SERIE B — INFORMATIK

Randomized simplex algorithms on Klee-Minty cubes$^{\diamond}$

Bernd Gärtner*
Günter M. Ziegler**

B 94–13
May 1994

Abstract

We investigate the behavior of randomized simplex algorithms on special linear programs.

For this, we develop combinatorial models for the Klee-Minty cubes [17] and similar linear programs with exponential decreasing paths. The analysis of two most natural randomized pivot rules on the Klee-Minty cubes leads to (nearly) quadratic lower bounds for the complexity of linear programming with random pivots. Thus, we disprove two bounds conjectured in the literature.

At the same time, we establish quadratic upper bounds for random pivots on the linear programs under investigation. This motivates the question whether some randomized pivot rules possibly have quadratic worst-case behavior on general linear programs.

*Institut für Informatik, Fachbereich Mathematik und Informatik, Freie Universität Berlin, Takustr. 9, D-14195 Berlin, Germany, E-mail: gaertner@inf.fu-berlin.de.

**Konrad-Zuse Zentrum für Informationstechnik (ZIB), Heilbronner Str. 10, D-10711 Berlin, E-mail: ziegler@zib-berlin.de. Supported by a grant in the "Gerhard-Hess Programm" of the German Science Foundation (DFG).

$^{\diamond}$This work was supported by the ESPRIT Basic Research Action No. 7141 (project ALCOM II).

1 Introduction

Linear programming is the problem of minimizing a linear objective function over a polyhedron $P \subseteq \mathbb{R}^n$ given by a system of m linear inequalities.

Without loss of generality [23] we may assume that the problem is primally and dually nondegenerate, that the feasible region is full-dimensional and bounded, and that the objective function is given by the last coordinate. In other words, we consider the problem of finding the "lowest vertex" (minimizing x_n) of a simple n-dimensional polytope $P \subseteq \mathbb{R}^n$ with at most m facets, where the last coordinate x_n is not constant on any edge, and thus the lowest vertex is unique.

In this setting, the (geometric interpretation of the) simplex algorithm proceeds from some starting vertex of P along edges in such a way that the objective function decreases, until the unique lowest vertex of P is found. The (theoretical and practical) efficiency of the simplex algorithm [24] depends on a suitable choice of decreasing edges that "quickly leads to the lowest vertex". Connected to this are two major problems of linear programming: the diameter problem "Is there a short path to the lowest vertex?", and the algorithm problem "Is there an algorithm which quickly finds a (short) path to the lowest vertex?".

The **diameter problem** is closely related to the "Hirsch conjecture" (from 1957) and its variants [5, 15, 27]. Currently there is no counterexample to the "Strong monotone Hirsch conjecture" [27] that there always has to be a decreasing path, from the vertex which maximizes x_n to the lowest vertex, of length at most $m - n$. On the other hand, the best arguments known for upper bounds establish paths whose length is roughly bounded by $m^{\log_2 2n}$ [12, 27].

The **algorithm problem** includes the quest for a strongly polynomial algorithm for linear programming. Klee & Minty [17] showed in 1972 that linear programs with exponentially long decreasing paths exist, and that the "steepest descent" pivot rule can be tricked into selecting such a path. Using variations of the Klee-Minty constructions, it has been shown that the simplex algorithm may take an exponential number of steps for virtually every deterministic pivot rule [15]. (A notable exception is Zadeh's rule [26, 15], locally minimizing revisits, for which Zadeh's \$1,000.– prize [15, p. 730] has not been collected, yet.)

No such evidence exists for some natural *randomized* pivot rules, among them the following three rules:

RANDOM-EDGE: At any nonoptimal vertex x of P, follow one of the decreasing edges leaving x with equal probability.

RANDOM-FACET: If x admits only one decreasing edge, then take it. Otherwise restrict the program to a randomly chosen facet containing x. This yields a linear program of smaller dimension in which x is nonoptimal, and which can be solved by recursive call to RANDOM-FACET.

RANDOM-SHADOW: Start at the unique vertex $y \in P$ which maximizes x_n. Choose a random unit vector $\mathbf{c}$ orthogonal to $\mathbf{e}_n$. Now take the path from y to the lowest vertex given by $\{x \in P : \mathbf{c}x \leq \mathbf{c}z \text{ for all } z \in P \text{ with } z_n = x_n\}$.

RANDOM-FACET is a randomized version, due to Kalai [12], of Bland's PROCEDURE A [2], which assumes that the facets are numbered, and always restricts to the facet with the smallest index. Interestingly enough, very elementary arguments imply a recursion

$$f(n,m) \leq f(n-1,m-1) + \frac{1}{n}\sum_{i=1}^{n} f(n,m-i)$$

for the maximal expected number of steps $f(n,m)$ on an n-dimensional linear program with m inequalities. From this one can get subexponential upper bounds of roughly $e^{O(\sqrt{n \log m})}$ for the number of steps of RANDOM-FACET — see Kalai [12], and (in a very similar dual setting) Matoušek, Sharir & Welzl [20].

