Random-Edge Is Slower Than Random-Facet on Abstract Cubes

Thomas Dueholm Hansen*[1] and Uri Zwick†[2]

1 **Department of Computer Science, Aarhus University, Aarhus, Denmark**
`tdh@cs.au.dk`

2 **Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel**
`zwick@tau.ac.il`

Abstract

Random-Edge and Random-Facet are two very natural randomized pivoting rules for the simplex algorithm. The behavior of Random-Facet is fairly well understood. It performs an expected *sub-exponential* number of pivoting steps on *any* linear program, or more generally, on any *Acyclic Unique Sink Orientation* (AUSO) of an arbitrary polytope, making it the fastest known pivoting rule for the simplex algorithm. The behavior of Random-Edge is much less understood. We show that in the AUSO setting, Random-Edge is *slower* than Random-Facet. To do that, we construct AUSOs of the n-dimensional hypercube on which Random-Edge performs an expected number of $2^{\Omega(\sqrt{n \log n})}$ steps. This improves on a $2^{\Omega(\sqrt[3]{n})}$ lower bound of Matoušek and Szabó. As Random-Facet performs an expected number of $2^{O(\sqrt{n})}$ steps on *any* n-dimensional AUSO, this established our result. Improving our $2^{\Omega(\sqrt{n \log n})}$ lower bound seems to require radically new techniques.

1998 ACM Subject Classification G.1.6 Optimization – Linear Programming, G.2.1 Combinatorics – Combinatorial Algorithms

Keywords and phrases Linear programming, the Simplex Algorithm, Pivoting rules, Acyclic Unique Sink Orientations

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.51

1 Introduction

Linear programming and the simplex algorithm. Linear programs (LPs) [3, 4, 28, 32] are among the most important mathematical optimization problems. The *simplex algorithm* (Dantzig [4]) is one of the most widely used methods for solving linear programs. It starts at a *vertex* of the polytope corresponding to the linear program. (We assume, for simplicity, that the LP is feasible, bounded, non-degenerate, that no two vertices have the same objective value, and that an initial vertex of the polytope is available.) If the current vertex is not optimal, then at least one of the *edges* incident to it leads to a neighboring vertex with a larger objective value. A *pivoting rule* determines which one of these vertices to move to. The simplex algorithm, with any pivoting rule, is guaranteed to find an optimal solution of an LP in a finite number of steps.

Unfortunately, with essentially all known *deterministic* pivoting rules, the simplex method requires an *exponential* number of steps on some LPs (see Klee and Minty [24] and [1, 2, 7,

* Thomas Dueholm Hansen was supported by the Carlsberg Foundation, grant no. CF14-0617.
† Uri Zwick was supported by BSF grant no. 2012338 and by the The Israeli Centers of Research Excellence (I-CORE) program, (Center No. 4/11).

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 51; pp. 51:1–51:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

15, 18]). While there are polynomial time algorithms for solving LP problems, most notably the *ellipsoid algorithm* (Khachian [23]) and *interior point methods* (Karmarkar [22]), these algorithms are not *strongly* polynomial, i.e., their running time, in the unit-cost model, depends on the number of *bits* in the representation of the coefficients of the LP, and not just on the combinatorial size of the problem, i.e., the number of variables and constraints. The question whether there exists a strongly polynomial time algorithm for solving linear programs is of great theoretical importance.

Randomized pivoting rules. Kalai [20] and Matoušek, Sharir and Welzl [29] devised a *randomized* pivoting rule, Random-Facet, that performs a *subexponential* number of pivoting steps, in expectation, on *any* LP. In particular, for an LP with n variables and $O(n)$ constraints, the expected number of pivoting steps performed by Random-Facet is at most $2^{O(\sqrt{n \log n})}$. A slightly improved version that performs an expected number of at most $2^{O(\sqrt{n})}$ pivoting steps on such LPs was recently obtained by the authors [17]. This improved version is currently the fastest known pivoting rule for the simplex algorithm.

