Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

On finding exact solutions of linear programs in the oracle model *

Daniel Dadush[1], László A. Végh[2], and Giacomo Zambelli[2]

[1]Centrum Wiskunde & Informatica, The Netherlands, dadush@cwi.nl
[2]Department of Mathematics, London School of Economics and Political Science,
{l.vegh,g.zambelli}@lse.ac.uk

Abstract

We consider linear programming in the oracle model: $\min c^\top x$ s.t. $x \in P$, where the polyhedron $P = \{x \in \mathbb{R}^n : Ax \le b\}$ is given by a separation oracle that returns violated inequalities from the system $Ax \le b$. We present an algorithm that finds exact primal and dual solutions using $O(n^2 \log(n/\delta))$ oracle calls and $O(n^4 \log(n/\delta) + n^5 \log\log(1/\delta))$ arithmetic operations, where δ is a geometric condition number associated with the system (A, b). These bounds do not depend on the cost vector c.

The algorithm works in a black box manner, requiring a subroutine for approximate primal and dual solutions; the above running times are achieved when using the cutting plane method of Jiang, Lee, Song, and Wong (STOC 2020) for this subroutine. Whereas approximate solvers may return primal solutions only, we develop a general framework for extracting dual certificates based on the work of Burrell and Todd (Math. Oper. Res. 1985).

Our algorithm works in the real model of computation, and extends results by Grötschel, Lovász, and Schrijver (Prog. Comb. Opt. 1984), and by Frank and Tardos (Combinatorica 1987) on solving LPs in the bit-complexity model. We show that under a natural assumption, simultaneous Diophantine approximation in these results can be avoided.

1 Introduction

We consider linear programming (LP) in the oracle model. Let $P = \{x \in \mathbb{R}^n : Ax \le b\}$ be a polyhedron for $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$; the i-th row of A is denoted by $a_i^\top$. In the *linear feasibility problem*, our goal is to find a feasible $x \in P$ or the conclusion $P = \emptyset$; in the *linear optimization problem*, we are given an objective function $c \in \mathbb{R}^n$, and we want to find a solution $x \in P$ maximizing $c^\top x$, or the conclusion that the problem is infeasible or that it is unbounded. The focus of this paper is on *exact* rather than approximate solutions to these problems, along with dual certificates.

We say that the LP is *explicitly given*, if the matrix A and vector b are given as part of the input. In the *oracle model*, these are represented implicitly, via a separation oracle. Our main example will be what we call a *polyhedral separation oracle*: given $\bar{x} \in \mathbb{R}^n$, the oracle returns the answer $\bar{x} \in P$, or a violated constraint $b_i > a_i^\top \bar{x}$ from the system $Ax \le b$; see Section 2 for the discussion of different oracle models. The number of constraints m may be exponentially large in n.

LP algorithms in the Turing model For an explicit rational input (A, b, c), the first polynomial time LP algorithm in the Turing model was given by Khachiyan in 1979, using the ellipsoid method [22]. Degeneracy, i.e., P being contained in a lower dimensional subspace, is a particular challenge for the ellipsoid method, since the volumetric progress measure is not directly applicable. Khachiyan used perturbation, based on bit-complexity arguments, to obtain a system $\tilde{P}$ such that $\tilde{P} = \emptyset$ if and only if $P = \emptyset$, and $\tilde{P}$ is full-dimensional whenever nonempty.

Grötschel, Lovász, and Schrijver [16, 17] used the ellipsoid method to tackle LPs given implicitly by a strong separation oracle, and developed the theory of *rational polyhedra*. They showed that for rational polyhedra with bounded 'facet complexity', the ellipsoid method either finds a feasible solution in polynomial time, or a lower dimensional subspace containing P can be identified using *simultaneous Diophantine approximation*, by an application of the basis reduction algorithm by Lenstra, Lenstra, and Lovász [27].

*The full version of the paper is available on arXiv. 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).

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

LP in the real model of computation In the context of LP it is natural to use a *real model of computation*: we assume the input is given by real numbers, each requiring unit storage, and one can perform a set of arithmetic operations in unit time. Arithmetics include basic operations (+, −, ×, /); certain models allow further operations such as $\sqrt{}$ and log. In the context of LP, Traub and Woźniakowski [35] advocated using such a model. A computational theory over reals was developed by Blum, Shub and Smale [5], see also the book [4]. The ultimate goal for an explicit LP in this model is to develop a *strongly polynomial algorithm*: one where the number of arithmetic operations only depends on the number of variables n and constraints m.[1] This was listed as the 9th question by Smale on his list of eighteen mathematical challenges for the 21st century [32]. The existence of a strongly polynomial algorithm remains wide open; such algorithms are only known for special classes of LP.

For explicitly given LPs, interior point methods (IPMs) yield algorithms with excellent theoretical and practical performance; for recent developments, as well as for pointers to the literature, see [8, 25, 37, 38]. IPMs naturally work in the real model; most variants output approximate solutions. In the Turing model, an approximate solution with sufficiently high accuracy can be converted to an exact optimal solution. Consider $\min c^\top x,\ Ax \le b,\ A \in \mathbb{R}^{m\times n}$, and let L denote the total bit-complexity of (A, b, c). Then, van den Brand [37] gives an $O(m^\omega L \log^{O(1)}(m))$ deterministic algorithm, and van den Brand et al. [38] gives a randomized $O((mn + n^3)L \log^{O(1)}(m))$ randomized algorithm for finding exact primal and dual optimal solutions.

Tardos [34] gave an algorithm in the Turing model with running time dependence only on the bit-complexity of A, but independent of b and c. This was generalized to the real model of computation by Vavasis and Ye [39], who gave a $\mathrm{poly}(n, m, \log \bar{\chi}_A)$ algorithm for solving explicit LPs exactly, where $\bar{\chi}_A$ is a certain condition number associated with the constraint matrix. We discuss more recent developments along these lines in Section 1.2.

LP in the oracle model Several important problems in combinatorial optimization, including matching, network design, and submodular optimization problems, can be formulated by LPs with an exponential number of constraints. For such LPs the explicit description would be exponential; at the same time, one can efficiently find violated constraints for infeasible points. This motivated the development of oracle algorithms by Grötschel, Lovász, and Schrijver [17], based on the ellipsoid method.

Vaidya [36] gave a more efficient cutting plane algorithm in the oracle setting; see [1, 2, 21, 26] for improvements and related algorithms. These algorithms return approximate solutions. Given a convex set $K \subseteq \mathbb{R}^n$ defined by a strong separation oracle and contained in a ball of radius r, the algorithm by Jiang, Lee, Song, and Wong [21] (henceforth referred to as the JLSW algorithm) either finds a feasible point in K, or concludes that K does not contain a ball of radius ε. The algorithm makes $O(n \log(nr/\varepsilon))$ oracle calls and uses $O(n^3 \log(nr/\varepsilon))$ arithmetic operations. This oracle complexity is the same as for Vaidya's algorithm [36] and is asymptotically optimal [29]. Moreover, [21] presents evidence that the arithmetic complexity of $O(n^2)$ operations per oracle call may also be optimal.

Even though ellipsoid and other cutting plane methods deliver approximate solutions only, finding exact solutions is crucial for the applications in combinatorial optimization. Prior to our work, all known results on finding exact LP solutions in the oracle model were based on bit complexity assumptions. Strengthening the result of Grötschel, Lovász, and Schrijver [16, 17], Frank and Tardos [15] showed that, assuming that the matrix A and vector b describing the system $Ax \le b$ are integral with the absolute values of the entries bounded by B, then linear optimization in the oracle model can be solved in time $\mathrm{poly}(n, \log B)$. This is independent of the encoding length of the cost function c. The result is achieved by rounding c using simultaneous Diophantine approximation.

Whereas [17] and [15] gave strongly polynomial algorithms for solving LPs in strongly polynomial time for many important problems such as submodular function minimization or minimum-cost matchings. Still, in contrast to explicitly given LPs, one cannot hope for strongly polynomial algorithms in the oracle model. Indeed, according to the next claim, there may not even exist a deterministic algorithm using $f(n)$ oracle calls for any function f; a proof is provided in the full version.

Proposition 1.1. *There exists no function $f : \mathbb{N} \to \mathbb{N}$ and deterministic algorithm $\mathcal{A}$ that solves the optimization problem $\max c^\top x,\ x \in P$ using at most $f(n)$ oracle calls, where $P \subseteq \mathbb{R}^n$ is a nonempty full-dimensional polyhedron and $c \in \mathbb{R}^n$, and P is accessed via the following oracle: for each $\bar{x} \in \mathbb{R}^n$, either returns $\bar{x} \in P$, or a facet defining inequality violated by $\bar{x}$.*

[1] A strongly polynomial algorithm in the Turing model is further required to be in PSPACE, so that the bit-complexity of the numbers used in the algorithm remains bounded in terms of the input.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

1.1 Our contributions Assume the polyhedron $P = \{x : Ax \leq b\} \subseteq \mathbb{R}^n$ is given by a polyhedral separation oracle, and consider the problem of maximizing $c^\top x$ for an objective function $c \in \mathbb{R}^n$. Our main result is an algorithm such that the number of arithmetic operations and oracle calls is polynomial in n, m, and the logarithm of a certain positive condition number dependent on (A, b), but independent from c. This can be seen as extension and strengthening of the results in [15, 16, 17]. Further, our results imply simpler, more efficient, and potentially more practical algorithms for many applications in the bit-complexity model. We now introduce the main condition number of interest.

Definition 1. *Let $V \subseteq \mathbb{R}^n$ be a set of vectors. We define δ_V to be the largest value such that, for any set of linearly independent vectors $\{v_i : i \in I\} \subseteq V$ and $\lambda \in \mathbb{R}^I$,*

$$\left\| \sum_{i \in I} \lambda_i v_i \right\| \geq \delta_V \max_{i \in I} |\lambda_i| \cdot \|v_i\| .$$

We note that $\delta_V > 0$ if and only if the set $\{v/\|v\| : v \in V\}$ is finite (Lemma 2.2). For a matrix $M \subseteq \mathbb{R}^{m \times n}$, we let δ_M denote the value corresponding to the rows of M.

This condition number was previously studied in the context of the shadow simplex algorithm by Brunsch and Röglin [6], by Eisenbrand and Vempala [14], and by Dadush and Hähnle [9]. They used the following equivalent characterization (see Lemma 2.2): δ_V is the largest number such that for any set of linearly independent vectors $\{v_i : i \in I\}$, the sine of the angle between the vector v_i and the subspace spanned by the vectors $\{v_j : j \in I \setminus \{j\}\}$ is at least δ_V. Further, δ_V bounds the minimum singular value of a matrix with columns v_j (see [6]). For an integer matrix $M \in \mathbb{Z}^{m \times n}$, let Δ_M denote the largest absolute value of any non-singular subdeterminant; then, $1/(n\Delta_M^2) \leq \delta_M$ [6]. In particular, in the rational model, the encoding size of δ_M is polynomially bounded by the sizes of numbers. The quantity δ_A was also studied in the context of lattice base reduction by Seysen [31]. A related condition number appears in the characterization of Hoffman constants [18, 23, 30].

In what follows, for vectors $v \in \mathbb{R}^k, w \in \mathbb{R}^l$, we use the shorthand $(v \mid w) := \begin{pmatrix} v \\ w \end{pmatrix} \in \mathbb{R}^{k+l}$ to denote corresponding column vector and $(v^\top, w^\top)$ to denote the corresponding row vector interpreted as an element in $(\mathbb{R}^{k+l})^*$. For the system $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, we consider the matrix

$$M = \begin{pmatrix} \mathbf{0} & 1 \\ -A & b \end{pmatrix} , \tag{1.1}$$

corresponding to the conic embedding of P as $K = \{(x \mid t) \in \mathbb{R}^{n+1} : M(x \mid t) \geq 0\}$. Let $\delta_{(A,b)+(\mathbf{0},1)}$ denote δ_M for this matrix M.

Our algorithms provide dual certificates of infeasibility and optimality. For a feasibility problem, a *Farkas certificate* of $P = \emptyset$ is a vector of nonnegative coefficients $\lambda \in \mathbb{R}^J_+$ for a subset $J \subseteq [m]$, $|J| \leq n$ such that $\sum_{j \in J} \lambda_j a_j = 0$, $\sum_{j \in J} \lambda_j b_j < 0$. For $c \in \mathbb{R}^n$, the dual polyhedron corresponding to $\max\{c^\top x : x \in P\}$ is $D_c = \{y \in \mathbb{R}^m_+ : A^\top y = c\}$. The LP has a finite optimum if and only if both primal and dual programs are feasible. In this case, a *dual certificate of optimality* for the solution $x^* \in P$ is defined by a subset $J \subseteq [m]$ and a vector of nonnegative coefficients $\lambda \in \mathbb{R}^J_+$ such that $\sum_{j \in J} \lambda_j a_j = c$ and $a_i^\top x^* = b$ for all $j \in J$. By duality theory, such coefficients always exists with $|J| \leq n$.

Our main result shows that the one can find an exact solution in time $O(n)$ times the running time of the best approximate algorithm [21], replacing r/ε by $1/\delta_{(A,b)+(\mathbf{0},1)}$, and an additional polynomial term multiplied by $\log\log(1/\delta_{(A,b)+(\mathbf{0},1)})$.

Theorem 1.1. *Consider the LP problem* $\max\{c^\top x : Ax \leq b\}$ *for $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, given by a polyhedral separation oracle. For parts* (ii) *and* (iii)*, assume that a polyhedral separation oracle for the recession cone* $\mathrm{rec}(P) := \{x \in \mathbb{R}^n : Ax \leq 0\}$ *is also provided.*

(i) A primal feasible solution or a Farkas certificate of infeasibility can be found using $O(n^2 \log(n/\delta_{(A,b)+(\mathbf{0},1)}))$ oracle queries and $O(n^4 \log(n/\delta_{(A,b)+(\mathbf{0},1)}))$ arithmetic operations.

(ii) A dual feasible solution or a Farkas certificate of dual infeasiblity can be found in $O(n^2 \log(n/\delta_A))$ oracle queries and $O(n^4 \log(n/\delta_A) + n^5 \log\log(1/\delta_A))$ arithmetic operations.

(iii) If both primal and dual systems are feasible, then primal and dual optimal solutions can be found in $O(n^2 \log(n/\delta_{(A,b)+(\mathbf{0},1)}))$ oracle queries and $O(n^4 \log(n/\delta_{(A,b)+(\mathbf{0},1)}) + n^5 \log\log(1/\delta_{(A,b)+(\mathbf{0},1)}))$ arithmetic operations.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

A few remarks about the result are in order.

- We use a *black box* approach. The algorithms work in the conic setting via the conic embedding described above, and require a subroutine that produces *'approximate dual certificates'*. The running time stated in Theorem 1.1 refers to the JLSW algorithm [21]. In Section 5, we present a general scheme that allows to extract dual certificates from a broad range of methods, including the ellipsoid method and geometric rescaling methods method [12, 19].
- Assuming P is given by a polyhedral separation oracle, our result strengthens that by Frank and Tardos [15]: for $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ with all entries having absolute value B, for M as in (1.1), the maximum subdeterminant Δ_M is bounded as $\Delta_M \le B^n n^{n/2}$ by the Hadamard–inequality, and we have $\delta_M \ge 1/(n\Delta_M^2) \ge 1/(B^{2n}n^{n+1})$. Thus, our algorithm makes $O(n^3 \log(nB))$ oracle calls to solve a linear optimization program with an arbitrary objective function max $c^\top x$.[2]
- The algorithms in [15, 16, 17] rely on bit-complexity arguments. In contrast, our algorithms are in the real model of computation and are entirely geometric. For the rational settings, our running time bounds depend on the condition number δ_M that can be significantly better than the lower bounds implied by the bit-complexity.
- Cutting planes methods typically require the feasible region to be enclosed in a ball of known radius. In the rational setting, the enclosing radius is usually estimated based on the encoding size of the coefficients. Our method does not require any such assumptions.

