

Revisiting Tardos’s Framework for Linear Programming: Faster Exact Solutions using Approximate Solvers

Daniel Dadush
*Networks & Optimization Group
 Centrum Wiskunde & Informatica
 Amsterdam, The Netherlands
 dadush@cwi.nl*

Bento Natura, László A. Végh
*Department of Mathematics
 London School of Economics and Political Science
 London, United Kingdom
 {b.natura, l.vegh}@lse.ac.uk*

Abstract—In breakthrough work, Tardos (Oper. Res. ’86) gave a *proximity* based framework for solving linear programming (LP) in time depending only on the constraint matrix in the bit complexity model. In Tardos’s framework, one reduces solving the LP $\min\langle c, x \rangle$, $Ax = b$, $x \geq 0$, $A \in \mathbb{Z}^{m \times n}$, to solving $O(nm)$ LPs in A having small integer coefficient objectives and right-hand sides using *any exact* LP algorithm. This gives rise to an LP algorithm in time $\text{poly}(n, m \log \Delta_A)$, where Δ_A is the largest subdeterminant of A . A significant extension to the real model of computation was given by Vavasis and Ye (Math. Prog. ’96), giving a specialized interior point method that runs in time $\text{poly}(n, m, \log \bar{\chi}_A)$, depending on Stewart’s $\bar{\chi}_A$, a well-studied condition number.

In this work, we extend Tardos’s original framework to obtain such a running time dependence. In particular, we replace the exact LP solves with *approximate* ones, enabling us to directly leverage the tremendous recent algorithmic progress for approximate linear programming. More precisely, we show that the fundamental “accuracy” needed to *exactly* solve any LP in A is inverse polynomial in n and $\log \bar{\chi}_A$. Plugging in the recent algorithm of van den Brand (SODA ’20), our method computes an optimal primal and dual solution using $O(mn^{\omega+1+o(1)} \log(\bar{\chi}_A+n))$ arithmetic operations, outperforming the specialized interior point method of Vavasis and Ye and its recent improvement by Dadush et al (STOC ’20). By applying the preprocessing algorithm of the latter paper, the dependence can also be reduced from $\bar{\chi}_A$ to $\bar{\chi}_A^*$, the minimum value of $\bar{\chi}_{AD}$ attainable via column rescalings. Our framework is applicable to achieve the $\text{poly}(n, m, \log \bar{\chi}_A^*)$ bound using essentially any weakly polynomial LP algorithm, such as the ellipsoid method.

At a technical level, our framework combines together approximate LP solutions to compute exact ones, making use of constructive proximity theorems—which bound the distance between solutions of “nearby” LPs—to keep the required accuracy low.

Keywords—linear programming, strongly polynomial algorithms, condition numbers, proximity, circuits

I. INTRODUCTION

In this paper, we consider the task of computing exact primal and dual solutions for linear programs (LP) in standard

Full version available at <https://arxiv.org/abs/2009.04942>. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreements ScaleOpt–757481 and QIP–805241).

form:

$$\begin{aligned} \min \langle c, x \rangle & & \max \langle y, b \rangle \\ Ax = b & & A^\top y + s = c \\ x \geq 0, & & s \geq 0. \end{aligned} \quad (\text{LP})$$

Here, $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m \leq n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are given in the input, and $x, s \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ are the variables. We consider the program in x to be the primal problem and the program in y, s to be the dual problem.

After the work of Khachiyan [1], who gave the first polynomial algorithm for LP using the ellipsoid method, Megiddo [2] asked whether there exists a “genuinely polynomial”, now known as *strongly polynomial*, algorithm for LP. Informally, the goal is to find an algorithm that uses $\text{poly}(n)$ *basic arithmetic operations* (e.g. addition, multiplication, etc.), where each such operation must be performed on numbers of size polynomial in the instance encoding length. While no such algorithm is known, the search for a strongly polynomial LP algorithm has spurred tremendous algorithmic advances for many classical combinatorial problems.

Strongly polynomial algorithms have indeed been found for important combinatorial classes of linear programs. Examples include feasibility for two variable per inequality systems [2], minimum-cost circulations [3]–[5], generalized flow maximization, [6], [7], and discounted Markov Decision Processes [8], [9].

To generalize these results to larger problem classes, a natural attempt is to seek abstract frameworks that capture known algorithms. In this vein, a recurring principle in strongly polynomial algorithm design is that “good enough” approximate solutions can be used to glean combinatorial information about exact optimal ones. Such information is used to reduce the underlying instance in a way that preserves all optimal solutions.

This was in fact the key idea in Tardos’s seminal paper on minimum-cost circulations [5]: solving a problem instance with a suitable rounded cost function reveals an arc that cannot be tight in any dual optimal solution; consequently, we can fix the flow value to 0. As another example, in submodular function minimization any sufficiently small

norm point in the base polytope can be used to infer relations in a ring-family containing all minimizers [10], [11].

At a higher level, it can be useful to view strongly polynomial algorithms as reductions from an exact optimization problem to a suitable approximate version of itself. To achieve fast strongly polynomial algorithms using these principles, important considerations are the complexity of the individual approximate solves, e.g., the degree of accuracy required, and the total required number of them.

Tardos's Framework for Linear Programming: Generalizing the above idea from minimum-cost flows to general linear programming, Tardos [12] provided such a framework for solving any standard form primal-dual LP with integer constraint matrix $A \in \mathbb{Z}^{m \times n}$ using a number of operations depending only on n and the logarithm of Δ_A , the maximum absolute value of the determinant of any square submatrix of A . This algorithm is strongly polynomial for minimum-cost flow, noting that digraph incidence matrices are totally unimodular, and therefore $\Delta_A = 1$. At a high level, Tardos's framework reduces getting exact LP solutions to getting exact solutions for "nearby LPs" with simpler coefficient structure, heavily relying on LP *proximity* theorems (e.g., see [13], [14]). More precisely, Tardos reduces computing exact primal-dual solutions to $\max \langle c, x \rangle, Ax = b, x \geq 0$ to computing exact primal-dual solutions to $O(nm)$ LPs in A with "rounded" objectives c' and right hand sides b' having integer coefficients of size $O(n^2 \Delta_A)$. In particular, after $O(n)$ such LP solves, one can determine a coefficient x_i in the support of some optimal solution, allowing to delete the $x_i \geq 0$ constraint. Due to their small coefficients, the LPs in the reduction can be solved using any weakly polynomial algorithm. We note that the fundamental property enabling the polynomial solvability of these rounded LPs is that the minimum non-zero slack of their basic solutions, i.e., $\min\{x_i : x_i > 0\}$, is lower bounded by $1/(n^{O(1)} \Delta)$ by Cramer's rule.

Achieving $\bar{\chi}_A$ dependence: While Tardos's framework is powerful, it inherently relies on the determinant bound Δ_A . This is only applicable for integer constraint matrices; one can obtain bounds for rational constraint matrices via multiplying by the least common denominator of the entries, but this leads to weak bounds that are highly volatile under small changes in the entries. A significant strengthening of [12] was given by Vavasis and Ye [15]. They gave an interior point method (IPM) in the real model of computation based on *layered least squares* (LLS) steps that outputs exact primal-dual solutions in $O(n^{3.5} \log(\bar{\chi}_A + n))$ iterations. Improved iteration bounds were later given for certain special cases, in particular, $O(\sqrt{n} \log(\bar{\chi}_A + n))$ for homogeneous conic feasibility [16] and $O(n^{2.5} \log(\bar{\chi}_A + n))$ for LP feasibility [17]. In a conceptual advance, Vavasis and Ye's result showed that the polynomial solvability of LP does not require any minimum non-zero slack assumption.

The condition measure replacing Δ_A is Stewart's $\bar{\chi}_A$ [18],

which for integer matrices satisfies $\bar{\chi}_A \leq n \Delta_A$. In contrast with Δ_A that relies on the entry numerics, $\bar{\chi}_A$ is a geometric measure that depends only on the kernel of A ; Formally, letting $W := \ker(A)$ and $\pi_I(W) = \{x_I : x \in W\}$, one may define $\bar{\chi}_A := \bar{\chi}_W$ as the minimum number $M \geq 1$ such that for any $\emptyset \neq I \subseteq [n]$ and $z \in \pi_I(W)$, there exists $y \in W$ with $y_I = z$ and $\|y\| \leq M \|z\|$. In words, it represents the cost of lifting partial fixings of coordinates into the subspace W .

Very recently, the authors and Huiberts [19], building on the work of Monteiro and Tsuchiya [20], [21], gave an improved LLS optimization algorithm and analysis requiring only $O(n^{2.5} \log n \log(\bar{\chi}_A^* + n))$ iterations, where $\bar{\chi}_A^*$ is the minimum $\bar{\chi}_{AD}$ over positive diagonal matrices $D > 0$. The paper [19] further gave a nearly optimal rescaling algorithm which runs in $O(m^2 n^2 + n^3)$ time and computes $D > 0$ satisfying $\bar{\chi}_{AD} \leq n(\bar{\chi}_A^*)^3$. Thus, by suitable preprocessing, any algorithm achieving $\bar{\chi}_A$ dependence can be converted into one with $\bar{\chi}_A^*$ dependence.

A key tool in [19] is to study the 'circuit imbalance measure' κ_A . This closely approximates $\bar{\chi}_A$, with $\log(\bar{\chi}_A + n) = \Theta(\log(\kappa_A + n))$, and has very favourable combinatorial properties. Our approach also relies on κ_A and κ_A^* , even though we state the results in terms of the better known $\bar{\chi}_A$ and $\bar{\chi}_A^*$.

