Frédéric Meunier

Linear Programming: an introduction

February 16, 2026

CERMICS

Contents

CHAPTER 1

Overview

1.1 Introduction

1.1.1 Linear programming, what for?

The problem

Optimizing a linear criterion on a set of points defined by linear equalities and inequalities. Without loss of generality:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geqslant \boldsymbol{b} \,. \end{aligned} \tag{P}$$

A *polyhedron* is the intersection of finitely many closed half-spaces (bounded by hyperplanes). So linear programming is exactly the same thing as optimizing a linear criterion on a polyhedron.

Applications

- Direct industrial applications (transportation, production, energy, etc.).

  Example. m resources, n products. Producing one unit of j requires a_{ij} units of i, and generates a revenue of c_j. Maximize the revenue, under constraint resources.

- A bit more than linearity. maximum, absolute value, convex piecewise affine objective functions. Examples:

  - Finding the hyperplane minimizing the L_1- or the L_∞-distance to a collection of points.
  - A production problem, with inventory costs, and production costs where variation is penalized.

- Integer polytopes, e.g., matchings in bipartite graph.

- Integer linear programming. Useful to solve "integer linear programming": bounds, TSP

For instance, on an instance of more that 1.3 billions stars from the galaxy, a tour within 0.38% of optimality has been recently computed [1]. Proving this optimality gap relies on linear programming.

1.1.2 History

Fourier (1827): a method for solving system of inequality constraints (not very efficient). This method is now known as the “Fourier–Motzkin” elimination (used for theory).

Kantorovich (1939) and Koopmans (1942): Nobel prize 1975 (in economics). Pioneers of tranportation problems. Kantorovich was motivated by concrete applications to the production of plywood. Koopmans considered duality.

Dantzig (1947): simplex algorithm.

Dantzig (1948): strong duality (suggested by Von Neumann).

Khachyian (1979): ellipsoids, linear programming is polynomial.

Karmarkar (1984): interior-points methods, practically efficient polynomial algorithm for linear programming.

1.1.3 Algorithms

Several algorithms have been proposed to solve linear programs. Simplex, interior-point methods, ellipsoids are the most famous.

$10,000$ variables and constraints can be solved within seconds with best solvers on a standard laptop, and within one hour when dealing with integer variables. (Very rough numbers, and depend highly on the structure of the problem.)

1.2 Main results

1.2.1 The Minkowski–Weyl theorem and consequences

The *convex hull* of a finite set X of points is

$$\operatorname{conv}(X) \coloneqq \left\{ \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \boldsymbol{x} \colon \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} = 1,\, \lambda_{\boldsymbol{x}} \geqslant 0 \right\}.$$

The *positive hull* of a finite set X of points is

$$\operatorname{pos}(X) \coloneqq \left\{ \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \boldsymbol{x} \colon \lambda_{\boldsymbol{x}} \geqslant 0 \right\}.$$

Theorem 1.1 (Minkowski–Weyl)**.** *Let P be a polyhedron of $\mathbb{R}^n$. Then there exit two finite sets $V, R \subseteq \mathbb{R}^n$ such that $P = \operatorname{conv}(V) + \operatorname{pos}(R)$.*

The proof relies on the following lemma, which is actually a special case.

Lemma 1.2. *Given a matrix M, we have $\{\boldsymbol{x} \colon M\boldsymbol{x} \geqslant \mathbf{0}\} = \operatorname{pos}(S)$ for some finite set S of points.*

Proof. Up to working on a supplementary space of $\ker M$, we can assume that the columns of M are linearly independent. For a subset I of the rows of M, we denote by M^I the matrix obtained from M by keeping the rows from I. The set of subsets I such that M^I has corank equal to 1 is denoted by $\mathcal{I}$. For each I, there exists $\boldsymbol{r}^I \neq \boldsymbol{0}$ (unique up to the sign) such that $M^I = \boldsymbol{0}$. Write $\mathcal{I}^+ := \{I \in \mathcal{I} \colon M\boldsymbol{r}^I \geqslant \boldsymbol{0}\}$. We claim that $\{\boldsymbol{x} \colon M\boldsymbol{x} \geqslant \boldsymbol{0}\} = \operatorname{pos}(S)$ for $S := \{r^I \colon I \in \mathcal{I}^+\}$. This is the purpose of the rest of the proof. One direction being obvious, we consider $\boldsymbol{x} \neq \boldsymbol{0}$ such that $M\boldsymbol{x} \geqslant \boldsymbol{0}$, and we are going to prove that $\boldsymbol{x} \in \operatorname{pos}(S)$.

Let $\boldsymbol{r}$ be such that $M\boldsymbol{r} \geqslant \boldsymbol{0}$ and such that the set $\operatorname{supp}(M\boldsymbol{r})$ is minimal for inclusion and included in $\operatorname{supp}(M\boldsymbol{x})$. Denote by J the rows of M orthogonal to $\boldsymbol{r}$, i.e., $M^J\boldsymbol{r} = \boldsymbol{0}$, and there is no strict superset of J with the same property. If there were a vector in $\ker(M^J)$ linearly independent from $\boldsymbol{r}$, then we could build $\boldsymbol{r}'$ such that $M\boldsymbol{r} \geqslant \boldsymbol{0}$, and with $\operatorname{supp}(M\boldsymbol{r}')$ strictly included in $\operatorname{supp}(M\boldsymbol{r})$, which is not possible by definition of $\boldsymbol{r}$. Thus, the corank of M^J is one. There exits $\lambda > 0$ such that $\operatorname{supp}(\boldsymbol{x} - \lambda\boldsymbol{r})$ is a strict subset of $\operatorname{supp}(\boldsymbol{x})$. By repeating this argument, we get

$$M\Big(\boldsymbol{x} - \sum_{\boldsymbol{s} \in S} \lambda_{\boldsymbol{s}} \boldsymbol{s}\Big) = \boldsymbol{0}\,,$$

for some $\lambda_{\boldsymbol{s}} \geqslant 0$. The desired conclusion follows from the linear independent of the columns of M. ☐

