Computational Geometry Column 66

Khaled Elbassioni[*] Adrian Dumitrescu[†]

October 29, 2017

Abstract

This column is dedicated to the memory of Leonid Khachiyan on the 65th anniversary of his birthday. In writing this column we remember some problems and results due to him that are most closely related to geometry. The column may also suggest that many optimization problems have a geometric face which is sometimes hidden.

Keywords: linear programming, polytope, rounding, vertex enumeration, hypergraph transversal, extremal determinant, affine degeneracy, isoperimetric inequality, random generation, counting, poset, linear extension.

1 Introduction

Besides being famous for his polynomial-time algorithm for linear programming, Leonid Khachiyan (1952–2005) will be remembered for the gentle nature of his character and for his inspiring example in his interactions with students and collaborators. This column pays homage to him by highlighting some of the problems he had worked on, mostly from the area of geometry. Even problems that did not seem to be apriori of a geometric nature have often received quality solutions when viewed from a geometric perspective. In recognizing the diversity of the areas in which Khachiyan made substantial contributions in collaboration with many other researchers, tools from linear algebra and the geometry of polytopes play an important role. For a short but comprehensive survey of Khachiyan's scientific contributions the reader is referred to [16].

2 Vertex enumeration and related problems

The well-known Minkowski-Weyl theorem states that any convex polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ can be represented as the Minkowski sum of the convex hull of the set $\mathcal{V}(\mathcal{P})$ of its extreme points and the conic hull of the set $\mathcal{D}(\mathcal{P})$ of its extreme directions; see for instance [91]. Further, for pointed polyhedra, i.e., those that do not contain lines, this representation is unique. Given a polyhedron $\mathcal{P}$ by its linear description (called $\mathcal{H}$-representation) as the intersection of finitely many halfspaces, obtaining the set $\mathcal{V}(\mathcal{P}) \cup \mathcal{D}(\mathcal{P})$, required by the other representation (called $\mathcal{V}$-representation), is a well-known problem, studied in the literature in different (but polynomially equivalent) forms, e.g., the *vertex enumeration* (VE) problem [18], the *convex hull* problem [4] or the *polytope-polyhedron*

[*]Department of Computer Science, Masdar Institute, Khalifa University of Science and Technology, Abu Dhabi, UAE. Email: `kelbassioni@masdar.ac.ae`

[†]Department of Computer Science, University of Wisconsin–Milwaukee, USA. Email: `dumitres@uwm.edu`

problem [78]. In the following, we write $\mathcal{P}_{\mathcal{H}}$ and $\mathcal{P}_{\mathcal{V}}$ to distinguish whether $\mathcal{P}$ is given by its $\mathcal{H}$-representation or $\mathcal{V}$-representation, respectively. Clearly, the size of the extreme set $\mathcal{V}(\mathcal{P}) \cup \mathcal{D}(\mathcal{P})$ can be (and typically is) exponential in the dimension n and the number of linear inequalities m. Thus when considering the computational complexity of the vertex enumeration problem, one is usually interested in *output-sensitive* algorithms, i.e., whose running time depends not only on n and m, but also on $|\mathcal{V}(\mathcal{P}) \cup \mathcal{D}(\mathcal{P})|$. Alternatively, we may consider the following, polynomially equivalent, decision variant of the problem; in this description, $\mathcal{C}(\mathcal{P})$ could be either $\mathcal{V}(\mathcal{P})$, $\mathcal{D}(\mathcal{P})$, or $\mathcal{V}(P) \cup \mathcal{D}(P)$.

Dec($\mathcal{P}_{\mathcal{H}}, \mathcal{X} \subseteq \mathcal{C}(\mathcal{P})$): Given a polyhedron $\mathcal{P}$, represented by a system of linear inequalities, and a subset $\mathcal{X} \subseteq \mathcal{C}(\mathcal{P})$, is $\mathcal{X} = \mathcal{C}(\mathcal{P})$?

It is well-known and also easy to see that the decision problems for $\mathcal{D}(\mathcal{P})$ or for $\mathcal{V}(\mathcal{P}) \cup \mathcal{D}(\mathcal{P})$ are equivalent to that for $\mathcal{V}(\mathcal{P}')$ where $\mathcal{P}'$ is some polytope whose description size is polynomial in that of $\mathcal{P}$. It is also well-known that if the decision problem is NP-hard, then no polynomial-time algorithm can generate the elements of the set $\mathcal{C}(\mathcal{P})$ unless P=NP; see for instance [13].

Vertex enumeration is an outstanding open problem in computational geometry and polyhedral combinatorics (see, e.g., [31, 78, 89]), and has numerous applications. For example, understanding the structure of the vertices helps in designing approximation algorithms for combinatorial optimization problems [107]; finding all vertices can be used for computing Nash equilibria for bimatrix games [7]. Numerous algorithmic ideas have been introduced in the literature either for vertex or for facet enumeration, see, e.g., [1, 4, 5, 6, 18, 22, 25, 28, 31, 90, 92].

According to the main result in [66], the problem Dec($\mathcal{P}_{\mathcal{H}}, \mathcal{X} \subseteq \mathcal{V}(\mathcal{P})$) is NP-hard in the case of unbounded polyhedra, more precisely, when $|\mathcal{D}(\mathcal{P})|$ is exponentially large in the input size. This negative result holds, even when restricted to 0/1-*polyhedra* [15], that is, when $\mathcal{V}(\mathcal{P}) \subseteq \{0,1\}^n$. This is also in contrast with the fact that for 0/1-polytopes, the VE problem is known to be solvable with polynomial delay; that is, the vertices are generated such that the delay between any successive outputs is polynomial only in the input description of $\mathcal{P}$. The complexity of the VE problem for general polytopes (i.e., when $\mathcal{D}(\mathcal{P}) = \emptyset$) remains a challenging open problem.

Problem 1. *Is problem Dec($\mathcal{P}_{\mathcal{H}}, \mathcal{X} \subseteq \mathcal{V}(\mathcal{P})$) solvable in polynomial time for any polytope $\mathcal{P}$?*

Let $A \in \{0,1\}^{m\times n}$ be a 0/1-matrix such that the polyhedron $\mathcal{P} = \mathcal{P}(A, \mathbf{1}) = \{x \in \mathbb{R}^n \mid Ax \geq \mathbf{1},\ x \geq \mathbf{0}\}$ has only integral vertices, where $\mathbf{0} \in \mathbb{R}^m$ and $\mathbf{1} \in \mathbb{R}^m$ are the m-dimensional vectors of all zeros and all ones, respectively. Then the vertices of $\mathcal{P}$ are in one-to-one correspondence with the minimal transversals of the hypergraph $\mathcal{H}(A) \subseteq 2^{[n]}$, whose characteristic vectors of hyperedges are the rows of A. For instance, if the matrix A is *totally unimodular* (i.e., every square subdeterminant has value in $\{-1, 0, 1\}$), then the polyhedron $\mathcal{P}(A, \mathbf{1})$ has integral vertices, and VE is equivalent to finding all minimal transversals (i.e., hitting sets) of a *unimodular hypergraph*. Consequently, it follows from a well-known result of Fredman and Khachiyan [40] that all vertices of such polyhedra can be enumerated in *quasi-polynomial* time, and hence the VE problem in this case is unlikely to be NP-hard. A very recent result [37] shows that, for totally unimodular matrices A, the VE problem for the polyhedron $\mathcal{P}(A, \mathbf{1})$ can be solved in polynomial time.

A 0/1-matrix is said to be *balanced* if it does not contain any square submatrix of odd order having row and column sum equal to 2. 0/1-totally unimodular matrices are a proper subset of balanced matrices. An interesting open question is whether the above mentioned result for totally unimodular matrices can be extended to the case when the right hand side of the polyhedron is an arbitrary non-negative vector b, or when the matrix A is a balanced matrix.

