RANDOM EDGE can be exponential on abstract cubes

JIŘÍ MATOUŠEK

Department of Applied Mathematics and
Institute of Theoretical Computer Science (ITI)
Charles University
Malostranské nám. 25, 118 00 Praha 1
Czech Republic and
Institut für Theoretische Informatik
ETH Zentrum, Zürich, Switzerland

TIBOR SZABÓ*

Institut für Theoretische Informatik
ETH Zentrum, Zürich, Switzerland

Abstract

We prove that RANDOM EDGE, the simplex algorithm that always chooses a random improving edge to proceed on, can take a mildly exponential number of steps in the model of abstract objective functions (introduced by Wiliamson Hoke [27] and by Kalai [16] under different names). We define an abstract objective function on the n-dimensional cube for which the algorithm, started at a random vertex, needs at least $\exp(\text{const} \cdot n^{1/3})$ steps with high probability. The best previous lower bound was quadratic. So in order for RANDOM EDGE to succeed in polynomial time, geometry *must* help.

1 Introduction

The simplex method from 1947 is the oldest linear programming algorithm. It can safely be declared one of the most important algorithms of the twentieth century, and probably it still remains the linear programming algorithm most widely used in practice. Since its discovery, numerous variants of it (pivot rules) have been proposed, and many of them work quite well in practice. However, no variant is known to be polynomial or close to polynomial in the worst case, and many variants have exponential lower bounds in the worst case.

While linear programming can be solved in time polynomially bounded in the *bit size* of the input, a major open problem is its complexity in the *unit-cost model*. That is, what is the smallest $f(n, m)$ such that any linear program in n variables and with m constraints can be solved in time at most $f(n, m)$ if all arithmetic operations are assumed to incur unit cost? It is natural to look for good algorithms in this model among simplex-type algorithms, since a pivot step in the simplex method can usually be implemented with polynomially many arithmetic operations.

Pivot rules and worst-case bounds. Geometrically, the simplex method can be viewed as follows: We have a convex polyhedron $P \subset \mathbf{R}^d$ (given as an intersection of n halfspaces) and a linear objective function c, and we seek a vertex of P minimizing c. There is no substantial loss of generality in assuming that P is bounded and simple and that no two vertices of P have the same value of c. A simplex algorithm starts at some initial vertex of P and at each step it moves from the current vertex v along an edge of P to another vertex w with $c(w) < c(v)$ (this is called a *pivot step*). Typically there are several possible choices of w at each step, and the way of selecting one of them is called a *pivot rule*. The simplex algorithm terminates for every pivot rule, of course, but the difference in the number of steps for different pivot rules may

*Research is supported by the SNF grant 200021-100316/1

be enormous.

Earlier results on the worst-case complexity of various pivot rules are rather discouraging. For Dantzig's original pivot rule, Klee and Minty [19] constructed a class of examples where this rule leads to an exponential number of steps. It is a polytope isomorphic to the cube $[0,1]^n$, but the cube is slightly deformed in such a way that there is a Hamiltonian monotone path, that is, a directed path visiting all vertices such that a suitable linear objective function decreases along it. (We will discuss these Klee–Minty cubes in more detail since they are a key building block in our construction.) Subsequently such worst-case examples were found by various researchers for almost all known determinictic pivot rules; see Goldfarb [13] for an overview and Amenta and Ziegler [4] for a new unified view of these examples.

A substantial progress in worst-case upper bounds was made using *randomized* pivot rules. Kalai [17] and independently Matoušek, Sharir, and Welzl [21] established a subexponential[1] upper bound, $e^{O(\sqrt{m \log n})}$, for the *expected* number of pivot steps of a randomized pivot rule commonly called RANDOM FACET. This bound is still very far from being polynomial but a substantial improvement over straightforward exponential bounds.

Abstract objective functions and similar frameworks. The subexponential analysis of RANDOM FACET and similar pivot rules relies on rather simple and general properties of the objective function on the polytope. It can be phrased in an axiomatic framework that encompasses linear programming but also a number of other geometric optimization problems, such as the smallest enclosing ball for a given set of points in $\mathbf{R}^n$. Several such frameworks have actually been proposed in the literature: We mention *abstract objective functions* (the name is used, e.g., in Kalai [18]; the concept appeared, as far as we know, in Wiliamson Hoke [27] and in Kalai [16]), *LP-type problems* of Sharir and Welzl [25], and *abstract optimization problems* of Gärtner [7].[2] For more information on these frameworks and their relations see, e.g., Gärtner and Welzl [12]. Here we will discuss acyclic unique-sink orientations, which are also equivalent, up to some algorithmic subtleties that do not concern us here, to abstract objective functions. (Actually Kalai in [16] speaks of orientations, not objective functions. Interestingly, in that paper he used them not in a context of linear programming, but rather for proving that a simple polytope is determined by its graph.)

Given a simple convex polytope P with vertex set V, the *graph* of P is the graph $G(P)$ with vertex set V and with edges corresponding to the edges (1-dimensional faces) of P. An acyclic[3] unique-sink orientation (AUSO) of P is an acyclic orientation of $G(P)$ such that the restriction of $G(P)$ to the vertex set of every face of P has exactly one sink (vertex of outdegree zero). A generic linear function on P induces an AUSO: Orient every edge from the vertex with the larger value to the one with the smaller value. The optimum vertex with the smallest value of the objective function becomes the (unique) sink of $G(P)$. Typically, most of the AUSOs of a given polytope are not given by any linear function.

[1] An attentive reader might have noticed that while here we call the function $e^{\sqrt{n}}$ subexponential, the title implicitly calls $e^{n^{1/3}}$ exponential. We believe that this is excusable: what one calls a mountain depends very much on whether one lives in Holland or in Switzerland, for example.