Perhaps the most natural randomized pivoting rule is Random-Edge. When there are several improving edges from the current vertex, simply choose one of them uniformly at random. The best upper bound for Random-Edge on general LPs is exponential (Gärtner and Kaibel [13]). Whether this can be improved to a subexponential upper bound, similar to the one available for Random-Facet, is an intriguing open problem.

Friedmann et al. [8, 9, 11], building on results of Friedmann [6] and Fearnley [5], obtained a $2^{\tilde{\Omega}(n^{1/3})}$ lower bound on the expected number of pivoting steps performed by Random-Facet on LPs that correspond to *shortest paths* problems. [1] Friedmann et al. [9] also constructed LPs on which Random-Edge performs an expected number of $2^{\tilde{\Omega}(n^{1/4})}$ pivoting steps. We believe that this lower bound is not tight. However, as the $2^{\tilde{\Omega}(n^{1/4})}$ lower bound is already quite complicated, improving it seems to be a hard task. As a step in this direction we obtain a $2^{\Omega(\sqrt{n \log n})}$ lower bound for Random-Edge in the more general setting of *Acyclic Unique Sink Orientations* (AUSOs). In particular, this shows that in the AUSO setting, Random-Edge is slower than Random-Facet.

Acyclic Unique Sink Orientations (AUSOs). Each bounded LP has a polytope associated with it. The geometric vertices and edges of this polytope define a combinatorial graph. The objective function of the LP defines an *orientation* of the edges of this graph; An edge connecting vertices u and v is directed from u to v if and only if v has a better objective value. (We assume that no two vertices have the same objective value.) This orientation is clearly *acyclic*. It follows easily from the properties of LPs that every subgraph corresponding to a geometric *face* of the polytope has a *unique sink*, i.e., a unique vertex with no outgoing edges. The unique sink of the whole graph is then the optimal vertex of the LP.

This naturally motivates the definition of *Acyclic Unique Sink Orientations* (AUSOs). Let $G(P)$ be that graph that corresponds to a polytope P. An orientation of $G(P)$ is said to be an AUSO if and only if it is acyclic and any subgraph that corresponds to a face of P has a unique sink. The term AUSO was introduced by Szabó and Welzl [34]. The same notion was considered before, however, under several different names. Williamson Hoke [36] refers to them as *Completely Unimodal Numberings*, while Kalai [19, 20, 21] refers to them as *Abstract*

[1] In [9] we obtained a $2^{\tilde{\Omega}(\sqrt{n})}$ lower bound for a one-permutation variant of Random-Facet and erroneously claimed that the expected number of steps performed by this variant is equal to the expected number of steps performed by Random-Facet. Unfortunately, as we point out in [10], this is *not* the case.

Objective Functions. AUSOs provide an appealing abstraction of LPs. The subexponential upper bound on the behavior of Random-Facet only relies on the AUSOs properties and is thus valid also in the AUSO setting.

Of special interest are AUSOs of the n-dimensional *hypercube* whose properties we review in Section 2. In the sequel we restrict our attention to such AUSOs. Gärtner [12] proved a $2^{O(n^{1/2})}$ upper bound on the complexity of Random-Facet on AUSOs, and Matoušek [27] obtained an essentially tight $2^{\Omega(n^{1/2})}$ lower bound. (The improved version of Random-Facet of [17], which is not specialized to cubes, essentially matches Gärtner's bound. It is an interesting open problem whether a similar lower bound can be obtained for this algorithm.) Matoušek and Szabó [30, 31] obtained a lower bound of $2^{\Omega(n^{1/3})}$ on the complexity of Random-Edge on AUSOs. We improve their lower bound to $2^{\Omega(\sqrt{n \log n})}$, thus showing that Random-Edge is slower than Random-Facet on AUSOs. Hansen et al. [16] obtained a 1.8^n upper bound for Random-Edge on AUSOs. There is a large gap between the available upper and lower bounds. However, the $2^{\Omega(\sqrt{n \log n})}$ lower bound that we obtain is the best lower bound that can be obtained using current techniques. Improving it would require the use of *non-layered* AUSOs which are not yet known to exist.