The condition number $\bar{\chi}_A^*$ can be smaller than $\bar{\chi}_A$ by an arbitrary factor, and in turn, $\bar{\chi}_A$ can be much smaller Δ_A even for integer matrices A . Let $A \in \mathbb{R}^{n \times m}$ be the node-edge incidence matrix of an undirected graph on n nodes and m edges. If the graph has k node-disjoint odd cycles, then $\Delta_A \geq 2^k$. However, it is easy to verify that for any graph, $\kappa_A \leq 2$ (see the definition of κ_A in Section II-A). Using Proposition II.4, we get the bound $\bar{\chi}_A \leq 2m$.

Harnessing the progress in approximate solvers: The complexity of fast approximate LP algorithms has seen substantial improvements in recent years [22]–[27]. Taking the recent algorithm [24], given a feasible LP $\min \langle c, x \rangle, Ax = b, x \geq 0$, having an optimal solution of ℓ_2 norm at most R , for $\varepsilon > 0$ it computes a point $\tilde{x} \geq 0$ satisfying

$$\begin{aligned} \langle c, \tilde{x} \rangle &\leq \min_{Ax=b, x \geq 0} \langle c, x \rangle + \varepsilon \cdot \|c\|_2 \cdot R \\ \|A\tilde{x} - b\|_2 &\leq \varepsilon \cdot (\|A\|_F \cdot R + \|b\|_2), \end{aligned} \quad (\text{APX-LP})$$

in deterministic time $O(n^{\omega+o(1)} \log(n/\varepsilon))$, where $\omega < 2.38$ is the matrix multiplication exponent.

Tardos's framework requires an exact black box solver for systems with the same matrix A but replacing b and c by small integer vectors. It is possible to use the approximate solver (APX-LP) to obtain exact optimal solution for integer matrices for sufficiently small ε . Assume $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$ and $\|b\|_\infty, \|c\|_\infty \leq n^{O(1)} \Delta^t$, and let OPT denote the optimum value of (LP). We may call (LP) in a suitable extended system with $\varepsilon = 1/\left(n^{O(1)} \Delta_A^{O(t)}\right)$, and use a Carathéodory reduction to identify primal and

dual optimal basic solutions. Integrality is used in multiple parts of such a reduction: e.g., for establishing a bound $R = n^{O(1)} \Delta_A^{O(t)}$ from Cramer’s rule, and for showing that for any primal feasible solution x , $\langle c, x \rangle < \text{OPT}$ implies $\langle c, x \rangle < \text{OPT} - \varepsilon \|c\|_2 R$. For a matrix $A \in \mathbb{R}^{m \times n}$, we cannot obtain an exact solver by applying the approximate solver for high enough accuracy in terms of the condition numbers $\bar{\chi}_A$ or κ_A . This is the main reason why we cannot work with explicitly rounded systems, but require a more flexible approach. Let us also note that recovering an exact solution from the approximate solver comes at a high arithmetic cost that we can save if using the approximate solution directly.

Fast algorithms with $\bar{\chi}_A$ dependence: The layered least squares interior point methods discussed above represent substantial advances in the strongly polynomial solvability of LP, yet it is highly non-obvious how to combine these techniques with those of recent fast LP solvers. For example, for the results of [22], [25], one would have to develop analogues of LLS steps for weighted versions of the logarithmic barrier. Furthermore, the proofs of exact convergence are intricate and deeply tied to the properties of the central path, and may leave one wondering whether the $\bar{\chi}_A$ solvability of LP is due to “IPM magic”. It would therefore be desirable to have an elementary proof of the $\bar{\chi}_A$ solvability of LP.

Partial progress on this question was given by Tunçel and Ho [28], who generalized Tardos’s framework in the real number model. Firstly, they showed that one can still round instances to have minimum non-zero slack $\tau_A > 0$, depending only on A . Second, they showed that applying the Mizuno-Todd-Ye [29] predictor-corrector IPM on the homogeneous self-dual formulation, these rounded instances can be solved $\text{poly}(n, \log \tau_A, \log(\Delta_A/\delta_A))$ time, where δ_A is the absolute value of the minimum non-zero determinant of any square submatrix of A . Here, they prove the relation $\bar{\chi}_A \leq n \Delta_A/\delta_A$ and note that Δ_A/δ_A can be arbitrarily larger than $\bar{\chi}_A$. Lastly, they provide a different algorithm that removes the dependence on τ_A , assuming one has access to the Vavasis-Ye algorithm as a subroutine only on instances with $b \in \{\pm 1, 0\}^m$, $c \in \{0, \pm 1\}^n$.

A. Our Contributions

As our main contribution, we provide a substantially improved Tardos style framework for LP which achieves both $\bar{\chi}_A$ dependence and relies only on approximate LP solves: we use the output (APX-LP) of the approximate LP solvers in a black-box manner. Our main result using the deterministic solver in [24] is summarized below. The more precise technical statements generalized to non-deterministic solvers are given as Theorem V.3 for feasibility and Theorem VI.2 for optimization.

Theorem I.1 (Enhanced Tardos Framework for Feasibility). *Assume we are given a feasibility LP $Ax = b$, $x \geq 0$ with data $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m$, and $b \in \mathbb{R}^m$.*

- (i) *If the primal program is feasible, then one can find a feasible solution x using $O(m)$ approximate LP solves (APX-LP) with accuracy $\varepsilon = 1/(n\bar{\chi}_A)^{O(1)}$ together with additional $O(mn^{\omega+o(1)})$ arithmetic operations. This gives a total complexity $O(mn^{\omega+o(1)} \log(\bar{\chi}_A + n))$ using the solver of van den Brand [24].*
- (ii) *If the primal program is infeasible, then a Farkas certificate of infeasibility $y \in \mathbb{R}^m$, satisfying $A^\top y \geq 0$, $\langle b, y \rangle < 0$ can be found using the amount of computation as in (i), and $O(nm^2 + n^{\omega+o(1)}) \log \log(\bar{\chi}_A + n)$ additional arithmetic operations.*

Next, we state our result for optimization:

Theorem I.2 (Enhanced Tardos Framework for Optimization). *Assume we are given primal-dual (LP) with data $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.*

- (i) *If both primal and dual programs are feasible, then one can obtain an optimal primal-dual pair (x, y, s) of solutions, using at most $O(nm)$ approximate LP solves (APX-LP) as in Theorem I.1(i), together with an additional $O(mn^{\omega+1+o(1)})$ arithmetic operations. This gives a total complexity $O(mn^{\omega+1+o(1)} \log^2(n) \log(\bar{\chi}_A + n))$ using [24].*
- (ii) *If either of the primal or dual programs are infeasible, then we can obtain a Farkas certificate of primal or dual infeasibility in the same running time as in (i), plus $O(n^3 m^2 \log \log(\bar{\chi}_A + n))$ additional arithmetic operations.*

This theorem yields the first LP algorithm achieving $\bar{\chi}_A$ dependence that is not based of the analysis of the central path. At a high level, we achieve this by more deeply exploiting the power of LP proximity theorems, which are already at the core of Tardos’s framework. In the rest of this section, we explain some of the key ideas behind the above theorem and how it compares to Tardos’s original algorithm as well as that of Vavasis and Ye.

Overview of the approach: Both Tardos’s and our approach use variants of Hoffman’s proximity bounds, see Section III. The fundamental difference is that while Tardos uses an exact solver where the perturbed objective and right hand side vectors are fixed in advance before calling the solver, we decide these perturbations “on the fly” as a function of the returned approximate solutions we receive.

Let us illustrate Tardos’s and our approaches on the dual feasibility LP

$$A^\top y + s = c, s \geq 0. \quad (D)$$

The feasibility algorithm in [5] proceeds as follows. Define $\tilde{b}_i = \sum_{j=1}^n (\Delta_A + 1)^{i-1} a_j$, where a_i is the i -th column vector of A , and consider the primal system

$$\min \langle c, x \rangle \quad \text{s.t.} \quad Ax = \tilde{b}, x \geq 0. \quad (\tilde{P})$$

Note that by the choice of \tilde{b} , this system is always feasible.

If it is unbounded, then we may conclude infeasibility of (D) . The reason for the particular choice of \tilde{b} is that whenever the system is bounded, the dual of (\tilde{P}) has a unique optimal solution; this can be shown by a determinant argument. Consequently, for any optimal solution x^* to (\tilde{P}) and $S^* = \text{supp}(x^*)$, the system $a_i^\top y = c_i$, $i \in S^*$ yields a feasible solution to (D) . The exact LP solver will be applied to a series of rounded problem instances of the form

$$\min \langle \tilde{c}, x \rangle \quad \text{s.t. } Ax = \tilde{b}, \quad x \geq 0, \quad x_T = 0, \quad (\hat{P})$$

where $\tilde{c} \in \mathbb{Z}^n$, $\|\tilde{c}\|_\infty \leq n^2 \Delta_A$, and $T \subseteq [n]$ is a set of indices i where we have already concluded that $x_i^* = 0$ in every optimal solution to (\tilde{P}) . This is initialized as $T = \emptyset$, and every call to the LP solver enables the addition of at least one new index; thus, we need $O(n)$ oracle calls to solve feasibility. According to the definition of \tilde{b} , this is an integer vector with $\|\tilde{b}\| = \Theta(\sqrt{m} \Delta_A^n)$. As explained above, we can obtain an exact solution to (\hat{P}) by calling (APX-LP) for accuracy $\varepsilon = 1/(n^{O(1)} \tilde{\Delta}_A^{O(n)})$.

To conclude that $i \in T$ for some $i \in [n]$, Tardos uses a proximity theorem that is a variant of Lemma III.4. It implies that if $\|\tilde{c} - c\|_\infty$ is “small”, then (\hat{P}) has a dual optimal solution that is “close” to the dual optimal solution obtained for (\hat{P}) .