Problem 2. *Is problem* $Dec(\mathcal{P}, \mathcal{X} \subseteq \mathcal{V}(\mathcal{P}))$ *solvable in polynomial time, when* $\mathcal{P} = \mathcal{P}(A, b) = \{x \in \mathbb{R}^n \mid Ax \geq b,\ x \geq \mathbf{0}\}$ *for a balanced matrix* A *and a non-negative vector* b*?*

When $b = \mathbf{1}$, a solution to Problem 2 would follow from a polynomial-time algorithm for the hypergraph transversal problem: Given a hypergraph $\mathcal{H}$ by the list of its hyperedges, enumerate all minimal transversals of $\mathcal{H}$. The currently fastest algorithm for this problem runs in quasi-polynomial time [40].

Problem 3. *Does there exist a polynomial-time algorithm for the hypergraph transversal problem?*

An interesting special case of Problem 3 is when $\mathcal{H}$ is a *geometric* hypergraph: given a set of points $P \in \mathbb{R}^d$ and set of ranges $\mathcal{R} \subseteq \mathbb{R}^d$, let $\mathcal{H}$ be a hypergraph whose every hyperedge is defined by intersecting a range in $\mathcal{R}$ with the set of points P. Then all minimal transversals (resp., all minimal set covers) of $\mathcal{H}$, can be enumerated in polynomial time if $\mathcal{R}$ is a set of half-planes, balls, or polytopes with a fixed number of facets, in fixed dimension $d = O(1)$ [38]. An extension of this result to hypergraphs of bounded VC-dimension is a natural open problem.

Problem 4. *Given a hypergraph* $\mathcal{H}$ *of bounded VC-dimension* $d = O(1)$*, can one enumerate all minimal transversals (resp., all minimal set covers) of* $\mathcal{H}$ *in polynomial time?*

Clearly, a positive answer to Problem 1 would give a positive answer to Problem 2, while a positive answer to Problem 3 would also yield a positive answer to Problem 4.

Let $\mathcal{A} \subseteq \mathbb{R}^n$ be a given subset of points in $\mathbb{R}^n$. Fix a point $z \in \mathbb{R}^n$, say $z = \mathbf{0}$, and consider the following four geometric objects [12]:

- A *simplex*: a minimal subset $X \subseteq \mathcal{A}$ of points containing z in its convex hull: $z \in \text{conv hull}(X)$.
- An *anti-simplex*: a maximal subset $X \subseteq \mathcal{A}$ of points not containing z in its convex hull: $z \notin \text{conv hull}(X)$.
- A *body*: a minimal (full-dimensional) subset $X \subseteq \mathcal{A}$ of points containing z in the interior of its convex hull: $z \in \text{int}(\text{conv hull}(X))$.
- An *anti-body*: a maximal subset $X \subseteq \mathcal{A}$ of points not containing z in the interior of its convex hull: $z \notin \text{int}(\text{conv hull}(X))$.

Equivalently, a simplex (resp., body) is a minimal collection of points not contained in an *open* (resp., *closed*) half-space through the origin. An anti-simplex (resp., anti-body) is a maximal collection of points contained in an open (resp., closed) half-space through the origin. It is known that $|X| \leq n + 1$ for any simplex $X \subseteq \mathcal{A}$ and that $n + 1 \leq |X| \leq 2n$ for any body $X \subseteq \mathcal{A}$.

Let $A \in \mathbb{R}^{m \times n}$, where $m = |\mathcal{A}|$, be the matrix whose rows are the points of $\mathcal{A}$. It follows from the above definitions (via Farkas' Lemma) that simplices and anti-simplices are in one-to-one correspondence respectively with the *minimal infeasible* and *maximal feasible* subsystems of the linear system of inequalities:

$$Ax \geq \mathbf{1}, \quad x \in \mathbb{R}^n.$$

Similarly, it follows that bodies and anti-bodies correspond respectively to the minimal infeasible and maximal feasible subsystems of the system:

$$Ax \geq \mathbf{0}, \quad x \neq \mathbf{0}. \tag{1}$$

Given a set of points $\mathcal{A} \subseteq \mathbb{R}^n$, and a partial list $\mathcal{X}$ of anti-bodies of $\mathcal{A}$, it was shown in [12] that it is NP-hard to determine if $\mathcal{X}$ is the complete list of anti-bodies of $\mathcal{A}$; in other words, the enumeration problem for anti-bodies is NP-hard. Equivalently, given an infeasible system (1), and a partial list of maximal feasible subsystems of (1), it is NP-hard to determine if the given partial list is complete. In contrast, the problem of enumerating the set of bodies for a given set of points $\mathcal{A} \subseteq \mathbb{R}^n$ is at least as hard as the hypergraph transversal problem [12].

The situation for simplices and anti-simplices is much less understood: the problem of generating simplices for a given set of points $\mathcal{A} \subseteq \mathbb{R}^n$ is essentially equivalent with Problem 1. In fact, if the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq \mathbf{0}\}$ is bounded, i.e., if it is a polytope, then the vertices of $\mathcal{P}$ are in one-to-one correspondence with the simplices of the point set $\mathcal{A}$ whose elements are the columns of the augmented matrix $[A \mid -b]$. For anti-simplices the complexity is completely open.

Problem 5. *Given a set of points $\mathcal{A} \subseteq \mathbb{R}^n$, is it NP-hard to enumerate the anti-simplices of $\mathcal{A}$?*

3 Rounding of polytopes and related problems

A well-known result of John [51] states that an arbitrary polytope $\mathcal{Q} \subset \mathbb{R}^n$ can be n-rounded, i.e., there exists an ellipsoid $\mathcal{E} \subset \mathbb{R}^n$, such that

$$\frac{1}{n}\mathcal{E} \subseteq \mathcal{Q} \subseteq \mathcal{E}. \tag{2}$$

In particular, (2) holds for the minimum-volume ellipsoid $\mathcal{E} = \text{MVEE}(\mathcal{Q})$ circumscribed about $\mathcal{Q}$, the so-called the *Löwner* ellipsoid. Furthermore, if $\mathcal{Q}$ is *centrally symmetric* (i.e., $\mathcal{Q} = -\mathcal{Q}$), the factor $\frac{1}{n}$ in (2) can be improved to $\frac{1}{\sqrt{n}}$. Given $\epsilon > 0$, an ellipsoid $E \subset \mathbb{R}^n$ is said to be a $(1+\epsilon)n$-rounding of $\mathcal{Q}$ if the factor $\frac{1}{n}$ in (2) is replaced by $\frac{1}{n(1+\epsilon)}$. Similarly, for any $\epsilon > 0$, an ellipsoid $\mathcal{E}$ is said to be a $(1+\epsilon)$-approximation of $\text{MVEE}(\mathcal{Q})$ if

$$\mathcal{Q} \subseteq \mathcal{E}, \qquad \text{Vol}(\mathcal{E}) \leq (1+\epsilon)\text{Vol}(\text{MVEE}(\mathcal{Q})). \tag{3}$$

The problem of approximating the minimum volume enclosing ellipsoid arises in several applications such as integer programming [45, 77], polytope volume computation [30, 80], statistics and optimal design [99, 105], computational geometry [24, 110], and computer graphics [17, 32]. Another important related problem is the computation of the maximum-volume ellipsoid $\text{MVIE}(\mathcal{Q})$ inscribed in a polytope $\mathcal{Q}$. For any $\epsilon > 0$, an ellipsoid $\mathcal{E}$ is said to be a $(1-\epsilon)$-approximation of $\text{MVIE}(\mathcal{Q})$ if

$$\mathcal{E} \subseteq Q, \qquad \text{Vol}(\mathcal{E}) \geq (1-\epsilon)\text{Vol}(\text{MVIE}(\mathcal{Q})). \tag{4}$$

The problem of approximating the maximum volume inscribed ellipsoid appears as a basic step in a number of convex optimization algorithms, see, e.g., [103].