Organization of paper. In the next section we review some basic properties of AUSOs of the Boolean hypercube. In Section 3 we describe a randomized product construction of AUSOs that plays a central role in the lower bound of Matoušek and Szabó [30, 31] as well as in our improved lower bound of $2^{\Omega(\sqrt{n})}$ which is described in Section 4. Section 5 contains an analysis of a random walk with *reshuffles* on a simple *path AUSO*. This analysis is needed to complete the proof of the lower bound presented in Section 4. In Section 6 we make the final push and improve the lower bound from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(\sqrt{n \log n})}$. In Section 7 we note that the AUSOs used to prove all known lower bounds for Random-Edge are *layered* and explain why it is unlikely that layered AUSOs could be used to obtain further improved lower bounds. A similar observation was made independently by Gärtner and Thomas [14]. We end in Section 8 with some concluding remarks and open problems.

2 Acyclic Unique Sink Orientations of the Hypercube

In this section we review the definition of AUSOs and describe some of their basic properties. We also introduce a simple *path* AUSO which plays a central role in our lower bound.

The n-cube is the undirected graph $C_n = (V_n, E_n)$, where $V_n = \{0,1\}^n$ and $E_n = \{\{\mathbf{x},\mathbf{y}\} \mid \mathbf{x},\mathbf{y} \in V_n, d_H(\mathbf{x},\mathbf{y}) = 1\}$, where $d_H(\mathbf{x},\mathbf{y})$ is the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$. Every string $\mathbf{s} \in \{0,1,*\}^n$ defines a *subcube* of C_n induced by the vertex set $V_s = \{\mathbf{x} \in \{0,1\}^n \mid \mathbf{x}_i = \mathbf{s}_i \text{ or } \mathbf{s}_i = *\}$. The subcube defined by $\mathbf{s}$ is clearly isomorphic to a cube whose dimension is the number of $*$'s in $\mathbf{s}$.

An *orientation* of the n-cube is a directed graph obtained by orienting each edge $\{\mathbf{x},\mathbf{y}\} \in E_n$ either from $\mathbf{x}$ to $\mathbf{y}$ or from $\mathbf{y}$ to $\mathbf{x}$. An orientation can be specified using a mapping $A : V_n \to \{0,1\}^n$ that satisfies the condition $A(\mathbf{x})_i \neq A(\mathbf{x} \oplus i)_i$, for every $\mathbf{x} \in V_n$. [2] The i-th edge of $\mathbf{x}$, i.e., $\{\mathbf{x}, \mathbf{x} \oplus i\}$ is directed *away* from $\mathbf{x}$ if and only if $A(\mathbf{x})_i = 1$. (Some authors refer to A as the *out-map* of the orientation. We consider it to be the orientation itself.) An orientation of a cube clearly induces orientations on all its subcubes. An orientation is

[2] Here, $A(\mathbf{x})_i$ denotes the i-th bit of $A(\mathbf{x}) \in \{0,1\}^n$, and $\mathbf{x} \oplus i$ is a shorthand for $\mathbf{x} \oplus \mathbf{e}_i$, where $\mathbf{e}_i = 0^{i-1}10^{n-i}$ is the i-th unit vector.