In contrast, our approach in Section V proceeds as follows. If $c \geq 0$, we simply return $s = c$. Otherwise, the norm of the negative coordinates $\|c^-\|_1$ will play a key role. We can strengthen (D) by adding the constraint

$$\|s - c\|_\infty \leq 16\kappa_A^2 n \|c^-\|_1, \quad (1)$$

where κ_A is the circuit imbalance measure; for integer matrices $\kappa_A \leq \Delta_A$. A proximity result (Corollary III.2) implies that whenever (D) is feasible, there is a feasible solution also satisfying (1).

We can use (APX-LP) directly to obtain a solution (\tilde{y}, \tilde{s}) such that $A^\top \tilde{y} + \tilde{s} = c$, $\|\tilde{s} - c\|_\infty \leq 3\kappa_A^2 n \|c^-\|_1$, and $\|\tilde{s}^-\|_\infty \leq \varepsilon \|c^-\|_1$ for $\varepsilon = 1/O(n^4 \kappa_A^4)$. Again, note that in addition to approximate feasibility, we also require proximity of s to c ; we can obtain such a solution with this extra property without an increase in the running time cost.

From here, we can identify a set K of coordinates such that \tilde{s}_i is large enough to conclude that there exists a feasible solution s to (D) with $\tilde{s}_i > 0$ for $i \in K$; this is done similarly as in Tardos’s approach.

We project out all variables in K , meaning that we remove the inequalities $a_i^\top y + s_i = c_i$ for $i \in K$ from the system. We recurse on the smaller subsystem. From the recursive call, we obtain a feasible solution y' to (D) in the smaller system that also satisfies (1). The proximity constraints enables us to easily map back y' to a feasible solution y to (D) by a simple ‘pullback’ operation.

As noted above, the very existence of an exact LP oracle

heavily relies on the integrality assumption of A . This integrality is also used to establish the relation between the optimal solutions of (\tilde{P}) and the solutions of (D) , using a determinant argument. In contrast, the proximity arguments as in Lemma III.4 and Corollary III.2 do not rely on integrality; we can use here κ_A instead of Δ_A .

Even for integer matrices and $\kappa_A = \Theta(\Delta_A)$, and using the same solver for (APX-LP), our algorithm is faster by a factor $\Omega(n^2/m)$. A key ingredient in the running time improvement is to strengthen the system with (1). This allows us to use $\varepsilon = 1/(n^{O(1)} \kappa_A^{O(1)})$; otherwise, we would need to require a higher precision $\varepsilon = 1/(n^{O(1)} \kappa_A^{O(n)})$. This yields a factor n improvement over [5].

Another factor n/m improvement is obtained as follows. In the approach sketched above, if the set of “large” coordinates K is nonempty, we get a bound n on the number of recursive calls. Using a slightly more careful recursive setup, we can decrease the rank of the system at each iteration, improving this bound to m .

Let us now turn to optimization. Our algorithm will be more similar to the one in [5], and for integer matrices with $\kappa_A = \Theta(\Delta_A)$ and $m = \Omega(n)$, the asymptotic running time bounds will be the same.

We now outline Tardos’s approach. Given an optimization problem (LP), we first check for both primal and dual feasibility. If these are both feasible, then we go through $\leq m$ main loops. In each main loop, we use the same approach as above to solve (\tilde{P}) with a perturbed $\tilde{b} \in \mathbb{Z}^m$ with $\|\tilde{b}\|_\infty \leq n^2 \Delta_A$. Using $\leq n$ oracle calls, we obtain optimal primal and dual solutions (x, y, s) . Again, proximity guarantees that if \tilde{b} is “close” to b , then we can identify an index i with a “large” $x_i > 0$ where we can conclude $s_i^* = 0$ in every optimal solution. Equivalently, x_i is in the support of some optimal solution, and hence we may delete the constraint $x_i \geq 0$, and proceed to the next main loop after projecting out the variable x_i . We note that the bound n on the inner loops is in reality $n - m$, and this can be improved to m by swapping the primal and dual sides.

In our approach in Section VI, the goal is to end up in the same place as Tardos at the end of the main loop, where the difference will be how we get there. As mentioned above, in Tardos’s setting, one already knows beforehand that the final objective and right hand side for which one will have optimal primal-dual solutions will be \tilde{b} , a rounded version of b , and the original c . However, the only important property is that at the end of the loop we end up with a primal-dual optimal pair for the original objective c , and *some* right hand side b' close enough to the original b . In particular, b' need not be known at the beginning of the algorithm and can thus be chosen *adaptively* depending on the outcome of the approximate LP solves.

For the above purpose, we utilize proximity theorems (see Section III for precise statements) to allow us to

stitch together the “large” coordinates of approximate dual solutions to achieve feasibility. At the same time, we perform a similar complementary stitching of primal approximate solutions, where we judiciously perturb “small” coordinates to 0, inducing a corresponding change of right hand side, to enforce complementarity with the dual solution. Here proximity allows us to control how much the solutions will change in future iterations, which is crucial to not destroying the structure of the solutions built so far.

We also note that Grötschel, Lovász, and Schrijver [30, Theorem 6.6.3] give a different proof for Tardos’s result using simultaneous Diophantine approximation (see also [31]). This shows that (LP) can be solved by creating a single perturbed instance with integer \tilde{b} and \tilde{c} bounded in terms of the encoding length of A such that the set of optimal bases coincide in the two systems. The perturbed instance can be solved in $\text{poly}(n, m, \log \Delta_A)$; we simply take an optimal basis and compute the corresponding primal and dual optimal solutions for the original b and c . However, this reduction inherently relies on integrality arguments.

Failure will be certified: Our algorithm requires an estimate on the circuit imbalance parameter κ_A (see definition in Section II-A). This is a common assumption shared by most previous literature: Tardos’s algorithm uses an estimate of Δ_A ; Vavasis and Ye require a bound on $\bar{\chi}_A$. These parameters are hard to compute [32], [33]. However, knowing these values are not required, and we can use the following simple guessing procedure, attributed to J. Renegar in [15]. We start with a low guess on $\bar{\chi}_A$ (or some other parameter), say $M = 100$. If the algorithm fails to return the required solution, then we conclude that the estimate was too low, and replace the guess M by M^2 . Thus, we can still obtain a dependence on $\log(\bar{\chi}_A + n)$, without knowing the value.

A new aspect of our algorithm is that in case of a failure, we do not simply conclude that our estimate was too low indirectly from the failure of the algorithm, but we also obtain an explicit *certificate*. Namely, an elementary operation is to compute *lifts* mentioned previously: for the subset $W = \ker(A)$, an index set $I \subseteq [n]$, and a vector $y \in \pi_I(W)$, we compute the minimum-norm vector $z \in W$ such that $z_I = y$. Our parameter κ_A satisfies $\|z\|_\infty \leq \kappa_A \|y\|_1$ (Proposition II.3). Whenever our algorithm fails due to underestimating $M < \kappa_A$, this will be certified by an index set $I \subseteq [n]$ and a vector $y \in \pi_I(W)$, and lift z with $\|z\|_\infty > M \|y\|_1$.

II. PRELIMINARIES

For vectors $v, w \in \mathbb{R}^n$ we denote by $\min\{v, w\}$ the vector $z \in \mathbb{R}^n$ with $z_i = \min\{v_i, w_i\}, i \in [n]$; analogously for $\max\{v, w\}$. Further, we use the notation $v^+ = \max\{v, 0_n\}$ and $v^- = \max\{-v, 0_n\}$.

For a vector $v \in \mathbb{R}^n$, we denote by $\text{diag}(v)$ the diagonal matrix whose i -th diagonal entry is v_i . For two vectors

$x, y \in \mathbb{R}^n$, we let $\langle x, y \rangle = x^\top y$ denote their scalar product. We denote by the binary operation \circ the element-wise multiplication $x \circ y = \text{diag}(x)y$. We let \mathbf{D} denote the set of all positive definite $n \times n$ diagonal matrices.

For an index subset $I \subseteq [n]$, we use $\pi_I : \mathbb{R}^n \rightarrow \mathbb{R}^I$ for the coordinate projection. 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 let $\mathbb{R}_I^n = \{x \in \mathbb{R}^n : x_{[n] \setminus I} = 0\}$.

For a subspace $W \subseteq \mathbb{R}^n$, we let $W_I = \pi_I(W \cap \mathbb{R}_I^n)$. It is easy to see that $\pi_I(W)^\perp = (W^\perp)_I$. Assume we are given a matrix $A \in \mathbb{R}^{m \times n}$ such that $W = \ker(A)$. Then, $W_I = \ker(A_I)$, and we can obtain a matrix A' from A such that $\pi_I(W) = \ker(A')$ by performing a Gaussian elimination of the variables in $[n] \setminus I$.

For a subspace $W \subseteq \mathbb{R}^n$ and a vector $d \in \mathbb{R}^n$ we define by d/W the orthogonal projection of d onto W^\perp , that is $d/W = \pi_{W^\perp}(d)$. In particular, d/W is the minimum-norm vector in $W + d$. Further, for a subspace $W \subseteq \mathbb{R}^n$, we let $W_+ = W \cap \mathbb{R}_+^n$.

Linear programming in subspace formulation: Let $A \in \mathbb{R}^{m \times n}$, and $W = \ker(A) \subseteq \mathbb{R}^n$. For $c, d \in \mathbb{R}^n$, we can write (LP) in the following equivalent form, where $d \in \mathbb{R}^n$ such that $Ad = b$. Define the system *Primal-Dual*(W, d, c):

$$\begin{array}{ll} \min \langle c, x \rangle & \max \langle d, c - s \rangle \\ x \in W + d & s \in W^\perp + c \\ x \geq 0, & s \geq 0. \end{array}$$

Note that (x, s) are optimal primal and dual solutions if and only if they are feasible and $\langle x, s \rangle = 0$. Thus, *Primal-Dual*(W, d, c) is equivalent to the following feasibility problem:

$$x \in W + d, s \in W^\perp + c, \langle x, s \rangle = 0, (x, s) \geq 0. \quad (2)$$

Circuits: For a linear subspace $W \subseteq \mathbb{R}^n$ and a matrix A such that $W = \ker(A)$, a circuit is an inclusion-wise minimal dependent set of columns of A . The set of circuits of W is denoted \mathcal{C}_W .