The RANDOM-SHADOW rule is a randomized version of Borgwardt's SHADOW VERTEX ALGORITHM [1] (also known as the GASS-SAATY RULE [16]), for which the auxiliary function **c** is deterministically obtained in such a way that it is minimized on the starting vertex. Borgwardt [1] has successfully analyzed this algorithm under the assumption that P is random in a suitable model (where the secondary objective function **c** can be fixed arbitrarily), and obtained polynomial upper bounds for the expected number of simplex steps.

None of the available evidence contradicts the possibility that the expected running time of all three randomized algorithms we consider is bounded from above by a polynomial, even a quadratic function, in n and m. In this connection, we report investigations of the performance of such algorithms on infinite families of "test problems": specific linear programs which have decreasing paths of exponential length.

It is not generally believed that polynomial upper bounds can be achieved; it is equally conceivable that subexponential bounds such as those by Kalai [12] are essentially best possible. An interesting open problem in this context is to find linear programs on which the algorithms in [12, 20] actually behave superpolynomially; Matoušek [19] has constructed abstract optimization problems — more general than linear programs — for which the subexponential analysis is tight.

In this extended abstract we concentrate on the analysis of the "Klee-Minty cubes", see Section 2. These are very interesting linear programs whose polytope is a deformed n-cube, but for which some pivot rules follow a path through all the vertices and thus need an exponential number of steps.

Our main results are quadratic, respectively nearly quadratic, lower bounds for the expected number of steps taken by the RANDOM-FACET and the RANDOM-EDGE simplex algorithms. For the RANDOM-EDGE rule this seems to be the first superlinear bound.

Specifically, our analysis of random pivots on the Klee-Minty cubes yields the following two theorems.

Theorem 1. *The* RANDOM-FACET *simplex algorithm on the n-dimensional Klee-Minty cube, started at the vertex $\overline{v}$ "opposite" (on the n cube) to the optimal vertex,* *takes a quadratic expected number of steps $F_n(\overline{v})$:*

$$F_n(\overline{v}) = n + 2\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k+2}\binom{n-k}{2} \approx \left(\frac{\pi}{4} - \frac{1}{2}\right) n^2.$$

Moreover, for a random starting vertex the expected number of steps is

$$F_n = \frac{n^2 + 3n}{8}.$$

We note that one gets a linear lower bound and a quadratic upper bound $F_n(x) \leq (n^2 + 3n)/4$ for the expected number of steps from an arbitrary starting vertex x. Furthermore, there are starting points in the upper facet for which the FACET RANDOM rule will take only linearly many steps. The fact that for some starting vertices the expected number of steps is quadratic follows from an explicit formula for the expectation value, given in Section 2, or from the bound for a random starting vertex.

A result very similar to Theorem 1, in the setting of dual simplex algorithms, was earlier obtained by Matoušek [19], who analyzed the behavior of the Matoušek-Sharir-Welzl dual simplex algorithm on a special class of linear programs.

Similarly, for RANDOM-EDGE one gets an upper bound $E_n(x) \leq \binom{n+1}{2}$ for the expected number of steps starting at any vertex x of the n-dimensional Klee-Minty cube, see Section 2. This was first observed by Kelly [13], see also [25].

Theorem 2. *The expected number E_n of steps that the* RANDOM-EDGE *rule will take, starting at a random vertex on the n-dimensional Klee-Minty cube, is bounded by*

$$\frac{n(n+1)}{8H_n} \leq E_n \leq \binom{n+1}{2},$$

where $H_n = 1 + 1/2 + \ldots + 1/n$ is the n-th harmonic number.

This superlinear lower bound requires substantially harder work, see Section 3. It implies that there is a vertex x with $E_n(x) = \Omega(n^2/\log n)$, but compared to the case of RANDOM-FACET we are not able to show this bound for a specific starting vertex, e.g. the top vertex.

Our proof is based on a combinatorial model for the Klee-Minty cubes, which describes the RANDOM-EDGE algorithm as a random walk on an acyclic directed graph (see Section 2).

The combinatorial model also makes it possible to do simulation experiments. Our tests in the range $n \leq 1,000$ suggest that the quadratic upper bound is close to the truth. Also, it seems that a (nearly) quadratic lower bound is valid also if the starting vertex is chosen to be the top vertex of the program, but as mentioned above, our method does not prove this.

Still, our result contradicts Exercise 8.10* in [22, p. 188], where it is claimed that $E_n(x) = O(n)$. It also disproves a conjecture of Kelly [13] that $E_n(x) = O(n(\log n)^2)$ for all starting vertices x.

Another conjecture of Kelly [13], according to which the expected number of RANDOM-EDGE pivots is maximal if the starting vertex is diametrically opposite to the lowest vertex, also turned out to be false. We found, by explicit computation of expectation values (in rational arithmetic, using REDUCE) that the smallest dimension in which this fails is $n = 18$.

The RANDOM-SHADOW algorithm has not yet been studied on special programs. Goldfarb [7, 8] has constructed a variant of the Klee-Minty cubes for which the deterministic SHADOW VERTEX ALGORITHM takes an exponential number of steps. There is hope for a successful analysis since Borgwardt's work [1] shows that methods of integral geometry can be very powerful when applied in this context.