Given a point $a \in \text{int}(\mathcal{Q})$, a related problem is to approximate the minimum-volume (resp., maximum-volume) ellipsoid $\text{MVEE}(\mathcal{Q}, a)$ (resp., $\text{MVIE}(\mathcal{Q}, a)$) enclosing (resp., inscribed in) $\mathcal{Q}$, and centered at a.

Given a (full-dimensional) polytope $\mathcal{Q}$, let us call for brevity the problems of computing ellipsoids $\mathcal{E}$ satisfying (3) and (4) as $\overline{\text{MVEE}}(\mathcal{Q}, \epsilon)$ and $\overline{\text{MVIE}}(\mathcal{Q}, \epsilon)$, respectively. Given also a point $a \in \text{int}(\mathcal{Q})$, let us call the corresponding problems of approximating $\text{MVEE}(\mathcal{Q}, a)$ and $\text{MVIE}(\mathcal{Q}, a)$ as $\overline{\text{MVEE}}(\mathcal{Q}, a, \epsilon)$ and $\overline{\text{MVIE}}(\mathcal{Q}, a, \epsilon)$, respectively.

The current best bounds on the complexity of these problems depend on how the polytope $\mathcal{Q}$ is described. If $\mathcal{Q}$ is given by a list of its vertices $\mathcal{Q} = \operatorname{conv\,hull}(\mathcal{A})$ (resp., by a list of its facets $\mathcal{Q} = \{x \in \mathbb{R}^n : a_i^T x \leq 1, \text{for } i = 1, \ldots, m\}$) for some explicitly given set of m points $\mathcal{A} := \{a_1, \ldots, a_m\} \subseteq \mathbb{R}^n$, the problems $\overline{\text{MVEE}}(\mathcal{Q}, \epsilon)$ and $\overline{\text{MVEE}}(\mathcal{Q}, a, \epsilon)$ (resp., $\overline{\text{MVIE}}(\mathcal{Q}, \epsilon)$ and $\overline{\text{MVIE}}(\mathcal{Q}, a, \epsilon)$) can be formulated as convex optimization problems over the cone of positive semidefinite matrices. We write $\mathcal{Q}_{\mathcal{H}}$ and $\mathcal{Q}_{\mathcal{V}}$ to distinguish whether $\mathcal{Q}$ is given by its $\mathcal{H}$-representation (i.e., linear inequalities) or $\mathcal{V}$-representation (i.e., vertices), respectively. Khachiyan and Todd [67] considered the above four problems. They gave linear time reductions between the problems resulting in the following equivalences/implications:

$$\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, \epsilon) \longleftrightarrow \overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, a, \epsilon) \longleftrightarrow \overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, a, \epsilon) \longrightarrow \overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon). \tag{5}$$

(For example, the reduction $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, \epsilon) \longrightarrow \overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, a, \epsilon)$ is established by showing that solving problem $\overline{\text{MVEE}}(\mathcal{Q}, \epsilon)$ for $\mathcal{Q} = \operatorname{conv\,hull}(\{a_1, \ldots, a_m\}) \subseteq \mathbb{R}^n$ can be done by solving $\overline{\text{MVEE}}(\mathcal{Q}', \mathbf{0}, \epsilon)$ for $\mathcal{Q}' := \operatorname{conv\,hull}(\{\pm \begin{pmatrix} a_1 \\ 1 \end{pmatrix}, \ldots, \pm \begin{pmatrix} a_m \\ 1 \end{pmatrix}\}) \subseteq \mathbb{R}^{n+1}$, then taking the intersection of the resulting ellipsoid with the hyperplane $\Pi = \{x \in \mathbb{R}^{n+1} : x_{n+1} = 1\}$.) A reduction in the opposite direction $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon) \longrightarrow \overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, a, \epsilon)$ in (5) is not known.

Problem 6. *Is there a "low-order" polynomial (say, linear) reduction from problem* $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ *to problem* $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, a, \epsilon)$*?*

Based on interior-point methods, Khachiyan and Todd [67] showed that the (most general) problem $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ can be solved, in

$$O\left(m^{3.5} \ln\left(\frac{mR}{\epsilon}\right) \ln\left(\frac{n \ln R}{\epsilon}\right)\right) \tag{6}$$

arithmetic operations and comparisons, where m is the number of inequalities in the description of $\mathcal{Q}$ and R is an apriori upper bound on the ratio between two Euclidean balls, the first of which is circumscribed about $\mathcal{Q}$ and the second one is inscribed in $\mathcal{Q}$; such a ratio can be derived from the input bit length of the description assuming rational input. A slight improvement on the bound (6), removing the factor $\ln\left(\frac{n \ln R}{\epsilon}\right)$, was obtained in [3].

In contrast, it was conjectured in [67] that problems $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ and $\overline{MVIE}(\mathcal{Q}_{\mathcal{V}}, \epsilon)$ are NP-hard. It is known [41] that these problems are NP-hard when balls are considered instead of ellipsoids.

Problem 7. *What is the complexity of solving the problems* $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ *and* $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{V}}, \epsilon)$*?*

In a later paper [65], Khachiyan gave a first order method requiring

$$O\left(mn^2\left(\frac{n}{\epsilon} + \log\log m\right)\right) \tag{7}$$

arithmetic operations and comparisons, for solving problem $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, \epsilon)$, where m is the number of vertices in the description of $\mathcal{Q}$. Using this result as a "pre-rounding" step for the Newton path-following method developed by Nesterov and Nemirovskiĭ [86, 87], the dependence on $\frac{1}{\epsilon}$ in (7) can be improved to $O(m^{3.5} \log \frac{m}{\epsilon})$. Note that this gives an algorithm with the same bound for $(1+\epsilon)n$-rounding of $\mathcal{Q}$; in fact, a better bound of $O\left(mn^2\left(\frac{1}{\epsilon} + \log n + \log\log m\right)\right)$, for the rounding, is obtained in [65].

Note also that the bound (7) does not depend on the *bit length* of the input description of $\mathcal{Q}$. In contrast, Khachiyan [65] also showed that obtaining a similar *strongly polynomial* bound for problem $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ is equivalent to solving the long-standing open problem of obtaining a strongly polynomial-time algorithm for linear programming (see Section 6).

Problem 8. *Can the bounds (6) and (7) on the running time complexity of solving problems $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ and $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, \epsilon)$, respectively, be improved in terms of m n, and/or ϵ?*

Geometric interpretations of Khachiyan's algorithm were provided in [72, 106]. Moreover, a slight improvement of the bound (7), removing the $\log \log m$ term was obtained in [72] using a simple initialization scheme. Additionally, given the description of $\mathcal{Q} = \text{conv hull}(\mathcal{A})$, the algorithm in [72] computes a "small" ϵ*-core set*, that is, a subset $\mathcal{X} \subseteq \mathcal{A}$ of size $O(\frac{d}{\epsilon^2})$ (independent of $|\mathcal{A}|$) such that for the ellipsoid $\mathcal{E} \supseteq \mathcal{A}$ computed by the algorithm,

$$\text{Vol}(\text{MVEE}(\mathcal{X})) \leq \text{Vol}(\text{MVEE}(\mathcal{A})) \leq \text{Vol}(\mathcal{E}) \leq (1+\epsilon)\text{Vol}(\text{MVEE}(\mathcal{X})) \leq (1+\epsilon)\text{Vol}(\text{MVEE}(\mathcal{A})).$$