For a subset $I \subseteq [n]$, we let $\text{cl}(I)$ denote its *closure* in the matroidal sense. We will make the assumption that $|C| > 1$ for all $C \in \mathcal{C}_W$.

A. The condition numbers $\bar{\chi}$ and κ

For a matrix $A \in \mathbb{R}^{m \times n}$, the condition number $\bar{\chi}_A$ is defined as

$$\bar{\chi}_A = \sup \left\{ \left\| A^\top (ADA^\top)^{-1} AD \right\| : D \in \mathbf{D} \right\}. \quad (3)$$

This quantity was first studied by Dikin [34], Stewart [18], and Todd [35], and has been extensively studied in the context of interior point methods; we refer the reader to [15], [20], [28] for further results and references.

It is important to note that $\bar{\chi}_A$ only depends on the subspace $W = \ker(A)$. Hence, we can also write $\bar{\chi}_W$ for a subspace $W \subseteq \mathbb{R}^n$, defined to be equal to $\bar{\chi}_A$ for some

matrix $A \in \mathbb{R}^{k \times n}$ with $W = \ker(A)$. We will use the notations $\bar{\chi}_A$ and $\bar{\chi}_W$ interchangeably.

Let us define the *lifting map* $L_I^W : \pi_I(W) \rightarrow W$ by

$$L_I^W(p) = \arg \min \{ \|z\| : z_I = p, z \in W \}.$$

A useful characterization of $\bar{\chi}_W$ can be given in terms of the operator norm of the lifting map. This was shown in [19], by using results from [18] and [36].

Proposition II.1 ([19]). *For a linear subspace $W \subseteq \mathbb{R}^n$,*

$$\bar{\chi}_W = \max \{ \|L_I^W\| : I \subseteq [n], I \neq \emptyset \}.$$

The circuit imbalance measure: Consider now the circuits \mathcal{C}_W of the subspace W . For a circuit $C \in \mathcal{C}_W$, let $g^C \in W$ be such that $\text{supp}(g^C) = C$. Note that g^C is unique up to multiplication by scalar. We define the *circuit imbalance measure*

$$\kappa_W = \max \left\{ \frac{\max_{i \in C} |g_i^C|}{\min_{j \in C} |g_j^C|} : C \in \mathcal{C}_W \right\},$$

as the largest ratio between two entries of any minimum support nonzero vector in W . This was studied in [19], [28], [37]. Note that $\kappa_W = 1$ corresponds to totally unimodular spaces. As shown in Proposition II.4 below, the condition measures $\bar{\chi}_W$ and κ_W are closely related: $O(\log(\bar{\chi}_W + n)) = O(\log(\kappa_W + n))$ holds. However, κ_W has several advantageous combinatorial properties. In fact, the argument in the proof of Tardos's main proximity result using Cramer's rule is implicitly bounding circuit imbalances, see discussion of Lemma III.4. Therefore, we will use κ_W instead of $\bar{\chi}_W$ throughout the paper.

We can give a characterization using max-norm instead of ℓ_2 -norm.

Proposition II.2. *For every matrix $A \in \mathbb{R}^{m \times n}$ with $\text{rk}(A) = m$,*

$$\kappa_A = \max \{ \|A_B^{-1}A\|_{\max} : B \text{ basis} \}.$$

Proposition II.1 asserts that $\bar{\chi}_W$ is the maximum $\ell_2 \rightarrow \ell_2$ operator norm of the mappings L_I^W over $I \subseteq [n]$. We can show that the maximum $\ell_1 \rightarrow \ell_\infty$ operator norm of these mappings is κ_W , even though the lifting map is defined with respect to ℓ_2 norms.

Proposition II.3. *For a linear subspace $W \subseteq \mathbb{R}^n$,*

$$\kappa_W = \max \left\{ \frac{\|L_I^W(p)\|_\infty}{\|p\|_1} : I \subseteq [n], I \neq \emptyset, p \in \pi_I(W) \setminus \{0\} \right\}.$$

Using Proposition II.3, we can easily relate the quantities $\bar{\chi}_W$ and κ_W . The upper bound was already shown in [37]; the slightly weaker lower bound $\sqrt{\bar{\chi}_W^2 - 1}/n \leq \kappa_W$ was given in [19].

Proposition II.4. *For a linear subspace $W \subseteq \mathbb{R}^n$,*

$$\frac{1}{n} \bar{\chi}_W \leq \kappa_W \leq \sqrt{\bar{\chi}_W^2 - 1}.$$

The estimate M and lifting certificates: The value of κ_W and $\bar{\chi}_A$ may not be known. In fact, these are hard to approximate even within a factor $2^{\text{poly}(m)}$ [33]. Throughout our algorithms, we maintain a guess M on the value of $2\kappa_W$, initialized as $M = 2$. At certain points in the algorithm, we may find an index set $I \subseteq [n]$ and a vector $p \in \pi_I(W)$ such that $\|L_I^W(p)\|_\infty > M\|p\|_1$. In this case, we conclude that $M < \kappa_W$ by Proposition II.3. Such a pair (I, p) is called a *lifting certificate* of $M > \kappa$. We can then restart the algorithm with an updated estimate $M' = \max\{2\|L_I^W(p)\|_\infty/\|p\|_1, M^2\}$.

Optimal rescalings: For every $D \in \mathbf{D}$, we can consider the condition numbers $\bar{\chi}_{WD} = \bar{\chi}_{AD^{-1}}$ and $\kappa_{WD} = \kappa_{AD^{-1}}$. We let

$$\begin{aligned} \bar{\chi}_W^* &= \bar{\chi}_A^* = \inf \{ \bar{\chi}_{WD} : D \in \mathbf{D} \} \\ \kappa_W^* &= \kappa_A^* = \inf \{ \kappa_{WD} : D \in \mathbf{D} \} \end{aligned}$$

denote the best possible values of $\bar{\chi}$ and κ that can be attained by rescaling the coordinates of W . A near-optimal rescaling can be found in strongly polynomial time [19]. As a consequence, after using this preprocessing step, any algorithm that has running time dependence on $\log(\kappa_W + n)$ is turned into an algorithm with dependence on $\log(\kappa_W^* + n)$. We note however that for small values of $\log(\kappa_W + n)$, this preprocessing may turn out to be a bottleneck operation for our feasibility algorithm.

III. PROXIMITY VIA HOFFMAN-BOUNDS

Hoffman's seminal work [13] has analyzed proximity of LP solutions. Given $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, $x_0 \in \mathbb{R}^n$, and norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$, we are interested in the minimum of $\|x - x_0\|_\alpha$ over $x \in P$. Hoffman showed that this can be bounded as $H_{\alpha,\beta}(A)\|(Ax_0 - b)^+\|_\beta$, where the Lipschitz-bound $H_{\alpha,\beta}(A)$ is a constant that only depends on A and the norms. Such bounds have been shown for several different problem forms and norms; we refer the reader to [38] for results and references.

We will use a Hoffman-bound for a system of the form $x \in W$, $\ell \leq x \leq u$. We show that $H_{\infty,1} = \kappa_W$ for such a system. Related bounds using $\bar{\chi}_A$ have been shown in [28].

For vectors $d, c \in \mathbb{R}^n$, let us define the set

$$\Lambda(d, c) := \text{supp}(d^-) \cup \text{supp}(c^+). \quad (4)$$

Theorem III.1 (Hoffman Proximity Theorem). *Let $W \subseteq \mathbb{R}^n$ be a subspace and $\ell \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{\infty\})^n$ be lower and upper bounds, and assume that $P = \{x \in W : \ell \leq x \leq u\}$ is non-empty. Then, for every $x \in P$ we have*

$$\|\ell^+ + u^-\|_1 \leq \|x_{\Lambda(u,\ell)}\|_1,$$

and there exists $x \in P$ such that

$$\|x\|_\infty \leq \kappa_W \|\ell^+ + u^-\|_1.$$

We can derive useful corollaries for feasibility and optimization problems.

Corollary III.2. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $d \in \mathbb{R}^n$. If the system $x \in W + d$, $x \geq 0$ is feasible, then the system*

$$x \in W + d, \|x - d\|_\infty \leq \kappa_W \|d^-\|_1, x \geq 0,$$

is also feasible.

Corollary III.3. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $c, d \in \mathbb{R}^n$, and assume $c \geq 0$. If $\text{Primal-Dual}(W, d, c)$ is feasible, then there is an optimal solution (x, s) such that*

$$\|x - d\|_\infty \leq \kappa_W \|d_{\Lambda(d,c)}\|_1.$$

The next lemma will be used to conclude that a primal variable $s_i^* = 0$ in every solution (x^*, s^*) to $\text{Primal-Dual}(W, d, c)$. For integer matrices, a similar statement was given by Cook et al. [14, Theorem 5], see also [39, Theorem 10.5] with a bound in terms of the maximum subdeterminant Δ_A . A variant of this statement is used by Tardos [5, Lemma 1.1] as the main underlying proximity statement of her algorithm. Ho and Tunçel [28, Theorem 6.3] generalized this bound to arbitrary matrices, using the condition number $\bar{\chi}_A$. This implies our statement with $n\kappa_W$ instead of $\kappa_W + 1$. We note that the arguments in [5], [14] are based on Cramer's rule. In essence, this is used to bound the circuit imbalances in terms of Δ_A . Hence, our formulation with κ_W can be seen as a natural extension.

