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: $\max\{c^\top x : x \in P\}$, where the polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is given by a separation oracle that returns violated inequalities from the system $Ax \leq b$. 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 and do not require any a-priori knowledge of δ.

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 $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a polyhedron given by $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$; the i-th row of A is denoted by $a_i^\top$. The *linear feasibility problem* is to either find $x \in P$ or conclude that $P = \emptyset$. 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 exact 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 $Ax \leq b$; see Section 2 for a 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 [24]. 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 a perturbation $\tilde{b}$ of the right-hand-side, based on bit-complexity arguments, such that the polyhedron $\tilde{P} = \{x : Ax \leq \tilde{b}\}$ satisfies $\tilde{P} = \emptyset$ if and only if $P = \emptyset$, and $\tilde{P}$ is full-dimensional whenever nonempty.

Grötschel, Lovász, and Schrijver [18, 19] 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

*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).

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 [29].

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 ($+$, $-$, $\times$, $/$); certain models allow further operations such as $\sqrt{}$ and log. In the context of LP, Traub and Woźniakowski [38] 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 [35]. The existence of a strongly polynomial algorithm remains wide open; such algorithms are only known for special classes of LPs.

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, 27, 41, 42]. 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 $\min c^\top x$, $Ax \leq b$, $A \in \mathbb{R}^{m\times n}$, and let L denote the total bit-complexity of (A, b, c). Then, van den Brand [41] gives an $O(m^\omega L \log^{O(1)}(m))$ deterministic algorithm, and van den Brand et al. [42] gives a randomized $O((mn + n^3)L \log^{O(1)}(m))$ randomized algorithm for finding exact primal and dual optimal solutions.

Tardos [37] 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 [43], who gave a $\text{poly}(n, m, \log \bar{\chi}_A)$ 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 [19], based on the ellipsoid method.

Vaidya [40] gave a more efficient cutting plane algorithm in the oracle setting; see [1, 2, 23, 28] 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 [23] (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 [40] and is asymptotically optimal [31]. Moreover, [23] 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 [18, 19], Frank and Tardos [17] showed that, assuming that the matrix A and vector b describing the system $Ax \leq b$ are integral with the absolute values of the entries bounded by B, then linear optimization in the oracle model can be solved in time $\text{poly}(n, \log B)$. This is independent of the encoding length of the cost function c. The result is achieved by rounding c using simultaneous Diophantine approximation.

Strongly polynomial algorithms for solving LPs in strongly polynomial time for many important problems such as submodular function minimization or minimum-cost matchings were given in [19] and [17]. 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; this is proved in Appendix A.

[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.

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 $\bar{x}$.*

1.1 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 [17, 18, 19]. 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 1.2. *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 [16], 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, $\log(1/\delta_M)$ is polynomially bounded by n and the sizes of numbers. The quantity δ_M was also studied in the context of lattice basis reduction by Seysen [34]. A related condition number appears in the characterization of Hoffman constants [20, 25, 32].

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 $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, we consider the matrix

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

corresponding to the conic embedding of P defined by the cone $K = \{(x \mid t) \in \mathbb{R}^{n+1} : M(x \mid t) \geq 0\}$. 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 $a_j^\top x^* = b_j$ 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 [23], replacing r/ε by $1/\delta_{(A,b)+(\mathbf{0},1)}$, and an additional $O(n^5 \log\log(1/\delta_{(A,b)+(\mathbf{0},1)}))$ term.

Theorem 1.3. *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.

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 1.3 refers to the JLSW algorithm [23]. In Section 6, 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, 21].
- Assuming P is given by a polyhedral separation oracle, our result strengthens that by Frank and Tardos [17]: 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), the maximum subdeterminant Δ_M is bounded as $\Delta_M \leq B^n n^{n/2}$ by the Hadamard–inequality, and we have $\delta_M \geq 1/(n\Delta_M^2) \geq 1/(B^{2n} n^{n+1})$. 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 [17, 18, 19] 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 require the feasible region to be enclosed in a ball of known radius. In the rational setting, the enclosing radius is 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 [43] 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. the proof is given in Appendix A.

Proposition 1.4. *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. $Ax \leq b$ 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 1.3 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 1.3 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 [19]. 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 Section 7.

Reduction to the conic setting The algorithms in Theorem 1.3 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 $v^\top \bar{x} \leq 0$ and $v^\top x \geq 0$, $\forall x \in K$. We do not assume that K is closed, non-empty or even polyhedral in this definition. We note that the requirement $v^\top \bar{x} \leq 0$ is automatically satisfied by any separator if $K \neq \emptyset$ (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.

[2] We note that we do not generalize [17] for arbitrary oracle settings. The result in [17] is a preprocessing step replacing c by an equivalent $\tilde{c}$ of small encoding length, but does not require any assumptions on the oracle.

We present black box algorithms using the following subroutine:

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

$$\left\| \sum_{j \in J} \lambda_j m_j \right\| < \varepsilon, \qquad \sum_{j \in J} \lambda_j \|m_j\| \geq 1. \tag{2}$$

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 (2) is bounded by a function $\tau(n)$. In Section 6 we show the following.

Theorem 1.5. *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 1.7. 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 $\operatorname{supp}(y^{(d)}) = I$. Find

$$\min \left\{ \frac{c^\top x}{d^\top x} : x \in K,\, d^\top x > 0 \right\}. \tag{3}$$

  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 (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 (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 1.6. *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 $i \notin I$.*

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 1.7. *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 $d = M^\top y^{(d)}$, $I = \text{supp}(y^{(d)})$, $|I| \leq n$, 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 1.3. 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), we can use the polyhedral separation oracle for P and $\text{rec}(P)$ to implement all required oracles (with $d = m_1^\top$) in Theorem 1.7. 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 1.7 in Section 2.1. The proof uses this conic embedding and combines Theorems 1.5 and 1.7.

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* [19], 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:

The encoding sizes of the vectors returned by the strong separation oracle are polynomially bounded by the facet complexity φ. (4)

Under this assumption, one can find a *optimal dual solutions with oracle inequalities* [19, Lemma 6.5.15]. This assumption is not without loss of generality; we discuss this and different concepts of dual solutions in Section 7. In Section 7.2, we sketch how one can still recover the results of [19] from our approach in case (4) does not hold.]

For finding the optimal dual solutions, [19] 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.

Under the same assumption (4), Theorem 1.7 enables a much simpler and more efficient algorithm. Even though Theorem 1.7 requires a polyhedral separation oracle, in Section 7.1 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. Lemma 7.2 shows that $\delta_M \geq 1/2^{O(n^3\varphi)}$ for the associated conic system.Compared to the general framework in [19], this method has the following advantages in the setting of (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 [19] on a higher degree polynomial of φ; our running time depends linearly on $\log(1/\delta_M)$.
- The algorithms in [19] 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 [19] 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. [36] 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 [19]. 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 [19] and Theorem 1.7 are black-box methods. However, [19] requires a seemingly weaker 'primal-only' subroutine, whereas Theorem 1.7 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 1.5 is based on the JLSW [23] cutting plane method. In Section 6, 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 [19] 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(Section 6.3).
- Volumetric cutting plane methods [23, 28, 40] can be used to provide an algorithm for Approx-Conic-Dual with $\mathcal{T}_o(n,\varepsilon) = O(n\log(1/\varepsilon))$ oracle calls(Section 6.4). We note that [28] 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 6.2.
- The geometric rescaling algorithm [12, 21] 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(Section 6.5).

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 6 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.

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 [30] 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

$$\min c^\top x, \quad Ax = b, x \geq 0\,, \tag{5}$$

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

$$\max b^\top y, \quad A^\top y \leq c\,. \tag{6}$$

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

Using the JLSW algorithm [23] to implement the approximate oracle for (6), Theorem 1.3 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 (5) (that is, the dual of (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 $i, j \in \operatorname{supp}(g)$. 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 [37] on combinatorial linear programs, gave an algorithm with running time $\operatorname{poly}(n, m, \log(\kappa_A + n))$ for solving linear programs of the form (5).

The condition number κ_A is within a factor $1/n$ from the Dikin–Stewart–Todd condition number $\bar{\chi}_A$ used in [43], 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 [41] 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. [42], 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 reconciled in Section 8. In Lemma 8.2 and Corollary 8.3 we show that for a matrix of the form $A=(I_m|A')$, $\log(n/\delta_{A^\top})=\Theta(\log(\kappa_A+n))$. We can therefore use our conic validity algorithm in Theorem 1.7 to find a feasible solution to (5) in $\mathrm{poly}(n,m,\log(\kappa_A+n))$ time. In particular, using the JLSW algorithm [23] 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 (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 $\mathrm{poly}(n,m,\log(\kappa_A+n))$ 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.

Finally, in Section 8.1, we also show that for an explicitly given matrix $A\in\mathbb{R}^{m\times n}$, one can find a nonsingular $T\in\mathbb{R}^{m\times m}$ that approximately maximizes $\delta_{(TA)^\top}$, based on the result of Dadush et al. [10] on optimal column rescaling κ_A^* of κ_A.

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 4.4, 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 [22] improved the complexity bounds of minimizing convex functions over integers. This is achieved by a more direct application of lattice basis reduction than in [19]. 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 [26] 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 [26] 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 [15], 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 part of the input, some variants, including those described in [12, 21], can be naturally extended to the oracle setting. We implement the approximate conic oracle in Section 6.5 for these variants.

Theorem 1.7 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 [32]. 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 Sections 3, 4, and 5 describe the algorithms for the strong conic feasibility, conic validity, and minimum conic ratio problems, along with their analyses. Sections 6.1 and 6.2 describe our general approach to finding approximate Farkas certificates from certified quadratic forms. Sections 6.3–6.5 describe different implementations of the oracle Approx-Conic-Dual, based on different methods (ellipsoid, volumetric cutting plane, and geometric rescaling). Section 7 relates the results presented to the classical framework of rational polyhedra. Finally, Section 8 shows the connection between the condition number δ and the circuit imbalance measure. Appendix A includes the proofs of the impossibility results of Propositions 1.1 and 1.4.

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 $\text{span}(v_j : j \in J) \subseteq \mathbb{R}^n$ the linear subspace they span; for a matrix $B \subseteq \mathbb{R}^{n \times m}$, let $\text{span}(B) \subseteq \mathbb{R}^n$ 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 $\text{cone}(v_1, \ldots, v_k) := \{\sum_{i=1}^k \lambda_i v_i : \lambda_1, \ldots, \lambda_k \geq 0\}$ denote the *closed* cone generated 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 $\|v\|_Q \stackrel{\text{def}}{=} \sqrt{v^\top Q v}$; 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 $\|\cdot\|_2$.

Except for Section 7 on rational polyhedra, weuse the real model of computation, allowing basic arithmetic operations $+$, $-$, $\times$, $/$ 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 stepsfor the ellipsoid method in Section 6.3.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 1.3 and Theorem 1.7, 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 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.

In Section 7 we show that for rational polyhedra satisfying assumption (4) on bounded bit-complexity, a strong separation oracle can be converted into a polyhedral separation oracle.

2.1 Reducing LP to the conic setting: proof of Theorem 1.3

We now give the proof of Theorem 1.3 using Theorems 1.5 and 1.7. 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 $\text{rec}(P) = \{x \in \mathbb{R}^n : -Ax \geq 0\}$, both of which are given by a polyhedral separation oracle.

We derive parts *(i), (ii), (iii)* of Theorem 1.7 using the standard homogenization of P into $\mathbb{R}^{n+1}$. Namely, we examine

$$K = \{(x \mid t) \in \mathbb{R}^{n+1} : tb - Ax \geq 0, t \geq 0\} := \{(x \mid t) \in \mathbb{R}^{n+1} : M(x \mid t) \geq 0\}, \tag{7}$$

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

(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 1.7 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 1.7 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 1.7 on the cone $K = \text{rec}(P) := \{x \in \mathbb{R}^n : -Ax \geq 0\}$ 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 $\text{rec}(P)$ 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 1.7. 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 $K_I^= = \{(x \mid t) \in K : d^\top x = 0\} = \{(x \mid 0) \in K\} := (\text{rec}(P) \mid 0)$. As explained in the previous paragraphs, these separators can be directly constructed from the corresponding polyhedral separators for P and $\text{rec}(P)$. 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 1.5 with the corresponding part of Theorem 1.7.

2.2 Properties of the δ-measure

We start by showing that the δ-measure introduced in Definition 1.2 is equivalent to the definition of the "δ-distance property" studied in [6, 9, 16], 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 $v \in V \setminus \text{span}(W)$, the Euclidean distance between v and $\text{span}(W)$ is at least $\delta_V \|v\|$. Further, $\delta_V > 0$ if and only if $|\{v/\|v\| : v \in V\}|$ is finite.*

Proof. For the first part, let δ'_V be the quantity in the statement; we prove $\delta'_V = \delta_V$. We can assume $0 \notin V$, and that $\|v_j\| = 1$ for all $j \in V$, noting that both quantities δ_V and δ'_V are invariant under rescaling vectors in V. We first show that $\delta_V \geq \delta'_V$. Let $\{v_j : j \in I\} \subseteq V$ be a linearly independent set of vectors; for any vector v_i let z_i be the projection of v_i to $\text{span}(v_j : j \in I \setminus \{i\})^\perp$. By definition, $\|z_i\| \geq \delta'_V \|v_i\| = \delta'_V$. It follows that, for every $\lambda \in \mathbb{R}^I$, $\left\|\sum_{k \in I} \lambda_k v_k\right\| \geq |\lambda_i| \|z_i\| \geq \delta'_V |\lambda_i|$. This shows $\delta_V \geq \delta'_V$.

For the direction $\delta'_V \geq \delta_V$, let $W \subseteq V$ and $v \in V \setminus \text{span}(W)$. W.l.o.g., we can assume that the vectors in W are linearly independent, hence the vectors in $W \cup \{v\}$ are linearly independent. Denoting by z be the projection of v to $\text{span}(W)^\perp$, it follows that there exist a unique vector

$\lambda \in \mathbb{R}^{|W|}$ such that $W\lambda = v - z$. Since the vectors in $W \cup \{v\}$ are linearly independent, it follows from the definition of δ_V that $\|z\| = \|W\lambda - v\| \geq \delta_V \|v\|$, showing that $\delta'_V \geq \delta_V$.

If $|V|$ is finite, then δ'_V is defined as the minimum of a finite number of positive numbers, showing $\delta_V = \delta'_V > 0$. If $|V|$ is infinite, then there exists a convergent sequence in V (recalling that all vectors are renormalized to 1 and thus $V \subseteq \mathbb{B}^n(0,1)$); let v_k, $k \in \mathbb{N}$ be such a sequence. Thus, for any $\varepsilon > 0$, there exists $v_j, v_k \in V$ such that $\|v_j - v_k\| < \varepsilon$. Setting $I = \{j, k\}$, $\lambda_j = 1$, $\lambda_k = -1$, we have $\|\sum_{i \in I} \lambda_i v_i\| = \|v_j - v_k\| < \varepsilon$; on the other hand, $|\lambda_j| = |\lambda_k| = 1$. This shows that $\delta_V < \varepsilon$ for every $\varepsilon > 0$. □

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,*

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

We recall the classical Minkowski–Weyl theorem for polyhedral cones.

Theorem 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, the following holds:*

- *$L = \ker(M) = \operatorname{span}(M^\top)^\perp$ is the lineality space of K, $K \cap L^\perp$ is a closed pointed cone and $K = L + (K \cap L^\perp)$.*
- *$K \cap L^\perp = \operatorname{cone}(v_1, \ldots, v_k)$, where $v_1, \ldots, v_k$ are the extreme rays of $K \cap L^\perp$. Furthermore, for each $i \in [k]$, there exists $S \subseteq [m]$, $|S| = \operatorname{rk}(M_S) = \dim(\operatorname{span}(M^\top)) - 1$, and $j \in [m] \setminus S$, such that $v_i = \lambda \Pi^S m_j$ for some $\lambda > 0$ where Π^S denotes the orthogonal projection onto $\ker(M_S)$.*

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 2.5. *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 $m_i^\top v \geq \delta_M \|m_i\| \cdot \|v\|$, $\forall i \in [m]$.*

Proof. Let $v \in K \cap \operatorname{span}(M^\top) \setminus \{0\}$ be an extreme ray and let $i \in [m]$. Then, by Minkowski–Weyl (Theorem 2.4), there exists $S \subseteq [m]$, $|S| = \operatorname{rk}(M_S) = \dim(\operatorname{span}(M^\top)) - 1$ and $j \in [m]$ such that $v = \lambda \Pi^S m_j$, $\lambda > 0$, where Π^S is the orthogonal projection onto $\ker(M_S)$. Since $\Pi^S(\operatorname{span}(M^\top))$ is 1-dimensional and $v^\top m_i \geq 0$, we see that $v^\top m_i = v^\top \Pi^S m_i = \|v\| \cdot \|\Pi^S m_i\|$. If $m_i \in \operatorname{span}(M_S^\top)$, then clearly $\Pi^S m_i = 0$ and thus $v^\top m_i = 0$. If $m_i \notin \operatorname{span}(M_S^\top)$, then $\|\Pi^S m_i\| \geq \delta_M \|m_i\|$ follows by Lemma 2.2. The statement thus follows. □

3 The strong conic feasibility algorithm

In this section, we prove the part of Theorem 1.7 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]; we include the proof for completeness.