Proof of Theorem 1.1. Let $P = \{\boldsymbol{x} \colon A\boldsymbol{x} \geqslant \boldsymbol{b}\}$. Apply Lemma 1.2 to $M = \left(\begin{smallmatrix} A & -\boldsymbol{b} \\ \boldsymbol{0}^\top & 1 \end{smallmatrix}\right)$, we get the existence of S such that for every $\boldsymbol{x} \in P$ we have

$$\begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} = \sum_{\boldsymbol{s} \in S} \lambda_{\boldsymbol{s}} \boldsymbol{s}\,,$$

for some $\lambda_{\boldsymbol{s}} \geqslant 0$. Up to some scaling, we suppose w.l.o.g. that the last component of every $\boldsymbol{s} \in S$ is either 0, or 1. Let V' be the points in S with a last component equal to 1, and let R' be those with a last component equal to 0. Then setting V (resp. R) as the points of V' (resp. R') from which the last component has been removed, we get that $P \subseteq \operatorname{conv}(V) + \operatorname{pos}(R)$.

Conversely, since P is feasible, we have $V \neq \varnothing$, and let $\boldsymbol{x} \in \operatorname{conv}(V)$ and $\boldsymbol{z} \in \operatorname{pos}(R)$ (note that this latter set is always non-empty). Then, by construction, $A\boldsymbol{x} \geqslant \boldsymbol{b}$ and $A\boldsymbol{z} \geqslant \boldsymbol{0}$, which implies that $\boldsymbol{x} + \boldsymbol{z} \in P$. ☐

Moreover, the proof makes clear the following fact: when P contains no infinite line (we say it is *pointed*), the elements of V can be chosen to be of the form $(A')^{-1}\boldsymbol{b}$ for some non-singular submatrix A' of A. This has several important consequences. Let us mention one here. A *vertex* of P is a point of P that cannot be written as the convex combination of two distinct points of P. Suppose that P is pointed; then the elements in V can be chosen to be the vertices of P.

The Minkwoski–Weyl theorem can be used to establish a result, which is sometimes called "the fundamental theorem of linear programming."

Theorem 1.3. *Suppose* (P) *is feasible and has its objective function lower bounded on the set of its feasible solutions. Then* (P) *admits an optimal solution. If moreover the polyhedron formed by the feasible solutions is pointed, then there always exists a vertex of the polyhedron that is an optimal solution.*

The fact that the optimal value is attained is a real specificity of linear programming: consider for instance the minimization of $1/x$ over $[1, +\infty)$ to see that in general this might not hold.

Proof of Theorem 1.3. Denote by P the set of feasible solutions. Let V and R as in Theorem 1.1. If $\boldsymbol{c} \cdot \boldsymbol{r} < 0$ for some $\boldsymbol{r} \in R$, then clearly the objective function would be unbounded. So $\boldsymbol{c} \cdot \boldsymbol{r} \geqslant 0$ for all $\boldsymbol{r} \in R$. Consider

$\boldsymbol{v}^\star \in V$ minimizing $\boldsymbol{c} \cdot \boldsymbol{v}$ on V. By convexity, $\boldsymbol{c} \cdot \boldsymbol{v}^\star \leqslant \boldsymbol{c} \cdot \boldsymbol{x}$ for all $\boldsymbol{x} \in \operatorname{conv}(V)$. Since $\boldsymbol{c} \cdot \boldsymbol{r} \geqslant 0$ for all $\boldsymbol{r} \in R$, we have further $\boldsymbol{c} \cdot \boldsymbol{v}^\star \leqslant \boldsymbol{c} \cdot \boldsymbol{x}$ for all $\boldsymbol{x} \in P$.

For the case when P is pointed, the statement results from the remark above: V can then be chosen as the set of vertices of P. □

1.2.2 Duality

Consider

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{b} \cdot \boldsymbol{y} \\ \text{s.c.} \quad & A^\top \boldsymbol{y} = \boldsymbol{c} \\ & \boldsymbol{y} \geqslant \boldsymbol{0}\,. \end{aligned} \tag{D}$$

The motivation comes the following fact: every $\boldsymbol{y}$ that is a feasible solution of (D) provides a lower bound on the optimal value of (P). Indeed, consider $\boldsymbol{x}$ satisfying $A\boldsymbol{x} \geqslant \boldsymbol{b}$; multiplying this system by such an $\boldsymbol{y}$ leads to $\boldsymbol{c} \cdot \boldsymbol{x} \geqslant \boldsymbol{b} \cdot \boldsymbol{y}$. In other words, the optimal value of (P) is lower bounded by the optimal value of (D). This is 'weak duality.

The linear program (D) models the problem of finding such an $\boldsymbol{y}$ providing the best lower bound. Except in pathological cases, the best bound we get this way is tight. This is *strong duality* and it is the message of the next theorem.

Theorem 1.4. *If (P) or (D) is feasible then, their optimal values are equal.*

Strong duality is a consequence of Farkas's lemma, itself being a consequence of separation.

Lemma 1.5 (Farkas lemma)**.** *Exactly one of the following occurs:*

- *There exists $\boldsymbol{u}$ such that $M\boldsymbol{u} \geqslant \boldsymbol{p}$.*
- *There exists $\boldsymbol{v} \geqslant \boldsymbol{0}$ such that $M^\top \boldsymbol{v} = \boldsymbol{0}$ and $\boldsymbol{p} \cdot \boldsymbol{v} = 1$.*

Proof. Suppose that there is no $\boldsymbol{u}$ such that $M\boldsymbol{u} \geqslant \boldsymbol{p}$, and consider $C = \{\boldsymbol{z} \colon \exists \boldsymbol{u} \text{ s.t. } \boldsymbol{z} \leqslant M\boldsymbol{u}\}$. By the separation theorem, there exist $\boldsymbol{v}$ and γ such that $\boldsymbol{v} \cdot \boldsymbol{z} \leqslant \gamma$ for all $\boldsymbol{z} \in C$ and $\boldsymbol{p} \cdot \boldsymbol{v} > \gamma$. Because C is unbounded, we can actually choose γ to be 0. Using again unboundedness, we see that necessarily $\boldsymbol{v} \geqslant \boldsymbol{0}$. By a simple scaling operation, we get $\boldsymbol{p} \cdot \boldsymbol{v} = 1$, as desired.

