Math. Program., Ser. A (2017) 164:325–339
DOI 10.1007/s10107-016-1089-0

FULL LENGTH PAPER

Geometric random edge

Friedrich Eisenbrand[1] · **Santosh Vempala**[2]

Received: 31 August 2014 / Accepted: 1 November 2016 / Published online: 21 November 2016

Abstract We show that a variant of the random-edge pivoting rule results in a strongly polynomial time simplex algorithm for linear programs $\max\{c^T x : x \in \mathbb{R}^n,\ Ax \leqslant b\}$, whose constraint matrix A satisfies a geometric property introduced by Brunsch and Röglin: The sine of the angle of a row of A to a hyperplane spanned by $n-1$ other rows of A is at least δ. This property is a geometric generalization of A being integral and each sub-determinant of A being bounded by Δ in absolute value. In this case $\delta \geqslant 1/(\Delta^2 n)$. In particular, linear programs defined by totally unimodular matrices are captured in this framework. Here $\delta \geqslant 1/n$ and Dyer and Frieze previously described a strongly polynomial-time randomized simplex algorithm for linear programs with A totally unimodular. The expected number of pivots of the simplex algorithm is polynomial in the dimension and $1/\delta$ and independent of the number of constraints of the linear program. Our main result can be viewed as an algorithmic realization of the proof of small diameter for such polytopes by Bonifas et al., using the ideas of Dyer and Frieze.

Mathematics Subject Classification 90C05 · 60J10 · 68W20

1 Introduction

Our goal is to solve a linear program

✉ Friedrich Eisenbrand
friedrich.eisenbrand@epfl.ch

Santosh Vempala
vempala@gatech.edu

[1] EPFL, Lausanne, Switzerland

[2] Georgia Tech, Atlanta, GA, USA

$$\begin{aligned} &\max c^T x \\ &Ax \leqslant b \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ is of full column rank, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. The rows of A are denoted by a_i, $1 \leqslant i \leqslant m$. We assume without loss of generality that the rows have Euclidean length one, i.e., that $\|a_i\| = 1$ holds for $i = 1, \dots, m$. The rows shall have the following *δ-distance property* (we use $\langle . \rangle$ to denote linear span):

> For any $I \subseteq [m]$, and $j \in [m]$, if $a_j \notin \langle a_i : i \in I \rangle$ then $d(a_j, \langle a_i : i \in I \rangle) \geqslant \delta$. In other words, if a_j is not in the span of the a_i, $i \in I$, then the distance of a_j to the subspace that is generated by the a_i, $i \in I$ is at least δ.

In this paper, we analyze the simplex algorithm [8] to solve (1) with a variant of the *random edge* pivoting rule. Our main result is a strongly polynomial running time bound for linear programs satisfying the δ-distance property.

Theorem 1 *There is a random edge pivot rule that solves a linear program using poly$(n, 1/\delta)$ pivots in expectation. The expected running time of this variant of the simplex algorithm is polynomial in n, m and $1/\delta$.*

The δ-distance property is a geometric generalization of the algebraic property of an integer matrix having small sub-determinants in absolute value. (Recall that a $k \times k$ *sub-determinant* of A is the determinant of a sub-matrix that is induced by a choice of k rows and k columns of A.) To see this, let $A \in \mathbb{Z}^{m \times n}$ with each of its sub-determinants bounded by Δ in absolute value. Let $B \subseteq \{1, \dots, m\}$ be a *basis of* A, i.e., an index set satisfying $|B| = n$ and $\langle a_i : i \in B \rangle = \mathbb{R}^n$. Let A_B be the sub-matrix of A that is induced by the rows indexed by B and let $w \in \mathbb{Q}^n$ be the column of A_B^{-1} with $a_i^T w = 1$. The distance of a_i to $\langle a_j : j \in B - i \rangle$ is the absolute value of

$$\frac{a_i^T w}{\|w\|}. \tag{2}$$

By Cramer's rule, each entry of w is a $(n-1) \times (n-1)$ sub-determinant of A_B divided by $\det(A_B)$. The division by $\det(A_B)$ cancels out in (2). After this cancellation, the numerator is an integer and the denominator is at most $\sqrt{n} \cdot \Delta$. This shows that the absolute value of (2) is at least $1/(\sqrt{n} \cdot \Delta)$. To bound δ from below, this distance needs to be divided by $\|a_i\| \leqslant \sqrt{n} \cdot \Delta$ since the rows of the matrix should be scaled to length one when we measure the distance. Thus an integer matrix satisfies the δ-distance property with $\delta \geqslant 1/(n \cdot \Delta^2)$.

This shows that our result is an extension of a randomized simplex-type algorithm of Dyer and Frieze [9] that solves linear programs (1) for *totally unimodular* $(\Delta = 1)$ $A \in \{0, \pm 1\}^{m \times n}$ and arbitrary b and c. In this case, $\delta \geqslant 1/n$.

1.1 Related work

General linear programming problems can be solved in weakly polynomial time [16, 17]. This means that the number of basic arithmetic operations performed by the algorithm is bounded by a polynomial in the binary encoding length of the input.

It is a longstanding open problem whether there exists a *strongly polynomial time* algorithm for linear programming. Such an algorithm would run in time polynomial in the dimension and the number of inequalities on a RAM machine. For general linear programming, the simplex method is sub-exponential [15,21].