Besides the Klee-Minty cubes and their variants, there are other natural classes of "test problems" for (randomized) linear programming algorithms. They include the deformed products of Klee & Minty [17], for which a combinatorial model is produced in Section 4. Also there is a natural model on polars of cyclic polytopes, for which the actual program has not been constructed, yet. This relates to the unsolved "upper bound problem for linear programs".

2 Combinatorial Models

The *Klee-Minty cubes* [17, 22] are the polytopes of the linear programs in $\mathbb{R}^n$ with $m = 2n$ facets given by

$$\begin{aligned} \min\ x_n :& \\ 0 \leq x_1 &\leq 1 \\ \varepsilon x_{i-1} \leq x_i &\leq 1 - \varepsilon x_{i-1} \end{aligned}$$

for $2 \leq i \leq n$ and $0 < \varepsilon < 1/2$. Our illustration shows the 3-dimensional Klee-Minty cube for $\varepsilon = 1/3$.

Considering the geometry in the limit $\varepsilon \to 0$, one sees that the feasible region is a (slightly) deformed unit cube. Thus the feasible vertices of the program are in bijection with the set $\{0, 1\}^n$ of all 0/1-vectors of length n, where we obtain the 0/1-vector for any vertex by rounding the coordinates. Two vertices are adjacent if the corresponding 0/1-vectors differ in exactly one coordinate. (The identification of $\{0, 1\}$ with $GF(2)$ will turn out useful in the next section, where linear algebra over $GF(2)$ is a key tool in our approach to lower bounds.)

In the following, we identify the vertices of the Klee-Minty cubes with the corresponding 0/1-vectors. Since the simplex algorithm proceeds along decreasing edges, we have to describe the edge orientations. It is easy to see, by induction on the dimension, that if x is a 0/1-vector with k ones, at positions $s_1 < s_2 < \ldots < s_k$, then the x_n-coordinate of the corresponding vertex of the Klee-Minty cube is

$$\varepsilon^{n-s_k} - \varepsilon^{n-s_{k-1}} + \ldots + (-1)^{k-1}\varepsilon^{n-s_1}$$

From this we obtain that if $x, x' \in \{0, 1\}^n$ differ in their i-th component, then the corresponding edge is directed from x to x' if and only if the sum $x_i + x_{i+1} + \ldots + x_n$ is odd. We write $x \to x'$ in this situation.

Figure 1: Klee-Minty cube for $n = 3, \varepsilon = 1/3$

This completes the description of the combinatorial model: a directed, acyclic graph with 2^n vertices, $n2^{n-1}$ directed arcs, and a unique source and sink. It can be used as a combinatorial model for the linear program.

For instance, one can derive that the average length Φ_n of a decreasing path from the highest to the lowest vertex — taking all paths with equal probability — satisfies $\Phi_n \geq (1 + 1/\sqrt{5})^{n-1}$ [11]: it is exponential. Thus, the "average" path is exponentially long, but the RANDOM-EDGE and RANDOM-FACET pivot rules take the long paths with low probability.

The RANDOM-EDGE algorithm moves on the digraph of the Klee-Minty cube by leaving the current vertex, using one of the outgoing edges with equal probability, until it reaches the unique sink in the digraph. For example, a legal sequence of steps for $n = 3$, starting at the highest vertex and ending at the lowest, is given by

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{1} \\ 0 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Here any coordinate that can be flipped is typeset bold: from this one can read off that the first step is taken with probability $p = \frac{1}{3}$, the second one with $p = \frac{1}{2}$, and the third with probability 1. Thus this path is taken with probability $\frac{1}{6}$.

The expected number of steps $E_n(x)$ from a vertex x to the lowest vertex satisfies the recursion

$$E_n(x) = 1 + \frac{1}{\#\{x' : x \to x'\}} \sum_{x' : x \to x'} E_n(x').$$

If $i(x)$ denotes the highest index i for which $x_i = 1$, then we can easily show

$$i(x) \leq E_n(x) \leq \binom{i(x)+1}{2} \leq \binom{n+1}{2}$$

— this implies the upper bound of Theorem 2, but only a linear lower bound. A complete analysis seems to be surprisingly difficult. In Section 3 we develop a method, based on linear algebra over $GF(2)$, that yields the nearly quadratic lower bounds "on average" of Theorem 2.

The RANDOM-FACET pivot rule can, however, be completely analyzed on the Klee-Minty cubes. For this, one first derives that

$$F_n(\mathbf{e}_i) = F_n(\mathbf{e}_i + \mathbf{e}_{i-1}) = i.$$

In particular, started at the highest vertex $\mathbf{e}_n$, the RANDOM-FACET rule only needs an expected number of $F_n(\mathbf{e}_n) = n$ steps. For an arbitrary starting vertex $x \in \{0,1\}^n$, the solution of the program restricted to a facet $x_i = 0$ delivers the lowest vertex; restricted to a facet $x_i = 1$ the algorithm yields the vector $\mathbf{e}_i + \mathbf{e}_{i-1}$, where we set $\mathbf{e}_0 = \mathbf{0}$. From this we get a recursion

$$F_n(x) = \frac{1}{n}\left(\sum_{i=1}^{n} i x_i + \sum_{i=1}^{n} F_{n-1}(x^{(i)})\right),$$