Note that solving the dual feasibility problem only depends on δ_A of the constraint matrix A, but not on b or c. One may ask whether also the optimization problem could be solved in time dependent only on A. This would be the analogue in the oracle model of the Vavasis–Ye [39] result, and would be the best one can hope for in the oracle model in light of Proposition 1.1. However, we show that this is not possible; the proof is given in the full version.

Proposition 1.2. *Let θ_A be a condition number associated with a matrix A that remains unchanged by creating a duplicate copy of a row. There exists no function $f : \mathbb{N}\times\mathbb{R} \to \mathbb{N}$ and algorithm $\mathcal{A}$ that solves max $c^\top x$ s.t. $Ax \le b$ for $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ in $f(n, \theta_A)$ oracle queries, assuming the system is given by a polyhedral separation oracle.*

In light of the negative results, Theorem 1.1 is conceptually the best possible one can hope for in the oracle model for linear optimization. The only scope for improvement may be to find algorithms that depend on better condition numbers of (A, b), or use fewer oracle calls or arithmetic operations.

Even though Theorem 1.1 uses a more restrictive oracle model than the standard strong separation oracle assumption, we show that it can reproduce many important results for rational polyhedra in [17]. In particular, simultaneous Diophantine approximation can be avoided in most applications, and dual optimal solutions can be found much more efficiently. These are discussed in the full version.

Reduction to the conic setting The algorithms in Theorem 1.1 are derived from conic optimization problems using the conic embedding.

We recall that a cone $K \subseteq \mathbb{R}^n$ is a convex set that is closed under positive scalings, that is, $\lambda K = K$ for any $\lambda > 0$. We define a conic separation oracle for K, to be an oracle that on input $\bar{x} \in \mathbb{R}^n$, either outputs that $\bar{x} \in K$, or if $\bar{x} \notin K$, outputs a *non-zero* vector $v \in \mathbb{R}^n \setminus \{0\}$ such that $v^\top\bar{x} \le 0$ and $v^\top x \ge 0$, $\forall x \in K$. We do not assume that K is closed, non-empty or even polyhedral in this definition. We note that the requirement $v^\top\bar{x} \le 0$ is automatically satisfied by any separator if $K \ne \emptyset$ (since 0 must be in the closure of K), so it is only a non-trivial requirement when $K = \emptyset$ (this case will be important for the computation of Farkas certificates). We further note that the separator produced by the oracle is not required to be strict if $\bar{x} \notin K$, i.e., we do not require $\bar{v}^\top x > 0$, $\forall x \in K$ (such a separator need not exist). The oracle is however required to exactly decide feasibility in K.

We present black box algorithms using the following subroutine:

[2] We note that we do not generalize [15] for arbitrary oracle settings. The result in [15] is a preprocessing step replacing c by an equivalent $\tilde{c}$ of small encoding length, but does not require any assumptions on the oracle.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Oracle Approx-Conic-Dual
Input: A cone K given by a conic separation oracle, and $\varepsilon > 0$.
Output: Either a point $x \in K$, or an *ε-approximate conic Farkas certificate*, which is defined by a set $\{m_j : j \in J\}$ of vectors returned by the separation oracle, along with multipliers $\lambda \in \mathbb{R}^J_{++}$ such that

$$\left\| \sum_{j \in J} \lambda_j m_j \right\| < \varepsilon, \qquad \sum_{j \in J} \lambda_j \|m_j\| \geq 1. \tag{1.2}$$

We let $\mathcal{T}_o(n,\varepsilon)$ denote the number of oracle calls and $\mathcal{T}_a(n,\varepsilon)$ the number of arithmetic operations of this subroutine. We assume these are of the form $\mathcal{T}_o(n,\varepsilon) = g_o(n)\log^{\nu_o}(n/\varepsilon)$ and $\mathcal{T}_a(n,\varepsilon) = g_a(n)\log^{\nu_a}(n/\varepsilon)$ for some $\nu_o, \nu_a \geq 1$. We assume that the the number $|J|$ of oracle separators involved in the ε-approximate conic Farkas certificate (1.2) is bounded by a function $\tau(n)$. In Section 5 we show the following.

Theorem 1.2. *There exists an oracle polynomial algorithm for* Approx-Conic-Dual *with* $\mathcal{T}_o(n,\varepsilon) = O(n\log(1/\varepsilon))$, $\mathcal{T}_a(n,\varepsilon) = O(n^3\log(1/\varepsilon))$, *and* $\tau(n) = O(n)$.

The above corresponds to the requisite approximate problem we will need to solve certain conic problems exactly. For the purpose of exact solutions, we will require further assumptions on the possible outputs of the oracle as in the preceding section.

For this purpose, we will work with cones of the form $K = \{x \in \mathbb{R}^n : M_T x \geq 0, M_S > 0\}$, where $M \in \mathbb{R}^{m\times n}$, $S \cup T = [m]$ is a (possibly trivial) partition, and $M_S \in \mathbb{R}^{S\times n}$, $M_T \in \mathbb{R}^{T\times n}$ denote the corresponding rows of M. Slightly abusing notation, we let $m_i \in \mathbb{R}^n$, $i \in [m]$, satisfy $m_i^\top = M_{\{i\}}$, i.e., the column vector whose transpose is the i^{th} row of M. Compared to the previous section, note that we allow (and we will need) strict inequalities in the definition of K.

A *polyhedral conic separation oracle* for K is an oracle that, given $\bar{x} \in \mathbb{R}^n$, either returns that $\bar{x} \in K$, or a vector $v \in \mathbb{R}^n$, such that $\exists i \in [m]$ satisfying $v = m_i$ and for which $v^\top \bar{x} < 0$ if $i \in T$ or $v^\top \bar{x} \leq 0$ if $i \in S$. From the perspective of implementation, the separator does not specify the index i, it needs only reveal whether $v^\top$ is a row indexed by S or by T. In the applications, the list of strict inequalities induced by $M_S x > 0$, will in fact be known be in advance and will satisfy $|S| = O(n)$.

We now formulate our three main conic problems. In each case, our goal is to provide algorithms that are oracle polynomial in n and $\log(\delta_M)$. The problems are defined over a closed polyhedral cone $K \subseteq \mathbb{R}^n$ of the form $K = \{x \in \mathbb{R}^n : Mx \geq 0\}$; that is, $S = \emptyset$ as above. The particular oracle assumptions will be detailed in Theorem 1.3. In the first problem, the first row $m_1^\top$ of M plays a special role and is given to us.

- **Strong conic feasibility problem:** either find an $x \in K$ and $m_1^\top x > 0$, or find a $y \in \mathbb{R}^m_+$ with $y_1 = 1$ such that $M^\top y = 0$ certifying that no such x exists.
- **Conic validity problem:** Given $c \in \mathbb{R}^n$, either find a certificate $y \in \mathbb{R}^k_+$ such that $M^\top y = c$ showing that $c^\top x \geq 0$ holds for every $x \in K$, or return an $\bar{x} \in K$ with $c^\top \bar{x} < 0$.
- **Conic minimum-ratio problem:** Given $c, d \in \mathbb{R}^n$, along with a certificate that $d^\top x \geq 0$ is valid for K, expressed by a set $\{m_i : i \in I\}$ for some $I \subseteq [m]$ with $|I| \leq n$ and $y^{(d)} \in \mathbb{R}^m_+$ with $M^\top y^{(d)} = d$ and $\text{supp}(y^{(d)}) = I$. Find

$$\min\left\{\frac{c^\top x}{d^\top x} : x \in K,\, d^\top x > 0\right\}. \tag{1.3}$$

  This is equivalent to finding the maximum value γ^* of $\gamma \in \mathbb{R}$ such that $(c + \gamma d)^\top x \geq 0$ holds for every $x \in K$, if such value exists. In such case, the optimum value of (1.3) is $-\gamma^*$. Depending on the outcome, we ask for the following output.
  - *Optimality:* if γ^* is finite, provide $x^* \in K$ with $(c + \gamma^* d)^\top x^* = 0$, $d^\top x^* > 0$, along with a dual certificate $y \in \mathbb{R}^m_+$ such that $M^\top y = c + \gamma^* d$.
  - *Infeasibility:* if $d^\top x = 0$ for all $x \in K$, then return $y \in \mathbb{R}^m_+$ such that $M^\top y = -d$.
  - *Unboundedness:* if (1.3) is unbounded, return $\bar{x} \in K$ with $d^\top \bar{x} > 0$, and $x \in K$ with $c^\top x < 0$ and $d^\top x = 0$.

We note that the above cases are all disjoint, and also cover all possibilities by standard LP duality.

Remark 1. *Throughout the paper, as above, we often refer to vectors $y \in \mathbb{R}^m$, and require the computation of $M^\top y$, even though M is only implicitly defined by a separation oracle. Whenever we use such notation, what we mean is that y is represented by a set of rows $\{m_i : i \in I\}$ for some $I \subseteq [m]$, $|I| \leq \text{poly}(n)$, and by a vector $\tilde{y} \in \mathbb{R}^I$ such that $y_i = \tilde{y}_i$ for $i \in I$, $y_i = 0$ for $i \notin I$.*

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The strong feasibility problem and the validity problem are special cases of each other: the validity problem is the strong feasibility problem over the matrix $M' = \binom{c^\top}{M}$, whereas the strong feasibility problem is the validity problem for $c = -m_1$. We differentiate them since for the validity problem our goal is to find an algorithm whose running time only depends on n and δ_M, but not on c. The strong feasibility algorithm is also significantly simpler than the validity algorithm.

Theorem 1.3. *For $n \in \mathbb{N}$, $\varepsilon > 0$, assume there exists an oracle polynomial-time algorithm for* Approx-Conic-Dual *using $\mathcal{T}_o(n, \varepsilon)$ oracle calls, $\mathcal{T}_a(n, \varepsilon)$ arithmetic operations, and that $\tau(n)$ is the size of the ε-approximate conic Farkas certificate returned. Letting $K = \{x \in \mathbb{R}^n : Mx \geq 0\}$, $M \in \mathbb{R}^{m\times n}$, the following holds:*

(i) Given a polyhedral conic separation oracle for

$$K_1 = \{x \in \mathbb{R}^n : Mx \geq 0, m_1^\top x > 0\},$$

the strong conic feasibility problem can be solved using $O(n) \cdot \mathcal{T}_o(n, \delta_M/O(n))$ oracle calls and $O(n) \cdot \mathcal{T}_a(n, \delta_M/O(n)) + O(n^3) \cdot \mathcal{T}_o(n, \delta_M/O(n)) + O((n^4 + n^2\tau(n)^2) \log\log(1/\delta_M))$ arithmetic operations.

(ii) Given $c \in \mathbb{R}^n$ and a polyhedral conic separation oracle for

$$K_c = \{x \in \mathbb{R}^n : Mx \geq 0, -c^\top x > 0\},$$

the conic validity problem can be solved using $O(n) \cdot \mathcal{T}_o(n, \delta_M/O(n))$ oracle calls and $O(n) \cdot \mathcal{T}_a(n, \delta_M/O(n)) + O(n^3) \cdot \mathcal{T}_o(n, \delta_M/O(n)) + O((n^5 + n^2\tau(n)^2) \log\log(1/\delta_M))$ arithmetic operations.

(iii) Given $c, d \in \mathbb{R}^n$, $y^{(d)} \in \mathbb{R}^m_+$ such that $d = M^\top y^{(d)}$, $I = \operatorname{supp}(y^{(d)})$, $|I| \leq n$, and polyhedral conic separation oracles for the two cones

$$K_{-d} = \{x \in \mathbb{R}^n : Mx \geq 0, d^\top x > 0\} \quad \text{and} \quad K_I^= = \{x \in \mathbb{R}^n : Mx \geq 0, M_I x = 0\},$$

the conic minimum-ratio problem can be solved using $O(n) \cdot \mathcal{T}_o(n, \delta_M/O(n))$ oracle calls and $O(n) \cdot \mathcal{T}_a(n, \delta_M/O(n)) + O(n^3) \cdot \mathcal{T}_o(n, \delta_M/O(n)) + O((n^5 + n^2\tau(n)^2) \log\log(1/\delta_M))$ arithmetic operations.

Note that if a polyhedral conic separation oracle for K is available, one can implement the required oracles for K_1, K_c, K_{-d} with $O(n)$ additional arithmetic operations. The separation oracle for $K_I^=$ can be implemented with $O(n^2)$ additional arithmetic operations, assuming that a projection matrix to $\ker(M_I)$ is pre-computed. The reason for stating the theorem with the specific oracle requirements in each part is for applicability to Theorem 1.1. Using the standard conic embedding of $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ as $K = \{(x \mid t) \in \mathbb{R}^{m+1} : Mx \geq 0\}$ with M as defined in (1.1), we can use the polyhedral separation oracle for P and $\operatorname{rec}(P)$ to implement all required oracles (with $d = m_1^\top$) in Theorem 1.3. However, we do not directly get separation for K for points of the form $(x \mid 0)$; this is further explained in the proof of Theorem 1.3 in Section 2.1. The proof uses this conic embedding and combines Theorems 1.2 and 1.3.

Application to rational polyhedra Let us now focus on *rational polyhedra*, i.e. polyhedra where all facets can be described by rational inequalities of bit complexity at most φ, called the *facet complexity*. Bounded facet complexity guarantees bounded *vertex complexity*, i.e. all extreme point solutions are rational numbers of bounded encoding length. The seminal work of Grötschel, Lovász, and Schrijver, summarized in the book *Geometric Algorithms and Combinatorial Optimization* [17], provided a polynomial-time algorithm for optimizing over rational polyhedra given by a strong separation oracles.

They use a black-box argument that requires a subroutine to find either a feasible point or a small-volume enclosing ellipsoid for the a convex set. Such a subroutine can be implemented using the ellipsoid method. Exploiting that a small volume ellipsoid must be sufficiently thin in a certain direction, they use *simultaneous Diophantine approximation* to identify an affine subspace containing the feasible region.

For the sake of simplicity, let us discuss the problem of finding dual optimal solutions under the following simplifying assumption:

(1.4) *The encoding sizes of the vectors returned by the strong separation oracle are polynomially bounded by the facet complexity φ.*

Under this assumption, one can find a *optimal dual solutions with oracle inequalities* [17, Lemma 6.5.15]. This assumption is not without loss of generality; in the full version we discuss this and different concepts of dual solutions, and explain how one can still recover the results of [17] from our approach in case (1.4) does not hold.]

For finding the optimal dual solutions, [17] needs several runs of the (primal) ellipsoid method, including the final one where the variable set corresponds to a large (albeit still polynomially bounded) set of inequalities. The running time depends on a higher power of φ.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Under the same assumption (1.4), Theorem 1.3 enables a much simpler and more efficient algorithm. Even though Theorem 1.3 requires a polyhedral separation oracle, in the full version we show that one can convert a strong separation oracle to a polyhedral separation oracle by rounding the right hand sides of the inequalities using the continued fractions method, and prove that $\delta_M \geq 1/2^{O(n^3\varphi)}$ for the associated conic system. Compared to the general framework in [17], this method has the following advantages in the setting of (1.4).

- We can identify lower dimensional subspaces without simultaneous Diophantine approximation. The only 'number theoretic' subroutine we use is the continued fractions method; otherwise, we rely on the purely geometric measure δ_M. Our algorithm can recurse by setting some inequalities returned by the oracle to equality.
- We recover dual certificates along with the primal solutions, without the need of solving a second, much larger linear program. The running time in [17] on a higher degree polynomial of φ; our running time depends linearly on $\log(1/\delta_M)$.
- The algorithms in [17] require accuracy depending on φ from the approximate subroutines. The running time of our algorithm depends on the condition number δ_M that can be drastically better than the lower bound implied by φ. In a sense, we work directly with the condition numbers implicit in [17] and lower bounded using the facet complexity.

Dual optimal solutions for LPs in the oracle model can be important for applications in combinatorial optimization. For example, the recent paper Svensson et al. [33] on the asymmetric travelling salesman problem crucially uses an optimal dual solution to the Held–Karp relaxation; prior to our work, this could only be obtained using the method in [17]. For this relaxation, one can naturally obtain a polyhedral separation oracle that returns a violated degree constraint or blossom inequality. Therefore, we do not even need to round the right hand sides. Our algorithm proceeds directly by identifying tight inequalities in an optimal solution, and terminates with exact primal and dual optimal solutions in strongly polynomial time.