The δ-distance property is a geometric property and not an algebraic one. In fact, the matrix A can be irrational even. This is in contrast to a result of Tardos [26], who gave a strongly polynomial time algorithm for linear programs whose constraint matrix has integer entries that are bounded by a constant. The algorithm of Tardos is not a simplex-type algorithm.

Spielman and Teng [25] have shown that the simplex algorithm with the *shadow-edge pivoting rule* runs in expected polynomial time if the input is randomly perturbed. This *smoothed analysis* paradigm was subsequently applied to many other algorithmic problems. Brunsch and Röglin [5] have shown that, given two vertices of a linear program satisfying the δ-distance property, one can compute in expected polynomial time a path joining these two vertices with $O(mn^2/\delta^2)$ edges in expectation with the shadow-edge rule. However, the two vertices need to be known in advance. The authors state the problem of finding an optimal vertex w.r.t. a given objective function vector c in polynomial time as an open problem. We solve this problem and obtain a path whose length is independent of the number m of inequalities.

Klee and Minty [18] have shown that the simplex method is exponential if Dantzig's original pivoting rule is applied. More recently, Friedmann et al. [11] have shown that the *random edge* pivot rule results in a superpolynomial number of pivoting operations. Here, random edge means to choose *an improving* edge uniformly at random. The authors also show such a lower bound for *random facet*. Friedmann [10] also recently showed a superpolynomial lower bound for *Zadeh's* pivoting rule. Nontrivial, but exponential upper bounds for random edge are given in [12,14].

Bonifas et al. [3] have shown that the diameter of a polytope defined by an integral constraint matrix A whose sub-determinants are bounded by Δ is polynomial in n and Δ and independent of the number of facets. In the setting of the δ-distance property, their proof yields a polynomial bound in $1/\delta$ and the dimension n on the diameter that is independent of m. Our result is an extension of this result in the setting of linear programming. We show that there is a variant of the simplex algorithm that uses a number of pivots that is polynomial in $1/\delta$ and the dimension n.

1.2 Assumptions

Throughout we assume that c and the rows of A, denoted by a_i, $1 \leqslant i \leqslant m$, have Euclidean norm $\|\cdot\|_2$ one. We also assume that the linear program is *non-degenerate*, meaning that for each feasible point x^*, there are at most n constraints that are satisfied by x^* with equality. It is well known that this assumption can be made without loss of generality [23]. Furthermore, we assume that $n \geqslant 4$, i.e., that the number of variables is at least 4.

2 Identifying an element of the optimal basis

Before we describe our variant of the random-edge simplex algorithm, we explain the primary goal, which is to identify one inequality of the optimal basis. Then, we can

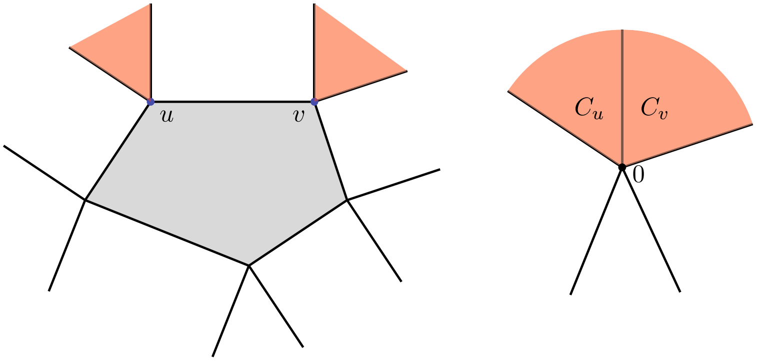

Fig. 1 Two neighboring vertices u and v and their normal-cones C_u and C_v

continue with the search for other basis elements by running the simplex algorithm in one dimension lower.

Let $K = \{x \in \mathbb{R}^n : Ax \leqslant b\}$ be the *polyhedron* of *feasible solutions* of (1). Without loss of generality, see Sect. 1, we can assume that K is a *polytope*, i.e., that K is bounded. Let $v \in K$ be a *vertex*. The *normal cone* C_v of v is the set of vectors $w \in \mathbb{R}^n$ such that, if $c^T x$ is replaced by $w^T x$ in (1), then v is an optimal solution of that linear program. Equivalently, let $B \subseteq \{1, \dots, m\}$ be the n row-indices with $a_i^T v = b_i$, $i \in B$, then the normal cone of v is the set $\operatorname{cone}\{a_i : i \in B\} = \{\sum_{i \in B} \lambda_i a_i : \lambda_i \geqslant 0,\ i \in B\}$. The cones C_u and C_v of two vertices $u \neq v$ intersect if and only if u and v are neighboring vertices of K. In this case, they intersect in the common facet $\operatorname{cone}\{a_i : i \in B_u \cap B_v\}$, where B_u and B_v are the indices of tight inequalities of u and v respectively, see Fig. 1.

Suppose that we found a point $c' \in \mathbb{R}^n$ together with a vertex v of K whose normal cone C_v contains c' such that:

$$\|c - c'\| < \delta/(2 \cdot n). \tag{3}$$

The following variant of a lemma proved by Cook et al. [6] shows that we then can identify at least one index of the optimal basis of (1). We provide a proof of the lemma in the appendix.