[2] Also the *abstract polytopes* studied by Adler and his coworkers, see e.g. [1], can be considered related. On the other hand, the beautiful work of Aldous [2], which also deals with certain abstract objective functions on cubes, uses a rather different model, where provably no subexponential algorithm exists.

[3] For some purposes, it is also very interesting to consider unique-sink orientations of polytopes that are not necessarily acyclic (see, e.g., [22, 26, 23]), but acyclicity is natural in the context of the simplex algorithm and we will consider exclusively the acyclic case.

Many pivot rules for the simplex algorithm make sense also for polytopes with AUSOs, and as was mentioned above, the only known subexponential worst-case bound for linear programming also works in this more general setting. Moreover, it was shown in [20] that the analysis of RANDOM FACET in [17, 21] is nearly tight: There are AUSOs of the n-dimensional cube for which RANDOM FACET really requires $e^{\Omega(\sqrt{n})}$ steps. So, in order to improve the upper bound for RANDOM FACET for linear programming, one would have to use some property of realizable AUSOs (those induced by actual linear functions) not shared by general AUSOs. A nice initial step in this direction was made by Gärtner [8], who showed that RANDOM FACET runs in expected quadratic number of steps for all realizable AUSOs from the (very restricted) class used as a lower bound in [20]. However, extending such kind of analysis to arbitrary linear programs, or even to all linear programs whose polytopes are isomorphic to cubes, appears very challenging.

Random edge. One can also hope that some of the known pivoting rules, or a newly designed one, could be shown to be polynomial, or at least substantially better than $e^{\sqrt{n}}$, even for arbitrary AUSOs.

Arguably the simplest randomized pivot rule is RANDOM EDGE: among all neighbors of the current vertex with smaller value of the objective function, select one uniformly at random as the next vertex. For example, this is the first among six pivot rules whose deeper study was suggested by Kalai in his survey paper [18].

Despite the simplicity of RANDOM EDGE, very little has been known about its running time, either for AUSOs or for actual linear programs. There are interesting special results, such as an example showing that RANDOM EDGE can be exponential in the *height* (the length of the shortest directed path to the sink) by Broder et al. [6] and an analysis of RANDOM EDGE for d-dimensional polytopes with $d+2$ facets by Gärtner et al. [11], but the best known lower bound in terms of the dimension and number of facets was $\Omega(n^2)$ for the n-dimensional Klee–Minty cube (Balogh and Pemantle [5], slightly improving on Gärtner, Henk, Ziegler[10]). On the other hand, on the examples from [20], which are hard for RANDOM FACET, RANDOM EDGE is easily seen to be at most quadratic. It was quite tempting to believe that it could be polynomial for arbitrary AUSOs or, more modestly, polynomial on all AUSOs of cubes.

Here we partially destroy these hopes by constructing AUSOs on which RANDOM EDGE almost surely needs mildly exponentially many steps to reach the sink. Here is a more precise statement of the result.

Theorem 1 *There is a positive constant c such that for all sufficiently large n there exists an acyclic unique-sink orientation (AUSO) of the n-dimensional cube $[0,1]^n$ such that the algorithm RANDOM EDGE, started at a randomly chosen vertex, with probability at least $1-e^{-cn^{1/3}}$ makes at least $e^{cn^{1/3}}$ steps before reaching the sink.*

Remarks and further work.

1. It seems that only technical obstacles prevent improving the lower bound to $e^{\Omega(\sqrt{n})}$. On the other hand, getting substantially above this looks more challenging, and perhaps one might now try to work more on a nontrivial upper bound.

2. Several deterministic algorithms, like BottomAntipodal and BottomTop suggested by Volker Kaibel were given exponential lower bounds recently by a deterministic construction in a spirit similar to our construction [24]. We believe that our construction, perhaps in combination with other known constructions, could also provide strong lower bounds for other deterministic/randomized pivoting rules. This is a topic for further research.

3. We do not expect our methods to yield *realizable* AUSOs that would be hard for RANDOM EDGE. However, perhaps one could modify the construction so that it satisfies some further necessary conditions for realizability. For example, one can consider the *Holt–Klee condition* [14], which requires that there are k vertex-disjoint oriented paths from the source to the sink in every k-dimensional face. This is known to hold for all realizable AUSOs of arbitrary polytopes but not necessarily for general AUSOs, even in the case of the 3-cube.

2 Preliminaries and a simpler construction

We begin with a construction simpler than the one used in the proof of Theorem 1. We believe that the main idea is best explained on this simpler construction. We conjecture that the simpler construction also gives mildly exponentially long running time for RANDOM EDGE, but a proof appears more difficult and technical than our current proof of Theorem 1.

Preliminaries on AUSOs. Let $e_i \in \{0,1\}^n$ be the vector having 1 at position i and zeros elsewhere. For zero-one vectors v and w, $v+w$ is understood as the modulo 2 sum of v and w. The notation (v,w) stands for the concatenation of the vectors v and w. The zero vector of any dimension is denoted by 0 and the reader is trusted to figure out the correct length of the vector.

From now on, by an AUSO we will mean an acyclic unique-sink orientation of the cube $[0,1]^n$ (we will not consider any other polytopes). The graph of the n-dimensional cube is the usual n-dimensional (graph-theoretic) cube with vertex set $\{0,1\}^n$. The neighbors of a vertex v are $v+e_i$, $i=1,2,\dots,n$.

Formally we identify an n-dimensional AUSO A with its *outmap* $s_A\colon\{0,1\}^n \to \{0,1\}^n$, where $s_A(v)_i = 1$ if the edge $\{v, v+e_i\}$ is oriented from v towards $v+e_i$, and $s_A(v)_i = 0$ otherwise, i.e. if that edge is oriented from $v+e_i$ towards v. Hence we have $s_A(v)_i = 1 - s_A(v+e_i)_i$. It is known that the outmap s_A is a bijection for any AUSO A. For this and other facts about unique sink orientations of cubes see, for example, [26].