The reverse implication is immediate. □

Proof of Theorem 1.4. If the optimal value of (P) is $-\infty$ or the optimal value of (D) is $+\infty$, then the theorem results easily from weak duality. So it is sufficient to consider the case where (D) has a finite optimal value. (The case where (P) has a finite value will be a consequence, since the dual of the dual is the primal.) Denote by α the optimal value of (D). Let $M \coloneqq \left(\begin{smallmatrix} A \\ -\boldsymbol{c}^\top \end{smallmatrix}\right)$ and $\boldsymbol{p} \coloneqq \left(\begin{smallmatrix} \boldsymbol{b} \\ -\alpha \end{smallmatrix}\right)$.

Suppose for a contradiction that there exists $\boldsymbol{v} = \left(\begin{smallmatrix} \boldsymbol{y} \\ w \end{smallmatrix}\right) \geqslant \boldsymbol{0}$ such that $M^\top \boldsymbol{v} = 0$ and $\boldsymbol{p} \cdot \boldsymbol{v} = 1$. Denote by $\bar{\boldsymbol{y}}$ a feasible solution of (D). It exists since the optimal value of (D) is finite. If $w = 0$, then we would have $\boldsymbol{y} \geqslant \boldsymbol{0}$ such that $A^\top \boldsymbol{y} = \boldsymbol{0}$ and $\boldsymbol{b} \cdot \boldsymbol{y} = 1$, which would imply that $\bar{\boldsymbol{y}} + t\boldsymbol{y}$ is a feasible solution of (D) for all $t > 0$, which would in turn imply that the optimal value of (D) is $+\infty$; a contradiction. So $w > 0$, but this implies that there exists a feasible solution of (D) giving to the objective function a value larger than α; again a contradiction. Thus there is no such $\boldsymbol{v}$, and Farkas's lemma implies that there exists $\boldsymbol{u} = \boldsymbol{x}$ such that $A\boldsymbol{x} \geqslant \boldsymbol{b}$ and $\boldsymbol{c} \cdot x \leqslant \alpha$. Weak duality leads then to the desired conclusion. □

1.2.3 Polynomial solvability

1.3 Applications and special cases

1.3.1 Zero-sum games

A nice application of strong duality is the following theorem, one of the first results in game theory. It was established by Von Neumann in 1928 and popularized as the miminam theorem. Denote by $\triangle^k$ the set of probabilities on $[k]$:

$$\triangle^k := \left\{\boldsymbol{x} \in \mathbb{R}_+^k \colon \sum_{i=1}^k x_i = 1\right\}.$$

Then this theorem states:

Theorem 1.6 (Von Neumann minimax theorem)**.** *For every $m \times n$ matrix A, there exists $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \triangle^m \times \triangle^n$ such that the following holds:*

$$\boldsymbol{x}^\top A\boldsymbol{y}^\star \geqslant (\boldsymbol{x}^\star)^\top A\boldsymbol{y}^\star \geqslant (\boldsymbol{x}^\star)^\top A\boldsymbol{y} \qquad \forall(\boldsymbol{x},\boldsymbol{y}) \in \triangle^m \times \triangle^n .$$

It is then an easy exercise to show furthermore that

$$(\boldsymbol{x}^\star)^\top A\boldsymbol{y} = \min_{\boldsymbol{x}\in\triangle^m} \max_{\boldsymbol{y}\in\triangle^n} \boldsymbol{x}^\top A\boldsymbol{y} = \max_{\boldsymbol{y}\in\triangle^n} \min_{\boldsymbol{x}\in\triangle^m} \boldsymbol{x}^\top A\boldsymbol{y} ,$$

which legitimates the name of this theorem.

Proof of Theorem 1.6. We start by adding a big positive constant to every entry of A, making all entries of A positive. Note that proving the theorem for this new matrix A implies it also holds for the original matrix A.

Consider the following primal-dual pair of linear programs, where $\mathbf{1}$ denotes the all-one vector:

$$\begin{array}{lll} \text{maximize} & \mathbf{1}\cdot\boldsymbol{x} & \\ \text{s.c.} & A^\top\boldsymbol{x} \leqslant \mathbf{1} & \text{(P')} \\ & \boldsymbol{x} \geqslant \mathbf{0}. & \end{array} \qquad\qquad \begin{array}{lll} \text{minimize} & \mathbf{1}\cdot\boldsymbol{y} & \\ \text{s.c.} & A\boldsymbol{y} \geqslant \mathbf{1} & \text{(D')} \\ & \boldsymbol{y} \geqslant \mathbf{0}. & \end{array}$$

Notice that (P') admits $\mathbf{0}$ as a feasible solution, and that it is bounded because all entries of A are positive. By Theorem 1.4, (P') and (D') both admit optimal solutions, which we denote respectively by $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$. Since $\mathbf{0}$ is not a feasible solution of (D'), the optimal value of the programs is a positive number, which we denote by α. Set

$$\boldsymbol{x}^\star := \frac{1}{\alpha}\bar{\boldsymbol{x}} \quad \text{and} \quad \boldsymbol{y}^\star := \frac{1}{\alpha}\bar{\boldsymbol{y}} .$$