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. We initialize $\bar{\lambda} = \lambda$, and maintain $\sum_{j\in J} \bar{\lambda}_j v_j = \sum_{j\in J} \lambda_j v_j$, $c^\top \bar{\lambda} \geq c^\top \lambda > 0$ throughout. At every iteration, either $\mathrm{supp}(\bar{\lambda})$ becomes strictly smaller, or we find a vector μ as in (ii). If we do not end with outcome (ii) at some iteration, then we terminate once $\{v_j : \bar{\lambda}_j > 0\}$ are linearly independent.

If the vectors in the support of $\bar{\lambda}$ are linearly dependent, then let $\mu \neq 0$ be a support-minimal vector such that $\sum_j \mu_j v_j = 0$, $\mathrm{supp}(\mu) \subseteq \mathrm{supp}(\bar{\lambda})$. If $\mu \geq 0$ and $c^\top \mu \geq 0$ or $\mu \leq 0$ and $c^\top \mu \leq 0$, then we we output μ or $-\mu$, respectively, and terminate with outcome (ii). Otherwise, possibly by replacing μ by $-\mu$, we can assume that $c^\top \mu \geq 0$ and μ has a negative component. Let $\alpha > 0$ be the largest number such that $\bar{\lambda} + \alpha\mu \geq 0$, and update $\bar{\lambda} := \bar{\lambda} + \alpha\mu$. Note that $\mathrm{supp}(\bar{\lambda})$ decreases by at least one, and by the choice of μ, we have $\sum_{j\in J} \bar{\lambda}_j v_j = \sum_{j\in J} \lambda_j v_j$, $c^\top \bar{\lambda} \geq c^\top \lambda$.

The running time bound is standard; it takes $O(n^2|J|)$ arithmetic operation to bring the system $\sum_{j\in J} v_j \mu_j = 0$ (in the variables μ_j, $j \in J$) in normal echelon form via Gaussian elimination. At every iteration, if we reduce the support of $\bar{\lambda}$, we remove the vectors corresponding to zero components of $\bar{\lambda}$, and it take $O(n|J|)$ operations per vector removed to bring the system back to echelon normal form, for a total of $O(n|J|^2)$ operations. □

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 $\{m_j : j \in \mathrm{supp}(\lambda)\}$ 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 $J = \mathrm{supp}(\lambda)$. 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 $\mathrm{supp}(\bar{\lambda}) \subseteq J$, $\sum_{j=1}^m \bar{\lambda}_j \|m_j\| \geq 1$ 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

$$\frac{\hat{\delta}}{2n} \geq \left\|M^\top \lambda\right\| = \left\|M^\top \bar{\lambda}\right\| \geq \hat{\delta} \max_{i\in\mathrm{supp}(\bar{\lambda})} \bar{\lambda}_i \|m_i\| \geq \frac{\hat{\delta}}{n}\,, \tag{8}$$

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 1.7. 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 $M_F x = 0$ 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

$$K^F = \{x \in \mathbb{R}^n : M^F x \geq 0\} \quad \text{and} \quad K_1^F = \{x \in \mathbb{R}^n : M^F x \geq 0, (m_1^F)^\top x > 0\}. \tag{9}$$

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, $K^F = (K \cap \ker(M_F)) + \operatorname{span}(M_F^\top)$. 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.*

Proof. For the first part, note that $\bar{x} \in K^F \Leftrightarrow M(\Pi^F \bar{x}) \geq 0 \Leftrightarrow \Pi^F \bar{x} \in K$. The second statement follows directly from the fact that $\Pi^F \bar{x} \in \{x \in \mathbb{R}^n : m_i^\top x = 0, i \in F\}$. For the separation oracle for K_1^F, if $m_1^F = 0$ then we have $K_1^F = \emptyset$; for the rest of the proof, let us assume $m_1^F \neq 0$. Take any $\bar{x} \in \mathbb{R}^n$, and run the conic separation oracle for K_1 for the orthogonal projection $\Pi^F \bar{x}$. If $\Pi^F \bar{x} \in K_1$, we can return $\bar{x} \in K_1^F$. If not, then we obtain an inequality $m_1^\top x > 0$ or $m_i^\top x \geq 0$ for $i > 1$ violated by $\Pi^F \bar{x}$. In the first case this means $m_1^\top (\Pi^F \bar{x}) \leq 0$, meaning $(m^F)^\top x \leq 0$. In the second case, $m_i^\top (\Pi^F \bar{x}) < 0$ for some $i \in T_F$. Note that $\Pi^F m_i \neq 0$ and $(m_i^F)^\top \bar{x} = m_i^\top (\Pi^F \bar{x}) < 0$, hence $(m_i^F)^\top x \geq 0$ is a violated inequality in the system defining K^F. Computing $\Pi^F \bar{x}$ and $\Pi^F m_i$ requires time $O(n^2)$. □

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

Proof. Consider a set $J \subseteq T^F$ such that $\{m_j^F : j \in J\}$ are linearly independent. Consider $\lambda \in \mathbb{R}^J$. There exists $\theta \in \mathbb{R}^F$ such that $\{m_i : i \in \operatorname{supp}(\theta)\}$ is linearly independent, and

$$\left\| \sum_{j \in J} \lambda_j \Pi^F m_j \right\| = \left\| \sum_{j \in J} \lambda_j m_j + M_F^\top \theta \right\| \geq \delta_M \max \left\{ \max_{j \in J} \lambda_j \|m_j\|, \max_{i \in F} \theta_i \|m_i\| \right\} \geq \delta_M \max_{j \in J} \lambda_j \|\Pi^F m_j\|,$$

where the first inequality follows from the definition of δ_M, since the vectors in $\{\Pi m_j : j \in J\} \cup \{m_i : i \in \operatorname{supp}(\theta)\}$ are linearly independent, and the second inequality follows from $\|m_j\| \geq \|\Pi m_j\|$ for all $j \in J$. □

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 $\operatorname{supp}(\mu) \subseteq F \cup \{1\}$ 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 $\Pi^F x \in K$ with $m_1^\top (\Pi^F x) > 0$, 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 $J = \operatorname{supp}(\hat{\lambda})$, and define $F' = F \cup J$ and $\xi' = \hat{\lambda} + \theta + \alpha \xi$. We have that $M^\top \xi' = 0$, $\operatorname{supp}(\xi') = F'$, $\xi' \geq 0$. Furthermore, $\operatorname{rk}(M_{F'}) = \operatorname{rk}(M_F) + \operatorname{rk}(M_J)$ and $|J| \leq \operatorname{rk}(M_J) + 1$ since $\hat{\lambda}$ is support minimal, hence $|F'| \leq 2\operatorname{rk}(M_{F'})$, 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 1.7 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

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

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

$$\frac{\|M^\top \lambda - c\|}{\hat{\delta}} = \frac{\|M^\top \bar{\lambda} - \tau c\|}{\tau \hat{\delta}} \leq \frac{4n\|c\|}{\hat{\delta}^2} \cdot \varepsilon = \frac{\|c\|}{2n}. \tag{10}$$

From the triangle inequality, (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*

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

Proof. Let Π denote the orthogonal projection onto $\ker(M)$. We examine two cases, $\Pi c \neq 0$ or $\Pi c = 0$. If $\Pi c \neq 0$, note that $c^\top(-\Pi c) = -\|\Pi c\|^2 < 0$ and $m_i^\top(-\Pi c) = (\Pi m_i)^\top(-\Pi c) = 0$, $\forall i \in [m]$. In particular, $-\Pi c \in K_c \cap \{x \in \mathbb{R}^n : m_j^\top x = 0,\ j \in J\}$, and thus the desired conclusion holds in this case.

We may now examine the case $\Pi c = 0 \Leftrightarrow c \in \text{span}(M^\top)$. If $K_c = \emptyset$, there is nothing to prove, so we may assume $\exists x \in K_c$. Let $\Pi' := I - \Pi$ denote the orthogonal projection onto $\text{span}(M^\top)$ and let $x' = \Pi' x$. Note that $m_i^\top x' = m_i^\top x \geq 0$, $i \in [m]$, and $c^\top x' = c^\top x < 0$, since $m_i, c \in \text{span}(M^\top)$ by assumption. Thus, $x' \in K \cap \text{span}(M^\top) \cap \{x : c^\top x < 0\} \subseteq K_c$. By Minkowski-Weyl, we can write $x' = \sum_{i=1}^k v_i$, where $v_1, \ldots, v_k$ are extreme rays of $K \cap \text{span}(M^\top)$.

Since $c^\top x' < 0$, we may choose $\ell \in [k]$ such that $c^\top v_\ell < 0$. Let J and $\lambda \in \mathbb{R}^m_+$ as in Lemma 4.1(iii). We claim that $m_j^\top v_\ell = 0$, for $j \in J$. This will prove the lemma, since then $v_\ell \in K_c \cap \{x \in \mathbb{R}^n : m_j^\top x = 0, j \in J\}$. Since $v_\ell \in K_c$, applying the Cauchy–Schwartz inequality we get that

$$0 \leq -c^\top v_\ell + \sum_{j \in [m]} \lambda_j m_j^\top v_\ell \leq \|M^\top \lambda - c\| \cdot \|v_\ell\| \,. \tag{11}$$

By the definition of J, $\alpha := \max_{j \in J} \frac{\|M^\top \lambda - c\|}{\hat{\delta} \lambda_j \|m_j\|} \in [0, 1)$. For $j \in J$, we thus obtain $m_j^\top v_\ell \leq \alpha \hat{\delta} \|v_\ell\| \cdot \|m_\ell\| \leq \alpha \delta_M \|v_\ell\| \cdot \|m_j\|$, $\forall j \in J$. Since v_ℓ is an extreme ray of $K \cap \text{span}(M^\top)$, by Lemma 2.5 we have that $m_j^\top v_\ell > 0 \Rightarrow m_j^\top v_\ell \geq \delta_M \|m_j\| \cdot \|v_\ell\|$. We conclude that $m_j^\top v_\ell = 0$, $\forall j \in J$, as needed. □

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 $|F| \leq 2\text{rk}(M_F)$. 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 4.3. *Let $F \subset F' \subseteq [m]$ such that $J := F' \setminus F \subseteq T_F$, $|J| \leq 2\text{rk}(M_J^F)$. Let $v \in \mathbb{R}^n$ and $y' \in \mathbb{R}^{T_{F'}}$ such that $(M^{F'})^\top y' = v^{F'}$ and $\text{supp}(y') \leq 2\text{rk}(M^{F'})$. Then, in time $O(n^\omega)$ we can compute $y \in \mathbb{R}^{T_F}$ such that $(M^F)^\top y = v^F$, $\text{supp}(\bar{y}) \subseteq T_{F'} \cup J$, $|\text{supp}(y)| \leq 2\text{rk}(M^F)$ and $y_i = y'_i$ for $i \in T_{F'}$.*

Proof. Note that each row $(m_i^{F'})^\top$ of $M^{F'}$, $i \in T_{F'}$, $m_i^{F'}$ is the orthogonal projection of m_i^F onto $\ker(M_J^F)$, and similarly $v^{F'}$ is the orthogonal projection of v^F onto $\ker(M_J^F)$. It follows that there exists a matrix $Q \in \mathbb{R}^{T_{F'} \times J}$ such that $M^{F'} = M^F_{T_{F'}} + Q M_J^F$. Thus, $(M^F_{T_{F'}})^\top y' + (M_J^F)^\top (Q^\top y') = v^F$. In particular, the system $v^F - (M^F_{T_{F'}})^\top y' = (M_J^F)^\top \mu$, $\mu \in \mathbb{R}^J$, has a solution. For any such μ, the vector $y_i = y'_i, i \in T_{F'}$, $y_i = \mu_i$, $i \in J$, satisfies the requirements of the lemma noting that $2(\text{rk}(M_J^F) + \text{rk}(M^{F'})) = 2\text{rk}(M^F)$. To compute μ, we must first compute the left hand side $v^F - (M^F_{T_{F'}})^\top y' = \Pi^F v - \Pi^F M^\top_{T_{F'}} y'$ which requires $O(n^2) = O(n)$ time since $|\text{supp}(y')| \leq 2n$ (we assume $\Pi^F, \Pi^{F'}$ have already been computed). From here, the matrix $(M_J^F)^\top = \Pi^F M_J^\top$ requires $O(n^\omega)$ to compute (noting that $|J| \leq 2n$), and finally solving for μ also requires $O(n^\omega)$ time. □

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:

(a) A vector $y \in \mathbb{R}^{T_F}_+$ with $|\text{supp}(y)| \leq 2\text{rk}(M^F)$, 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 $\{m_j^F : j \in \text{supp}(\lambda)\}$ 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 4.3, 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 4.3, 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. The proof will be given in Section 4.1.1.

Lemma 4.4. *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 4.4 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 4.4. If outcome (*ii*) of Lemma 4.4 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 4.4 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.

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 4.3, 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 4.4, 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 $O(n^5 \log\log(1/\delta_M))$.

4.1.1 The proof of Lemma 4.4

In this section, we prove Lemma 4.4, which is a slight adaptation of [11, Lemma 4.3].

Lemma 4.5 ([11, Lemma 4.3])**.** *Let $B \in \mathbb{R}^{t\times n}$ with row vectors $b_i^\top$, $\ell \in (\mathbb{R} \cup \{-\infty\})^t$, $u \in (\mathbb{R} \cup \{\infty\})^t$, and let $z \in \ker(B^\top)$, $\ell \leq z \leq u$. Then, in $O(t^3 n)$ time, we can compute $r, v \in \ker(B^\top)$, along with a set $S \subseteq [t]$ such that:*

- *$\ell \leq r \leq u$, $\|r\|_\infty \leq \|v\|_\infty$*
- *$S \supseteq \{i \in [t] : l_i = u_i\}$ and $r_i = v_i = \ell_i^+ - u_i^-$ for all $i \in S$.*
- *the vectors in $\{b_i : i \in \mathrm{supp}(v) \setminus S\}$ are linearly independent.*

In [11, Lemma 4.3], the only difference is that we express $r = \sum_{j=1}^k \lambda_j v^j$, where $\lambda \geq 0$ is a convex combination, and $v^1, \ldots, v^k \in \ker(B^\top)$ satisfy the last two bullets. In the above, we simply take $v := v^{j_*}$ where $j_* := \mathrm{argmax}_{j\in[k]}\|v^j\|_\infty$, which clearly satisfies the first bullet since r is a convex combination of $v^1, \ldots, v^j$. The role of the vector v above is to serve as an "algebraic witness" that the norm of r is controlled by the "difficult" bound constraints induced by l^+ and u^- (i.e., those that cut off 0 as a solution).

This lemma was previously used in the following context; recall the definition of the circuit imbalance measure κ_A from the Introduction. According to [11, Theorem 3.1], whenever the system $Ax = 0$, $\ell \leq x \leq u$ is feasible, there exists a feasible solution x with $\|x\|_\infty \leq \kappa_A\|\ell^+ + u^-\|$. Note that the right hand side is 0 if and only if $\ell \leq 0 \leq u$, in which case we can select $x = 0$. Algorithmically, given a feasible solution $Az = 0$, $\ell \leq z \leq u$, Lemma 4.5 enables one to find another feasible $Ar = 0$ with $\|r\|_\infty \leq \kappa_A\|\ell^+ + u^-\|$ in strongly polynomial time. See Section 8 for further discussion of the circuit imbalance measure.

Proof of Lemma 4.4. The statement is invariant under renormalizing the rows of H, so let us assume $\|h_i\| = 1$ for all $i \in [k]$. Let $\tau = \|H^\top(y' - y)\|$, and define $B^\top = (H^\top, \frac{1}{\tau}H^\top(y - y')) \in \mathbb{R}^{n\times(k+1)}$; that is, we add to B the row $(y - y')^\top H$ normalized to 1. Let $z \in \mathbb{R}^{k+1} = (y' - y; \tau)$; clearly, $z \in \ker(B^\top)$.