Implementing the approximate conic oracle Both [17] and Theorem 1.3 are black-box methods. However, [17] requires a seemingly weaker 'primal-only' subroutine, whereas Theorem 1.3 requires an approximate dual certificate. We next explain that this difference is illusory: a ε-approximate conic Farkas certificates can be naturally extracted from the ellipsoid method as well as other convex feasibility algorithms.

The algorithm of Theorem 1.2 is based on the JLSW [21] cutting plane method. In the full version of the paper, we present a general technique to extract ε-approximate conic Farkas certificates from various methods to solve convex feasibility problems; we only list the oracle complexities here.

- The ellipsoid method [17] can be modified to provide an algorithm for Approx-Conic-Dual with $\mathcal{T}_o(n,\varepsilon) = O(n^2\log(1/\varepsilon))$ oracle calls.
- Volumetric cutting plane methods [21, 26, 36] can be used to provide an algorithm for Approx-Conic-Dual with $\mathcal{T}_o(n,\varepsilon) = O(n\log(1/\varepsilon))$ oracle calls. We note that [26] contains an almost explicit statement that gives the bounds $\mathcal{T}_o(n,\varepsilon) = O(n\log(n/\varepsilon))$ and $\mathcal{T}_a(n,\varepsilon) = O(n^3\log^{O(1)}\log(n/\varepsilon))$, see Theorem 5.1.
- The geometric rescaling algorithm [12, 19] can be modified to provide an algorithm for Approx-Conic-Dual with $\mathcal{T}_o(n,\varepsilon) = O(n^3\log(1/\varepsilon))$ oracle calls.

A common feature of the above methods applied to the intersection $K \cap \mathbb{B}^n(1)$ of the cone and the unit ball is that they can find a "ε-thin direction", that is, an oracle inequality m_t such that $m_t^\top x \leq \varepsilon\|m_t\| \cdot \|x\|$ for every $x \in K$. By convex duality, there must exist a dual certificate of this bound using inequalities returned by the oracle during the course of the algorithm; such certificate would provide an ε-approximate Farkas certificate.

One aspect that is ostensibly absent from the original ellipsoid method or from Vaidya's method is duality: at first sight, they appear to be "primal" methods only, where infeasibility is concluded by a volumetric argument, relying on the assumption that the feasible region has a sufficiently large volume, without returning a Farkas certificate of infeasibility. Furthermore, in the ellipsoid method no certificate is maintained of the fact the feasible region is contained within the current ellipsoid.

In a remarkable paper, Burrell and Todd [7] showed that, in the context of the ellipsoid method, both these shortcomings are illusory. They introduced a new view of the ellipsoid method in terms of what we will refer to in Section 5 as *'certified concave quadratic forms'*. The ellipsoid E produced by the algorithm at any iteration is maintained in the form $E = \{x \in \mathbb{R}^n : q(x) \geq 0\}$, where the strictly concave quadratic form $q(x)$ is built from the defining constraints of P and the initial ball constraint $\|x\| \leq r$ in a way that immediately verifies the containments $P \subseteq E$. Furthermore Burrell and Todd showed that, from such a representation, one can construct dual certificates for any bound that holds for a linear function over the elliposid E.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We extend Burrell and Todd's framework beyond the ellipsoid method. A further illustration is given on the geometric rescaling algorithm, by showing how certified quadratic forms can be maintained during execution of the algorithm. For volumetric cutting plane methods, there is no additional overhead in maintaining the quadratic forms. We show that the final output of the algorithm can be converted to a certified concave quadratic form.

Nemirovski, Onn, and Rothblum [28] extended the work of Burrell and Todd by giving a very general certification procedure for the oracle model. Consider any convex minimization problem given by oracle access, returning separators for infeasible points and subgradients of the objective function for feasible points, and consider an algorithm (such as variants of cutting plane methods), that can find a feasible solution with objective value within $\varepsilon > 0$ from the optimum value. Under mild assumptions, they show that it is possible to construct a dual certificate of the approximate optimality of the solution as an appropriate conic combination of the separators and subgradients obtained during the algorithm. Any such certification procedure should be applicable to implement Approx-Conic-Dual.

1.2 Explicit linear programs and connections with circuit imbalance

Let us now consider the implications of our results on explicitly given LP, and compare the running time achieved by our algorithm with the currently fastest algorithms for this setting. We consider linear programs of the form

$$\min c^\top x, \quad Ax = b, x \geq 0\,, \tag{1.5}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. The dual can be written as

$$\max b^\top y, \quad A^\top y \leq c\,. \tag{1.6}$$

One can obtain an $O(mn)$ time polyhedral separation oracle for this problem by computing the vector $A^\top y$.

Using the JLSW algorithm [21] to implement the approximate oracle for (1.6), Theorem 1.1 yields a complexity bound $O\left(nm^3 \log(n/\delta_{A^\top}) + m^5 \log\log(n/\delta_{A^\top})\right)$ for the feasibility of (1.5) (that is, the dual of (1.6)), and $O\left(nm^3 \log(n/\delta_{(A^\top,c)+(\mathbf{0},1)}) + m^5 \log\log(n/\delta_{(A^\top,c)+(\mathbf{0},1)})\right)$ for optimization.

We compare this with recent work on explicitly given linear programs [10, 11]. For the comparison, we need to introduce the following condition number. For a linear space $W \subseteq \mathbb{R}^n$, the set of *elementary vectors* $\mathcal{E}(W)$ is the set of support minimal nonzero vectors in W; the support of elementary vectors corresponds to the set of circuits in the associated linear matroid. The *circuit imbalance measure* κ_W is defined as the maximum ratio $|g_j/g_i|$ over all $g \in \mathcal{E}(W)$ and all $i, j \in \text{supp}(g)$. For a matrix $A \in \mathbb{R}^{m\times n}$, we let κ_A denote κ_W for $W = \ker(A)$. In particular, $\kappa_A = 1$ for totally unimodular matrices.

Dadush et al. [11], strengthening Tardos's result [34] on combinatorial linear programs, gave an algorithm with running time $\text{poly}(n, m, \log(\kappa_A + n))$ for solving linear programs of the form (1.5).

The condition number κ_A is within a factor $1/n$ from the Dikin–Stewart–Todd condition number $\bar{\chi}_A$ used in [39], see [10, 11]. Hence, $\log(\kappa_A + n) = \Theta(\log(\bar{\chi}_A + n))$.

The algorithm in [11] is of a black-box nature: for linear optimization, it requires $O(nm)$ calls to an approximate linear programming solver with accuracy $\varepsilon = 1/(n\kappa_A)^{O(1)}$. For the linear feasibility problem $Ax = b$, $x \geq 0$, $O(m)$ calls suffice. Combined with the solver of van den Brand [37] the running time is $O\left(mn^\omega \log^2(n)\log(\kappa_A + n)\right)$, and combined with the solver of van den Brand et al. [38], it is $O\left((nm^2 + m^4)\log^{O(1)}(n)\log(\kappa_A + n) + m^5 \log\log(\kappa_A + n)\right)$.

The condition numbers $1/\delta$ and κ are related as follow: for a matrix of the form $A = (I_m|A')$, we have $\log(n/\delta_{A^\top}) = \Theta(\log(\kappa_A + n))$ (see full version). We can therefore use our conic validity algorithm in Theorem 1.3 to find a feasible solution to (1.5) in $\text{poly}(n, m, \log(\kappa_A + n))$ time. In particular, using the JLSW algorithm [21] to implement the approximate oracle gives us a running time of $O(nm^3 \log(n + \kappa_A) + m^5 \log\log(n + \kappa_A))$ for feasibility of (1.5).

At a high level, the feasibility algorithm in [11] and our conic validity algorithm both use approximate solutions to $Ax \approx b$, $x \geq 0$ to reduce the problem size, and project out variables with high x_i values. The main difference is that [11] requires a stronger approximate oracle that enables a more efficient 'pullback' of a Farkas certicate in case of infeasibility. Our algorithm has an additional term $O(m^5 \log\log(n + \kappa_A))$ compared to $O(\min\{m^5, mn^\omega\}\log\log(\kappa_A + n))$ in [11]. We note that our method cannot reproduce the main result of [11] of a $\text{poly}(n, m, \log(\kappa_A + n))$ algorithm for linear optimization: our running time depends on $\delta_{(A^\top,c)+(\mathbf{0},1)}$. On the other hand, [11] heavily uses that the system is explicitly given, while our method extends to the oracle setting.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

1.3 Overview of techniques

Adaptive bound on δ_M In general, computing δ_M may be difficult. Nevertheless, our algorithms are all "oblivious" to the value of δ_M: we do not need to know this parameter to terminate within the claimed number of oracle calls. Let us start with the optimistic estimate $\hat{\delta} = 1/n$, and run the algorithm with this value. The precision ε required from Approx-Conic-Dual will depend on our adaptive estimate of $\hat{\delta}$. The algorithm may succeed even if $\hat{\delta} > \delta_M$. In case the algorithm fails to deliver the required conclusions, it will be able "certify" such failure, by returning a set of linearly independent rows $\{m_i : i \in J\}$ along with coefficients $\lambda_i \in \mathbb{R}^J$ such that $\sum_{i\in J} \lambda_i m_i < \hat{\delta} \max_{i\in J} \lambda_i \|m_i\|$, thus showing $\hat{\delta} > \delta_M$. We can then update our guess to the bound implied by these vectors or to $\hat{\delta}^2$, whichever is smaller, and simply restart the algorithm.

Hence, if our algorithm has not succeeded for the very first time, we will have the guarantee that $\hat{\delta} \geq \delta_M^2$ for all subsequent trials. Assuming the running time of each trial is bounded as $\mathrm{poly}(n, \log(1/\hat{\delta}))$, the overall running time of all trials will be dominated by the running time of the final, successful trial.

Strong conic feasibility Consider the strong conic feasibility algorithm for a cone K and constraint $m_1^\top x > 0$. We call the subroutine Approx-Conic-Feasible for the cone $K_1 = K \cap \{x \in \mathbb{R}^n : m_1^\top x > 0\}$ and $\varepsilon = \hat{\delta}/O(n)$. The algorithm terminates if a feasible solution is found. Otherwise, an ε-approximate conic Farkas certificate $\lambda \in \mathbb{R}^m_+$ is returned.

Assume first that such vector λ satisfies $M^\top \lambda = 0$. The certificate shows that $K \subseteq \ker(M_J)$, where $J = \mathrm{supp}(\lambda)$. In particular, if $\lambda_1 > 0$, then this shows that $m_1^\top x = 0$ for all $x \in K$, and the algorithm stops. Otherwise, the algorithm recurses on the lower dimensional space $\ker(M_J)$. In case $M^\top \lambda \neq 0$, we use a Carathéodory-style subroutine which either succeeds in finding another nonzero $\lambda' \in \mathbb{R}^m_+$, with linearly independent support such that $M^\top \lambda' = 0$, in which case we proceed as above, or fails in finding such a vector, in which case it will output a certificate that our adaptive estimate $\hat{\delta}$ was incorrect (that is, $\hat{\delta} > \delta_M$).

Conic validity and conic minimum-ratio The algorithms for conic validity and conic minimum-ratio are more involved, due to the fact that the number of iterations should only depend on n and δ_M, but not on c and d. In particular, the simple strategy adopted for strong conic validity does not work, as it would require a level of precision ε dependent on c and d.

We briefly outline the main idea for the conic validity algorithm; the conic-minimum ratio algorithm is a further extension of this idea. The conic validity algorithm for the cone K and vector c, calls the subroutine Approx-Conic-Feasible for the cone $K_c = K \cap \{x \in \mathbb{R}^n : c^\top x > 0\}$ and $\varepsilon = \hat{\delta}^2/O(n^2)$. We terminate in case a feasible solution is found. Otherwise, we consider the ε-approximate conic Farkas certificate $(\lambda, \tau) \in \mathbb{R}^m_+ \times \mathbb{R}_+$ returned, where τ is the multiplier for the inequality $-c^\top x < 0$. If τ is sufficiently small, then we can recurse as in the feasibility algorithm. If τ is large, then—assuming $\hat{\delta} \leq \delta_M$—the inequalities of $Mx \geq 0$ corresponding to suitably large entries of λ have to be satisfied at equality for every $x \in K$. We recurse on the lower dimensional space and continue. At any step we will therefore have a set $F \subseteq [m]$ with the guarantee that $K_c \cap \ker(M_F) \neq \emptyset$, assuming $\hat{\delta} \leq \delta_M$. However, observe that, if our estimate $\hat{\delta}$ was actually not correct, it is possible that we recursed on the wrong subspace, that is, $K_c \cap \ker(M_F) = \emptyset$. (This is in contrast with the feasibility algorithm described above, where $K_1 \subseteq \ker(M_F)$ is always guaranteed when recursing.) The algorithm recurses until it either finds $x \in K \cap \ker(M_F)$ such that $c^\top x > 0$, in which case we stop with the feasible solution x, or a dual certificate $\tilde{\lambda}$ of the fact that $c^\top x \leq 0$ for all $x \in K \cap \ker(M_F)$. The main technical tool at this stage is an algorithm, described in Lemma 4.3, is a *pullback* subroutine. Starting from $\tilde{\lambda}$, it either produces a dual certificate $\lambda \in \mathbb{R}^m_+$ such that $M^\top \lambda = -c$, in which case we stop, or detects a failure for $\hat{\delta}$, in which case we update our bound $\hat{\delta}_M$ and continue.

1.4 Further related results

In recent work, Jiang [20] improved the complexity bounds of minimizing convex functions over integers. This is achieved by a more direct application of lattice basis reduction than in [17]. However, this does not seem to lead to an improvement for rational polyhedra in the bounded facet complexity model.

The Burrell–Todd representation [7] was also used recently by Lamperski, Freund, and Todd [24] to developed an "oblivious ellipsoid algorithm" that terminates in finite time, assuming P is explicitly given by inequalities, and that P is either full-dimensional or empty. In contrast, our result is applicable also for degenerate systems. We also note that whereas [24] use a modification of the standard ellipsoid method, our approach uses the standard method in a black-box manner.

Geometric rescaling is a more recent class of polynomial-time linear programming algorithms: the common theme of such algorithms is to boost simple iterative algorithms by adaptively changing the scalar product. The first such algorithms were given by Betke [3] and by Dunagan and Vempala [13], and a number of papers have since appear on the subject. We refer the reader to [12] for an overview of such results. Whereas most of these algorithms work only under the assumption that the constraints defining the cone are explicitly given as

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

part of the input, some variants, including those described in [12, 19], can be naturally extended to the oracle setting. In the full version of the paper, we implement the approximate conic oracle for these variants.

Theorem 1.3 also gives an answer to a question raised in [12] on finding a "primal-dual" geometric rescaling algorithm for the conic maximum support problem that does not depend on a priori bounds on the condition numbers. Such an algorithm was also recently obtained for explicitly given systems by Pena and Soheili [30]. Their algorithm runs the rescaling algorithms in parallel on the primal and dual problems, with increasing estimates on a certain condition number.

Organization of the paper The paper is organized as follows. Section 2 introduces some basic notation and concepts, provides the proof of Theorem 1.1, and proves some fundamental facts about the condition measure δ_M. Sections 3 and 4 describe the algorithms for the strong conic feasibility and conic validity problems, along with their analyses. The algorithm for minimum conic ratio, while more involved, shares several of the ideas used for strong conic validity, and will be presented in the full version of this paper. Sections 5.1 and 5.2 describe our general approach to finding approximate Farkas certificates from certified quadratic forms. In the full version, we will use these results to provide implementations of the oracle Approx-Conic-Dual based on different methods (ellipsoid, volumetric cutting plane, and geometric rescaling). All omitted proofs can be found in the full version.