Note that $\boldsymbol{x}^\star$ belongs to $\triangle^m$ and $\boldsymbol{y}^\star$ belongs to $\triangle^n$, as required. The constraints of (P') show that $A^\top\boldsymbol{x}^\star \leqslant \frac{1}{\alpha}\mathbf{1}$, and the constraints of (D') show that $A\boldsymbol{y}^\star \leqslant \frac{1}{\alpha}\mathbf{1}$. Multiplying the first system of inequalities by $\boldsymbol{y}$ and the second system of inequalitites by $\boldsymbol{x}$ shows that

$$(\boldsymbol{x}^\star)^\top A\boldsymbol{y} \leqslant \frac{1}{\alpha} \quad \forall\boldsymbol{y} \in \triangle^n \qquad \text{and} \qquad \boldsymbol{x}^\top A\boldsymbol{y}^\star \geqslant \frac{1}{\alpha} \quad \forall\boldsymbol{x} \in \triangle^m .$$

We get thus the desired conclusion, noticing further that the above inequalitites imply the equality $(\boldsymbol{x}^\star)^\top A\boldsymbol{y}^\star = \frac{1}{\alpha}$. ∎

1.3.2 Flows

1.3.3 TU matrices

1.4 Open questions

1.4.1 Strong polynomiality

Even though the ellipsoid method and the interior-point method are polynomial, they are not *strongly* polynomial, which means that the number of iterations does not admit a polynomial bound depending only on the number of rows and columns of the linear program: the number of iterations depends also on the entries of the program. Whether there exists a strongly polynomial algorithm solving linear programming is the current big open question of the field.

1.4.2 Diameter of polytopes

CHAPTER 2

The ellipsoid method

Even though the simplex algorithm is very efficient in practice, it suffers from a serious drawback: it is not known to be a polynomial algorithm, in the sense that for most pivot rules, there are inputs for which the number of iterations is exponential, and for the other pivot rules, it is not known whether it runs in polynomial time. In the 60's and the 70's—period where the complexity of algorithms has started to be considered—the question whether there exists a polynomial algorithm for solving linear programs has become a major open problem in optimization.

In 1979, Khachyian solved this question by describing a polynomial algorithm for solving linear programs: the *ellipsoid algorithm*. This algorithm has actually another interesting feature, with deep theoretical consequences, and which will be discussed: its execution does not need an explicit list of the constraints. However, nobody has ever succeeded in proposing an efficient implementation of the ellipsoid algorithm, which makes of it essentially a theoretical object.

2.1 The algorithm

The ellipsoid algorithm actually solves the following problem: given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$, decide whether there exists $\boldsymbol{x} \in \mathbb{R}^n$ such that $A\boldsymbol{x} \geqslant \boldsymbol{b}$. This is the "feasibility" version of linear programming. Solving this latter is enough to solve linear programming in full generality. Indeed, given for instance a linear program in the inequality form

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \geqslant \boldsymbol{b}\,, \end{array}$$

a point $\boldsymbol{x}^\star$ is an optimal solution if and only if it forms with some $\boldsymbol{y}^\star$ a feasible solution of the following system of equalities and inequalities:

$$\begin{aligned} & A\boldsymbol{x} \geqslant \boldsymbol{b} \\ & A^\top \boldsymbol{y} = \boldsymbol{c} \\ & \boldsymbol{y} \geqslant \boldsymbol{0} \\ & \boldsymbol{c} \cdot \boldsymbol{x} - \boldsymbol{b} \cdot \boldsymbol{y} = 0\,. \end{aligned}$$

This is a direct consequence of strong duality in linear programming. To describe the algorithm, the following notation will be useful. Denote by $\operatorname{vol}(K)$ the volume of a convex body K, and by $E(\boldsymbol{s}, Q)$ the ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \colon (\boldsymbol{x} - \boldsymbol{s})^\top Q^{-1}(\boldsymbol{x} - \boldsymbol{s}) \leqslant 1\}$ (note that $\boldsymbol{s}$ is its center and that Q is definite positive). For integers p, q, we set $\langle p/q \rangle := \lceil \log_2(p+1) \rceil + \lceil \log_2(q+1) \rceil + 1$; this is the number of bits needed to encode the rational number p/q.

To avoid unnecssary discussion, we assume in the remaining of this section that $n \geqslant 2$. The algorithm applies on systems $A\boldsymbol{x} \geqslant \boldsymbol{b}$ for which there exists $R > 0$ such that $\{\boldsymbol{x} \in \mathbb{R}^n \colon A\boldsymbol{x} \geqslant \boldsymbol{b}\}$ is included in $\bar{B}(\boldsymbol{0}, R)$ (the closed ball of radius R centered at $\boldsymbol{0}$). It consists then in the following algorithm:

Input : A matrix A (with m rows and n columns), a vector $\boldsymbol{b} \in \mathbb{R}^m$, a positive number R such that $\{\boldsymbol{x} \in \mathbb{R}^n \colon A\boldsymbol{x} \geqslant \boldsymbol{b}\} \subseteq \bar{B}(\boldsymbol{0}, R)$, a positive number ε
Output : An $\boldsymbol{x}$ such that $A\boldsymbol{x} \geqslant \boldsymbol{b}$ or the assertion that $\operatorname{vol}(\{\boldsymbol{x} \in \mathbb{R}^n \colon A\boldsymbol{x} \geqslant \boldsymbol{b}\}) \leqslant \varepsilon$

$E_0 \leftarrow \bar{B}(\boldsymbol{0}, R)$;
$\boldsymbol{s}^0 \leftarrow \boldsymbol{0}$;
$k \leftarrow 0$;
while $A\boldsymbol{s}^k \not\geqslant \boldsymbol{b}$ *or* $\operatorname{vol}(E_k) > \varepsilon$ **do**
  Choose i such that $\boldsymbol{a}^i \cdot \boldsymbol{s}^k < b_i$;
  Find an ellipsoid E_{k+1} such that $E_k \cap \{\boldsymbol{x} \in \mathbb{R}^n \colon \boldsymbol{a}^i \cdot \boldsymbol{x} \geqslant \boldsymbol{a}^i \cdot \boldsymbol{s}^k\} \subseteq E_{k+1}$ and $\operatorname{vol}(E_{k+1}) \leqslant e^{-\frac{1}{2(n+1)}} \operatorname{vol}(E_k)$;
  $\boldsymbol{s}^{k+1} \leftarrow$ center of E_{k+1};
  $k \leftarrow k+1$;
end
if $A\boldsymbol{s}^k \geqslant \boldsymbol{b}$ **then**
  return $\boldsymbol{s}^k$
else
  return *"volume of polytope smaller than ε"*
end

Algorithme 1 : Ellipsoid algorithm

It generates a sequence $E_0, E_1, \ldots$ of ellipsoids, of decreasing volume, each containing $\{\boldsymbol{x} \in \mathbb{R}^n \colon A\boldsymbol{x} \geqslant \boldsymbol{b}\}$.

