sigma \eqdef \Sigma_{A_x} (\vec{g}^{-\alpha}(\vec{x}))$, $\Pi \eqdef \Pi_{A_x} (\vec{g}^{-\alpha}(\vec{x}))$, $\Lambda \eqdef \Lambda_{A_x} (\vec{g}^{-\alpha}(\vec{x}))$. Also, fix an arbitrary $\vec{g} \in \mathbb{R}^m$ and let $\vec{z} \eqdef B \vec{g}$. 

20
By Lemma 14, $G' = -G(G + \alpha \Lambda)^{-1} \Lambda (\Phi''')^{-1} \Phi'''$ and therefore

$$B = -G^{-1} \left( G(G + \alpha \Lambda)^{-1} \Lambda (\Phi''')^{-1} \Phi'' \right) (\Phi')^{-1/2}$$

$$= (G + \alpha \Lambda)^{-1} (2 \Lambda) \text{diag} \left( \frac{-\Phi'''}{2(\Phi'^3/2)} \right).$$

Let $C \equiv (G + \alpha \Lambda)^{-1} (2 \Lambda)$ and let $\vec{y}' \equiv \text{diag} \left( \frac{-\Phi'''}{2(\Phi'^3/2)} \right) \vec{y}$. By the self concordance of $\Phi$ (Definition 2) we know that $\|\vec{y}'\| \leq \|\vec{y}\|$ for both norms in the lemma statement. Since $\vec{z} = B\vec{y} = C\vec{y}'$, it suffices to bound $\|\vec{C}\vec{y}'\|$ in terms of $\|\vec{y}'\|$ for the necessary norms.

Letting $\Lambda \equiv G^{-1/2} \Lambda G^{-1/2}$, we simplify the equation further and note that

$$\|\vec{C}\|_G = \|G^{1/2} (G + \alpha \Lambda)^{-1} (2 \Lambda) G^{-1/2}\|_2 = \| \left( I + \alpha \Lambda \right)^{-1} (2 \Lambda) \|_2.$$

Now, for any eigenvector, $\vec{v}$, of $\Lambda$ with eigenvalue $\lambda$, we see that $\vec{v}$ is an eigenvector of $(I + \alpha \Lambda)^{-1}(2 \Lambda)$ with eigenvalue $2\lambda/(1 + \alpha \lambda)$. Furthermore, since $0 \leq \Lambda \leq I$, we have that $\|\vec{C}\|_G \leq 2/(1 + \alpha)$ and hence $\|\vec{z}\|_G \leq 2(1 + \alpha)^{-1}\|\vec{y}'\|_G \leq 2(1 + \alpha)^{-1}\|\vec{y}\|_G$ as desired.

Next, to bound $\|\vec{z}\|_\infty$, we use that $(G + \alpha \Lambda)\vec{z} = 2\Lambda \vec{y}'$ and since $\Lambda = \Sigma - P^{(2)}$ and $G = \Sigma + \beta I$, we have

$$(1 + \alpha) \Sigma \vec{z} + \beta \vec{z} - \alpha P^{(2)} \vec{z} = 2 \Sigma \vec{y}' - 2 P^{(2)} \vec{y}'.$$}

Looking at the $i^{th}$ coordinate of both sides and using that $\sigma_i \geq 0$, we have

$$((1 + \alpha)\sigma_i + \beta) |\vec{z}_i| \leq \alpha \left| [P^{(2)} \vec{z}][i] + 2\sigma_i |\vec{y}'|_\infty + 2 \left| [P^{(2)} \vec{y}'][i] \right| \right| \leq \alpha \sigma_i ||\vec{z}\|_\Sigma + 2\sigma_i |\vec{y}'|_\infty + 2\sigma_i ||\vec{y}'\|_\Sigma \quad \text{(Lemma 28)}$$

$$(\Sigma \leq G \text{ and } ||\vec{z}\|_G \leq 2(1 + \alpha)^{-1} ||\vec{y}'\|_G$$)

Hence, we have

$$|\vec{z}_i| \leq \frac{2}{1 + \alpha} ||\vec{y}'\|_\infty + \frac{1}{1 + \alpha} \left( \frac{2\alpha}{1 + \alpha} + 2 \right) ||\vec{y}'\|_G$$

$$\leq \frac{2}{1 + \alpha} \left[ ||\vec{y}'\|_\infty + 2 ||\vec{y}'\|_G \right].$$

Therefore, $\|B\vec{y}\|_\infty = ||\vec{z}\|_\infty \leq 2(1 + \alpha)^{-1}(||\vec{y}'\|_\infty + 2 ||\vec{y}'\|_G).$ To get the final inequality, we note that

$$\|B\vec{y}\|_{\vec{y} + \infty} = \|B\vec{y}\|_\infty + C \|B\vec{y}\|_G$$

$$\leq \frac{2}{1 + \alpha} ||\vec{y}'\|_\infty + \frac{2}{1 + \alpha} \cdot 2 ||\vec{y}'\|_G + \frac{2}{1 + \alpha} C \|\vec{y}'\|_G$$

$$\leq \frac{2}{1 + \alpha} \left( 1 + \frac{2}{C} \right) ||\vec{y}'\|_{\vec{y} + \infty}.$$

$\Box$
Theorem 16. Choosing parameters

\[ \alpha = 1 + \frac{1}{\log_2 \left( \frac{2m}{\text{rank}(A)} \right)} , \quad \beta = \frac{\text{rank}(A)}{2m} , \text{ and } \quad C_{\text{norm}} = 18 \log_2 \left( \frac{2m}{\text{rank}(A)} \right) \]

yields

\[ c_1(\bar{g}) \leq 2 \text{rank}(A) \quad , \quad c_\gamma(\bar{g}) \leq 1 + \frac{1}{9 \log_2 \left( \frac{2m}{\text{rank}(A)} \right)} \quad , \quad \text{and} \quad c_\delta(\bar{g}) \leq 1 - \frac{2}{9 \log_2 \left( \frac{2m}{\text{rank}(A)} \right)} . \]

In particular, we have

\[ c_\gamma(\bar{g}) c_\delta(\bar{g}) \leq 1 - \frac{1}{9 \log_2 \left( \frac{2m}{\text{rank}(A)} \right)} . \]

Proof. The bounds on \( c_1(\bar{g}) \) and \( c_\delta(\bar{g}) \) follow immediately from Lemma 14 and Lemma 15. Now, we estimate the \( c_\gamma(\bar{g}) \) and let \( \frac{5}{4} \bar{g} \leq \bar{w} \leq \frac{5}{4} \bar{g} \). Fix an arbitrary \( \bar{x} \in \Omega^0 \) and let \( \bar{g} \equiv \bar{g} \). Recall that by Lemma 14, we have \( \bar{g} \geq \beta \). Furthermore, since \( \bar{g}^{-1} = \bar{g}^{\alpha-1} \bar{g}^{-\alpha} \) and \( \beta^{\alpha-1} \geq \frac{1}{4} \), the following holds

\[ 4 \bar{g}^{-\alpha} \leq \frac{4}{5} \beta^{\alpha-1} \bar{g}^{-\alpha} \leq 4 \bar{g}^{-1} \leq \bar{w}^{-1} \quad (4.7) \]