2 Preliminaries

For a natural number $k < m$, let $[m] = \{1, 2, \ldots, m\}$, $[k, m] = \{k, k+1, \ldots, m\}$. For any number $\alpha \in \mathbb{R}$, we let $\alpha^+ = \max\{\alpha, 0\}$ and $\alpha^- = \max\{-\alpha, 0\}$. For a vector $x \in \mathbb{R}^n$, x^+ and x^- in $\mathbb{R}^n$ are defined by $(x^+)_i = x_i$, $(x^-)_i = x_i^-$, $i \in [i]$. Thus, $x = x^+ - x^-$. Let $\vec{e}_j$ denote the jth unit vector in $\mathbb{R}^n$. For a set of vectors $\{v_j : j \in J\} \subseteq \mathbb{R}^n$, we let $\operatorname{span}(v_j : j \in J) \subseteq \mathbb{R}^n$ the linear subspace they span; for a matrix $B \subseteq \mathbb{R}^{n\times m}$, let $\operatorname{span}(B) \subseteq \mathbb{R}^n$ denote the linear subspace spanned by the columns of B.

For any matrix $H \in \mathbb{R}^{k\times n}$ and every $J \subseteq [k]$, we denote by H_J the submatrix of H defined by the rows indexed by J, and similarly, for $v \in \mathbb{R}^k$, v_J defined the restriction of v to the entries indexed by J.

$K \subseteq \mathbb{R}^n$ is a *cone* if K is convex and K is closed under positive scalings, that is, $x \in K \Rightarrow \lambda x \in K, \forall \lambda > 0$. For a set of vectors $v_1, \ldots, v_k \in \mathbb{R}^n$, we let $\operatorname{cone}(v_1, \ldots, v_k) := \{\sum_{i=1}^k \lambda_i v_i : \lambda_1, \ldots, \lambda_k \geq 0\}$ denote the *closed* cone induced by $v_1, \ldots, v_k$.

For a convex set $C \subseteq \mathbb{R}^n$, we say that $F \subseteq C$ is a *face* of C if F is convex and if for all $x, y \in C$, we have that $\lambda x + (1-\lambda)y \in F, \lambda \in (0,1)$, implies that $x, y \in F$. The *lineality space* of C is the largest linear subspace W such that $C + W = C$. We say that closed convex set C is *pointed* if its lineality space is $W = \{0\}$. For a closed pointed cone K, the set of 1-dimensional faces of K are called the *extreme rays* of K. Slightly abusing notation, we will also say that $v \in K \setminus \{0\}$ is an extreme ray of K if $\mathbb{R}_+ v$ is a 1-dimensional face of K.

Given $p \in \mathbb{R}^n$ and $r > 0$, we denote by $\mathbb{B}^n(p, r)$ the ball of radius r in $\mathbb{R}^n$ centered at p. We use the notation $\mathbb{B}^n(r)$ for $\mathbb{B}^n(0, r)$. We denote by $\mathbb{S}^n_{++}$ and $\mathbb{S}^n_+$ the sets of symmetric $n \times n$ positive definite and positive semi-definite matrices, respectively. For $P, Q \in \mathbb{S}^n_+$, we use $P \preceq Q$ if $Q - P \in \mathbb{S}^n_+$. For $Q \in \mathbb{S}^n_{++}$ and a vector $v \in \mathbb{R}^n$, we let $\|v\|_Q \overset{\text{def}}{=} \sqrt{v^\top Q v}$; this defines a norm over $\mathbb{R}^n$. We use $\|\cdot\|_1$ for the ℓ_1-norm and $\|\cdot\|_2$ for the Euclidean norm. When there is no risk of confusion we simply write $\|\cdot\|$ for $\|\cdot\|_2$.

We use the real model of computation, allowing basic arithmetic operations +, −, ×, / and comparisons. We avoid using square roots exactly: instead of unit norm vectors, we sometime assume that $\|v\| \in [1, 2]$ for certain vectors. The results can be easily adapted to the Turing model; however, this requires rounding steps. The black box algorithms in Sections 3–4 only use simple linear algebra subroutines that can be implemented in the Turing model without any modification.

The following simple claim will be needed for running time estimations when using an adaptive bound on the condition number δ_M.

Lemma 2.1. *Let $1/n = \delta_1 > \delta_2 > \ldots > \delta_t$ and $\delta > 0$ be real numbers such that $\delta_{i+1} < \delta_i^2$ for $i \in [t-1]$, $\delta_t > \delta^2$, and $\nu \geq 1$. Then,*

$$\sum_{i=1}^{t} \log^\nu(n/\delta_i) = O(1) \cdot \log^\nu(n/\delta)\,.$$

Separation oracle variants For a convex set $K \subseteq \mathbb{R}^n$, a *strong separation oracle* takes as input a point $\bar{x} \in \mathbb{R}^n$, and either either returns the answer $\bar{x} \in K$, or a nonzero vector $a \in \mathbb{R}^n$, such that $a^\top x < a^\top \bar{x}$ for every $x \in K$. This is the standard separation oracle model used for the ellipsoid and other cutting plane methods. The notion of conic separation oracle required for Approx-Conic-Dual oracle is, as discussed in the Introduction, identical to the strong separation oracle if the cone K it defines is non-empty.

Recall that our main results Theorem 1.1 and Theorem 1.3, make stronger oracle assumptions. Namely, we assume that a polyhedron P, is defined by a *polyhedral separation oracle*: if $\bar{x} \notin P$, the oracle returns

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

an inequality $a^\top x \leq \beta$ violated by $\bar{x}$, where the set of all inequalities returned by the oracle for all possible choices of $\bar{x} \notin P$ is finite. We will often write that $P = \{x : Ax \leq b\}$ is defined by a polyhedral separation oracle to mean that $Ax \leq b$ comprises all such possible inequalities, with the understanding that $Ax \leq b$ is not explicitly given, but we have access to it via the oracle.

2.1 Reducing LP to the conic setting: proof of Theorem 1.1 We now give the proof of Theorem 1.1 using Theorems 1.2 and 1.3. Consider a polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ for some $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and its recession cone $\mathrm{rec}(P) = \{x \in \mathbb{R}^n : -Ax \geq 0\}$, both of which are given by a polyhedral separation oracle.

We derive parts *(i), (ii), (iii)* of Theorem 1.3 using the standard homogenization of P into $\mathbb{R}^{n+1}$. Namely, we examine

$$K = \{(x \mid t) \in \mathbb{R}^{n+1} : tb - Ax \geq 0, t \geq 0\} := \{(x \mid t) \in \mathbb{R}^{n+1} : M(x \mid t) \geq 0\}, \tag{2.7}$$

where $M \in \mathbb{R}^{(m+1)\times(n+1)}$ is as in (1.1). Note that $(x \mid 0) \in K \Leftrightarrow x \in \mathrm{rec}(P)$ and that $(x \mid t) \in K, t > 0 \Leftrightarrow x/t \in P$.[3]

(i) Primal feasibility We must compute a solution to $Ax \leq b$ or a Farkas certicate of infeasibility $A^\top\lambda = 0, b^\top\lambda < 0, \lambda \geq 0$. We reduce to solving strong conic feasibility using Theorem 1.3 on K as above with the constraint $t > 0$ corresponding to m_1. For this purpose, we require a polyhedral separator for $K_1 := \{(x \mid t) \in K : t > 0\}$, which can be derived directly from the polyhedral separation oracle for P. Namely, given $(x' \mid t')$, if $t' \leq 0$, we return the separator $t > 0$, and if $t > 0$, we call the separator for P on x'/t'. If x'/t' violates $a_i x \leq b_i$ for P, we return $(-a_i \mid b_i)$ for K'. From here, if Theorem 1.3 returns $(x \mid t) \in K_1$, we return $x/t \in P$, and if it returns $\lambda \in \mathbb{R}^m_+$ satisfying $\lambda^\top(-A, b) + 1 \cdot (0, 1) = 0$, we return λ as the Farkas certificate.

(ii) Dual feasibility We must compute a solution to $A^\top\lambda = c$, $\lambda \geq 0$, or find a solution to $Ax \leq 0$, $c^\top x > 0$. For this purpose, we reduce to the conic validity problem using Theorem 1.3 on the cone $K = \mathrm{rec}(P) := \{x \in \mathbb{R}^n : -Ax \geq 0\}$ and the vector $\bar{c} := -c$. This requires a polyhedral conic separation oracle for $K_{\bar{c}} = \{x \in \mathbb{R}^n : -Ax \geq 0, c^\top x > 0\}$. This is direct to implement since we have access to a polyhedral separation oracle for $\mathrm{rec}(P)$ and c is known to us. Since the conic validity problem is a direct restatement of the dual feasibility, the correctness of the reduction is evident.

(iii) Optimization Assuming both $Ax \leq b$ and $A^\top\lambda = c, \lambda \geq 0$ are feasible, we must find an optimal primal-dual pair x^*, λ^* satisfying complementary slackness, namely $(\lambda^*)^\top(b - Ax^*) = 0$. We reduce this to the conic minimum-ratio problem on K given by $\min\{-c^\top x/t : (x \mid t) \in K, t > 0\}$ using Theorem 1.3. Note that this problem can be rewritten as $\min\{\bar{c}^\top(x \mid t)/d^\top(x \mid t) : (x \mid t) \in K, d^\top(x \mid t) > 0\}$, where $\bar{c} = (-c \mid 0)$ and $d = (0 \mid 1)$.

For this purpose, we first require that $d = (0 \mid 1)$ be given as a conic combination of the original constraints of K, which trivially holds since $(0 \mid 1)$ induces an original constraint itself; hence $I = \{1\}$. We also require polyhedral separators for $K_{-d} = \{(x \mid t) \in K : d^\top x > 0\} = \{(x \mid t) \in K : t > 0\}$ and for $K_I^= = \{(x \mid t) \in K : d^\top x = 0\} = \{(x \mid 0) \in K\} := (\mathrm{rec}(P) \mid 0)$. As explained in the previous paragraphs, these separators can be directly constructed from the corresponding polyhedral separators for P and $\mathrm{rec}(P)$. Furthermore, we recall that feasibility of $Ax \leq b$ is equivalent to $K_{-d} \neq \emptyset$ (i.e., $d^\top(x \mid t) = 0$ is not valid for K) and feasibility of $A^\top\lambda = c, \lambda \geq 0$ is equivalent to $-c^\top x = \bar{c}^\top(x \mid t) \geq 0$ being a valid inequality for $K_I^=$.

Given the above, the conic minimum-ratio solve must output $\gamma^* \in \mathbb{R}$, $(x^* \mid t^*) \in K_d$, $\lambda^* \in \mathbb{R}^m_+, \beta^* \geq 0$ such that $\gamma^* = -\bar{c}^\top(x^* \mid t^*)/(d^\top(x^* \mid t^*)) = c^\top x/t$ and $(\lambda^*)^\top(-A, b) + \beta^*(0, 1) = \bar{c}^\top + \gamma^* d^\top = (-c, \gamma^*)$. We claim that $x^*/t^*, \lambda^*$ are the desired optimal primal-dual pair. To begin, we note that the inclusion $x^*/t^* \in P$ is direct since $t^* > 0$. Furthermore, by the guarantees of the output we have that

$$0 = (\bar{c} + \gamma^* d)^\top(x^* \mid t^*) = ((\lambda^*)^\top(-A, b) + \beta^*(0, 1))(x^* \mid t^*) = (\lambda^*)^\top(t^* b - Ax^*) + \beta^* t^* \geq 0 + 0 = 0.$$

Since $t^* > 0$, the above implies that $\beta^* = 0$ and that $(\lambda^*)^\top(t^* b - Ax^*) = 0$. Since $\beta^* = 0$, we see that λ^* is a valid dual solution. Finally, complementary slackness follows from $(\lambda^*)^\top(t^* b - Ax^*)$ after dividing by t^*.

For all three problems above, the desired running times now follow directly by combining Theorem 1.2 with the corresponding part of Theorem 1.3.

[3] We cannot directly build a polyhedral conic separation oracle for K given our assumptions. In particular, for an input $(x, 0)$ to the oracle, if $(x, 0) \notin K$, we would need to return $(-a_i \mid b_i)$ such that $-a_i^\top x < 0$. Our polyhedral separator for $\mathrm{rec}(P)$ would give us access to a_i but not to b_i, noting that the inequality $(-a_i \mid 0)^\top(x \mid t) \geq 0$ is not necessarily valid for K. We will be able to circumvent this issue however, as the problems we wish to solve will only require polyhedral conic seperation oracles for sub-cones of K, that we will be able to build directly.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

2.2 Properties of the δ-measure We start by showing that the δ-measure introduced in Definition 1 is equivalent to the definition of the "δ-distance property" studied in [6, 9, 14], and that it is positive if and only if V is finite.

Lemma 2.2. *For a set of vectors $V \subseteq \mathbb{R}^n$, δ_V is the largest value such that, for every $W \subseteq V$ and every $v \in V \setminus \operatorname{span}(W)$, the Euclidean distance between v and $\operatorname{span}(W)$ is at least $\delta_V \|v\|$. Further, $\delta_V > 0$ if and only if $|\{v/\|v\| : v \in V\}|$ is finite.*

The following characterization can be shown with a similar argument.

Lemma 2.3. ([6, Lemma 5(i)]) *Consider a matrix $M \in \mathbb{R}^{m\times n}$ such that all rows $m_i^\top$ have norm one. For a matrix $B \in \mathbb{R}^{m\times m}$, let $\gamma(B)$ denote the maximum column norm of B. Then,*

$$\frac{1}{\delta_M} = \max\left\{\gamma(N^{-1}) : N \textit{ is an } m \times m \textit{ submatrix of } M\right\}$$

The following is the key property of δ_M in the conic setting. Namely, it gives a lower bound in terms of δ_M, on the angles extreme rays of $Mx \geq 0$ can form with constraints that they are not incident to.

Lemma 2.4. *Let $K = \{x \in \mathbb{R}^n : Mx \geq 0\}$ be a closed polyhedral cone with $M \in \mathbb{R}^{m\times n}$. Then, for any extreme ray v of $K \cap \operatorname{span}(M^\top)$ and $i \in [m]$, we have that either $m_i^\top v = 0$ or $m_i^\top v \geq \delta_M \|m_i\| \cdot \|v\|$, $\forall i \in [m]$.*

3 The strong conic feasibility algorithm

In this section, we prove the part of Theorem 1.3 for the strong conic feasiblity problem. We assume a polyhedral conic separation oracle is available for

$$K_1 = \{x \in \mathbb{R}^n : Mx \geq 0, m_1^\top x > 0\},$$

and that the subroutine Approx-Conic-Dual for K_1 is provided as in Section 1.1, requiring $\mathcal{T}_o(n,\varepsilon)$ oracle calls, $\mathcal{T}_a(n,\varepsilon)$ arithmetic operations, and returning an ε-approximate conic Farkas certificate of size at most $\tau(n)$.

The next lemma captures the key recursive step:

Lemma 3.1. *Let $K = \{x \in \mathbb{R}^n : Mx \geq 0\}$ for $M \in \mathbb{R}^{m\times n}$, given by a conic separation oracle, and let $m_1^\top$ be the first row. There exists an oracle polynomial-time algorithm using $O(\mathcal{T}_o(n, \delta_M/(2n)))$ oracle calls and $O(\mathcal{T}_a(n, \delta_M/(2n)) + (n^3 + n\tau(n)^2)\log\log(1/\delta_M))$ that either finds an $x \in K$ with $m_1^\top x > 0$, or $\lambda \in \mathbb{R}^m_+$ that is a minimal support solution to $M^\top\lambda = 0$.*

In Section 3.1, we show how the strong conic feasibility algorithm can be obtained by at most n calls to this subroutine. The proof of Lemma 3.1 relies on the decomposition stated in the next lemma. This is essentially a careful reading of the proof of Carathéodory's theorem. It is a consequence of [11, Lemma 4.1].

Lemma 3.2. *There exists an $O(n|J|^2 + n^2|J|)$ time algorithm that, given vectors $\{v_j : j \in J\}$, $\lambda \in \mathbb{R}^J_+$, and $c \in \mathbb{R}^J$ such that $c^\top\lambda > 0$, outputs one of the following.*

(i) A vector $\bar\lambda \in \mathbb{R}^J_+$ such that $\sum_{j\in J}\bar\lambda_j v_j = \sum_{j\in J}\lambda_j v_j$, $c^\top\bar\lambda \geq c^\top\lambda$, and the vectors $\{v_j : \bar\lambda_j > 0\}$ are linearly independent.