Define $\ell, u \in \mathbb{R}^{k+1}$ as follows. For $i \in \mathcal{L}_1$, we set $\ell_i = -\infty$, $u_i = \infty$. For $i \in \mathcal{L}_2$, we set $\ell_i = -y_i$ and $u_i = \infty$, and let $\ell_{k+1} = u_{k+1} = \tau$. Note that $\ell \leq z \leq u$ holds due to the assumptions on y and y'. Using Lemma 4.5, we obtain $r, v \in \ker(B^\top)$ and $S \subseteq [k+1]$, such that $k+1 \in S$, $\ell \leq r \leq u$, $\|r\|_\infty \leq \|v\|_\infty$, $r_i = v_i = (\ell_i^+ - u_i^-)$ for all $i \in S$, and the vectors in $\{h_i : i \in \mathrm{supp}(v)\setminus S\}$ are linearly independent. By our assumptions, note that $(\ell^+ - u^-)_i = 0$ for $i \in [k]$ and that $(\ell^+ - u^-)_{k+1} = \tau$. Let $\bar{r} := r_{[k]}$ and $\bar{v} := v_{[k]}$ be the vectors in $\mathbb{R}^k$ obtained by dropping the $(k+1)$'st coordinate τ. Since $r, v \in \ker(B^\top)$, we see that $H^\top\bar{r} = H^\top\bar{v} = H^\top(y' - y)$.

Note that, since $v_i = 0$ for $i \in S \cap [k]$, we have $\mathrm{supp}(\bar{v}) \subseteq [k] \setminus S$, hence $\{h_i : i \in \mathrm{supp}(\bar{v})\}$ are linearly independent. If $\|H^\top \bar{v}\| < \hat{\delta}\|\bar{v}\|_\infty$, then we output $\lambda = \bar{v}$ and stop with outcome (ii). Assume that $\|H^\top \bar{v}\| \geq \hat{\delta}\|\bar{v}\|_\infty$. We next observe that

$$\|\bar{r}\|_\infty \leq \tau/\hat{\delta}. \tag{12}$$

Indeed, $\|\bar{v}\|_\infty \leq \left\|H^\top \bar{v}\right\|/\hat{\delta} = \left\|H^\top (y' - y)\right\|/\hat{\delta} = \tau/\hat{\delta}$, hence $\|\bar{r}\|_\infty \leq \|r\|_\infty \leq \|v\|_\infty = \max\{\|\bar{v}\|_\infty, \tau\} \leq \tau/\hat{\delta}$, since $\hat{\delta} \leq 1$.

We claim that the vector $q := y + \bar{r}$ satisfies $H^\top q = H^\top y'$, $q \geq 0$. Recalling that $H^\top \bar{r} = H^\top (y' - y)$, we have $H^\top q = H^\top y'$. For $i \in \mathcal{L}_2$, $q_i = y_i + r_i \geq y_i + \ell_i = 0$. For $i \in \mathcal{L}_1$,

$$q_i = y_i + \bar{r}_i \geq y_i - \|\bar{r}\|_\infty \geq y_i - \frac{\tau}{\hat{\delta}} \geq 0\,,$$

where the last inequality follows by (12) and the assumptions $y_i\|h_i\| \geq \tau/\hat{\delta}$, $\|h_i\| = 1$.

Finally, in time $O(k^2 n + n^2 k)$ we can turn q into a basic solution of $H^\top q = H^\top y'$, $q \geq 0$, ensuring $|\mathrm{supp}(q)| \leq \mathrm{rk}(H)$. □

5 The conic minimum-ratio algorithm

Recall the conic minimum-ratio 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 via a conic separation oracle, and vectors $c, d \in \mathbb{R}^n$, $y^{(d)} \in \mathbb{R}^m_+$ such that $M^\top y^{(d)} = d$, $I = \mathrm{supp}(y^{(d)})$, $|I| \leq n$. The goal is to find the minimum value $\gamma^* \in \mathbb{R}$ such that $(c + \gamma^* d)^\top x \geq 0$ for all $x \in K$, or conclude that the problem is infeasible or unbounded. We assume polyhedral separation oracles are available for the cones

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

Further, provide certificates as described in Section 1.1. For an index set $F \subseteq [m]$, we will use the same notation Π^F, v^F, T_F, M^F, and K^F as in Sections 3 and 4.

Detecting infeasibility and unboundedness First, let us decide if problem (3) is infeasible, i.e. if $d^\top x = 0$ for all $x \in K$. The combination $y^{(d)}$ already certifies that $d^\top x \geq 0$ for all $x \in K$. Let us run the conic validity algorithm for the cost vector $-d$, using the oracle for K_{-d}. Then, we either find $\bar{x} \in K$ such that $d^\top \bar{x} > 0$, or a certificate $y \in \mathbb{R}^m_+$ such that $M^\top y = -d$.

Next, let us check if problem (3) is unbounded. We need to verify if there exists an $x \in K$ with $c^\top x < 0$ and $d^\top x = 0$. According to the combination $y^{(d)} \in \mathbb{R}^m_+$, $M^\top y^{(d)} = d$, $I = \mathrm{supp}(y^{(d)})$, we have $d^\top x = 0$ if and only if $x \in K^=_I$. We thus run the conic validity algorithm for $K^=_I$ and the projection c^I. (The required oracle for $(K^=_I)_{c^I}$ can be implemented using the oracle assumed for $K^=_I$ and checking the additional inequality $(c^I)^\top x < 0$.) If the original problem is unbounded, then we find an $x \in K^=_I$ with $(c^F)^\top x < 0$. If we establish that $c^\top x \geq 0$ for all $x \in K^=_I$ then we conclude that (3) is not unbounded.

The optimal set and the main subroutine After the above preprocessing, from now on we assume that

$$K_{-d} \neq \emptyset \text{ and } c^\top x \geq 0 \text{ for all } x \in K^=_I. \tag{13}$$

Hence, a finite optimal γ^* exists, in which case $(c + \gamma^* d)^\top x \geq 0$ is valid for K and there exists an $x^* \in K$ with $(c + \gamma^* d)^\top x^* = 0$, $d^\top x^* > 0$. Let

$$K_0 = \left\{x : Mx \geq 0\,,\, (c + \gamma^* d)^\top x = 0\,,\, d^\top x > 0\right\}$$

denote the set of optimal solutions. By the above assumptions, $K_0 \neq \emptyset$. Note that if $c = -\gamma d$ for some $\gamma \in \mathbb{R}$, then $c^\top x / d^\top x = -\gamma$ for every $x \in K_{-d}$. Hence, $K_0 = K_{-d}$ in this case.

The purpose of the next lemma, which is an analogue of Lemma 4.1 for the set K_0, is to identify inequalities of $Mx \geq 0$ that can be set at equality for K_0.

Lemma 5.1. *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 $c, d \in \mathbb{R}^n$ such that (13) holds, and c and d are linearly independent. Furthermore, let $y^{(d)} \in \mathbb{R}^m_+$ such that $M^\top y^{(d)} = d$. 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) a nonzero vector $\lambda \in \mathbb{R}^m_+$ that is a minimum support solution to $M^\top \lambda = 0$;

(ii) a vector $\lambda \in \mathbb{R}^m_+$ such that $\{m_j : j \in \operatorname{supp}(\lambda)\}$ are linearly independent, a point $\bar{x} \in K$, such that $d^\top \bar{x} > 0$, along with a nonempty subset $J \subseteq [m]$ such that, for $\gamma = -c^\top \bar{x}/d^\top \bar{x}$, every $j \in J$ satisfies $\lambda_j \|m_j\| > \|M^\top \lambda - (c + \gamma d)\|/\hat{\delta}$;

(iii) a certificate of $\hat{\delta} > \delta_M$, namely, a vector $\lambda \in \mathbb{R}^m_+$ such that $\{m_j : j \in \operatorname{supp}(\lambda)\}$ are linearly independent and $\varphi = \|M\lambda\|/\max_{j\in[m]} \lambda_j \|m_j\| < \hat{\delta}$.

Proof. Let $\hat{K}$ be the cone implicitly defined by the following conic separation oracle. We can obtain the following separation oracle based on the one provided for K_{-d}. For every $\bar{x} \in \mathbb{R}^n$, the oracle returns one of the following:

(i) If $\bar{x} \notin K_{-d}$, then return a violated inequality $m_i^\top x \geq 0$ or $d^\top x > 0$.

(ii) If $\bar{x} \in K_{-d}$, then set $\bar{\gamma} = -c^\top \bar{x}/d^\top \bar{x}$, and return the inequality $(c + \bar{\gamma} d)^\top x < 0$.

Observe that, by definition of the oracle, $\hat{K} = \emptyset$, since every point in $\mathbb{R}^n$ is separated. By the assumption that c and d are linearly independent, $c + \bar{\gamma} d \neq 0$. For every inequality returned in case (ii), we have $\bar{\gamma} \leq \gamma^*$, since $(c + \bar{\gamma} d)^\top \bar{x} = 0 \leq (c + \gamma^* d)^\top \bar{x}$, and $d^\top \bar{x} > 0$.

We call the subroutine Approx-Conic-Dual with $\varepsilon = \hat{\delta}^2/(8n^2)$ using the above separation oracle. Since $\hat{K} = \emptyset$, Approx-Conic-Dual always terminates with a ε-certificate consisting of inequalities returned by the oracle. The combination may include vectors corresponding to the inequalities of $Mx \geq 0$, $d^\top x \geq 0$, and inequalities $-(c + \gamma_t d)^\top x \geq 0$, $t \in S$, for different values of γ_t. The certificate is a vector of the form $(\bar{\lambda}, \bar{\tau}, \bar{\mu}) \in \mathbb{R}^m_+ \times \mathbb{R}^S_+ \times \mathbb{R}_+$ such that

$$\left\| M^\top \bar{\lambda} - \sum_{t\in S} \bar{\tau}_t (c + \gamma_t d) + \bar{\mu} d \right\| \leq \varepsilon\,, \quad \sum_{j=1}^m \bar{\lambda}_j \|m_j\| + \sum_{t\in S} \bar{\tau}_t \|c + \gamma_t d\| + \bar{\mu}\|d\| \geq 1\,.$$

Recall that $d = M^\top y^{(d)}$ for a given vector $y^{(d)} \in \mathbb{R}^k_+$. Let us define $(\lambda, \tau) \in \mathbb{R}^m_+ \times \mathbb{R}_+$ as follows. If $S = \emptyset$, then we let $\lambda = \bar{\lambda} + \bar{\mu} y^{(d)}$ and $\tau = 0$. If $S \neq \emptyset$, let $\gamma = \max_{t\in S} \gamma_t$. Define

$$\lambda = \bar{\lambda} + \left(\bar{\mu} + \sum_{t\in S} \bar{\tau}_t (\gamma - \gamma_t) \right) y^{(d)}\,, \quad \text{and} \quad \tau = \sum_{t\in S} \bar{\tau}_t\,.$$

In both cases, $M^\top \lambda - \tau(c + \gamma d) = M^\top \bar{\lambda} - \sum_{t\in S} \bar{\tau}_t (c + \gamma_t d) + \bar{\mu} d$. Furthermore,

$$\sum_{j=1}^m \lambda_j \|m_j\| + \tau \|c + \gamma d\| \geq \sum_{j=1}^m \bar{\lambda}_j \|m_j\| + \sum_{t\in S} \bar{\tau}_t \|c + \gamma_t d\| + \bar{\mu}\|d\| \geq 1$$

by the triangle inequality. Note that during the algorithm the inequality $-(c + \gamma d)^\top x \geq 0$ was returned to separate a point $\bar{x} \in K$ such that $d^\top \bar{x} > 0$ and $\gamma = -c^\top \bar{x}/(d^\top \bar{x})$. The rest of the proof is now identical to the proof of Lemma 4.1, applied for the vector $c + \gamma d$ in place of c. □

Lemma 5.2. *In Lemma 5.1, if outcome* (ii) *occurs and $\hat{\delta} \leq \delta_M$, then*

$$\emptyset \neq K_0 \subseteq \{x \in \mathbb{R}^n : m_j^\top x = 0,\, j \in J\}.$$

Proof. Let $\bar{x} \in K$, γ, J and $\lambda \in \mathbb{R}^m_+$ as in Lemma 4.1(iii). Recall that $K_0 \neq \emptyset$ by assumption (13). Let K' be the face of K defined by the valid inequality $(c + \gamma^* d)^\top x \geq 0$, so that $K_0 = \{x \in K' : d^\top x > 0\}$. We will show that $K' \subseteq \{x \in \mathbb{R}^n : m_j^\top x = 0,\, j \in J\}$, which clearly suffices. By Minkowski-Weyl, $K' = \ker(M) + \operatorname{cone}(v_1, \ldots, v_q)$, where $v_1, \ldots, v_q$ are the extreme rays of $K' \cap \operatorname{span}(M^\top)$. To verify the desired containment, it clearly suffices to show that $m_j^\top v_i = 0, \forall i \in [q], j \in J$. Since $K' \cap \operatorname{span}(M^\top)$ is a face of $K \cap \operatorname{span}(M^\top)$, we note that $v_1, \ldots, v_q$ are also extreme rays of $K \cap \operatorname{span}(M^\top)$. Therefore, by Lemma 2.5, for $j \in J, i \in [q]$, $m_j^\top v_i > 0 \Rightarrow m_j^\top v_i \geq \delta_M \|m_j\| \|v_i\|$.

On the other hand, for any $z \in K'$, by the Cauchy–Schwartz inequality we get that

$$0 \leq -(c + \gamma d)^\top z + \sum_{j\in[m]} \lambda_j m_j^\top z \leq \|M^\top \lambda - (c + \gamma d)\| \cdot \|z\|\,.$$

By the definition of J, $\alpha := \max_{j\in J} \frac{\|M^\top \lambda - (c+\gamma d)\|}{\hat{\delta} \lambda_j \|m_j\|} \in [0, 1)$. For $j \in J$, we thus obtain $m_j^\top z \leq \alpha \hat{\delta} \|z\| \cdot \|m_j\| \leq \alpha \delta_M \|z\| \cdot \|m_j\|$ for every $z \in K'$. Applying the above to $z = v_i$, $i \in [q]$, it follows that $m_j^\top v_i = 0$, $\forall j \in J$. □

The recursive algorithm The algorithm follows the same lines as those in Section 3.1 for strong conic feasibility and in Section 4.1 for conic validity. We assume that (13) holds. For $F \subseteq [m]$ we use the same notation M^F, Π^F, T_F, K^F and c^F; further, let $d^F = \Pi^F d$.

As in the previous algorithms, we will maintain a set of inequalities $F \subseteq [m]$, $|F| \leq 2\text{rk}(M_F)$ and an estimate $\hat{\delta} \in (0,1)$ of δ_M. We will defined an algorithm RECURSIVE-MIN-RATIO$(K, c, d, F, \hat{\delta})$ which outputs either of the following:

(a) A point $x^* \in K$ such that $d^\top x^* > 0$ along with the value $\gamma^* := -c^\top x^*/d^\top x^*$ and a certificate $y \in \mathbb{R}_+^{T^F}$ with $|\text{supp}(y)| = O(n)$ such that $(M^F)^\top y = c^F + \gamma^* d^F$.

(b) Claim a failure for $\hat{\delta}$.

Our algorithm is the following: initialize $\hat{\delta} := 1/n$ and call RECURSIVE-MIN-RATIO$(K, c, d, \emptyset, \hat{\delta})$. If outcome (a) occurs, then γ^* is the optimal value, x^* and optimal solution, and $y \in \mathbb{R}^m$ a dual certificate of optimality. If outcome (b) occurs, then update $\hat{\delta} := \hat{\delta}^2$ and repeat. (For simplicity, in the description we give below, we do not compute an explicit combination of linearly independent rows of M certifying a failure, as we had done in the previous sections. Hence, we do not have an explicit upper bound $\delta_M \leq \varphi < \hat{\delta}$. This does not make a difference in term of the worst-case running time.)

We now describe RECURSIVE-MIN-RATIO$(K, c, d, F, \hat{\delta})$. If $d^F = 0$, then we declare a failure, i.e., outcome (b). This is because according to (13), $K_0 \neq 0$, and $d^\top x > 0$ for all $x \in K_0$. Thus, $K_0 \subseteq K^F$ is not possible, since it would imply $d^\top x = (d^F)^\top x = 0$ for all $x \in K_0$.