Two things are not specified in the algorithm: how to compute the volume of an ellipsoid and how to find the ellipsoid E_{k+1}. The first thing is easy: the volume of an ellipsoid $E(\boldsymbol{s}, Q)$ is given by $\det(Q^{1/2}) \operatorname{vol}(\bar{B}(\boldsymbol{0}, 1))$ (classical and easy result). The second thing relies on the following lemma, which gives the general expression of an ellipsoid containing the intersection of another ellipsoid with some specific half-space. It turns out—but we do not prove this here—that this ellipsoid is nothing else than the Löwner–John ellipsoid of this intersection. (The *Löwner–John ellipsoid* of a convex body is the unique ellipsoid of minimal volume containing it.)

Lemma 2.1. *Let Q be an $n \times n$ definite positive matrix and let $\boldsymbol{a}, \boldsymbol{s} \in \mathbb{R}^n$. Then $E(\boldsymbol{s}', Q')$ contains $E(\boldsymbol{s}, Q) \cap \{\boldsymbol{x} \in \mathbb{R}^n \colon \boldsymbol{a} \cdot \boldsymbol{x} \geqslant \boldsymbol{a} \cdot \boldsymbol{s}\}$ where*

$$\boldsymbol{s}' := \boldsymbol{s} + \frac{1}{n+1}\boldsymbol{\omega} \quad \text{and} \quad Q' := \frac{n^2}{n^2-1}\Big(Q - \frac{2}{n+1}\boldsymbol{\omega}\boldsymbol{\omega}^\top\Big) \qquad \text{with} \quad \boldsymbol{\omega} := \frac{1}{\sqrt{\boldsymbol{a}^\top Q \boldsymbol{a}}} Q\boldsymbol{a}\,.$$

Moreover, we have $\operatorname{vol}(E(\boldsymbol{s}', Q')) \leqslant e^{-\frac{1}{2(n+1)}} \operatorname{vol}(E(\boldsymbol{s}, Q))$.

Proof. We prove first that every $\boldsymbol{x}$ such that $(\boldsymbol{x}-\boldsymbol{s})^\top Q^{-1}(\boldsymbol{x}-\boldsymbol{s}) \leqslant 1$ and $\boldsymbol{a}\cdot(\boldsymbol{x}-\boldsymbol{s}) \geqslant 0$ satisfies $(\boldsymbol{x}-\boldsymbol{s}')^\top (Q')^{-1}(\boldsymbol{x}-\boldsymbol{s}') \leqslant 1$. Performing an affine transformation, this amounts to consider any $\boldsymbol{y}$ such that $\|\boldsymbol{y}\|_2 \leqslant 1$ and $\boldsymbol{u}\cdot\boldsymbol{y} \geqslant 0$ and to show that it satisfies $(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u})^\top M^{-1}(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u}) \leqslant 1$, where

$$M := \frac{n^2}{n^2-1}\Big(I_n - \frac{2}{n+1}\boldsymbol{u}\boldsymbol{u}^\top\Big)\,,$$

where $\boldsymbol{u} := \frac{1}{\|\boldsymbol{a}\|_2}\boldsymbol{a}$. For such an $\boldsymbol{y}$, write $\boldsymbol{y} = \lambda\boldsymbol{u} + \boldsymbol{z}$, with $\boldsymbol{z}$ orthogonal to $\boldsymbol{u}$ and with $\lambda \in [0,1]$. Notice that

$$M^{-1} = \frac{n^2-1}{n^2}\Big(I_n + \frac{2}{n-1}\boldsymbol{u}\boldsymbol{u}^\top\Big)\,.$$

(This also shows that M is definite positive.) We have then

$$\Big(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u}\Big)^\top M^{-1}\Big(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u}\Big) = \frac{n^2-1}{n^2}\Big[\Big(\lambda-\frac{1}{n+1}\Big)^2\Big(1+\frac{2}{n-1}\Big) + \|\boldsymbol{z}\|_2^2\Big]\,.$$

Since we have $\|\boldsymbol{z}\|_2^2 \leqslant 1-\lambda^2$, we get

$$\Big(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u}\Big)^\top M^{-1}\Big(\boldsymbol{y}-\frac{1}{n+1}\boldsymbol{u}\Big) \leqslant \frac{n^2-1}{n^2}\Big[\Big(\lambda-\frac{1}{n+1}\Big)^2\Big(1+\frac{2}{n-1}\Big) + 1-\lambda^2\Big]\,.$$

The right-hand term is a convex function in λ. On $[0,1]$, it is thus upper bounded by the values its takes in $\lambda=0$ and $\lambda=1$. In $\lambda=0$, this value is 1; in $\lambda=1$, this value is 1 as well.