We say that two AUSO A and B are *isomorphic* if there is a bijection between the vertices of A and the vertices of B that preserves the oriented edges. We note that there are $2^n n!$ isomorphic copies of a single n-dimensional AUSO, and $n!$ among them (induced by permutations of coordinates) have their sink at 0.

Here are two lemmas, special cases of results of [23], which allow us to construct new AUSOs from old ones. The first lemma uses the product structure of the cube.

Lemma 2 (Blowup construction) [23, Lemma 3] *Let A be an m-dimensional AUSO and for each $u \in \{0,1\}^m$ let B_u be a d-dimensional AUSO. Then the map $s_C\colon\{0,1\}^{m+d} \to \{0,1\}^{m+d}$ defined by*

$$s_C(u,v) = (s_A(u), s_{B_u}(v))$$

is the outmap of an $(m+d)$-dimensional AUSO C.

One can imagine that we blow up each vertex of A to a d-dimensional cube, which is oriented according to some AUSO, generally different for different vertices. For us, however, a complementary view will be more useful: We can obtain C by taking 2^d copies of A and, for each vertex u of A, interconnecting all the 2^d copies of u by a d-dimensional cubic "frame" oriented according to B_u. This is illustrated in Fig. 1.

The second lemma, the heart of our recursion, allows to change the orientation on a smaller subcube under appropriate conditions. Let A be an n-dimensional AUSO and let S be a face of the n-dimensional cube (isomorphic to an m-dimensional cube for some $m \le n$). We call S a *hypersink* of A if all edges

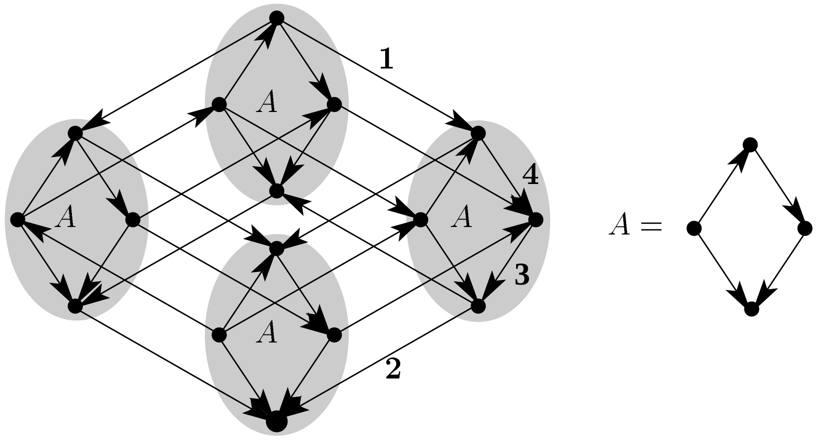

Figure 1: The blowup construction

connecting vertices of S to vertices outside S are oriented towards S.

Lemma 3 (Hypersink reorientation) [23, Lemma 5] *Let A be an n-dimensional AUSO and let S be an m-dimensional hypersink of A. If the edges within S are reoriented according to an arbitrary m-dimensional AUSO B, and the orientations of all other edges are left as in A, then the resulting orientation of the n-dimensional cube is an AUSO.*

The Klee–Minty cube. A basic building block in the simple construction, as well as in the proof of our main result, is the m-dimensional Klee-Minty cube KM_m. First we describe it as an AUSO. The usual definition is recursive. The zero-dimensional cube KM_0 is just a vertex. To construct KM_m, one takes two copies K and K' of KM_{m-1} and flips the orientations of all edges in one of them, say in K'. Then one adds a perfect matching between the vertices of K and K' having identical coordinates and orients these edges from K' towards K. See Fig. 2 for a 3-dimensional illustration.

Here is a more explicit, nonrecursive description of the outmap s_{KM_m}. Let $v \in \{0,1\}^m$ be a vertex, and let us suppose that in the above recursive construction of KM_m,

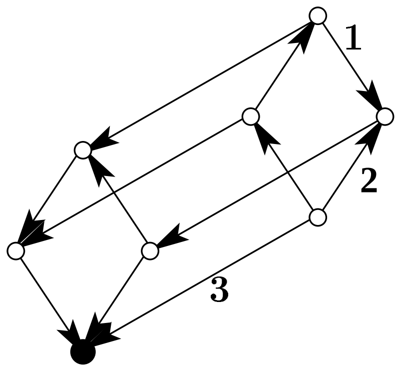

Figure 2: The 3-dimensional Klee-Minty cube. The dimensions are added in the indicated order.

the new coordinate (i.e. the direction of the edges connecting K to K') is always added to the end, so that $v_m = 0$ means that v lies in K and $v_m = 1$ means that v lies in K'. Then it is easy to see that

$$s_{\mathrm{KM}_m}(v) = v^{\Sigma},$$

where the ith coordinate of v^{Σ} is $(v_m + v_{m-1} + \cdots + v_i) \bmod 2$, $i = 1, 2, \ldots, m$.

A simpler construction. In this subsection our explanation will be somewhat informal. Let us suppose that we have already constructed an n_0-dimensional AUSO A_0, with sink at 0, such that RANDOM EDGE started at a random vertex of A_0 almost surely needs a rather long time T_0 to reach the sink. We choose some suitable $m = m(n_0)$, say $m = \sqrt{n_0}$, and for each vertex u of A_0, we choose an m-dimensional AUSO B_u by randomly permuting the coordinates of KM_m, each of the $m!$ coordinate permutations having the same probability and the choices independent for different u. We let C be the blowup of A_0 by these B_u. So, according to our preferred view of the blowup construction, we take 2^m copies of A_0 and interconnect them by the m-dimensional frames B_u, the Klee–Minty cubes with permuted coordinates. All B_u have sink at 0, thus in the same copy S of A_0, and this copy of A_0 is a hypersink in C. We now reorient this hypersink: We form a new n_0-dimensional AUSO A'_0 by selecting one of the $2^{n_0}n_0!$ isomorphic copies of A_0 uniformly at random (note that the sink of A'_0 is not at 0 but rather at a random vertex!). We then orient the hypersink S of C according to A'_0 and we denote the resulting (n_0+m)-dimensional AUSO by A_1.