If $d^F \neq 0$, but $c^F = -\gamma d^F$ for some $\gamma \in \mathbb{R}$, then we run the conic validity problem for K^F and $-d^F$ (Section 4). If this returns a feasible solution $\bar{x}$, then we return outcome (a) with $\bar{x}$, γ, and the trivial dual certificate $y = 0$. If this returns infeasibility, then we again declare the failure outcome (b) as above.

Finally, if $d^F \neq 0$ and c^F and d^F are linearly independent, then we call the subroutine in Lemma 5.1 for the lower dimensional problem K^F, c^F, and d^F, and consider the possible outcomes:

(i) From Lemma 5.1 we obtain a nonzero $\lambda \in \mathbb{R}_+^{T_F}$ that is a minimal support solution to $(M^F)^\top \lambda = 0$. Let $J = \text{supp}(\lambda)$ and $F' = F \cup J$, and call RECURSIVE-MIN-RATIO$(K, c, d, F', \hat{\delta})$. If this call returns $x^* \in K$ such that $d^\top x^* > 0$, $\gamma^* = c^\top x^*/d^\top x^*$, and $y \in \mathbb{R}_+^{T_{F'}}$ such that $(M^{F'})^\top y = c^{F'} + \gamma^* d^{F'}$ (outcome (a)), then we compute $y' \in \mathbb{R}^{T_F}$ as in Claim 4.3, and return x^* and $\bar{y} := y' + \alpha\lambda \geq 0$, for some α large enough, satisfying $(M^F)^\top \bar{y} = c^F + \gamma^* d^F$. If the recursive call returns a failure (outcome (b)), then we return a failure.

(ii) From Lemma 5.1 we obtain $\lambda \in \mathbb{R}_+^{T_F}$ such that $\{m_j : j \in \text{supp}(\lambda)\}$ are linearly independent, $\bar{x} \in K^F$ with $(d^F)^\top \bar{x} > 0$, $\gamma = -(c^F)^\top \bar{x}/(d^F)^\top \bar{x}$, and a nonempty $J \subseteq \text{supp}(\lambda)$ such that $\lambda_j \|m_j^F\| > \|(M^F)^\top \lambda - (c^F + \gamma d^F)\|/\delta_M$ for all $j \in J$. If $(M^F)^\top \lambda = c^F + \gamma d^F$, then return outcome (a), along $x^* = \Pi^F \bar{x} \in K$, $\gamma^* = \gamma$, and $y = \lambda$. Otherwise, we set $F' = F \cup J$, and call RECURSIVE-MIN-RATIO$(K, c, d, F', \hat{\delta})$. Indeed, recall that, if $\hat{\delta} \leq \delta_M$, then by Lemma 5.2 $\emptyset \neq K_0 \subset \ker(M_{F'})$. It this call returns a failure (outcome (b)), then we return a failure.

Consider the case where RECURSIVE-MIN-RATIO$(K, c, d, F', \hat{\delta})$ returns $x^* \in K$ such that $d^\top x^* > 0$, $\gamma^* = -c^\top x^*/d^\top x^*$, and $\bar{y} \in \mathbb{R}_+^{T_{F'}}$ with $|\text{supp}(\bar{y})| = O(n)$ such that $(M^{F'})^\top \bar{y} = c^{F'} + \gamma^* d^{F'}$ (outcome (a)). If $\gamma > \gamma^*$, then we return a failure. Assume $\gamma^* \geq \gamma$. By Claim 4.3, we can compute $y' \in \mathbb{R}_+^{T_F}$ such that $(M^F)^\top y' = c^F + \gamma^* d^F$; $\text{supp}(y') \subseteq T_{F'} \cup J$ and $y'_i = \bar{y}_i$ for all $i \in T_{F'}$, where possibly $y_i < 0$ for some $i \in J$.

Recall that the input included a vector $y^{(d)} \in \mathbb{R}_+^k$ such that $M^\top y^{(d)} = d$; observe that the restriction $\tilde{y} = y^{(d)}_{T^F}$ satisfies $(M^F)^\top \tilde{y} = d^F$. Define $y = \lambda + (\gamma^* - \gamma)\tilde{y}$. Observe that, by construction, $(M^F)^\top y - (c^F + \gamma^* d^F) = (M^F)^\top \lambda - (c^F + \gamma d^F)$ and $y \geq \lambda$. It follows that we can apply Lemma 4.4 to $H = M^F$, y and y'. The outcome will be either a failure for $\hat{\delta}$, in which case we return a failure, or a vector $q \in \mathbb{R}_+^{T_F}$ such that $(M^F)^\top q = c^F + \gamma^* d^F$. We output x^*, γ^*, and the certificate $q \in \mathbb{R}_+^{T^F}$.

(iii) From Lemma 5.1 we obtain a failure for $\hat{\delta}$, in which case we return a failure.

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-MIN-RATIO$(K, c, d, \emptyset, \hat{\delta})$ will make at most $\text{rk}(M) \leq n$ recursive calls of the form RECURSIVE-MIN-RATIO$(K, c, d, F, \hat{\delta})$. Furthermore, if $\hat{\delta} \leq \delta_M$, then RECURSIVE-MIN-RATIO$(K, c, d, \emptyset, \hat{\delta})$ will not detect any failure, and so it will terminate with one of the two desired outcomes.

Running time analysis The running time analysis is identical to the one in Section 4.1, so we omit it.

Remark 5.3. The above recursive algorithm can be modified to explicitly compute a "certified failure" for the current $\hat{\delta}$, instead of simply declaring that $\hat{\delta} > \delta_M$. There are three places in the above recursive procedure where a failure is detected. One is when $d^F = 0$, the second when d^F and c^F are linearly independent, and the third when $\gamma < \gamma^*$ in outcome (ii). In the first two cases, we proceed as in the conic-validity algorithm to repeatedly apply Lemma 4.4 to pull-back certificates of the form $(M^F)^\top \lambda = d^F$, $\lambda \geq 0$. Since we cannot recover a certificate $M^\top \lambda = d$, $\lambda \geq 0$, at some point Lemma 4.4 must compute a failure. If $\gamma < \gamma^*$ in outcome (ii), then we proceed as in the above algorithm to repeatedly apply Lemma 4.4 to pull-back the certificate $(M^{F'})^\top \bar{y} = c^{F'} + \gamma^* d^{F'}$ to a certificate $M^\top y = c + \gamma^* d$, $y \geq 0$. Since such a certificate does not exist, because γ^* is not optimal, it follows that at some point Lemma 4.4 must compute a failure.

6 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 6.1. *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 [28] (LSW algorithm) explicitly provides dual certificates; the proof follows easily using Theorem 31 in the paper.

Theorem 6.2. *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

$$\operatorname{width}_K(v) \stackrel{\text{def}}{=} \max\{v^\top z \,:\, z \in K\} - \min\{v^\top x \,:\, z \in K\}. \tag{14}$$

We say that v is an *ε-thin direction for K* if $\operatorname{width}_K(v) \leq \varepsilon$. 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 6.3. *Given $R \in \mathbb{S}^n_{++}$, and $p \in \mathbb{R}^n$, let $E := E(R, p)$. For any $v \in \mathbb{R}^d$, $\operatorname{width}_E(v) = 2\|v\|_{R^{-1}}$. In particular, for $K = E(R, p)$, v is an ε-thin direction if and only if $\|v\|_{R^{-1}} \leq \varepsilon/2$.*

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 [40], as well as the geometric rescaling algorithms [12, 21]—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 6.4 ([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 $a_1, \ldots, a_m \in \mathbb{R}^n$, and $\gamma_0, \ldots, \gamma_m \geq 0$, with $\gamma_0 \leq 1$. Then, for every $i \in [t]$, $\gamma_i \|a_i\|^2_{R^{-1}} < 1$ holds, and $\sum_{i=1}^{t} \gamma_i \|a_i\|^2_{R^{-1}} \leq n$. Further, if $\det(R) > 1$, then there exists $k \in [t]$ such that

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

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

$$\begin{aligned} \sum_{i=1}^{m} \gamma_i \|a_i\|^2_Q = \sum_{i=1}^{m} \gamma_i \left(a_i^\top Q a_i \right) &= \operatorname{tr}\left(Q \sum_{i=1}^{m} \gamma_i a_i a_i^\top \right) \\ &= \operatorname{tr}(Q(R - \gamma_0 I_n)) = \operatorname{tr}(I_n) - \gamma_0 \operatorname{tr}(Q) < n . \end{aligned} \tag{15}$$

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

For the third claim, we see that $\operatorname{tr}(R) = \gamma_0 n + \sum_{i=1}^{m} \gamma_i \|a_i\|^2_2$ since $\|a_i\|_2 \leq 2$. Noting that $\gamma_0 \leq 1$, $\sum_{i=1}^{m} \gamma_i \|a_i\|^2_2 \geq \operatorname{tr}(R) - n \geq n(\det(R)^{1/n} - 1)$, using the well-known inequality $\det(R)^{1/n} \leq \operatorname{tr}(R)/n$ 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

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

□

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 $a_i^\top x \leq u_i$ and the initial ball constraint $\|x\|_2 \leq r$. 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 6.1, 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 6.2. The remaining three subsections demonstrate the use of certified concave quadratic forms for the ellipsoid method (Section 6.3), volumetric cutting plane methods (Section 6.4), and for geometric rescaling methods (Section 6.5).

6.1 Dual certificates from certified quadratic forms

Duality theory provides the following variant of Farkas' lemma.

Lemma 6.5. *Given $A \in \mathbb{R}^{m \times n}$, $u \in \mathbb{R}^m$, $r > 0$, the system $Ax \leq u$ has no solution in $\mathbb{B}^n(r)$ if and only if there exists $\lambda \in \mathbb{R}^m_+$ such that*

$$r\|A^\top \lambda\| < -\lambda^\top u . \tag{16}$$

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

$$r\|A^\top \lambda + v\| + \nu \leq -\lambda^\top u . \tag{17}$$

Consider a polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq u\}$, 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 (17) as a *dual certificate of validity* of $v^\top x \geq \nu$ for $P \cap \mathbb{B}^n(r)$.

Definition 6.6. *Let $P = \{x : Ax \leq u\}$ 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*

$$q(x) := \gamma_0(r^2 - \|x\|^2) + \sum_{i=1}^m \gamma_i(u_i - a_i^\top x)(a_i^\top x - \ell_i) + d^\top x - \beta \tag{18}$$

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*

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

We will also say that the quadratic form (18) *is* certified *by $\mu^{(i)}$, $i \in [m]$, and ϑ.*

The certificates $\mu^{(i)}$, $i \in [m]$, and ϑ guarantee that $P \cap \mathbb{B}^n(r) \subseteq \{x : q(x) \geq 0\}$. We will show in Lemma 6.8 that any inequality $v^\top x \geq \nu$ that is valid for $\{x : q(x) \geq 0\}$ 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 6.7. *Let q be a strictly concave quadratic form q for $A, u, r, \ell, \gamma, d, \beta$ as in Definition 6.6. In particular, assume we are also give $\vartheta \in \mathbb{R}^m_+$ certifying $d^\top x \geq \beta$ for $P \cap \mathbb{B}^n(r)$ as $r\|A^\top \vartheta + d\| + \beta \leq -\vartheta^\top u$. 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*

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

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 $\bar{\ell}_i$:

$$\sum_{i=1}^m \gamma_i(\bar{u}_i - \bar{\ell}_i)a_i + d = 2\gamma_0 p \tag{19}$$

$$\gamma_0(r^2 - \|p\|^2) + \sum_{i=1}^m \gamma_i \bar{u}_i \bar{\ell}_i + d^\top p - \beta = q(p)\,. \tag{20}$$

Note that $\lambda_i = \gamma_i(\bar{u}_i - \bar{\ell}_i)$ for all $i \in [m]$. Hence, (19) can be written as $A^\top \lambda + d = 2\gamma_0 p$. Let $P := \{i \in [m] : \lambda_i \geq 0\}$ and $N := [m] \setminus P$. The proof is completed by

$$\begin{aligned}
\ell^\top \lambda^+ - u^\top \lambda^- - u^\top \vartheta &= -\sum_{i\in P} \lambda_i \bar{\ell}_i + \sum_{i\in N} \lambda_i \bar{u}_i + \sum_{i=1}^m \lambda_i a_i^\top p - u^\top \vartheta \\
\text{(by (19))} &= -\sum_{i\in P} \gamma_i(\bar{u}_i - \bar{\ell}_i)\bar{\ell}_i + \sum_{i\in N} \gamma_i(\bar{u}_i - \bar{\ell}_i)\bar{u}_i + 2\gamma_0\|p\|^2 - d^\top p - u^\top \vartheta \\
&= -\sum_{i=1}^m \gamma_i \bar{u}_i \bar{\ell}_i + \sum_{i\in P} \gamma_i \bar{\ell}_i^2 + \sum_{i\in N} \gamma_i \bar{u}_i^2 + 2\gamma_0\|p\|^2 - d^\top p - u^\top \vartheta \\
\text{(by (20))} &\geq -q(p) + \gamma_0(r^2 + \|p\|^2) - \beta - u^\top \vartheta \\
\text{(by the definition of } \vartheta) &\geq -q(p) + 2r\gamma_0\|p\| + r\|A^\top \vartheta + a\| \\
\text{(by (19))} &= -q(p) + r\left\|A^\top \lambda + a\right\| + r\|A^\top \vartheta + a\| \\
&\geq -q(p) + r\left\|A^\top(\lambda - \vartheta)\right\|\,.
\end{aligned}$$

□

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 6.8. *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 (18), 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 $O(n^2m + n^\omega)$.

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 $i = 0,\dots,m$. 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 6.7 to $\tilde q$, and observing that $\lambda_i = \tilde\gamma_i(u_i + \ell_i - 2a_i^\top x^*)$, $i \in [m]$, we obtain

$$\ell^\top\lambda^+ - u^\top\lambda^- - u^\top(\sigma\vartheta) - \nu \geq \tilde q(x^*) + r\left\|A^\top(\lambda - \sigma\vartheta) - v\right\| = r\left\|A^\top(\lambda - \sigma\vartheta) - v\right\|. \tag{21}$$

We need to show that $r\|A^\top\tilde\lambda + v\| + \nu \leq -\tilde\lambda^\top u$. From (21), we have

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

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 (Section 6.3), from volumetric cutting plane methods [23, 28, 40] (Section 6.4), and from the geometric rescaling algorithms [12, 21] (Section 6.5). 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.

Throughout, these algorithms maintain a set of oracle inequalities $Ax \leq u$ along with a strictly feasible point with respect to these inequalities. The main technical ingredient will be the following lemma. This is only needed for Section 6.4 and readers interested in the other algorithms only may skip it.

Lemma 6.9. *Let $P = \{x \in \mathbb{R}^n : Ax \leq u\}$ be a polytope, where $A \in \mathbb{R}^{m\times n}$ and $u \in \mathbb{R}^m$. Let z be a point in the interior of P, and $\gamma \in \mathbb{R}^m_{++}$. Define*

$$R := \sum_{i=1}^m \frac{\gamma_i a_i a_i^\top}{(u_i - a_i^\top z)^2}, \qquad w = \sum_{i=1}^m \frac{\gamma_i a_i}{u_i - a_i^\top z} \tag{22}$$

Let $\rho := \|w\|_{R^{-1}}$ and $\varphi := \max_{i\in[m]} \|a_i\|_{R^{-1}}/(u_i - a_i^\top z)$. Assume that $\varphi\rho \le 1/2$. Let $\ell \in \mathbb{R}^m$ and the quadratic form q be defined by

$$\ell_i := a_i^\top z - 3\varphi^2\|\gamma\|_1(u_i - a_i^\top z), \qquad q(x) = \sum_{i=1}^m \gamma_i \frac{(u_i - a_i^\top x)(a_i^\top x - \ell_i)}{(u_i - a_i^\top z)^2}. \tag{23}$$

The following hold.

