Interior point methods are not worse than Simplex

Xavier Allamigeon
Inria, CMAP, CNRS, Ecole Polytechnique, IP Paris
Paris, France
xavier.allamigeon@inria.fr

Daniel Dadush
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
dadush@cwi.nl

Georg Loho
University of Twente
Enschede, The Netherlands
g.loho@utwente.nl

Bento Natura
London School of Economics and Political Science
London, UK
b.natura@lse.ac.uk

László A. Végh
London School of Economics and Political Science
London, UK
l.vegh@lse.ac.uk

***Abstract*—Whereas interior point methods provide polynomial-time linear programming algorithms, the running time bounds depend on bit-complexity or condition measures that can be unbounded in the problem dimension. This is in contrast with the simplex method that always admits an exponential bound. We introduce a new polynomial-time path-following interior point method where the number of iterations also admits a combinatorial upper bound $O(2^n n^{1.5}\log n)$ for an n-variable linear program in standard form. This complements previous work by Allamigeon, Benchimol, Gaubert, and Joswig (SIAGA 2018) that exhibited a family of instances where any path-following method must take exponentially many iterations.**

The number of iterations of our algorithm is at most $O(n^{1.5}\log n)$ times the number of segments of any piecewise linear curve in the wide neighborhood of the central path. In particular, it matches the number of iterations of any path following interior point method up to this polynomial factor. The overall exponential upper bound derives from studying the 'max central path', a piecewise-linear curve with the number of pieces bounded by the total length of $2n$ shadow vertex simplex paths.

From the existence of a line segment in the wide neighborhood we derive strong implications on the structure of the corresponding segment of the central path. Our algorithm is able to detect this structure from the local geometry at the current iterate, and constructs a step direction that descends along this segment. The bound $O(n^{1.5}\log n)$ that applies for arbitrarily long line segments is derived from a combinatorial progress measure.

Our algorithm falls into the family of layered least squares interior point methods introduced by Vavasis and Ye (Math. Prog. 1996). In contrast to previous layered least squares methods that partition the kernel of the constraint matrix into coordinate subspaces, our method creates layers based on a general subspace providing more flexibility. Our result also implies the same bound on the number of iterations of the trust region interior point method by Lan, Monteiro, and Tsuchiya (SIOPT 2009).

I. Introduction

In this paper, we explore connections between interior point methods and the simplex method, the two most commonly used classes of algorithms for linear programming. We consider linear programming (LP) in the following primal-dual form:

$$\begin{array}{ll} \min\ \langle c,x\rangle & \max\ \langle b,y\rangle \\ Ax=b & A^\top y+s=c \\ x\geq 0\,, & s\geq 0\,, \end{array} \tag{LP}$$

where $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$ and $c\in\mathbb{R}^n$, and $\mathrm{rk}(A)=m$. We let

$$\mathcal{P}=\{x\in\mathbb{R}^n : Ax=b, x\geq 0\}\,,$$
$$\mathcal{D}=\{s\in\mathbb{R}^n : \exists y\ A^\top y+s=c\,,\ s\geq 0\}$$

denote the primal and dual feasible regions. Our focus is on LP algorithms that find exact primal and dual optimal solutions.

The *simplex method* traverses a path formed by vertices and edges of $\mathcal{P}$ according to a certain *pivot rule*. Albeit efficient in practice, there is no polynomial-time variant known, and there are exponential worst case examples for several pivot rules. The first such construction was given by Klee and Minty [1] for Dantzig's pivot rule.

Breakthrough developments in the seventies and eighties led to the first polynomial-time algorithms for linear programming: the *ellipsoid method* by Khachiyan [2], and *interior point methods* introduced by Karmarkar [3]. The running time of these algorithms is $\mathrm{poly}(n,L)$, where L denotes the encoding-length L of the rational input (A,b,c) of (LP).

While the simplex method may be exponential, it is never worse: for any non-cycling pivot rule, the number of pivot steps can be bounded by the number of bases, at most $\binom{n}{m}<2^n$. Whereas the bound $\mathrm{poly}(n,L)$ is typically much better, the encoding length L may be arbitrarily large. To the extent of our knowledge, no variant of the ellipsoid or interior point methods have been shown to admit a bound $f(n)$ on the number of iterations for any function $f\colon\mathbb{N}\to\mathbb{N}$ prior to our work.

Even though LPs with exponential encoding length do not frequently appear in practice, there are examples when the binary encoding is exponential yet one could efficiently implement arithmetic operations using a different encoding, see Megiddo [4]. The net present value problem in project scheduling is a particular example of a natural optimization

The full version of this paper is available at https://arxiv.org/abs/2206.08810. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme: DD from grant agreement no. 805241-QIP, BN and LAV from grant agreements no. 757481-ScaleOpt. We thank the 2021 Hausdorff Research Institute for Mathematics Program Discrete Optimization during which part of this work was developed.

problem that can be reformulated as an LP of exponential encoding length, see Grinold [5]. From a theoretical perspective, finding an interior point method with an absolute bound $f(n)$ on the number of iterations connects to the fundamental open question on finding a strongly polynomial algorithm to solve linear programming. Besides being polynomial time, such an algorithm must achieve a number of arithmetic operations in $\mathrm{poly}(m, n)$. This question takes its roots in the development of the simplex method, and appears in Smale's list of open problems for the 21st century [6].

a) Interior point methods and the central path: Whereas the simplex method moves on the boundary of the feasible region $\mathcal{P}$, interior point methods (IPM) reach an optimal solution by iterating through the strict interior of $\mathcal{P}$. Path-following interior point methods are driven to an optimal point by following a smooth trajectory called the *central path*. In the most standard setting [7], the latter is defined as the parametric curve $\mu \in (0, \infty) \mapsto z(\mu) := (x(\mu), s(\mu))$, where $x(\mu)$ and $(y(\mu), s(\mu))$ are the unique solutions to the system

$$\begin{aligned} Ax(\mu) = b\,, \;\; x(\mu) > 0\,, \\ A^\top y(\mu) + s(\mu) = c\,, \;\; s(\mu) > 0\,, \\ x(\mu)_i s(\mu)_i = \mu \quad \text{for all } \; i \in [n]\,. \end{aligned} \tag{1}$$

This system arises from the optimality conditions of convex problems obtained by penalizing the original linear programs with the logarithmic barrier, i.e., respectively adding terms of the form $-\mu \sum_{i=1}^n \log x_i$ and $\mu \sum_{i=1}^n \log s_i$ to the objective functions of the primal and dual (LP). The weight of the penalty is given by the parameter $\mu > 0$. When $\mu \searrow 0$, the central path $z(\mu)$ converges to a pair of optimal solutions $(x^\star, s^\star)$ of (LP), which can be easily deduced from the fact that the duality gap of $z(\mu)$ is given by $\langle c, x(\mu)\rangle - \langle b, y(\mu)\rangle = \langle x(\mu), s(\mu)\rangle = n\mu$. Accordingly, we define the quantity $\overline{\mu}(z) := \langle x, s\rangle / n$ for any feasible point $z = (x, s) \in \mathcal{P} \times \mathcal{D}$, which we refer to as the *normalized duality gap* of z.

Interior point methods iteratively compute approximations of the points on the central path associated with successive values of μ that decrease geometrically; at most $O(\sqrt{n} \log(\mu/\mu'))$ iterations are needed to decrease the normalized duality gap from μ to μ'. The iterations follow an improvement direction, e.g., a Newton step, while remaining in a certain neighborhood of the central path, and can be implemented in polynomial time. The classical analysis yields a running time $O(n^{3.5} L)$ for solving (LP) for a rational input (A, b, c) of total encoding length L. There have been significant improvements in recent years both for general LP as well as for special classes, see Section I-B.

A running time bound dependent on L requires a rational input; in contrast, the simplex method can be implemented in $2^n \mathrm{poly}(n)$ even in the *real model of computation*. Whereas standard IPMs use bit-complexity arguments to terminate, they have also been extended to the real model of computation, e.g., by Vavasis and Ye [8]. The running time of such algorithms is parametrized by *condition numbers* that capture geometric properties of the input. In a remarkable paper, Vavasis and Ye [9] introduced a *layered least squares (LLS)* interior point method that runs in $O(n^{3.5} \log(\bar{\chi}_A + n))$ iterations, where $\bar{\chi}_A$ is the Dikin–Stuart–Todd condition number associated with the kernel of A (but independent of b and c). As a consequence, they also derive a structural characterization of the central path: there are at most $\binom{n}{2}$ 'short and curved' segments, possibly separated by 'long and straight' segments. The LLS directions are refined Newton steps that can traverse the latter segments.

Lan, Monteiro and Tsuchiya [10] gave a scaling invariant *trust region IPM* taking $O(n^{3.5} \log(\bar{\chi}^*_A + n))$ iterations. Here, $\bar{\chi}^*_A$ is the minimum value of $\bar{\chi}_A$ that can be achieved by any column rescaling. However, computing the step directions in this algorithm has a weakly polynomial dependence on the right hand side. In recent work, [11] gave a scaling invariant LLS algorithm with iteration bound $O(n^{2.5} \log(n) \log(\bar{\chi}^*_A + n))$, where the step directions can be computed in strongly polynomial time, by solving linear systems. We discuss the literature on such IPM methods in more detail in Section I-B.

b) Lower bounds on interior point methods: LLS methods provide strongly polynomial LP algorithms whenever $\bar{\chi}^*_A \in 2^{\mathrm{poly}(n)}$; this is always the case if the encoding-length of A is polynomially bounded. One may wonder if some variant of IPM could be strongly polynomial for all LPs. A negative answer to this question was given in recent work by Allamigeon, Benchimol, Gaubert, and Joswig: they used tropical geometry to build pathological linear programs on which the number of iterations of IPM has to be exponential (in m, n) [12], [13]. Their construction shows that, when the entries of A, b, and c are of very different orders of magnitude, the central path can be significantly deformed to the boundary of the feasible set. Allamigeon, Gaubert and Vandame later extended this result to the broad class of path-following IPMs using any self-concordant barrier function [14]. They exhibited a counterexample where the feasible set is an n-dimensional combinatorial cube and the shape of the central path is analogous to the simplex paths on pathological instances of LP for the simplex method, akin to the Klee–Minty cube [1].

c) The shadow vertex simplex rule: We introduce a new IPM in this paper whose analysis can be related to the *shadow vertex simplex rule*. Originally dubbed 'parametric simplex' by Gass and Saaty [15], this is one of the most extensively analyzed simplex rules from a theoretical perspective. The shadow vertex rule was used in Borgwardt's average case analysis [16] and in Spielman and Teng's smoothed analysis [17]. The interested reader may refer to a recent survey [18] for a detailed exposition.