We introduce some terminology for the following discussion. An edge of A_1 is called a

- *frame edge* if it belongs to one of the 2^{n_0} m-dimensional Klee–Minty frames,
- A_0*-edge* if it belongs to one of the $2^m - 1$ identical copies of A_0, and
- A'_0*-edge* if it belongs to the random isomorphic copy A'_0 of A_0 placed to the hypersink.

Let us consider the behavior of RANDOM EDGE on A_1. First we consider the phase before the walk enters the hypersink (a random starting vertex almost surely doesn't lie in the hypersink). A step along a frame edge can be interpreted as a step of RANDOM EDGE within the appropriate frame B_u, which is isomorphic to KM_m. Each step along an A_0-edge (an A_0*-step*) corresponds to a step of RANDOM EDGE within A_0, but, crucially, an A_0-step also has an interpretation within KM_m: If an A_0-step goes from a vertex (u_1, v) to a vertex (u_2, v) of A_1, where $\{u_1, u_2\}$ is an edge of A_0, we move from the frame B_{u_1} to the frame B_{u_2}. Since A_0 is acyclic, we never reenter an already visited frame, and so we can think of B_{u_2} as obtained from B_{u_1} by a random permutation of coordinates. So if v_1 is the vertex of KM_m corresponding to v in the canonical isomorphism of KM_m with B_{u_1}, and v_2 is the vertex of KM_m corresponding to v in the canonical isomorphism with B_{u_2}, then v_2 *is obtained from* v_1 *by a random permutation of coordinates.* Thus RANDOM EDGE on A_1 induces a random process on KM_m: each frame step corresponds to one step of RANDOM EDGE on KM_m, and each A_0-step corresponds to passing from the current vertex to a vertex obtained by a random permutation of coordinates (we call this a *reshuffle* step).

Conceptually, the hypersink S of A_1 can be reached by two mechanisms:

- We reach the sink of some copy of A_0. Then there will be no more reshuffles and we have the usual RANDOM EDGE on the current frame that, as is well known, reaches the sink of that frame in $O(m^2)$ steps.
- Alternatively, the hypersink is reached without entering the sink of any of the copies of A_0. This means that the random walk with reshuffles reaches the sink of KM_m.

By the assumption of A_0, the first mechanism needs at least T_0 steps almost surely. Intuitively, the second mechanism will also need quite long time, since the random walk with reshuffles is typically going to last long, longer than T_0, provided that reshuffles happen sufficiently often. (The intuition is that even if, from the point of view of RANDOM EDGE on KM_m, we got quite "near the sink" at some moment, a reshuffle is likely going to ruin most of our progress and move us to a vertex quite far away, again in terms of the progress of RANDOM EDGE, not in terms of Hamming distance, say.)

So altogether we almost surely need at least T_0 time before reaching the hypersink S. Since S is a randomly reoriented A_0, no matter where we enter it, the rest of the walk is equivalent to RANDOM EDGE started at a random vertex of A_0, and this is going to last T_0 steps almost surely. So the time at least doubles by passing from A_0 to A_1. If we iterate this construction $\sqrt{n_0}$ times, say, we obtain an AUSO of dimension $n_0 + m\sqrt{n_0} = 2n_0$ where RANDOM EDGE will need $2^{\sqrt{n}_0}T_0$ steps.

The reason why we do not use the simple construction for the proof of our main result is that, in this setting, *the probability of a reshuffle could become too low.* In the next section we will show that a random walk with reshuffles on the Klee–Minty cube almost surely takes exponential time provided that the probability of reshuffle is considerably larger than the probability of a RANDOM EDGE step, but we cannot guarantee this condition in the simple construction. Thus, after the analysis of a random walk with reshuffles, we present a more complicated construction that gets around this obstacle.

3 Walk with reshuffles on the Klee-Minty cube

Let us introduce more formally the random walk with reshuffles discussed in the previous section. The walk starts in a vertex $v^{(0)}$ of KM_m chosen uniformly at random. Being at a vertex $v^{(i-1)}$, the vertex $v^{(i)}$ is chosen as follows:

- With some probability $p^{(i)}_{\mathrm{step}}$ we make one step of RANDOM EDGE; that is, we choose one of the edges going out from $v^{(i-1)}$ uniformly at random and go to the corresponding adjacent vertex, which becomes $v^{(i)}$.
- With some probability $p^{(i)}_{\mathrm{resh}}$ we *reshuffle*: $v^{(i)}$ is obtained from $v^{(i-1)}$ by a random permutation of the coordinates (all the possible $m!$ permutations having the same probability).
- With probability $1 - p^{(i)}_{\mathrm{step}} - p^{(i)}_{\mathrm{resh}}$ we do not move: $v^{(i)} = v^{(i-1)}$.

We assume that $p^{(i)}_{\mathrm{resh}} \geq p_{\mathrm{resh}}$ and $p^{(i)}_{\mathrm{step}} \leq p_{\mathrm{step}}$ for all i, where p_{step} and p_{resh} are some given parameters. The random choice of the upgoing edge and of the reshuffling permutation are independent of all other random choices in the walk. The particular probabilities $p^{(i)}_{\mathrm{step}}$ and $p^{(i)}_{\mathrm{resh}}$ depend on the current state of the random process. In the setting of the previous section, these probabilities are determined by the current position of another random walk, in a certain n-dimensional AUSO with n much larger than m.