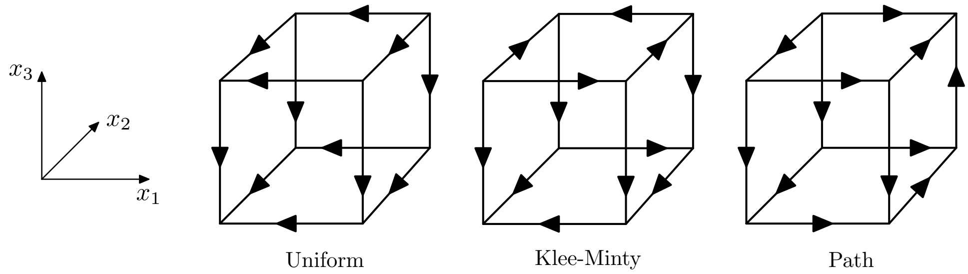

Figure 1 The uniform, Klee-Minty and the path AUSOs in dimension 3.

acyclic if there are no directed cycles. A *sink* of an orientation A of C_n is a vertex $\mathbf{x} \in V_n$ with no outgoing edges, i.e., $A(\mathbf{x}) = 0^n$. An acyclic orientation clearly has at least one sink, but it may have several.

▶ **Definition 1** (AUSOs). An orientation A of the n-cube is an n-AUSO if and only if it is *acyclic* and if every subcube has a *unique sink* under A.

Perhaps the simplest AUSO is the *uniform* AUSO defined as follows $U(\mathbf{x}) = \mathbf{x}$. The sink of this AUSO is clearly 0^n. A more interesting and famous AUSO is the Klee-Minty cube [24] defined as follows $KM(\mathbf{x})_i = x_i \oplus x_{i+1} \oplus \cdots x_n$, for $i = 1, 2, \ldots, n$, where $\mathbf{x} = x_1x_2 \ldots x_n$. The sink is again 0^n. (It is an instructive exercise to verify that the Klee-Minty cube is indeed an AUSO.) The Klee-Minty cube plays a prominent role in the lower bound of Matoušek and Szabó [30, 31]. One of the major steps in obtaining our improved lower bound is replacing the Klee-Minty cubes used in the lower bound of Matoušek and Szabó [30, 31] by the following much simpler AUSO.

▶ **Definition 2** (Path AUSO). The *path* n-AUSO is defined as follows

$$P(\mathbf{x})_i = x_i \oplus (x_1 \wedge x_2 \wedge \cdots \wedge x_{i-1}) \ .$$

In other words, if $\mathbf{x} = x_1x_2 \ldots x_n \in \{0, 1\}^n$, then the i-th edge of $\mathbf{x}$ is directed away from $\mathbf{x}$ if and only if $x_1x_2 \ldots x_i = 1^{i-1}0$ or $x_i = 1$ but $x_1x_2 \ldots x_{i-1} \neq 1^{i-1}$. (Informally, a bit wants to become, or remain, a 1 if and only if all the bits preceding it are 1.) The sink of the path AUSO is clearly at 1^n. The uniform, Klee-Minty and the path AUSO for $n = 3$ are shown in Figure 1. The path AUSO is also used in a building block in [9, 11]. A realization of the path AUSO as a shortest paths problem in a weighted directed graph is given in Figure 2. Each n-bit string $\mathbf{x} = x_1x_2 \ldots x_n$ corresponds to a *tree* of directed paths directed towards the target vertex 0. If $x_i = 0$, then $(i, 0)$ is included in the tree, otherwise $(i, i-1)$ is included in the tree. The i-th edge of $\mathbf{x}$ in the n-cube is directed away from $\mathbf{x}$ if and only if switching the outgoing edge of i in the tree reduces the distance to the target of at least one vertex. With this interpretation, it is clear that the path AUSO is an AUSO. (It is of course also easy to prove it directly.)

▶ **Lemma 3.** *The path AUSO is indeed an AUSO.*

Schurr and Szabó [33] describe two useful methods of constructing new AUSOs from existing ones. The first method takes an n-AUSO and *expands* each one of its 2^n vertices into an m-AUSO, resulting in an $(n + m)$-AUSO. (Each one of the 2^n vertices may be expanded into a different m-AUSO.) Schurr and Szabó [33] call it the *blowup* construction. We view it as a *product* of AUSOs.