Applying this and using the definition of \( P_{A_\bar{x}} \) yields

\[ A_\bar{x}(A_\bar{x}^T W^{-1} A_\bar{x})^{-1} A_\bar{x}^T \leq \frac{10}{4} A_\bar{x}(A_\bar{x}^T G^{-\alpha} A_\bar{x})^{-1} A_\bar{x}^T = \frac{10}{4} G^{\alpha/2} P_{A_\bar{x}}(\bar{g}^{-\alpha}) G^{\alpha/2} . \quad (4.8) \]

Hence, we have

\[ \bar{\sigma}_i \left( \frac{1}{w_{i'}} \right) = \bar{\sigma}_i \left( \frac{1}{w_{i'}} \right) \leq \frac{10}{4} \bar{g}^{-\alpha} \bar{\sigma}_i \left( \frac{1}{w_{i'}} \right) \leq \frac{10}{4} \left( \frac{5}{4} \right) ^2 \bar{\sigma}_i \left( \frac{1}{w_{i'}} \right) \leq 4. \]

Since \( P_{\bar{x},\bar{w}} \) is an orthogonal projection in \( \| \cdot \|_{\bar{w}} \), we have \( \| P_{\bar{x},\bar{w}} \|_{\bar{w} \to \bar{w}} = 1 \). Let \( P_{\bar{x},\bar{w}} = I - P_{\bar{x},\bar{w}} \), we have

\[ \| P_{\bar{x},\bar{w}} \|_{\bar{w} \to \infty} = \max_i \| \bar{g} \|_{\bar{w}} \leq 1 \]

\[ \leq \max_i \| (\bar{w})^{-1/2} P_{\bar{x},\bar{w}} \|_{\bar{w} \to \bar{w}}^2 \]

\[ = \max_i \bar{\sigma}_i \left( \frac{1}{w_{i'}} \right) \leq 2. \]
For any \( \tilde{y} \), we have
\[
\| P \tilde{x}, w \tilde{y} \|_\infty + C_{\text{norm}} \| P \tilde{x}, w \tilde{y} \|_1 \\
\leq \| \tilde{y} \|_\infty + (2 + C_{\text{norm}}) \| \tilde{y} \|_1 \\
\leq \frac{C_{\text{norm}} + 2}{C_{\text{norm}}} \| \tilde{y} \|_1.
\]

Hence, we have \( c_\gamma \leq \frac{C_{\text{norm}} + 2}{C_{\text{norm}}} \). Thus, we have picked \( C_{\text{norm}} = \frac{18}{\alpha - 1} \) and have \( c_\gamma \leq 1 + \frac{\alpha - 1}{9} \).

### 4.6 Computing Weight Function

\[
\tilde{w} = \text{computeWeight}(\tilde{x}, \tilde{w}, K)
\]

1. Let \( \alpha = 1 + \frac{1}{\log_2 \left( \frac{2m}{\text{rank}(A)} \right)} \), \( \beta = \frac{\text{rank}(A)}{2m} \), \( k = 10^k \log^2 (4m/K) \log(m) \).

2. For \( j = 1, \ldots, 20 \log (2m/K) \)
   1a. Let \( \bar{c} = \left( \tilde{w}_j \right)^{-1} \).
   1b. Let \( \bar{q}_j \) be \( k \) random \( \pm 1/\sqrt{k} \) vectors of length \( n \).
   1c. Compute \( \bar{\sigma}_t = \sum_{j=1}^k \left( \frac{1}{2} C^{1/2} A (A^T C A)^{-1} A^T C^{1/2} \bar{q}_j \right) \).
   1d. \( \tilde{w}_{j+1} = \text{med} \left( (1 - \frac{1}{13}) \tilde{w}_0, \frac{3}{4} \bar{w}_j + \frac{1}{4} \bar{\sigma}_t + \frac{1}{4} (1 + \frac{1}{13}) \tilde{w}_0 \right) \).

3. Output \( \tilde{w}_{\text{last}} \).

**Theorem 17.** \([\text{Approximate Weight}] \) Suppose \( \| W_0^{-1} (G(\tilde{x}) - \tilde{W}_0) \|_\infty \leq \frac{1}{13} \) and \( K < 1 \). The algorithm \( \text{computeWeight} \) return \( \tilde{w} \) such that
\[
\| G(\tilde{x})^{-1} (G(\tilde{x}) - \tilde{W}) \|_\infty \leq K
\]
using only \( O(\log(m) \log^3 (m/K) / K^2) \) times linear system solves.

**Proof.** The proof is basically the same as \([18]\). However, since \( \bar{g} \) is strongly convex compared to the corresponding \( \tilde{g} \) in that paper, the gradient descent converges faster.

### 5 The Linear Programming Method

In this section we show how to use the centering procedure and weight analysis of the previous section to solve (3.1). First we bound the difference in cost of a central path point to the cost of the optimal path point using a proof adapted from \([22, \text{Thm 4.2.7}]\). This allows us to reason about our distance from the optimal point after multiple iterations of our path following procedure.

**Lemma 18.** \([22, \text{Thm 4.2.7}]\)Let \( x^* \) denote an optimal solution to 3.1 and \( \tilde{x}_t = \arg \min f_t (\tilde{x}, \tilde{w}) \). We have that for any \( \{ \tilde{x}, \tilde{w} \} \in \{ \Omega^b \times \mathbb{R}^m \} \)
\[
\bar{c}^T \tilde{x}_t (\tilde{w}) - \bar{c}^T \tilde{x}^* \leq \frac{\| \tilde{w} \|_1}{t}.
\]

23
Proof. We note that by the optimality conditions we know that \( \nabla_x f_t(\vec{x}_t(\vec{w})) = t \cdot \vec{c} + \vec{w} \vec{\phi}'(\vec{x}_t(\vec{w})) \) is orthogonal to the kernel of \( A^T \) furthermore since \( \vec{x}_t(\vec{w}) - \vec{x}^* \in \text{ker}(A^T) \) we have that 

\[
(t \cdot \vec{c} + \vec{w} \vec{\phi}'(\vec{x}_t(\vec{w})))^T (\vec{x}_t(\vec{w}) - \vec{x}^*) = 0.
\]

Using that \( \phi'_i(x_t(\vec{w})_i) \cdot (x^*_i - x_t(\vec{w})_i) \leq 1 \) by Lemma 4 we then have 

\[
\frac{\vec{c}}{t}(\vec{x}_t(\vec{w}) - \vec{x}^*) = \frac{1}{t} \left( \vec{w} \vec{\phi}'(\vec{x}_t(\vec{w})) \right)^T (\vec{x}^* - \vec{x}_t(\vec{w}))
= \frac{1}{t} \sum_{i \in m} w_i \cdot \phi'_i(x_t(\vec{w})_i) \cdot (x^*_i - x_t(\vec{w})_i)
\leq \frac{\|\vec{w}\|_1}{t}.
\] (5.1)

Next we bound how close the weighted central path can go to the boundary of the polytope. This allows us to reason about how ill-conditioned the linear system we need to solve become over the course of the algorithm.