- *(i) In $O(n^\omega)$ time, we can find coefficients $\mu^{(k)} \in \mathbb{R}^m_+$, such that $A^\top\mu^{(k)} = -a_k$, $u^\top\mu^{(k)} \le -\ell_k$ for $k \in [m]$, i.e. a dual certificate of validity of $a_k^\top x \ge \ell_k$ for P.*
- *(ii) q is strictly concave, $\max_{x\in\mathbb{R}^n} q(x) \le (3\varphi\|\gamma\|_1)^2$, proving that $P \subseteq 3\varphi\|\gamma\|_1 E(R,p)$ for $p = \arg\max q(x)$.*

Proof. Define $\bar a_i = a_i/(u_i - a_i^\top z)$ for $i \in [m]$. With this notation, $R = \sum_{i=1}^m \gamma_i \bar a_i \bar a_i^\top$ and $w = \sum_{i=1}^m \gamma_i \bar a_i$. Observe that $P = \{x \in \mathbb{R}^n : \bar a_i^\top(x - z) \le 1,\ i \in [m]\}$. For part *(a)*, for any $k \in [m]$, let us define $\bar\mu^{(k)}$ by

$$\bar\mu_i^{(k)} := \gamma_i \left(2\varphi^2(1 - \bar a_i^\top R^{-1} w) - \bar a_i^\top R^{-1}\bar a_k\right), \qquad i = 1, \dots, m. \tag{24}$$

Observe that $\bar\mu^{(k)} \ge 0$ because $|\bar a_i R^{-1} w| \le \|\bar a_i\|_{R^{-1}}\|w\|_{R^{-1}} \le \rho\varphi \le 1/2$ and $|\bar a_i R^{-1}\bar a_k| \le \|\bar a_i\|_{R^{-1}}\|\bar a_k\|_{R^{-1}} \le \varphi^2$, by definition of φ and ρ.

Next, observe that

$$\sum_{i=1}^m \bar\mu_i^{(k)}\bar a_i = 2\varphi^2\left(\sum_{i=1}^m \gamma_i\bar a_i - \sum_{i=1}^m \gamma_i \bar a_i \bar a_i^\top R^{-1} w\right) - \sum_{i=1}^m \gamma_i \bar a_i \bar a_i^\top R^{-1}\bar a_k = -\bar a_k$$

and

$$\sum_{i=1}^m \bar\mu_i^{(k)} = 2\varphi^2(\|\gamma\|_1 - \sum_{i=1}^m \gamma_i \bar a_i R^{-1} w) - (\sum_{i=1}^m \gamma_i \bar a_i) R^{-1}\bar a_k = 2\varphi^2(\|\gamma\|_1 - \rho^2) - w^\top a_k \le 2\varphi^2\|\gamma\|_1 + \varphi\rho \le 3\varphi^2\|\gamma\|_1,$$

where the last inequality follows from $\rho \le \varphi\|\gamma\|_1$, because $\rho^2 = \sum_{i=1}^m \gamma^i a_i^\top R^{-1} w \le \sum_{i=1}^m \gamma^i \varphi\rho$.

It follows that the vector $\mu^{(k)} \in \mathbb{R}^m_+$ defined by

$$\mu_i^{(k)} := \frac{u_k - a_k^\top z}{u_i - a_i^\top z}\bar\mu_i^{(k)},$$

satisfies $A^\top\mu^{(k)} = -a_k$, $u^\top\mu^{(k)} \le -\ell_k$, since

$$A^\top\mu^{(k)} = (u_k - a_k^\top z)\sum_{i=1}^m \bar\mu_i^{(k)}\bar a_i = -(u_k - a_k^\top z)\bar a_k = -a_k,$$

and

$$u^\top\mu^{(k)} = \sum_{i=1}^m (u_i - a_i^\top z)\mu_i^{(k)} + \sum_{i=1}^m a_i^\top z\,\mu_i^{(k)} = (u_k - a_k^\top z)\sum_{i=1}^m \bar\mu_i^{(k)} - a_k^\top z \le (u_k - a_k^\top z)3\varphi^2\|\gamma\|_1 - a_k^\top z = -\ell_k.$$

For part *(b)*, note that q is strictly concave because P is a polytope, hence $\mathrm{rk}(A) = n$. Consider any $x \in \mathbb{R}^n$ and let $y = x - z$. We have

$$\begin{aligned} q(x) &= q(y+z) = \sum_{i=1}^m \gamma_i(1 - \bar a_i^\top y)(\bar a_i^\top y + 3\varphi^2\|\gamma\|_1) \\ &= -y^\top\left(\sum_{i=1}^m \gamma_i \bar a_i a_i^\top\right) y - \left(1 + 3\varphi^2\|\gamma\|_1\right)\sum_{i=1}^m \gamma_i \bar a_i^\top y + 3\varphi^2\|\gamma\|_1 \sum_{i=1}^m \gamma_i \\ &= -y^\top R y - (1 + 3\varphi^2\|\gamma\|_1) w^\top y + 3\varphi^2\|\gamma\|_1^2\,. \end{aligned}$$

If we define $p = z - (1/2)(1 + 3\varphi^2\|\gamma\|_1)R^{-1}w$, we have that

$$\begin{aligned} (x-p)^\top R(x-p) &= (x-z)^\top R(x-z) + 2(x-p)^\top R(z-p) + (z-p)^\top R(z-p) \\ &= (x-z)^\top R(x-z) + (1 + 3\varphi^2\|\gamma\|_1) w^\top(x-z) + (1/4)(1 + 3\varphi^2\|\gamma\|_1)^2\rho^2 \\ &= y^\top R y + (1 + 3\varphi^2\|\gamma\|_1) w^\top y + (1/4)(1 + 3\varphi^2\|\gamma\|_1)^2\rho^2\,. \end{aligned}$$

It follows that

$$q(x) = -(x-p)^\top R(x-p) + 3\varphi^2\|\gamma\|_1^2 + (1/4)(1+3\varphi^2\|\gamma\|_1)^2\rho^2\,.$$

In particular, p is the unique maximizer of q, and

$$\max_{x\in\mathbb{R}^n} q(x) = 3\varphi^2\|\gamma\|_1^2 + (1/4)(1+3\varphi^2\|\gamma\|_1)^2\rho^2 \le (3\varphi\|\gamma\|_1)^2\,,$$

where the last inequality follows from $(1+3\varphi^2\|\gamma\|_1)\rho \le \varphi\|\gamma\|_1 + (3/2)\varphi\|\gamma\|_1$, because $\rho \le \varphi\|\gamma\|_1$ and $\varphi\rho \le 1/2$. □

6.2 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 (18) 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 6.10. *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 6.8, 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 $(\mu')^+, (\mu'')^+$ 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

$$r\|A^\top\lambda' - v\| - v^\top p - \varepsilon/6 \le -u^\top\lambda', \quad r\|A^\top\lambda'' + v\| + v^\top p - \varepsilon/6 \le -u^\top\lambda''. \tag{25}$$

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 (25) 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

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

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 $\alpha \leq \frac{\varepsilon}{3r}$. Note that

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

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

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

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

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

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

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

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

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

6.3 Approximate Farkas certificates from the ellipsoid method

In this section we describe an algorithm, which we will call the *Certified Ellipsoid Method* and prove the following.

Theorem 6.11. *Let K be a convex set given by a strong separation oracle, $r > 0$, and $\varepsilon \in (0, 2r)$. Then, the Certified Ellipsoid Method runs in oracle-polynomial time and, by making $O(n^2 \log(nr/\varepsilon))$ calls to the strong separation oracle, either returns a point $x \in K$, or an ε-approximate Farkas certificate for $K \cap \mathbb{B}^n(r)$ comprising only oracle inequalities.*

As previously noted, this yields a subroutine Approx-Conic-Dual with $\mathcal{T}(n, \varepsilon) = O(n^2 \log(1/\varepsilon))$ oracle calls.

Remark 6.12. *In the remainder of the section, we will assume $r = 1$. Indeed, if we define $K' = K/r$, and $\varepsilon' = \varepsilon/r$, observe that an inequality $a^\top x \leq \beta$ is valid for K if and only if $ra^\top x \leq \beta$ is valid for K'. Hence, given a system $rAx \leq u$ of m valid inequalities for K', an ε'-approximate Farkas certificate for $K' \cap \mathbb{B}^n(1)$ is of the form $\lambda' \in \mathbb{R}^m_+$ such that $\|rA^\top \lambda'\| + u^\top \lambda' \leq \varepsilon'$, $\sum_{i=1}^m \lambda'_i r\|a_i\| \geq 1$, hence $\lambda = r\lambda'$ is an ε-approximate Farkas certificate for $K \cap \mathbb{B}^n(r)$.*

We give a self-contained exposition of the ellipsoid method and the certification procedure that may provide new insights into the classical algorithm. Section 6.3.1 describes the Basic Ellipsoid Method in a slightly stronger form. Section 6.3.2 introduces the Certified Ellipsoid Method, where we modify the original framework to show how the containing ellipsoid can always be maintained in terms of certified quadratic forms, which will imply Theorem 6.11.

6.3.1 The Basic Ellipsoid Method

Theorem 6.13. *Let K be a convex set given by a strong separation oracle, and $\varepsilon > 0$. Then, there exists an oracle-polynomial time algorithm that, by making $O(n^2 \log(n/\varepsilon))$ calls to the strong separation oracle, either returns a point $x \in K$, or returns a vector $a \in \mathbb{R}^n$ with $\|a\|_2 \in [1, 2]$ such that $\operatorname{width}_{K \cap \mathbb{B}^n(1)}(a) \leq \varepsilon$. Moreover, the vector a will be one of the vectors returned by the separation oracle during the algorithm.*

The statement differs from the usual form in the case when no feasible solution is found. For this case, the usual outcome as in [19, Theorem 3.2.1] is a small volume ellipsoid containing K. Given such an ellipsoid E, one can show that E (and thus K) is thin in the direction in one of the principal axes of E (see e.g. the proof of [19, Lemma 6.4.2]). Instead, using Lemma 6.4, we observe that a small volume ellipsoid is thin in one of the directions returned by the oracle.

At the t-th iteration, the algorithm maintains positive definite matrices $R_t, Q_t \in \mathbb{S}^n_{++}$ such that $Q_t = R_t^{-1}$, and a vector $p_t \in \mathbb{R}^n$. We let $E_t = E(R_t, p_t)$. Throughout, we maintain

$$K \cap \mathbb{B}^n(1) \subseteq E_t \,. \tag{26}$$

We initialize $E_0 = \mathbb{B}^n(1) = E(I_n, 0)$. At iteration $t = 1, 2, \ldots$, the strong separation oracle is called to check if $p_{t-1} \in K$. If the answer is yes, the algorithm terminates. Otherwise, the oracle returns a direction a_t such that $a_t^\top p_{t-1} > a_t^\top z$ for any $z \in K$. In this case, the matrices are updated as follows:

$$R_t = \frac{n^2-1}{n^2} \cdot R_{t-1} + \frac{2n+2}{n^2} \cdot \frac{a_t a_t^\top}{\|a_t\|^2_{Q_{t-1}}}\,, \quad Q_t = R_t^{-1}\,, \quad p_t = p_{t-1} - \frac{Q_t a_t}{(n+1)\|a_t\|_{Q_t}}\,. \tag{27}$$

We note that computing Q_t does not require a matrix inversion, but can be computed via a simple rank-1 update formula from Q_{t-1}, similarly to R_t.

The following lemma is the key in showing the progress in the volumetric potential. In Section 6.3.2, we present a new proof using quadratic forms.

Lemma 6.14. *If $K \cap \mathbb{B}^n(1) \subseteq E_{t-1}$, then $K \cap \mathbb{B}^n(1) \subseteq E_t$ also holds, and $\det(R_t) > e^{1/n} \det(R_{t-1})$.*

The next lemma is immediate from the construction sequence.

Lemma 6.15. *At iteration $t \geq 1$ of the Basic Ellipsoid Method, there exist coefficients $0 < \gamma_0^{(t)} < 1$, and $\gamma_i^{(t)} > 0$ for $i \in [t]$ such that*

$$R_t = \gamma_0^{(t)} I_n + \sum_{i=1}^{t} \gamma_i^{(t)} a_i a_i^\top. \tag{28}$$

The next lemma completes the proof of Theorem 6.13.

Lemma 6.16. *If the Basic Ellipsoid Method does not terminate in $T = O(n^2 \log(n/\varepsilon))$ iterations, then $\mathrm{width}_{K \cap \mathbb{B}^n(1)}(a_t) \leq \varepsilon$ for some $t \in [T]$.*

Proof. Initially, $\det(R_0) = 1$. Using Lemma 6.14, we see that after $t = O(n^2 \log(n/\varepsilon))$ iterations, $\det(R_t)^{1/n} > 1/\varepsilon^2 + 1$. The claim follows using Lemmas 6.3 and 6.4. □

6.3.2 The certified ellipsoid method

Let us denote by $q_t(x)$ the strictly concave quadratic form defined as

$$q_t(x) := 1 - (x - p_t)^\top R_t (x - p_t)\,, \tag{29}$$

so that $E_t = \{x \in \mathbb{R}^n \,:\, q_t(x) \geq 0\}$. Note that initially $q_0(x) = 1 - \|x\|^2$. The Certified Ellipsoid Method will maintain a q_t as a certified quadratic form as in Definition 6.6. Combined with Theorems 6.10 and 6.13, this will imply Theorem 6.11.

Assume that for the current ellipsoid $E_t = E(R_t, p_t)$, the separation oracle returned a_t such that $a_t^\top p_{t-1} > a_t^\top z$ for all $z \in K$. If we let

$$u_t := a_t^\top p_{t-1}$$

then the inequality $a_t^\top x \leq u_t$ is the inequality returned by the oracle, and it is valid for K. Further, if we let

$$\ell_t := a_t^\top p_{t-1} - \|a_t\|_{Q_{t-1}}\,,$$

then the fact that $K \cap \mathbb{B}^n(1) \subseteq E_{t-1}$ ensures that $a_t^\top x \geq \ell_t$ is also valid for $K \cap \mathbb{B}^n(1)$.

The following lemma shows that the ellipsoid update (27) corresponds to the following update of the quadratic form, for some values $\alpha, \beta > 0$.

$$q_t(x) = \alpha q_{t-1}(x) + \frac{\beta}{\|a_t\|^2_{Q_{t-1}}} (u_t - a_t^\top x)(a_t^\top x - \ell_t)\,, \tag{30}$$

Lemma 6.17. *For some $\alpha, \beta > 0$, consider the expression $q_t(x)$ as in (30) with $u_t = a_t^\top p_{t-1}$ and $\ell_t = a_t^\top p_{t-1} - \|a_t\|_{Q_{t-1}}$, and let p_t be the unique maximizer of $q_t(x)$. There exists $R_t \in \mathbb{S}^n_{++}$ such that $q_t(x) = 1 - (x - p_t)^\top R_t (x - p_t)$ if and only if*

$$\alpha = \frac{1-2\gamma}{(1-\gamma)^2}\,, \quad \beta = \frac{2\gamma}{(1-\gamma)^2}\,, \quad \text{for some } 0 < \gamma < \frac{1}{2}\,. \tag{31}$$

Furthermore, for any such α, β, and γ,

$$R_t = \alpha R_{t-1} + \beta \frac{a_t a_t^\top}{\|a_t\|^2_{Q_{t-1}}}, \quad p_t = p_{t-1} + \gamma \frac{Q_{t-1} a_t}{\|a_t\|_{Q_{t-1}}}. \tag{32}$$

Finally, the choice of γ that minimizes the volume of E_t (or, equivalently, maximizes $\det(R_t)$) is $\gamma^* := 1/(n+1)$, for which choice $\det(R_t)/\det(R_{t-1}) \geq e^{1/n}$.

Proof. Denote $q_t := q$, $a := a_t/\|a_t\|_{Q_{t-1}}$, $p := p_{t-1}$, $p' := p_t$, $R := R_{t-1}$, $Q := Q_{t-1}$, and $R' = R_t$. If there exists a positive definite matrix R' and p' such that $q(x) \equiv 1 - (x-p')^\top R'(x-p')$ then $q(p') = 1$. Since p' is the unique maximizer of $q(x)$, we have $\nabla q(p') = 0$. Using that

$$\nabla q(x) = -2\alpha R(x-p) - \beta(1 + 2a^\top(x-p))a, \tag{33}$$

from $\nabla q(p') = 0$ we see that $p' = p - \gamma Q a$, where $\gamma \in \mathbb{R}$ satisfies

$$2\alpha\gamma - \beta(1-2\gamma) = 0. \tag{34}$$

The condition $q(p') = 1$ implies

$$\alpha(1-\gamma^2) + \beta\gamma(1-\gamma) = 1. \tag{35}$$

Solving the linear system given by (34), (35) for α and β in terms of the parameter γ, we obtain (31), and $\alpha, \beta > 0$ if only if $0 < \gamma < 1/2$.

Let us now fix a value of $0 < \gamma < 1/2$ and compute α and β as above. Let us set $R' = \alpha R + \beta a a^\top$. We show that $q(x) \equiv q'(x)$ for $q(x) = q_t(x)$ as defined in (30), and for $q'(x) = 1 - (x-p')^\top R'(x-p')$. This is a consequence of the following simple claim.