Given a pointed polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and two objectives $c^{(1)}, c^{(2)} \in \mathbb{R}^n$, the shadow vertex rule consists in iterating over the vertices of $\mathcal{P}$ successively maximizing the objectives $(1 - \lambda)c^{(1)} + \lambda c^{(2)}$ as λ goes from 0 to 1. Under non-degeneracy assumptions, the vertices of the path correspond to those vertices of the two-dimensional projection $\{(\langle c^{(1)}, x\rangle, \langle c^{(2)}, x\rangle) \colon x \in \mathcal{P}\}$ that maximize some open interval of objectives $(1 - \lambda)e^1 + \lambda e^2$, $\lambda \in [0, 1]$ (where e^1 and e^2 stand for the unit vectors of $\mathbb{R}^2$). We denote by $S_{\mathcal{P}}(c^{(1)}, c^{(2)})$ the number of vertices of the projection of the simplex path in this two-dimensional projection; this corresponds to the number of non-degenerate pivots.

A. Contributions

The purpose of this work is to establish a natural connection between the complexity of IPM and that of the simplex method, and deduce combinatorial bounds on the number of iterations. To this end, we introduce an interior point method called IPM WITH SUBSPACE LLS (see Algorithm 1), whose number of iterations is, up to a factor $O(n^{1.5}\log n)$, bounded by the number of pieces of any piecewise linear curve contained a *wide neighborhood* of the central path. This neighborhood is defined as

$$\mathcal{N}^{-\infty}(\theta) := \{z = (x, s) \in \mathcal{P} \times \mathcal{D} : xs \geq (1-\theta)\overline{\mu}(z)\mathbf{1}\} \tag{2}$$

for $0 < \theta < 1$, where $xs \in \mathbb{R}^n$ denotes the Hadamard-product and $\mathbf{1} \in \mathbb{R}^n$ is the n-dimensional all-one vector. Our algorithm will however navigate through the narrower ℓ_2-neighborhood of the central path:

$$\mathcal{N}(\beta) := \left\{ z = (x, s) \in \mathcal{P} \times \mathcal{D} : \left\| \frac{xs}{\overline{\mu}(z)} - \mathbf{1} \right\| \leq \beta \right\} \tag{3}$$

for $0 < \beta < 1/4$.

Theorem I.1. *Assume that there exists a piecewise linear curve* $\Gamma\colon (0, \mu_0) \to \mathcal{N}^{-\infty}(\theta)$*, for some* $\mu_0 > 0$ *and* $\theta \in (0,1)$*, satisfying* $\overline{\mu}(\Gamma(\mu)) = \mu$*,* $\forall \mu \in (0, \mu_0)$*. Starting from any point* $z^0 \in \mathcal{N}(\beta)$ *such that* $\overline{\mu}(z^0) \leq \mu_0$*, the algorithm* IPM WITH SUBSPACE LLS *finds an optimal solution of* (LP) *in* $O(n^{1.5}\log(\frac{n}{1-\theta})\,T)$*, where* T *is the number of linear segments in* Γ.

At a high level, our strategy is to show that any 'somewhat straight' segment of the central path, corresponding to a single straight segment in the wide neighborhood $\mathcal{N}^{-\infty}(\theta)$, can be decomposed into at most n short segments of length $\mathrm{poly}(n/(1-\theta))$ (as measured by the ratio of the start and end parameter), where consecutive short segments are possibly separated by 'long and straight' segments. To traverse the long and straight segments we develop a novel *subspace* LLS step, which generalizes prior LLS steps from coordinate subspaces to general ones. Before describing this in more details, we present two applications of Theorem I.1.

a) An exponential upper bound on the number of iterations: The first application relies on a piecewise linear curve that we call the *max central path*, and that is related with $2n$ simplex paths. It is defined as the parametric curve $g \mapsto z^{\mathrm{m}}(g) := (x^{\mathrm{m}}(g), s^{\mathrm{m}}(g)) \in \mathbb{R}^{2n}_{+}$, where $x^{\mathrm{m}}_i(g)$ and $s^{\mathrm{m}}_i(g)$ are the optimal values of the following parametric LP, respectively:

$$\begin{array}{ll} \max\ x_i & \max\ s_i \\ \quad Ax = b\,,\ x \geq 0 & \quad A^\top y + s = c\,,\ s \geq 0 \\ \langle c, x\rangle \leq v^\star + g\,, & \langle b, y\rangle \geq v^\star - g\,, \end{array} \tag{4}$$

where we denote by $v^\star$ the optimal value of (LP). The following theorem shows that the max central path shares important similarities with the central path:

Theorem I.2 (Centrality of the max central path)**.** *For all* $g \geq 0$*, we have that*

$$g \leq x^{\mathrm{m}}_i(g)s^{\mathrm{m}}_i(g) \leq 2g \quad \forall i \in [n]\,.$$

For $\mu > 0$*, we have that*

$$z^{\mathrm{m}}(n\mu) \geq z^{\mathrm{cp}}(\mu) \geq \frac{z^{\mathrm{m}}(n\mu)}{2n}\,.$$

While the max central path does not correspond to a feasible path inside $\mathcal{P} \times \mathcal{D}$, it is in fact close to a piecewise linear path that lives inside the wide neighborhood of the central path having the same number of breakpoints.

Theorem I.3. *There exists a piecewise linear curve* $\Gamma\colon \mathbb{R}_+ \to \mathcal{N}^{-\infty}(1 - \frac{1}{2n})$ *with at most as many linear segments as* $g \mapsto z^{\mathrm{m}}(g)$ *satisfying* $\overline{\mu}(\Gamma(s)) = s$*,* $\forall s \geq 0$.

The theorem follows by taking the average of n primal-dual optimal solutions to the programs (4) for $i = 1, 2, \ldots, n$.

It can be shown that the maps $x^{\mathrm{m}}_i(\cdot)$ and $s^{\mathrm{m}}_i(\cdot)$ are piecewise linear, and the number of pieces can be related to the complexity of the simplex method with the shadow vertex rule.

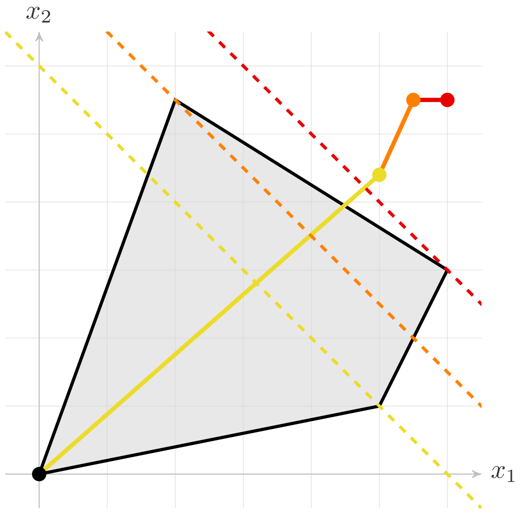

Fig. 1. The max central path in the coordinates (x_1, x_2) with cost function $x_1 + x_2$. Dashed lines correspond to level sets at breakpoints.

Recall that $(x^\star, s^\star)$ is the optimal solution of (LP) at the central path limit point, and $S_{\mathcal{P}}(-s^\star, e_i)$ and $S_{\mathcal{D}}(-x^\star, e_i)$ denote the number of nondegenerate pivots in the primal and dual shadow vertex paths for the indicated objective functions. We let $\mathcal{V}_{\mathcal{P}}$ and $\mathcal{V}_{\mathcal{D}}$ denote the number of vertices of the primal and dual feasible polyhedra, respectively. We show the following relation.

Lemma I.4 (Piecewise Linearity of MCP). *$g \mapsto z^{\mathrm{m}}(g)$ is piecewise linear and entry-wise non-decreasing with at most*

$$\min\left\{\sum_{i=1}^{n} S_{\mathcal{P}}(-s^\star, e^i) + S_{\mathcal{D}}(-x^\star, e^i), \mathcal{V}_{\mathcal{P}} + \mathcal{V}_{\mathcal{D}}\right\}$$

pieces.

This lemma, along with Theorem I.1 and Theorem I.2, implies the following bound:

Theorem I.5. *From any point $z^0 \in \mathcal{N}(\beta)$, the algorithm* IPM WITH SUBSPACE LLS *finds an optimal solution of* (LP) *in a number of iterations bounded by $O(2^n n^{1.5} \log n)$.*

This complements the results of [12], [13] by giving a singly exponential upper bound. We note that the max central path also plays an important if implicit role in the papers [12]–[14], as it can be directly related to the tropical central path by the log-limit, see discussion in Section I-B.

Theorem I.5 assumes that a feasible starting point $z^0 \in \mathcal{N}(\beta)$ is given. This assumption can be removed e.g. by using the standard homogeneous self-dual embedding [19, Section 5.3.1]. Then, the bounds in the theorem will refer to the shadow vertex paths and the number of vertices in the self-dual program.

b) Matching the complexity of any path following method: The second implication of Theorem I.1 provides polynomial-time bounds in case the bit-complexity or a condition number such as $\bar{\chi}^*_A$ is bounded. We show that—apart from a factor $O(n^{1.5} \log \frac{n}{1-\theta})$—the number of iterations of IPM WITH SUBSPACE LLS *is at most that of any IPM that stays in the wide neighborhood.*

Indeed, any IPM induces a piecewise linear curve formed by the line segments between the successive iterates. Already the wide neighborhood $\mathcal{N}^{-\infty}(1/2)$ is known to contain this piecewise linear curve for a large class of IPM based on the logarithmic barrier; we refer to [12, Section 2] for a detailed discussion. We note that our algorithm matches—up to a polynomial factor—even any IPM that only stays in the extremely wide neighborhood $\mathcal{N}^{-\infty}\left(1 - 1/2^{\mathrm{poly}(n)}\right)$.