We prove now that $\operatorname{vol}(E(\boldsymbol{s}', Q')) \leqslant e^{-\frac{1}{2(n+1)}} \operatorname{vol}(E(\boldsymbol{s}, Q))$. We start by noticing the following consequence of the "matrix-determinant lemma":

$$\det\Big(Q - \frac{2}{n+1}\boldsymbol{\omega}\boldsymbol{\omega}^\top\Big) = \Big(1-\frac{2}{n+1}\Big)\det(Q)\,.$$

We have thus

$$\frac{\operatorname{vol}(E(\boldsymbol{s}', Q'))}{\operatorname{vol}(E(\boldsymbol{s}, Q))} = \sqrt{\frac{\det(Q')}{\det(Q)}} = \Big(\frac{n^2}{n^2-1}\Big)^{n/2}\sqrt{\frac{n-1}{n+1}} = \Big(1-\frac{1}{n+1}\Big)\Big(1+\frac{1}{n^2-1}\Big)^{(n-1)/2}\,.$$

Using $1+t < e^t$ for $t \neq 0$, we get

$$\frac{\operatorname{vol}(E(\boldsymbol{s}', Q'))}{\operatorname{vol}(E(\boldsymbol{s}, Q))} < e^{-\frac{1}{n+1}+\frac{n-1}{2(n^2-1)}} = e^{-\frac{1}{2(n+1)}}\,,$$

as desired. □

This lemma involves a square root and this makes clear that an actual implementation of the algorithm requires some approximations. It is actually possible to achieve such an approximation in the computation while returning an exact answer when the input uses only rational numbers. Because of the inequality $\operatorname{vol}(E_{k+1}) \leqslant e^{-\frac{1}{2(n+1)}} \operatorname{vol}(E_k)$, the maximal number of iterations is $O\big((\langle R\rangle + \langle\varepsilon\rangle)n^2\big)$.

2.2 How to deal with any instance

To deal with any instance, we need to explain how we can slightly change the matrix A and the vector $\boldsymbol{b}$ so that we know in advance that the polytope $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b}\}$ is included in $\bar{B}(\mathbf{0}, R)$ and that its volume exceeds ε, where R and ε are now rational numbers that can be computed in polynomial time.

Lemma 2.2. *Set*

$$M := 2^{\varphi}\,, \quad \eta := 2^{-2\varphi}, \quad \textit{and} \quad \varepsilon := 2^{-3n\varphi}\,,$$

where $\varphi = n\lceil \log_2(n) \rceil + n\langle A \rangle + \langle \boldsymbol{b} \rangle$. Then the following two properties hold:

(i) $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b}\} \neq \varnothing \iff \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b}\} \cap [-M, M]^n \neq \varnothing.$

(ii) $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b}\} \neq \varnothing \iff \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b} - \eta \mathbf{1}\} \neq \varnothing$ *and* $\mathrm{vol}(\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b} - \eta\mathbf{1}\}) \geqslant \varepsilon.$

Proof. Without loss of generality, we assume that A and $\boldsymbol{b}$ have integer entries.

We prove first item (i). We prove the implication $\Rightarrow$, the other one being obvious. Assume there exists $\bar{\boldsymbol{x}}$ such that $A\bar{\boldsymbol{x}} \geqslant \boldsymbol{b}$. Using existence of feasible basic solutions for linear programs in the standard form and their expression via the Cramer formula (which involves here only integer numbers), we know that there exists a feasible solution $\tilde{\boldsymbol{x}}$ whose components are all upper bounded by $n!\|A\|_\infty^{n-1}\|\boldsymbol{b}\|_\infty$ in absolute value, where $\|A\|_\infty$ is the maximal entry of A in absolute value. The conclusion comes from the inequality $\log_2(n!\|A\|_\infty^{n-1}\|\boldsymbol{b}\|_\infty) \leqslant \varphi$.

We prove then item (ii), implication $\Rightarrow$. Assume there exists $\bar{\boldsymbol{x}}$ such that $A\bar{\boldsymbol{x}} \geqslant \boldsymbol{b}$. We have $A\bar{\boldsymbol{x}} \geqslant \boldsymbol{b} - \eta\mathbf{1}$ by the positivity of η. It remains to show the statement about the volume. Consider an arbitrary vector $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_\infty \leqslant \frac{1}{2}\varepsilon^{1/n}$. Notice that each entry of $A\boldsymbol{\delta}$ is lower bounded by $-\frac{1}{2}n\|A\|_\infty\varepsilon^{1/n}$, which is nonsmaller than η, by definition of η and ε. Therefore, we have $A\boldsymbol{\delta} \geqslant -\eta\mathbf{1}$, and $A(\bar{\boldsymbol{x}} + \delta) \geqslant \boldsymbol{b} - \eta\mathbf{1}$. This last inequality being valid for all $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_\infty \leqslant \frac{1}{2}\varepsilon^{1/n}$, considering a cube centered at $\bar{x}$, we get that the volume of $\{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geqslant \boldsymbol{b} - \eta\mathbf{1}\}$ is at least $(\varepsilon^{1/n})^n = \varepsilon$, as desired.

We finish by proving item (ii), implication $\Leftarrow$. Assume there exists $\bar{\boldsymbol{x}}$ such that $A\bar{\boldsymbol{x}} \geqslant \boldsymbol{b} - \eta\mathbf{1}$. Suppose by contradiction that there is no $\boldsymbol{x}$ such that $A\boldsymbol{x} \geqslant \boldsymbol{b}$. By Farkas's lemma, there exists $\boldsymbol{y} \geqslant \mathbf{0}$ such that $A^\top\boldsymbol{y} = 0$ and $\boldsymbol{b} \cdot \boldsymbol{y} > 0$. Up to a normalization, we can actually assume that $\boldsymbol{b} \cdot \boldsymbol{y} = 1$. Using again existence of feasible basic solutions for linear programs in the standard form and their expression via the Cramer formula (which involves here only integer numbers), we can further assume that $\|\boldsymbol{y}\|_\infty \leqslant n!\|A\|_\infty^{n-1}\|\boldsymbol{b}\|_\infty$, i.e., $\|\boldsymbol{y}\|_\infty \leqslant 2^{\varphi}$. Then

$$(\boldsymbol{b} - \eta\mathbf{1}) \cdot \boldsymbol{y} = 1 - \eta\|\boldsymbol{y}\|_1 \geqslant 1 - n2^{-2\varphi}2^{\varphi} > 0\,,$$

where the last inequality results from $\varphi > \log_2(n)$. Applying Farkas's lemma again (in the other direction), we get that there cannot exist any $\boldsymbol{x}$ such that $A\boldsymbol{x} \geqslant \boldsymbol{b} - \eta\mathbf{1}$, which contradicts the definition of $\bar{\boldsymbol{x}}$. $\square$

From this, we get the following theorem.