Claim 6.18. *Two strictly concave quadratic functions are identical if and only if the following are the same for the two functions:* (i) *the quadratic terms;* (ii) *the unique maximizers; and* (iii) *the maximum values.*

The quadratic term in both $q(x)$ and $q'(x)$ is $-x^\top(\alpha R + \beta a a^\top)x$. The maximizer of both functions is p', and the maximum value is 1 in both cases. This completes the proof of (32).

Finally, we need to determine the choice of γ in order to minimize the volume of E_t. This is equivalent to maximizing $\det(R')$. Note that

$$\begin{aligned} \det(R') &= \alpha^n \det\left(R^{1/2}\left(I + \frac{\beta}{\alpha} Q^{1/2} a a^\top Q^{1/2}\right) R^{1/2}\right) = \det(R)\alpha^n\left(1 + \frac{\beta}{\alpha}\right) \\ &= \det(R)\alpha^{n-1}(\alpha+\beta) = \det(R)\frac{(1-2\gamma)^{n-1}}{(1-\gamma)^{2n}}. \end{aligned} \tag{36}$$

This is maximized for $\gamma = 1/(n+1)$, for which choice

$$\frac{\det(R_t)}{\det(R_{t-1})} = \left(\frac{n-1}{n}\right)^{n-1}\left(\frac{n+1}{n}\right)^{n+1} \geq e^{1/n}.$$

□

Observe that the above lemma immediately implies Lemma 6.14, hence this provides an alternative exposition of the standard volumetric argument for the ellipsoid method.

Corollary 6.19. *At the t-th iteration of the Basic Ellipsoid Method, we can maintain $q_t(x)$ in the form*

$$q_t(x) = \gamma_0^{(t)}(1 - \|x\|^2) + \sum_{i=1}^{t} \gamma_i^{(t)}(u_i - a_i^\top x)(a_i^\top x - \ell_i), \tag{37}$$

where $\gamma_0^{(t)} = \alpha^t$, and for each $i \in [t]$, a_i is the vector returned by the separation oracle at the i-th iteration, $u_i = a_i^\top p_{i-1}$, $\ell_i = u_i - \|a_i\|_{Q_{i-1}}$, and $\gamma_i^{(t)} = \beta\alpha^{t-i}/\|a_i\|^2_{Q_i}$, where α, β are defined as in (31) for $\gamma = 1/(n+1)$. Furthermore, for every $k \in [t]$, we can compute a certificate $\mu^{(k)} \in \mathbb{R}^t_+$ for the validity of $a_k^\top x \ell_k$, that is

$$\left\|\sum_{i=1}^{t} \mu^{(i)} a_i + a_k\right\| + \ell^{(k)} \leq -\sum_{i=1}^{t} \lambda_i^{(k)} u_i. \tag{38}$$

Proof. The first statement follows by construction and by Lemma 30. For the last statement, we observe that $\mu^{(k)} \in \mathbb{R}^t_+$, $k \in [t]$ can be computed throughout the execution of the ellipsoid method. Indeed, suppose that up to iteration $t-1$ we have computed q_{t-1} in the form (37), along with $\mu^{(1)}, \ldots, \mu^{(t-1)}$. Since the inequality $a_t^\top x \geq \ell_t$ is valid for $E_{t-1} = \{x : q_{t-1}(x) \geq 0\}$, it follows from Lemma 6.8 that we can compute $\mu^{(t)}$ satisfying (38). □

Maintaining a compact representation So far, we kept adding a new term $a_i a_i^\top$ to R_t in every iteration, and thus R_t will be the weighted sum of the identity and t rank-1 matrices $a_i a_i^\top$. This would lead to $O(n^2 \log(n/\varepsilon))$ terms in R when running the Basic Ellipsoid Method to obtain a ε-thin direction, according to Lemma 6.16. Thus, the space complexity of the algorithm is large, albeit polynomial. Furthermore, we must maintain certificates $\mu^{(i)}$ for each inequality $a_i^\top x \geq \ell_i$, $i \in [t]$. In what follows, we observe that one can maintain each quadratic form q_t in terms of only $O(n^2)$ terms of the form $(u_i - a_i^\top x)(a_i^\top x - \ell_i)$, and that one needs to only keep $O(n^3)$ vectors a_i to certify the inequalities $a_i^\top x \geq \ell_i$.

Indeed, one can readily verify from the expression (37) that

$$q_t(x) = (1, -x^\top) H \begin{pmatrix} 1 \\ -x \end{pmatrix},$$

where the vector $(\gamma_1^{(t)}, \ldots, \gamma_t^{(t)})$ is a feasible solution to the following system of $\binom{n+1}{2}$ equations in the nonnegative variables $\gamma_1, \ldots, \gamma_t$

$$\sum_{i=1}^t \gamma_i \left(\begin{array}{c|c} \ell_i u_i & a_i^\top (\ell_i + u_i)/2 \\ \hline a_i (\ell_i + u_i)/2 & a_i a_i^\top \end{array} \right) = H - \gamma_0^{(t)} I_{n+1}.$$

If we choose $\gamma \in \mathbb{R}^t_+$ to be a basic solution and let $I_t \subseteq [t]$ be its support, it follows that $|I_t| \leq \binom{n+1}{2}$ and

$$q_t(x) = \gamma_0^{(t)}(1 - \|x\|^2) + \sum_{i \in I_t} \gamma_i (u_i - a_i^\top x)(a_i^\top x - \ell_i).$$

Furthermore, for every $k \in I_t$, given $\mu^{(k)} \in \mathbb{R}^t_+$ satisfying (38), we can compute a basic solution $\tilde{\mu}^{(k)} \in \mathbb{R}^t_+$ to the system $\sum_{i=1}^t \tilde{\mu}^{(i)} a_i = \sum_{i=1}^t \mu^{(i)} a_i$ with $\sum_{i=1}^t \tilde{\mu}^{(i)} u_i \leq \sum_{i=1}^t \mu^{(i)} u_i$, hence obtaining a certificate for $a_k^\top x \geq \ell_k$ of support size at most n. If we denote by C_k the support of $\tilde{\mu}^{(k)}$, then we only need to maintain the vectors a_i, $i \in I^t \cup \bigcup_{i \in I_t} C_i$, along with the certificates $\tilde{\mu}^{(k)} \in \mathbb{R}^{I_t}_+$.

By induction, we will guarantee that we obtain $\mu^{(t)}$ as in (38) of support size $O(n^3)$. This can be reduced to an $O(n)$ size basic solution in time $O(n^5)$. This amounts to a substantial running time overhead: whereas an update takes $O(n^2)$ time for the Basic Ellipsoid Method, maintaining a small I amounts to $O(n^4)$ per update, and maintaining the certificates to $O(n^5)$.[4] The increased complexity bound is however still much lower than the algorithm for finding a dual optimal solution in [19, Lemma 6.5.15] that would amount to running the Ellipsoid Method for a second time with $O(n^2 \log(n/\varepsilon))$ variables (for an appropriate ε in the context of rational polyhedra).

6.4 Approximate Farkas certificates from volumetric cutting plane methods

We now show how to derive Farkas certificates from volumetric cutting plane methods. This method was introduced by Vaidya [40]; similar cutting plane methods with improved arithmetic complexity were given by Lee, Sidford, and Wong [28] and by Jiang, Lee, Song, and Wong [23]. For simplicity, we present the implementation for Vaidya's original algorithm, but the same framework is applicable for the subsequent variants as well. As previously noted, [28] also includes an explicit statement on dual certificates.

In contrast to the ellipsoid method, we do not maintain the certified concave quadratic form during the algorithm, but construct it at termination using Lemma 6.9.

Given a polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq u\}$, $A \in \mathbb{R}^{m \times n}$, $u \in \mathbb{R}^m$, the log-barrier function is defined over the interior of P by $f(x) = -\sum_{i=1}^m \log(u_i - a_i^\top x)$, and its Hessian is

$$\nabla^2 f(x) = \sum_{i=1}^m \frac{a_i a_i^\top}{(u_i - a_i^\top x)^2}.$$

[4] It is possible to improve the complexity of maintaining a small I to $O(n^3)$, by only approximately maintaining $q(x)$.

Let F be the function defined over the interior of P by $F(x) = \log(\det(\nabla^2 f(x)))$.

The function F is strictly convex [40], and the minimizer $\tilde{\omega}$ of F is called the *volumetric center* of P. The gradient of F is

$$\nabla F(x) = \sum_{i=1}^{m} \sigma_i(x) \frac{a_i}{u_i - a_i^\top x}, \quad \text{where} \quad \sigma_i(x) = \frac{a_i^\top (\nabla^2 f(x))^{-1} a_i}{(u_i - a_i^\top x)^2}, \qquad i \in [m].$$

Observe that $\sum_{i=1}^m \sigma_i(x) = n$ and $\sigma_i(x) \leq 1$ for $i \in [m]$. Finally, let us define

$$Q(x) = \sum_{i=1}^{m} \sigma_i(x) \frac{a_i a_i^\top}{(u_i - a_i^\top x)^2}.$$

The $\sigma_i(x)$ values are the *leverage scores* of the subspace

$$\operatorname{span}(\operatorname{diag}((u_1 - a_1^\top x)^{-1}, \ldots, (u_m - a_m^\top x))A).$$

Since $\sigma_i(x) \leq 1$, $i \in [m]$, $\nabla^2 f(x) \succeq Q(x)$. Let $\mu(x)$ be the largest number $\tilde{\mu}$ such that $Q(x) \succeq \tilde{\mu} \nabla^2 f(x)$, and note that $\mu(x) \geq \min_{i \in [m]} \sigma_i(x)$.

Let K defined by a strong separation oracle, $\varepsilon > 0$, and assume we want to find a point in K or an ε-approximate Farkas certificate for $K \cap \mathbb{B}^n(1)$; as in Remark 6.12, we can assume $r = 1$. Let $\delta \leq 10^{-4}$, $c \leq 10^{-3}\delta$. At every iteration, Vaidya's algorithm maintains a set of inequalities $Ax \leq u$, where $a_i^\top x \leq u_i$ is either an inequality returned by the oracle, or one of the bound inequalities $x_j \leq 1$ or $-x_j \leq 1$. The algorithm also maintains a point $z \in \mathbb{R}^n$ satisfying $Az < u$ such that

$$F(z) - F(\tilde{\omega}) \leq c^4 \mu(\tilde{\omega}). \tag{39}$$

The system is initialized to $-1 \leq x_j \leq 1$, $j \in [n]$, and $z = \tilde{\omega} = 0$. At any given iteration, if $\sigma_i(z) < c$ for some $i \in [m]$, the inequality $a_i^\top x \leq u_i$ is removed from the system. If $\min_{i \in [m]} \sigma_i(z) \geq c$, the separation oracle is queried for z; if $z \in K$ then the algorithm stops, else the oracle returns a vector $a_i \in \mathbb{R}^n$ such that $a_i^\top x < a_i^\top z$, and the algorithm adds the inequality $a_i^\top x \leq u_i$ to the system, for some appropriately chosen value $u_i > a_i^\top z$. In both cases (whether an inequality has been added or removed), the algorithm performs a fixed number of Newton steps to compute a new point z satisfying (39) with respect to the new system. Each iteration requires $O(n^\omega)$ arithmetic operations; the algorithms [23, 28] improve on the arithmetic complexity.

Let us denote by ρ^k the value of $F(\tilde{\omega})$ at the kth iteration. Note that at the beginning $\nabla^2 f(z) = I_n$, hence $\rho^0 = \log(\det(I_n)) = 0$. Vaidya [40] shows

$$\rho^k \geq \frac{ck}{2}. \tag{40}$$

Theorem 6.20. *Let K be a convex set given by a strong separation oracle, $r > 0$, and $\varepsilon \in (0, 2r)$. Then, in $O(n \log(nr/\varepsilon))$ calls to the strong separation oracle and $O(n^{\omega+1} \log(nr/\varepsilon))$ arithmetic operations, Vaidya's 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 support of the certificate has cardinality $O(n)$.*

Proof. As previously noted, we may assume $r = 1$. Observe that, since $\sum_{i=1}^m \sigma_i(z) = n$, then at every iteration $m \leq n/c$, since otherwise $\min_{i \in [m]} \sigma_i(z) < c$, and an inequality is removed from the system. By (40), after $O(n \log(n/\varepsilon))$ iterations we have $\det(\nabla^2 f(z)) \geq c^{-1.5n}(20n/\varepsilon)^{2n}$. Furthermore, continuing Vaidya's algorithm by removing constraints as long as $\min_{i \in [m]} \sigma_i(z) < c$, which requires at most $m \leq n/c$ further iterations, we can also guarantee that $\sigma_i(z) \geq c$ for all $i \in [m]$. In particular, it follows that $\det(Q(z)) \geq c^n \det(\nabla^2 f(z)) \geq c^{-n/2}(20n/\varepsilon)^{2n}$.

Vaidya (Lemma 10 in [40]) shows that, if $F(z) - F(\tilde{\omega}) \leq \delta\sqrt{\mu(\tilde{\omega})}$, then $\nabla F(z)^\top Q(z)^{-1} \nabla F(z) \leq (F(z) - F(\tilde{\omega}))/0.14$. Since we maintain a point z satisfying (39), and since $c^4 \mu(\tilde{\omega}) < \delta\sqrt{\mu(\tilde{\omega})}$, it follows that $\nabla F(z)^\top Q(z)^{-1} \nabla F(z) \leq c^4 \mu(\tilde{\omega})/0.14$.

Vaidya (Claim 4 in [40]) also shows that $a_i^\top Q(z)^{-1} a_i / (u_i - a_i^\top z)^2 \leq 1/\sqrt{\mu(z)} \leq c^{-1/2}$, $i \in [m]$.

Let us now define $\gamma \in \mathbb{R}^m_{++}$ by $\gamma_i := \sigma_i(x)$, $i \in [m]$, $R := Q(z)$, $w = \nabla F(z)$. Observe that γ, z, R, and w satisfy (22). By the above discussion, we have $\rho := \|w\|_{R^{-1}} \leq \sqrt{c^4 \mu(\tilde{\omega})/0.14}$ and $\varphi := \max_{i \in [m]} \|a_i\|_{R^{-1}} / (u_i - a_i^\top z) \leq c^{-1/4}$. Note that $\rho\varphi \leq 1/2$, hence γ satisfies the assumptions of Lemma 6.9, which shows that the quadratic form defined in (23) is certified for $Ax \leq u$, and that the ellipsoid $E = \{x \in \mathbb{R}^n : q(x) \geq 0\}$ satisfies $E = \alpha E(R, p)$, p being the center of E

and $\alpha \le 3\varphi\|\gamma\|_1 \le 3nc^{-1/4}$. It now follows from Lemmas 6.4 that there exists $k \in [m]$ such that $\|a_k\|_{R^{-1}}/\|a_k\|_2 \le (\det(R)^{1/n} - 1)^{-1/2}$, and from Lemma 6.3 we have that, for $v = a_k/\|a_k\|_2$,

$$\mathrm{width}_E(v) = 2\alpha\|v\|_{R^{-1}} \le 6nc^{-1/4}(\det(R)^{1/n} - 1)^{-1/2} \le 6nc^{-1/4}(c^{-1/2}(20n/\varepsilon)^2 - 1)^{-1/2} \le \varepsilon/3.$$

It follows from Theorem 6.10 that we can compute an ε-approximate Farkas certificate for $K \cap \mathbb{B}^n(1)$ comprising only oracle inequalities. □

6.5 Approximate conic certificates from geometric rescaling methods

We consider a particular variant of the geometric rescaling algorithms introduced by Dadush et al. [12, 13] and by Hoberg and Rothvoß [21]. Similar certification should be applicable for other variants as well. The algorithm discussed here takes as input a cone $K \subseteq \mathbb{R}^n$ defined by a conic separation oracle, along with $\varepsilon > 0$, and returns either a point $x \in K$ or determines a "thin direction", that is, a vector $v \in \mathbb{R}^n$, $\|v\| = 1$, such that $K \subseteq \{x \in \mathbb{R}^n : |v^\top x| \le \varepsilon\}$. Here we briefly describe the algorithm and the analysis, to show that in the second case we can compute an ε-approximate conic Farkas certificate for K.