Theorem I.6. *Suppose that an IPM reduces the duality gap from μ_0 to $\mu_1 \geq 0$ in T iterations staying throughout in the wide neighborhood $\mathcal{N}^{-\infty}(\theta)$ for some $0 < \theta < 1$. Then, from any point $z^0 \in \mathcal{N}(\beta)$ satisfying $\overline{\mu}(z^0) \leq \mu_0$, the algorithm* IPM WITH SUBSPACE LLS *finds a solution z^1 with $\overline{\mu}(z^1) \leq \mu_1$ in at most $O(n^{1.5} \log(\frac{n}{1-\theta}) T)$ iterations.*

c) Comparison to the Trust Region IPM: IPM WITH SUBSPACE LLS also has an interesting relation to the Trust Region IPM algorithm by Lan, Monteiro, and Tsuchiya [10]. The trust region steps are obtained as optimal solutions to primal and dual quadratic programs (18). These programs in essence capture the longest possible step achievable at the current point (up to a certain factor). However, solving these programs to sufficient accuracy requires weakly polynomial dependence on the input. Lan, Monteiro, and Tsuchiya show in [10] that the number of iterations of the trust region algorithm can be bounded as $O(n^{3.5} \log(\bar{\chi}^*_A + n))$, by adapting the analysis of the LLS methods [9], [20].

The step directions used by our algorithm are feasible solutions to (18) for a suitable parameter. This implies that the steps of the Trust Region algorithm are always at least as long as the steps in our algorithm; as a consequence, the iteration bounds of our algorithm are also applicable to the Trust Region algorithm. Whereas any individual step of our algorithm could be arbitrarily worse than the one using the trust region step, Theorem I.6 implies that overall we may only take $O(n^{1.5} \log n)$ more iterations. We emphasize that [10] only provides the $\bar{\chi}^*_A$ dependent iteration bound, and we do not see any obvious ways to obtain any $f(n)$ bound on their algorithm, other than comparing it to IPM WITH SUBSPACE LLS.

A further advantage of our algorithm is that the iterations can be implemented in strongly polynomial time, using simple projection steps. The description of this paper requires a subspace V that can be obtained as a singular value subspace from a singular value decomposition. However, it suffices to compute rough approximations on the singular values.

B. Related Work

Interior points methods have been a tremendously active and fruitful research area since the seminal works of Karmarkar [3] and Renegar [7] in the 80's. Remarkable advances have been made both in speed as well as applicability of IPMs. We first briefly review works that—unlike the present paper—aim for ε-approximate solutions. A key ingredient has been the use of different, self-concordant barrier functions. Like the logarithmic barrier, every such function gives rise to a notion of central path. In the general setting, the iteration complexity to get an ε-approximation of the optimal value is bounded by $O(\vartheta^{1/2} \log \varepsilon^{-1})$, where ϑ is a complexity parameter specific to the barrier function. General bounds on self-concordant barriers were given by Nesterov and Nemirovski [21], improved recently by Lee and Yue [22]. Specific barrier functions include Vaidya's volumetric barrier [23], the entropic barrier by Bubeck and Eldan [24], and the weighted log-barrier by Lee and Sidford [25], [26].

Recent improvements make use of efficient data structures to amortize the cost of the iterative updates, and work with approximate computations, see Cohen, Lee and Song [27], van den Brand [28], and van den Brand, Lee, Sidford, and Song [29]. For special classes of LP such as network flow and matching problems, even faster algorithms have been obtained using, among other techniques, fast Laplacian solvers [30], see e.g. [31]–[36], culminating in the very recent near-linear time minimum-cost flow algorithm [37].

Layered least squares IPMs, initiated by Vavasis and Ye [9] find exact optimal solutions and their running time bound is independent of b and c. Improved LLS algorithms were given by Megiddo, Mizuno, and Tsuchiya [38] and Monteiro and Tsuchiya [20], [39]. As discussed previously, scaling invariant algorithms with a $\bar{\chi}^*_A$ dependence are the Trust Region algorithm by Lan, Monteiro, and Tsuchiya [10], and the LLS algorithm [11] that relies on approximating circuit imbalances.

There is an interesting connection between IPMs and differential geometry. Sonnevend, Stoer, and Zhao [40] introduced a primal-dual curvature concept for the central path, and related the curvature integral to the iteration complexity of IPMs. Monteiro and Tsuchiya [41] showed that a curvature integral is bounded by $O(n^{3.5}\log(\bar{\chi}_A^*+n))$. This has been extended to SDP and symmetric cone programming [42], and it was also studied in the context of information geometry [43].

Relating the central path with a simplex path has been mainly used to build LP with pathological properties. On top of the construction of [14] that we already discussed, Deza, Nematollahi and Terlaky [44] built a Klee–Minty cube with exponentially many redundant inequalities where the central path is distorted into the neighborhood of the simplex path that visits the 2^m vertices.

The max central path studied in this paper is related to the tropical central path in [12]–[14]. The latter arises when studying parametric families of LP where the input (A,b,c) depends on a parameter $t>1$. The tropical central path is defined as the log-limit, i.e., the limit as $t\to\infty$ of the image under the map $z\mapsto \log_t z = \frac{\log z}{\log t}$, of the central path of these LP. In [12]–[14], it was shown that the tropical central path corresponds to the greatest point (entrywise) of the log-limit of the feasible sets of (4). This turns out to be precisely the log-limit of the max central path.

II. Preliminaries

a) Notation: For $S\subseteq[n]$, $\mathbf{M}_S\in\mathbb{R}^{m\times|S|}$ refers to the submatrix formed by the columns in S. For $i\in[n]$, we set $\mathbf{M}_{\leq i}=\mathbf{M}_{\{j\in[n]:\, j\leq i\}}$ and $\mathbf{M}_{\geq i}=\mathbf{M}_{\{j\in[n]:\, j\geq i\}}$. We define $\sigma_i(\mathbf{M})$ to be its i-th smallest singular value.

For a vector $x\in\mathbb{R}^n$ and a subspace $W\subseteq\mathbb{R}^n$, we use the notation $xW=\{xw : w\in W\}$. For $p\in\mathbb{Q}$, we also use the notation x^p to denote the vector $(x_i^p)_{i\in[n]}$. Similarly, for $x,y\in\mathbb{R}^n$, we let x/y denote the vector $(x_i/y_i)_{i\in[n]}$.

For a subspace $W\subseteq\mathbb{R}^n$, we define $\Pi_W\colon\mathbb{R}^n\to W$ to be the orthogonal projection onto W. We define $W^\perp := \{x\in\mathbb{R}^n : \langle x,y\rangle=0, \forall y\in W\}$ as the orthogonal complement of W. For a non-empty index set $I\subseteq[n]$, we let $\pi_I\colon\mathbb{R}^n\to\mathbb{R}^I$ to denote the coordinate projection onto I, that is $\pi_I(x)=x_I$, and for a subset $S\subseteq\mathbb{R}^n$, $\pi_I(S)=\{x_I : x\in S\}$. We also define $\mathbb{R}^n_I := \{x\in\mathbb{R}^n : x_i=0, i\in[n]\setminus I\}$ to be the set of vectors in $\mathbb{R}^n$ with support contained in I.

b) Subspace formulation: It will be more convenient for our algorithm and analysis to represent (LP) in an equivalent subspace language. Throughout the extended abstract, we let $W=\ker(A)\subseteq\mathbb{R}^n$. Using this notation, (LP) can be written in the form

$$\begin{aligned}
&\min\ \langle c,x\rangle &\qquad& \max\ \langle d,c-s\rangle\\
&\quad x\in W+d && \quad s\in W^\perp+c\\
&\quad x\geq 0, && \quad s\geq 0,
\end{aligned}\tag{5}$$

where $d\in\mathbb{R}^n$ is any solution $Ad=b$ and $W^\perp=\mathrm{im}(A^\top)$ is the orthogonal complement of W.

Note that $s\in W^\perp+c$ is equivalent to $\exists y\in\mathbb{R}^m$ such that $A^\top y+c=s$. Hence, the original variable y is implicit. The feasible regions can be written as

$$\begin{aligned}
\mathcal{P}&=\{x\in\mathbb{R}^n : x\in W+d, x\geq 0\}\,,\\
\mathcal{D}&=\{s\in\mathbb{R}^n : s\in W^\perp+c, s\geq 0\}\,.
\end{aligned}$$

c) Preliminaries on Interior-Point Methods: We recall standard properties of the central path and IPM that will be required for our algorithm. To ensure that the central path is well-defined, we assume that $\mathcal{P}$ and $\mathcal{D}$ admit strictly feasible solutions, i.e., the sets

$$\mathcal{P}_{++}\coloneqq\{x\in\mathcal{P} :\ x>0\}\,,\quad \mathcal{D}_{++}\coloneqq\{s\in\mathcal{D} :\ s>0\}$$

are both nonempty. We use $z^{\mathrm{cp}}(\mu)=(x^{\mathrm{cp}}(\mu),s^{\mathrm{cp}}(\mu))$ to denote the central path point at μ rather than $z(\mu)=(x(\mu),s(\mu))$ used in the Introduction.

Given $z=(x,s)\in\mathcal{P}\times\mathcal{D}$, we recall that the normalized duality gap is defined as $\overline{\mu}(z)=\frac{\langle x,s\rangle}{n}$.

A key property of the central path is *'near monotonicity'*, formulated in the following lemma, see [9, Lemma 16].