Lemma 2 *Let $c' \in \mathbb{R}^n$ and let $B \subseteq \{1, \dots, m\}$ be the optimal basis of the linear program (1) and let B' be an optimal basis of the linear program (1) with c being replaced by c'. Consider the conic combination*

$$c' = \sum_{j \in B'} \mu_j a_j. \tag{4}$$

For $k \in B' \backslash B$, one has

$$\|c - c'\| \geqslant \delta \cdot \mu_k.$$

Following the notation of the lemma, let B' be the optimal basis of the linear program with objective function $c'x$. Since (3) holds and since $\|c\| = 1$ we have $\|c'\| > 1 - \delta/(2 \cdot n)$. This means that there exists a μ_k with $\mu_k > 1/n \cdot (1 - \delta/(2 \cdot n))$. But then k must be in B since $\delta \cdot \mu_k > \delta/n\,(1 - \delta/(2 \cdot n)) \geqslant \delta/(2 \cdot n)$.

Once we have identified an index k of the optimal basis B of (1) we set this inequality to equality and let the simplex algorithm search for the next element of the basis on the induced face of K. This is in fact a $n-1$-dimensional linear program with the δ-distance property. We explain now why this is the case.

Suppose that the element from the optimal basis is a_1. Let $U \in \mathbb{R}^{n\times n}$ be a non-singular orthonormal matrix that rotates a_1 into the first unit vector, i.e.

$$a_1^T \cdot U = e_1^T.$$

The linear program $\max\{c^T x \colon x \in \mathbb{R}^n,\ Ax \leqslant b\}$ is equivalent to the linear program $\max\{c^T U \cdot x \colon x \in \mathbb{R}^n,\ A \cdot U \cdot x \leqslant b\}$. Notice that this transformation preserves the δ-distance property. Therefore we can assume that a_1 is the first unit vector.

Now we can set the constraint $x_1 \leqslant b_1$ to equality and subtract this equation from the other constraints such that they do not involve the first variable anymore. The a_i are in this way projected into the orthogonal complement of e_1. We scale the vectors and right-hand-sides with a scalar $\geqslant 1$ such that they lie on the unit sphere and are now left with a linear program with $n-1$ variables that satisfies the δ-distance property as we show now.[1]

Lemma 3 *Suppose that the vectors $a_1, \ldots, a_m$ satisfy the δ-distance property, then $a_2^*, \ldots, a_m^*$, after being scaled to unit length, satisfy the δ-distance property as well, where a_i^* is the projection of a_i onto the orthogonal complement of a_1.*

Proof Let $I \subseteq \{2, \ldots, m\}$ and $j \in \{2, \ldots, m\}$ such that a_j^* is not in the span of the a_i^*, $i \in I$. Let $d(a_j^*, \langle a_i^* \colon i \in I\rangle) = \gamma$. Clearly, $d(a_j^*, \langle a_i \colon i \in I \cup \{1\}\rangle \leqslant \gamma$ and since a_j^* stems from a_j by subtracting a suitable scalar multiple of a_1, we have $d(a_j, \langle a_i \colon i \in I \cup \{1\}\rangle \leqslant \gamma$ and consequently $\gamma \geqslant \delta$. □

This shows that we can solve the linear programming problem efficiently if we can efficiently solve the following algorithmic problem.

Problem 1 Given an initial vertex of the linear program (1), compute a vector $c' \in \mathbb{R}^n$ together with a vertex v of (1) such that

a) The vector c' is contained in the normal cone C_v of v and
b) $\|c - c'\| < \delta/(2 \cdot n)$ hold.

3 A random walk controls the simplex pivot operations

Next we show how to solve Problem 1 with the simplex algorithm wherein the pivoting is carried out by a random walk. Consider the function

$$f(x) = \exp(-\|x - \alpha \cdot c\|_1), \tag{5}$$

[1] A similar fact holds for totally unimodular constraint matrices, see, e.g., [22, Proposition 2.1, p. 540] meaning that after one has identified an element of the optimal basis, one is left with a linear program in dimension $n-1$ with a totally unimodular constraint matrix. A similar fact fails to hold for integral matrices with sub-determinants bounded by 2.

where $\alpha \geqslant 1$ is a constant whose value will be determined later. Imagine that we could sample a point x^* from the distribution with density proportional to (5) and that, at the same time, we are provided with a vertex v that is an optimal solution of the linear program (1) where c has been replaced by x^*. We are interested in the probability that $c' = x^*/\alpha$ together with v is not a solution of Problem 1. This happens if $\|x^* - \alpha \cdot c\|_2 \geqslant \alpha \cdot \delta/2n$ and since $\|\cdot\|_1 \geqslant \|\cdot\|_2$ one has

$$\|x^* - \alpha \cdot c\|_1 \geqslant \alpha \cdot \delta/2n \tag{6}$$

in this case. The probability of the event (6) is equal to the probability that a random y^* chosen with a density proportional to $\exp(-\|y\|_1)$ has ℓ_1 norm at least $\alpha \cdot \delta/2n$. Since at least one component of y needs to have absolute value at least $\alpha \cdot \delta/2n^2$ to satisfy (6), this probability is upper bounded by

$$n \int_{\alpha \cdot \delta/2n^2}^{\infty} \exp(-x)\, dx = n/\exp(\alpha \cdot \delta/2n^2) \tag{7}$$

by applying the union bound. Thus if $\alpha \geqslant 2n^3/\delta$, this probability is exponentially small in n.

Approximate sampling for log-concave distributions can be dealt with by random-walk techniques [1]. As in the paper of Dyer and Frieze [9] we combine these techniques with the simplex algorithm to keep track of the optimal vertex of the current point of the random walk.