(ii) A nonzero vector $\mu \in \mathbb{R}^J_+$ which is a support-minimal solution to $\sum_{j\in J}\mu v_j = 0$, and such that $c^\top\mu \geq 0$.

Proof. [Proof of Lemma 3.1] We maintain an estimate $\hat\delta$ on δ_M, initializing $\hat\delta := 1/n$. A nonzero vector $\lambda \in \mathbb{R}^m$ is a *failure for* $\hat\delta$ if $\{m_j : j \in \operatorname{supp}(\lambda)\}$ are linearly independent and $\varphi := \|M^\top\lambda\|/(\max_{j\in[m]}\lambda_j\|m_j\|) < \hat\delta$, proving $\delta_M \leq \varphi$. Whenever we detect a failure, we update $\hat\delta := \min\{\hat\delta^2, \varphi\}$.

We call the subroutine Approx-Conic-Dual for K_1 for $\varepsilon := \hat\delta/(2n)$. This requires $\mathcal{T}_o(n, \hat\delta/(2n))$ oracle calls and $\mathcal{T}_a(n, \hat\delta/(2n))$ operations. Either we obtain an $x \in K$ with $m_1^\top x > 0$, or $\lambda \in \mathbb{R}^m_+$ such that $\sum_{j=1}^m \lambda_j\|m_j\| \geq 1$ and $\|M^\top\lambda\| \leq \hat\delta/(2n)$; let $J = \operatorname{supp}(\lambda)$. If $M^\top\lambda = 0$ then we can readily return the vector λ.

If $M^\top\lambda \neq 0$, then we apply Lemma 3.2 to λ, the vectors $\{m_j : j \in J\}$, and to the vector $c \in \mathbb{R}^m$ defined by $c_j = \|m_j\|$, $j \in [k]$. Recall that $|J| \leq \tau(n)$, hence this step requires $O(n^2|J| + n|J|^2) = O(n^3 + n\tau(n)^2)$ arithmetic operations. If outcome *(i)* occurs, then we obtain $\bar\lambda \in \mathbb{R}^m_+$ with $\operatorname{supp}(\bar\lambda) \subseteq J$, $\sum_{j=1}^m \bar\lambda_j\|m_j\| \geq 1$ such that $M^\top\bar\lambda = M^\top\lambda$, and the rows of M in the support of $\bar\lambda$ are linearly independent. We claim that $\bar\lambda$ is a failure for $\hat\delta$. Suppose not. Then we obtain a contradiction

$$\frac{\hat\delta}{2n} \geq \left\|M^\top\lambda\right\| = \left\|M^\top\bar\lambda\right\| \geq \hat\delta \max_{i\in\operatorname{supp}(\bar\lambda)} \bar\lambda_i\|m_i\| \geq \frac{\hat\delta}{n}, \tag{3.8}$$

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where the last inequality follows from $\sum_{j\in J}\bar{\lambda}_j\|m_j\|\geq 1$ and $\mathrm{supp}(\bar{\lambda})\leq n$. In this case we update $\hat{\delta}$ and ε accordingly, and call $\varepsilon=\hat{\delta}/(2n)$ for the new value of ε. If outcome *(ii)* occurs, we obtain a nonzero $\mu\in\mathbb{R}^m_+$, $\mathrm{supp}(\mu)\subseteq J$ such that $M^\top\mu=0$ and μ is support minimal; we can return μ as the output. The running time bound follows using Lemma 2.1, using the choice of the estimates $\hat{\delta}$. □

3.1 The recursive algorithm We now describe the overall strong conic feasibility algorithm, with the running time bound stated in Theorem 1.3. This can be achieved by making at most n calls to the algorithm in Lemma 3.1. We gradually identify a subset $F\subseteq[m]$ and find coefficients $\xi\in\mathbb{R}^m_+$, such that $\mathrm{supp}(\xi)=F$, $M^\top\xi=0$, $|F|\leq 2\mathrm{rk}(M_F)$. This certifies that $M_Fx=0$ for all $x\in K$.

This set is initialized as $F=\emptyset$; after the first call to Lemma 3.1, if the output is a support minimal solution $\lambda\geq 0$ to $M^\top\lambda\geq 0$, then we select $F=\mathrm{supp}(\lambda)$. F will be extended in every iteration; thus, the algorithm terminates by making at most n calls.

The following notation and subsequent Lemmas 3.3 and 3.4 will also be used in later sections, and apply for any $F\subseteq[m]$ (that is, we do not require $K\subseteq\ker(M_F)$). Given an index set $F\subseteq[m]$, let $\Pi^F\in\mathbb{R}^{n\times n}$ be the orthogonal projection matrix onto $\ker(M_F)$. Computing Π^F requires $O(n^3)$ operations. For every vector $v\in\mathbb{R}^n$, let

$$v^F:=\Pi^F v.$$

Let $T_F=\{i\in[m]\,:\,\Pi^F m_i\neq 0\}$ and let $M^F\in\mathbb{R}^{T_F\times n}$ be the matrix with rows $(m_i^F)^\top$. Let

$$K^F=\{x\in\mathbb{R}^n\,:\,M^Fx\geq 0\}\quad\text{and}\quad K_1^F=\{x\in\mathbb{R}^n\,:\,M^Fx\geq 0,(m_1^F)^\top x>0\}\,. \tag{3.9}$$

Lemma 3.3. *For any index set $F\subseteq[m]$ and $\bar{x}\in\mathbb{R}^n$, $\bar{x}\in K^F$ if and only if $\Pi^F\bar{x}\in K$. In particular, $K^F=(K\cap\ker(M_F))+\mathrm{span}(M_F^\top)$. Given a conic separation oracle for K_1, we can implement a conic separation oracle for K_1^F, requiring $O(n^2)$ time for each oracle call.*

Lemma 3.4. *For any $F\subseteq[m]$, we have $\delta_{M^F}\geq\delta_M$.*

Equipped with the above notation, the description of the algorithm follows. We initialize $F=\emptyset$. If at any iteration $m_1^F=0$ then $m_1^\top x=0$ must hold for all $x\in K$; thus, no strong feasible solution exists. We can obtain an infeasibility certificate as follows. Let $\mu\in\mathbb{R}^m$ with $M^\top\mu=0$ such that $\mathrm{supp}(\mu)\subseteq F\cup\{1\}$ and $\mu_1=1$. Then, for sufficiently large $\alpha>0$, $\lambda'=\mu+\alpha\xi$ is a nonnegative vector with $M^\top\lambda'=0$ such that $\lambda_1'=1$.

Each iteration calls the algorithm described in Lemma 3.1 for M^F and K^F with the projected separation oracle as in Lemma 3.3. If the output is a point $x\in K^F$ with $(m_1^F)^\top x>0$, then we return the point $\Pi^Fx\in K$ with $m_1^\top(\Pi^Fx)>0$, which is a solution to the strong feasibility problem.

The other possible output is a nonzero vector $\hat{\lambda}\in\mathbb{R}^{T^F}_+$ that is a support minimal solution to $(M^F)^\top\hat{\lambda}=0$. It follows that $M^\top\hat{\lambda}$ is orthogonal to $\ker(M_F)$, so there exists a $\theta\in\mathbb{R}^F$ such that $M^\top\hat{\lambda}+M_F^\top\theta=0$. Such vector θ can be computed in time $O(n^3)$ by Gaussian elimination, recalling that $|F|\leq 2n$. If we extend $\hat{\lambda}$ and θ to vectors in $\mathbb{R}^m$ by setting to zero the entries outside of their support, choose $\alpha\geq 0$ such that $\theta_i+\alpha\xi_i>0$ for all $i\in F$. Let $J=\mathrm{supp}(\hat{\lambda})$, and define $F'=F\cup J$ and $\xi'=\hat{\lambda}+\theta+\alpha\xi$. We have that $M^\top\xi'=0$, $\mathrm{supp}(\xi')=F'$, $\xi'\geq 0$. Furthermore, $\mathrm{rk}(M_{F'})=\mathrm{rk}(M_F)+\mathrm{rk}(M_J)$ and $|J|\leq\mathrm{rk}(M_J)+1$ since $\hat{\lambda}$ is support minimal, hence $|F'|\leq 2\mathrm{rk}(M_{F'})$, so we can update $F:=F'$, $\xi:=\xi'$.

If none of the recursive calls finds a strongly feasible solution, within at most n iterations we reach $m_1^F=0$ and obtain an infeasibility certificate as above.

Running time analysis Each call to the algorithm in Lemma 3.1 needs $O(\mathcal{T}_o(n,\delta_M/(2n)))$ oracle calls and $O(\mathcal{T}_a(n,\delta_M/(2n))+(n^3+n\tau(n)^2)\log\log(1/\delta_M))$ arithmetic operations, and we call this algorithm at most n times. By Lemma 3.3, each oracle call for K^F requires $O(n^2)$ arithmetic operations. This gives a total number of operations of $O(n^3\mathcal{T}_o(n,\delta_M/(2n)))+O(n\mathcal{T}_a(n,\delta_M/(2n))+(n^4+n^2\tau(n)^2)\log\log(1/\delta_M))$. Whenever we call such an algorithm, we need to update F and ξ, which requires $O(n^3)$ arithmetic operations, as well as Π^F, also in $O(n^3)$ arithmetic operations, for a total of $O(n^4)$ operations.

4 The conic validity algorithm

Next, we prove the part of Theorem 1.3 on conic validity. Recall that in the conic validity problem, the input is a cone $K\subseteq\mathbb{R}^n$ of the form $K=\{x\in\mathbb{R}^n\,:\,Mx\geq 0\}$ with $M\in\mathbb{R}^{m\times n}$, given by a conic separation oracle, and an objective vector $c\in\mathbb{R}^n$, $c\neq 0$. The goal is to either find $y\in\mathbb{R}^m_+$ with $M^\top y=c$, or an $x\in K$ with $c^\top x<0$. Here, we assume a polyhedral conic separation oracle is available for

$$K_c=K\cap\{x\in\mathbb{R}^n\,:\,c^\top x<0\}\,,$$

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

and that a subroutine Approx-Conic-Dual for K_c is provided with running time $\mathcal{T}_o(n,\varepsilon)$ oracle calls and $\mathcal{T}_a(n,\varepsilon)$ arithmetic operations, and returns an ε-approximate conic Farkas certificate comprised of at most $\tau(n)$ oracle separators. The next lemma formulates the main recursive step, analogously to Lemma 3.1. Note that here we use an arbitrary estimate $\hat{\delta} \in (0,1)$ as opposed to the true value δ_M. Outcome (iv) provides a certificate that $\hat{\delta} > \delta_M$.

Lemma 4.1. *Let $K = \{x \in \mathbb{R}^n : Mx \geq 0\}$ for $M \in \mathbb{R}^{m\times n}$, given by a conic separation oracle, let $c \in \mathbb{R}^n$, $c \neq 0$, and let K_c be defined as above. Let $\hat{\delta} \in (0,1)$. There exists an oracle polynomial-time algorithm using $\mathcal{T}_o(n, \hat{\delta}^2/(8n^2))$ oracle calls and $\mathcal{T}_a(n, \hat{\delta}^2/(8n^2)) + O(n^3 + n\tau(n)^2)$ operations that returns one of the following:*

(i) an $x \in K$ with $c^\top x < 0$;

(ii) a nonzero vector $\lambda \in \mathbb{R}^m_+$ which is a support minimal solution to $M^\top \lambda = 0$;

(iii) a vector $\lambda \in \mathbb{R}^m_+$ such that $\{m_j : j \in \operatorname{supp}(\lambda)\}$ are linearly independent, along with a nonempty subset $J \subseteq [m]$ such that for every $j \in J$, $\lambda_j\|m_j\| > \|M^\top\lambda - c\|/\hat{\delta}$.

(iv) a vector $\lambda \in \mathbb{R}^m_+$ such that $\{m_j : j \in \operatorname{supp}(\lambda)\}$ are linearly independent and $\|M\lambda\| < \hat{\delta}\max_{j\in[m]} \lambda_j\|m_j\|$.

Proof. Observe that outcome (i) corresponds to finding a point in K_c. Let us call the subroutine Approx-Conic-Dual for K_c with $\varepsilon = \hat{\delta}^2/(8n^2)$, using the separation oracle just described. This will require $\mathcal{T}_o(n, \hat{\delta}^2/(8n^2))$ oracle calls and $\mathcal{T}_a(n, \hat{\delta}^2/(8n^2))$ operations. Either we obtain an $x \in K$ with $c^\top x < 0$, or an ε-certificate consisting of oracle inequalities. If these are only original separating inequalities m_i, then we have $\lambda \in \mathbb{R}^m_+$, such that $\sum_{j=1}^m \lambda\|m_j\| \geq 1$ and $\|M^\top\lambda\| \leq \varepsilon$. As in the proof of Lemma 3.1, we can obtain outcomes (ii) or (iv) using Lemma 3.2.

Assume next the combination also includes $-c$: we get $\|M^\top\bar{\lambda} - \tau c\| \leq \varepsilon$ for $(\bar{\lambda},\tau) \in \mathbb{R}^m_+ \times \mathbb{R}_+$ with $\sum_{j=1}^m \lambda\|m_j\| + \tau\|c\| \geq 1$. First, assume that $\tau\|c\| \leq \hat{\delta}/(4n)$. Then, $\|M^\top\bar{\lambda}\| \leq \varepsilon + \tau\|c\| \leq \varepsilon + \hat{\delta}/(4n) < 3\hat{\delta}/(8n)$. At the same time, $\sum_{j=1}^m \lambda\|m_j\| \geq 1 \geq 1 - \tau\|c\| > 3/4$. For $\lambda := 4\bar{\lambda}/3$, we have $\|M^\top\lambda\| \leq \hat{\delta}/(2n)$ and $\sum_{j=1}^m \bar{\lambda}\|m_j\| \geq 1$. As in the previous case, we can obtain outcomes (ii) or (iv).

For the rest, assume that $\tau\|c\| > \hat{\delta}/(4n)$. We perform a Carathéodory reduction to find a vector $\lambda \in \mathbb{R}^m_+$ such that $M^\top\lambda = M^\top\bar{\lambda}/\tau$ and such that $\{m_i : i \in \operatorname{supp}(\lambda)\}$ are linearly independent. Similar to the proof of Lemma 3.2, this requires $O(n^3 + n\tau(n)^2)$ time. We derive outcome (iii) by showing that the set $J := \left\{j \in [m] : \lambda_j\|m_j\| > \|M^\top\lambda - c\|/\hat{\delta}\right\}$ is nonempty. Note that

$$\frac{\|M^\top\lambda - c\|}{\hat{\delta}} = \frac{\|M^\top\bar{\lambda} - \tau c\|}{\tau\hat{\delta}} \leq \frac{4n\|c\|}{\hat{\delta}^2}\cdot\varepsilon = \frac{\|c\|}{2n}\,. \tag{4.10}$$

From the triangle inequality, (4.10), and the assumption $\hat{\delta} < 1$, we obtain

$$\sum_{j\in[m]} \lambda_j\|m_j\| \geq \|c\| - \|M^\top\lambda - c\| \geq (2n - \hat{\delta})\frac{\|M^\top\lambda - c\|}{\hat{\delta}} > n\cdot\frac{\|M^\top\lambda - c\|}{\hat{\delta}}.$$

By the linear independence assumption, $|\operatorname{supp}(\lambda)| \leq n$ and hence $\arg\max_{j\in[m]} \lambda_j\|m_j\| \in J$. ∎

Lemma 4.2. *In Lemma 4.1, if outcome* (iii) *occurs and $\hat{\delta} \leq \delta_M$, then*

$$K_c \neq \emptyset \Rightarrow K_c \cap \{x \in \mathbb{R}^n : m_j^\top x = 0,\ j \in J\} \neq \emptyset.$$

4.1 The recursive algorithm

Similarly to Section 3.1, the recursive calls restrict the problem to a subspace $\ker(M_F)$ for an index set $F \subseteq [m]$ such that $|F| \leq 2\mathrm{rk}(M_F)$. We use the same notation Π^F, v^F, T_F, M^F, and K^F. Similarly to Lemma 3.3, we can implement the required oracle for $(K^F)_c$. We recall from Lemma 3.4 that $\delta_{M^F} \geq \delta_M$. Hence, if we find a certificate $\hat{\delta} > \delta_{M^F}$ in a recursive call then this also implies that $\hat{\delta}$ was a wrong estimate on δ_M.