Lemma II.1. *For the central path points at $0\leq\mu'\leq\mu$, we have*

$$\left\|\frac{x^{\mathrm{cp}}(\mu')}{x^{\mathrm{cp}}(\mu)}+\frac{s^{\mathrm{cp}}(\mu')}{s^{\mathrm{cp}}(\mu)}\right\|_\infty\leq n\,.$$

A. Predictor-Corrector Methods

Given $z=(x,s)\in\mathcal{P}_{++}\times\mathcal{D}_{++}$, the search directions commonly used in interior-point methods are obtained as the solution $(\Delta x,\Delta s)$ to the following linear system for some $\nu\in[0,1]$.

$$\Delta x\in W \tag{6}$$

$$\Delta s\in W^\perp \tag{7}$$

$$s\Delta x+x\Delta s=\nu\mu\mathbf{1}-xs \tag{8}$$

Predictor-corrector methods, such as the Mizuno–Todd–Ye Predictor-Corrector (MTY P-C) algorithm [45], alternate between two types of steps. In *predictor steps*, we use $\nu=0$. This direction is also called the *affine scaling direction*, and will be denoted as $\Delta z^{\mathrm{a}}=(\Delta x^{\mathrm{a}},\Delta s^{\mathrm{a}})$ throughout. In *corrector steps*, we use $\nu=1$. This gives the *centrality direction*, denoted as $\Delta z^{\mathrm{c}}=(\Delta x^{\mathrm{c}},\Delta s^{\mathrm{c}})$.

In the predictor steps, we make progress along the central path. Given the search direction on the current iterate $z=(x,s)\in\mathcal{N}(\beta)$, the step-length is chosen maximal such that we remain in $\mathcal{N}(2\beta)$, i.e.

$$\alpha^{\mathrm{a}}\coloneqq\sup\{\alpha\in[0,1]:\forall\alpha'\in[0,\alpha]: z+\alpha'\Delta z^{\mathrm{a}}\in\mathcal{N}(2\beta)\}.$$

Thus, we obtain a point $z^+=z+\alpha^{\mathrm{a}}\Delta w^{\mathrm{a}}\in\mathcal{N}(2\beta)$. The corrector step finds a next iterate $z^{\mathrm{c}}=z^++\Delta z^{\mathrm{c}}$, where Δz^{c} is the centrality direction computed at z^+. The next proposition summarizes well-known properties, see e.g. [19, Section 4.5.1].

Proposition II.2. *Let $z=(x,s)\in\mathcal{N}(\beta)$ for $\beta\in(0,1/4]$.*

(i) For the affine scaling step, we have $\overline{\mu}(z^+) = (1 - \alpha^{\mathrm{a}})\overline{\mu}(z)$.

(ii) The affine scaling step-length is

$$\alpha^{\mathrm{a}} \geq \max\left\{\frac{\beta}{\sqrt{n}}, 1 - \frac{\|\Delta x^{\mathrm{a}}\Delta s^{\mathrm{a}}\|}{\beta\overline{\mu}(z)}\right\}.$$

(iii) For $z^+ \in \mathcal{N}(2\beta)$, and $z^{\mathrm{c}} = z^+ + \Delta w^{\mathrm{c}}$, we have $\overline{\mu}(z^{\mathrm{c}}) = \overline{\mu}(z^+)$ and $z^{\mathrm{c}} \in \mathcal{N}(\beta)$.

(iv) After a sequence of $O(\sqrt{n}t)$ predictor and corrector steps, we obtain an iterate $z' = (x', s') \in \mathcal{N}(\beta)$ such that $\overline{\mu}(z') \leq \overline{\mu}(z)/2^t$.

a) Minimum-norm viewpoint: We introduce some useful notation for the algorithm, and derive the minimum-norm interpretation of the affine scaling steps. For $z = (x, s) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$, we let

$$\begin{aligned} \xi(z) &= \frac{x^{1/2}s^{1/2}}{\overline{\mu}(z)^{1/2}} \in \mathbb{R}^n, \\ \hat{x} = x\xi^{-1}(z) &= x^{1/2}s^{-1/2}\overline{\mu}(z)^{1/2} \in \mathbb{R}^n, \\ \hat{s} = s\xi^{-1}(z) &= s^{1/2}x^{-1/2}\overline{\mu}(z)^{1/2} \in \mathbb{R}^n. \end{aligned} \tag{9}$$

If clear from the context, we simply use ξ. If $z = (x, s)$ falls on the central path, that is, $xs = \overline{\mu}(z)\mathbf{1}$, then $\xi(z) = \mathbf{1}$, $\hat{x} = x$ and $\hat{s} = s$. The variables $\hat{x}$ and $\hat{s}$ represent natural adjustments for points off the central path.

We will frequently use the rescaled subspaces $\hat{x}^{-1}W$ and $\hat{s}^{-1}W^\perp$ that correspond to using the local geometry at the point $z = (x, s)$. Throughout, we will refer to $\|w/\hat{x}\|$ and $\|w/\hat{s}\|$ as the *primal and dual local norms* of the vector $w \in \mathbb{R}^n$ at the point $z = (x, s) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$. The following statement is immediate from the definitions using $\hat{x}\hat{s} = \overline{\mu}(z)\mathbf{1}$.

Proposition II.3. *The subspaces $\hat{x}^{-1}W$ and $\hat{s}^{-1}W^\perp$ are orthogonal.*

Equation (8) for the predictor step ($\nu = 0$) with update direction $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ can be written as

$$x^{-1}\Delta x^{\mathrm{a}} + x^{-1}\Delta s^{\mathrm{a}} = -\mathbf{1}, \tag{10}$$

or equivalently,

$$\hat{x}^{-1}\Delta x^{\mathrm{a}} + \hat{s}^{-1}\Delta s^{\mathrm{a}} = -\xi, \tag{11}$$

which serves the purpose that now $\hat{x}^{-1}\Delta x^{\mathrm{a}} \in \hat{x}^{-1}W$ and $\hat{s}^{-1}\Delta s^{\mathrm{a}} \in \hat{s}^{-1}W^\perp$ are orthogonal vectors (Proposition II.3). Thus, $\hat{x}^{-1}\Delta x^{\mathrm{a}}$ and $\hat{s}^{-1}\Delta s^{\mathrm{a}}$ give an orthogonal decomposition of $-\xi$. This leads to the following formulas:

$$\Delta x^{\mathrm{a}} = -\hat{x}\Pi_{\hat{x}^{-1}W}(\xi) \qquad \Delta s^{\mathrm{a}} = -\hat{s}\Pi_{\hat{s}^{-1}W^\perp}(\xi). \tag{12}$$

Equivalently, we can see $\Delta z^{\mathrm{a}} = (\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ as the optimal solutions of the following minimum-norm problems:

$$\begin{aligned} \Delta x^{\mathrm{a}} &= \operatorname*{arg\,min}_{\Delta x \in W} \|\hat{x}^{-1}(x + \Delta x)\|, \\ \Delta s^{\mathrm{a}} &= \operatorname*{arg\,min}_{\Delta s \in W^\perp} \|\hat{s}^{-1}(s + \Delta s)\|. \end{aligned} \tag{13}$$

A further equivalent way to express these movement directions is by projections in the rescaled subspaces $\hat{x}^{-1}W$ and $\hat{s}^{-1}W^\perp$; this viewpoint will be used in Section IV.

$$\Delta x^{\mathrm{a}} = \hat{x} \operatorname*{arg\,min}_{\delta \in \hat{x}^{-1}W} \|\xi + \delta\|, \qquad \Delta s^{\mathrm{a}} = \hat{s} \operatorname*{arg\,min}_{\delta \in \hat{s}^{-1}W^\perp} \|\xi + \delta\|. \tag{14}$$

The equivalence of the two forms follows by noting that $\hat{x}^{-1}x = \hat{s}^{-1}s = \xi$.

b) Step-length estimates: We will also need good estimates on the size on predictor steps beyond affine scaling. Our main estimate in this regard is given below.

Lemma II.4 (Step-length estimate for general directions)**.** *Let $z = (x, s) \in \mathcal{N}(\beta)$, $\beta \in (0, 1/6]$. Consider directions $\Delta x \in W$, $\Delta s \in W^\perp$ that satisfy $\|\Delta x\Delta s\| \leq \beta\mu/4$. Let*

$$\gamma = \frac{\|(x + \Delta x)(s + \Delta s)\|}{\mu}.$$

For $0 \leq \alpha \leq 1 - \frac{4\gamma}{\beta}$, then $(x + \alpha\Delta x, s + \alpha\Delta s) \in \mathcal{N}(2\beta)$ and $\overline{\mu}(x + \alpha\Delta x, s + \alpha\Delta s) \leq (1 + \frac{3}{2}\beta/\sqrt{n})(1 - \alpha)\mu$.

B. Lifting Maps

Our algorithm in Section IV uses a layered least squares step that consists of solving a minimum-norm point problem in a smaller subspace first, and then extending it to the entire space. A crucial operation in both computing the layered step and in identifying the appropriate subspaces is the lifting map defined next.

Definition II.5. *Given a partition $I \cup J = [n]$ and a subspace $W \subseteq \mathbb{R}^n$, we define the lifting map $L_I^W : \mathbb{R}^I \to W \subseteq \mathbb{R}^n$ as follows:*

$$L_I^W(x) := \arg\min\{\|w\| : w \in W, w_I = \Pi_{\pi_I(W)}(x)\}. \tag{15}$$

We further define $\ell_I^W : \mathbb{R}^I \to \mathbb{R}^J$ by

$$\ell_I^W(x) := (L_I^W(x))_J = \pi_J\left(L_I^W(x)\right). \tag{16}$$

Note that if $x \in \pi_I(W)$, $w = L_I^W(x)$ is the minimum-norm point in W with $w_I = x$. The following lists the key properties of the lifting map.