Remember that the normal cones C_v of the vertices v partition the space $\mathbb{R}^n$. Each of these cones is now again partitioned into countably infinitely many parallelepipeds whose axes are parallel to the unit vectors defining the cone of length $1/n^2$ see Fig. 2. More precisely, a given cone $C_v = \{\sum_{i \in B} \lambda_i a_i : \lambda_i \geqslant 0,\ i \in B\}$ is partitioned into translates of the parallelepiped

$$\left\{ \sum_{i \in B} \lambda_i \cdot a_i : 0 \leq \lambda_i \leq 1/n^2,\ i \in B \right\}.$$

The *volume* of such a parallelepiped P is $\text{vol}(P) = (1/n^2)^n |\det(A_B)|$ where A_B is the sub-matrix of A consisting of the rows indexed by the basis B. For a parallelepiped P, we define

$$f(P) = f(z_P)\text{vol}(P)$$

where z_P is the center of P. The state space of the random walk will be all parallelepipeds used to partition all cones, a countably infinite collection. One iteration is as follows.

At the parallelepiped P, pick one of the $2n$ parallelepipeds that share a facet with P uniformly at random, i.e. with probability $(1/2)\,n$. Let this parallelepiped be P' and go to it with probability $\frac{1}{2}\min\{1, f(P')/f(P)\}$.

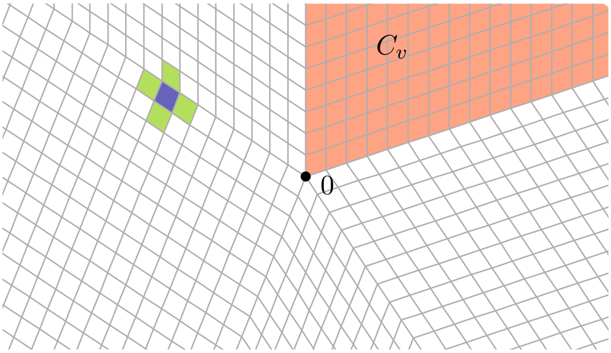

Fig. 2 The partitioning of $\mathbb{R}^2$ into parallelepipeds for the polytope in Fig. 1 together with the illustration of a parallelepiped (*blue*) and its neighbors (*green*) (color figure online)

Notice that P' can be a parallelepiped that is contained in a different cone. If we then make the transition to P' we perform a *simplex pivot*. In this way we can always keep track of the normal cone containing the current parallelepiped. We assume that we are given some vertex of the polytope, x^0 and its associated basis $B_0 \subseteq \{1, \dots, m\}$ to start with. We explain in the appendix how this assumption can be removed. We start the random walk at the parallelepiped of the cone C_{x_0} that has vertex 0. The algorithm is completely described below.

```
Algorithm 1  (Geometric random edge)

  Input:    An LP specified by $A, b, c$; a basic feasible solution $x^0$ of
            $Ax \leq b$ and an associated basis $B_0$.
  Output:   If successful, an element of the optimal basis

1. Let $f(P) = \exp(-\|z_P - \alpha c\|_1)\,\mathrm{vol}(P)$. Start with the parallelepiped $P_0$ in $C_{x_0}$
   containing the point 0.
2. Repeat for $\ell$ iterations:
   • If $c$ is in the current cone, the algorithm stops. It has found the optimal basis.
   • Otherwise pick a neighboring parallelepiped, say $P'$, uniformly at random.
   • Go to $P'$ with probability $\frac{1}{2}\min\left\{1, \frac{f(P')}{f(P)}\right\}$.
     (this is a pivot whenever $P'$ and $P$ are in different cones.)
3. If the center $z_P$ of the final parallelepiped satisfies $\|z_P/\alpha - c\| < \delta/2n$,
   then return an element of the optimal basis as described in Section 2. Return
   failure otherwise.
```

Before we analyze the convergence of the random walk, we first explain the reason behind our choice for $1/n^2$ for the length of the edges of the parallelepipeds. In short, this is because the value of the function $f(x) = \exp(-\|x - \alpha \cdot c\|_1)$ does not vary too much for points in neighboring parallelepipeds. More precisely, we can state the following lemma.

Lemma 4 *Let P_1 and P_2 be two parallelepipeds that share a common facet and let $x_1, x_2 \in P_1 \cup P_2$. If $n \geqslant 4$ then*

$$f(x_1)/f(x_2) \leqslant e.$$

Proof The Euclidean distance between any two points in one of the two parallelepipeds is at most $1/n$ and thus $\|x_1 - x_2\| \leqslant 2/n$. This implies $\|x_1 - x_2\|_1 \leqslant 2/\sqrt{n} \leqslant 1$ since $n \geqslant 4$. Consequently

$$\begin{aligned} f(x_1)/f(x_2) &= \exp(-\|x_1 - \alpha \cdot c\|_1 + \|x_2 - \alpha \cdot c\|_1) \\ &\leqslant \exp(\|x_1 - x_2\|_1) \\ &\leqslant \exp(1). \end{aligned}$$

In the second line of the above equation, we have applied the triangle inequality. □