with $x^{(i)} := (x_1, \ldots, x_{i-2}, x_{i-1} + x_i, x_{i+1}, \ldots, x_n)^t \in \{0,1\}^{n-1}$ for $1 \leq i \leq n$. Using this recursion, it is easy to derive a linear lower bound and a quadratic upper bound for $F_n(x)$, namely

$$i(x) \leq F_n(x) \leq \frac{i(x)^2 + 3i(x)}{4} \leq \frac{n^2+3n}{4}.$$

Equality in the linear lower bound holds for the infinite series of vectors $\mathbf{e}_i$ and $\mathbf{e}_i + \mathbf{e}_{i-1}$. Surprisingly, one can explicitly solve the above recursion. In particular, quadratic lower bounds as in Theorem 1 can be derived from the following result.

Proposition 3. *Started at a vertex $x \in \{0,1\}^n$ of the n-dimensional Klee-Minty cube, with*

$$\{t : x_t = 1\} = \{s_1, s_2, \ldots, s_k\}_<,$$

the expected number of steps of the RANDOM-FACET *simplex algorithm is*

$$F_n(x) = \sum_{i=1}^{k} s_i + 2 \sum_{1 \leq i < j \leq k} \frac{(-1)^{j-i} s_i}{s_j - s_i + 1}.$$

For a random starting vertex, the situation is substantially simpler: Let

$$G_n = \sum_{x \in \{0,1\}^n} F_n(x).$$

From the recursion we get

$$G_n = 2^{n-2}(n+1) + 2G_{n-1}$$

with $G_1 = 1$. This gives

$$G_n = 2^{n-2}(\binom{n+2}{2} - 1),$$

and the second part of Theorem 1 follows.

3 A Lower Bound

Our analysis of the RANDOM-EDGE rule on the Klee-Minty cubes starts with a coordinate transformation in $\mathcal{V} := GF(2)^n$. Namely, we associate with every vertex $x \in \mathcal{V}$ the label

$$Tx := (x_n, x_n + x_{n-1}, \ldots, x_n{+}x_{n-1}{+}\ldots{+}x_1)^t \in \mathcal{V}.$$

With these new labels, the vertex set of the digraph is again given by $\mathcal{V}$. An arc of the digraph now corresponds to vertices $x, x' \in \mathcal{V}$ such that $x_i = 1$, and x' arises from x by replacing x_j by $x_j + 1 (\bmod 2)$ for every $j \geq i$. (In particular, this yields $x'_i = 0$.)

Thus, for any vector $x \in \mathcal{V}$, we consider the *game* KM(x):

> choose a random coordinate r for which $x_r = 1$, and flip this coordinate together with all coordinates of higher index. This operation is repeated until the zero vector is reached.

For example, the flipping sequence considered in Section 2 corresponds, after this coordinate transformation, to the sequence

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The version in which we prove the lower bound of Theorem 2 in this section is the following: starting with a random vector $x \in \mathcal{V}$, the expected number $L(x)$ of *rounds played* is at least $cn^2/\log n$ for some $c > 0$.

The flipping operation. The flip at index r (in the new coordinate system) can conveniently be expressed as a linear transformation over $\mathcal{V}$, i.e., there is a matrix A^r such that

$$x^r := (x_1, \ldots, x_{r-1}, 0, x_{r+1} + x_r, \ldots, x_n + x_r)^t = A^r x$$

for all vectors $x = (x_1, \ldots, x_n)$.

The columns of A^r are the images of the unit vectors under the flip at r, i.e.

$$A^r = \left(\begin{array}{ccccccc}
1 & & & \downarrow \text{column } r & & & \\
& \ddots & & & & & \\
& & 1 & & & & \\
& & & 0 & & \leftarrow \text{row } r & \\
& & & 1 & 1 & & \\
& & & \vdots & & \ddots & \\
& & & 1 & & & 1
\end{array}\right),$$

and all other entries are zero. Note that for $j \neq r$, $\mathbf{e}_j{}^r = \mathbf{e}_j$; in general, a flip with $x^r = x$ is called *void*, and although KM(x) does not perform void flips, this more general flipping concept is a key ingredient in our approach.

Flip sequences. Let $\mathcal{S}$ be the set of all (formally infinite) sequences $(s_1, s_2, ...)$ with elements in $\{1, ..., n\}$, where $\text{prob}_{\mathcal{S}}(s_k = r) = 1/n$ independently for all k. We refer to the members of $\mathcal{S}$ as *flip sequences*. For a flip sequence s and an integer k we let $x^{(s,k)}$ be the result of 'applying' the first k flips of s to x, i.e.,

$$x^{(s,k)} := A^{(s,k)}x, \text{ with } A^{(s,k)} := A^{s_k}...A^{s_2}A^{s_1}.$$

The analysis of game KM. It is clear that one can simulate game KM by flipping with a random $r \in \{1, ..., n\}$ in each step and ignoring the void flips. This means that the expected length $L(x)$ of game KM(x) is just the expected number of nonvoid flips encountered during the simulation. Using the linearity of expectation, this boils down to the following formula:

$$L(x) = \sum_{k=1}^{\infty} \underset{\mathcal{S}}{\text{prob}}(x^{(s,k)} \neq x^{(s,k-1)}).$$

Note that

$$L^*(x) := \sum_{k=1}^{\infty} \underset{\mathcal{S}}{\text{prob}}(x^{(s,k)} \neq \mathbf{0})$$

is just the expected length of the simulation, including the void flips - this will be important later. Let us refer to the simulation as game KM*.