Lemma II.6. *For a partition $I \cup J = [n]$ and a linear subspace W, $L_I^W : \mathbb{R}^I \to W$ and $\ell_I^W : \mathbb{R}^I \to \mathbb{R}^J$ are linear maps. Moreover, for $x \in \mathbb{R}^I$, $w = L_I^W(x)$ is the unique solution to the following linear system:*

$$w \in W, \quad w_I \in \pi_I(W)^\perp + x, \quad w_J \in \pi_J(W^\perp). \tag{17}$$

We now give the fundamental duality relation between lifting maps. For this purpose, we define $\ell_I^{W*} : \mathbb{R}^J \to \mathbb{R}^I$ to denote the adjoint of ℓ_I^W, namely, the map satisfying $\langle \ell_I^{W*}(y), x\rangle = \langle y, \ell_I(x)\rangle$, $\forall y \in \mathbb{R}^J, x \in \mathbb{R}^I$. If we are expressing ℓ_I^W as matrix $M \in \mathbb{R}^{J \times I}$, then ℓ_I^{W*} is represented by $M^\top$. That is, if $\ell_I^W(x) = Mx$, then $\ell_J^{W*}(y) = M^\top y$.

Lemma II.7. *For a partition $I \cup J = [n]$ and a linear subspace W, we have that $\ell_I^W = \ell_J^{W^\perp *}$. In particular, the non-zero singular values of ℓ_I^W and $\ell_J^{W^\perp *}$ are the same.*

III. Polarization of the Central Path

A segment of the central path $\mathrm{CP}[\mu_1, \mu_0] := \{z(\mu) : \mu \in [\mu_1, \mu_0]\}$, $0 \le \mu_0 < \mu_1$, is polarized if it admits a partition $B \cup N = [n]$ such that the primal variables in B are essentially fixed and those in N are scaling down linearly with the parameter μ (vice versa for the dual variables).

Definition III.1 (Polarization). *For $\gamma \in [0,1]$ and $\mu_0 \ge \mu_1 \ge 0$, we say that the segment $\mathrm{CP}[\mu_1, \mu_0]$ is* γ-polarized *if there exists a partition $B \cup N = [n]$ such that for all $\mu \in [\mu_1, \mu_0]$:*

$$x_i^{\mathrm{cp}}(\mu) \ge \gamma x_i^{\mathrm{cp}}(\mu_0)\,, \quad \forall i \in B\,,$$
$$s_i^{\mathrm{cp}}(\mu) \ge \gamma s_i^{\mathrm{cp}}(\mu_0)\,, \quad \forall i \in N\,.$$

As a direct consequence of the definition together with near-monotonicity (Lemma II.1), we deduce the following crucial corollary:

Corollary III.2. *Let $\mathrm{CP}[\mu_1, \mu_0]$, $0 \le \mu_1 \le \mu_0$, be γ-polarized with respect to the partition $B \cup N = [n]$. Then, for all $\mu \in [\mu_1, \mu_0]$, the following holds:*

(1) $\gamma x_i(\mu_0) \le x_i(\mu) \le n x_i(\mu_0)$, $i \in B$.
(2) $\gamma s_i(\mu_0) \le s_i(\mu) \le n s_i(\mu_0)$, $i \in N$.
(3) $\frac{\mu}{n\mu_0} x_i(\mu_0) \le x_i(\mu) \le \frac{\mu}{\gamma\mu_0} x_i(\mu_0)$, $i \in N$.
(4) $\frac{\mu}{n\mu_0} s_i(\mu_0) \le s_i(\mu) \le \frac{\mu}{\gamma\mu_0} s_i(\mu_0)$, $i \in B$.

Section IV introduces the algorithm IPM with subspace LLS that can traverse γ-polarized segments in $O(n^{1.5}\log(n/\gamma))$ iterations. Theorem I.1 follows by combining this algorithm with the following decomposition result that is the main result of this section.

Theorem III.3. *Let $\Gamma : (\mu_1, \mu_0) \to \mathcal{N}^{-\infty}(\theta)$, $\theta \in (0,1)$, $0 \le \mu_1 \le \mu_0 \le \infty$, be a piecewise linear curve satisfying $\overline{\mu}(\Gamma(\mu)) = \mu$, $\forall \mu \in (\mu_1, \mu_0)$ consisting of T linear segments. Then, $\mathrm{CP}[\mu_1, \mu_0]$ can be decomposed into T segments that are $\frac{(1-\theta)^2}{16n^3}$-polarized.*

This is a direct consequence of the following key lemma.

Lemma III.4. *For $\theta \in (0,1)$, let $[z^{(0)}, z^{(1)}] \subseteq \mathcal{N}^{-\infty}(\theta)$, $\overline{\mu}(z^{(0)}) > \overline{\mu}(z^{(1)})$. Then, $\mathrm{CP}[\overline{\mu}(z^{(1)}), \overline{\mu}(z^{(0)})]$ is $\frac{(1-\theta)^2}{16n^3}$-polarized.*

One should read this last statement as saying that a segment of the central path is 'approximately linear' if and only if it is polarized. The link between polarization and linearity is surprisingly elementary; it follows from the analysis of the inequalities of the wide neighborhood (2):

$$\begin{aligned}((1-\alpha)x_i(\mu_0) + \alpha x_i(\mu_1))\ ((1-\alpha)s_i(\mu_0) + \alpha s_i(\mu_1))\\ \ge (1-\theta)((1-\alpha)\mu_0 + \alpha\mu_1)\,, \forall \alpha \in [0,1]\,,\ i \in [n],\end{aligned}$$

where we recall that $z(\mu_0) = (x(\mu_0), s(\mu_0)), z(\mu_1) = (x(\mu_1), s(\mu_1))$. For example, if $\theta = 0$, it is not hard to check that for each $i \in [n]$, one must have either $x_i(\mu_0) = x_i(\mu_1)$ and $s_i(\mu_1) = \frac{\mu_1}{\mu_0} s_i(\mu_0)$ (i.e., $i \in B$) or $x_i(\mu_0) = \frac{\mu_1}{\mu_0} x_i(\mu_1)$ and $s_i(\mu_1) = s_i(\mu_0)$ (i.e., $i \in N$).

We note that polarization plays an important if implicit role in prior layered least squares analyses [9], [11], [39]. In particular, the 'long and straight' segments in these works are all polarized. What was unclear in these works, however, is whether polarization *by itself* was sufficient to make a segment easy to traverse. Indeed, these works all crucially rely upon numerical condition numbers of the instance which can be effectively unbounded in the present context. Beyond the LLS context, we are further unaware of central path analyses exploiting the tight connection between approximate linearity and polarization, and we hope this will encourage future study.

As is clear from the definition, polarization provides us extremely useful 'long-range' control over the evolution of variables on a segment. We note that γ-polarization is mostly interesting when the segment itself is *long*, namely, when $\mu_0/\mu_1 \gg 1/\gamma$. We now explain how to leverage this control to traverse any γ-polarized segment using *subspace LLS steps*.

IV. The Subspace Layered Least Squares IPM

In this section, we introduce the algorithm IPM with subspace LLS (Algorithm 1) and prove the following result.

Theorem IV.1. *Let $\mathrm{CP}[\mu_1, \mu_0]$ be γ-polarized. Then, given an iterate $z \in \mathcal{N}(\beta)$ with parameter $\overline{\mu}(z) \in (\mu_1, \mu_0)$, the algorithm* IPM with subspace LLS *(Algorithm 1) takes $O\left(n^{1.5}\log(n/\gamma)\right)$ many iterations to find $z' \in \mathcal{N}(\beta)$ such that $\overline{\mu}(z') \le \mu_1$.*

We stress that IPM with subspace LLS does not have any information about the polarization of $\mathrm{CP}[\mu_1, \mu_0]$, but only defines the steps using local information. Theorem I.1 directly follows from Theorem III.3 and Theorem IV.1.

A. Overview: Traversing a Polarized Segment

Before the more technical description, we highlight the main intuition behind our algorithm and compare it to the Trust Region IPM. Let $\mathrm{CP}[\mu_1, \mu_0]$, $0 \le \mu_1 < \mu_0$, be a γ-polarized segment with partition $B \cup N = [n]$.

For simplicity of presentation, let us assume that given any iterate (x, s) in the narrow neighborhood $\mathcal{N}(1/6)$ used in our algorithm, we can jump to the exact central path point $z(\mu) \in \mathrm{CP} = \mathcal{N}(0)$ with $\mu = \overline{\mu}(x, s)$ for free. Let us further assume that the algorithm knows the partition B, N (we discuss how to effectively compute it at the end) and that we are given the starting point $z^{(0)} := z(\mu_0)$.

Our abstract algorithm will thus compute iterates $z^{(0)}, z^{(1)}, \ldots$ on the central path CP with $\overline{\mu}(z^{(0)}) > \overline{\mu}(z^{(1)}) > \ldots$. To move from $z^{(t)}$ to $z^{(t+1)}$, we first compute a *movement direction*

$$\Delta z^{(t)} = (\Delta x^{(t)}, \Delta s^{(t)}) \in \ker(A) \times \mathrm{im}(A^\top) =: W \times W^\perp,$$

together with a step-length $\alpha^{(t)} \in [0,1]$, chosen such that $z^{(t)} + \alpha\Delta z^{(t)} \in \mathcal{N}(1/6)$, $0 \le \alpha \le \alpha^{(t)}$. Lastly, we jump for free to $z^{(t+1)} \in \mathrm{CP}$ satisfying $\overline{\mu}(z^{t+1}) = \overline{\mu}(z^{(t)} + \alpha^{(t)}\Delta z^{(t)})$.

Given this setup, our goal is to compute movement directions, such that after $k = O(n^{1.5}\log(n/\gamma))$ iterations, we have $\overline{\mu}(z^{(k)}) \le \mu_1$, i.e., that we have crossed the segment. We would like to emphasize that our algorithm will in fact compute the movement direction $\Delta z^{(t)}$ using only *local information* at $z^{(t)}$, without any explicit knowledge of the polarized segment.