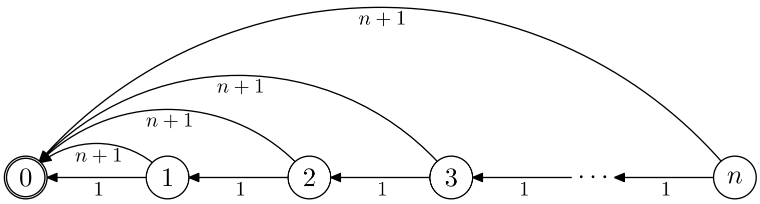

Figure 2 A realization of the path AUSO as a shortest paths problem in a weighted directed graph. Numbers attached to edges are edge weights.

▶ **Definition 4** (Product of AUSOs [33]). Let A be an n-AUSO, and let $m > 1$. For every $\mathbf{x} \in \{0,1\}^n$, let $B_{\mathbf{x}}$ be an m-AUSO. Define an $(n+m)$-AUSO $C = A \times \langle B_{\mathbf{x}} \rangle$ as follows:

$$C(\mathbf{x}, \mathbf{y}) \;=\; (A(\mathbf{x}), B_{\mathbf{x}}(\mathbf{y}))\ , \quad \text{for } \mathbf{x} \in \{0,1\}^n, \mathbf{y} \in \{0,1\}^m\ .$$

The proof that the product AUSO is indeed an AUSO is straightforward. (For the details, see [33].) Before describing the second method we need another definition.

▶ **Definition 5** (Hypersink [33]). A subcube of an AUSO is a *hypersink* if and only if all edges between vertices of the subcube and vertices outside the subcube are directed towards the subcube.

▶ **Lemma 6** (Hypersink replacement [33]). *Let A be an AUSO, let B be a subcube of A which is a hypersink, and let B' be an AUSO of the same dimension as B. Then, the AUSO A' obtained by modifying the orientation of the edges within B so that the resulting orientation is isomorphic to B' is again an AUSO.*

The proof of the lemma is again straightforward and can be found in [33]. Finally, the following well-known lemma will be used in Section 6.

▶ **Lemma 7** (Unique outmaps [34]). *Let A be an n-AUSO. Then, for every $\mathbf{x}_1 \neq \mathbf{x}_2 \in \{0,1\}^n$ we have $A(\mathbf{x}_1) \neq A(\mathbf{x}_2)$. In particular, the number of vertices of A of out-degree k is $\binom{n}{k}$.*

3 Random-Edge on Randomized Products

Matoušek and Szabó [30, 31] introduced the following simple and elegant *randomized product* construction. Let A be an n-AUSO and let B be an m-AUSO. Assume, for simplicity, that the sinks of A and B are at 1^n and 1^m, respectively. For every $\mathbf{x} \in \{0,1\}^n$, let $B_{\mathbf{x}}$ be a version of B in which the m coordinates are *randomly permuted*, each permutation being equally likely. (Note that the sink of $B_{\mathbf{x}}$ is still at 1^m.) Consider now the product $A \times \langle B_{\mathbf{x}} \rangle$. It is easy to check that the copy of A corresponding to $\mathbf{y} = 1^m$ is a hypersink of $A \times \langle B_{\mathbf{x}} \rangle$. We replace this copy of A by a *random translation* A' of A. A random translation of an n-AUSO A is obtained by choosing a random vertex $\mathbf{x}_0 \in \{0,1\}^n$ and making it the sink of an AUSO A' isomorphic to A. More precisely, we let $A'(\mathbf{x}) = A(\mathbf{x} \oplus \mathbf{x}_0)$. A formal definition of randomized products follows.

▶ **Definition 8** (Randomized product [30, 31]). Let A be an n-AUSO and let B be an m-AUSO whose sinks are at 1^n and 1^m, respectively. For every $\mathbf{x} \in \{0,1\}^n$, let $B_{\mathbf{x}}$ be a randomly permuted version of B, and let A' be a random translation of A. Then, the *randomized product* $C = A \times_R B$ is a distribution over AUSOs defined as follows:

$$C(\mathbf{x}, \mathbf{y}) \;=\; \begin{cases} (A(\mathbf{x}), B_{\mathbf{x}}(\mathbf{y})) & , \text{ if } \mathbf{y} \neq 1^m\ , \\ (A'(\mathbf{x}), B_{\mathbf{x}}(\mathbf{y})) & , \text{ otherwise}\ . \end{cases}$$

As $C = A \times_R B$ is obtained by performing a product of AUSOs followed by a hypersink replacement, it is clearly an AUSO, or rather a probability distribution over AUSOs.