Lemma III.4. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $c, d, \tilde{d} \in \mathbb{R}^n$. Let (\tilde{x}, s) be an optimal solution to $\text{Primal-Dual}(W, \tilde{d}, c)$. Then there exists an optimal solution (x^*, s^*) to $\text{Primal-Dual}(W, d, c)$ such that*

$$\|x^* - \tilde{x}\|_\infty \leq (\kappa_W + 1) \|d - \tilde{d}\|_1.$$

We can immediately use this theorem to derive a conclusion on the support of the optimal dual solutions to $\text{Primal-Dual}(W, d, c)$, using the optimal solution to $\text{Primal-Dual}(W, \tilde{d}, c)$.

Corollary III.5. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $c, d, \tilde{d} \in \mathbb{R}^n$. Let (\tilde{x}, s) be an optimal solution to $\text{Primal-Dual}(W, \tilde{d}, c)$. Let*

$$R := \{i \in [n] : \tilde{x}_i > (\kappa_W + 1) \|d - \tilde{d}\|_1\}.$$

Then for every dual optimal solution s^* to $\text{Primal-Dual}(W, d, c)$, we have $s_R^* = 0$.

We now formulate a strengthening of this corollary. We show that besides setting dual variables in R to 0, we are also able to set certain primal variables to 0. This will be the key to decrease the number of recursive calls from n to m .

More precisely, we show the following. Assume x' in the previous proof contains a 'large' coordinate set I_L , significantly larger than the threshold for R in Corollary III.5. Assume that the closure $\text{cl}(I_L)$ contains some indices from $[n] \setminus R$. Then, we can transform x' in the proof to another optimal solution x'' where all these coordinates are set to 0. This can be achieved by changing the coordinates in I_L only, and their high value in x'' guarantees that they remain positive.

Theorem III.6. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $c, d, \tilde{d} \in \mathbb{R}^n$. Let (\tilde{x}, s) be an optimal solution to $\text{Primal-Dual}(W, \tilde{d}, c)$, and let $\tau \geq (\kappa_W + 1) \|d - \tilde{d}\|_1$ and $T \geq (2n\kappa_W + 1)\tau$. Let us define the following partition of $[n]$ into large, medium, and small indices.*

$$I_L = \{i \in [n] : \tilde{x}_i > T\}, \quad I_M = \{i \in [n] : T \geq \tilde{x}_i > \tau\},$$

and $I_S = [n] \setminus (I_L \cup I_M)$. We further partition I_S as

$$I_S^0 = I_S \cap \text{cl}(I_L), \quad I_S^+ = I_S \setminus \text{cl}(I_L).$$

Then, there exists a primal optimal solution x'' to $\text{Primal-Dual}(W, d, c)$ such that $x''_{I_L \cup I_M} > 0$, and $x''_{I_S^0} = 0$.

IV. BLACK-BOX LINEAR PROGRAMMING SOLVERS

Our feasibility and optimization algorithms in Sections V and VI use oracles that return approximate LP solutions. These can be implemented by using any weakly-polynomial algorithm that returns approximately optimal approximately feasible solutions as in (APX-LP). We will use the following result that summarizes recent developments on interior point methods. Whereas the papers only formulate the main statements on primal solutions, they all use primal-dual interior-point methods, and also find dual solutions with similar properties. We present the results in such a primal-dual form.

Theorem IV.1 ([22], [24], [25], [27]). *Consider (LP) for $A \in \mathbb{R}^{m \times n}$ with $\text{rk}(A) = m$. Assume both the primal and dual programs are feasible, let $\delta \in [0, 1]$ and $d \in \mathbb{R}_+^n$ be such a feasible primal solution i.e. $Ad = b$. Let R_P and R_D be the diameters of the primal and dual solution sets in ℓ_2 norm, i.e., $\|x\|_2 \leq R_P$ for all primal feasible solutions x , and $\|s\|_2 \leq R_D$ for all dual feasible solutions (y, s) . Then, we can find a vector $(x, y, s) \in \mathbb{R}^{n+m+n}$ with $x, s \geq 0$ such that*

- (i) $\langle c, x \rangle \leq \langle b, y \rangle + \delta \cdot (\|c\|_2 \cdot R_P + \|d\|_2 \cdot R_D)$,
- (ii) $\|Ax - b\|_2 \leq \delta \cdot (\|A\|_F \cdot R_P + \|b\|_2)$, and
- (iii) $\|A^\top y + c - s\|_2 \leq \delta \cdot (\|A\|_F \cdot R_D + \|c\|_2)$.

in the following running time bounds:

- (1) In $O((mn + m^3) \log^{O(1)}(n) \log(n/\delta))$ expected running time [25].
- (2) In $O(n^{\omega+o(1)} \log(n/\delta))$ deterministic running time, assuming $\omega \geq 13/6$ [24]. The same expected running time is achievable assuming $\omega \geq 2 + 1/18$ [27].

(3) In $O((\text{nnz}(A)+m^2)\sqrt{m}\log^{O(1)}(n)\log(n/\delta))$ expected running time [22], where $\text{nnz}(A)$ denotes the number of nonzero entries in A .

We use the notation $\Psi(A)$ to denote the ‘cost per unit’ in these results. Namely, a δ -approximate solution can be obtained in time $O(\Psi(A)\log(n/\delta))$, where

$$\Psi(A) \leq \log^{O(1)}(n) \min\{mn + m^3, n^{\omega+o(1)}, \sqrt{m}(\text{nnz}(A) + m^2)\}. \quad (5)$$

We note that the third bound will not be directly applicable, since we will use the oracle in various subspaces, where the number of nonzero entries may increase.

V. THE FEASIBILITY ALGORITHM

Given a matrix $A \in \mathbb{R}^{m \times n}$ and $d \in \mathbb{R}^n$, we let $W = \ker(A)$. In this section, we consider the feasibility problem $x \in W + d, x \geq 0$.

A key insight is to work with a stronger system, including a proximity constraint. According to Corollary III.2, whenever the problem $x \in W + d, x \geq 0$ is feasible and $\kappa_W \leq M$, then the following system is also feasible. In fact, this would be true even with the stronger bound M instead of $16M^2n$; we use this weaker bound to leave sufficient slack for the recursive argument. Note that if $d \geq 0$, then the only feasible solution is $x = d$.

$$\begin{aligned} x &\in W + d \\ \|x - d\|_\infty &\leq 16M^2n\|d^-\|_1 \quad (\text{Feas-LP}(W, d, M)) \\ x &\geq 0. \end{aligned}$$

We use a black-box approach assuming an oracle that returns an approximately feasible solution. We will assume that an oracle $\text{Prox-Feas-Solver}(W, d, M, \varepsilon)$ is given as in Oracle 1. Outcome (i) gives an approximately feasible solution with a bound on the negative components and a somewhat stronger proximity guarantee as in $(\text{Feas-LP}(W, d, M))$. Outcome (ii) gives a Farkas certificate of infeasibility, whereas outcome (iii) gives a lifting certificate of $M < \kappa_W$.

Input : A subspace $W \subseteq \mathbb{R}^n$, given as $W = \ker(A)$ for $A \in \mathbb{R}^{m \times n}$, a vector $d \in \mathbb{R}^n$, $M, \varepsilon > 0$.

Output: One of the following three outcomes

(i) A solution x to the system

$$\begin{aligned} x &\in W + d \\ \|x - d\|_\infty &\leq 3M^2n\|d^-\|_1 \quad (\text{Prox-Feas}(W, d, M, \varepsilon)) \\ \|x^-\|_\infty &\leq \varepsilon\|d^-\|_1 \end{aligned}$$

(ii) A vector $y \in W^\perp$, $y \geq 0$, $\langle d, y \rangle < 0$,

(iii) A subset $I \subseteq [n]$ and a vector $p \in \pi_I(W)$ such that $\|L_I^W(p)\|_\infty > M\|p\|_1$.

Oracle 1: $\text{Prox-Feas-Solver}(W, d, M, \varepsilon)$

The running time is stated as follows. Recall the definition of $\Psi(A)$ from (5).

Lemma V.1. *There exists an $O(\Psi(A) \cdot \log(M + n) + nm^{\omega-1+o(1)})$ time algorithm, that either returns a solution to $\text{Prox-Feas}(W, d, M, \varepsilon)$, or concludes that (ii) or (iii) should be the outcome of $\text{Prox-Feas-Solver}(W, d, M, \varepsilon)$. In the latter case, these outcomes can be obtained in additional time $O(nm^2 + n^{\omega+o(1)})$.*

The next lemma will be the key technical tool of the algorithm. It allows to solve $\text{Feas-LP}(W, d, M)$ by combining an approximate solution to $\text{Prox-Feas}(W, d', M, \varepsilon)$ for some $d' \in W + d$ with an exact solution to $\text{Feas-LP}(W, d, M)$ obtained recursively from a smaller system.

Recall that for a set $K \subseteq [n]$, $\text{cl}(K)$ denotes the closure of K i.e., the unique largest set $J \subseteq [n]$ such that $K \subset J$ and $\text{rk}(A_J) = \text{rk}(A_K)$.

We select a set K of indices i where x_i is very large in the approximate solution x ; for such indices, proximity guarantees that there must be a feasible solution $x^* \in W + d$, $x^* \geq 0$ with $x_i^* > 0$. We project out all these indices, along with all other indices $J = \text{cl}(K) \setminus K$ in their closure, and recurse on the remaining index set I . We note that the purpose of the set J is to avoid creating loops from the recursive instances.