A natural movement direction is *affine scaling* used in predictor-corrector methods, see Section II-A. This direction guarantees $1-\Omega(1/\sqrt{n})$ decrease in normalized duality gap per step. Hence, if $\mu_0/\mu_1 \le \mathrm{poly}(n, 1/\gamma)$, then simply using $\sqrt{n}\ln(\mu_0/\mu_1)$ affine scaling iterations is sufficient for our purposes.

Thus, we may assume that $\mu_0/\mu_1 \gg \mathrm{poly}(n, 1/\gamma)$. In this case, we show that computing an affine scaling direction $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ at the current iterate $(x^{(t)}, s^{(t)})$ reveals the correct partition $B \cup N = [n]$ of the current polarized segment. This is because the standard affine scaling step itself exhibits a polarized behaviour: we can simply select B as the set of coordinates i where $|\Delta x_i^{\mathrm{a}}/x_i^{(t)}| < |\Delta s_i^{\mathrm{a}}/s_i^{(t)}|$, i.e., the relative primal movement is smaller than the relative dual movement (see Definition IV.4).

a) Trust Region Programs and Subspace LLS: The trust region programs by Lan, Monteiro, Tsuchiya [10] provide a good starting point for defining our movement direction $\Delta z^{(t)} = (\Delta x^{(t)}, \Delta s^{(t)}) \in W \times W^\perp$ from an iterate $z^{(t)} = (x^{(t)}, s^{(t)}) \in \mathrm{CP}[\mu_1, \mu_0]$ and a given a partition $[n] = B \cup N$:

$$\begin{aligned}
&\min_{\Delta x \in W} \{\|(x_N^{(t)} + \Delta x_N)/x_N^{(t)}\| \,:\, \|\Delta x_B/x_B^{(t)}\| \le \varrho\} \\
&\min_{\Delta s \in W^\perp} \{\|(s_B^{(t)} + \Delta s_B)/s_B^{(t)}\| \,:\, \|\Delta s_N/s_N^{(t)}\| \le \varrho\}
\end{aligned} \tag{18}$$

where $\varrho = 1/100$ is sufficient for our purposes. We use the notation $\Delta x/x^{(t)} := (\Delta x_1/x_1^t, \dots, \Delta x_n/x_n^{(t)})$ and similarly for $\Delta s/s^{(i)}$. The norms $\|x/x^{(t)}\|$ and $\|s/s^{(t)}\|$ are the so-called primal and dual *local norms* at $x^{(t)}$ and $s^{(t)}$. By definition, the optimal primal trust region direction Δx^* achieves a maximal multiplicative decrease on the coordinates in N while 'barely moving' the coordinates in B as measured in the local norm. The optimal dual direction Δs^* achieves the same on the dual side with the role of N and B swapped.

Note that these directions mesh very well with polarization of the segment $\mathrm{CP}[\mu_1, \mu_0]$. In particular, they reflect the idea that the coordinates of $x(\mu)$ in N should be linearly scaling down while those in B are staying mostly fixed, and vice versa for $s(\mu)$. As shown in [10], moving in any direction $\Delta z^{(t)} = (\Delta x^{(t)}, \Delta s^{(t)})$ corresponding to feasible solutions to (18), the normalized duality gap can be reduced as

$$\frac{\overline{\mu}(z^{(t+1)})}{\overline{\mu}(z^{(t)})} \le \|(x_N^{(t)} + \Delta x_N^{(t)})/x_N^{(t)}\| + \|(s_B^{(t)} + \Delta s_B^{(t)})/s_B^{(t)}\|. \tag{19}$$

That is, we can achieve a drop that corresponds to the sum of primal and dual objective values.

In many ways, the trust region direction can be seen as the 'optimal' movement direction. However, [10] solves the quadratic convex programs in (18) in weakly polynomial time with dependence on the vectors b and c in (LP). It is not known whether a strongly polynomial algorithm (with dependence only on n) exists. Further, the analysis in [10] relies on combinatorial progress measures adapted from the LLS analyses. It remains unclear how to analyze the convergence of the trust region steps by using only the fact that they are maximally long.

Instead of optimally solving (18), we introduce what we call *subspace LLS steps* that yield 'good enough' approximate solutions for our purposes. We restrict the set of primal and dual directions to come from carefully selected subspaces $V^{(t)} \subseteq W$ and $U^{(t)} \subseteq W^\perp$ satisfying:

$$\|\Delta x_B/x_B^{(t)}\| \le \tau \|\Delta x_N/x_N^{(t)}\|, \quad \forall \Delta x \in V^{(t)} \tag{20}$$

$$\|\Delta s_N/s_N^{(t)}\| \le \tau \|\Delta s_B/s_B^{(t)}\|, \quad \forall \Delta s \in U^{(t)}, \tag{21}$$

where we set $\tau = \varrho/(16\sqrt{n})$. We call any such subspaces $U^{(t)}, V^{(t)}$ *cheap lift subspaces*. Note that every such solution automatically satisfies the constraints in program (18). Hence, the optimal solutions can be computed by solving systems of linear equations that correspond to minimum-norm points in the local norms.

In terms of the choice of subspaces, there is quite a lot of flexibility. A canonical choice, which we use for simplicity in the analysis, consists of choosing a space spanned by the singular vectors of a certain 'lifting map' (in the sense of Section II-B) whose corresponding singular values are at most τ.

B. The algorithm

We now formally introduce *subspace layered least squares update direction* and describe our main algorithm. At a given point $z \in \mathcal{N}(\beta)$, this step direction is specified by a partition $B \cup N = [n]$ and two subspaces $V \subseteq \pi_N(\hat{x}^{-1}W)$ and $U \subseteq \pi_B(\hat{s}^{-1}W^\perp)$.

Recall the notation ξ for the local error scaling and the thereby adjusted versions $\hat{x}$ and $\hat{s}$ of x and s defined in (9). For a given partition $B \cup N = [n]$ such that $B, N \neq \emptyset$, recall the lifting maps introduced in Section II-B. We use the following shorthand notation

$$\ell_{z,N} := \ell_N^{\hat{x}^{-1}W} \qquad \text{and} \qquad \ell_{z,B}^\perp := \ell_B^{\hat{s}^{-1}W^\perp}.$$

Thus, the linear map $\ell_{z,N} \colon \mathbb{R}^N \to \mathbb{R}^B$ computes a minimum-norm lift in the rescaling $\hat{x}^{-1}W$ corresponding to the local geometry at the point z. Recall from Lemma II.7 that the linear map $\ell_{z,B}^\perp \colon \mathbb{R}^B \to \mathbb{R}^N$ is the adjoint of $\ell_{z,N}$, thus, they are represented by matrices that are transposed of each other.

Definition IV.2 (Subspace LLS direction)**.** *Let $z \in \mathcal{N}(\beta)$, and let ξ, $\hat{x}$, $\hat{s}$ be defined as in* (9)*. Assume we are given a partition $B \cup N = [n]$ and two subspaces $V \subseteq \pi_N(\hat{x}^{-1}W)$ and $U \subseteq \pi_B(\hat{s}^{-1}W^\perp)$. The* Subspace LLS *update direction $(\Delta x^\ell, \Delta s^\ell) \in W \times W^\perp$ at z with respect to (B, N, U, V) is defined as follows. If $B, N \neq \emptyset$, then*

$$\begin{aligned}
\delta^V &:= \operatorname*{arg\,min}_{\delta \in V} \|\xi_N + \delta\|, & \delta^U &:= \operatorname*{arg\,min}_{\delta \in U} \|\xi_B + \delta\|, \\
\delta^x &:= \left(\ell_{z,N}(\delta^V), \delta^V\right), & \delta^s &:= \left(\delta^U, \ell_{z,B}^\perp(\delta^U)\right), \\
\Delta x^\ell &:= \hat{x}\delta^x, & \Delta s^\ell &:= \hat{s}\delta^s.
\end{aligned}$$

If $N = \emptyset$, then we let $(\Delta x^\ell, \Delta s^\ell) = (\mathbf{0}, \Delta s^{\mathrm{a}})$ and if $B = \emptyset$, then we let $(\Delta x^\ell, \Delta s^\ell) = (\Delta x^{\mathrm{a}}, \mathbf{0})$.

The formula defining δ^V is similar to the definition (14) of the affine scaling direction. However, when defining δ^V, we restrict ourselves to norm minimization in $\pi_N(\hat{x}^{-1}W)$, and within that, we require $\delta^V \in V$. This step is then extended to the coordinates in B using a minimum norm lift in the rescaled subspace $\hat{x}^{-1}W$.

Note that we can equivalently write $\delta^V = -\Pi_V(\xi_N)$ and $\delta^U = -\Pi_U(\xi_B)$. An equivalent definition of Δx^ℓ, using the primal local norm in the original space W—similarly to (13)—is the following:

$$\begin{aligned}\Delta v^\ell &:= \arg\min\left\{\left\|\hat{x}_N^{-1}(x_N + \Delta v)\right\| \,:\, \Delta v \in \hat{x}V\right\}\,,\\ \Delta x^\ell &:= \arg\min\left\{\left\|\hat{x}^{-1}\Delta x\right\| \,:\, \Delta x_N = \Delta v^\ell\,, \Delta x \in W\right\}\,.\end{aligned} \tag{22}$$

Analogous formulas can be given for Δs^ℓ.

For the subspace LLS direction as in Definition IV.2, we define the residuals

$$\varrho^V := \xi + \delta^x = \frac{x + \Delta x^\ell}{\hat{x}}\,,\quad \varrho^U := \xi + \delta^s = \frac{s + \Delta s^\ell}{\hat{s}}\,. \tag{23}$$

a) Cheap lifts and singular values: We now turn to the selection of the subspaces V and U for a given partition (B, N). Simply selecting $V = \pi_N(\hat{x}^{-1}W)$ and $U = \pi_B(\hat{s}^{-1}W^\perp)$ would attain the smallest possible objective values as in (19); however, the constraints bounding the local norms of Δx_B^ℓ and Δs_N^ℓ in (18) could be arbitrarily violated. Therefore, we select the subspaces V and U such that no matter how δ^V and δ^U are selected, the lifts $\Delta x_B^\ell = \hat{x}_N \ell_{z,N}(\delta^V)$ and $\Delta s_N^\ell = \hat{s}_N \ell_{z,B}^\perp(\delta^U)$ will have small local norms; that is, $\|\ell_{z,N}(\delta^V)\|$ and $\|\ell_{z,B}^\perp(\delta^U)\|$ are suitably bounded.