**Lemma 19.** Let \( \vec{x}_t = \arg \min f_t(\vec{x}, \vec{w}) \). Assume optimum solution(s) exists and are all on the boundary. Let \( 0 < a < b \). Let \( s_i(\vec{x}) \) be the slack of \( \vec{x} \) for the \( i \)th constraint. (For bounded inequality, it holds for either side). Then, we have 

\[
s_i(\vec{x}_b) \geq \left( 1 + \frac{b - a}{a} \frac{\|\vec{w}\|_1}{\min \vec{w}} \right)^{-1} s_i(\vec{x}_a).
\]

**Proof.** Consider the straight line from \( \vec{x}_a \) to \( \vec{x}_b \). It must hit the boundary of the polytope at some point otherwise there is an optimum solution in the interior or the optimum value is unbounded. Now, we parametrize the straight line \( \vec{p}(t) \) from \( \vec{x}_a \) to \( \vec{x}_b \) such that \( \vec{p}(1) = \vec{x}_a \), \( \vec{p}(\theta) = \vec{x}_b \) for some \( \theta \) and \( \vec{p}(0) \) on the boundary of the polytope. The line is parametrize by same speed, hence \( \frac{d\vec{p}}{dt} \) is a constant vector. Without loss of generality, we assume \( \frac{d\vec{p}}{dt} \) is an unit vector.

Let \( \phi(t) = \sum \vec{w}_i \phi_i(\vec{p}(t)) \). Then, the optimality of \( \vec{x}_a \) shows that \( a\vec{c}^T \frac{d\vec{p}}{dt} + \frac{d\phi}{dt} \bigg|_{t=1} = 0 \). Since \( \vec{p}(t) \) is feasible from 0 to 1, Lemma 4 shows that 

\[
\frac{d\phi_i}{dt} \bigg|_{x=\vec{p}(t)} \geq \frac{-1}{t}.
\]

Therefore, \( \frac{d\phi}{dt} \bigg|_{t=1} \geq -\|\vec{w}\|_1 \). Therefore, we have \( a\vec{c}^T \frac{d\vec{p}}{dt} \leq \|\vec{w}\|_1 \). Without loss of generality, assume the term \( \phi_1(p(t)) \to \infty \) as \( t \to 0 \). Lemma 3 shows that 

\[
\frac{d^2\phi_i}{dt^2} \bigg|_{x=\vec{p}(t)} \geq \frac{1}{t^2}.
\]

24
Integrating it, we have

\[
\frac{d\phi_i}{dt} \bigg|_{x=\bar{p}_i(t)} \leq \frac{d\phi_i}{dt} \bigg|_{x=\bar{p}_i(1)} - \int_t^1 \frac{1}{t^2} dt \\
\leq \frac{d\phi_i}{dt} \bigg|_{x=\bar{p}_i(1)} - \frac{1}{t} + 1.
\]

Since each term of $\phi$ is convex, we have

\[
\frac{d\phi}{dt} \bigg|_t \leq \frac{d\phi}{dt} \bigg|_{t=1} - \min \bar{w} \left( \frac{1}{t} - 1 \right) \\
= -ac^T \frac{dp}{dt} - \min \bar{w} \left( \frac{1}{t} - 1 \right).
\]

Since $bc^T \frac{dp}{dt} + \frac{d\phi}{dt} \bigg|_{t=\theta} = 0$, we have

\[
bc^T \frac{dp}{dt} = - \frac{d\phi}{dt} \bigg|_{t=\theta} \geq \min \bar{w} \left( \frac{1}{\theta} - 1 \right) + ac^T \frac{dp}{dt}.
\]

Thus, we have

\[
\frac{b-a}{a} \left\| \bar{w} \right\|_1 \geq (b-a) c^T \frac{dp}{dt} \geq \min \bar{w} \left( \frac{1}{\theta} - 1 \right).
\]

Next we bound how our distance to the weighted central path in terms of the centrality quantity.

**Lemma 20.** Suppose $\delta_t(\bar{x}(1), \bar{g}(\bar{x}(1))) \leq \frac{1}{800c_k \log(400m)}$. Let $\bar{x}_t = \arg \min f_t(\bar{x}, \bar{w})$. Then, we have

\[
\left\| \sqrt{\phi''(\bar{x}_t)} (\bar{x}(1) - \bar{x}_t) \right\|_\infty \leq 16c_k c \delta_t(\bar{x}(1), \bar{g}(\bar{x}(1))).
\]

**Proof.** We use Theorem 12 with exact weight computation and starts with $\bar{x}(1)$. The iteration $\bar{x}(k)$ decrease $\delta_t$ by factor of $\left(1 - \frac{1}{4c_k}\right)$. (3.4) shows that $\left\| \sqrt{\phi''(\bar{x}(k))} (\bar{x}(k+1) - \bar{x}(k)) \right\|_\infty \leq c \delta_t(\bar{x}(k), \bar{g}(\bar{x}(k)))$. The Lemma 3 shows that

\[
\left\| \log (\sqrt{\phi''(\bar{x}(k))}) - \log (\sqrt{\phi''(\bar{x}(k+1))}) \right\|_\infty \leq \left(1 - 2c \delta_t(\bar{x}(k), \bar{g}(\bar{x}(k)))\right)^{-1} \leq e^{4c \delta_t(\bar{x}(k), \bar{g}(\bar{x}(k)))}.
\]

Therefore, for any $k$, we have

\[
\left\| \log (\sqrt{\phi''(\bar{x}(1))}) - \log (\sqrt{\phi''(\bar{x}(1))}) \right\|_\infty \leq e^{4c \sum \delta_t(\bar{x}(k), \bar{g}(\bar{x}(k)))} \leq e^{32c_k c \delta_t(\bar{x}(1), \bar{g}(\bar{x}(1)))} \leq 2.
\]

25
Hence, for any \(k\), we have
\[
\left\| \sqrt{\phi''(\vec{x}_t)} \left( \vec{x}^{(1)} - \vec{x}^{(k)} \right) \right\|_\infty \leq \sum 2c_s \delta_t(\vec{x}^{(k)}, \vec{g}(\vec{x}^{(k)})) \\
\leq 16c_k \delta_t(\vec{x}^{(1)}, \vec{g}(\vec{x}^{(1)})).
\]

It is clear now \(\vec{x}^{(k)}\) forms a Cauchy sequence and converges to \(\vec{x}_t\) because \(\delta_t\) continuous and \(\vec{x}_t\) is the unique point such that \(\delta_t = 0\).