The choice of the proximity bounds allow us to ‘stitch together’ the solution obtained on $\pi_I(W)$ from the recursive call with the approximate solution x to a feasible solution to the original system. Roughly speaking, the amount of change required to cancel out all negative coordinates in x_I is small enough so that x remains positive on K .

An important feature in the scheme is the choice of the vector d' for the approximate system. This will be either $d' = d$ or $d' = d/W$; hence $W + d' = W + d$. However, this choice makes a difference due to the proximity bounds: the system $\text{Feas-LP}(W, d, M)$ features $\|d^-\|_1$ as well as a bound on $\|x - d\|_\infty$.

In particular, if $\|d^-\|_1$ is ‘too big’, then we may end up with an empty index set K and cannot recurse. In this case, we swap to $d' = d/W$; otherwise, we keep $d' = d$. We note that always swapping to $d' = d/W$ does not work either: $\text{Feas-LP}(W, d, M)$ features the bound $\|x - d\|_\infty$, and using $\|x - d/W\|_\infty$ in the approximate system may move us too far from d . Fortunately, the bad cases for these two choices turn out to be complementary.

Lemma V.2. *Let $M, \varepsilon > 0$ such that $\varepsilon \leq 1/(16M^4n^4)$, let $d \in \mathbb{R}^n$ and define*

$$d' = \begin{cases} d & \text{if } \|d^-\|_1 < \max\left\{M\|d/W\|_1, \frac{\|d\|_\infty}{4M^2n}\right\}, \\ d/W & \text{otherwise.} \end{cases}$$

Let x be a feasible solution to $\text{Prox-Feas}(W, d', M, \varepsilon)$, and

let

$$\begin{aligned} K &= \{i \in [n] : x_i \geq 16n^2 M^3 \|x^-\|_1\}, \\ J &= \text{cl}(K) \setminus K, \\ I &= [n] \setminus \text{cl}(K). \end{aligned} \quad (6)$$

Then $K \neq \emptyset$. Further, if $\text{Feas-LP}(\pi_I(W), x_I, M)$ is feasible, then let $w \in \mathbb{R}^I$ be any feasible solution, and let

$$x' = x + L_{I \cup J}^W((w - x_I, x_J^-)).$$

Then, either x' is feasible to $\text{Feas-LP}(W, d, M)$, or $\|L_{I \cup J}^W((w - x_I, x_J^-))\|_\infty > M\|(w - x_I, x_J^-)\|_1$, that is, $M < \kappa_W$.

If $y \in \pi_I(W)^\perp$, $y \geq 0$, $\langle x_I, y \rangle < 0$ is an infeasibility certificate to $x' \in \pi_I(W) + x_I$, $x_I \geq 0$, then $y' = (y, 0_{J \cup K})$ is an infeasibility certificate to $x \in W + d$, $x \geq 0$.

Algorithm 1: FEAS-ALG

Input : $A \in \mathbb{R}^{m \times n}$, $W = \ker(A) \subseteq \mathbb{R}^n$, $\text{rk}(A) = m$, $d \in \mathbb{R}^n$, $M \geq 2$.

Output: One of the following: (i) a solution x to $\text{Feas-LP}(W, d, M)$; (ii) a Farkas certificate of infeasibility, or (iii) a lifting certificate that $M < \kappa_W$.

```

1 if  $\|d^-\|_1 \geq \max\{M\|d/W\|_1, \|d\|_\infty/4M^2n\}$  then
2    $d \leftarrow d/W$ ;
3 if  $d \geq 0$  then return solution  $d$ ;
4  $\varepsilon \leftarrow 1/(2Mn)^4$ ;
5 switch  $\text{Prox-Feas-Solver}(W, d, M, \varepsilon)$  do
6   case (i)
7      $x \leftarrow \text{Prox-Feas-Solver}(W, d, M, \varepsilon)$ ;
8      $K \leftarrow \{i \in [n] : x_i \geq 16n^2 M^3 \|x^-\|_1\}$ ;
9      $J \leftarrow \text{cl}(K) \setminus K$ ;
10     $I \leftarrow [n] \setminus (J \cup K)$ ;
11    switch  $\text{output of FEAS-ALG}(\pi_I(W), d_I, M)$  do
12      case (i)
13         $x' \leftarrow \text{FEAS-ALG}(\pi_I(W), d_I, M)$ ;
14         $w \leftarrow L_{I \cup J}^W(x' - x_I, x_J^-)$ ;
15        if  $\|w\|_\infty \leq M\|(x' - x_I, x_J^-)\|_1$  then return
16           $x + w$ ;
17        else return lifting certificate of  $M < \kappa_W$ . ;
18      case (ii)
19         $y \leftarrow \text{FEAS-ALG}(\pi_I(W), d_I, M)$ ;
20         $y' \leftarrow (y, 0_{J \cup K})$ ;
21        return Farkas certificate  $y'$  to  $\text{Feas-LP}(W, d, M)$ 
22      case (iii) return lifting certificate of  $M < \kappa_W$ ;
23   case (ii) return Farkas certificate ;
24   case (iii) return lift. certificate of  $M < \kappa_W$ ;

```

The overall feasibility algorithm is given in Algorithm 1, a recursive implementation of Lemma V.2. The output can be (i) a feasible solution to $\text{Feas-LP}(W, d, M)$; (ii) a Farkas

certificate of infeasibility, or (iii) a lifting certificate of $M < \kappa_W$. The latter will always be of the form of an index set $I \subseteq [n]$ and a vector $p \in \pi_I(W)$ such that $\|L_I^W(p)\|_\infty > M\|p\|_1$. In this case, we can restart the entire algorithm, after updating M to $\max\{\|L_I^W(p)\|_\infty/\|p\|_1, M^2\}$.

The algorithm calls Oracle 1. For outputs (ii) and (iii), we return the Farkas certificate or the lifting certificate for $M < \kappa_W$. For output (i), we construct the sets I , J , and K and recurse on $\pi_I(W)$, as in Lemma V.2.

We are now ready to state the central theorem of this section, which in particular proves Theorem I.1.

Theorem V.3. *Algorithm 1 is correct. If $M > \kappa_W$ and the system $x \in W + d$, $x \geq 0$ is feasible, then the algorithm returns a solution in time $O(m \Psi(A) \log(M+n) + mn^{\omega+o(1)})$. If the system is infeasible or $M < \kappa_W$, then a Farkas certificate or a lifting certificate can be obtained in additional running time $O(nm^2 + n^{\omega+o(1)})$.*

VI. THE OPTIMIZATION ALGORITHM

In this section, we show how $\text{Primal-Dual}(W, d, c)$ can be solved using an approximate LP solver. As in the feasibility algorithm, we let M denote our current upper estimate on $2\kappa_W$. We present an algorithm that comprises an Inner and an Outer Loop. The calls to the approximate LP solver will happen inside the Inner Loop.

The outer loop gives an algorithmic implementation of Theorem III.6. The subroutine $\text{INNERLOOP}(W, d, c, M)$ returns a solution $(\tilde{d}, \tilde{x}, s)$, where \tilde{d} is a ‘perturbed’ version of d , and (\tilde{x}, s) are optimal solutions to $\text{Primal-Dual}(W, \tilde{d}, c)$. We get $(\tilde{d}, \tilde{x}, \tilde{s})$ as solutions to the following system.

$$\begin{aligned} \|\tilde{d} - d\|_1 &\leq \frac{\|\tilde{x}\|_\infty}{4n^2 M^2} \\ \tilde{x} &\in W + \tilde{d} \\ \tilde{s} &\in W^\perp + c \\ \langle \tilde{x}, \tilde{s} \rangle &= 0 \\ \tilde{x}, \tilde{s} &\geq 0. \end{aligned} \quad (\text{F-Primal}(W, d, c, M))$$

The subroutine will be described in Section VI-B; we now state the running time.

Theorem VI.1. *Assume we are given a matrix $A \in \mathbb{R}^{m \times n}$, vectors $c \in \mathbb{R}^n$ and $d \in \mathbb{R}_+^n$; let $W = \ker(A)$, and M be an estimate on κ_W . There exists an $O(n \Psi(A) \log(M+n) + n^{\omega+1+o(1)})$ time algorithm (Algorithm 3) that returns a solution $(\tilde{d}, \tilde{x}, \tilde{s})$ to $\text{F-Primal}(W, d, c, M)$, or decides that the system is either primally infeasible or $M < \kappa_W$. To obtain a Farkas certificate of primal infeasibility or a certificate that $M < \kappa_W$ is obtained in additional time $O(n^3 m)$.*

The overall algorithm described in Section VI-A repeatedly calls INNERLOOP to set primal and dual variables to 0 according to Theorem III.6, and recurses to lower dimensional subspaces. The final optimal solutions are obtained via calling the feasibility algorithm on both primal and

dual side. The drawback of this variant is that, in case the algorithm fails, we do not obtain lifting certificates of $M < \kappa_W$ —a guarantee we can achieve for feasibility in Section V. A post-processing, as described in Lemma VI.5 is able to create a certificate of $M < \kappa_W$, at an additional computational expense.

A. The Outer Loop

Consider an instance of $\text{Primal-Dual}(W, d, c)$ and an estimate M on κ_W . We first use the feasibility algorithm and check if both systems $x \in W + d$, $x \geq 0$ and $s \in W^\perp + c$, $s \geq 0$ are feasible. For the remainder of this section, let us assume both these systems are feasible, and consequently, $\text{Primal-Dual}(W, d, c)$ is also feasible. Moreover, we can write an equivalent system with nonnegative $d \geq 0$. (We could also impose $c \geq 0$, but this will not be used).