The random walk with reshuffles ends when it reaches the sink ($v^{(i)} = 0$). We want to prove that under suitable restriction on p_{step} and p_{resh}, it almost surely needs exponentially many steps:

Proposition 4 *Suppose that $p_{\mathrm{resh}} \geq 11 p_{\mathrm{step}}$ (the constant 11 is rather arbitrary; any sufficiently large constant would do). Then with probability at least $1 - e^{-\alpha m}$ the random walk with reshuffles makes at least $e^{\beta m}$ steps, where α and β are positive constants.*

For a vertex $v \in \{0,1\}^m$ we define the *level* $\ell(v)$ as the number of ones in v. We note that if V is a vertex of level ℓ, then the vertex obtained by a random permutation of the coordinates of v is a random vertex of level ℓ (with all $\binom{m}{\ell}$ choices for the positions of the ℓ ones having the same probability). First we need to bound from above the probability that for a random vertex of level ℓ, with ℓ in a certain range, one step of RANDOM EDGE in the Klee–Minty cube decreases the level (we note that such a step can either increase the level by 1 or decrease it by 1).

Lemma 5 *Let ℓ be given with $\ell_0 \leq \ell \leq m/8$, where ℓ_0 is a sufficiently large constant. Let v be a random vertex of KM_m of level ℓ, and let v' be a random successor of v as in RANDOM EDGE. Then the probability of $\ell(v') = \ell(v)-1$ is at most 0.4.*

The constant 0.4 is certainly not optimal (and the actual bound depends on the rather arbitrary choice of the upper bound $m/8$). For us it is sufficient to have the probability bounded away from $\frac{1}{2}$.

Proof. The number of successors of v is the number of ones in the outmap $s_{\mathrm{KM}_m}(v) = v^{\Sigma}$. If we choose the 1 at the ith position of v^{Σ}, then $v'_i = 1 - v_i$ and $v'_j = v_j$ for $j \neq i$. So the number of successors v' of v with $\ell(v') = \ell(v) - 1$ is the number of indices i with both $v_i = 1$ and $v_i^{\Sigma} = 1$. These i are the positions of ones in v that are followed by an even number of ones in v, and their number is $\lceil \ell/2 \rceil$, for every v of level ℓ.

Now one could easily derive an explicit expression, involving products of binomial coefficients, for the number of v of level ℓ that have a given number of ones in v^{Σ}, and prove the lemma (or a more precise result) by suitable estimates. We present another proof with almost no calculation, using a concentration result for the hypergeometric distribution.

It suffices to prove that with probability at least 0.9 the string v^{Σ} has at least 2ℓ ones. For such strings, the probability of $\ell(v') = \ell(v) - 1$ is at most $\lceil \ell/2 \rceil / 2\ell < \frac{1}{3}$, so the overall probability decreasing the level is at most $0.1 + 0.9 \cdot \frac{1}{3} = 0.4$.

Let $k_1 > k_2 > \cdots > k_\ell$ be the positions of the ℓ ones in v, for convenience numbered in decreasing order; as was remarked above, $L = \{k_1, \ldots, k_\ell\}$ is a random ℓ-subset of $[m]$. Let $g_i = k_{i-1} - k_i$ denote the ith gap in L, with the convention $k_0 = m$ and $k_{\ell+1} = 0$. Then the number of ones in v^{Σ} is $g_2 + g_4 + \cdots + g_{2\lceil \ell/2 \rceil}$, the total size of the "even gaps". Let us construct another ℓ-element subset $L' = \{k'_1, \ldots, k'_\ell\}$ of $[m]$ by taking first the even gaps and then the odd gaps. Namely, we set $k'_1 = g_2$, $k'_2 = k'_1 + g_4, \ldots,$ $k'_{\lceil \ell/2 \rceil} = k'_{\lceil \ell/2 \rceil - 1} + g_{2\lceil \ell/2 \rceil}$, $k'_{\lceil \ell/2 \rceil+1} = k'_{\lceil \ell/2 \rceil} + g_1$, $k'_{\lceil \ell/2 \rceil+2} = k'_{\lceil \ell/2 \rceil+1} + g_3$, and so on. The correspondence of L and L' is bijective, and so L' is also a random ℓ-subset of $[m]$.

The probability that v^{Σ} has less than $q = 2\ell$ ones equals the probability that the even gaps in L sum to less than q, and this is also the probability that the first $\lceil \ell/2 \rceil$ gaps in L' sum to less than q, which in turn is at most the probability that $|L' \cap [q]| > \lceil \ell/2 \rceil$. Here we can apply a tail estimate for the hypergeometric distribution. The expected number of elements of a random ℓ-subset of $[m]$ lying in $[q]$ is $\lambda = \ell q/m$, and the probability of at least $\lambda + t$ elements falling into $[q]$ is at most $e^{-t^2/2(\lambda+t/3)}$; see, for example, the book of Janson et al. [15], Theorem 2.10. In our situation we have $q = 2\ell$, $t = \ell/2 - \lambda$, and with $\ell \leq \frac{m}{8}$ we get $\lambda \leq \frac{\ell}{4}$ and $t \geq \frac{\ell}{4}$. Then $e^{-t^2/2(\lambda+t/3)} = e^{-t/(2\lambda/t+2/3)} \leq e^{-t/3} \leq e^{-\ell/12} \leq e^{-\ell_0/12}$. This can be made as small as desired by choosing ℓ_0 sufficiently large. Lemma 5 is proved. □

Proof of Proposition 4. We consider the sequence $W = (\ell^{(0)}, \ell^{(1)}, \ldots)$, $\ell^{(i)} = \ell(v^{(i)})$, of levels of the vertices in the random walk with reshuffles; this is a kind of random walk on $\{0, 1, \ldots, n\}$. We assume that m is sufficiently large and, for simplicity of notation, that it is divisible by 24. We define a "critical level" $\ell_{\mathrm{crit}} = m/12$.