Theorem 2.3. *Given a rational matrix A and a rational vector $\boldsymbol{b}$ with the same number of rows. Solving $A\boldsymbol{x} \geqslant \boldsymbol{b}$ can be done in polynomial time.*

Note that, in full generality, we have only proved that deciding whether the system is feasible can be done in polynomial time (the returned solution satisfies instead $A\boldsymbol{x} \geqslant \boldsymbol{b} - \eta\mathbf{1}$) but with a small η an extra "rounding" procedure we do not discuss here leads to the exact solution.

2.3 Polynomial-time separation implies polynomial-time optimization

Consider now a system $A\boldsymbol{x} \geqslant \boldsymbol{b}$ for which we do not have the full matrix A and vector $\boldsymbol{b}$ in input. Instead, we suppose available an *separation oracle*, i.e., an algorithm that given a point $\boldsymbol{x}^0$ either returns 'yes' if $A\boldsymbol{x}^0 \geqslant \boldsymbol{b}$, or returns 'no' and some row index i of A with $\boldsymbol{a}^i \cdot \boldsymbol{x}^0 < b_i$ if $A\boldsymbol{x}^0 \not\geqslant \boldsymbol{b}$. Combining the ellipsoid algorithm with the ideas of Section 2.2 leads to the following theorem, which we state without a proof; see the book by Grötschel, Lovász, and Schrijver [**?**].

Theorem 2.4. *Consider a linear program*

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \in P\,, \end{aligned}$$

where $P \subseteq \mathbb{R}^n$ is a polyhedron whose (unknown) linear constraints have rational coefficients of encoding size at most φ. Assume that we can access P only through a separation oracle. Then the linear program can be solved using only a polynomial number of calls to the separation oracle.

A typical application of this theorem shows that some linear program with an exponential number of constraints can actually be solved in polynomial time.

CHAPTER 3

Interior-point methods

Kamarkar proposed in 1984 another polynomial algorithm for solving linear programs, and this one does have efficient implementations: the interior-point algorithm. We should actually say "the interior-point algorithms," with the plural, since there exist many variations. Most professional solvers rely nowadays on the simplex algorithm and on some interior-point algorithm, since there are the two efficient methods in practice, and show complementary behaviors in that respect.

3.1 The algorithm

We present one interior-point algorithm that is quite easy to describe, and which does run in polynomial time. In order to solve

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c} \cdot \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqslant \boldsymbol{0}\,, \end{aligned} \tag{P}$$

the idea is to consider instead the family of convex problems parametrized by the nonnegative real number μ:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c} \cdot \boldsymbol{x} - \mu \sum_{j=1}^{n} \ln(x_j) \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} > \boldsymbol{0}\,, \end{aligned} \tag{P(μ)}$$

where $\boldsymbol{x} > \boldsymbol{0}$ means that all components of $\boldsymbol{x}$ are positive. We assume throughout the section that A has full-row rank.

Let $\mu > 0$ and assume (P(μ)) feasible. Then it admits a unique optimal solution $\boldsymbol{x}^\star(\mu)$. The existence of such an optimal solution results from the change of variable $x_j = e^{y_j}$ and by arguing of the coercivity of the objective function; its uniqueness results from the strict convexity of the objective function. Moreover, by the theorem of Kuhn et Tucker for convex programming, $\boldsymbol{x}^\star(\mu)$ is the unique solution $\boldsymbol{x}$ of the system with $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ as unknown:

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{b} \\ A^\top \boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c} \\ (s_1x_1, s_2x_2, \ldots, s_nx_n) &= \mu\boldsymbol{1} \\ \boldsymbol{x}, \boldsymbol{s} &\geqslant \boldsymbol{0}\,. \end{aligned} \tag{3.1}$$

The algorithm will start with a large μ, and make it converge to 0. At each iteration, the system above is solved approximately.

A solution $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ is updated as $\boldsymbol{x}+\Delta\boldsymbol{x}$, $\boldsymbol{y}+\Delta\boldsymbol{y}$, $\boldsymbol{s}+\Delta\boldsymbol{s}$ when μ is changed. By substituting in equation (3.1) and neglecting the terms $\Delta s_j \Delta x_j$, this suggests to solve the following linear system (where $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ are regarded as constant and $\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s}$ are the unknown):

$$\begin{aligned} &A\Delta\boldsymbol{x} = \mathbf{0} \\ &A^\top \Delta\boldsymbol{y} + \Delta\boldsymbol{s} = \mathbf{0} \\ &(s_1\Delta x_1 + x_1\Delta s_1, s_2\Delta x_2 + x_2\Delta s_2, \dots, s_n\Delta x_n + x_n\Delta s_n) = \mu\mathbf{1} - (s_1x_1, s_2x_2, \dots, s_nx_n)\,. \end{aligned} \tag{3.2}$$

Since A has full-row rank, this system always admits a unique solution in $\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s}$.