Here, we show how to put together the results of the previous section and analyze the running time of our linear program solver.

| Step | Description |
|------|-------------|
| 1.   | \(c_k = 9 \log_2 \left( 2m / \text{rank} \right), t = t_{\text{start}}, K = \frac{1}{20c_k} \). |
| 2.   | While \((t < t_{\text{end}} \text{ if } t_{\text{start}} < t_{\text{end}}) \text{ or } (t > t_{\text{end}} \text{ if } t_{\text{start}} > t_{\text{end}})\) |
| 2a.  | \((\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) = \text{centeringInexact}(\vec{x}, \vec{w}, K)\) |
| 2b.  | \(t \leftarrow t \left( 1 \pm \frac{10c_k^2 \log \left( 400m \sqrt{\text{rank}} \right) }{t_{\text{end}} - t_{\text{start}}} \right) \text{ where the sign is the sign of } t_{\text{end}} - t_{\text{start}}\) |
| 2c.  | \(\vec{x} \leftarrow \vec{x}^{\text{final}}, \vec{w} \leftarrow \vec{w}^{\text{new}}.\) |
| 3.   | Repeat \(4c_k \log(1/\epsilon)\) times |
| 3a.  | \((\vec{x}, \vec{w}) = \text{centeringInexact}(\vec{x}, \vec{w}, K)\) |
|      | where it use the function \(\text{computeWeight}\) to find the approximation of \(\vec{g}(\vec{x})\). |
| 4.   | Output \((\vec{x}, \vec{w})\). |

**Theorem 21.** Using the notation in \(\text{pathFollowing}\). We have
\[
\delta_{t_{\text{start}}}(\vec{x}, \vec{w}) \leq \frac{1}{960c_k^2 \log (400m)} \quad \text{and} \quad \Phi_{\alpha}(\vec{\Psi}(\vec{x}, \vec{w})) \leq 10^5 m^2.
\]

Let \((\vec{x}^{\text{final}}, \vec{w}^{\text{new}}) = \text{pathFollowing}(\vec{x}, \vec{w}, t_{\text{start}}, t_{\text{end}})\), then
\[
\delta_{t_{\text{end}}}(\vec{x}^{\text{final}}, \vec{w}^{\text{new}}) \leq \epsilon \quad \text{and} \quad \Phi_{\alpha}(\vec{\Psi}(\vec{x}^{\text{final}}, \vec{w}^{\text{new}})) \leq 10^5 m^2.
\]

Also, it takes \(\tilde{O} \left( \sqrt{\text{rank}} \left( \log \left( \frac{t_{\text{end}}}{t_{\text{start}}} \right) + \log(1/\epsilon) \right) (T + m) \right)\) where \(T\) is the time to compute one linear system solve.

In particular, it implies there is an \(\tilde{O} \left( \sqrt{\text{rank}} \right) \) iterations algorithm for solving linear programming problems of the form
\[
\min \vec{c}^T \vec{x} \text{ given } \vec{A}^T \vec{x} = \vec{b} \text{ and } l_i \leq x_i \leq u_i
\]
where \(-\infty \leq l_i \leq u_i \leq +\infty\) and each iterations involves solving \(\tilde{O}(1)\) linear systems to \(O(L)\) bit accuracy.

**Proof.** This algorithm maintain the invariant that \(\delta_t(\vec{x}, \vec{w}) \leq \frac{1}{960c_k^2 \log (400m)}\) and \(\Phi_{\alpha}(\vec{\Psi}(\vec{x}, \vec{w})) \leq 10^5 m^2\) on each iteration in the beginning of the step (2a). Theorem 12 shows that
\[
\left\| \log(\vec{g}(\vec{x}^{\text{new}})) - \log(\vec{w}) \right\|_\infty \leq K \leq \frac{1}{20c_k}.
\]
Thus, it satisfies the condition of Theorem 17. Hence, the algorithm centerInexact can use the function computeWeight to find the approximation of \( \tilde{g}(\bar{x}^{\text{new}}) \). Hence, we have

\[
\delta_t(\bar{x}^{\text{(final)}}, \bar{w}^{\text{(new)}}) \leq \left( 1 - \frac{1}{4c_k} \right) \delta_t \quad \text{and} \quad \Phi_\alpha(\tilde{\Psi}(\bar{x}^{\text{(final)}}, \bar{w}^{\text{(new)}})) \leq 10^5 m^2.
\]

Using Lemma 7, we have

\[
\delta_t(\bar{x}^{\text{(final)}}, \bar{w}^{\text{(new)}}) \leq \frac{1}{960c_k^2 \log (400m)}.
\]

Hence, we proved that for every step (2c), we have the invariant. The \( \delta_t < \epsilon \) bounds follows from the last loop.

### Table 1

| \( \bar{x}^{\text{(final)}} \) = LPSolve(\( \bar{x} \), \( \epsilon \)) |
|---|
| 1. \( \bar{w} = \tilde{\sigma}_A \left( (\tilde{d}'(\bar{x}))^{-1} \right) + \beta \tilde{1} \), \( d = -\bar{w}_i \phi_i(\bar{x}) \). |
| 2. Rescale the linear programs by \( A = \bar{w}^{\alpha/2} A \). |
| 3. \( t_1 = (10^4 U^2 m^6)^{-1}, t_2 = 1/\epsilon, \epsilon_1 = \frac{1}{2000c_k^2 \log (400m)}, \epsilon_2 = \frac{\epsilon}{10^3 c_k^2 m^4 U^2} \). |
| 4. \((\bar{x}^{\text{(new)}}, \bar{w}^{\text{(new)}}) = \text{pathFollowing}(\bar{x}, \bar{w}, 1, t_1, t_2, \epsilon_1)\) on the rescaled LP with cost vector \( d \). |
| 5.\((\bar{x}^{\text{(final)}}, \bar{w}^{\text{(final)}}) = \text{pathFollowing}(\bar{x}, \bar{w}, 1, t_1, t_2, \epsilon_2)\) on the rescaled LP with cost vector \( \bar{c} \). |
| 6. Output \( \bar{x}^{\text{final}} \). |

In this master theorem, we will explain how this algorithm can works using inexact linear solves.