We initialize $F = \emptyset$. If at any iteration we find a solution $(c^F)^\top x < 0$ for some $x \in K^F$, then we obtain a solution $\Pi^F x \in K$ and $c^\top(\Pi^F x) = (c^F)^\top x < 0$ to the original system.

Claim 1. *Let $F \subset F' \subseteq [m]$ such that $J := F' \setminus F \subseteq T_F$, $|J| \leq 2\mathrm{rk}(M_J^F)$. Let $v \in \mathbb{R}^n$ and $y' \in \mathbb{R}^{T_{F'}}$ such that $(M^{F'})^\top y' = v^{F'}$ and* $\operatorname{supp}(y') \leq 2\mathrm{rk}(M^{F'})$*. Then, in time $O(n^\omega)$ we can compute $y \in \mathbb{R}^{T_F}$ such that $(M^F)^\top y = v^F$,* $\operatorname{supp}(\bar{y}) \subseteq T_{F'} \cup J$*, $|\operatorname{supp}(y)| \leq 2\mathrm{rk}(M^F)$ and $y_i = y'_i$ for $i \in T_{F'}$.*

As in the proof of Lemma 3.1, we maintain an estimate $\hat{\delta}$ of δ_M, updated whenever we detect a failure. We define a subroutine Recursive-Conic-Validity$(K, c, F, \hat{\delta})$, which takes as arguments $F \subseteq [m]$, $c \in \mathbb{R}^n$, and $\hat{\delta} \in (0,1)$. The output of this algorithm is one of the following:

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

(a) A vector $y \in \mathbb{R}_+^{T_F}$ with $|\mathrm{supp}(y)| \leq 2\mathrm{rk}(M^F)$, such that $(M^F)^\top y = c^F$, certifying that $(c^F)^\top x \geq 0$ for all $x \in K^F$.

(b) A point $\bar{x} \in K$ with $c^\top \bar{x} < 0$.

(c) A certificate for $\hat{\delta} > \delta_{M_F}$, namely, a vector $\lambda \in \mathbb{R}_+^{T_F}$ with $\{m_j^F : j \in \mathrm{supp}(\lambda)\}$ linearly independent and $\varphi = \|M^F\lambda\|/(\max_{j\in T^F} \lambda_j \|m_j^F\|) < \hat{\delta}$.

Our algorithm is the following: initialize $\hat{\delta} := 1/n$, and call Recursive-Conic-Validity$(K, c, \emptyset, \hat{\delta})$. If outcomes (a) or (b) occur, then terminate with the desired solution. If outcome (c) occurs, then update $\hat{\delta} := \min\{\hat{\delta}^2, \varphi\}$ and restart the entire algorithm.

We now describe Recursive-Conic-Validity on input $K, c, F, \hat{\delta}$. If $c^F = 0$, we return the trivial solution $y = 0$ to the system $(M^F)^\top y = c^F$, $y \geq 0$. If $c^F \neq 0$, we run the subroutine in Lemma 4.1 for K^F, c^F and $\hat{\delta}$, and perform one of the following actions according to the outcome:

(i) Lemma 4.1 returns $x \in K^F$ with $(c^F)^\top x < 0$; we return $\bar{x} := \Pi^F x$ and the algorithm terminates.

(ii) Lemma 4.1 returns a nonzero $\lambda \in \mathbb{R}_+^{T_F}$ that is a support minimal solution to $(M^F)^\top \lambda = 0$. Let $J := \mathrm{supp}(\lambda)$; λ provides a proof that $(K^F)_c \subseteq K^F \subseteq \{x : M_J^F x = 0\}$. We set $F' := F \cup J$. Note that $|J| \leq \mathrm{rk}(M_J) + 1$, by the minimality of λ, hence $|F'| \leq 2\mathrm{rk}(M_{F'})$. We call Recursive-Conic-Validity$(K, c, F', \hat{\delta})$. If this recursive call outputs $\bar{x} \in K$ with $c^\top \bar{x} < 0$ (outcome (b)), we return $\bar{x}$. If the recursive call outputs a failure (outcome (c)), we return the corresponding φ and λ.

Finally, assume that the recursive call outputs $y' \in \mathbb{R}_+^{T_{F'}}$ such that $(M^{F'})^\top y' = c^{F'}$ (outcome (a)) and $|\mathrm{supp}(y')| \leq 2\mathrm{rk}(M^{F'})$. By Claim 1, we can compute a vector $y \in \mathbb{R}^{T_F}$ with $\mathrm{supp}(y) \in T_{F'} \cup J$, $|\mathrm{supp}(y)| \leq 2\mathrm{rk}(M^F)$, such that $(M^F)^\top y = c^F$, and $y_i = y'_i$ for $i \in T_{F'}$. However, $y_i < 0$ is possible for $i \in J$. Since $M^F \lambda = 0$ and $\lambda_J > 0$, for sufficiently large $\alpha > 0$, we obtain $\bar{y} = y + \alpha\lambda \geq 0$ such that $(M^F)^\top \bar{y} = c^F$; we return the vector $\bar{y}$.

(iii) Lemma 4.1 returns $\lambda \in \mathbb{R}_+^{T_F}$ such that $\{m_i^F : i \in \mathrm{supp}(\lambda)\}$ are linearly independent, along with a nonempty $J \subseteq \mathrm{supp}(\lambda)$ such that $\lambda_j \|m_j^F\| > \|(M^F)^\top \lambda - c^F\|/\hat{\delta}$ for all $j \in J$. If $(M^F)^\top \lambda = c^F$, we can output this vector λ as outcome (a).

By Lemma 4.2, if $\hat{\delta} \leq \delta_M \leq \delta_{M^F}$, then $(K^F)_c \neq \emptyset \Rightarrow (K^F)_c \cap \{x : M_J^F x = 0\} \neq \emptyset$. Therefore, we set $F' = F \cup J$ and call Recursive-Conic-Validity$(K, c, F', \hat{\delta})$. Note that $|F'| \leq 2\mathrm{rk}(M_{F'})$ by the linear independence assumption on λ.

Note that, unlike in case *(ii)* above, we do not have a proof that $M_J x \geq 0$ can be set at equality; indeed, if $\hat{\delta} > \delta_{M^F}$, then we may have set at equality an incorrect set of inequalities. As we will now explain, the algorithm will either return a correct solution, or detect a failure, in which case we will restart from $F = \emptyset$ and an updated value of $\hat{\delta}$.

As in the previous case, if the output of Recursive-Conic-Validity$(K, c, F', \hat{\delta})$ is $\bar{x} \in K$ with $c^\top \bar{x} < 0$ (outcome (b)), we return $\bar{x}$, whereas if the output is a failure (outcome (c)), we return the corresponding φ and λ.

Assume the output is $\bar{y} \in \mathbb{R}_+^{T_{F'}}$ with $|\mathrm{supp}(\bar{y})| \leq 2\mathrm{rk}(M^{F'})$, such that $(M^{F'})^\top \bar{y} = c^{F'}$. By Claim 1, we can compute a vector $y' \in \mathbb{R}^{T_F}$ with $\mathrm{supp}(y') \in T_{F'} \cup J$, $|\mathrm{supp}(y')| \leq 2\mathrm{rk}(M^F)$ such that $(M^F)^\top y' = c^F$, and $y'_i = \bar{y}_i$ for $i \in T_{F'}$. In the case that $y'_i < 0$ for some $i \in J$ we invoke the following lemma.

Lemma 4.3. *Let $H \in \mathbb{R}^{k\times n}$, let $[k] = \mathcal{L}_1 \cup \mathcal{L}_2$, and let $\hat{\delta} \in (0,1)$. Consider $y, y' \in \mathbb{R}^k$ such that $y \geq 0$, $y'_{\mathcal{L}_2} \geq 0$ and*

$$y_i \|h_i\| \geq \|H^\top (y' - y)\|/\hat{\delta}, \qquad \forall i \in \mathcal{L}_1.$$

In time $O(k^3 n)$ we can find one of the following:

(i) A nonnegative vector $q \in \mathbb{R}_+^k$ such that $H^\top q = H^\top y'$ and $|\mathrm{supp}(q)| \leq \mathrm{rk}(H)$.

(ii) $\lambda \in \mathbb{R}^k$, such that $\{h_i : i \in \mathrm{supp}(\lambda)\}$ are linearly independent and $\left\|H^\top \lambda\right\| < \hat{\delta} \max_{i\in[k]} |\lambda_i| \cdot \|h_i\|$.

To remove the negative components of y'_J, we apply the algorithm in Lemma 4.3 with the choice $H = M^F$, $y = \lambda$, $\mathcal{L}_1 = J$ and $\mathcal{L}_2 = T_F \setminus J$. Observe that $H^T(y' - y) = c^F - (M^F)^\top \lambda$, hence y, y' satisfy the assumptions of Lemma 4.3. If outcome *(ii)* of Lemma 4.3 occurs, then we detected a fail since $\hat{\delta} > \delta_{M^F} \geq \delta_M$, and output the corresponding bound φ and combination λ. Otherwise, outcome *(i)* of Lemma 4.3 occurs, we obtain $q \in \mathbb{R}_+^{T_F}$ with $(M^F)^\top q = c^F$, $|\mathrm{supp}(q)| \leq \mathrm{rk}(M^F)$, and we return q as outcome (a).

(iv) Lemma 4.1 returns a failure for $\hat{\delta}$, in which case we return outcome (c), along with the corresponding bound φ and combination λ.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Correctness. It is clear that, if the procedure terminates, it terminates with a correct output, so we only need to argue termination. Note that, for every value of $\hat{\delta}$, each call to Recursive-Conic-Validity$(K, c, \emptyset, \hat{\delta})$ will make at most $\mathrm{rk}(M) \leq n$ recursive calls of the form Recursive-Conic-Validity$(K, c, F, \hat{\delta})$. To see this, note that $\mathrm{rk}(M^F)$ decrease in cardinality by at least 1 at every successive recursive call. Furthermore, once $\mathrm{rk}(M^F) = 0 \Leftrightarrow T^F = \emptyset$, one of the following two things will happen. Either $c^F = 0$, and we return the trivial combination $y = 0$, or $c^F \neq 0$, and then the call to Lemma 4.1 on $K^F, c^F, \hat{\delta}$ must return a solution $x \in K_c$. To justify the latter, simply note that $T^F = \emptyset$ and $\|c^F\| \neq 0$ excludes all outcomes except outcome (i) (indeed, $-c^F \in K_c$).

Lastly, if $\hat{\delta} \leq \delta_M$, then Recursive-Conic-Validity$(K, c, \emptyset, \hat{\delta})$ will not detect any failure, and so it will terminate with one of the two desired outcomes.

Running time analysis For each value of $\hat{\delta}$ set by the algorithm, we have at most n recursive calls to Recursive-Conic-Validity. Recall that each time we update $\hat{\delta}$ to a value which is less than or equal to $\hat{\delta}^2$, and we terminate with $\hat{\delta} \geq \delta_M^2$, for a maximum of $O(\log\log(\delta_M))$ updates. In each recursive call to Recursive-Conic-Validity we call to the algorithm in Lemma 4.1, which requires $\mathcal{T}_o(n, \hat{\delta}^2/(8n^2))$ oracle calls and $\mathcal{T}_a(n, \hat{\delta}^2/(8n^2)) + O(n^3 + n\tau(n)^2)$ arithmetic operations. By Lemma 3.3, each oracle call for K^F requires $O(n^2)$ arithmetic operations. By Lemma 2.1, it follows that the total time required by the calls to Lemma 4.1 is dominated by the time for the last value of $\hat{\delta}$, hence it requires $O(n\mathcal{T}_o(n, \delta_M^2/O(n)))$ oracle calls and $O(n^3\mathcal{T}_o(n, \delta_M^2/O(n)) + n\mathcal{T}_a(n\delta_M^2/O(n))) + O(n^4 + n^2\tau(n)^2)\log\log(1/\delta_M))$ arithmetic operations.

At each recursive call, we need to compute the projection matrix Π^F, which requires $O(n^3)$ arithmetic operations. In case (ii) of the recursion, the running time is dominated by the application of Claim 1, which requires $O(n^\omega)$ operations (since $|F| \leq 2n$). In case (iii) of the recursion, the running time is dominated by the application of Lemma 4.3, which requires $O(n^4)$ operations (observe that this is because, when we apply the lemma to $H = M^F$, we can limit ourselves to the rows of H corresponding to $\mathrm{supp}(y) \cup \mathrm{supp}(y')$, and by construction $|\mathrm{supp}(y)|, |\mathrm{supp}(y')| = O(n)$). Since we have n recursive call per value of $\hat{\delta}$, and $\hat{\delta}$ is updated at most $\log\log(\delta_M)$ times, it follows that the running time of all these operations is bounded by $O(n^5 \log\log(1/\delta_M))$.

5 Computing approximate dual certificates

Our goal in this section is to exhibit a general technique for implementing the Approx-Conic-Dual oracle using various methods. We define a more general notion of dual certificates also applicable for the non-conic setting.

Definition 2. *Given a convex set $K \subseteq \mathbb{R}^n$ and $r, \varepsilon > 0$, an* ε-approximate Farkas certificate *for $K \cap \mathbb{B}^n(r)$ is given by a system $Ax \leq u$ of valid inequalities for K, $A \in \mathbb{R}^{m\times n}$, $u \in \mathbb{R}^m$, and multipliers $\lambda \in \mathbb{R}^J_{++}$*

$$\lambda^\top u + r\left\|A^\top \lambda\right\| < \varepsilon, \qquad \sum_{i=1}^m \lambda_i \|a_i\| \geq 1.$$

If $K \subseteq \mathbb{R}^n$ is a cone given by a conic separation oracle, then we can assume $u_i = 0$ for all oracle inequalities $a_i^\top x \leq 0$. Setting $r = 1$, an *ε-approximate Farkas certificate* for $K \cap \mathbb{B}^n(1)$ using oracle inequalities coincides with the notion of an *ε-approximate conic Farkas certificate* for K as required in Approx-Conic-Dual.

The Lee, Sidford, and Wong's cutting plane method [26] (LSW algorithm) explicitly provides dual certificates; the proof follows easily using Theorem 31 in the paper.

Theorem 5.1. *Let K be a convex set given by a strong separation oracle, $r > 0$, and $\varepsilon \in (0, 2r)$. Then, in expected $O(n\log(nr/\varepsilon))$ calls to the separation oracle, and expected $O(n^3 \log^{O(1)}(nr/\varepsilon))$ arithmetic operations, the LSW algorithm either returns a point $x \in K$, or an ε-approximate Farkas certificate for $K \cap \mathbb{B}^n(r)$ comprising only oracle inequalities.*

The rest of this section is dedicated to showing that ε-Farkas certificates can be recovered from a broad class of algorithms, including seemingly 'primal-only' methods such as the ellipsoid method. For any $R \in \mathbb{S}^n_{++}$ and $p \in \mathbb{R}^n$, we define the ellipsoid

$$E(R, p) \stackrel{\text{def}}{=} \{z \in \mathbb{R}^d \,:\, \|z - p\|_R \leq 1\}.$$

Given a compact set $K \subseteq \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$, we define the *width of K along v* as

$$\mathrm{width}_K(v) \stackrel{\text{def}}{=} \max\{v^\top z \,:\, z \in K\} - \min\{v^\top x \,:\, z \in K\}. \tag{5.11}$$

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We say that v is an *ε-thin direction for K* if $\mathrm{width}_K(v) \le \varepsilon$. The width of an ellipsoid can be characterized as follows. Recall that for every $v \in \mathbb{R}^n$, $\min\{v^\top x : x \in E(R,p)\} = v^\top p - \|v\|_{R^{-1}}$, achieved by $x^* = p - R^{-1}v/\|v\|_{R^{-1}}$.