First we claim that with probability at least $1 - e^{-\Omega(m)}$, the walk starts above the critical level. Indeed, the expected level of a random vertex in $\{0,1\}^m$ is $\frac{m}{2}$, and by the standard Chernoff inequality, the probability of the random vertex having level smaller than $\frac{m}{2} - t$ is at most $e^{-t^2/2m}$.

Let us call the ith step of W *level-changing* if $i = 0$ or $\ell^{(i)} \neq \ell^{(i-1)}$ (that is, if $v^{(i)}$ was obtained from $v^{(i-1)}$ by a RANDOM EDGE step in the Klee–Minty cube). If the random walk with reshuffling on KM_m does move at the ith step, i.e. if $v^{(i)} \neq v^{(i-1)}$, then this move is at least 11 times more likely to be a reshuffling than a RANDOM EDGE step by the assumptions of the proposition. In particular, when the walk does move the first time after the $(j-1)^{st}$ level-changing step, it reshuffles with probability at least $\frac{11}{12}$. So with at least this probability there is at least one reshuffling between the $(j-1)$st level-changing step and the jth level-changing step.

Let $k_j = \ell^{(i_j)}$ be the level at the jth level-changing step. Let us now assume k_{j-1} is in the range $[\ell_0, \frac{m}{8}]$, and let us estimate the probability of $k_j = k_{j-1}-1$. If there is a reshuffling between the $(j-1)$st level-changing step and the jth level-changing step, which happens with probability at least $\frac{11}{12}$, then by Lemma 5 this probability is at most 0.4. Hence the overall probability of $k_j = k_{j-1} - 1$ is at most

$$\frac{1}{12} + \frac{11}{12} \cdot 0.4 = 0.45,$$

and the conditional expectation of k_j given k_{j-1} is at least $k_{j-1} + 0.45(-1) + 0.55 \cdot 1 = k_{j-1} + 0.1$. So there is an expected drift of at least $+0.1$ per level-changing step.

If the walk starts above the critical level, which happens with high probability, it needs to pass the critical level in order to reach 0 and finish. Let j_0 be the smallest j with $k_j = \ell_{\text{crit}}$. Let us call the level-changing steps $j_0, j_0+1, \ldots, j_0+m/24$ the *first attempt*, and we call the first attempt *successful* if it ends up below the critical level, i.e. if $k_{j_0+m/24} < \ell_{\text{crit}}$. If the first attempt was not successful, we can define the second attempt similarly, starting at the first level-changing step $j_1 \geq j_0 + m/24$ with $k_{j_1} = \ell_{\text{crit}}$, and so on. We prove that each attempt succeeds with probability at most $e^{-\Omega(m)}$; this will imply that exponentially many attempts are needed with probability exponentially close to 1, and thus also the desired proposition.

Let the considered attempt start at the jth level-changing step, and for $i = 0, 1, 2, \ldots, m/24$, we define $X_i = k_{j+i} - 0.1i$; this is the level after i level-changing steps of the attempt minus the expected drift. During the whole attempt the level stays in the range $[\ell_{\text{crit}} - m/24, \ell_{\text{crit}} + m/24] \subseteq [\ell_0, m/8]$, and so the conditional expectation of X_i given X_{i-1} is at least X_{i-1}. In other words, the X_i form a *submartingale* (if the conditional expectation of X_i were *equal* to X_{i-1}, we would get the perhaps more familiar notion of a martingale). As is well known and easy to check, standard proofs of Azuma's inequality (see, e.g., Alon and Spencer [3] or Janson et al. [15]) also yield the lower tail estimate for a submartingale instead of a martingale. In our case, we always have $|X_i - X_{i-1}| \leq 1.1$, and Azuma's inequality gives that the probability of $X_{m/24} \leq X_0 - t = \ell_{\text{crit}} - t$ is at most $e^{-24t^2/3m}$. A successful attempt requires $X_{m/24} \leq \ell_{\text{crit}} - 0.1m/24$, so we can set $t = m/240$ and we indeed obtain that the probability of a successful attempt is $e^{-\Omega(m)}$ as needed. Proposition 4 is proved. $\square$

4 The construction

Here we prove Theorem 1. The construction is quite similar to the one from Section 2 but in one iteration we make k blowups by m-dimensional Klee–Minty cubes rather than one.

Let a sufficiently large n be given. We define integer parameters $n_0 = n/2$, $m = n^{1/3}$, $k = Cn^{1/3}$ for a sufficiently large constant C, and $t = n/2km$ (for simplicity, let us assume that the defining expressions indeed come out integral). We define a sequence of AUSO $A_0, A_1, A_2, \ldots, A_t$, where $\dim(A_i) = n_0 + ikm$. The final product A_t has dimension $n_0 + tkm = n$.

We can take more or less any n_0-dimensional AUSO for A_0; as a tribute to Klee and Minty let it be KM_{n_0}.

Having defined A_{i-1}, we construct A_i by describing the orientation of each edge. The vertex set of A_i is $\{0,1\}^{ikm+n_0}$. We partition the coordinates into blocks $B_0, B_1, \ldots, B_k$: the initial block B_0 has length $(i-1)km + n_0$, and it is followed by the blocks $B_1, B_2, \ldots, B_k$

of length m each. For an edge e, the coordinate where the two endpoints of e differ is called the *label* of e. The orientation of an edge will partly depend on its label. As in the simple construction, we fix an AUSO A'_{i-1} selected uniformly at random among all isomorphic copies of A_{i-1}. An edge of A_i with a label in B_0 is oriented as the corresponding edge in A_{i-1} if none of the B_j, $j = 1, \ldots, k$ is all zeros, and otherwise, it is oriented as in A'_{i-1}. To orient the edges with labels in other blocks, we choose a Klee-Minty cube $K_{u,j}$ by randomly permuting the coordinates of KM_m for each $u \in \{0,1\}^{|B_0|}$ and each $j = 1, 2, \ldots, k$. Note that each $K_{u,j}$ has its sink at 0. We orient the edges with label in block B_j according to these Klee-Minty cubes (so the orientation of an edge in B_j depends *only* on the coordinates in B_0 and B_j).