The overall algorithm is presented in Algorithm 2. We let (W^0, d^0, c^0) denote the original input, where W^0 is a subspace of \mathbb{R}^n , and $d^0, c^0 \in \mathbb{R}^n$, $d^0 \geq 0$. We will maintain an index set $I \subseteq [n]$, initialized as $I = [n]$. We gradually move every index into the set B or N . We apply Theorem III.6 with thresholds $\tau = \|x\|_\infty / (3n^2M)$ and $T = \|x\|_\infty / n$. The bound $\|\tilde{d} - d\|_1 \leq \|\tilde{x}\|_\infty / 4n^2M^2$ in $F\text{-Primal}(W, d, c, M)$ guarantees that these are suitable choices. Assuming that $M \geq \kappa_W$, we obtain $s_B^* = 0$ for every dual optimal solution s^* , and that there exists a primal optimal solution x^* with $x_N^* = 0$. Note the asymmetry between the two sides; the weaker guarantee on N already suffices for correctness.

At every iteration, we have an index set $I \subseteq [n]$ of ‘undecided indices’ and a subspace $W \subseteq \mathbb{R}^I$. We consider the partition $I = I_L \cup I_M \cup I_S^+ \cup I_S^0$ according to Theorem III.6, and add $I_L \cup I_M$ to B and I_S^0 to N . The optimal solution x'' guaranteed in the theorem implies that the optimum value is the same on W and $W' = W \cap \mathbb{R}_{I_L \cup I_S \cup I_S^+}^I$, i.e. the subspace where all entries in I_S^0 are forced to 0. We then update $I = I_S^+$ as the remaining set of undecided indices and recurse on the subspace $\pi_I(W')$.

The algorithm terminates when d is contained in W . The remaining indices I are split up between B and N based on whether they are in the support of the optimal dual solution of the perturbed system. Finally, we obtain the primal and dual solutions by solving feasibility problems on the subsets B and N . If both are feasible, they form a complementary pair of primal and dual solutions, and hence they are optimal. In case of a failure, we conclude that the underlying assumption $M \geq \kappa_W$ was wrong.

Theorem VI.2. *Assuming that $\text{Primal-Dual}(W^0, d^0, c^0)$ is both primal and dual feasible, Algorithm 2 either finds an optimal solution to $\text{Primal-Dual}(W^0, d^0, c^0)$ or correctly concludes that $M < \kappa_{W^0}$ by at most m calls to the subroutine INNERLOOP. The total runtime is $O(mn \Psi(A) \log(M +$*

Algorithm 2: Optimization Algorithm

Input : $W^0 \subseteq \mathbb{R}^n$; $c^0 \in \mathbb{R}^n$, $d^0 \in \mathbb{R}_+^n$ such that $\text{Primal-Dual}(W^0, d^0, c^0)$ is feasible, and $M \geq 2$.

Output: Solution (x^*, s^*) to $\text{Primal-Dual}(W^0, d^0, c^0)$ or the answer $M < \kappa_{W^0}$.

```

1  $W \leftarrow W^0$  ;  $d \leftarrow d^0$  ;  $c \leftarrow c^0 / W^\perp$  ;
2  $I \leftarrow [n]$  ;  $B \leftarrow \emptyset$  ;  $N \leftarrow \emptyset$  ;
3 while  $I \neq \emptyset$  and  $d \notin W$  do
4   if INNERLOOP( $W, d, c, M$ ) returns  $M < \kappa_W$  then
     return  $M < \kappa_W$  ;
5    $(\tilde{d}, \tilde{x}, \tilde{s}) \leftarrow \text{INNERLOOP}(W, d, c, M)$  ;
6    $I_L \leftarrow \{i \in I : \tilde{x}_i > \|\tilde{x}\|_\infty / n\}$  ;
7    $I_M \leftarrow \{i \in I : \|\tilde{x}\|_\infty / n \geq \tilde{x}_i > \|\tilde{x}\|_\infty / (3n^2M)\}$  ;
8    $I_S \leftarrow \{i \in I : \|\tilde{x}\|_\infty / (3n^2M) \geq \tilde{x}_i\}$  ;
9    $I_S^0 \leftarrow I_S \cap \text{cl}(I_L)$  ;  $I_S^+ \leftarrow I_S \setminus \text{cl}(I_L)$  ;
10   $B \leftarrow B \cup I_L \cup I_M$  ;  $N \leftarrow N \cup I_S^0$  ;
11   $W' \leftarrow W \cap \mathbb{R}_{I_L \cup I_M \cup I_S^+}^I$  ;
12   $I \leftarrow I_S^+$  ;
13   $W \leftarrow \pi_I(W')$  ;  $d \leftarrow d_I$  ;  $c \leftarrow \tilde{s}_I$  ;
14   $N \leftarrow N \cup (I \cap \text{supp}(\tilde{s}))$  ;  $B \leftarrow B \cup (I \setminus \text{supp}(\tilde{s}))$  ;
15  if FEAS-ALG( $W_B^0, d^0$ ) and FEAS-ALG( $(W^0)_N^\perp, c^0$ )
     are feasible then
16     $x^* \leftarrow \text{FEAS-ALG}(W_B^0, d^0)$  ;
17     $s^* \leftarrow \text{FEAS-ALG}((W^0)_N^\perp, c^0)$  ;
18    return  $(x^*, s^*)$ 
19 else return  $M < \kappa_{W^0}$  ;
```

$n) + mn^{\omega+1+o(1)})$ to find a primal-dual optimal pair. Obtaining a lifting certificate requires additional time $O(n^3m^2)$.

B. The Inner Loop

For the Inner Loop and $d \geq 0$ we formulate the stronger version $F\text{-Primal-Prox}(W, d, c, M, \varepsilon)$ of $F\text{-Primal}(W, d, c, M)$, which maintains dual proximity and therefore—in similar vein as the feasibility algorithm—only needs an oracle with precision $(Mn)^{-O(1)}$ for recursive calls. We define $F\text{-Primal-Prox}(W, d, c, M, \varepsilon)$:

$$\begin{aligned}
x \in W + d, \quad s \in W^\perp + c, \quad s \geq 0, \\
\|s - c\|_\infty \leq 16M^2n \|c_{\Lambda(c,d)}\|_1, \\
\|x_{\Lambda(x,s)}\|_\infty \leq 2\varepsilon n \|x\|_\infty.
\end{aligned}$$

Given an output (x, s) to this system with $\varepsilon \leq 1/(32M^4n^4)$, we obtain a solution to $F\text{-Primal}(W, d, c, M)$ as

$$\tilde{x}_i = \begin{cases} 0 & \text{if } i \in \Lambda(x, s), \\ x_i & \text{otherwise,} \end{cases}, \quad \tilde{d} = d - x + \tilde{x}, \quad \tilde{s} = s. \quad (7)$$

We will assume that the following oracle (Oracle 2) is available, that returns a solution (\tilde{c}, x, s) to the system $\text{Prox-Opt}(W, d, c, M, \varepsilon)$, a primal or dual infeasibility certificate,

or a lifting certificate. For the input, we require the nonnegativity $c \geq 0$. This will be perturbed to \tilde{c} , and (x, s) will be near-optimal and near-feasible primal and dual solutions with respect to the perturbed system, satisfying a primal proximity constraint.

Input : $W \subseteq \mathbb{R}^n, c \in \mathbb{R}_+^n, d \in \mathbb{R}^n, M \geq 2, \varepsilon > 0$.

Output: One of the following:

(i) A solution to the system $\text{Prox-Opt}(W, d, c, M, \varepsilon)$:

$$x \in W + d, s \in W^\perp + \tilde{c}, s \geq 0, c - \tilde{c} \geq 0,$$

$$\|x_{\Lambda(x,s)}\|_\infty \leq \varepsilon \|d_{\Lambda(d,c)}\|_1$$

$$\|x - d\|_\infty \leq 3M^2 n \|d_{\Lambda(d,c)}\|_1$$

$$\|c - \tilde{c}\|_\infty \leq \frac{\varepsilon}{n} \|c/W^\perp\|_1$$

(ii) A vector $y \in W^\perp, y \geq 0, \langle d, y \rangle < 0$,

(iii) A vector $x \in W, x \geq 0, \langle c, x \rangle < 0$,

(iv) A lifting certificate of $M < \kappa_W = \kappa_{W^\perp}$.

Oracle 2: $\text{Prox-Opt-Solver}(W, d, c, M, \varepsilon)$

Lemma VI.3. Assume we are given a matrix $A \in \mathbb{R}^{m \times n}$, vectors $c \in \mathbb{R}_+^n$ and $d \in \mathbb{R}^n$; let $W = \ker(A)$, and M be an estimate on κ_W . Further, let $0 < \varepsilon < 1$. There exists an $O(\Psi(A) \cdot \log(M+n) + nm^{\omega-1+\alpha(1)})$ time algorithm, that either returns a solution to $\text{Prox-Opt}(W, d, c, M, \varepsilon)$ or concludes that (ii), (iii) or (iv) should be the outcome of $\text{Prox-Opt-Solver}(W, d, c, M, \varepsilon)$. These latter outcomes require an additional computational time $O(n^3 m)$.

$\text{INNERLOOP}(W, d, c, M, \varepsilon)$ recursively calls itself and Oracle 2, while maintaining dual proximity. The Oracle will be called for the dual system.

Lemma VI.4. Let M be an estimate on $\kappa_W, 0 < \varepsilon < 1/(32M^4 n^4)$. Let us define

$$c' = \begin{cases} c & \text{if } \|c_{\Lambda(c,d)}\|_1 < \max \left\{ M \|c/W^\perp\|_1, \frac{\|c\|_\infty}{4M^2 n} \right\}, \\ c/W^\perp & \text{otherwise.} \end{cases}$$

Let (x, s) be a feasible solution to $\text{Prox-Opt}(W^\perp, c', d, M, \varepsilon)$ and let

$$I = \{i \in [n] : s_i \leq 16n^3 M^3 \|s_{\Lambda(s,x)}\|_1\}, \quad J = [n] \setminus I.$$

Then, the following hold:

(i) If $I = \emptyset$, then we must have $s \geq 0$ and $d \in W$, and $(0, s)$ is feasible to $F\text{-Primal-Prox}(W, d, c, M, \varepsilon)$.