The algorithm starts with $\mu = 1$, and applies under the assumption that $\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{s}^0$ satisfy

$$\begin{aligned} &A\boldsymbol{x}^0 = \boldsymbol{b} \\ &A^\top \boldsymbol{y}^0 + \boldsymbol{s}^0 = \boldsymbol{c} \\ &(s_1^0x_1^0, s_2^0x_2^0, \dots, s_n^0x_n^0) = \mathbf{1} \\ &\boldsymbol{x}^0, \boldsymbol{s}^0 > \mathbf{0}\,. \end{aligned} \tag{3.3}$$

To describe the algorithm, we need the "distance to centrality"

$$\mathrm{cdist}_\mu(\boldsymbol{x}, \boldsymbol{s}) \coloneqq \left\| \big(\rho(s_1x_1, \mu), \rho(s_2x_2, \mu), \dots, \rho(s_nx_n, \mu)\big) \right\|_2\,,$$

where $\rho(a, \mu) = \sqrt{a/\mu} - \sqrt{\mu/a}$. It finds its motivation in the following lemma, which implies the following: if $\mathrm{cdist}_\mu(\boldsymbol{x}, \boldsymbol{s})$ remains bounded, then a small μ means a small $\boldsymbol{s}\cdot\boldsymbol{x}$.

Lemma 3.1. *The inequality $a \leqslant \big(2 + \rho^2(a, \mu)\big)\mu$ holds for every positive a and μ.*

Proof. Let $\alpha \coloneqq \rho^2(a, \mu)$. We have then $a^2 - (2 + \alpha)\mu a + \mu^2 = 0$. Solving the corresponding degree-two polynomial leads to $0 \leqslant a \leqslant (2 + \alpha)\mu$. □

Here is the algorithm.

Input : Points $\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{s}^0$ satisfying the system (3.1), a positive number ε
Output : A solution $\tilde{\boldsymbol{x}}$ within ε of optimality

```
x ← x⁰;
y ← y⁰;
s ← s⁰;
μ ← 1;
while μ > ε/(2(n+1)) do
    μ ← (1 − 1/(2√n)) μ;
    Compute the unique solution of the linear system (3.2);
    x ← x + Δx;
    y ← y + Δy;
    s ← s + Δs;
end
x̃ ← x;
return x̃;
```

Algorithme 2 : Interior-point algorithm

The correctness of the algorithm relies on the following two lemmas. The first one ensures that we keep a primal-dual pair $(\boldsymbol{x}, \boldsymbol{y})$ of feasible solutions.

Lemma 3.2. *Suppose that* $\boldsymbol{x} > \boldsymbol{0}$, $\boldsymbol{s} > \boldsymbol{0}$, *and* $\operatorname{cdist}_\mu(\boldsymbol{x}, \boldsymbol{s}) < 2$. *Then,* $\boldsymbol{x} + \Delta\boldsymbol{x} > \boldsymbol{0}$ *and* $\boldsymbol{s} + \Delta\boldsymbol{s} > \boldsymbol{0}$, *where* $(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s})$ *is the unique solution of the linear system* (3.2).

Proof. Omitted. □

The second one we stay close to a solution of (3.1).

Lemma 3.3. *The invariant* $\operatorname{cdist}_\mu(\boldsymbol{x}, \boldsymbol{s}) \leqslant \sqrt{2}$ *is kept along the algorithm.*

Proof. At the first iteration, we have $\operatorname{cdist}_\mu(\boldsymbol{x}, \boldsymbol{s}) = 0$. The rest of the proof is omitted. □

Denote by $\tilde{\boldsymbol{s}}$ the last $\boldsymbol{s}$ when the algorithm stops. According to Lemmas 3.1 and 3.3, we have

$$\tilde{\boldsymbol{s}} \cdot \tilde{\boldsymbol{x}} \leqslant \big(2 + \operatorname{cdist}_\mu(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{s}})\big) n\mu \,.$$

The quality of the solution $\tilde{\boldsymbol{x}}$ returned by the algorithm is guaranteed by the following property. To be *within* ε *of optimality* means that not to exceed the optimal value by more than ε.

Proposition 3.4. *Let* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ *be a solution of*

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{b} \\ A^\top \boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c} \\ \boldsymbol{x}, \boldsymbol{s} &\geqslant \boldsymbol{0} \,, \end{aligned}$$

and let $\varepsilon > 0$. *If* $\boldsymbol{s} \cdot \boldsymbol{x} \leqslant \varepsilon$, *then* $\boldsymbol{x}$ *is a feasible solution of* (P) *within* ε *of optimality.*

Proof. Note that $\boldsymbol{x}$ is a feasible solution of (P). By multiplying the second block by $\boldsymbol{x}$, we get

$$\boldsymbol{x}^\top A^\top \boldsymbol{y} + \boldsymbol{s} \cdot \boldsymbol{x} = \boldsymbol{c} \cdot \boldsymbol{x} \,.$$

Using the first block, we get $\boldsymbol{b} \cdot \boldsymbol{y} + \boldsymbol{s} \cdot \boldsymbol{x} = \boldsymbol{c} \cdot \boldsymbol{x}$. We conclude by noticing that $\boldsymbol{y}$ is a feasible solution of the dual of (P) and that weak duality implies then that $\boldsymbol{b} \cdot \boldsymbol{y}$ is a lower bound on the optimal value of (P). □

All together, the algorithm works correctly: it returns a feasible solution thanks to Lemma 3.2; this solution is within ε of optimality thanks to Proposition 3.4. It requires $O(\sqrt{n} \log \frac{n}{\varepsilon})$ iterations. So we get:

Theorem 3.5. *Provided we have an initial solution* $\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{s}^0$ *of* (3.1) *with* $\mu = 1$, *Algorithm 2 computes in polynomial time a feasible solution of* (P) *that is within* ε *of optimality.*

In order to solve any linear program, two additional ingredients are needed: a way to transform an approximate optimal solution into an exact optimal solution; an initial solution $\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{s}^0$ satisfying the system (3.1). The first ingredient relies on “strictly complementary pairs” of optimal solutions, whose existence results from the Goldman–Tucker theorem; see Section 3.2 The second one is not too difficult—just a bit long—and relies on a “self-dual linear program”; see Section 3.3.

3.2 Rounding

3.3 Initialization

Bibliography

[1] David Applegate, William Cook, Keld Helsgaun, and the Star Tours Team, *1,331,906,450 stars: Gaia dr2*, Web page, 2026, Accessed on February 16, 2026. 2