Consider now the behavior of Random-Edge on $C = A \times_R B$, starting from some vertex $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \{0,1\}^n$, $\mathbf{y} \in \{0,1\}^m$. Assume at first that $\mathbf{x} \neq 1^n$ and $\mathbf{y} \neq 1^m$. Each step of Random-Edge either changes a coordinate of $\mathbf{x}$ or of $\mathbf{y}$. If we only consider the steps in which $\mathbf{x}$ changes, and look only at $\mathbf{x}$, then as long as $\mathbf{y} \neq 1^m$, we get a standard Random-Edge walk on A.

The random walk on $\mathbf{y}$ induced by the Random-Edge walk on $C = A \times_R B$ is more interesting. When $\mathbf{x}$ is fixed, $\mathbf{y}$ performs a standard Random-Edge walk on $B_{\mathbf{x}}$. But, when a step from $\mathbf{x}$ to $\mathbf{x}'$ in A is made, the walk on $\mathbf{y}$ finds itself in a new AUSO $B_{\mathbf{x}'}$. Due to the acyclicity of A, $B_{\mathbf{x}'}$ was never visited before, and is thus completely random. The resulting random walk on $\mathbf{y}$ is what Matoušek and Szabó [30, 31] call a *random walk with reshuffles* on B. Such a random walk is likely to be much longer than a standard Random-Edge walk on B as the random reshuffles tend to destroy progress made since the last reshuffle.

When the random walk with reshuffles on B reaches its sink $\mathbf{y} = 1^m$, the hypersink A' is entered, and $\mathbf{y}$ remains fixed. Note that $\mathbf{y} = 1^m$ may be reached either before $\mathbf{x}$ reaches 1^n, or after it does. We prefer the second option, as then the Random-Edge walk on A ran to completion. Our construction will ensure that this happens with very high probability. As A' is a random translation of A, the ensuing random walk on A', even if it starts at $\mathbf{x} = 1^n$, is equivalent to a Random-Edge walk starting at a *random* vertex of A. Thus, under appropriate conditions, a Random-Edge walk on $C = A \times_R B$ is expected to be at least *twice* as long as a Random-Edge walk on A itself. This is the crux of the matter.

We next consider the probability of a reshuffle in a random walk with reshuffles on B. This probability must be high to make the random walk long. If the current vertex in $C = A \times_R B$ is $(\mathbf{x}, \mathbf{y})$ and there are s and t outgoing edges from $\mathbf{x}$ in A and $\mathbf{y}$ in B, respectively, then a reshuffle occurs with probability $s/(s+t)$, as Random-Edge on $C = A \times_R B$ chooses each outgoing edge with equal probability and each step in A causes a reshuffle of B. We let Random-Reshuffle$_k$ denote a random walk with reshuffles on B induced by a Random-Edge walk on $C = A \times_R B$, conditioned on all visited vertices of A having outdegree at least k. The probability of a reshuffle when $\mathbf{y}$ has outdegree t in B is then *at least* $k/(k+t)$.

To ensure that reshuffles occur with sufficiently high probability, we need a slight generalization of the randomized product construction.