This inequality implies that the minimum volume enclosing ellipsoid of $\mathcal{X}$ provides a good approximation to that of $\mathcal{A}$. The existence of small core sets has been established for several geometric optimization problems, see, e.g., [10, 9, 23, 71, 113], and plays an important role in developing efficient and practical algorithms for various large-scale geometric problems. A similar small core-set result for problem $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$ is not known. More precisely, given $Q = \{x \in \mathbb{R}^n : a_i^T x \leq 1, \text{ for } i = 1, \ldots, m\}$, define an ϵ-core set to be a subset $\mathcal{X} \subseteq \{a_1, \ldots, a_m\}$ such that if we write $\mathcal{Q}' = \{x \in \mathbb{R}^n : a_i^T x \leq 1, \text{ for } a_i \in \mathcal{X}\}$ then there is an ellipsoid $\mathcal{E} \subseteq \mathcal{Q}$ such that

$$\text{Vol}(\text{MVIE}(\mathcal{Q}')) \geq \text{Vol}(\text{MVIE}(\mathcal{Q})) \geq \text{Vol}(\mathcal{E}) \geq (1-\epsilon)\text{Vol}(\text{MVEE}(\mathcal{Q}')) \geq (1-\epsilon)\text{Vol}(\text{MVEE}(\mathcal{Q})).$$

Problem 9. *Provide an (as small as possible) upper bound $f(n, \epsilon)$ on the size of an ϵ-core set $\mathcal{X}$ for $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}}, \epsilon)$, such that $\mathcal{X}$ can be computed in polynomial time.*

Note that for fixed dimension $n = O(1)$, problems $\overline{\text{MVEE}}(\mathcal{Q}_{\mathcal{V}}, 0)$ and $\overline{\text{MVIE}}(\mathcal{Q}_{\mathcal{H}})$ can be solved in $n^{O(n^2)}m$ time the real RAM model of computation, where m is the number of vertices or facets of $\mathcal{Q}$ [2, 24, 84, 110].

4 General position and extremal determinants

The input to a geometric problem is said to be in *general position* if no nontrivial algebraic relation with integer coefficients holds among the parameters that specify the input [95]. Typical assumptions of this type are: no three points are collinear, no four points are cocircular, no two lines are parallel, no three lines are concurrent, no four points in $\mathbb{R}^3$ are coplanar, etc.

Let us first recall some standard terms, see, e.g., [33, App. A.4],[83, Ch. 1]. Let $P = \{p_1, \ldots, p_n\}$ be a finite set of points in $\mathbb{R}^d$. A point x is a *linear combination* of P if

$$x = \sum_{i=1}^{n} \lambda_i p_i,$$

for suitable real numbers λ_i. If $\sum_{i=1}^{n} \lambda_i = 1$, then x is also called an *affine combination* of P; if in addition, $0 \leq \lambda_i \leq 1$ for $i = 1, \ldots, n$, then x is a *convex combination* of P. P is said to be *linearly* *dependent* (or *linearly degenerate*) if there exists a point $p_i \in P$ which is a linear combination of $P \backslash \{p_i\}$; otherwise, P is *linearly independent*. Similarly, P is said to be *affinely dependent* (or *affinely degenerate*) if there exists a point $p_i \in P$ which is an affine combination of $P \setminus \{p_i\}$; otherwise, P is *affinely independent*. A point $p \in P$ is said to be *extreme* if $\mathrm{conv}(P) \neq \mathrm{conv}(P \setminus \{p\})$. A k-dimensional affine subspace is often called a *k-flat*. A set of points P in $\mathbb{R}^d$ is in general position if no unnecessary affine dependencies exist, i.e., no $k \leq d+1$ points lie in a common $(k-2)$-flat. (The precise meaning of general position may depend on the context and may include other conditions when convenient.)

For a real $d \times n$ matrix A of rank d (where $d \leq n$), let $\mathcal{B} = \mathcal{B}(A)$ be the set of all nondegenerate $d \times d$ submatrices of A, and let

$$\delta(A) = \min\{|\det B| \ : \ B \in \mathcal{B}\}, \quad \Delta(A) = \max\{|\det B| \ : \ B \in \mathcal{B}\}.$$

In particular, if $P = \{p_1, \ldots, p_n\}$ is a set of points in $\mathbb{R}^d$, then

$$\delta \begin{pmatrix} 1 & \cdots & 1 \\ p_1 & \cdots & p_n \end{pmatrix}$$

equals $d!$ times the minimum nonzero volume of a simplex determined by $d+1$ points from P; similarly,

$$\Delta \begin{pmatrix} 1 & \cdots & 1 \\ p_1 & \cdots & p_n \end{pmatrix}$$

equals $d!$ times the maximum nonzero volume of a simplex determined by $d+1$ points from P.

Khachiyan [64] proved that for any polynomial $p = p(d,n)$, the problem of approximating $\delta(A)$ within a factor of 2^p is NP-hard. Moreover, in the same paper he proved that determining whether a set of n rational points in d-space is affinely or linearly degenerate is NP-hard. The current fastest algorithm for testing the affine degeneracy of a set of n points in $\mathbb{R}^d$, due to Edelsbrunner and Guibas [34], runs in $O(n^d)$ time and $O(n)$ space in the real RAM model of computation. The same running time is attained by an earlier algorithm of Edelsbrunner, O'Rourke, and Seidel [35] that uses quadratic space.

Papadimitriou [88] proved that the problem of computing $\Delta(A)$ is NP-hard. Khachiyan [64] gave an algorithm for approximating $\Delta(A)$ within a factor of $[(1+\varepsilon)d]^{(d-1)/2}$ in time $O(nd^2(\varepsilon^{-1} + \log d + \log\log n))$ in the same model. As such, the maximum nonzero volume of a simplex determined by $d+1$ points from P can be approximated within a factor of $[(1+\varepsilon)(d+1)]^{d/2}$ in this time.

We list a few open problems:

Problem 10. *Can the time complexity of testing the affine degeneracy of a set of n points in $\mathbb{R}^d$ be improved?*

Problem 11. *Can a better approximation of the maximum nonzero volume of a simplex be found efficiently?*

5 Isoperimetric inequalities and counting linear extensions

We start by recalling a few terms in relation to partial orders. A partially ordered set (or poset) is a set P equipped with an irreflexive transitive relation "$<$". A *linear extension* of a partially ordered set P on n vertices is a linear ordering "$\prec$" of the vertices such that $x \prec y$ whenever $x < y$ in P. For a poset P, let $\Lambda(P)$ denote the set of linear extensions of P and set $N(P) = |\Lambda(P)|$ as the number of linear extensions of P. For two incomparable elements $x, y \in P$, $\Pr(x \prec y)$ denotes the probability that x precedes y in a uniformly randomly chosen linear extension of P.

The problem of determining the number of linear extensions of a partially ordered set is intrinsically connected with sorting. At each stage of a comparison-based sorting algorithm current information makes up a poset, while future comparisons will lead to the unique linear extension of this poset describing the sorted order. If the number of linear extensions of a poset were easy to compute, one could try to find the best pair of elements to compare at each step in the sorting process so as to minimize the maximum number of linear extensions of the two resulting posets, i.e., $N(P \cup (x, y))$ and $N(P \cup (y, x))$. According to a result of Kahn and Saks [53], there is always a good pair to compare, i.e., one that splits no worse than $3/11 : 8/11$. It is worth mentioning that results of this kind, however with weaker constants but simpler proofs were also obtained by Karzanov and Khachiyan [62] and by Kahn and Linial [52], independently, with both derivations using geometric arguments. We note that the classic $1/3 : 2/3$ conjecture of Kislitsyn says that $3/11$ could be replaced by $1/3$; see also the survey by Brightwell [19] dedicated to balanced pairs.