Using that $\|\delta^V\| \le \|\xi_N\|$ as δ^V is the projection of ξ_N, and the definition of ξ, we get $\|\delta^V\| \le \|\xi_N\| \le \sqrt{n}$. Hence, if we can guarantee that the lifting map satisfies $\|\ell_{z,N}(\delta)\| \le \tau\|\delta\|$ for all $\delta \in V$ for a suitably small threshold τ, we can guarantee Δx^ℓ to be feasible to the primal trust region program (18); analogous arguments can be made for the dual direction. Naturally, we would like to select the largest subspaces V and U with this property.

Let us define the threshold

$$\tau := \frac{\beta}{16\sqrt{n}}\,. \tag{24}$$

Our goal is to ensure that

$$\|\ell_{z,N}(\delta)\| \le \tau\|\delta\|\,, \forall \delta \in V \text{ and} \|\ell_{z,B}^\perp(\delta)\| \le \tau\|\delta\|\,, \forall \delta \in U\,. \tag{25}$$

This holds if V is the subspace spanned by the singular vectors of the map $\ell_{z,N}$ corresponding to the singular values $\le \tau$ in absolute value; analogously for $\ell_{z,B}^\perp$ and U. In the following definition, we use a full Singular Value Decomposition (SVD) of the matrix $\mathbf{M} \in \mathbb{R}^{|B|\times|N|}$ representing the map $\ell_{z,N}$. The SVD gives $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{|B|\times|B|}$ and $\mathbf{V} \in \mathbb{R}^{|N|\times|N|}$ are orthogonal matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{|B|\times|N|}$ is a rectangular diagonal matrix also including the zero singular values.

Definition IV.3 (Cheap Subspaces)**.** *For a partition $B \cup N = [n]$ with $B, N \neq \emptyset$ and $z \in \mathcal{N}(\beta)$, consider an SVD decomposition*

$$\ell_{z,N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\,,\quad \ell_{z,B}^\perp = \mathbf{V}\mathbf{\Sigma}^\top\mathbf{U}^\top\,.$$

Depending on a threshold parameter $t \in \mathbb{R}_{>0}$, we set

$V_{z,N}(t) := \mathrm{im}(\mathbf{V}_S) \cap \pi_N(\hat{x}^{-1}W)$ *for* $S := \{i : \|\mathbf{\Sigma}_{\cdot,i}\| \le t\}$,
$U_{z,B}(t) := \mathrm{im}(\mathbf{U}_T) \cap \pi_B(\hat{s}^{-1}W^\perp)$ *for* $T := \{i : \|\mathbf{\Sigma}_{i,\cdot}\| \le t\}$.

Further, we let $\overline{V}_{z,N}(t)$ be the orthogonal complement of $V_{z,N}(t)$ in $\pi_N(\hat{x}^{-1}W)$, that is, $V_{z,N}(t) \oplus \overline{V}_{z,N}(t) = \pi_N(\hat{x}^{-1}W)$ and $V_{z,N}(t) \perp \overline{V}_{z,N}(t)$. Analogously, we define $\overline{U}_{z,B}(t)$ such that $U_{z,B}(t) \oplus \overline{U}_{z,B}(t) = \pi_B(\hat{s}^{-1}W^\perp)$ and $U_{z,B}(t) \perp \overline{U}_{z,B}(t)$.

Note that $\|\mathbf{\Sigma}_{\cdot,i}\|$ and $\|\mathbf{\Sigma}_{\cdot,i}\|$ are the absolute values of the singular values corresponding to the i-th columns of $\mathbf{V}$ and $\mathbf{U}$, respectively. It is clear from the definition that $V = V_{z,N}$ and $U = U_{z,B}$ satisfy (25).

In this extended abstract, we define the subspaces $V = V_{z,N}(\tau)$ and $U = U_{z,B}(\tau)$ according to Definition IV.3. However, we note that this requires an exact SVD decomposition of the matrix M representing the lifting map $\ell_{z,N}$. Our arithmetic model does not allow for computing an exact decomposition. Since we would like to implement each iteration of the algorithm in strongly polynomial time, we cannot use a numerical approximation for the SVD decomposition, as such approximations would depend on the norm of the matrix. However, a weaker property suffices for our analysis. Namely, we need that the subspace $V \subseteq \pi_{\widetilde{N}}(\hat{x}^{-1}W)$ has singular values at most τ—and thus satisfies (25)—and further it includes the subspace of right singular vectors corresponding to the singular values $\le \tau/n^c$ for some constant $c > 0$; analogously for U. A strongly polynomial subroutine finding such a subspace is given in the full version.

b) The associated partition: Definition IV.2 and Definition IV.3 are applicable for any partition $B \cup N = [n]$, $B, N \neq \emptyset$ and $z \in \mathcal{N}(\beta)$. Our algorithm chooses a natural partition derived from the relative step lengths in the affine scaling step:

Definition IV.4 (Associated partition)**.** *For $z = (x, s) \in \mathcal{N}(\beta)$, let $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ be the affine scaling step as in* (12)*. Let us define the associated partition $\widetilde{B}_z \cup \widetilde{N}_z = [n]$ as*

$$\widetilde{B}_z := \left\{i : \left|\frac{\Delta x_i^{\mathrm{a}}}{x_i}\right| < \left|\frac{\Delta s_i^{\mathrm{a}}}{s_i}\right|\right\}\qquad \widetilde{N}_z := [n] \setminus \widetilde{B}_z\,.$$

Further, let $\widetilde{\ell}_z := \ell_{z,\widetilde{N}_z}$ and $\widetilde{\ell}_z^\perp := \ell_{z,\widetilde{B}_z}^\perp$ denote the lifting maps corresponding to this partition, assuming $\widetilde{B}_z, \widetilde{N}_z \neq \emptyset$.

The affine scaling step is the canonical candidate for an improving direction. Namely, for each $i \in \widetilde{B}_z$ the variable s_i decreases at a faster rate than x_i, and vice versa for $i \in \widetilde{N}_z$. It turns out that $(\widetilde{B}_z, \widetilde{N}_z)$ reveals the polarizing partition for a sufficiently long polarized segment.

c) Description of the algorithm: We are ready to describe the predictor-corrector algorithm IPM WITH SUBSPACE LLS, shown in Algorithm 1. We are given a starting point

$(x^0, s^0) \in \mathcal{N}(\beta)$. In each iteration, we compute the affine scaling direction $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ and identify the associated partition $(\widetilde{B}, \widetilde{N})$. Using this partition, we select the subspaces U and V according to Definition IV.3. We then compute the subspace LLS direction $(\Delta x^{\ell}, \Delta s^{\ell})$ for $(\widetilde{B}, \widetilde{N}, V, U)$. For both directions, compute the feasible step-lengths according to the bounds in Proposition II.2 and Lemma II.4. We use the better of these two possible steps, and obtain the next iterate after a corrector step.

```
Algorithm 1: IPM WITH SUBSPACE LLS
Input  : Instance of (LP) and initial $(x^0, s^0) \in \mathcal{N}(\beta)$,
         $\beta \in (0, 1/6]$.
Output: Optimal solution $(x^\star, s^\star)$ to (LP).
$x \leftarrow x^0, s \leftarrow s^0$;
while $\langle x, s \rangle > 0$ do
    Compute affine scaling direction $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$;
    Set $\alpha^{\mathrm{a}}$ for $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ according to
      Proposition II.2(ii);
    $(\hat{x}^{\mathrm{a}}, \hat{s}^{\mathrm{a}}) \leftarrow (x + \alpha^{\mathrm{a}} \Delta x^{\mathrm{a}}, s + \alpha^{\mathrm{a}} \Delta s^{\mathrm{a}})$;
    $\widetilde{B} \leftarrow \left\{ i : \left| \frac{\Delta x_i^{\mathrm{a}}}{x_i} \right| < \left| \frac{\Delta s_i^{\mathrm{a}}}{s_i} \right| \right\}, \widetilde{N} \leftarrow [n] \setminus \widetilde{B}$ ;
    $V \leftarrow V_{z,\widetilde{N}}(\tau)$ ; $U \leftarrow U_{z,\widetilde{B}}(\tau)$ according to
      Definition IV.3 ;
    Find subspace LLS direction $(\Delta x^{\ell}, \Delta s^{\ell})$ according
      to Definition IV.2 for $V$ and $U$ ;
    Set $\alpha^{\ell}$ for $(\Delta x^{\ell}, \Delta s^{\ell})$ according to Lemma II.4;
    $(\hat{x}^{\ell}, \hat{s}^{\ell}) \leftarrow (x + \alpha^{\ell} \Delta x^{\ell}, s + \alpha^{\ell} \Delta s^{\ell})$;
    if $\overline{\mu}(\hat{x}^{\mathrm{a}}, \hat{s}^{\mathrm{a}}) \le \overline{\mu}(\hat{x}^{\ell}, \hat{s}^{\ell})$ then
        $(x, s) \leftarrow (\hat{x}^{\mathrm{a}}, \hat{s}^{\mathrm{a}})$
    else
        $(x, s) \leftarrow (\hat{x}^{\ell}, \hat{s}^{\ell})$
    $(\Delta x^c, \Delta s^c) = \textsc{Corrector}(x, s)$;
    $x \leftarrow x + \Delta x^c, s \leftarrow s + \Delta s^c$;
return $(x, s)$;
```

C. Analysis overview

We sketch the key idea in showing that subspace LLS steps can reach the end of the current γ-polarized segment $\mathrm{CP}[\mu_1, \mu_0]$ in $O(n^{1.5} \ln(n/\gamma))$ iterations. Given any iterate $z^{(t)} \in \mathrm{CP}[\mu_1, \mu_0]$, let $U^{(t)}$ and $V^{(t)}$ denote the corresponding subspaces at this iteration. If $\overline{\mu}(z^{(t+k)})$ is still significantly larger than μ_1—i.e., we have not reached the end of the segment—then we show that both $\dim(U^{(t+k)}) > \dim(U^{(t)})$ and $\dim(V^{(t+k)}) > \dim(V^{(t)})$. The overall bound follows since this can occur at most n times.