Recall that $x^{(s,k)} \neq x^{(s,k-1)}$ if and only if the k-th flip hits a one-entry of the current vector $x^{(s,k-1)}$, which implies that the probability for a nonvoid k-th flip is just the expected number of one-entries in $x^{(s,k-1)}$, divided by n. Thus

$$\underset{\mathcal{S}}{\text{prob}}(x^{(s,k)} \neq x^{(s,k-1)}) = \frac{1}{n} \sum_{r=1}^{n} \underset{\mathcal{S}}{\text{prob}}(x^{(s,k-1)}{}_r = 1).$$

Let

$$L(n) := \frac{1}{2^n} \sum_{x \in \mathcal{V}} L(x)$$

be the *average* expected length (over all vectors x) of game KM. We obtain

$$\begin{aligned} L(n) &= \sum_{k=1}^{\infty} \frac{1}{n} \sum_{r=1}^{n} \operatorname*{prob}_{\mathcal{S},\mathcal{V}}(x^{(s,k-1)}{}_r = 1) \\ &= \frac{1}{n} \sum_{r=1}^{n} \sum_{k=1}^{\infty} \operatorname*{prob}_{\mathcal{S},\mathcal{V}}((A^{(s,k-1)}x)_r = 1) \\ &= \frac{1}{n} \sum_{r=1}^{n} \sum_{k=1}^{\infty} \operatorname*{prob}_{\mathcal{S},\mathcal{V}}((\mathbf{e}_r{}^t A^{(s,k-1)})x = 1) \\ &= \frac{1}{2n} \sum_{r=1}^{n} \sum_{k=1}^{\infty} \operatorname*{prob}_{\mathcal{S}}(\mathbf{e}_r{}^t A^{(s,k-1)} \neq \mathbf{0}), \end{aligned}$$

since $\text{prob}_{\mathcal{V}}((\mathbf{e}_r{}^t A^{(s,k-1)})x = 1)$ is equal to $1/2$ if $\mathbf{e}_r{}^t A^{(s,k-1)} \neq \mathbf{0}$ (and 0 otherwise).

In general, $x^t A^{(s,k-1)}$ arises from x by playing $k-1$ rounds of another flipping game: choose in each round a random index r and replace x_r by $\sum_{r'>r} x_{r'} = x_r + \sum_{r' \geq r} x_{r'}$. But wait! This flipping game is nothing else than the simulation KM*, played in the original combinatorial model of the previous section: if $\sum_{r' \geq r} x_{r'}$ is odd, x_r gets flipped, inducing a proper round of KM – otherwise nothing happens, and a void flip occurs in the simulation. Putting together this observation and the previous derivation gives a simple relation between the average expected length $L(n)$ of game KM and the expected length L^* of game KM* for specific starting vectors.

Lemma 4.

$$L(n) = \frac{1}{2n} \sum_{r=1}^{n} (1 + L^*(T\mathbf{e}_r)),$$

where $T : x \mapsto (x_n, x_n + x_{n-1}, \ldots, x_n + x_{n-1} + \ldots + x_1)^t$ *is the coordinate transformation relating the two combinatorial models of the Klee-Minty cubes.*

Bounding L^*. Lemma 4 leaves us with the problem of determining how many flips (void or nonvoid) are necessary on the average to reduce the vector

$$T\mathbf{e}_r = (0, \ldots, 0, \underbrace{1, \ldots, 1}_{r})$$

to $\mathbf{0}$. To this end we will analyze how the vector evolves when applying a random flip sequence to it. Actually, the analysis will only trace the *dimension* which records the one-entry with lowest index. Therefore, the considerations for $T\mathbf{e}_r$ are valid as well for any other vector x with the same lowest one-entry.

Definition 5. For a nonzero vector $x \in \mathcal{V}$, the dimension is the number

$$d(x) := n + 1 - \min\{r \mid x_r = 1\}.$$

Furthermore, $d(\mathbf{0}) := 0$.

For example, $d(T\mathbf{e}_r) = r$. Now define for $0 \leq d \leq n$ the numbers

$$l(d) := \min_{x \in \mathcal{V}} \{L^*(x) \mid d(x) = d\}.$$

We get $l(0) = 0$ and $L^*(T\mathbf{e}_r) \geq l(r)$, and our objective will be to bound $l(d)$ from below. To this end fix any vector x with dimension d and $L^*(x) = l(d)$, and apply a random flip sequence to it. Eventually the sequence will hit the leading one-entry, thereby decreasing the dimension of the vector currently under consideration. The expected number of flips performed until this happens is exactly n (we have a sequence of Bernoulli-trials with probability of success equal to $1/n$ independently in every trial). The expected number of flips performed after the dimension has decreased depends on the actual dimension obtained. For $i < d$ let p_i denote the probability that the dimension goes down from d to i. Then