Karzanov and Khachiyan [59] showed that a very natural random walk on the set of linear extensions of a poset is *rapidly mixing*. Specifically, they gave a randomized algorithm with the following properties. The input is a partially ordered set P on n elements and a positive number $\varepsilon > 0$. The output is a linear extension of P, where for any $\lambda \in \Lambda(P)$,

$$\left|\Pr(\lambda \text{ is output}) - \frac{1}{N(P)}\right| \leq \frac{\varepsilon}{N(P)}.$$

For a poset P on n elements, the *linear extension graph* $G(P)$ has vertex set $\Lambda(P)$, and two linear extensions λ, μ are adjacent if they differ by an adjacent transposition. As such, the degree $d(\lambda)$ of a vertex in $G(P)$ is at most $n - 1$. The Markov chain used in [59] works according to the following transition matrix:

$$p(\lambda, \mu) = \begin{cases} 1/(2n-2) & \lambda \text{ and } \mu \text{ are adjacent}, \\ 1 - d(\lambda)/(2n-2) & \lambda = \mu, \\ 0 & \text{otherwise}. \end{cases}$$

They used geometric arguments in conjunction to conductance arguments specific to the analysis of Markov chains to show that the algorithm runs in $O(n^6 \log n \log(1/\varepsilon))$ time. Specifically, they proved the following isoperimetric inequality. Let Q be a convex body in $\mathbb{R}^n$ partitioned into two volumes u and v by a cut of surface area s. Then $s > \min(u, v)/\mathrm{diam}(Q)$. Having an efficient (polynomial-time) almost uniform generator of linear extensions leads to an efficient algorithm for approximating $\Pr(x \prec y \mid P)$, where x and y are elements of poset P. Further, this yields a *fully polynomial randomized approximation scheme* (FPRAS) for counting linear extensions; see [20, 50, 100]. For instance, the Karzanov-Khachiyan chain yields a randomized algorithm with the following properties. The input is a partially ordered set P on n elements and positive numbers ε, β. The output is a number L (the estimate count) such that

$$\Pr(\left(\left|\frac{L}{N(P)} - 1\right| > \varepsilon\right) \leq \beta.$$

The algorithm runs in $O(n^9 \log^6 n \log(1/\varepsilon)\, \varepsilon^{-2} \log(1/\beta))$ time.

Several Markov chains techniques have yielded randomized algorithms that generate approximately uniform samples in $O(n^3 \log n)$ expected time [21]. More recently, Wilson [111] obtained a tight bound of $\Theta(n^3 \log n)$ on the mixing time of the original Karzanov-Khachiyan chain, while Huber [48] constructed an algorithm that generates uniform samples within the same time.

The problem of counting linear extensions of a partial order P on n elements $\{a_1, \ldots, a_n\}$ can be viewed as a special case of the problem of computing the volume of a polytope in $\mathbb{R}^n$. To see this connection [20, 59], define the *order polytope* $Q(P)$ as

$$\{\mathbf{x} \in [0,1]^n : \; x_i \leq x_j \text{ whenever } a_i < a_j \text{ in } P\}.$$

Apart from a set of measure zero (where some pair of coordinates are equal), $[0,1]^n$ can be partitioned into simplices

$$Q_\sigma = \{\mathbf{x} \in [0,1]^n : x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}\},$$

each of volume $1/n!$. Finally, one sees that $Q(P)$ is made up of those simplices Q_σ where $a_{\sigma(1)} \prec a_{\sigma(2)} \prec \cdots \prec a_{\sigma(n)}$ is a linear extension of P. Consequently, the volume of $Q(P)$ equals $N(P)/n!$, and the relation is established.

On one side, positive results about computing the volume of convex bodies in $\mathbb{R}^n$ yield positive results about counting linear extensions. In particular, a fully polynomial randomized approximation scheme (FPRAS) for computing the volume of a convex polytope (Vol($Q(P)$) in particular) yields one for counting linear extensions of a partial order [30]. It is worth noting key progress in the field of random generation and approximate counting in the late 1980s [50, 58, 100] followed by work dedicated to volume computation by randomized algorithms in the early 1990s [30, 79, 80]. The hardness of exact volume computation and the limitations of deterministic algorithms for this task have been shown in several works [8, 29, 39, 76].

On the other side, negative results about counting linear extensions imply negative results about volume computation. In particular the result of Brightwell and Winkler [20] that the problem of counting linear extensions of a poset is # P-complete implies that the problem of computing the volume of an n-dimensional polytope is strongly # P-complete. This was first pointed out by Khachiyan [63]; see also [20, p. 240]. A few other problems at the interface between combinatorial algorithms and volume computation are outlined in his survey [63].

As remarked in [20], none of the schemes mentioned earlier (either relying on randomized algorithms for volume computation or on Markov chains for uniform or almost uniform generation of linear extensions) gives an algorithm for approximately counting linear extensions that is truly practical.

Problem 12. *Does there exist an efficient scheme for approximately counting linear extensions that does not rely on (almost) uniform generation of such extensions?*

Uniform generation and approximate counting is of high interest in intrinsically geometric problems too. In this regard, we like to recall the following problem from [117]:

Problem 13. *Does there exist an efficient algorithm for uniform generation of a simple polygon on a given set of n points in the plane? What is the complexity of the corresponding counting problem?*

6 The state of the art of linear programming

Linear programming has a rich and beautiful history that dates back to at least the time of Fourier (around 1824-1827); see, e.g., [91, pp. 209–225] for a brief history. The simplex method, formulated by Dantzig around 1947, has proved to be a *practically* efficient method for solving large-scale real-world linear programs (LPs), and is still in use today with many engineered variants and software packages available (see e.g., [94]). However, beginning with the famous example of Klee and Minty [69], it was realized that the simplex algorithm can take exponential time, in the worst case, to reach an optimal solution, even for very simple problem instances and under several pivoting rules.

In 1979, Khachiyan [60, 61] proved that any linear programming problem can be solved in polynomial time in the *bit model* of computation. More precisely, he showed that the *ellipsoid method*, previously proposed by Shor [98] and by Yudin and Nemirovskiĭ [114] for nonlinear programming, can be used to solve linear programs with n variables and m constraints in $O(mn^3L)$ arithmetic operations, where L is the *total bit length* of the input LP. Even though this method did not compete practically with the simplex method on real-world instances, it provided a powerful tool to prove polynomial-time solvability of many problems in combinatorial optimization, due to its flexibility with respect to how the constraints are presented (e.g., using *separation* or *membership* oracles) [45]. In fact, it is now becoming a standard tool in the area of approximation algorithms (see, e.g., [107]) to approximately solve NP-hard problems based on linear programming relaxations having an exponential (or even infinite) number of constraints, which are shown to be solvable in polynomial time by the ellipsoid method; see also [14, 36, 49, 73, 97] for examples of more recent applications of the ellipsoid method in algorithmic game theory, geometry, and stochastic optimization. Typically, combinatorial or more efficient algorithms were later discovered for many of these problems, but there is still a good number of problems for which the ellipsoid method is the only *deterministic* polynomial-time algorithm known till today.

Besides its theoretical importance, Khachiyan's result also paved the road for other polynomial-time algorithms for linear programming; the first major result came in 1984, when Karmarkar [57] introduced an *interior point algorithm* that required $O(nL)$ iterations, each performing $O(n^{2.5})$ arithmetic operations (essentially solving a system of linear equations); his algorithm was also claimed to be practically comparable with the simplex method. Following this, a large body of work on interior point algorithms emerged, most notably improving the number of iterations to $O(\sqrt{n}L)$, and extending these methods to more general convex programs; see, e.g., [86, 87]. Among these, for example, is the matrix scaling/balancing algorithm due to Kalantari and Khachiyan [56]. Further theoretical and practical improvements were obtained over the last two decades (either in terms of the number of iterations or the number of operations per iteration); the notably most recent progress is reported in [74], where the number of iterations is improved to $O(\sqrt{r}L)$ (while solving a polylogarithmic number of linear systems per iteration), with r being the rank of the constraint matrix; see also [75] for subsequent improvements on the time per iteration.