(ii) If $I \neq \emptyset$, then let (w, z) be a solution to $F\text{-Primal-Prox}(W_I, x_I, s_I, M, \varepsilon)$, and define

$$\tilde{x} = (w, x_J) + d - \tilde{d}, \quad \text{and} \quad \tilde{s} = s + L_I^{W^\perp}(z - s_I).$$

Then either (\tilde{x}, \tilde{s}) is feasible to $F\text{-Primal-Prox}(W, d, c, M, \varepsilon)$ or we obtain a lifting certificate of $M < \kappa_{W^\perp} = \kappa_W$.

(iii) $J \neq \emptyset$.

Algorithm 3 implements the recursive calls in accordance with Lemma VI.4. This is the algorithm asserted in Theorem VI.1.

Algorithm 3: INNERLOOP

Input : $W \subseteq \mathbb{R}^n, c \in \mathbb{R}^n, d \in \mathbb{R}_+^n$ such that $F\text{-Primal-Dual}(W, d, c)$ is feasible, $M \geq 1$.

Output: A solution $(\tilde{d}, \tilde{x}, \tilde{s})$ to $F\text{-Primal-Prox}(W, d, c, M, \varepsilon)$ or lifting certificate of $M < \kappa_W$.

```

1  $\varepsilon \leftarrow 1/(32M^4 n^4)$ ;
2 if  $\|c_{\Lambda(c,d)}\|_1 \geq \max \left\{ M \|c/W^\perp\|_1, \frac{\|c\|_\infty}{4M^2 n} \right\}$  then
    $c \leftarrow c/W^\perp$ ;
3 if  $\text{Prox-Opt-Solver}(W^\perp, c, d, M, \varepsilon)$  in (i) then
4    $(\tilde{d}, s, x) \leftarrow \text{Prox-Opt-Solver}(W^\perp, c, d, M, \varepsilon)$ ;
5    $I = \{i : s_i < 16(Mn)^3 \|s_{\Lambda(s,x)}\|_1\}, J = [n] \setminus I$ ;
6   if  $I = \emptyset$  then return  $(x, s)$ ;
7   else
8     if  $\text{INNERLOOP}(W_I, x_I, s_I, M, \varepsilon)$  returns a solution
9       then
10       $(w, z) \leftarrow \text{INNERLOOP}(W_I, x_I, s_I, M, \varepsilon)$ ;
11      if  $\|L_I^{W^\perp}(z - s_I)\|_\infty \leq M \|z - s_I\|_1$  then
12       $\tilde{x} \leftarrow (w, x_J) + d - \tilde{d}$ ;
13       $\tilde{s} \leftarrow s + L_I^{W^\perp}(z - s_I)$ ;
14      return  $(\tilde{x}, \tilde{s})$ ;
15      else return Lifting certificate of  $M < \kappa_W$ ;
16 else return Lifting certificate of  $M < \kappa_W$ ;

```

C. Certificate for the wrong guess

Algorithm 2 only provides the verdict $M < \kappa_W$ without providing the corresponding certificate. The INNERLOOP is able to provide a certificate, but if infeasibility is detected in the calls to the FEAS-ALG in lines Line 16 and Line 17 then this means that the partition $B \cup N$ is wrong. Note that failed calls to FEAS-ALG could also result in $M < \kappa_W$ for which the feasibility solvers provide a certificate. We are nonetheless able to provide a certificate of $M < \kappa_W$.

Lemma VI.5. In line 17 of Algorithm 2, the dual feasibility solver always succeeds to find a dual solution or finds a certificate of $M < \kappa_W$. If the primal feasibility solver in line 16 fails to find a primal feasible solution, then we can find a lifting certificate of $M < \kappa_W$ in $O(n^3 m^2)$ time.

REFERENCES

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

- [2] N. Megiddo, "Towards a genuinely polynomial algorithm for linear programming," *SIAM Journal on Computing*, vol. 12, no. 2, pp. 347–353, 1983.
- [3] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *Journal of the ACM (JACM)*, vol. 36, no. 4, pp. 873–886, 1989.
- [4] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, vol. 41, no. 2, pp. 338–350, 1993.
- [5] É. Tardos, "A strongly polynomial minimum cost circulation algorithm," *Combinatorica*, vol. 5, pp. 247–255, Sep 1985.
- [6] L. A. Végh, "A strongly polynomial algorithm for generalized flow maximization," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 179–211, 2017.
- [7] N. Olver and L. A. Végh, "A simpler and faster strongly polynomial algorithm for generalized flow maximization," in *Proceedings of the Forty-Ninth Annual ACM Symposium on Theory of Computing (STOC)*, pp. 100–111, 2017.
- [8] Y. Ye, "A new complexity result on solving the Markov decision problem," *Mathematics of Operations Research*, vol. 30, no. 3, pp. 733–749, 2005.
- [9] Y. Ye, "The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate," *Mathematics of Operations Research*, vol. 36, no. 4, pp. 593–603, 2011.
- [10] S. Iwata, L. Fleischer, and S. Fujishige, "A combinatorial strongly polynomial algorithm for minimizing submodular functions," *Journal of the ACM (JACM)*, vol. 48, no. 4, pp. 761–777, 2001.
- [11] D. Dadush, L. A. Végh, and G. Zambelli, "Geometric rescaling algorithms for submodular function minimization," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 832–848, SIAM, 2018.
- [12] É. Tardos, "A strongly polynomial algorithm to solve combinatorial linear programs," *Operations Research*, pp. 250–256, 1986.
- [13] A. J. Hoffman, "On approximate solutions of systems of linear inequalities," *J. Res. Natl. Bur. Stand.*, vol. 49, no. 4, pp. 263–265, 1952.
- [14] W. Cook, A. M. Gerards, A. Schrijver, and É. Tardos, "Sensitivity theorems in integer linear programming," *Mathematical Programming*, vol. 34, no. 3, pp. 251–264, 1986.
- [15] 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.
- [16] S. A. Vavasis and Y. Ye, "Condition numbers for polyhedra with real number data," *Operations Research Letters*, vol. 17, pp. 209–214, 06 1995.
- [17] Y. Ye, "Improved complexity results on solving real-number linear feasibility problems," *Mathematical Programming*, vol. 106, pp. 339–363, Apr. 2006.
- [18] G. Stewart, "On scaled projections and pseudoinverses," *Linear Algebra and its Applications*, vol. 112, pp. 189 – 193, 1989.
- [19] 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.
- [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.
- [21] 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.
- [22] Y. T. Lee and A. Sidford, "Solving linear programs with $\tilde{O}(\sqrt{\text{rank}})$ linear system solves." arXiv preprint 1910.08033, 2019.
- [23] 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.
- [24] J. van den Brand, "A deterministic linear program solver in current matrix multiplication time," in *Proceedings of the Symposium on Discrete Algorithms (SODA)*, 2020.
- [25] J. van den Brand, Y. T. Lee, A. Sidford, and Z. Song, "Solving tall dense linear programs in nearly linear time," 2020.
- [26] Y. T. Lee, Z. Song, and Q. Zhang, "Solving empirical risk minimization in the current matrix multiplication time," in *COLT*, vol. 99 of *Proceedings of Machine Learning Research*, (Phoenix, USA), pp. 2140–2157, PMLR, 25–28 Jun 2019.
- [27] S. Jiang, Z. Song, O. Weinstein, and H. Zhang, "Faster Dynamic Matrix Inverse for Faster LPs," arXiv:2004.07470 [cs], Apr. 2020.
- [28] J. C. Ho and L. Tunçel, "Reconciliation of various complexity and condition measures for linear programming problems and a generalization of Tardos' theorem," in *Foundations of Computational Mathematics*, pp. 93–147, World Scientific, 2002.
- [29] S. Mizuno, M. Todd, and Y. Ye, "On adaptive-step primal-dual interior-point algorithms for linear programming," *Mathematics of Operations Research - MOR*, vol. 18, pp. 964–981, 11 1993.
- [30] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2. Springer Science & Business Media, 2012.
- [31] A. Frank and É. Tardos, "An application of simultaneous diophantine approximation in combinatorial optimization," *Combinatorica*, vol. 7, no. 1, pp. 49–65, 1987.
- [32] L. G. Khachiyan, "On the complexity of approximating extremal determinants in matrices," *J. Complex.*, vol. 11, pp. 138–153, Mar. 1995.
- [33] L. Tunçel, "Approximating the complexity measure of Vavasis-Ye algorithm is NP-hard," *Mathematical Programming*, vol. 86, pp. 219–223, Sep 1999.
- [34] I. Dikin, "Iterative solution of problems of linear and quadratic programming," *Doklady Akademii Nauk*, vol. 174, no. 4, pp. 747–748, 1967.
- [35] M. J. Todd, "A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm," *Operations Research*, vol. 38, no. 6, pp. 1006–1018, 1990.
- [36] D. P. O'Leary, "On bounds for scaled projections and pseudoinverses," *Linear Algebra and its Applications*, vol. 132, pp. 115–117, Apr. 1990.
- [37] S. A. Vavasis, "Stable numerical algorithms for equilibrium systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 4, pp. 1108–1131, 1994.
- [38] J. Pena, J. C. Vera, and L. F. Zuluaga, "New characterizations of Hoffman constants for systems of linear constraints," *Mathematical Programming*, pp. 1–31, 2020.
- [39] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.