**Theorem 22.** Suppose we have an interior point \( \bar{x}_0 \) for the linear programs

\[
\min c^T \bar{x} \text{ given } A^T \bar{x} = \bar{b} \text{ and } l_i \leq x_i \leq u_i.
\]

Let \( U > 1 \) be such that \( \| \max \left( \frac{\bar{u} - \bar{I}}{\bar{u} - \bar{I}}, \frac{\bar{I} - \bar{l}}{\bar{I} - \bar{l}} \right) \|_\infty \leq U, \| \bar{u} - \bar{l} \|_\infty \leq U, \| \bar{c} \|_\infty \leq U \). Then, the algorithm LPSolve outputs \( \bar{x} \) such that \( \bar{c}^T \bar{x} \leq \text{OPT} + \epsilon \) in \( \tilde{O} \left( \sqrt{\text{rank}(A)} \log (U/\epsilon) (T + m) \right) \) where \( T \) is the time to compute one linear system solve up to \( O \left( \log (mU \kappa/\epsilon) \right) \) bit accuracy and \( \kappa \) is the condition number of \( A^T A \). The linear systems solved in time \( T \) are of the form \( A^T S A \) where the condition number of \( S \) is \( \text{poly}(mU/\epsilon) \).

**Proof.** Using \( \bar{w} = \tilde{\sigma}_A \left( (\tilde{d}'(\bar{x}))^{-1} \right) + \beta \tilde{1} \) and \( \bar{A} = \bar{w}^{\alpha/2} A \), we have

\[
\bar{w} = \tilde{\sigma}_A \left( (\tilde{d}'(\bar{x}))^{-1} \right) + \beta \bar{1} = \tilde{\sigma}_A \left( (\tilde{d}'(\bar{x}) \tilde{w}^\alpha)^{-1} \right) + \beta \bar{1}.
\]

Therefore, we have \( \tilde{g}(\bar{x}) = \bar{w} \). By the definition of \( \bar{d} \), we have \( \bar{x} \) is the minimum of

\[
\min \bar{d}^T \bar{x} - \sum \bar{w}_i \phi_i(\bar{x}) \text{ given } A^T \bar{x} = \bar{b}.
\]

\[10\text{The linear systems that need to be solved in time } T \text{ have condition number } O(\text{poly}(\kappa, m, \epsilon^{-1})) \text{ and therefore most notions of precision for linear system solvers are equivalent up to a } O(\text{poly}(\kappa, m)) \text{ factor.}\]
Therefore, \( (\vec{x}_0, \vec{w}_0) \) satisfies the assumption of theorem 21 because \( \delta_t = 0 \) and \( \Phi_{\alpha} = 0 \). Hence, we have

\[
\delta_{t_{\text{end}}} (\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) \leq \frac{1}{2000\sqrt{k} \log (400m)} \quad \text{and} \quad \Phi_{\alpha}(\vec{y}(\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})})) \leq 10^5 m^2.
\]

Since we rescale the system, note that all size guarantee become at worst \( m^2 U \). Since \( \| \max \left( \frac{u-l}{u-x}, \frac{u-l}{x-l} \right) \|_\infty \leq m^2 U \), Lemma 4 shows that \( \| \phi(t) \|_\infty \leq m^2 U \) and hence \( \| \bar{c} - \bar{d} \|_\infty \leq 2m^2 U \). Also, since \( \| u-l \|_\infty \leq m^2 U \), Lemma 3 shows that \( \sqrt{\phi''(\bar{x})} \geq \frac{1}{m^2 U} \). Therefore, we have

\[
\delta_{t_1} (\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) \quad \text{def} \quad \left\| \frac{t_1 \bar{c} + \vec{w} \phi'(\bar{x})}{\left( \frac{1}{m^2 U} \right)} \right\|_{Q \min(\bar{x}, \vec{w})} \\
\leq \left\| \frac{t_1 \bar{d} + \vec{w} \phi'(\bar{x})}{\left( \frac{1}{m^2 U} \right)} \right\|_{Q \min(\bar{x}, \vec{w})} + t_1 \left\| \frac{\bar{c} - \bar{d}}{\left( \frac{1}{m^2 U} \right)} \right\|_{\vec{w} \infty} \\
= \delta_{t_1} (\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) + 2m^5 U^2 t_1 \left\| \bar{x} \right\|_{\vec{w} \infty} + 2000 \sqrt{k} \log (400m).
\]

Since we have \( t_1 \) small enough, we have \( \delta_{t_1} (\vec{x}^{(\text{new})}, \vec{w}^{(\text{new})}) \) is still very small. So, we only need to prove how large \( t_2 \) should be and how small \( \epsilon_2 \) should be in order to get \( \vec{x} \) such that \( \epsilon^T \vec{x} \leq \text{OPT} + \epsilon \). By Lemma 18, we have

\[
\epsilon^T \vec{x}_{t_2} \leq \text{OPT} + \frac{3m}{t_2}.
\]

Also, Lemma 20 shows that we have

\[
\left\| \sqrt{\phi''(\vec{x}_{t_2})} (\vec{x}^{(\text{final})} - \vec{x}_{t_2}) \right\|_\infty \leq 32 \epsilon_2 c_k.
\]

Using \( \sqrt{\phi''(\vec{x})} \geq \frac{1}{m^2 U} \), we have \( \| (\vec{x}^{(\text{final})} - \vec{x}_{t_2}) \|_\infty \leq 32 \epsilon_2 c_k m^2 U \) and hence our choice of \( t_2 \) and \( \epsilon_2 \) gives the result

\[
\epsilon^T \vec{x}^{(\text{final})} \leq \text{OPT} + \frac{3m}{t_2} + 32 \epsilon_2 c_k m^4 U^2 \leq \text{OPT} + \epsilon.
\]

Since the convergence of our algorithm depends only on the invariant of \( \delta_t \) and \( \Phi \) and the invariant that \( A^T \vec{x} = \vec{b} \), as long as the inexact linear solver does not break this invariant, it is stable. For the first two invariants, we can prove it does not increase \( \delta_t \) and \( \Phi \) too much by triangle inequality. To do this, we only need to prove that \( \vec{w} \phi'' \) is upper and lower bounded by polynomial of \( m \), \( U \) and \( 1/\epsilon \). For \( \vec{w} \), we know that \( m^{-1} \leq \vec{w} \leq 2 \). For \( \phi'' \), Lemma 19 shows that the distance of \( \vec{x} \) to the boundary of polytope is lower bounded by polynomial of \( 1/m \), \( 1/U \) and \( \epsilon \). Hence it is easy to see from Lemma 4 that \( \phi'' \) is not too large, otherwise, we can use it to bound \( \phi' \). On the other hand, \( \phi'' \) is not too small because of the size of the domain \( \| \vec{u} - \vec{l} \|_\infty \leq U \). Therefore, we have the linear system is never too ill-conditioned compared to
the original system. Therefore, the accuracy we needed is only $O(\log(\eta U \kappa / \epsilon))$ bits where $\kappa$ is the condition number of the original system. To keep the invariant that $A^T \tilde{x} = \tilde{b}$, the last step of centeringInexact projects the $\tilde{x}$ back to $A^T \bar{x} = \bar{b}$. This step would $\delta_t$ and $\Phi$ a little bit, but since we do it every iteration, the distance of $A^T \tilde{x}$ and $\bar{b}$ is never too large and hence it will not affects the stability of the solver.