$$l(d) \geq n + \sum_{i=0}^{d-1} p_i l(i). \tag{1}$$

Lemma 6. *Let $i < d$. Then*

$$p_0 + \ldots + p_i \leq \frac{1}{d-i}.$$

Proof. $p_0 + \ldots + p_i$ is the probability that the dimension goes down by at least $d - i$, and if this event is possible at all, the flip sequence must necessarily hit the leading one-entry before it hits any of the $d - i - 1$ next higher indices – otherwise there is a zero-entry at the lowest such index which was hit, and this entry turns into one by the time the leading position is flipped, preventing the dimension from advancing by more than $d - i - 1$. However, the probability of hitting the leading one-entry first is exactly $1/(d - i)$. □

From the fact that l is monotone in d (the easy argument is omitted here) it follows that the right hand side of (1) is minimized if the tuple $(p_{d-1}, \ldots, p_0)$ is lexicographically smallest subject to $\sum_{i=0}^{d-1} p_i = 1$ and the inequalities established by Lemma 6. This is the case if $p_i = 1/(d-i) - 1/(d-i+1)$ for $i > 0$, and we get

Lemma 7.

$$l(d) \geq n + \sum_{i=1}^{d-1} (\frac{1}{d-i} - \frac{1}{d-i+1}) l(i).$$

Theorem 8.

$$l(d) \geq \frac{dn}{2(H_{d+1}-1)}.$$

Proof. The inequality of Lemma 7 can be rewritten as

$$\sum_{i=1}^{d} \frac{l(i)}{d-i+1} \geq n + \sum_{i=1}^{d-1} \frac{l(i)}{d-i},$$

and after setting $f(d) := \sum_{i=1}^{d} l(i)/(d-i+1)$ reads as

$$f(d) \geq n + f(d-1)$$

with $f(0)=0$. This implies $f(d') \geq d'n$ for all $d' \leq d$, so

$$\sum_{i=1}^{d'} \frac{l(i)}{d'-i+1} \geq d'n.$$

Summing up the inequalities for all values of d' up to d gives

$$\begin{aligned} \binom{d+1}{2} n &\leq \sum_{d'=1}^{d} \sum_{i=1}^{d'} \frac{l(i)}{d'-i+1} \\ &= \sum_{i=1}^{d} l(i) \sum_{d'=i}^{d} \frac{1}{d'-i+1} \\ &= \sum_{i=1}^{d} l(i) H_{d-i+1}. \end{aligned}$$

While $l(i)$ increases with i, H_{d-i+1} decreases, and Chebyshev's summation inequality [9, p. 38] can be applied to yield

$$\begin{aligned} \sum_{i=1}^{d} l(i) H_{d-i+1} &\leq \frac{1}{d} \left(\sum_{i=1}^{d} l(i) \right) \left(\sum_{i=1}^{d} H_{d-i+1} \right) \\ &= \frac{1}{d}(d+1)(H_{d+1}-1) \sum_{i=1}^{d} l(i). \end{aligned}$$

From this we obtain

$$\frac{d^2 n}{2} \leq (H_{d+1}-1) \sum_{i=1}^{d} l(i),$$

so

$$l(d) \geq \frac{1}{d} \sum_{i=1}^{d} l(i) \geq \frac{dn}{2(H_{d+1}-1)},$$

as claimed. □

Putting everything together. From Lemma 4 we know that

$$L(n) = \frac{1}{2n}\sum_{r=1}^{n}(1 + L^*(T\mathbf{e}_r)) \geq \frac{1}{2n}\sum_{r=1}^{n} l(r),$$

and by Theorem 8 we can argue that

$$\begin{aligned} L(n) &\geq \frac{1}{4}\sum_{r=1}^{n}\frac{r}{H_{r+1}-1} = \frac{1}{4H_n}\sum_{r=1}^{n} r\frac{H_n}{H_{r+1}-1} \\ &\geq \frac{1}{4H_n}\sum_{r=1}^{n} r = \frac{n(n+1)}{8H_n}, \end{aligned}$$

which finally implies Theorem 2.

It can be shown that the lower bound on $l(d)$ derived in Theorem 8 is asymptotically sharp for the recursion of Lemma 7; in order to beat this bound (and show e.g. that $l(d) = \Omega(dn)$) one will have to keep track of more information than the dimension of vectors during the flipping process.

4 Related Models

In this final section, we provide two more combinatorial models for classes of linear programs with exponentially long decreasing paths. A main feature of these two classes — as compared to the Klee-Minty cubes — is that they include polytopes with arbitrarily large number of facets in any fixed dimension. In both classes, we can prove quadratic upper bounds for the running time of RANDOM-EDGE with arbitrary starting vertex.

Deformed products. This class of linear programs was also constructed by Klee & Minty [17]. Its polytopes are combinatorially equivalent to products of 1- and 2-dimensional polytopes. For the following, we restrict to the special case where the dimension n is even, and $P := (C_k)^{n/2}$ is a product of k-gons: an n-dimensional polytope with $m = \frac{kn}{2}$ facets. Such polytopes are now realized in $\mathbb{R}^n$ ("deformed") in such a way that they have an x_n-decreasing path through all the vertices. The geometric construction of these programs is tricky [17], but the combinatorial model is very simple, as follows.