4 Analysis of the walk

We assume that the reader is familiar with basics of *Markov chains* (see e.g., [20]). The random walk above with its transition probabilities is in fact a Markov chain $\mathcal{M} = (\mathscr{P}, p)$ with a countably infinite *state space* $\mathscr{P}$ which is the set of parallelepipeds and *transition probabilities* $p \colon \mathscr{P} \times \mathscr{P} \longrightarrow \mathbb{R}_{\geqslant 0}$. Let $Q \colon \mathscr{P} \longrightarrow \mathbb{R}_{\geqslant 0}$ be a probability distribution on $\mathscr{P}$. The distribution Q is called *stationary* if for each $P \in \mathscr{P}$ one has

$$Q(P) = \sum_{P' \in \mathscr{P}} p(P', P) \cdot Q(P').$$

Our Markov chain is *lazy* since, at each step, with probability $\geqslant 1/2$ it does nothing. The *rejection sampling* step where we step from P to P' with probability $1/2 \cdot \min\{1, f(P')/f(P)\}$ is called a *Metropolis filter*. Laziness and the Metropolis filter ensure that $\mathcal{M} = (\mathscr{P}, p)$ has the unique stationary distribution (see e.g., Section 1, Thm 1.4 of [19] or Thm 2.1 of [27]):

$$Q(P) = f(P) \Big/ \sum_{P' \in \mathscr{P}} f(P')$$

which is a discretization of the continuous distribution with density $2^{-n} \exp(\|x - \alpha c\|_1)$ from (5).

Performing ℓ steps of the walk induces a distribution Q^ℓ on $\mathscr{P}$ where $Q^\ell(P)$ is the probability that the walk is in the parallelepiped P in the end. In the limit, when ℓ tends to infinity, Q^ℓ converges to Q. We now show that, due to the δ-distance property of the matrix A, the walk quickly converges to Q. More precisely, we only need to run it for a polynomial number (in n and $1/\delta$) iterations. Then Q^ℓ will be sufficiently close to Q which shows that Algorithm 1 solves Problem 1 with high probability.

To prove convergence of the walk, we bound the conductance of the underlying Markov chain [24]. The conductance of $\mathcal{M}$ is

$$\phi = \min_{\substack{S \subseteq \mathscr{P}: \\ 0 < Q(S) \leqslant 1/2}} \frac{\sum_{P \in S, P' \in \mathscr{P} \setminus S} Q(P) \cdot p(P, P')}{Q(S)}. \tag{8}$$

Jerrum and Sinclair [24] related the conductance to the convergence of a finite Markov chain to its stationary distribution. Lovász and Simonovits [19] extended their result to general Markov chains and in particular to Markov chains with a countably infinite set of states like our chain $\mathcal{M} = (\mathscr{P}, p)$. We state their theorem in our setting.

Theorem 5 [19, Corollary 1.5] *Let Q^ℓ be the distribution obtained after ℓ steps of the Markov chain $\mathcal{M} = (\mathscr{P}, p)$ started at the initial parallelepiped P_0 and let ϕ be the conductance as in* (8). *Then for any $T \subseteq \mathscr{P}$ one has*

$$|Q^\ell(T) - Q(T)| \leq Q(P_0)^{-1/2} \left(1 - \frac{\phi^2}{2}\right)^\ell.$$

The rate of convergence is thus $O(1/\phi^2)$. Our goal is now to bound ϕ from below.

4.1 Bounding the conductance

Inspecting (8), we first note that the transition probability $p(P, P')$ is zero, unless the parallelepiped P' is a neighbor of P, i.e., shares a facet with P. In this case one has

$$p(P, P') = \frac{1}{4n} \min\{1, f(P')/f(P)\}.$$

The ratio $f(P')/f(P)$ can be bounded from below by δ/e. This is because $f(z_{P'})/f(z_P) \geqslant 1/e$ by Lemma 4 and since $\text{vol}(P')/\text{vol}(P) \geqslant \delta$. The latter inequality is a consequence of the δ-distance property, as the ratio $\text{vol}(P')/\text{vol}(P)$ is equal to the ratio of the height of P' and the height of P measured from the common facet that they share respectively. This ratio is at least δ. Consequently we have for neighboring parallelepipeds P and P',

$$p(P, P') \geqslant \delta/(4en).$$

Clearly $Q(P) \cdot p(P, P') = Q(P') \cdot p(P', P)$ which implies that the conductance can be bounded by

$$\phi \geqslant \min_{\substack{S \subseteq \mathscr{P}: \\ 0 < Q(S) \leqslant 1/2}} (\delta/4en) \frac{Q(N(S))}{Q(S)}. \tag{9}$$

where $N(S) \subseteq \mathscr{P} \setminus S$ denotes the *neighborhood* of S, which is the set of parallelepipeds $P' \notin S$ for which there exists a neighboring $P \in S$.

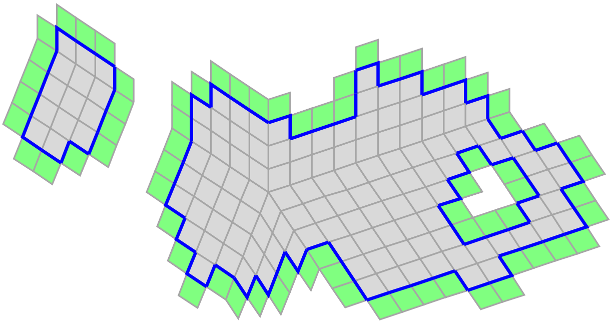

Fig. 3 An illustration of Theorem 6 in our setting. The *gray parallelepipeds* are the set S and the boundary ∂S is in *blue*. The *green parallelepipeds* are $N(S)$ (color figure online)

We next make use of the following isoperimetric inequality that was shown by Bobkov and Houdre [2] for more general product probability measures. For our function $f(x) = \exp(-\|x - \alpha \cdot c\|_1)$ it reads as follows, where $\partial(A)$ denotes the boundary of A.