In a third line of work, random sampling methods [11, 55] have been used to design randomized algorithms for solving linear (and convex) programming problems. This work was inspired by randomized algorithms for volume computation of convex bodies, starting from the work of Dyer, Frieze and Kannan [30]. The success of these methods relies heavily on the efficient sampling from a *logconcave* distribution defined on a convex set, and the standard method for this is to design a random walk and show that the corresponding Markov chain has a small mixing time (this is essentially the time until the Markov chain is close to its stationary distribution). The current best bound on the mixing time, due to Lovász and Vemplala [81], is[1] $\tilde{O}(n^3)$ (after some preprocessing), for convex sets in $\mathbb{R}^n$. An improvement on this bound would imply a better algorithm for the class of linear programming algorithms based on random sampling.

As linear programming is known to be *P-complete* [27], it is natural to look for parallel algorithms for important special cases. Among these is the class of *positive* LPs, also known as *packing-covering* LPs. An early and inspiring paper in this area was written by Grigoriadis and Khachiyan [44], who considered the "equivalent" problem of zero-sum games, and gave a parallel randomized algorithm for approximately solving such games; the (expected) sequential complexity of this algorithm is $O(\rho^2\varepsilon^{-2}(n+m)\log(n+m))$ for matrices with m rows, n columns and entries in $[-\rho,\rho]$. Independently, Luby and Nisan [82] gave a *width-independent* parallel fully polynomial time approximation scheme (FPTAS) for packing and covering LPs, where the dependence of the running time on the accuracy ε is $\tilde{O}(\varepsilon^{-4})$, independent of the *width* parameter ρ. A series of subsequent papers provided more efficient (sequential or parallel) FPTASs or extended the results to more general settings; see, e.g., [43, 68, 112, 70]. An important issue in this area is the dependence of the running time on the accuracy parameter ε; for more than twenty years, the best bounds stood at $O(\varepsilon^{-2})$ for sequential algorithms and $\tilde{O}(\varepsilon^{-4})$ for parallel algorithms. A major progress has been recently achieved, improving these bounds to $\tilde{O}(\varepsilon^{-1})$ [115, 109] and $\tilde{O}(\varepsilon^{-2})$ [108, 116], respectively.

It was observed by practitioners that worst-case examples for the simplex method, such as the Klee-Minty example [69], are not robust with respect to random perturbations, and thus the performance of the simplex method is much better in reality than its worst-case. In an attempt to understand this phenomena, Spielmann and Teng [102] proposed the *smoothed analysis* of algorithms as a framework to study the performance of algorithms under random perturbations of worst-case instances. They showed that under Gaussian perturbations of the constraint matrix, the *shadow vertex* simplex algorithm has smoothed polynomial-time complexity, that is, its expected running time on any such perturbed instance is polynomial in the dimensions of the constraint matrix and the inverse of the standard deviation of the perturbations; this strongly extends earlier work on *average case analysis* of the simplex method, such as the one given by Smale [101].

An algorithm for linear programming is said to be *strongly polynomial* if the number of arithmetic operations required by the algorithm in the worst-case depends polynomially on the number of variables n and the number of constraints m, but is independent of the input size (bit length) L (and the space used by the algorithm is bounded by a polynomial in the size of the input). A number of special cases of LP are known to be solvable in strongly polynomial time, e.g., Tardos' algorithm for the minimum cost and multicommodity flow problems [104]. Perhaps the most important open question in the area of linear programming is:

Problem 14. *Does there exist a strongly polynomial-time algorithm for linear programming?*

Megiddo gave a *deterministic* (strongly) linear time algorithm for fixed dimension $n = O(1)$ [85]; the running time of his algorithm is $O(2^{2^{n+2}}m)$. Subsequent improvements on this bound, some using randomization, were given in [26, 93]. A major progress was made by Kalai [54] and by Matoušek, Sharir and Welzl [84], who devised a simplex-type method with a *randomized* pivoting rule that takes an expected number of $O(n^2m + \exp(O(\sqrt{n\log n})))$ arithmetic operations to solve any linear program. An almost matching lower bound for two natural randomized pivoting rules was given recently in [42]. It is interesting to note that the subexponential algorithms in [54, 84]

[1]The notation $\tilde{O}()$ suppresses polylogarithmic factors.

can, in fact, be used to solve a more general class of problems known as *LP-type* problems [96], which include for example, the problem of computing an optimal pair of stationary strategies for *mean payoff games* [47], a problem for which Gurvich, Karzanov and Khachiyan [46] gave one of the earliest known (pseudo-polynomial time) algorithms. This latter problem is known to lie in the intersection of NP and co-NP, and resembles in many senses LP. Finding a polynomial-time algorithm for solving mean payoff games remains an outstanding open problem:

Problem 15. *What is the complexity of solving mean payoff games?*

References

[1] S. D. Abdullahi, Vertex Enumeration and Counting for Certain Classes of Polyhedra, Ph.D. thesis, University of Leeds, U.K., 2003.

[2] I. Adler and R. Shamir, A randomized scheme for speeding up algorithms for linear and convex programming problems with high constraints-to-variables ratio, *Mathematical Programming* **61(1)** (1993), 39–52.

[3] K. M. Anstreicher, Improved complexity for maximum volume inscribed ellipsoids, *SIAM Journal on Optimization* **13(2)** (2002), 309–320.

[4] D. Avis, B. Bremner, and R. Seidel, How good are convex hull algorithms? *Computational Geometry: Theory and Applications* **7** (1997), 265–301.

[5] D. Avis and K. Fukuda, A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete and Computational Geometry* **8(3)** (1992), 295–313.

[6] D. Avis and K. Fukuda. Reverse search for enumeration, *Discrete Applied Mathematics* **65(1-3)** (1996), 21–46.

[7] D. Avis, G. D. Rosenberg, R. Savani, and B. von Stengel, Enumeration of nash equilibria for two-player games, *Economic Theory* **42(1)** (2010), 9–37.

[8] I. Bárány and Z. Füredi, Computing the volume is difficult, in *Proc. 18th Annual ACM Symposium on Theory of Computing*, ACM, 1986, pp. 442–447.

[9] M. Bădoiu and K. L. Clarkson, Optimal core-sets for balls, *Computational Geometry: Theory and Applications* **40** (2008), 14–22.

[10] M. Bădoiu, S. Har-Peled, and P. Indyk, Approximate clustering via core-sets, in *Proc. 34th Annual ACM Symposium on Theory of Computing*, ACM, 2002, pp. 250–257.

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

[12] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, Enumerating minimal dicuts and strongly connected subgraphs and related geometric problems, in *Proc. 10th Int. Conf. on Integer Programming and Combinatorial Optimization*, New York, 2004, Springer, vol. 3064 of LNCS, pp. 152–162.

[13] E. Boros, K. Elbassioni, V. Gurvich, and K. Makino, Generating vertices of polyhedra and related problems of monotone generation, in *Proc. of the Centre de Recherches Mathématiques*

at the Université de Montréal, special issue on polyhedral computation, (D. Avis, D. Bremner, and A. Deza, editors), vol. 49, 2009.

[14] E. Boros, K. Elbassioni, V. Gurvich, and K. Makino, A convex programming-based algorithm for mean payoff stochastic games with perfect information, *Optimization Letters*, to appear; `https://doi.org/10.1007/s11590-017-1140-y`.

[15] E. Boros, K. Elbassioni, V. Gurvich, and H. R. Tiwary, The negative cycles polyhedron and hardness of checking some polyhedral properties, *Annals of Operations Research* **188(1)** (2011), 63–76.

[16] E. Boros and V. Gurvich, Scientific contributions of Leo Khachiyan (a short overview), *Discrete Applied Mathematics* **156(11)** (2008), 2232–2240.

[17] C. Bouville, Bounding ellipsoids for ray-fractal intersection, *SIGGRAPH Comput. Graph.* **19(3)** (1985), 45–52.

[18] D. Bremner, K. Fukuda, and A. Marzetta, Primal-dual methods for vertex and facet enumeration, *Discrete and Computational Geometry* **20** (1998), 333–357.

[19] G. Brightwell, Balanced pairs in partial orders, *Discrete Mathematics* **201** (1999), 25–52.