The vertex set of P can naturally be identified with the set of vectors $\{1,\ldots,k\}^{n/2}$. Two vertices are adjacent if their vectors $x, x' \in \{1,\ldots,k\}^{n/2}$ differ in a single coordinate, and in this coordinate the difference is either 1 or $k-1$. Furthermore, the directions for these edges are given as follows: if x and x' differ in their i-th coordinate, then we get a directed edge $x \rightarrow x'$ if either $x_i < x'_i$ and $(x_1,\ldots,x_{i-1})$ contains an even number of even entries, or if $x_i > x'_i$ and $(x_1,\ldots,x_{i-1})$ contains an odd number of even entries. For example, for $n = 4$ and $k = 3$ $(m = 6)$ we get

$11 \to 12 \to 13 \to 23 \to 22 \to 21 \to 31 \to 32 \to 33$
as the directed path through all the vertices.

This explicitly describes a digraph, on which algorithms such as RANDOM-EDGE take a random walk.

Proposition 9. *For an arbitrary starting vertex x on a deformed product program, the expected number of steps taken by the* RANDOM-EDGE *algorithm is bounded by a quadratic function, namely,*

$$E_{n,m}(x) \leq n{\cdot}m.$$

The function $E_{n,m}(x)$ is, however, not even completely analyzed for the case $n = 4$.

For the deformed products, the shortest path from the highest to the lowest vertex visits only these two vertices, while the longest decreasing path visits all the $k^{n/2} = (\frac{2m}{n})^{n/2}$ vertices. In constant dimension this yields a longest decreasing path of length $O(m^{n/2})$, which is asymptotically sharp. However, for other interesting parameter settings, like $m = 2n$, there might be substantially longer paths — see the following construction.

Cyclic programs. Here the construction starts with the polars $C_n(m)^\Delta$ of cyclic polytopes [10,27]. These simple polytopes have the maximal number of vertices for given m and n, namely

$$V(n,m) = \binom{m - \lceil \frac{n}{2} \rceil}{\lfloor \frac{n}{2} \rfloor} + \binom{m - 1 - \lceil \frac{n-1}{2} \rceil}{\lfloor \frac{n-1}{2} \rfloor},$$

according to McMullen's upper bound theorem [21,27]. The facets of $C_n(m)^\Delta$ are identified with $[m] := \{1, 2, \ldots, m\}$; the vertices correspond to those n-subsets $F \subseteq [m]$ which satisfy "Gale's evenness condition": if $i, k \in [m] \backslash F$, then the set $\{j \in F : i < j < k\}$ has even cardinality.

Now any two sets $F = \{i_1, i_2, ..., i_n\}_<$ and $G = \{j_1, j_2, ..., j_n\}_<$ satisfying Gale's evenness condition are compared by the following *twisted lexicographic order*: $F < G$ if and only if $i_1 < j_1$,
or $i_1 = j_1, \ldots, i_k = j_k, i_{k+1} < j_{k+1}$, and i_k is even,
or $i_1 = j_1, \ldots, i_k = j_k, i_{k+1} > j_{k+1}$, and i_k is odd.

Thus one compares the first element in which the (sorted) sets F and G differ, and takes the natural order if the element before that is even (or doesn't exist), and the reversed order if the element before is odd. For example, for $C_4(8)^\Delta$ we get the ordering
$1678 < 1568 < 1458 < 1348 < 1238 < 1234 < 1245 < 1256 < 1267 < 1278 < 2378 < 2367 < 2356 < 2345 < 3456 < 3467 < 3478 < 4578 < 4567 < 5678$.

Now we use this ordering to construct the digraph model. Its vertices are the sets satisfying Gale's evenness condition. There is a directed edge $F \to F'$ if and only if $F' < F$ and F, F' differ in exactly one element, that is, the corresponding vertices of $C_n(m)^{\Delta}$ are adjacent.

The special property of the ordering is that every vertex is adjacent to the previous one. Thus the digraph is acyclic with unique source and sink, and with a directed path through all the vertices. (The construction is derived from Klee [14], where the order is constructed and described recursively.)

It is not clear if one can realize the polytope $C_n(m)^{\Delta}$ such that the x_n-coordinate orders the vertices according to twisted lexicographic order. (Equivalently, we cannot show that this order corresponds to a Bruggesser-Mani shelling [3, 27] of some realization of the cyclic polytope.) If such a realization is possible, then this solves the "upper bound problem for linear programs ":

> "What is the largest possible number $P(n, m)$ of vertices on a decreasing path in a linear program of dimension n with m facets?"

(a very natural unsolved problem!) by showing that the bound $P(n, m) \leq V(m, n)$, from the upper bound theorem for polytopes, holds with equality.

Even without such a realization, the twisted lexicographic ordering yields an abstract objective function in the sense of [25], and thus a shelling of the cyclic polytope $C_n(m)$. Thus our digraph model is a very reasonable "worst case" (?) scenario for the performance of randomized simplex algorithms. Both the RANDOM-EDGE and the RANDOM-FACET variants can, indeed, be analyzed in terms of the digraph model, without use of a metric realization.