6 Applications

6.1 Generalized Minimum Cost Flow

Our algorithm for the generalized minimum cost flow problem is essentially the same as our algorithm for the simpler special case of maximum flow same.\footnote{The algorithm could be simplified slightly for the maximum flow case and the dependence on polylogarithmic factors could possibly be improved. However, we are making no attempt to control the dependence on polylogarithmic factors in this paper.} Therefore, we present the algorithm for the generalized minimum cost flow problem directly.

The generalized minimum cost flow problem [5] is as follows. Let $G = (V, E)$ be a directed graph where each edge $e$ has capacity $c_e > 0$ multiplier $1 \geq \gamma_e > 0$. For each edge $e$, there can be only at most $c_e$ units of flow on that edge and the flow on that edge must be non-negative. Also, for each unit of flow entering edge $e$, there are only $\gamma_e$ units of flow going out. The generalized maximum flow problem is to compute how much flow can be sent into $t$ given a unlimited source $s$. The generalized minimum cost flow is to ask what is the minimum cost of sending the maximum flow given the cost of each edge is $q_e$. The maximum flow and the minimum cost flow are the case with $\gamma_e = 1$ for all edges $e$.

Since the generalized minimum cost flow includes all of these cases, we focus on this general formulation. The problem can be written as the following linear programs

$$\min_{0 \leq \bar{x} \leq \bar{c}} q^T \bar{x} \text{ such that } A\bar{x} = F\bar{1}_t$$

where $F$ is the generalized maximum flow value, $\bar{1}_t$ is an indicator vector of size $(n - 1)$ that is non-zero at vertices $t$ and $A$ is a $|V\setminus\{s\}| \times |E|$ matrix such that for each edge $e$, we have

$$A(e_{head}, e) = -1,$$
$$A(e_{tail}, e) = \gamma(e).$$

In order words, the constraint $A\bar{x} = F\bar{1}_t$ requires the flow to satisfies the flow conversation at all vertices except $s$ and $t$ and requires it flows $F$ unit of flow to $t$. We assume $c_e$ are integer and $\gamma_e$ is a rational number. Let $U$ be the maximum of $c_e$, $q_e$, the numerator of $\gamma_e$ and the denominator of $\gamma_e$. For the generalized flow problems, getting an efficient exact algorithm is difficult and we aim for approximation algorithms only.

\textbf{Definition 23.} We call a flow an $\epsilon$–approximate generalized maximum flow if it is a flow satisfies the flow conservation and the flow value is larger than maximum flow value minus $\epsilon$. We call a flow is an $\epsilon$-approximate generalized minimum cost maximum flow if it is an $\epsilon$-approximate maximum flow and has cost not greater than the minimum cost maximum flow value.
Note that \( \text{rank}(A) \leq n - 1 \) and hence our algorithm takes only \( O(\sqrt{nL}) \) iterations. Therefore, the problems remaining are to compute \( L \) and bound how much time is required to solve the linear systems involved. However, \( L \) is large in the most general setting and hence we cannot use the standard theory to say how to get the initial point, how to round to the vertex. Furthermore, the condition number of \( A^T \) can be very bad.

In [5], they used dual path following to solve the generalized minimum cost flow problem with the caveats that the dual polytope is not bounded, the problem of getting the initial flow, the problem of rounding it to a feasible flow. We use there analysis to formulate the problem in a manner more amenable to our algorithms. Since we are doing the primal path following, we will state a reformulation of the LP slightly different.

**Theorem 24.** [5] Given a directed graph \( G \). We can find a new directed graph \( \tilde{G} \) in \( \tilde{O}(m) \) time such that the modified linear program

\[
\min_{0 \leq x_i \leq c_i, 0 \leq y_i \leq 4mU^2, 0 \leq z_i \leq 4mU^2} \quad \tilde{q}^T \tilde{x} + \frac{256m^5U^5}{e^2} \left( \tilde{x}^T \tilde{y} + \tilde{x}^T \tilde{z} \right)
\]

such that \( A\tilde{x} + \tilde{y} - \tilde{z} = \tilde{F}m \tilde{t} \)

satisfies the following conditions:

1. \( \tilde{x} = \frac{1}{2} \tilde{u}, \tilde{y} = 2mU^2 \tilde{u} - (A\frac{1}{2} \tilde{u})^+ + F\tilde{t}, z = 2mU^2 \tilde{u} + (A\frac{1}{2} \tilde{u})^+ \) is an interior point of the linear program.

2. Given any \( (\tilde{x}, \tilde{y}, \tilde{z}) \) such that \( \|A\tilde{x} + \tilde{y} - \tilde{z}\|_2 \leq \frac{e^2}{128m^2U^2} \) and with cost value within \( \frac{e^2}{128m^2U^2} \) of the optimum. Then, one can compute an approximate minimum cost maximum flow for graph \( G \) in time \( \tilde{O}(m) \).

3. The linear system of the linear program is well-conditioned, i.e., the condition number of \( [A I -I] \begin{bmatrix} A^T \\ I \\ -I \end{bmatrix} \) is \( O(mU) \).

4. The linear system of the linear program can be solve in nearly linear time, i.e. for any diagonal matrix \( S \) with condition number \( \kappa \) and vector \( b \), it takes \( \tilde{O}(m \log (\frac{\kappa U}{\delta})) \) time to find \( x \) such that

\[
\|x - L^{-1}b\|_L \leq \delta \|x\|_L
\]

where \( L = [A I -I] S \begin{bmatrix} A^T \\ I \\ -I \end{bmatrix} \).

The main difference between what stated in [5] and here is that

1. Our linear program solver can support constraint \( l_i \leq x_i \leq u_i \) and hence we do not need to split the flow variable to positive part and negative part.

2. Our linear program solver is primal path following and hence we add the constraint \( y_i \leq 4mU^2 \) and \( z_i \leq 4mU^2 \). Since the maximum flow value is at most \( mU^2 \), it does not affect the optimal solution of the linear program.

3. We remove the variable \( x_3 \) in [5] because the purpose of that is to make the dual polytope is bounded and we do not need it here.
Using the reduction mentioned above, one can obtain the promised generalized minimum cost flow algorithm.

**Theorem 25.** Using definition 23. There is an algorithm to compute an approximate generalized minimum cost maximum flow in time \( \tilde{O}(m\sqrt{n}\log^2(U/\epsilon)) \). Furthermore, there is an algorithm to compute an exact standard minimum cost maximum flow in time \( \tilde{O}(m\sqrt{n}\log^2(U)) \).