Theorem 6 [2] *Let* $f(x) = \exp(-\|x - \alpha \cdot c\|_1)$. *For any measurable set* $A \subset \mathbb{R}^n$ *with* $0 < 2^{-n} \int_A f(x)\,dx < 1/2$ *one has*

$$\int_{\partial A} f(x)\,dx \geq \frac{1}{2\sqrt{6}} \int_A f(x)\,dx.$$

Theorem 6 together with the δ-distance property yields a lower bound on $Q(N(S))/Q(S)$ as follows. Each parallelepiped $P' \in N(S)$ has at least one facet F at the boundary $\partial(\cup_{P \in S} P)$, see Fig. 3. Lemma 4 implies that $\int_F f(x)\,dx \leqslant e \cdot \text{Area}(F) \cdot f(z_{P'})$ and since the height of P' w.r.t. F is at least δ one has $\text{Area}(F) \cdot \delta \leqslant \text{vol}(P')$ which implies

$$\int_F f(x)\,dx \leqslant e/\delta \cdot f(z_{P'})\text{vol}(P') = e/\delta \cdot f(P').$$

Since each $P' \in N(S)$ has $2n$ facets, this implies

$$2n\, e/\delta \cdot \sum_{P' \in N(S)} f(P') \geqslant \frac{1}{2\sqrt{6}} \int_{\cup_{P \in S} P} f(x)\,dx \geqslant \frac{1}{2\,e\,\sqrt{6}} \sum_{P \in S} f(P) \tag{10}$$

where we used Lemma 4 again in the last inequality. Thus $Q(N(S))/Q(S) \geqslant \delta/(4e^2\sqrt{6}n)$ which implies a bound of $\Omega(\delta^2/n^2)$ on the conductance.

Lemma 7 *The conductance of the random walk on the parallelepipeds is* $\Omega(\delta^2/n^2)$.

4.2 Bounding the failure probability of the algorithm

Algorithm 1 does not return an element of the optimal basis if

$$\|z_P - \alpha \cdot c\|_2 \geqslant \delta/2n.$$

What is the probability of this event if P is sampled according to Q, the stationary distribution of the walk? If this happens, then the final parallelepiped is contained in the set

$$\{x \in \mathbb{R}^n \colon \|x - \alpha \cdot c\|_2 \geqslant \alpha\delta/2n - 1/n\} \tag{11}$$

since the diameter of a parallelepiped is at most $1/n$. Let $T \subseteq \mathscr{P}$ be the set of parallelepipeds that are contained in the set (11). To estimate $Q(T)$ we use the notation $\widetilde{T}$ to denote the set of points in the union of the parallelepipeds, i.e., $\widetilde{T} = \cup_{P \in T} P$. Using Lemma 4, we see that

$$Q(T) = \sum_{P \in T} f(P) \Big/ \sum_{P \in \mathscr{P}} f(P) \leqslant e^2 \cdot 2^{-n} \cdot \int_{\widetilde{T}} f(x)\, dx.$$

Clearly $2^{-n} \cdot \int_{\widetilde{T}} f(x)\, dx$ is at most the probability that a random point $x^* \in \mathbb{R}^n$ sampled from the distribution with density $2^{-n} \cdot \exp(-\|x\|_1)$ has ℓ_2-norm at least $\alpha \cdot \delta/2n - 1/n \geqslant \alpha \cdot \delta/4n$ where we assume that $\alpha \geqslant 2n^3/\delta$ in the last inequality. Arguing as in (7) we conclude that

$$Q(T) \leqslant e^2 \cdot n/\exp(\alpha \cdot \delta/4n^2).$$

Thus is $\alpha = 4n^3/\delta$, then

$$Q(T) \leqslant e^2 \cdot n/\exp(n) \tag{12}$$

which is exponentially small in n.

How many steps ℓ of the walk are necessary until

$$|Q^\ell(T) - Q(T)| \leqslant \exp(-n) \tag{13}$$

holds? Remember that the conductance $\phi \geqslant \xi \cdot (\delta/n)^2$ for some constant $\xi > 0$. Thus (13) holds if

$$\sqrt{e^2 2^n / f(P_0)} \left(1 - \xi \cdot (\delta/n)^2\right)^\ell < \exp(-n). \tag{14}$$

Now $f(P_0) = f(z_{P_0})\mathrm{vol}(P_0) \geqslant \exp(-\|2\alpha c\|_1) n^{-n} \delta^n \geqslant \exp(-8 \cdot n^{3.5}/\delta)(\delta/n)^n$. Thus (14) holds if

$$\left(1 - \xi \cdot (\delta/n)^2\right)^\ell < \exp(-\nu n^{3.5}/\delta)(\delta/n)^n,$$

where $\nu > 0$ is an appropriate constant. Using the inequality $(1 + x) \leqslant \exp(x)$ and taking logarithms on both sides, this holds if

$$- \ell \cdot \xi \cdot (\delta/n)^2 < -\nu n^{3.5}/\delta + n \ln \delta/n \qquad (15)$$

and thus $\ell = \Omega(n^{5.5}/\delta^3)$ is a polynomial lower bound on ℓ such that (13) holds. With the union bound we thus have our main result.