Proposition 10. *For the* RANDOM-EDGE *rule, started at an arbitrary vertex F of the cyclic program, there is a linear lower bound and a quadratic upper bound for the expected number of steps. For this, we set $\ell(F) := m + 1 - \min(F)$, with $n \leq \ell(F) \leq m$, and obtain*

$$\ell(F) - n \leq E_{n,m}(x) \leq \binom{\ell(F)+1}{2} - \binom{n+1}{2}.$$

Since both the diameter problem [18, 15] and the algorithm problem [4, 12] have upper bounds that are linear in m, it would be interesting to know that $E_{n,m}(x)$ indeed grows at most linearly in m for such problems. On the other hand, it is certainly challenging to strive for a nonlinear lower bound for these models.

More details for the analysis of the models in this section will appear in [6].

Thanks

We wish to thank M. Reiss, E. Welzl and M. Henk for helpful discussions, comments and contributions, and D. Kühl, W. Schlickenrieder, C. Betz-Haubold, T. Takkula and W. Neun for extensive computations. We are indebted to J. Matoušek for a substantial simplification of the lower bound proof in Section 3.

References

[1] K.H. Borgwardt: *The Simplex Method. A Probabilistic Analysis,* Algorithms and Combinatorics **1**, Springer 1987.

[2] R.G. Bland: *New finite pivoting rules for the simplex method, Math. Operations Research* **2** (1977), 103-107.

[3] H. Bruggesser & P. Mani: *Shellable decompositions of cells and spheres, Math. Scand.* **29** (1971), 197-205.

[4] K. Clarkson: *A Las Vegas algorithm for linear programming when the dimension is small,* in: "Proc. 29. IEEE Symposium on Foundations of Computer Science (FOCS)" 1988, 452-457.

[5] G.B. Dantzig: *Linear Programming and Extensions,* Princeton University Press, Princeton 1963.

[6] B. Gärtner, G.M. Ziegler et al.: *Randomized simplex algorithms on special linear programs,* in preparation 1994.

[7] D. Goldfarb: *Worst case complexity of the shadow vertex simplex algorithm,* preprint, Columbia University 1983, 11 pages.

[8] D. Goldfarb: *On the complexity of the simplex algorithm,* preprint, Columbia University 1993, 21 pages; in: "IV. Workshop on Numerical Analysis and Optimization" in Oaxaca, Mexico (ed. S. Gomez), to appear.

[9] R.L. Graham, D.E. Knuth & O. Patashnik: *Concrete Mathematics,* Addison-Wesley, Reading, MA 1989.

[10] B. Grünbaum: *Convex Polytopes,* Interscience, London 1967.

[11] M. Henk: *Note on the Klee-Minty cube,* manuscript, 1993.

[12] G. Kalai: *A subexponential randomized simplex algorithm,* in: "Proc. 24th ACM Symposium on the Theory of Computing (STOC)", ACM Press 1992, 475-482.

[13] D.G. Kelly: *Some results on random linear programs, Methods of Operations Research* **40** (1981), 351-355.

[14] V. Klee: *Paths on polyhedra II, Pacific Journal of Mathematics* **17** (1966), 249-262.

[15] V. Klee & P. Kleinschmidt: *The d-step conjecture and its relatives, Math. Operations Research* **12** (1987), 718-755.

[16] V. Klee & P. Kleinschmidt: *Geometry of the Gass-Saaty parametric cost LP algorithm, Discrete Comput. Geometry* **5** (1990), 13-26.

[17] V. Klee & G.J. Minty, *How good is the simplex algorithm?,* in: "Inequalities III" (O. Sisha, ed.), Academic Press, New York 1972, 159-175.

[18] D. G. Larman: *Paths on polytopes, Proc. London Math. Soc.* **20** (1970), 161-178.

[19] J. Matoušek: *Lower bounds for a subexponential optimization algorithm, Random Structures & Algorithms*, to appear

[20] J. Matoušek, M. Sharir & E. Welzl: *A subexponential bound for linear programming,* in: "Proc. 8th Annual ACM Symp. Computational Geometry" (Berlin 1992), ACM Press 1992, 1-8.

[21] P. McMullen: *The maximum numbers of faces of a convex polytope, Mathematika* **17** (1970), 179-184.

[22] C.H. Papadimitriou & K. Steiglitz: *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall, Inc., Englewood Cliffs, NJ 1982.

[23] A. Schrijver: *Theory of Linear and Integer Programming,* Wiley-Interscience, Chichester 1986.

[24] R. Shamir: *The efficiency of the simplex method: a survey, Management Sci.* **33** (1987), 301-334.

[25] K. Williamson Hoke: *Completely unimodal numberings of a simple polytope, Discrete Applied Math.* **20** (1988), 69-81.

[26] N. Zadeh: *What is the worst case behavior of the simplex algorithm?,* Technical Report No. 27, Dept. Operations Research, Stanford 1980, 26 pages.

[27] G.M. Ziegler: *Lectures on Polytopes,* Graduate Texts in Mathematics, Springer-Verlag New York 1994, to appear.