[20] G. Brightwell and P. Winkler, Counting linear extensions, *Order* **8** (1991), 225–242.

[21] R. Bubley and M. Dyer, Faster random generation of linear extensions, *Discrete Mathematics* **201** (1999), 81–88.

[22] M. R. Bussieck and M. E. Lübbecke, The vertex set of a 0/1 polytope is strongly $\mathcal{P}$-enumerable, *Computational Geometry: Theory and Applications* **11(2)** (1998), 103–109.

[23] T. M. Chan, Faster core-set constructions and data-stream algorithms in fixed dimensions, *Computational Geometry: Theory and Applications* **35(1-2)** (2006), 20–35.

[24] B. Chazelle and J. Matoušek, On linear-time deterministic algorithms for optimization problems in fixed dimension, *Journal of Algorithms* **21(3)** (1996), 579–597.

[25] V. Chvátal, *Linear Programming*, Freeman, San Francisco, CA, 1983.

[26] K. L. Clarkson, Las Vegas algorithms for linear and integer programming when the dimension is small, *Journal of the ACM* **42(2)** (1995), 488–499.

[27] D. P. Dobkin and R. J. Lipton and S. P. Reiss, Linear Programming is logspace hard for P, *Information Processing Letters* **8** (1979), 96–97.

[28] M. E. Dyer, The complexity of vertex enumeration methods, *Mathematics of Operations Research* **8** (1983), 381–402.

[29] M. Dyer and A. Frieze, On the complexity of computing the volume of a polyhedron, *SIAM Journal on Computing* **17(5)** (1988), 967–974.

[30] M. Dyer, A. Frieze, and R. Kannan, A random polynomial time algorithm for approximating the volume of convex bodies, *Journal of the ACM* **38(1)** (1991), 1–17.

[31] M. E. Dyer and L. G. Proll, An algorithms for determining all extreme points of a convex polytope, *Mathematical Programming* **12** (1977), 81–96.

[32] D. H. Eberly, *3D Game Engine Design, 2nd Edition: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufmann Publishers, San Francisco, 2006.

[33] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, Heidelberg, 1987.

[34] H. Edelsbrunner and L. J. Guibas, Topologically sweeping an arrangement, *J. Comput. Syst. Sci.* **38** (1989), 165–194. Corrigendum: Topologically sweeping an arrangement, *J. Comput. Syst. Sci.* **42** (1991), 249–251.

[35] H. Edelsbrunner, J. O'Rourke, and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM Journal on Computing* **15** (1986), 341–363.

[36] F. Eisenbrand, S. Funke, A. Karrenbauer and D. Matijevic, Energy-aware stage illumination, *Proc. 21st ACM Symposium on Computational Geometry* 2005, pp. 336–345.

[37] K. Elbassioni and K. Makino, Enumerating vertices of 0/1-polyhedra associated with 0/1-totally unimodular matrices; manuscript, 2017.

[38] K. Elbassioni, I. Rauf, and S. Ray, Enumerating minimal transversals of geometric hypergraphs, in *Proc. 23rd Annual Canadian Conference on Computational Geometry*, Toronto, Canada, August 2011.

[39] G. Elekes, A geometric inequality and the complexity of computing volume, *Discrete & Computational Geometry* **1** (1986), 289–292.

[40] M. L. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, *Journal of Algorithms* **21** (1996), 618–628.

[41] R. M. Freund and J. B. Orlin, On the complexity of four polyhedral set containment problems, *Mathematical Programming* **33(2)** (1985), 139–145.

[42] O. Friedmann, T. D. Hansen, and U. Zwick, Subexponential lower bounds for randomized pivoting rules for the simplex algorithm, *Proc. 43rd annual ACM symposium on Theory of computing*, 2011, pp. 283–292.

[43] N. Garg and J. Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, *SIAM Journal on Computing* **37(2)** (2007), 630–652.

[44] M. D. Grigoriadis and L. Khachiyan, A sublinear-time randomized approximation algorithm for matrix games, *Operations Research Letters* **18(2)** (1995), 53–58.

[45] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, vol. 2 of *Algorithms and Combinatorics*, Springer, 1988.

[46] V. Gurvich, A. Karzanov, and L. Khachiyan, Cyclic games and an algorithm to find minimax cycle means in directed graphs, *USSR Computational Mathematics and Mathematical Physics* **28** (1988), 85–91.

[47] N. Halman, Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems, *Algorithmica* **49(1)** (2007), 37–50.

[48] M. Huber, Fast perfect sampling from linear extensions, *Discrete Mathematics* **306** (2006), 420–428.

[49] K. Jain, A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities, *SIAM Journal on Computing* **37(1)** (2007), 303–318.

[50] M. Jerrum, L. Valiant, and V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theoretical Computer Science* **43** (1986), 169–188.

[51] F. John, Extremum problems with inequalities as subsidiary conditions, in *Studies and Essays: Courant Anniversary Volume* (K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, editors), pp. 187–204, Wiley-Interscience, New York, 1948.

[52] J. Kahn and N. Linial, Balancing extensions via Brunn-Minkowski, *Combinatorica* **11(4)** (1991), 363–368.

[53] J. Kahn and M. Saks, Balancing poset extensions, *Order* **11(4)** (1984), 113–126.

[54] G. Kalai, A subexponential randomized simplex algorithm, *Proc. 24th ACM Symposium on Theory of Computing*, 1992, pp. 475–482.

[55] A. T. Kalai and S. Vempala, Simulated annealing for convex optimization, *Math. Oper. Res.* **31(2)** (2006), 253–266.

[56] B. Kalantari and L. Khachiyan, Diagonal matrix scaling and linear programming, *SIAM J. Optim.* **2(4)** (1992), 668–672.

[57] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* **4** (1984), 373-395.

[58] R. Karp, M. Luby, and N. Madras, Monte-Carlo approximation algorithms for enumeration problems, *Journal of Algorithms* **10(3)** (1989), 429–448.

[59] A. Karzanov and L. Khachiyan, On the conductance of order Markov chains, *Order* **8** (1991), 7–15.

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

[61] L. Khachiyan, Polynomial algorithms in linear programming, *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki* **20** (1980), 51-68. [English Translation: *USSR Computational Mathematics and Mathematical Physics* **20** (1980), 53-72.]

[62] L. Khachiyan, Optimal algorithms in convex programming, decomposition and sorting, in *Computers and Decision Problems* (J. Jaravlev, editor), pp. 161–205, Moscow, Nauka, 1989 (in Russian).

[63] L. Khachiyan, Complexity of polytope volume computation, in *New Trends in Discrete and Computational Geometry* (J. Pach, editor), pp. 91–101, Algorithms and Combinatorics, vol. 10, Springer, Berlin, 1993.

[64] L. Khachiyan, On the complexity of approximating extremal determinants in matrices, *Journal of Complexity* **11(1)** (1995), 138–153.

[65] L. Khachiyan, Rounding of polytopes in the real number model of computation, *Mathematics of Operations Research* **21(2)** (1996), 307–320.

[66] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, and V. Gurvich, Generating all vertices of a polyhedron is hard, *Discrete & Computational Geometry* **39(1-3)** (2008), 174–190.

[67] L. Khachiyan and M. J. Todd, On the complexity of approximating the maximal inscribed ellipsoid for a polytope, *Mathematical Programming* **61(2)** (1993), 137–159.

[68] R. Khandekar, Lagrangian relaxation based algorithms for convex programming problems, Ph.D. thesis, Indian Institute of Technology, Delhi, 2004.

[69] V. Klee and G. Minty, How good is the simplex algorithm? In *Inequalities*, volume III, (O. Shisha, editor), pp. 159–175, Academic Press, New York, 1972.

[70] C. Koufogiannakis and N. E. Young, A nearly linear-time PTAS for explicit fractional packing and covering linear programs, *Algorithmica* **70(4)** (2014), 648–674.