**Proof.** Using the reduction above and 22, we get an algorithm of generalized minimum cost flow by solving \( \tilde{O}(\sqrt{n}) \) linear systems to \( O(\log(mU/\epsilon)) \) bit accuracy and the condition number of those systems are \( \text{poly}(mU/\epsilon) \). Therefore, it takes \( \tilde{O}(m\log(U/\epsilon)) \) time to solve each systems.

For the standard minimum cost maximum flow problem, it is known that the solution set is a convex polytope with integer coordinates and we can use Isolation lemma to make sure there is unique minimum. Hence, we only need to take \( \epsilon = \text{poly}(1/mU) \) and round the solution to the closest integer. See Section 3.5 in [5] for details. \( \Box \)

## 7 Acknowledgments

We thank Yan Kit Chim, Andreea Gane, Jonathan A. Kelner, Lap Chi Lau, Aleksander Mądry, Lorenzo Orecchia for many helpful conversations. This work was partially supported by NSF awards 0843915 and 1111109, NSF Graduate Research Fellowship (grant no. 1122374) and Hong Kong RGC grant 2150701.

**References**

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows: theory, algorithms, and applications. 1993.

[2] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[3] András A Benczúr and David R Karger. Approximating st minimum cuts in \( \tilde{O}(n^2) \) time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 47–55. ACM, 1996.

[4] Paul Christiano, Jonathan A Kelner, Aleksander Madry, Daniel A Spielman, and Shang-Hua 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*, pages 273–282. ACM, 2011.

[5] Samuel I Daitch and Daniel A Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 451–460. ACM, 2008.

[6] Shimon Even and R Endre Tarjan. Network flow and testing graph connectivity. *SIAM journal on computing*, 4(4):507–518, 1975.

[7] Robert M Freund and Michael J Todd. Barrier functions and interior-point algorithms for linear programming with zero-, one-, or two-sided bounds on the variables. *Mathematics of Operations Research*, 20(2):415–440, 1995.
[8] Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, 1998.

[9] David Karger and Matthew Levine. Random sampling in residual graphs. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 63–66. ACM, 2002.

[10] David R Karger. Better random sampling algorithms for flows in undirected graphs. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 490–499. Society for Industrial and Applied Mathematics, 1998.

[11] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.

[12] Jonathan A Kelner, Lorenzo Orecchia, Yin Tat Lee, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. *arXiv preprint arXiv:1304.2338*, 2013.

[13] Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A Simple, Combinatorial Algorithm for Solving SDD Systems in Nearly-Linear Time. January 2013.

[14] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD systems. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, 2010.

[15] Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-\(m \log n\) time solver for sdd linear systems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 590–598, oct. 2011.

[16] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *The 45th ACM Symposium on Theory of Computing (STOC)*, pages 755–764, 2013.

[17] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *The 54th Annual Symposium on Foundations of Computer Science (FOCS)*, 2013.

[18] Yin Tat Lee and Aaron Sidford. Matching the universal barrier without paying the costs: Solving linear programs with \(\tilde{\Theta}(\sqrt{\text{rank}})\) linear system solves. *CoRR*, 2013.

[19] Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. 2012.

[20] Aleksander Madry. Fast approximation algorithms for cut-based problems in undirected graphs. In *FOCS*, pages 245–254, 2010.

[21] Aleksander Madry. Navigating central path with electrical flows: from flows to matchings, and back. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, 2013.
Lemma 26. For any norm $\| \cdot \|$ and $\| \vec{y} \|_Q = \min_{\vec{\eta} \in \mathbb{R}^n} \left\| \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}} \right\|$ we have

$$\|\vec{y}\|_Q \leq \|P_{\vec{x},\vec{w}}\vec{y}\| \leq \|P_{\vec{x},\vec{w}}\| \cdot \|\vec{y}\|_Q.$$ 

Proof. By definition $P_{\vec{x},\vec{w}}\vec{y} = \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}}$ for some $\vec{\eta} \in \mathbb{R}^n$. Consequently,

$$\|\vec{y}\|_Q = \min_{\vec{\eta} \in \mathbb{R}^n} \left\| \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}} \right\| \leq \|P_{\vec{x},\vec{w}}\vec{y}\|.$$ 

On the other hand, let $\vec{\eta}$ by such that such that $\|\vec{y}\|_Q = \left\| \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}} \right\|$. Then, since $P_{\vec{x},\vec{w}}W^{-1}(\Phi'')^{-1/2}A = 0$ we have

$$\|P_{\vec{x},\vec{w}}\vec{y}\| = \|P_{\vec{x},\vec{w}} \left( \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}} \right) \| \leq \|P_{\vec{x},\vec{w}}\| \cdot \left\| \vec{y} - \frac{A\vec{\eta}}{\vec{w} \sqrt{\phi''(\vec{x})}} \right\| = \|P_{\vec{x},\vec{w}}\| \cdot \|\vec{y}\|_Q.$$
Lemma 27. (Log Notation) Suppose $|\log(a) - \log(b)| = \epsilon \leq 1/2$ then
\[
|a - b| \leq \epsilon + \epsilon^2.
\]

If $|\frac{a - b}{b}| = \epsilon \leq 1/2$, then $|\log(a) - \log(b)| \leq \epsilon + \epsilon^2$.

Lemma 28. (18, Appendix) For any projection matrix $P \in \mathbb{R}^{n \times n}$, $\Sigma = \text{diag}(P)$, $i, j \in [n]$, $\vec{x} \in \mathbb{R}^n$, and $\vec{w} \in \mathbb{R}_{>0}$ we have

1. $\Sigma_{ii} = \sum_{j \in [n]} P^{(2)}_{ij}$,
2. $0 \preceq P^{(2)} \preceq \Sigma \preceq I$,
3. $P^{(2)}_{ij} \preceq \Sigma_{ii} \Sigma_{jj}$,
4. $|\vec{1}_i^T P^{(2)} \vec{x}| \leq \Sigma_{ii} \|\vec{x}\| \Sigma$,
5. $\nabla_{\vec{w}} \log \det(A^T W A) = \Sigma_A(\vec{w}) W^{-1} \vec{1}$.
6. $J_{\vec{w}}(\sigma_A(\vec{w})) = \Lambda_A(\vec{w}) W^{-1}$.