▶ **Definition 9** (k-fold randomized product [30, 31])**.** Let A be an n-AUSO and let B be an m-AUSO whose sinks are at 1^n and 1^m, respectively, and let $k \geq 1$. For every $\mathbf{x} \in \{0,1\}^n$ and $1 \leq i \leq k$, let $B_{\mathbf{x},i}$ be a randomly permuted version of B, and let A' be a random translation of A. Then, the *k-fold randomized product* $C = A \times_R^k B$ is a distribution over AUSOs defined as follows:

$$C(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_k) \;=\; \begin{cases} (A(\mathbf{x}), B_{\mathbf{x},1}(\mathbf{y}_1), \ldots, B_{\mathbf{x},k}(\mathbf{y}_k)) & \text{, if } \mathbf{y}_1 \neq 1^m \wedge \cdots \wedge \mathbf{y}_k \neq 1^m \;, \\ (A'(\mathbf{x}), B_{\mathbf{x},1}(\mathbf{y}_1), \ldots, B_{\mathbf{x},k}(\mathbf{y}_k)) & \text{, otherwise} \;. \end{cases}$$

It is not difficult to check that for every random choice $C = A \times_R^k B$ is indeed an AUSO. Matoušek and Szabó [30, 31] need to apply the k-fold randomized product with $k = n^{1/3}$. We only need $k = 2$. As a result, the lower bound is improved from $2^{\Omega(\sqrt[3]{n})}$ to $2^{\Omega(\sqrt{n})}$. Using $k = 1$ is actually enough, but the probabilistic analysis is slightly more complicated. The details will appear in the full version of the paper.

▶ **Lemma 10** ([30, 31])**.** *Let A be an n-AUSO, let B be an m-AUSO and let $k > 1$. Suppose that the probability that the* Random-Edge *walk on A, starting at a random vertex, reaches* *a vertex with outdegree less than k in less than T steps is at most p. Suppose that the probability that a* RANDOM-RESHUFFLE$_k$ *walk on B, starting at a random vertex, reaches the sink in less than T steps is at most q. Then, the probability that the* RANDOM-EDGE *walk on $C = A \times_R^k B$, starting at a random vertex, reaches a vertex of outdegree less than k in less than $2T$ steps is at most $2p + kq$.*

Proof. Partition the RANDOM-EDGE walk on $C = A \times_R^k B$, starting at a random vertex, into two stages. The first stage ends when one of $B_1, \ldots, B_k$ reaches its sink, i.e., when a state of the form $(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_k)$, where $\mathbf{y}_i = 1^m$, for some $1 \le i \le k$, is reached. When the first stage ends, the orientation on the first n-coordinates changes to A', a random translation of A, and the second stage begins. The second stage lasts until the sink of $C = A \times_R^k B$ is reached.

The probability that the first stage lasts less than T steps is at most $p + kq$. Indeed, during the first T steps on C, at most T steps on A are performed. Thus, the probability that the RANDOM-EDGE walk on A reaches a vertex of outdegree less than k is at most p. As long as the walk on A visits vertices of outdegree at least k, the induced random walks on $B_1, \ldots, B_k$ are RANDOM-RESHUFFLE$_k$ walks. Thus, the probability that a specific B_i reaches its sink during the first T steps on C is at most q, and the claim follows. During the first phase, there is at least one improving switch in every B_i, thus the outgedree of each visited vertex is at least k.

In the second stage, the orientation on the first n-coordinates changes to A'. As A' is a random translation of A, this is equivalent to starting a RANDOM-EDGE walk on A at a random starting vertex. The probability that this random walk reaches a vertex of outdegree less than k in less than T is steps is at most p.

Thus, the probability that the RANDOM-EDGE walk on $C = A \times_R^k B$, starting at a random vertex, reaches a vertex of outdegree less than k in less than $2T$ steps is at most $(p + kq) + p = 2p + kq$. ◀

4 The Lower Bound

A distribution of AUSOs on which RANDOM-EDGE makes $2^{\Omega(\sqrt{n})}$ steps, with high probability, is obtained by iterating the randomized product construction of the previous section. We follow the footsteps of Matoušek and Szabó [30, 31] making one crucial change; We use the *path* AUSO, defined in Section 2, instead of the Klee-Minty cube, as the main building block. Using the path AUSO allows us to simplify the proof and improve the lower bound. The following lemma, whose proof is given in the next section, is one of the main technical contributions of this paper.