[71] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim, Approximate minimum enclosing balls in high dimensions using core-sets, *ACM Journal of Experimental Algorithmics* **8** (2003), Article No. 1.1 (29 pages).

[72] P. Kumar and E. A. Yildirim, Minimum-volume enclosing ellipsoids and core sets, *Journal of Optimization Theory and Applications* **126(1)** (2005), 1–21.

[73] R. Lavi and C. Swamy, Truthful and near-optimal mechanism design via linear programming, *Journal of the ACM* **58(6)** (2011), 25:1–25:24.

[74] Y. T. Lee and A. Sidford, Path finding methods for linear programming: solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow, *Proc. 55th IEEE Annual Symposium on Foundations of Computer Science*, 2014, pp. 424–433.

[75] Y. T. Lee and A. Sidford, Efficient inverse maintenance and faster algorithms for linear programming, *Proc. 56th IEEE Annual Symposium on Foundations of Computer Science*, 2015, pp. 230–249.

[76] N. Linial, Hard enumeration problems in geometry and combinatorics, *SIAM Journal on Algebraic and Discrete Methods* **36(3)** (1986), 331–335.

[77] L. Lovász, Submodular functions and convexity, in *Mathematical Programming: The State of the Art* (M. Grotschel, A. Bachem, and B. Korte, editors), pp. 235–257, Springer, New York, 1983.

[78] L. Lovász, Combinatorial optimization: some problems and trends, DIMACS Technical Report 92-53, Rutgers University, 1992.

[79] L. Lovász and M. Simonovits, The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume, in *Proc. 31st IEEE Symposium on Foundations of Computer Science*, 1990, pp. 346–354.

[80] L. Lovász and M. Simonovits, On the randomized complexity of volume and diameter, in *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, 1992, pp. 482–491.

[81] L. Lovász and S. Vempala, Fast algorithms for logconcave functions: sampling, rounding, integration and optimization, *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006, pp. 57–68.

[82] M. Luby and N. Nisan, A parallel approximation algorithm for positive linear programming, *Proc. 25th Annual ACM symposium on Theory of computing*, 1993, pp. 448–457.

[83] J. Matoušek, *Lectures on Discrete Geometry*, Springer, 2002.

[84] J. Matoušek, M. Sharir, and E. Welzl, A subexponential bound for linear programming, *Algorithmica* **16(4)** (1996), 498–516.

[85] N. Megiddo, Linear programming in linear time when the dimension is fixed, *Journal of the ACM* **31** (1984), 114–127.

[86] Y. Nesterov and A. Nemirovskiĭ, Self-concordant functions and polynomial-time methods in convex programming, Report, Central Economical & Mathematical Institute, USSR Academy of Sciences, 1989.

[87] Y. Nesterov and A. Nemirovskiĭ, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in in Applied Mathematics 13, SIAM, 1994.

[88] C. Papadimitriou, The largest subdeterminant in a matrix, *Bulletin of the Greek Mathematical Society* **25** (1984), 95–105.

[89] M. E. Pfetsch, *The Maximum Feasible Subsystem Problem and Vertex-Facet Incidences of Polyhedra*, Dissertation, TU Berlin, 2002.

[90] J. S. Provan, Efficient enumeration of the vertices of polyhedra associated with network LP's, *Mathematical Programming* **63(1-3)** (1994), 47–64.

[91] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, 1986.

[92] R. Seidel, Output-size sensitive algorithms for constructive problems in computational geometry, Technical report, Cornell University, Ithaca, NY, 1986.

[93] R. Seidel, Small-dimensional linear programming and convex hulls made easy, *Discrete and Computational Geometry* **6** (1991), 423-434.

[94] R. Shamir, The efficiency of the simplex method: a survey, *Management Science* **33(3)** (1987), 301–334.

[95] M. Sharir, Algorithmic motion planning, in *Handbook of Discrete and Computational Geometry* (J. Goodman and J. O'Rourke, editors), pp. 1036–1064, Chapman & Hall, 2nd edition, 2004.

[96] M. Sharir and E. Welzl, A combinatorial bound for linear programming and related problems, *Proc. 9th Annual Symposium on Theoretical Aspects of Computer Science*, 1992, pp. 569–579.

[97] D. B. Shmoys and C. Swamy, An approximation scheme for stochastic linear programming and its application to stochastic integer programs, *Journal of the ACM* **53(6)** (2006), 978–1012.

[98] N. Z. Shor, Utilization of the operation of space dilatation in the minimization of convex functions (in Russian), *Kibernetika (Kiev)* **(1)** (1970), 6–12. [English translation: *Cybernetics* **6** (1970), 7–15.]

[99] B. W. Silverman and D. M. Titterington, Minimum covering ellipses, *SIAM Journal on Scientific and Statistical Computing* **1(4)** (1980), 401–409.

[100] A. Sinclair and M. Jerrum, Approximate counting, uniform generation and rapidly mixing Markov chains, *Information and Computation* **82(1)** (1989), 93–133.

[101] S. Smale, On the average number of steps of the simplex method of linear programming, *Mathematical Programming* **27(3)** (1983), 241–262.

[102] D. A. Spielman and S.-H. Teng, Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time, *Journal of the ACM* **51(3)** (2004), 385–463.

[103] S. P. Tarasov, L.G. Khachiyan, and I. I. Erlikh, The method of inscribed ellipsoids, *Soviet Math. Doklady* **37(1)** (1988), 230–266.

[104] É. Tardos, A strongly polynomial minimum cost circulation algorithm, *Combinatorica* **5(3)** (1985), 247–256.

[105] D. M. Titterington, Optimal design: Some geometrical aspects of d-optimality, *Biometrika* **62(2)** (1975), 313–320.

[106] M. J. Todd and E. A. Yildirim, On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids, *Discrete Applied Mathematics* **155(13)** (2007), 1731–1744.

[107] V. V. Vazirani, *Approximation Algorithms*, Springer, New York, 2001.

[108] D. Wang, Fast Approximation Algorithms for Positive Linear Programs, Ph.D. Thesis, University of California at Berkeley, Technical Report No. UCB/EECS-2017-126.

[109] D. Wang, S. Rao and M. W. Mahoney, Unified acceleration method for packing and covering problems via diameter reduction, *Proc. 43rd International Colloquium on Automata, Languages, and Programming*, 2016, 50:1–50:13.

[110] E. Welzl, Smallest enclosing disks (balls and ellipsoids), in *New Results and New Trends in Computer Science* (H. Maurer, editor), pp. 359–370, vol. 555 of LNCS, Springer, 1991.

[111] D. B. Wilson, Mixing times of lozenge tiling and card shuffling Markov chains, *The Annals of Applied Probability* **14(1)** (2004), 274–325.

[112] N. E. Young, Sequential and parallel algorithms for mixed packing and covering, *Proc. 42nd Annual Symposium on Foundations of Computer Science*, 2001, pp. 538–546.

[113] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets, *Algorithmica* **52(3)** (2008), 378–402.

[114] D. B. Yudin and A. S. Nemirovskiĭ, Informational complexity and efficient methods for the solution of convex extremal problems (in Russian), *Ekonomiku i Matematicheskie Metody* **12** (1976), 357–369. [English translation: *Mutekon* **13(3)** (1977), 25–45.]

[115] Z. A. Zhu and L. Orecchia, Nearly-linear time positive LP solver with faster convergence rate, *Proc. 47th Annual ACM Symposium on Theory of Computing*, 2015, pp. 229–236.

[116] Z. A. Zhu and L. Orecchia, Using optimization to break the epsilon barrier: a faster and simpler width-independent algorithm for solving positive linear programs in parallel, *Proc. 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015, pp. 1439–1456.

[117] C. Zhu, G. Sundaram, J. Snoeyink, and J. S. B. Mitchell, Generating random polygons with given vertices, *Computational Geometry: Theory & Applications* **6** (1996), 277–290.