Theorem 8 *If Algorithm 1 performs $\Omega(n^{5.5}/\delta^3)$ steps of the random walk, then the probability of failure is bounded by $e^2 \cdot n \exp(-n) + \exp(-n)$. Consequently, there exists a randomized simplex-type algorithm for linear programming with an expected running time that is polynomial in n/δ.*

4.3 Remarks

Remark 1 The δ-distance property is a global property of the matrix A. However, we used it only for the *feasible bases* of the linear program when we showed rapid mixing of the random walk. In the first submission of our paper, we asked whether an analog of Lemma 2 also holds for linear programs where a *local* δ-distance property holds. This was answered positively by Dadush and Hähnle [7]. Thus our random walk can be used to solve linear programs whose basis matrices A_B, for each feasible basis B, satisfy the δ-distance property in expected polynomial time in n/δ. The paper of Dadush and Hähnle shows this as well via a variant of the *shadow-vertex* pivoting rule. Their algorithm is much faster than ours. Also Brunsch et. al [4] could recently show that the shadow-vertex pivoting rule results in a polynomial-time algorithm for linear programs with the δ-distance property. Their number of pivot-operations however also depends on the number m of constraints.

Remark 2 Our random walk, and especially the partitioning scheme, is similar to Dyer and Frieze [9] and is directly inspired by their paper. The main differences are the following. Dyer and Frieze rely on a *finite* partitioning of a bounded subset of $\mathbb{R}^n$, adapting a technique of Applegate and Kannan [1]. Our walk, however, is on a countably infinite set of states. For this, we rely on the paper of Lovàsz and Simonovits [19] which extends the relation of the conductance and rapid mixing to more general Markov chains, in particular countably infinite Markow chains. We also choose the ℓ_1-norm, i.e., the function $\exp(-\|x - \alpha \cdot c\|_1)$, whereas Dyer and Frieze used the Euclidean norm. With this choice, the isoperimetric inequality of Bobkov and Houdré [2] makes the conductance analysis much simpler. Finally, we analyze the walk in terms of the δ-distance in a satisfactory way. We feel that this geometric property is better suited for linear programming algorithms, as it is more general.

Acknowledgements The authors are grateful to Daniel Dadush and Nicolai Hähnle, who pointed out an error in the sub-division scheme in a previous version of this paper.

Appendix

Proof of Lemma 2 We denote the normal cones of B and B' by

$$C = \left\{ \sum_{i \in B} \lambda_i a_i : \lambda_i \geqslant 0 \right\} \quad \text{and} \quad C' = \left\{ \sum_{j \in B'} \mu_j a_j : \mu_j \geqslant 0 \right\}.$$

By a gift-wrapping technique, we construct a hyperplane $(h^T x = 0)$, $h \in \mathbb{R}^n \backslash \{0\}$ such that the following conditions hold.

(i) The hyperplane separates the interiors of C and C'.
(ii) The row a_k does not lie on the hyperplane.
(iii) The hyperplane is spanned by $n - 1$ rows of A.

Once we construct $(h^T x = 0)$ we can argue as follows. The distance of $\mu_k \cdot a_k$ to the hyperplane $(h^T x = 0)$ is at least $\mu_k \cdot \delta$. Since c' is the sum of $\mu_k \cdot a_k$ and a vector that is on the same side of the hyperplane as a_k it follows that the distance of c' to this hyperplane is also at least $\mu_k \cdot \delta$. Since c lies on the opposite side of the hyperplane, the distance of c and c' is at least $\mu_k \cdot \delta$.

We start with a hyperplane $(h^T x = 0)$ strictly separating the interiors of C and C'. The conditions (i, ii) are satisfied. Suppose that (iii) is not satisfied and let $\ell < n - 1$ be the maximum number of linearly independent rows of A that are contained in $(h^T x = 0)$.

We tilt the hyperplane by moving its normal vector h along a chosen equator of the ball of radius $\|h\|$ to augment this number. Since $\ell < n - 1$ there exists an equator leaving the rows of A that are in contained in $(h^T = 0)$ invariant under each rotation.

However, as soon as the hyperplane contains a new row of A we stop. If this new row of A is not a_k then still, conditions (i, ii) hold and the hyperplane now contains $\ell + 1$ linearly independent rows of A.

If this new row is a_k, then we redo the tilting operation but this time by moving h in the opposite direction on the chosen equator. Since there are n linearly independent rows of A without the row a_k this tilting will stop at a new row of A which is not a_k and we end the first tilting operation.

This tilting operation has to be repeated at most $n - 1 - |B \cap B'|$ times to obtain the desired hyperplane. □

Phase 1

We now describe an approach to determine an initial basic feasible solution or to assert that the linear program (1) is infeasible. Furthermore, we justify the assumption that the set of feasible solutions is a bounded polytope. This *phase 1* is different from the usual textbook method since the linear programs that we need to solve have to comply with the *δ-distance property*.

To find an initial basic feasible solution, we start by identifying n linearly independent linear inequalities $\widetilde{a}_1^T x \leqslant \widetilde{b}_1, \ldots, \widetilde{a}_n^T x \leqslant \widetilde{b}_n$ of $Ax \leqslant b$. Then we determine a ball that contains all feasible solutions. This standard technique is for example described in [13]. Using the normal-vectors $\widetilde{a}_1, \dots, \widetilde{a}_n$ we next determine values $\beta_i, \gamma_i \in \mathbb{R}$, $i = 1, \dots, n$ such that this ball is contained in the parallelepiped $Z = \{x \in \mathbb{R}^n \colon \beta_i \leqslant \widetilde{a}_i^T x \leqslant \gamma_i,\ i = 1, \dots, n\}$. We start with a basic feasible solution x_0^* of this parallelepiped and then proceed in m iterations. In iteration i, we determine a basic feasible solution x_i^* of the polytope

$$P_i = Z \cap \left\{x \in \mathbb{R}^n \colon a_j^T x \leqslant b_j,\ 1 \leqslant j \leqslant i\right\}$$

using the basic feasible solution x_{i-1}^* from the previous iteration by solving the linear program

$$\min \left\{a_i^T x \colon x \in P_{i-1}\right\}.$$

If the optimum value of this linear program is larger than b_i, we assert that the linear program (1) is infeasible. Otherwise x_i^* is the basic feasible solution from this iteration.