The algorithm will maintain, at each iteration t, a matrix $R_t \in \mathbb{S}^n_{++}$ such that $K \cap \mathbb{B}^n(1) \subseteq E(R_t, 0)$ and such that $\det(R_t)$ increases at each iteration. Throughout we denote $Q_t = R_t^{-1}$. Initially $R_0 := I_n$. The algorithm sets a threshold $\beta := 1/(11n)$. At iteration $t+1$, the algorithm applies von Neumann's algorithm (first described by Dantzig in [14]) to compute a set of vectors $\{m_i : i \in J_t\}$ returned by the conic separation oracle and $\zeta^{(t)} \in \mathbb{R}^{J_t}_+$, $\|\zeta^{(t)}\|_1 = 1$, such that the vector

$$y := \sum_{i \in J_t} \frac{\zeta_i^{(t)} m_i}{\|m_i\|_{Q_t}},$$

either satisfies that $Q_t y \in K$ (in which case we stop), or $\|y\|_{Q_t} \le \beta$. In the latter case, we update

$$R_{t+1} := \frac{R_t + P_t}{1+\beta}, \qquad \text{where } P_t := \sum_{i \in J_t} \frac{\zeta_i^{(t)}}{\|m_i\|^2_{Q_t}} m_i m_i^\top. \tag{41}$$

We have $\det(R_{t+1}) \ge (16/9)\det(R_t)$ [12, Lemma 11].

Next we observe that the quadratic form $q_t(x) := 1 - x^\top R_t x$ can be maintained as a certified quadratic form for $K \cap \mathbb{B}^n(1)$. Inductively, assume that

$$q_t(x) = \gamma_0(1 - \|x\|^2) + \sum_{i \in I_t} \gamma_i m_i^\top x(u_i - m_i^\top x) + \nu - v^\top x, \tag{42}$$

where we have maintained dual certificates of validity over $K \cap \mathbb{B}^n(1)$ for the inequalities $m_i^\top x \le u_i$, $i \in I_t$, and $v^\top x \le \nu$.

Note that, for all $i \in J_t$, by Cauchy-Schwartz the inequality $m_i^\top x \le \|m_i\|_{Q_t}$ is valid for $K \cap \mathbb{B}^n(1)$, because $\|x\|_{R_t} \le 1$ for all $x \in K \cap \mathbb{B}^n(1) \subseteq E(R_t, 0)$. Since $E(R_t, 0) = \{x : q_t(x) \ge 0\}$ and q_t is a certified strictly concave quadratic form, by Lemma 6.8 we can compute dual certificates of validity for $m_i^\top x \le \|m_i\|_Q$ over $K \cap \mathbb{B}^n(1)$. Similarly, we have that $y^\top x \le \|y\|_{Q_t} \le \beta$ is valid for $K \cap \mathbb{B}^n(1)$, and we can compute a dual certificate of validity for it. Hence the quadratic form

$$r(x) = \sum_{i \in J_t} \frac{\zeta^{(t)}}{\|m_i\|^2_{Q_t}} m_i^\top x(\|m_i\|_{Q_t} - m_i^\top x) + \beta - y^\top x,$$

is certified for $K \cap \mathbb{B}^n(1)$. It follows from the definition of y and P_t that $r(x) = \beta - x^\top P_t x$, hence $q_{t+1}(x) = (1+\beta)^{-1}(q_t(x) + r_t(x))$. Since q_t and r_t are both certified, also q_{t+1} is certified, as can be seen by observing that inequality $(v+y)^\top x \le \nu + \beta$, corresponding to the linear-term in $q_{t+1}(x)$, can be certified as follows. If ϑ and ϑ' certify the validity of $v^\top x \le \nu$ and $y^\top x \le \beta$ for $K \cap \mathbb{B}^n(1)$, then $\vartheta + \vartheta'$ certifies the validity of $(v+y)^\top x \le \nu + \beta$, since $\|\sum_{i \in I_t}(\vartheta_i + \vartheta'_i)m_i + v + y\| \le \|\sum_{i \in I_t} \vartheta_i m_i + v\| + \|\sum_{i \in I_t} \vartheta'_i m_i + y\| \le \nu + \beta$.

Note that, exactly as explained in Section 6.3.2, we can apply Caratheodory to maintain the expression $\sum_{i \in I_t} \gamma_i m_i^\top x(u_i - m_i^\top x)$ in (42) so that $|I_t| \in O(n^2)$.

Theorem 6.21. *Let $K \subseteq \mathbb{R}^n$ be a convex cone given by a conic separation oracle, and $\varepsilon > 0$. Then, in $O(n^3 \log(n/\varepsilon))$ calls to the separation oracle, and $O(n^5 \log(n/\varepsilon))$ arithmetic operations, the geometric rescaling algorithm either returns a point $x \in K$, or an ε-approximate conic Farkas certificate.*

Proof. Since $\det(R_{t+1}) \geq (16/9)\det(R_t)$, after $T \in O(n\log(n/\varepsilon))$ iterations, there exists $k \in I_T$ such that $\|m_k\|_{Q_T} \leq \varepsilon\|m_k\|_2$. Since the quadratic form $q_T(x)$ is certified, we can compute a certificate of validity $\lambda \in \mathbb{R}_+^{I_T}$ for the inequality $(m_k/\|m_k\|_2)^\top x \leq \varepsilon$, that is, $\|\sum_{i\in I_T}\lambda_i m_i + m_k/\|m_k\|_2\| \leq \varepsilon$. This gives a ε-approximate conic Farkas certificate. For the running time, each von Neumann call requires $\lceil 1/\beta^2\rceil = O(n^2)$ calls to the separation oracle and $O(n^3)$ arithmetic operations [12, Lemma 8], so in particular $|J_t| \in O(n^2)$ at each iteration t. Hence the total number of oracle calls is $O(n^3\log(n/\varepsilon))$. Computing the new matrix R_t for $t = 1,\ldots,T$ requires time $O(n^2|J_t|) \in O(n^4)$, which gives $O(n^5\log(n/\varepsilon))$ arithmetic operations overall. □

7 Rational polyhedra

In this section, we consider rational polyhedra in the bit complexity model as in [19], and show some implications of our algorithms in this model. We denote by $\langle\alpha\rangle$ the binary encoding length of a rational number α, and $\langle D\rangle$ denote the encoding length of a matrix D, defined as follows: for an integer $n \neq 0$, $\langle n\rangle := \lceil\log_2|n|+1\rceil$; for a rational number $\alpha = p/q$ given as the ratio of co-prime integers with $q > 0$, $\langle\alpha\rangle := \langle p\rangle + \langle q\rangle$; and for a matrix $D \in \mathbb{Q}^{n\times n}$, $\langle D\rangle$ is the sum of the encoding length of all entries. Let us recall the definitions of facet and vertex complexity; see [19, Definition (6.2.2)].

Definition 7.1. *Let $P \subseteq \mathbb{R}^n$ be a polyhedron.*

(i) We say that P has facet-complexity at most φ*, if P can be defined by a system of linear inequalities with rational coefficients such that each inequality has encoding length at most φ. If $P = \mathbb{R}^n$, we require $\varphi \geq n+1$. The triple $(P; n, \varphi)$ is called a* well-described polyhedron*.*

(ii) We say that P has vertex-complexity at most ν*, if there exist finite sets of vectors $V_P, D_P \in \mathbb{Q}^n$ such that $P = \mathrm{conv}(V_P) + \mathrm{cone}(D_P)$, and all vectors in $V_P \cup D_P$ have encoding length at most at most ν. If $P = \emptyset$, then we require $\nu \geq n$.*

Dual solutions in the oracle model The results in [19] appear to be the only known methods in the literature for obtaining dual certificates of infeasibility and optimality in the oracle model. In this context, it is important to clarify which inequalities can be involved in the dual solution. In this respect, [19] considers two different concepts: *optimal standard dual solutions* and *optimal dual solutions with oracle inequalities.*

The former concept means that the dual solution assigns non-zero multipliers only to facet-defining inequalities for P, in some standard representation of P. Since there is typically no guarantee that the separation oracle returns facet-defining inequalities, it is not conceivable to be able to derive such certificates directly from the execution of the method, and indeed getting an optimal standard dual solution requires repeated applications of the ellipsoid method and the use of polarity (Theorem 6.5.14 in [19]).

The second concept, instead, requires that the nonzero entries of the dual solution correspond to inequalities that have been output by the oracle during the application of the method. Grötschel, Lovász, and Schrijver [19] point out that this is not possible in general. Assumption (4) on bounded encoding length of the oracle inequalities plays a key role in this context: under this assumption, one can obtain a dual certificate as follows (cf. Lemma 6.5.15 in [19]). We first run the (primal) ellipsoid method, obtaining a set $\mathcal{F}$ of oracle inequalities. Next, we "tighten" each inequality in $\mathcal{F}$, by another run of the ellipsoid per inequality. Finally, we apply the ellipsoid method to the dual LP, with the variable set corresponding to $\mathcal{F}$. Note that the dimension of this latter problem will be very large (albeit still polynomially bounded).

It has to be noted that, while assumption (4) is natural and applies widely in combinatorial optimization, it is not without loss of generality. A notable exception to assumption (4) is optimizing over the ϑ-body [19, Chapter 9]. In Section 7.2, we sketch how our method can be adapted to settings that do not satisfy assumption (4).

Bounding δ by φ The next lemma shows bounds on our condition number δ in terms of the bit-complexity.

Lemma 7.2. *If every row of a matrix $M \in \mathbb{R}^{m\times n}$ has bit-complexity at most φ, then $\delta_M \geq 1/2^{O(n^3\varphi)}$. Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ such that for all inequalities $a_i^\top x \leq b_i$, the vector $(a_i^\top, b_i)$ has bit-complexity at most φ. Then, $\delta_A, \delta_{(A,b)+(\mathbf{0},1)} \geq 1/2^{O(n^3\varphi)}$.*

Proof. The second part follows by the first by recalling that $\delta_{(A,b)+(\mathbf{0},1)} = \delta_M$ for the matrix M as in (1). Let us now show the first part; let $m_i^\top$ be the i-th row of M. Consider a set $\{m_i : i \in J\}$ of linearly independent vectors and nonzero coefficients $\lambda \in \mathbb{R}^J$; let $v = \sum_{i\in J} \lambda_i m_i$.

Let $B \in \mathbb{Q}^{n\times n}$ be a matrix whose first $|J|$ columns are the vectors m_i, $i \in J$, with arbitrary further $n-|J|$ unit vectors added such that B is non-singular. Then, for $x = B^{-1}v$, we have $x_i = \lambda_i$ for $i \in J$ and $x_j = 0$ otherwise. We have $\langle B\rangle = O(n\varphi)$, and consequently, $\langle B^{-1}\rangle \le 4n^2 \langle B\rangle = O(n^3\varphi)$ (see [19, Exercise 1.3.5(d)]). In particular, the norm of every row in B^{-1} is bounded by $2^{O(n^3\varphi)}$, and consequently, $|x_i| \le 2^{O(n^3\varphi)}\|v\|$. We also have $\|m_i\| \le 2^{O(\varphi)}$; consequently, $\max_{i\in J} |\lambda_i|\|m_i\|_2 \le 2^{O(n^3\varphi)}\|v\|$, showing the bound on δ_M. □

Together with Theorem 1.3, we yield the following.

Corollary 7.3. *Let $P = \{x \in \mathbb{R}^n : Ax \le b\}$, and assume we are given polyhedral separation oracles for both P as well as the recession cone $\mathrm{rec}(P) = \{x \in \mathbb{R}^n : Ax \le 0\}$. Further, assume for every inequality $a_i^\top x \le b_i$ in the system, $(a_i^\top, b_i)$ has bit-complexity at most φ. Then, there exists oracle polynomial algorithms for linear feasibility and linear optimization, also returning dual certificates, using $O(n^5\varphi)$ oracle calls.*

Remark 7.4. In order to obtain a polynomial algorithm in the bit-complexity model using an implementation of Approximate-Conic-Dual as in Section 2, one needs to provide an implementation of the subroutine that maintains rational numbers of bounded encoding length. This can be done by suitably rounding the coefficients encountered by the algorithms described in Section 6, but we do not elaborate the details here as the main focus of this paper is on the real model of computation.

7.1 Strong separation oracles with bounded bit complexity

The algorithms in [19] can return dual certificates with oracle inequalities under assumption (4). Recall that this assumption requires that the we have a strong separation oracle returning vectors whose encoding size is polynomially bounded by the facet complexity φ. However, in Corollary 7.3, we use a seemingly stronger polyhedral separation oracle model, where we require the oracle to return inequalities $a_i^\top x \le b_i$ such that (a_i, b_i) has encoding size polynomially bounded by φ. Furthermore, Corollary 7.3 also requires such separation oracle also for the recession cone.

In this section, we show that a strong separation oracle satisfying assumption (4) can be turned into polyhedral separation oracles of bounded bit-complexity. Assume that for the well-described polyhedron $(P; n, \varphi)$ with vertex complexity at most ν, and we have a strong separation oracle that for each $\bar{x} \in \mathbb{Q}^n$, returns a vector $a \in \mathbb{Q}^n$, such that $\max\{a^\top x : x \in P\} < a^\top \bar{x}$ with $\langle a\rangle = O(\varphi)$. This oracle returns the inequality $a^\top x < a^\top \bar{x}$, that is, the right hand side $a^\top \bar{x}$ depends on the point x queried, and a priori there could be an infinite number of potential inequalities returned.

Regarding the recession cone, [19, Lemma (6.4.8)] shows that the strong separation oracle of P implies a strong separation oracle for $\mathrm{rec}(P)$, returning inequalities $a^\top x \le 0$, where $a^\top x \le a^\top \bar{x}$ is an inequality that can be returned by the separation oracle for P. Hence, we obtain a polyhedral separation oracle fo $\mathrm{rec}(P)$, with the inequalities of encoding length φ.

Next, we show that a polyhedral separation oracle can be implemented for P by rounding the inequalities $a^\top x \le a^\top \bar{x}$ to $a^\top x \le b$ such that b has bit-complexity $O(\varphi + \nu)$. Thus, there exists a finite description $Ax \le b$ of P such that every inequality has encoding length $O(\varphi + \nu)$. Together with the rounding procedure, we can map the strong separation oracle to a polyhedral separation oracle with respect to this system, and therefore, Corollary 7.3 is applicable.

Let us note the following bounds relating facet- and vertex-complexity.

Lemma 7.5 ([19, (6.2.4)])**.** *Let $P \subseteq \mathbb{R}^n$ be a polyhedron.*

(i) If P has facet-complexity at most φ, then P has vertex-complexity at most $4n^2\varphi$.

(ii) If P has vertex-complexity at most ν, then P has facet-complexity at most $3n^2\nu$.

Rounding using the continued fractions method The key tool is the *continued fractions method*, an efficient algorithm for the following existential result by Legendre. We do not describe the method but summarize its properties in the next theorem, see [33, Section 6.1]

Theorem 7.6. *For given real number $\alpha \ge 0$ and integer $N > 0$, there exists at most one pair (p, q) of nonnegative integers such that $\left|\alpha - \frac{p}{q}\right| \le \frac{1}{2N^2}$ and $q \le N$. There exists an algorithm that, in $O(\log N)$ arithmetic operations, either finds such a pair (p, q), or concludes that no such pair exists. Further, if α is rational, then the space complexity of the algorithm is poly$(\langle\alpha\rangle, \log N)$.*

Lemma 7.7. *For given rational number $\alpha \geq 0$ and integer $\sigma > 0$, in $O(\sigma)$ iterations the continued fraction method finds the largest rational number $q \leq \alpha$ such that $\langle q \rangle \leq \sigma$. If α is rational, then the space complexity of the algorithm is poly$(\langle \alpha \rangle, \sigma)$.*

Proof. This follows as in the proof of [19, Theorem (5.1.9)]. The continued fractions method constructs a sequence of iterates $\beta_k = g_k/h_k$ such that h_k grows exponentially. Further, $\beta_{2k} \leq \alpha \leq \beta_{2k+1}$. Thus, there are $O(\sigma)$ iterates of size at most σ; the largest such β_{2k} provides the required answer. □

Consider now the well-described polyhedron $(P; n, \varphi)$ with vertex complexity at most ν, and a valid inequality $a^\top x \leq u$ such that $\langle a \rangle \leq \varphi$. Let b be the largest rational number $b < u$ such that $\langle b \rangle \leq \varphi + \nu$. Such a value b can be computed in polynomial time in $\langle u \rangle$ and $\varphi + \nu$ via continued fractions, as in Lemma 7.7. We claim that $a^\top x \leq b$ is a valid inequality for P. This follows since the linear function $a^\top x$ is bounded on P, and thus takes its maximum value on an extreme solution that has encoding length $\leq \nu$. Hence, the value $\max\{a^\top x : x \in P\}$ has encoding length $\leq \varphi + \nu$.