For $v \in \{0,1\}^{ikm+n_0}$, let $v_{[j]}$ denote the restriction of v to the block B_j, $j = 0, 1, \ldots, k$. Formally, an edge of label l from a vertex $v = (v_{[0]}, v_{[1]}, \ldots, v_{[k]})$ is oriented towards v if and only if

- $l \in B_0$, $v_{[j]} \neq 0$ for every $j = 1, \ldots, k$ and the edge of label l is oriented towards $v_{[0]}$ in A_{i-1}, or
- $l \in B_0$, $v_{[j]} = 0$ for some $j = 1, \ldots, k$ and the edge of label l is oriented towards $v_{[0]}$ in A'_{i-1}, or
- $l \in B_j$ and the edge of label l is oriented towards $v_{[j]}$ in $K_{v_{[0]},j}$.

Lemma 6 *A_i is an AUSO.*

Proof. We construct A_i from A_{i-1} using k iterations of the blowup construction (Lemma 2) and hypersink reorientation (Lemma 3). To this end, we decompose the construction of A_i from A_{i-1} into k phases, adding m dimensions in each phase. Let $A_i^0 = A_{i-1}$ and let A_i^j be the AUSO defined after the jth phase, of dimension$((i-1)k+j)m+n_0$. To obtain A_i^{j+1}, we first blow up the vertices of A_i^j by some copies of KM_m. More precisely, with the notation of Lemma 2 we take $A = A_i^j$ and $B_u = K_{u_{[0]},j+1}$. Then we reorient the $(((i-1)k+j)m+n_0)$-dimensional hypersink spanned by the sinks of these Klee–Minty cubes using the following AUSO C_i^j. Leting $C_i^0 = A'_{i-1}$, C_i^j is a blowup of C_i^{j-1}. More precisely, with the notation of Lemma 2, let $A = C_i^{j-1}$ and $B_u = K_{u_{[0]},j}$. □

Let us define some terminology. Let S be the set of the vertices v with $v_{[j]} = 0$ for some $j = 1, \ldots, k$. Note that no (directed) walk can leave S. If the label of an edge is in the block B_0, then depending on wheter its endpoints are in S or not, we call it an *A'_{i-1}-edge* or an *A_{i-1}-edge*, respectively. An edge with label in B_j, $j = 1, 2, \ldots, k$, is called a *j-frame edge.*

Proof of Theorem 1. We prove that with probability $1 - e^{-\Omega(n^{1/3})}$, RANDOM EDGE started at a random vertex of the AUSO A_t constructed above needs at least $e^{\Omega(n^{1/3})}$ steps. The probability in this statement is with respect to the random choices of the algorithm, the random choice of the initial vertex, *and* the random choices involved in the construction of A_t. This implies, by a consideration in the spirit of Fubini's theorem, that there is a *specific* instance of A_t with the behavior of RANDOM EDGE as advertised in Theorem 1.

We prove the following statement for all $i = 0, 1, \ldots, t$ by induction on i: *Assuming that the constant C in the definition of k and t is sufficiently large, the following holds with probability at least $1 - p_i$, where $p_i = 2^{-3t+2i}$: when RANDOM EDGE is started at a random vertex of A_i, the first 2^i steps visit only vertices with outdegree at least k.*

In particular, for $i = t$ we get that with probability at least $1 - 2^{-t} = 1 - e^{-\Omega(n^{1/3})}$ RANDOM EDGE on A_t makes at least $2^t = e^{\Omega(n^{1/3})}$ steps, which implies Theorem 1.

For $i = 0$ the statement holds, since a random vertex of any n_0-dimensional AUSO has outdegree less than k with probability $\frac{1}{2^{n_0}} \sum_{i=0}^{k-1} \binom{n_0}{i} = e^{-\Omega(n)}$ (here we use that the outmap of any AUSO is a bijection, so the number of vertices with outdegree k is $\binom{n}{k}$, plus the standard Chernoff inequality.)

For the inductive step from $i-1$ to i, let us consider the random walk W of RANDOM EDGE on A_i. The cardinality of S is $2^{ikm+n_0} - 2^{(i-1)km+n_0}(2^m-1)^k$, so W starts outside S with probability at least $1-k2^{-m}$. The steps along the A_{i-1}-edges (A_{i-1}-steps for short) made before reaching a vertex of S define a trajectory W_0 of RANDOM EDGE on A_{i-1}, and for each $j = 1, 2, \ldots, k$, the steps using the j-frame edges and the A_{i-1}-steps define a random walk with reshuffles on KM_m, which we call R_j. By the inductive assumption, the following statement $(*)$ holds with probability at least $1-p_{i-1}$:

$(*)$ *The first 2^{i-1} steps of W_0 visit only vertices of outdegree at least k in A_{i-1} and, in particular, they do not reach the sink.*

For a vertex v of A_i not lying in S, let d_j denote the number of outgoing j-frame edges and let d_0 be the number of outgoing A_{i-1}-edges. If RANDOM EDGE on A_i is at v, then the probability of a reshuffle in the corresponding step of R_j is equal d_0/d, where $d = d_0 + d_1 + \cdots + d_k$, while the probability of a RANDOM EDGE step in R_j is d_j/d. We have $d_j \le m$, and by $(*)$, we may assume $d_0 \ge k$ during the first 2^{i-1} steps. Since $k = Cm \ge 11m$, as long as $d_0 \ge k$ holds, the assumptions of Proposition 4 are met by R_j. So assuming $(*)$, with probability at least $1 - e^{-\alpha m}$ R_j does not reach the sink before step $\min(2^{i-1}, e^{\beta m}) = 2^{i-1}$ for large enough C (since $i \le t = n^{1/3}/2C$).