LEMMA 5.1. *Given $R \in \mathbb{S}^n_{++}$, and $p \in \mathbb{R}^n$, let $E := E(R,p)$. For any $v \in \mathbb{R}^d$, $\mathrm{width}_E(v) = 2\|v\|_{R^{-1}}$. In particular, for $K = E(R,p)$, v is an ε-thin direction if and only if $\|v\|_{R^{-1}} \le \varepsilon/2$.*

Consider the feasibility problem for a polyhedron $P \subseteq \mathbb{R}^n$ given by a strong separation oracle. The algorithms discussed in this section—the ellipsoid method, Vaidya's cutting plane methods [36], as well as the geometric rescaling algorithms [12, 19]—proceed by maintaining a containing ellipsoid $E(R,p) \supseteq P \cap \mathbb{B}^n(r)$. In every iteration, they either terminate with a feasible solution, or modify the containing ellipsoid; the main progress measure is decrease in the volume of the ellipsoid.

In particular, all these methods maintain the matrix R in the form $R = \gamma_0 I_n + \sum_{i=1}^m \gamma_i a_i a_i^\top$ for coefficients $\gamma \in \mathbb{R}^{m+1}$ and vectors a_i returned by the oracle calls. The next lemma shows that if the volume of $E(R,p)$ is small, or equivalently the determinant of R is large, then P must be thin in one of the directions a_i returned by the oracle. We include the simple proof for completeness.

LEMMA 5.2. ([12, LEMMA 4.11]) *Let $R \in \mathbb{S}^n_{++}$ be defined by*

$$R = \gamma_0 I_n + \sum_{i=1}^m \gamma_i a_i a_i^\top,$$

where $a_1, \dots, a_m \in \mathbb{R}^n$, and $\gamma_0, \dots, \gamma_m \ge 0$, with $\gamma_0 \le 1$. Then, for every $i \in [t]$, $\gamma_i \|a_i\|^2_{R^{-1}} < 1$ holds, and $\sum_{i=1}^t \gamma_i \|a_i\|^2_{R^{-1}} \le n$. Further, if $\det(R) > 1$, then there exists $k \in [t]$ such that

$$\|a_k\|_{R^{-1}} \le \frac{\|a_k\|_2}{\sqrt{\det(R)^{1/n} - 1}}\,.$$

Proof. Let $Q = R^{-1}$. The bound $\gamma_i \|a_i\|^2_Q < 1$ follows by

$$\|a_i\|^2_Q = a_i Q R Q a_i = a_i Q \left(\gamma_0 I_n + \sum_{j=1}^m \gamma_j a_j a_j^\top\right) Q a_i > \gamma_i \|a_i\|^4_Q\,.$$

For $\sum_{i=1}^m \gamma_i \|a_i\|^2_Q < n$, we see that

$$\begin{aligned}\sum_{i=1}^m \gamma_i \|a_i\|^2_Q &= \sum_{i=1}^m \gamma_i \left(a_i^\top Q a_i\right) = \mathrm{tr}\left(Q \sum_{i=1}^m \gamma_i a_i a_i^\top\right) \\ &= \mathrm{tr}(Q(R - \gamma_0 I_n)) = \mathrm{tr}(I_n) - \gamma_0 \mathrm{tr}(Q) < n\,.\end{aligned} \tag{5.12}$$

In the final inequality we used that $\mathrm{tr}(Q) > 0$, since Q is positive definite.

For the third claim, we see that $\mathrm{tr}(R) = \gamma_0 n + \sum_{i=1}^m \gamma_i \|a_i\|_2^2$ since $\|a_i\|_2 \le 2$. Noting that $\gamma_0 \le 1$, $\sum_{i=1}^m \gamma_i \|a_i\|_2^2 \ge \mathrm{tr}(R) - n \ge n(\det(R)^{1/n} - 1)$, using the well-known inequality $\det(R)^{1/n} \le \mathrm{tr}(R)/n$ for positive semidefinite matrices. Let $k = \arg\min_{i \in [m]}(\|a_i\|_Q / \|a_k\|_2)$. Using the bound $\sum_{i=1}^m \gamma_i \|a_i\|^2_Q < n$, we see that

$$\frac{\|a_k\|_Q^2}{\|a_k\|_2^2} \le \frac{n}{\sum_{i=1}^m \gamma_i \|a_k\|_2^2} < \frac{1}{\det(R)^{1/n} - 1}\,.$$

□

Thus, we can identify a thin direction a_k. Our goal in this section is to provide a dual certificate of thinness of $P \cap \mathbb{B}^n(r)$, using the oracle inequalities $a_i^\top x \le u_i$ and the initial ball constraint $\|x\|_2 \le r$. For a conic set P, this will imply ε-approximate conic Farkas certificate from the algorithms mentioned.

Our certification scheme builds on the work of Burrell and Todd [7] on the ellipsoid method. The key idea is to use an alternative representation of the strictly concave quadratic form $q(x) = -(x-p)^\top R(x-p)$ corresponding to the ellipsoid $E(R,p)$. In Section 5.1, we introduce *certified concave quadratic forms*, and show how this representation can be used to construct dual certificates for valid inequalities. These ingredients are combined to derive the Farkas certificate in Section 5.2. In the full version of the paper we demonstrate the use of certified concave quadratic forms for the ellipsoid method, volumetric cutting plane methods, and for geometric rescaling methods.

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

5.1 Dual certificates from certified quadratic forms Duality theory provides the following variant of Farkas' lemma.

LEMMA 5.3. *Given $A \in \mathbb{R}^{m\times n}$, $u \in \mathbb{R}^m$, $r > 0$, the system $Ax \le u$ has no solution in $\mathbb{B}^n(r)$ if and only if there exists $\lambda \in \mathbb{R}^m_+$ such that*

$$r\|A^\top \lambda\| < -\lambda^\top u\,. \tag{5.13}$$

Further, given $v \in \mathbb{R}^n$, the inequality $v^\top x \ge \nu$ is valid for $\{x \in \mathbb{R}^n : Ax \le u\} \cap \mathbb{B}^n(r)$ if and only if there exists $\lambda \in \mathbb{R}^m_+$ such that

$$r\|A^\top \lambda + v\| + \nu \le -\lambda^\top u\,. \tag{5.14}$$

Consider a polyhedron $P = \{x \in \mathbb{R}^n : Ax \le u\}$, where $A \in \mathbb{R}^{n\times m}$ and $u \in \mathbb{R}^m$. Let a_i, $i \in [m]$ be the rows of A. We will refer to $\lambda \in \mathbb{R}^m_+$ satisfying (5.14) as a *dual certificate of validity* of $v^\top x \ge \nu$ for $P \cap \mathbb{B}^n(r)$.

DEFINITION 3. *Let $P = \{x : Ax \le u\}$ for $A \in \mathbb{R}^{m\times n}$, $u \in \mathbb{R}^m$, and $r > 0$. Let $q : \mathbb{R}^n \to \mathbb{R}$ be a concave quadratic form given as*

$$q(x) := \gamma_0(r^2 - \|x\|^2) + \sum_{i=1}^m \gamma_i(u_i - a_i^\top x)(a_i^\top x - \ell_i) + d^\top x - \beta \tag{5.15}$$

for $\gamma \in \mathbb{R}^{m+1}_+$, $\ell \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, $\beta \in \mathbb{R}$. We say that P is a certified concave quadratic form *for $P \cap \mathbb{B}^n(r)$ if we are also given $\mu^{(i)} \in \mathbb{R}^m_+$, $i \in [m]$, $\vartheta \in \mathbb{R}^m_+$ such that*

- $r\|A^\top \mu^{(i)} + a_i\| + \ell_i \le -u^\top \mu^{(i)}$ *(certifying $a_i^\top x \ge \ell_i$ for $P \cap \mathbb{B}^n(r)$)*
- $r\|A^\top \vartheta + d\| + \beta \le -\vartheta^\top u$ *(certifying $d^\top x \ge \beta$ for $P \cap \mathbb{B}^n(r)$)*

We will also say that the quadratic form (5.15) *is* certified *by $\mu^{(i)}$, $i \in [m]$, and ϑ.*

The certificates $\mu^{(i)}$, $i \in [m]$, and ϑ guarantee that $P \cap \mathbb{B}^n(r) \subseteq \{x : q(x) \ge 0\}$. We will show in Lemma 5.5 that any inequality $v^\top x \ge \nu$ that is valid for $\{x : q(x) \ge 0\}$ admits a closed-form dual certificate of its validity for $P \cap \mathbb{B}^n(r)$ that can be derived from the representation of $q(x)$. As the first step, we need the following technical lemma.

LEMMA 5.4. *Let q be a strictly concave quadratic form q for $A, u, r, \ell, \gamma, d, \beta$ as in Definition 3. In particular, assume we are also give $\vartheta \in \mathbb{R}^m_+$ certifying $d^\top x \ge \beta$ for $P \cap \mathbb{B}^n(r)$ as $r\|A^\top \vartheta + d\| + \beta \le -\vartheta^\top u$. Let $p \in \mathbb{R}^n$ be a maximizer of $q(x)$. Define $\lambda \in \mathbb{R}^m$ by $\lambda_i = \gamma_i(\ell_i + u_i - 2a_i^\top p)$, $i \in [m]$. Then*

$$r\|A^\top(\lambda - \vartheta)\| \le \max_{x\in\mathbb{R}^n} q(x) + \ell^\top \lambda^+ - u^\top \lambda^- - u^\top \vartheta.$$

Proof. Since q is a strictly concave quadratic form, it achieves a maximum $p \in \mathbb{R}^n$, which must satisfy $\nabla q(p) = 0$. We use the notation $\bar{\ell} := Ap - \ell$ and $\bar{u} := u - Ap$. The following equation states that $\nabla q(p) = 0$, and the next one computes the value of $q(p)$, expressed with $\bar{u}_i$ and $\bar{\ell}_i$:

$$\sum_{i=1}^m \gamma_i(\bar{u}_i - \bar{\ell}_i)a_i + d \quad = \quad 2\gamma_0 p \tag{5.16}$$

$$\gamma_0(r^2 - \|p\|^2) + \sum_{i=1}^m \gamma_i \bar{u}_i \bar{\ell}_i + d^\top p - \beta \quad = \quad q(p)\,. \tag{5.17}$$

Note that $\lambda_i = \gamma_i(\bar{u}_i - \bar{\ell}_i)$ for all $i \in [m]$. Hence, (5.16) can be written as $A^\top \lambda + d = 2\gamma_0 p$. Let

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$P := \{i \in [m] \,:\, \lambda_i \geq 0\}$ and $N := [m] \setminus P$. The proof is completed by

$$
\begin{array}{rcl}
\ell^\top \lambda^+ - u^\top \lambda^- - u^\top \vartheta & = & -\sum_{i \in P} \lambda_i \bar{\ell}_i + \sum_{i \in N} \lambda_i \bar{u}_i + \sum_{i=1}^m \lambda_i a_i^\top p - u^\top \vartheta \\
\text{(by (5.16))} & = & -\sum_{i \in P} \gamma_i (\bar{u}_i - \bar{\ell}_i)\bar{\ell}_i + \sum_{i \in N} \gamma_i (\bar{u}_i - \bar{\ell}_i)\bar{u}_i + 2\gamma_0 \|p\|^2 - d^\top p - u^\top \vartheta \\
 & = & -\sum_{i=1}^m \gamma_i \bar{u}_i \bar{\ell}_i + \sum_{i \in P} \gamma_i \bar{\ell}_i^2 + \sum_{i \in N} \gamma_i \bar{u}_i^2 + 2\gamma_0 \|p\|^2 - d^\top p - u^\top \vartheta \\
\text{(by (5.17))} & \geq & -q(p) + \gamma_0 (r^2 + \|p\|^2) - \beta - u^\top \vartheta \\
\text{(by the definition of } \vartheta) & \geq & -q(p) + 2r\gamma_0 \|p\| + r\|A^\top \vartheta + a\| \\
\text{(by (5.16))} & = & -q(p) + r \left\| A^\top \lambda + a \right\| + r\|A^\top \vartheta + a\| \\
 & \geq & -q(p) + r \left\| A^\top (\lambda - \vartheta) \right\| .
\end{array}
$$

□

The next lemma shows that, if P is a polyhedron and q is a strictly concave quadratic form certified for $P \cap \mathbb{B}_n(r)$, then we can compute a dual certificate for $P \cap \mathbb{B}_n(r)$ for the inequality $v^\top x \geq \nu$ where $\nu = \min\{v^\top x \,:\, q(x) \geq 0\}$. This is a variant of [7, Proposition 3.1 and Theorem 3.2]. Recall that if q is a strictly concave quadratic form, then there exist $R \in \mathbb{S}^n_{++}$ such that $q(x) = -(x-p)^\top R(x-p) + q(p)$, where p is the unique maximizer of q. In particular, if $q(p) > 0$ then $\{x \in \mathbb{R}^n \,:\, q(x) \geq 0\} = \sqrt{q(p)}E(R,p)$.

Lemma 5.5. *Let $P = \{x \in \mathbb{R}^n \,:\, Ax \leq u\}$, where $A \in \mathbb{R}^{m \times n}$ and $u \in \mathbb{R}^m$. Let $r > 0$, and let $q : \mathbb{R}^n \to \mathbb{R}$ be a strictly concave quadratic form as in (5.15), certified for $P \cap \mathbb{B}^n(r)$ by $\mu^{(i)} \in \mathbb{R}^m_+$, $i \in [m]$ and $\vartheta \in \mathbb{R}^m_+$. Assume that $\max_{x \in \mathbb{R}^n} q(x) > 0$, and let $p := \arg\max q(x)$, $\alpha := \sqrt{q(p)}$. Define $R = \gamma_0 I_n + \sum_{i=1}^m \gamma_i a_i a_i^\top$.*

Given $v \in \mathbb{R}^n$, let $\nu := v^\top p - \alpha \|v\|_{R^{-1}}$ and $x^* = p - \alpha R^{-1} v / \|v\|_{R^{-1}}$. Define $\lambda, \tilde{\lambda} \in \mathbb{R}^m$ by

$$
\lambda_i = \frac{\|v\|_{R^{-1}}}{2\alpha} \gamma_i (\ell_i + u_i - 2a_i^\top x^*) \quad \forall i \in [m] , \qquad \tilde{\lambda} := \sum_{i=1}^m \lambda_i^+ \mu^{(i)} + \lambda^- + \frac{\|v\|_{R^{-1}}}{2\alpha} \vartheta.
$$

Then, $\tilde{\lambda}$ is a dual certificate for $v^\top x \geq \nu$ for $P \cap \mathbb{B}^n(r)$, that is, $r\|A^\top \tilde{\lambda} + v\| + \nu < -\tilde{\lambda}^\top u$. Furthermore, $\tilde{\lambda}$ can be computed in time $O(n^2 m + n^\omega)$.

Proof. Note that $x^* = \arg\min\{v^\top x \,:\, x \in E(R,p)\}$ and $v^\top x^* = \nu$. Define $\sigma := \frac{\|v\|_{R^{-1}}}{2\alpha}$ and $\tilde{\gamma}_i := \sigma \gamma_i$ for $i = 0, \ldots, m$. Consider the polyhedron $\tilde{P} = \{x \,:\, Ax \leq u, \, v^\top x \leq \nu\}$, and the quadratic form defined by $\tilde{q}(x) = \sigma q(x) - v^\top x + \nu$. Observe that

$$
\tilde{q}(x) := \tilde{\gamma}_0 (r^2 - \|x\|^2) + \sum_{i=1}^m \tilde{\gamma}_i (u_i - a_i^\top x)(a_i^\top x - \ell_i) + (\sigma d - v)^\top x - (\sigma \beta - \nu) ,
$$

hence $\tilde{q}$ is certified for $\tilde{P} \cap B^n(r)$ by $\lambda^{(i)}$, $i \in [m]$, and by $\|A^\top(\sigma\vartheta) + v + (\sigma d - v)\| + \sigma\beta - \nu \leq -\sigma\vartheta^\top u - \nu$.