To get this result, we analyze the evolution of what we call the 'empirical gradient' at $z^{(t)}$, which we define to be

$$\Delta \tilde{z}^{(t)} := z(\mu_1) - z^{(t)} ,$$

i.e., the difference between the current iterate and the end of the segment. A crucial observation is that if $\Delta \tilde{z}^{(t)}$ were a feasible solution to (18), then following this direction would get to within a $\mathrm{poly}(n/\gamma)$ factor for the end of the segment in one step (though we do not know how to compute it). Furthermore, the empirical gradient is never far from being feasible, in particular, it is feasible if the bound of β is replaced by $O(n)$. We show the following dichotomy. Given an iterate $z^{(t)}$, either the empirical gradient $\Delta \tilde{z}^{(t)}$ is mostly "aligned" with the LLS subspaces $U^{(t)} \times V^{(t)}$, in which case we get close to the end of the segment in one step, or we can extract from $\Delta \tilde{z}^{(t)}$ an additional "cheap lift" dimension in the next $O(\sqrt{n} \ln(n/\gamma))$ iterations. In the latter case, we use the polarization property to analyze the evolution of the singular values of lifting maps.

References

[1] V. Klee and G. J. Minty, "How good is the simplex algorithm?," in *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pp. 159–175, New York: Academic Press, 1972. 1, 2

[2] L. G. Khachiyan, "A polynomial algorithm in linear programming," in *Doklady Academii Nauk SSSR*, vol. 244, pp. 1093–1096, 1979. 1

[3] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 302–311, 1984. 1, 4

[4] N. Megiddo, "Is binary encoding appropriate for the problem-language relationship?," *Theoretical Computer Science*, vol. 19, no. 3, pp. 337–341, 1982. 1

[5] R. C. Grinold, "The payment scheduling problem," *Naval Research Logistics Quarterly*, vol. 19, pp. 123–136, 1972. 2

[6] S. Smale, "Mathematical problems for the next century," *The Mathematical Intelligencer*, vol. 20, pp. 7–15, 1998. 2

[7] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming," *Mathematical Programming*, vol. 40, no. 1-3, pp. 59–93, 1988. 2, 4

[8] S. A. Vavasis and Y. Ye, "Condition numbers for polyhedra with real number data," *Operations Research Letters*, vol. 17, pp. 209–214, 06 1995. 2

[9] S. A. Vavasis and Y. Ye, "A primal-dual interior point method whose running time depends only on the constraint matrix," *Mathematical Programming*, vol. 74, no. 1, pp. 79–120, 1996. 2, 4, 5, 7

[10] G. Lan, R. D. Monteiro, and T. Tsuchiya, "A polynomial predictor-corrector trust-region algorithm for linear programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1918–1946, 2009. 2, 4, 5, 8

[11] D. Dadush, S. Huiberts, B. Natura, and L. A. Végh, "A scaling-invariant algorithm for linear programming whose running time depends only on the constraint matrix," in *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 761–774, 2020. 2, 5, 7

[12] X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig, "Log-barrier interior point methods are not strongly polynomial," *SIAM Journal on Applied Algebra and Geometry*, vol. 2, no. 1, pp. 140–178, 2018. 2, 4, 5

[13] X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig, "What tropical geometry tells us about the complexity of linear programming," *SIAM Review*, vol. 63, pp. 123–164, Jan. 2021. 2, 4, 5

[14] X. Allamigeon, S. Gaubert, and N. Vandame, "No self-concordant barrier interior point method is strongly polynomial," in *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 515–528, 2022. 2, 4, 5

[15] S. Gass and T. Saaty, "The computational algorithm for the parametric objective function," *Naval Res. Logist. Quart.*, vol. 2, pp. 39–45, 1955. 2

[16] K. H. Borgwardt, *The Simplex Method: a probabilistic analysis*, vol. 1. Springer Science & Business Media, 2012. 2

[17] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM (JACM)*, vol. 51, no. 3, pp. 385–463, 2004. 2

[18] D. Dadush and S. Huiberts, "Smoothed analysis of the simplex method," in *Beyond the Worst-Case Analysis of Algorithms* (T. Roughgarden, ed.), Cambridge: Cambridge University Press, 2020. 2

[19] Y. Ye, *Interior-Point Algorithms: Theory and Analysis*. New York: John Wiley and Sons, 1997. 4, 5

[20] R. D. C. Monteiro and T. Tsuchiya, “A variant of the Vavasis-Ye layered-step interior-point algorithm for linear programming,” *SIAM Journal on Optimization*, vol. 13, no. 4, pp. 1054–1079, 2003. 4

[21] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994. 4

[22] Y. T. Lee and M.-C. Yue, “Universal barrier is n-self-concordant,” *Mathematics of Operations Research*, vol. 46, no. 3, pp. 1129–1148, 2021. 4

[23] P. M. Vaidya, “Speeding-up linear programming using fast matrix multiplication,” in *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pp. 332–337, 1989. 4

[24] S. Bubeck and R. Eldan, “The entropic barrier: a simple and optimal universal self-concordant barrier.” arXiv preprint arXiv:1412.1587, 2014. 4

[25] Y. T. Lee and A. Sidford, “Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\mathrm{rank}})$ iterations and faster algorithms for maximum flow,” in *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 424–433, 2014. 4

[26] Y. T. Lee and A. Sidford, “Solving linear programs with $\tilde{O}(\sqrt{\mathrm{rank}})$ linear system solves.” arXiv preprint arXiv:1910.08033, 2019. 4

[27] M. B. Cohen, Y. T. Lee, and Z. Song, “Solving linear programs in the current matrix multiplication time,” in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 938–942, 2019. 4

[28] J. van den Brand, “A deterministic linear program solver in current matrix multiplication time,” in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 259–278, SIAM, 2020. 4

[29] J. van den Brand, Y. Tat Lee, A. Sidford, and Z. Song, “Solving tall dense linear programs in nearly linear time,” in *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 775–788, 2020. 4

[30] D. A. Spielman and S.-H. Teng, “Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems,” in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, 2004. 4

[31] K. Axiotis, A. Madry, and A. Vladu, “Faster sparse minimum cost flow by electrical flow localization,” in *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 528–539, 2022. 4

[32] S. I. Daitch and D. A. Spielman, “Faster approximate lossy generalized flow via interior point algorithms,” in *Proceedings of the 40th annual ACM symposium on Theory of Computing (STOC)*, pp. 451–460, 2008. 4

[33] Y. Gao, Y. P. Liu, and R. Peng, “Fully dynamic electrical flows: Sparse maxflow faster than Goldberg–Rao,” in *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 516–527, IEEE, 2022. 4

[34] A. 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*, pp. 253–262, IEEE, 2013. 4

[35] J. van den Brand, Y.-T. Lee, D. Nanongkai, R. Peng, T. Saranurak, A. Sidford, Z. Song, and D. Wang, “Bipartite matching in nearly-linear time on moderately dense graphs,” in *61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 919–930, 2020. 4

[36] J. van den Brand, Y. T. Lee, Y. P. Liu, T. Saranurak, A. Sidford, Z. Song, and D. Wang, “Minimum cost flows, MDPs, and L1-regression in nearly linear time for dense instances,” in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 859–869, 2021. 4

[37] L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. P. Gutenberg, and S. Sachdeva, “Maximum flow and minimum-cost flow in almost-linear time,” in *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 612–623, IEEE, 2022. 4

[38] N. Megiddo, S. Mizuno, and T. Tsuchiya, “A modified layered-step interior-point algorithm for linear programming,” *Mathematical Programming*, vol. 82, no. 3, pp. 339–355, 1998. 4

[39] R. D. C. Monteiro and T. Tsuchiya, “A new iteration-complexity bound for the MTY predictor-corrector algorithm,” *SIAM Journal on Optimization*, vol. 15, no. 2, pp. 319–347, 2005. 4, 7

[40] G. Sonnevend, J. Stoer, and G. Zhao, “On the complexity of following the central path of linear programs by linear extrapolation II,” *Mathematical Programming*, vol. 52, no. 1-3, pp. 527–553, 1991. 5

[41] R. D. Monteiro and T. Tsuchiya, “A strong bound on the integral of the central path curvature and its relationship with the iteration-complexity of primal-dual path-following LP algorithms,” *Mathematical Programming*, vol. 115, no. 1, pp. 105–149, 2008. 5

[42] S. Kakihara, A. O. Ohara, and T. Tsuchiya, “Curvature integrals and iteration complexities in SDP and symmetric cone programs,” *Computational Optimization and Applications*, vol. 57, pp. 623–665, 2014. 5

[43] S. Kakihara, A. O. Ohara, and T. Tsuchiya, “Information geometry and interior-point algorithms in semidefinite programs and symmetric cone programs,” *Journal of Optimization Theory and Applications*, vol. 157, pp. 749–780, 2013. 5

[44] A. Deza, E. Nematollahi, and T. Terlaky, “How good are interior point methods? Klee–Minty cubes tighten iteration-complexity bounds,” *Math. Program.*, vol. 113, pp. 1–14, Oct. 2008. 5

[45] S. Mizuno, M. Todd, and Y. Ye, “On adaptive-step primal-dual interior-point algorithms for linear programming,” *Mathematics of Operations Research*, vol. 18, pp. 964–981, 11 1993. 5