After the first 2^{i-1} steps we cannot guarantee anymore that reshuffles will be frequent enough in R_j. But all vertices of A_i outside S have at least one outgoing j-frame edge for every $j = 1, 2, \ldots, k$, and hence outdegree is still at least k until one of the R_j reaches its sink. At that very moment, i.e. when W first reaches a vertex $v \in S$, $v_{[0]}$ is a random vertex of A_{i-1} and further moves induce a trajectory W_0' of RANDOM EDGE on A_{i-1}', the random isomorphic copy of A_{i-1}. By inductive assumption, with probability at least $1-p_{i-1}$, at least 2^{i-1} steps of W_0' are made through vertices of outdegree at least k in A_{i-1}'. If this happens then, of course, W also makes at least 2^{i-1} steps through vertices with outdegree at least k. Altogether we showed that W makes at least 2^i steps through vertices of outdegree at least k with probability at least $1-k2^{-m}-2p_{i-1}-ke^{-\alpha m}$. With a sufficiently large C we have $k2^{-m}+ke^{-\alpha m} < 2^{-3t}$, and so $k2^{-m}+2p_{i-1}+ke^{-\alpha m} < 2^{-3t+2i-1}+2^{-3t} < 2^{-3t+2i} = p_i$. This finishes the induction step and concludes the proof of Theorem 1. □

Acknowledgment

We would like to thank Ingo Schurr, Uli Wagner and Emo Welzl for stimulating discussions. We are also grateful to Ingo Schurr for performing computer experiments with a preliminary version of our construction.

References

[1] I. Adler, G. B. Dantzig. Maximum diameter of abstract polytopes. *Math. Programming Studies*, 1:20–40, 1974.

[2] D. Aldous. Minimization algorithms and random walk on the d-cube. *The Annals of Probability*, 11:403–413, 1983.

[3] N. Alon, J. Spencer, **The probabilistic method**, 2nd Edition, Wiley, 2000.

[4] N. Amenta and G. M. Ziegler. Shadows and slices of polytopes. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1996.

[5] J. Balogh, R. Pemantle, The Klee-Minty random edge chain moves with linear speed, *preprint*, 2004.

[6] A. Z. Broder, M. E. Dyer, A. M. Frieze, P. Raghavan, and E. Upfal. The worst-case running time of the random simplex

algorithm is exponential in the height. *Inform. Proc. Letters*, 56(2):79–81, 1995.

[7] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.*, 24:1018–1035, 1995.

[8] B. Gärtner. Combinatorial linear programming: Geometry can help. In *Proc. 2nd Int. Workshop on Randomization and Approximation Techniques in Computer Science*, 1518:82–96, 1998. Lecture Notes in Computer Science.

[9] B. Gärtner. The Random-Facet Simplex Algorithm on Combinatorial Cubes. In *Random Structures & Algorithms*, 20(3):353–381, 2002.

[10] B. Gärtner, M. Henk, and G.M. Ziegler. Randomized simplex algorithms on Klee-Minty cubes. *Combinatorica*, 18(3):349–372, 1998.

[11] B. Gärtner, J. Solymosi, F. Tschirschnitz, P. Valtr, and E. Welzl. One lines and n points. In *Proc. 33rd annual ACM Symposium on Theory of Computing (STOC)*, pages 306–315, 2001.

[12] B. Gärtner, E. Welzl. Linear programming – randomization and abstract frameworks. In *Proc. 13th Ann. ACM Symp. Theoretical Aspects of Computer Science*, volume 1046 of *Lecture Notes Comput. Sci.*, pages 669–687. Springer-Verlag, 1996.

[13] D. Goldfarb. On the complexity of the simplex algorithm. In *Advances in optimization and numerical analysis*, pages 25–38, Dordrecht, 1994. Kluwer.

[14] F. Holt, V. Klee, A proof of the strict monotone 4-steps conjecture, *Contemp. Math.* **223** (1999), 201-216.

[15] S. Janson, T. Łuczak, and A. Rućinski. **Random Graphs**, Wiley, 2000.

[16] G. Kalai. A simple way to tell a simple polytope from its graph. *J. Combin. Theory, Ser. A*, 49(2):381–383, 1988.

[17] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.

[18] G. Kalai. Combinatorics with a geometric flavor: Some examples. In *Visions in Mathematics Towards 2000 (GAFA, special volume), part II*, pages 742–792. Birkhäuser, Basel, 2001.

[19] V. Klee, G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, 1972.

[20] J. Matoušek. Lower bounds for a subexponential optimization algorithm. *Random Structures & Algorithms*, 5(4):591–607, 1994.

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

[22] W. D. Morris. Randomized principal pivot algorithms for P-matrix linear complementarity problems. *Mathematical Programming*, Ser. A, 92:285–296, 2002.

[23] I. Schurr, T. Szabó. Finding the sink takes some time, *Discrete and Computational Geometry*, to appear. An extended abstract appeared in the proceedings of the *European Symposium on Algorithms (ESA)*, 2002.

[24] I. Schurr, T. Szabó. Lower bounds for some deterministic algorithms in the model of acyclic unique sink orientations, *preprint*, 2003.

[25] M. Sharir, E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos.*

Theoret. Aspects Comput. Sci., volume 577 of *Lecture Notes Comput. Sci.*, pages 569–579. Springer-Verlag, 1992.

[26] T. Szabó, E. Welzl. Unique sink orientations of cubes. In *Proc.* 42^{nd} *IEEE Symp. on Foundations of Comput. Sci.*, pages 547–555, 2001.

[27] K. Williamson Hoke. Completely unimodal numberings of a simple polytope. *Discrete Appl. Math.*, 20:69–81, 1988.