Since $\tilde{q}(x) = \sigma(\alpha^2 - (x-p)^\top R(x-p)) + \nu - v^\top x$, we have that $\tilde{q}$ is strictly concave and $\nabla \tilde{q}(x) = -2\sigma R(x-p) - v$. This implies that x^* is the unique maximizer of $\tilde{q}$ since $\nabla \tilde{q}(x^*) = 0$. Furthermore, one can compute that $\tilde{q}(x^*) = 0$. Applying Lemma 5.4 to $\tilde{q}$, and observing that $\lambda_i = \tilde{\gamma}_i (u_i + \ell_i - 2a_i^\top x^*)$, $i \in [m]$, we obtain

$$
\ell^\top \lambda^+ - u^\top \lambda^- - u^\top (\sigma\vartheta) - \nu \geq \tilde{q}(x^*) + r \left\| A^\top (\lambda - \sigma\vartheta) - v \right\| = r \left\| A^\top (\lambda - \sigma\vartheta) - v \right\| . \tag{5.18}
$$

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We need to show that $r\|A^\top\tilde{\lambda}+v\|+\nu \le -\tilde{\lambda}^\top u$. From (5.18), we have

$$\begin{aligned}
\nu &\le \ell^\top\lambda^+ - u^\top(\lambda^- + \sigma\vartheta) - r\left\|A^\top(\lambda-\sigma\vartheta)-v\right\| \\
&\le \sum_{i=1}^m \lambda_i^+\left(-u^\top\mu^{(i)} - r\left\|A^\top\mu^{(i)}+a_i\right\|\right) - u^\top(\lambda^- + \sigma\vartheta) - r\left\|A^\top(\lambda-\sigma\vartheta)-v\right\| \\
\text{(triangle inequality)}\quad &\le -u^\top\left(\sum_{i=1}^m \lambda_i^+\mu^{(i)} + \lambda^- + \sigma\vartheta\right) - r\left\|\sum_{i=1}^m \lambda_i^+\left(A^\top\mu^{(i)}+a_i\right) - A^\top(\lambda-\sigma\vartheta)+v\right\| \\
&= -u^\top\left(\sum_{i=1}^m \lambda_i^+\mu^{(i)} + \lambda^- + \sigma\vartheta\right) - r\left\|A^\top\left(\sum_{i=1}^m \lambda_i^+\mu^{(i)} + \lambda^- + \sigma\vartheta\right)+v\right\| \\
&= -\tilde{\lambda}^\top u - r\|A^\top\tilde{\lambda}+v\|.
\end{aligned}$$

To compute $\tilde{\lambda}$, we need to compute R, which can be done in time $O(n^2m)$, as well as x^* and p. The time to compute these two points is dominated by the computation of R^{-1}, which can be performed in time $O(n^\omega)$. Computing λ and $\tilde{\lambda}$ requires time $O(nm)$. □

The above lemma will be used to compute ε-approximate Farkas certificates from the ellipsoid method, from volumetric cutting plane methods [21, 26, 36], and from the geometric rescaling algorithms [12, 19] (see full version). For all three, we will need to show that we can find an appropriate certified quadratic form for the polyhedron defined by the current set of oracle inequalities. For the ellipsoid method and the geometric rescaling algorithms, such quadratic form will need to be maintained explicitly at every iteration. For the volumetric cutting plane algorithms, we will instead show that, once the algorithm has achieved the required level of accuracy, we can a-posteriori compute a suitable certified quadratic form directly from the information that is maintained by the algorithm.

5.2 Finding approximate Farkas certificates The following theorem is the main technical tool for finding an ε-approximate Farkas certificate. The theorem shows that, given a convex set K, if we have a strictly concave certified quadratic form $q(x)$ for $K\cap\mathbb{B}^n(r)$, and if we have a direction v, $\|v\|\ge 1$, such that the ellipsoid $\{x\,:\,q(x)\ge 0\}$ has small width in the direction of v, then we can compute an ε-approximate Farkas certificate for $K\cap\mathbb{B}^n(r)$. All methods we consider start from some initial simple set containing $K\cap\mathbb{B}^n(r)$. For the ellipsoid method and the geometric rescaling algorithm, the initial relaxation is simply $\mathbb{B}^n(r)$, whereas for the volumetric cutting plane algorithms, the initial relaxation is $[-r,r]^n$. In particular, for the ellipsoid method and the geometric rescaling algorithms, the certified quadratic form (5.15) has $\gamma_0>0$ (this corresponds to the initial quadratic form $r^2-\|x\|^2\ge 0$), whereas for Vaidya's algorithm we will always have $\gamma_0=0$, but some of the inequalities $a_i^\top x\le u_i$ may be the initial box-constraints $x_j\le r$ or $-x_j\le r$. Note that, in both cases, the ε-approximate Farkas certificate computed using the theorem below will always be purely in terms of the oracle inequalities for K.

Theorem 5.2. *Let $K\subseteq\mathbb{R}^n$ be a convex set, $r>0$, and $\varepsilon\in(0,2r)$. Assume we are given inequalities $Ax\le u$ valid for K, $A\in\mathbb{R}^{n\times n}$, $u\in\mathbb{R}^m$, and let $\bar{A}x\le\bar{u}$, $\bar{A}\in\mathbb{R}^{k\times n}$, $\bar{u}\in\mathbb{R}^k$, $k\ge m$, be a system comprising all inequalities in $Ax\le u$ and some of the "box constraints" $x_j\le r$ or $-x_j\le r$, $j\in[n]$. Assume we are also given $\ell\in\mathbb{R}^k$ and $\mu^{(i)}\in\mathbb{R}^k_+$, $i\in[k]$ such that $r\|\bar{A}^\top\lambda^{(i)}+a_i\|+\ell_i\le-\bar{u}^\top\mu^{(i)}$, and $d\in\mathbb{R}^n$, $\beta\in\mathbb{R}$, $\vartheta\in\mathbb{R}^k_+$ such that $\|\bar{A}\vartheta+d\|+\beta\le-\bar{u}^\top\vartheta$. Let $\gamma\in\mathbb{R}^{k+1}_+$, and assume that the quadratic form*

$$q(x)=\gamma_0(r^2-\|x\|^2)+\sum_{i=1}^k\gamma_i(u_i-a_i^\top x)(a_i^\top x-\ell_i)+d^\top x-\beta.$$

is strictly concave. Let $E=\{x\,:\,q(x)\ge 0\}$. If we are given $v\in\mathbb{R}^n$ such that $\|v\|\ge 1$ and $\mathrm{width}_E(v)\le\varepsilon/3$, then in time $O(n^2m+n^\omega)$ we can compute an ε-approximate Farkas certificate for $K\cap\mathbb{B}^n(r)$ in terms of the inequalities $Ax\le u$, that is, $\lambda\in\mathbb{R}^m_+$ such that $\lambda^\top u+r\left\|A^\top\lambda\right\|<\varepsilon$, $\sum_{i=1}^m\lambda_i\|a_i\|\ge 1$.

Proof. Recall that E is an ellipsoid centered at $p:=\arg\max_{x\in\mathbb{R}^n}q(x)$. Since $\mathrm{width}_E(v)\le\varepsilon/3$, it follows that $E\subseteq\{x\in\mathbb{R}^n\,:\,-\varepsilon/6\le v^\top x-v^\top p\le\varepsilon/6\}$.

It will be convenient to assume $\|a_i\|\in[1,2]$ for $i\in[m]$, which is without loss of generality. By Lemma 5.5, in time $O(n^2m+n^\omega)$ we can compute dual certificates for $\{x\,:\,\bar{A}x\le\bar{u}\}\cap\mathbb{B}^n(r)$ of both inequalities $-v^\top x\ge -v^\top p-\varepsilon/6$ and $v^\top x\ge v^\top p-\varepsilon/6$. To distinguish the roles of the constraints in $Ax\le u$ from the box constraints, we express these certificates by two vectors $(\lambda',\mu'),(\lambda'',\mu'')\in\mathbb{R}^m_+\times\mathbb{R}^n$, where $(\mu')^+,(\mu'')^+$

Copyright © 2022
Copyright for this paper is retained by the authors

define the multipliers for the inequalities $x_j \le r$, and $(\mu')^-, (\mu'')^-$ define the multipliers for the inequalities $-x_j \le r$. Hence (λ', μ') and (λ'', μ'') satisfy

$$r\|A^\top \lambda' + \mu' - v\| - v^\top p - \varepsilon/6 \le -u^\top \lambda' - r\|\mu'\|_1, \quad r\|A^\top \lambda'' + \mu'' + v\| + v^\top p - \varepsilon/6 \le -u^\top \lambda'' - r\|\mu''\|_1.$$

Note that, since $\|\mu'\| \le \|\mu'\|_1$ and $\|\mu''\| \le \|\mu''\|_1$, the triangle inequality implies

$$r\|A^\top \lambda' - v\| - v^\top p - \varepsilon/6 \le -u^\top \lambda', \quad r\|A^\top \lambda'' + v\| + v^\top p - \varepsilon/6 \le -u^\top \lambda''. \tag{5.19}$$

In what follows, we show that $\lambda = (\lambda' + \lambda'')/\|\lambda' + \lambda''\|_1$ is a ε-approximate Farkas certificate. By definition, $\|\lambda\|_1 = 1$, hence $\sum_{i=1}^m \lambda_i \|a_i\| \ge 1$ by our assumption that $\|a_i\| \in [1, 2]$.

Adding up the two inequalities in (5.19) we obtain

$$\frac{\varepsilon}{3} \ge u^\top(\lambda' + \lambda'') + r\|A^\top \lambda' - v\| + r\|A^\top \lambda'' + v\|$$

From the triangle inequality, we have

$$\alpha := \|A^\top \lambda' - v\| + \|A^\top \lambda'' + v\| - \|A^\top(\lambda' + \lambda'')\| \ge 0\,.$$

If $\alpha > \frac{\varepsilon}{3r}$, then the two equations above give $u^\top(\lambda' + \lambda'') + r\|A^\top(\lambda' + \lambda'')\| < 0$, proving $u^\top\lambda + \|A^\top\lambda\| < 0$, and we are done.

Assume therefore that $\alpha \le \frac{\varepsilon}{3r}$. Note that

$$\frac{\varepsilon}{3\|\lambda' + \lambda''\|_1} \ge u^\top \lambda + r\|A^\top \lambda\|\,,$$

hence it suffices to show that $\|\lambda' + \lambda''\|_1 = \|\lambda'\|_1 + \|\lambda''\|_1 \ge 1/3$.

From the triangle inequality, using that $\|a_i\| \in [1, 2]$ for all $i \in [m]$, we obtain

$$1 \le \|v\| \le \left\|A^\top \lambda'\right\| + \left\|A^\top \lambda' - v\right\| \le 2\|\lambda'\|_1 + \left\|A^\top \lambda' - v\right\|\,,$$

and a similar inequality holds for λ''. Adding up the two bounds and using the definition of α, we get

$$\alpha + \left\|A^\top(\lambda' + \lambda'')\right\| = \left\|A^\top \lambda' - v\right\| + \left\|A^\top \lambda'' - v\right\| \ge 2 - 2(\|\lambda'\|_1 + \|\lambda''\|_1)\,.$$

Using the assumption $\alpha \le \varepsilon/(3r)$ and the upper bounds $\|a_i\|_2 \le 2$,

$$\frac{\varepsilon}{3r} + 2(\|\lambda'\|_1 + \|\lambda''\|_1) \ge 2 - 2(\|\lambda'\|_1 + \|\lambda''\|_1).$$

Since $\varepsilon \le 2r$, the above implies $\|\lambda'\|_1 + \|\lambda''\|_1 \ge \frac{1}{3}$ as required. □

References

[1] D. S. Atkinson and P. M. Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69(1):1–43, 1995.

[2] D. Bertsimas and S. Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

[3] U. Betke. Relaxation, new combinatorial and polynomial algorithms for the linear feasibility problem. *Discrete & Computational Geometry*, 32(3):317–338, 2004.

[4] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation.* Springer Science & Business Media, 1998.

[5] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, 1989.

[6] T. Brunsch and H. Röglin. Finding short paths on polytopes by the shadow vertex algorithm. In *International Colloquium on Automata, Languages, and Programming*, pages 279–290. Springer, 2013.

[7] B. P. Burrell and M. J. Todd. The ellipsoid method generates dual variables. *Mathematics of Operations Research*, 10(4):688–700, 1985.

[8] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021.

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © 2022
Copyright for this paper is retained by the authors

[9] D. Dadush and N. Hähnle. On the shadow simplex method for curved polyhedra. *Discrete & Computational Geometry*, 56(4):882–909, 2016.

[10] 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 SIGACT Symposium on Theory of Computing*, pages 761–774, 2020.

[11] D. Dadush, B. Natura, and L. A. Végh. Revisiting Tardos's framework for linear programming: Faster exact solutions using approximate solvers. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, 2020.

[12] D. Dadush, L. A. Végh, and G. Zambelli. Rescaling algorithms for linear conic feasibility. *Mathematics of Operations Research*, 45(2):732–754, 2020.

[13] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Mathematical Programming*, 114(1):101–114, 2008.

[14] F. Eisenbrand and S. Vempala. Geometric random edge. *Mathematical Programming*, 164(1-2):325–339, 2017.

[15] A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[16] M. Grötschel, L. Lovász, and A. Schrijver. Geometric methods in combinatorial optimization. In *Progress in combinatorial optimization*, pages 167–183. Elsevier, 1984.

[17] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[18] O. Güler, A. J. Hoffman, and U. G. Rothblum. Approximations to solutions to systems of linear inequalities. *SIAM Journal on Matrix Analysis and Applications*, 16(2):688–696, 1995.

[19] R. Hoberg and T. Rothvoß. An improved deterministic rescaling for linear programming algorithms. In *Integer Programming and Combinatorial Optimization (IPCO)*, volume 10328 of *Lecture Notes in Comput. Sci.*, pages 267–278. Springer, Cham, 2017.

[20] H. Jiang. Minimizing convex functions with integral minimizers. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 976–985. SIAM, 2021.

[21] H. Jiang, Y. T. Lee, Z. Song, and S. C.-w. Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 944–953, 2020.

[22] L. G. Khachiyan. A polynomial algorithm in linear programming (in Russian). *Doklady Akademiia Nauk SSSR 224*, 224:1093–1096, 1979. (English Translation: Soviet Mathematics Doklady 20, 191-194.).

[23] D. Klatte and G. Thiere. Error bounds for solutions of linear equations and inequalities. *Zeitschrift für Operations Research*, 41(2):191–214, 1995.

[24] J. Lamperski, R. M. Freund, and M. J. Todd. An oblivious ellipsoid algorithm for solving a system of (in) feasible linear inequalities. *arXiv preprint arXiv:1910.03114*, 2019.

[25] Y. T. Lee and A. Sidford. Solving linear programs with $\sqrt{\text{rank}}$ linear system solves. *arXiv preprint arXiv:1910.08033*, 2019.

[26] Y. T. Lee, A. Sidford, and S. C.-w. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1049–1065. IEEE, 2015.

[27] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[28] A. Nemirovski, S. Onn, and U. G. Rothblum. Accuracy certificates for computational problems with convex structure. *Mathematics of Operations Research*, 35(1):52–78, 2010.

[29] A. S. Nemirovski and D. B. Yudin. Problem complexity and method efficiency in optimization (in Russian). 1979. (English translation: Wiley-Intersci. Ser. Discrete Math. 15, John Wiley, New York, 1983.).

[30] J. Pena and N. Soheili. Projection and rescaling algorithm for finding most interior solutions to polyhedral conic systems. *arXiv preprint arXiv:2003.08911*, 2020.

[31] M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.

[32] S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998.

[33] O. Svensson, J. Tarnawski, and L. A. Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. *Journal of the ACM (JACM)*, 67(6):1–53, 2020.

[34] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, pages 250–256, 1986.

[35] J. F. Traub and H. Woźniakowski. Complexity of linear programming. *Operations Research Letters*,

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © 2022
Copyright for this paper is retained by the authors

Downloaded 04/11/25 to 84.29.92.82 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

1(2):59–62, 1982.

[36] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.

[37] J. van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 259–278, 2020.

[38] J. van den Brand, Y. T. 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)*, pages 775–788, 2020.

[39] S. A. Vavasis and Y. Ye. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming*, 74(1):79–120, 1996.

Copyright © 2022
Copyright for this paper is retained by the authors