A.1 Projection on Mixed Norm Ball

In the (18), we studied the following problem:

\[
\max_{\|\vec{x}\|_2 \leq 1, -l_i \leq x_i \leq l_i} \langle \vec{a}, \vec{x} \rangle
\]  \hspace{1cm} (A.1)

for some given vector $\vec{a}$ and $\vec{l}$. We proved that the following algorithm outputs a solution of (A.1) in depth $\tilde{O}(1)$ and work $\tilde{O}(n)$.

\[
\vec{x} = \text{projectOntoBallBoxParallel}(\vec{a})
\]

1. Set $\vec{d} = \vec{a}/\|\vec{a}\|_2$.
2. Sort the coordinate such that $|a_i|/l_i$ is in descending order.
3. Precompute $\sum_{k=0}^i l_k^2$ and $\sum_{k=0}^i a_k^2$ for all $i$.
4. Find the first $i$ such that $1 - \sum_{k=0}^i a_k^2 \leq l_i^2/\sum_{k=0}^i l_k^2$.
5. Output $\vec{x} = \left\{ \begin{array}{ll} \text{sign}(a_j) l_j & \text{if } j \in \{1, 2, \cdots, i\} \\ \sqrt{\frac{1 - \sum_{k=0}^i a_k^2}{1 - \sum_{k=0}^i l_k^2}} a_j & \text{otherwise} \end{array} \right.$

In this section, we show that the algorithm above can be transformed to solve the problem

\[
\max_{\|\vec{x}\|_w + \|\vec{x}\|_{\infty} \leq 1} \langle \vec{a}, \vec{x} \rangle
\]  \hspace{1cm} (A.2)
for some given vector \( \bar{a} \) and \( \bar{w} > 0 \). To do this, let study (A.1) more closely. Without loss of generality, we can assume \( \| \bar{a} \|_2 = 1 \) and \( |a_i|/l_i \) is in descending order. The key consequence of projectOntoBallBoxParallel is that the problem (A.1) always has a solution of the form

\[
\tilde{x}_{l,a}^{(i)} = \begin{cases} 
\text{sign}(a_j)l_j & \text{if } j \in \{1, 2, \ldots, i\} \\
\sqrt{\frac{1 - \sum_{k=0}^{i-1} l_k^2}{1 - \sum_{k=0}^{i} a_k^2}} a_j & \text{otherwise}
\end{cases}.
\]  

(A.3)

Let \( i_t \) be the first coordinate such that

\[
\frac{|a_j|}{\sqrt{l_j^2 \left(1 - \sum_{k=0}^{j} a_k^2\right) + a_j^2 \sum_{k=0}^{j} l_k^2}} \leq t < \frac{|a_{j-1}|}{\sqrt{l_{j-1}^2 \left(1 - \sum_{k=0}^{j-1} a_k^2\right) + a_{j-1}^2 \sum_{k=0}^{j-1} l_k^2}}.
\]  

(A.4)

Note that \( i_t \geq i_s \) if \( t \leq s \). Therefore, we have that the set of \( t \) such that \( i_t = j \) is simply\(^{12}\)

\[
\frac{|a_j|}{\sqrt{l_j^2 \left(1 - \sum_{k=0}^{j} a_k^2\right) + a_j^2 \sum_{k=0}^{j} l_k^2}} \leq t < \frac{|a_{j-1}|}{\sqrt{l_{j-1}^2 \left(1 - \sum_{k=0}^{j-1} a_k^2\right) + a_{j-1}^2 \sum_{k=0}^{j-1} l_k^2}}.
\]  

Define the function \( f \) by

\[
f(t) = \max_{\| \bar{z} \|_2 \leq 1, -u_t \leq x_i \leq u_t} \langle \bar{a}, \bar{z} \rangle.
\]

We know that

\[
f(t) = \langle \bar{a}, \tilde{x}_{l,t,a}^{(i_t)} \rangle
\]

\[
= t \sum_{j=1}^{i_t} |a_j| |l_j| + \sqrt{1 - t^2 \sum_{k=0}^{i_t} l_k^2} \sqrt{1 - \sum_{k=0}^{i_t} a_k^2}.
\]

Therefore, we have

\[
\max_{\| \bar{z} \|_2 + \| \bar{f} \|_\infty \leq 1 } \langle \bar{a}, \bar{z} \rangle = \max_{0 \leq t \leq 1} \max_{\| \bar{z} \|_2 \leq 1 - t \text{ and } -u_t \leq x_i \leq u_t } \langle \bar{a}, \bar{z} \rangle
\]

\[
= \max_{0 \leq t \leq 1} (1 - t) \max_{\| \bar{z} \|_2 \leq 1 \text{ and } -u_t \leq x_i \leq u_t } \langle \bar{a}, \bar{z} \rangle
\]

\[
= \max_{0 \leq t \leq 1} (1 - t) f \left( \frac{t}{1-t} \right)
\]

\[
= \max_{0 \leq t \leq 1} t \sum_{j=1}^{i_t} |a_j| |l_j| + \sqrt{(1 - t)^2 - t^2 \sum_{k=0}^{i_t} l_k^2} \sqrt{1 - \sum_{k=0}^{i_t} a_k^2}.
\]

Note that the function \( t \sum_{j=1}^{i_t} |a_j| |l_j| + \sqrt{(1 - t)^2 - t^2 \sum_{k=0}^{i_t} l_k^2} \sqrt{1 - \sum_{k=0}^{i_t} a_k^2} \) is concave and the solution has a close form. Therefore, one can compute the maximum value for each interval of \( t \) and find which is the best. Hence, we get the following algorithm.

\(^{12}\)There are some boundary cases we ignored for simplicity.
\[ \bar{x} = \text{projectOntoMixedNormBallParallel}(\bar{a}, l) \]

1. Set \( \bar{a} = \bar{a}/\|\bar{a}\|_2 \).
2. Sort the coordinate such that \( |a_i|/l_i \) is in descending order.
3. Precompute \( \sum_{k=0}^{i} l_k^2 \), \( \sum_{k=0}^{i} a_k^2 \) and \( \sum_{j=1}^{i} |a_j||l_j| \) for all \( i \).
4. Let \( g_i(t) = t \sum_{j=1}^{i} |a_j||l_j| + \sqrt{(1-t)^2 - t^2 \sum_{k=0}^{i} l_k^2} \sqrt{1 - \sum_{k=0}^{i} a_k^2} \).
5. For each \( j \in \{1, \cdots, n\} \), Find \( t_j = \arg \max_{t_i = j} g_j(t) \) using (A.4).
6. Find \( \hat{i} = \arg \max_{i} g_{\hat{i}}(t_{\hat{i}}) \).
7. Output \( (1 - t_{\hat{i}}) \bar{x}^{(t_{\hat{i}})}_{\bar{x} - t_{\hat{i}} l, a} \) defined by (A.3).

The discussion above leads to the following theorem. The problem in the form (A.2) can be solved by \text{projectOntoMixedNormBallParallel} and a change of variables.

**Theorem 29.** The algorithm \text{projectOntoMixedNormBallParallel} outputs a solution to the problem

\[
\max \left( \|\bar{x}\|_2^2 + \|\bar{r}^{-1}\bar{x}\|_\infty \right) \leq 1 \langle \bar{a}, \bar{x} \rangle
\]

in depth \( \tilde{O}(1) \) and work \( \tilde{O}(n) \).