Finally, we justify the assumption that $P = \{x \in \mathbb{R}^n \colon Ax \leqslant b\}$ is bounded as follows. Instead of solving the linear program (1), we solve the linear program $\max\{c^T x \colon x \in P \cap Z\}$ with the initial basic feasible solution x_m^*. If the optimum solution is not feasible for (1) then we assert that (1) is unbounded.

References

1. Applegate, D., Kannan, R.: Sampling and integration of near log-concave functions. In: STOC ’91: Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, pp. 156–163. ACM, New York (1991)
2. Bobkov, S.G., Houdré, C.: Isoperimetric constants for product probability measures. Ann. Probab. **25**(1), 184–205 (1997)
3. Bonifas, N., Di Summa, M., Eisenbrand, F., Hähnle, N., Niemeier, M.: On sub-determinants and the diameter of polyhedra. In: Proceedings of the 28th Annual ACM Symposium on Computational Geometry, SoCG ’12, pp. 357–362 (2012)
4. Brunsch, T., Großwendt, A., Röglin, H.: Solving totally unimodular lps with the shadow vertex algorithm. In: 32nd International Symposium on Theoretical Aspects of Computer Science, p. 171 (2015)
5. Brunsch, T., Röglin, H.: Finding short paths on polytopes by the shadow vertex algorithm. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) Automata, Languages, and Programming, pp. 279–290. Springer, Berlin(2013)
6. Cook, W., Gerards, A.M.H., Schrijver, A., Tardos, E.: Sensitivity theorems in integer linear programming. Math. Program. **34**, 251–264 (1986)
7. Dadush, D., Hähnle, N.: On the shadow simplex method for curved polyhedra. In: 31st International Symposium on Computational Geometry, SoCG 2015, June 22–25, 2015, Eindhoven, The Netherlands, pp. 345–359 (2015)
8. Dantzig, G.B.: Maximization of a linear function of variables subject to linear inequalities. In: Koopmans, T.C. (ed.) Activity Analysis of Production and Allocation, pp. 339–347. Wiley, New York (1951)
9. Dyer, M., Frieze, A.: Random walks, totally unimodular matrices, and a randomised dual simplex algorithm. Math. Program. **64**(1, Ser. A), 1–16 (1994)
10. Friedmann, Oliver.: A subexponential lower bound for Zadeh’s pivoting rule for solving linear programs and games. In: Integer Programming and Combinatoral Optimization, pp. 192–206. Springer, Berlin (2011)

11. Friedmann, O., Hansen, T.D., Zwick, U.: Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In: STOC'11—Proceedings of the 43rd ACM Symposium on Theory of Computing, pp. 283–292. ACM, New York (2011)
12. Gärtner, B., Kaibel, V.: Two new bounds for the random-edge simplex-algorithm. SIAM J. Discrete Math. **21**(1), 178–190 (2007)
13. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, Volume 2 of Algorithms and Combinatorics. Springer, Berlin (1988)
14. Hansen, T.D., Paterson, M., Zwick, U.: Improved upper bounds for random-edge and random-jump on abstract cubes. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 874–881. Society for Industrial and Applied Mathematics (2014)
15. Kalai, Gil.: A subexponential randomized simplex algorithm (extended abstract). In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC92), pp. 475–482 (1992)
16. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4), 373–395 (1984)
17. Khachiyan, L.G.: A polynomial algorithm in linear programming. Dokl. Akad. Nauk SSSR **244**, 1093–1097 (1979)
18. Klee, V., Minty, G.J.: How good is the simplex algorithm? In: Inequalities, III (Proceedings of Third Symposium, Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin), pp. 159–175. Academic Press, New York (1972)
19. Lovász, L., Simonovits, M.: Random walks in a convex body and an improved volume algorithm. Random Struct. Algorithms **4**(4), 359–412 (1993)
20. Lovász, L.: Random walks on graphs. Combinatorics Paul Erdos Eighty **2**, 1–46 (1993)
21. Matoušek, J., Sharir, M., Welzl, E.: A subexponential bound for linear programming. Algorithmica **16**(4–5), 498–516 (1996)
22. Nemhauser, G.L., Wolsey, L.A.: Integer programming. In: Nemhauser, G.L., et al. (eds.) Optimization, Volume 1 of Handbooks in Operations Research and Management Science, pp. 447–527. Elsevier, Amsterdam (1989)
23. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, London (1986)
24. Sinclair, A., Jerrum, M.: Approximate counting, uniform generation and rapidly mixing markov chains. Inf. Comput. **82**, 93–133 (1989)
25. Spielman, D.A., Teng, S.-H.: Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. J. ACM **51**(3), 385–463 (2004). (electronic)
26. Tardos, É.: A strongly polynomial algorithm to solve combinatorial linear programs. Oper. Res. **34**(2), 250–256 (1986)
27. Vempala, S.: Geometric random walks: a survey. MSRI Comb. Comput. Geom. **52**, 573–612 (2005)