Remark 7.8. *Finally, let us remark that it is not necessary to apply the rounding procedure to every inequality returned by the oracle. We can postpone the roundings to the inequalities that participate in the ε-approximate Farkas certificates.*

7.2 Removing the assumption on the complexity of oracle inequalities

Let us now briefly sketch how one can (inefficiently) recover the [19] result without the simplifying assumption (4) within the polyhedral separation oracle model. In this context, one assumes that the bit-complexity of the facets of the polyhedron P have bit-size at most φ, but one only makes the assumption that the complexity of the output of the separation oracle grows polynomially as a function of the bit-size of the query point (i.e., it is not bounded by a fixed function of φ). To convert any such oracle to a polyhedral one with lower bounded δ, one may simply post-process each inequality outputted by the oracle using Diophantine approximation. Using the iterated Diophantine approximation method of Frank and Tardos [17], one can convert any valid inequality $a^\top x \leq b$ for P to a "nearby" valid inequality $\tilde{a}^\top x \leq \tilde{b}$ for P, where $\tilde{a}, \tilde{b}$ have bit-size poly(n, φ). More precisely, the closeness of $\tilde{a}x \leq \tilde{b}$ to $ax \leq b$ is formalized by saying that for any low-complexity point $\bar{x} \in \mathbb{Q}^n$, i.e., with bit-size poly(n, φ), satisfies $\text{sign}(b - a^\top x) = \text{sign}(\tilde{b} - \tilde{a}^\top x)$. In particular, if we maintain that we only query the oracle with inputs of bit-size at most poly(n, φ), which is relatively straightforward to achieve in most settings, the "post-processed" oracle behaves analoguously to a polyhedral separation oracle with lower-bounded δ. In particular, $\log 1/\delta$ will be polynomial in φ and n, using standard bit-complexity arguments.

One can make the above reduction more efficient by only lazily post-processing the "important" oracle inequalities, which would reduce to number of Diophantine approximations to $O(n)$, similar to [19]. As the details are technical and somewhat orthogonal to our main contributions, we defer a thorough presentation of this reduction to the full version of the paper.

8 Relating the circuit imbalance and δ

In this section, we investigate the relation between the condition number δ and the circuit imbalance measure κ. Lemma 8.2 shows bounds between these numbers; Corollary 8.3 shows that if A is in the form $A = (I|A')$, then $\delta_{A^\top}$ is comparable to κ_A. Using this, Theorem 8.4 adapts our conic validity algorithm to solving linear feasibility systems with κ_A dependence. Finally, in Section 8.1, we study optimal rescalings of δ. We use the following result on self-duality of κ_W.

Lemma 8.1 ([10])**.** *For every linear subspace $W \subseteq \mathbb{R}^n$ and the dual subspace $W^\top$, $\kappa_W = \kappa_{W^\top}$.*

The relation between δ and κ is as follows.

Lemma 8.2. *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with full row rank and $m < n$ with $\|a_i\| = 1$ for all columns $i \in [n]$. Let $\sigma_{\min}(A^\top)$ be the minimum singular value of $A^\top$ (that equals the minimum nonzero singular value of A). Then,*

$$\frac{\sigma_{\min}(A^\top)}{\sqrt{n}\delta_{A^\top}} \leq \kappa_A \leq \frac{1}{\delta_{A^\top}}.$$

Proof. Let us start with the upper bound on κ_A. Let $g \in \mathcal{E}(\ker(A))$ be an elementary vector. Select an arbitrary $i \in \mathrm{supp}(g)$, and let $J = \mathrm{supp}(g) \setminus \{i\}$. Then, the columns $\{g_j : j \in J\}$ are linearly independent, and $-g_i a_i = \sum_{j \in J} g_j a_j$. Thus,

$$|g_i| \cdot \|a_i\| = \left\| \sum_{j \in J} g_j a_j \right\| \geq \delta_{A^\top} \max_{j \in J} |g_j| \cdot \|a_j\| ,$$

and using that all columns have unit norm, we get $|g_j/g_i| \leq 1/\delta_{A^\top}$ for all $j \in J$. This shows that $\kappa_A \leq 1/\delta_{A^\top}$.

We now turn to the lower bound on κ_A. According to Lemma 2.3, there exists an index set $B \subseteq [n]$, $|B| = m$, and an index $k \in B$, such that A_B is non-singular, and $1/\delta_{A^\top}$ equals the norm of the k-th row of A_B^{-1}. This row of A_B^{-1} is the unique solution $z \in \mathbb{R}^m$ to the system $(A_B)^\top z = e_k^m$, where e_k^m is the k-th unique vector in $\mathbb{R}^m$.

Let us now consider the vector $y = A^\top z \in \mathbb{R}^n$; this is a vector in $\mathrm{im}(A^\top)$, and we claim that it is an elementary vector. Indeed, suppose a nonzero vector $y' \in \mathrm{im}(A^\top)$ exists with strictly smaller support. We have $y_i = 0$ for all $i \in B \setminus \{k\}$. If $y'_k = 0$, then the nonsingularity of A_B implies that $y' = 0$. If $y'_k \neq 0$, then we can normalize to $y'_k = 1$; but now $y'_B = y_B$, and again by the nonsingularity of A_B, we must have $y' = y$.

Let us use duality of κ_W (Lemma 8.1) for $W = \ker(A)$ and $W^\perp = \mathrm{im}(A^\top)$. Since $y_k = 1$, we obtain $\|y\|_\infty \leq \kappa_{W^\perp} = \kappa_W$. We thus get

$$\frac{\sigma_{\min}(A^\top)}{\delta_{A^\top}} = \sigma_{\min}(A^\top)\|z\| \leq \|A^\top z\| = \|y\| \leq \sqrt{n}\|y\|_\infty \leq \sqrt{n}\kappa_W .$$

□

Corollary 8.3. *Let $A \in \mathbb{R}^{m \times n}$ be of the form $A = (I_m | A')$. Then,*

$$\frac{1}{\delta_{A^\top}} \leq \sqrt{nm}\kappa_A^3 .$$

Proof. For any column a_i of A, $i \in [m+1, n]$, we have a circuit g in $\ker(A)$ defined as $g_j = -a_{ij}$ for $j \in [m]$, $g_i = 1$, and $g_j = 0$ otherwise. Hence, all nonzero entries of A are between $1/\kappa_A$ and κ_A, and therefore all column norms are between $1/\kappa_A$ and $\sqrt{m}\kappa_A$.

Let $\hat{A}$ be the matrix arising by normalizing all columns of A so that all columns have norm 1, and let us apply Lemma 8.2 to A. Note that, since A contains an identity matrix, $\sigma_{\min}(\hat{A}^\top) \geq 1$. Hence, $1/\delta_{\hat{A}^\top} \leq \sqrt{n}\kappa_{\hat{A}}$. Renormalization may increase the ratio between two entries of an elementary vector by at most $\sqrt{m}\kappa_A^2$; thus, the claim follows. □

Theorem 8.4. *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and consider the linear feasibility problem $Ax = b$, $x \geq 0$; assume $\mathrm{rk}(A) = m$. There exists an $O(nm^3 \log(n + \kappa_A) + m^5 \log\log(n + \kappa_A))$ time algorithm that either finds a feasible solution, or a Farkas certificate $A^\top y \geq 0$, $b^\top y < 0$.*

Proof. First, we can use Gaussian elimination to bring A to the basic form $A_B^{-1}A$ in $O(nm^2)$ time. To simplify the notation, we henceforth replace A by $A_B^{-1}A$ and b by $A_B^{-1}b$. Consider the conic validity problem for the vector b and the cone $K = \{y \in \mathbb{R}^m : A^\top y \geq 0\}$. The output is either a primal feasible solution or a Farkas certificate to our problem.

We obtain the claimed running time from Theorem 1.7 with the oracle as in Theorem 1.5 implementing the JLSW algorithm [22], and noting that every oracle call takes nm time by checking each inequality $a_i^\top y$. Finally, note that by Corollary 8.3, $\log(n/\delta_A) = O(\log(n + \kappa_A))$. □

8.1 Optimizing the measure δ

Consider a full-row rank matrix $A \in \mathbb{R}^{m \times n}$. As we have already observed, the measures κ_A and $\delta_{A^\top}$ are invariant under different rescalings: for any positive diagonal matrix $D \in \mathbb{R}^{n \times n}$, $\delta_{(AD)^\top} = \delta_{A^\top}$, but κ_{AD} might be different from κ_A. On the other hand, for any nonsingular $T \in \mathbb{R}^{m \times m}$, the opposite holds: $\kappa_{TA} = \kappa_A$, but $\delta_{(TA)^\top}$ and $\delta_{A^\top}$ may be different. One can naturally ask for the following quantities.

$$\begin{aligned} \kappa_A^* &:= \inf\left\{\kappa_{AD} : D \in \mathbb{R}^{n \times n} \text{ positive diagonal}\right\} \\ \delta_{A^\top}^* &:= \sup\left\{\delta_{(TA)^\top} : T \in \mathbb{R}^{m \times m} \text{ nonsingular}\right\} \end{aligned}$$

Lemma 8.5. *Given a matrix $A \in \mathbb{R}^{m\times n}$, $\delta^*_{A^\top} \le 1/\kappa^*_A$.*

Proof. Let D be the diagonal matrix with ith diagonal entry equal to $1/\|a_i\|$, so that all columns of AD have unit norm. For any nonsingular $T \in \mathbb{R}^{m\times m}$, Lemma 8.2 implies

$$\delta_{(TA)^\top} = \delta_{(TAD)^\top} \le \frac{1}{\kappa_{TAD}} = \frac{1}{\kappa_{AD}} \le \frac{1}{\kappa^*_A}\,.$$

□

The quantity κ^*_A was studied in [10]; they gave a min-max characterization, as well as the following algorithmic result:

Theorem 8.6 ([10])**.** *Given a matrix $A \in \mathbb{R}^{m\times n}$, in $O(n^2m^2 + n^3)$ time one can find a positive diagonal matrix D such that $\kappa_{AD} \le (\kappa^*_A)^3$.*

As a consequence, all results with running time dependence on $\log(\kappa_A + n)$ can be strengthened to dependence on $\log(\kappa^*_A + n)$. Note that the approximability result is in surprising contrast with that of Tunçel [39] who showed that $\bar\chi_A$ (and consequently, κ_A) cannot be approximated within a factor $2^{\mathrm{poly}(m)}$.

Combined with Corollary 8.3, we obtain the following result on $\delta^*_{A^\top}$.

Corollary 8.7. *For a matrix $A \in \mathbb{R}^{m\times n}$, in $O(n^2m^2 + m^3)$ time we can find a nonsingular $T \in \mathbb{R}^{m\times m}$ such that*

$$\frac{(\delta^*_{A^\top})^9}{\sqrt{nm}} \le \delta_{(TA)^\top}\,.$$

Proof. Let us compute the near optimal rescaling D such that $\kappa_{AD} \le (\kappa^*_A)^3$ as in Theorem 8.6. Compute any non-singular $m \times m$ matrix B of AD, and return $T = B^{-1}$. We show that T satisfies the statement. Indeed, TAD contains an $m \times m$ identity matrix, therefore, by Corollary 8.3, we get

$$\delta_{(TA)^\top} = \delta_{(TAD)^\top} \ge \frac{1}{\sqrt{nm}(\kappa_{TAD})^3} = \frac{1}{\sqrt{nm}(\kappa_{AD})^3} \ge \frac{1}{\sqrt{nm}(\kappa^*_A)^9} \ge \frac{(\delta^*_{A^\top})^9}{\sqrt{nm}}\,,$$

where the last inequality follows from Lemma 8.5. □

Note that we are only able to use the above renormalization for an explicitly given matrix A, but not in the oracle model.

A Impossibility results

We now turn to the proofs of Proposition 1.1 and 1.4 showing the impossibility of strongly polynomial algorithms in the oracle model.

Proof of Proposition 1.1. Let $\mathcal{P}_m$ denote the set of polytopes $P \subseteq \mathbb{R}^2$ of the form

$$P = \left\{(x_1, x_2) \in \mathbb{R}^2 : \frac{x_1}{p_i} + \frac{x_2}{q_i} \le 1, i \in [m], (x_1, x_2) \ge 0\right\}, \tag{43}$$

for some numbers $p_1 > p_2 > \ldots > p_m > 0$, $0 < q_1 < q_2 < \ldots < q_m$. For any such choice of parameters, these define a full-dimensional polytope with $m + 2$ facets. Let $\mathcal{P}_m$ denote the class of all polytopes of this form.

Consider the problem $\max\ x_1$, $x \in P$ for such a polytope $P \in \mathcal{P}_m$. Clearly, the optimal solution is $(p_m, 0)$. We claim that an adversary can answer oracle queries such that, for any prescribed $m^* \in \mathbb{N}$, any algorithm will require at least m^* queries to compute an optimal solution. This proves the claim, since m^* can be chosen independently from $n = 2$.

The adversary strategy is as follows. At the current iteration, say iteration m, they maintain two polytopes $P_m \in \mathcal{P}_m$ and $Q_m \subseteq P_m$, starting with $m = 0$, $P_0 = \mathbb{R}^2_+$, $Q_0 = \{0\}$. For $m \ge 1$, the polyhedron P_m is given in the form (43) with $p_1 > p_2 > \ldots > p_m > 0$ and $0 < q_1 < q_2 < \ldots < q_m$. For $m = 1$ we let $z = (0, q_1)$, and for $m > 1$ we let $z = (z_1, z_2)$ be the point corresponding to the intersection of the last two separators $x_1/p_{m-1} + x_2/q_{m-1} = 1$ and $x_1/p_m + x_2/q_m = 1$. For $1 \le m < m^*$, we let $Q_m = P_m \cap \{(x_1, x_2) \in \mathbb{R}^2 : x_1 \le z_1\}$.

In each iteration, if the algorithm queries a point $x \in \mathbb{R}^2$ that is in Q_m, the oracle responds that the point is feasible. If $x \notin P_m$, then we return one of the facet defining inequalities of P_m that separates x. Finally, if $x \in P_m \setminus Q_m$, then we return a new inequality $x_1/p_{m+1} + x_2/q_{m+1} \le 1$ with

$0 < p_{m+1} < p_m$ and $q_{m+1} > q_m$; it is easy to see that such an inequality can always be added. We obtain P_{m+1} from P_m by adding this new inequality, update Q_{m+1}, and increase m by one. Once we reach $m = m^*$, we set $Q_m = P_m$.

Note that this oracle strategy maintains $P_{m+1} \subset P_m$, $Q_{m+1} \supset Q_m$, and all facets of P_m are also facets of P_{m+1}. Thus, all previous separators returned by the oracle are valid to P_{m+1}. Moreover, m gives a lower bound on the total number of oracle queries. □

Proof of Proposition 1.4. The proof is similar to the previous one, but even simpler, in 1-dimension. Consider $b \in \mathbb{R}^m$ with values $b_1 > b_2 > \ldots > b_m \in \mathbb{R}$. Let $A \in \mathbb{R}^{m \times 1}$ be a matrix with all entries 1. Let $P = \{x_1 \in \mathbb{R} : Ax_1 \leq b\}$; thus, $P = (-\infty, b_m]$. Consider the problem of max x_1 for $x_1 \in P$; the optimal solution is $x_1^* = b_m$. However, m oracle queries are necessary: Similarly to the previous proof, the adversary answering the oracle queries can always maintain P in this form such that the first k oracle queries are $x_1 \leq b_i$, $i \in [k]$.

Note that all rows of the matrix A are identical, hence, any condition number θ_A unchanged by duplicating copies must be the same as for the 1×1 matrix $A' = (1)$. Therefore, $f(n, \theta_A)$ is the same for all instances in this form, showing that no algorithm may terminate in $f(n, \theta_A)$ queries. □

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.

[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] D. Dadush, L. A. Végh, and G. Zambelli. Geometric rescaling algorithms for submodular function minimization. *Mathematics of Operations Research*, 2021. (to appear).

[14] G. B. Dantzig. An ε-precise feasible solution to a linear program with a convexity constraint in $1/\varepsilon^2$ iterations independent of problem size. Technical report, Technical Report 92-5, Stanford University, 1992.

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

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

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

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

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.).

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

[26] 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.

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

[28] 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.

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

[30] 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.

[31] 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.).

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

[33] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, New York, 1998.

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

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

[36] 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.

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

[38] J. F. Traub and H. Woźniakowski. Complexity of linear programming. *Operations Research Letters*, 1(2):59–62, 1982.

[39] L. Tunçel. Approximating the complexity measure of Vavasis-Ye algorithm is NP-hard. *Mathematical Programming*, 86(1):219–223, Sep 1999.

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

[41] 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.

[42] 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.

[43] 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.