▶ **Lemma 11.** *Let P_m be the path m-AUSO. There are constants $\alpha, \beta > 0$ such that the probability that* RANDOM-RESHUFFLE$_2$ *on P_m, starting from a random vertex, performs less than $2^{\alpha m}$ steps before reaching the sink is at most $2^{-\beta m}$.*

Let m be an integer, and let $\ell = \gamma m$, where $\gamma < \min\{\alpha, \beta\}$, where α, β are the constants in Lemma 11. Let A_0 be an arbitrary m-AUSO. For concreteness, we let $A_0 = P_m$, the *path* AUSO of Section 2. We construct a sequence of $A_0, A_1, \ldots, A_\ell$ of AUSOs, where $A_i = A_{i-1} \times_R^2 P_m$ is an $(2i+1)m$-AUSO, for $0 \le i \le \ell$. Note that A_ℓ is of dimension $(2\ell+1)m = O(m^2)$. The following lemma, which easily implies a $2^{\Omega(\sqrt{n})}$ lower bound, claims that with high probability, RANDOM-EDGE performs at least $2^{\Omega(m)}$ steps when started at a random vertex of A_ℓ. We give an improved version of the lemma in Section 6.

▶ **Lemma 12.** *The probability that* RANDOM-EDGE *performs less than 2^ℓ steps when started at a random vertex of A_ℓ, where $\ell < \alpha m$, is at most $4 \cdot 2^{\ell - \beta m}$.*

Proof. Let p_i be the probability that RANDOM-EDGE, started at a random vertex of A_i, performs less than 2^i steps before reaching a vertex of outdegree less than 2. Note that $p_0 = (m+1)2^{-m}$ is just the probability that the random starting vertex of A_0 has outdegree at most 1, which by Lemma 7 is exactly $(m+1)2^{-m}$.

Let q_i be the probability that the RANDOM-RESHUFFLE$_2$ walk on P_m, in which the probability of reshuffle from a vertex of outdegree j is at least $2/(j+2)$, reaches the sink in less than 2^i steps, when started at a random vertex. Clearly, for every $1 \le i \le \ell$ we have $q_i \le q_\ell$. As $\ell \le \alpha m$, Lemma 11 implies that $q_\ell \le 2^{-\beta m}$.

By Lemma 10, we have $p_i \le 2(p_{i-1} + q_{i-1})$. It follows easily by induction that $p_i \le 2^i p_0 + 2(2^i - 1)q_i$, for $1 \le i \le \ell$. Thus, $p_\ell \le 2^\ell(m+1)2^{-m} + 2^{\ell+1}2^{-\beta m} \le 4 \cdot 2^{\ell-\beta m}$. ◀

As a corollary, we get:

▶ **Theorem 13.** *There exist n-AUSOs and appropriate starting points from which* RANDOM-EDGE *performs $2^{\Omega(n^{1/2})}$ steps with probability at least $1 - 2^{-\delta\sqrt{n}}$, for some $\delta > 0$.*

5 Random Walk with Reshuffles on the Path AUSO

In this section we provide a proof of Lemma 11, completing the proof of our first lower bound.

Let $\mathbf{y} \in \{0,1\}^m$ be a state of RANDOM-RESHUFFLE$_2$ on the path AUSO P_m. Let k be the *weight*, i.e., the number of 1's in $\mathbf{y}$ and let i be the number of leading 1's in $\mathbf{y}$. For example, if $m = 7$ and $\mathbf{y} = 1100101$, then $k = 4$ and $i = 2$. Also let $j = k - i$, the number of non-leading 1's. We say that $\mathbf{y}$ is of *type* (i,j). The outdegree of a state of type (i,j) is clearly $j+1$.

From a state $\mathbf{y}$ of type (i,j) of RANDOM-RESHUFFLE$_2$ on P_m, there is a reshuffle with a probability of at *least* $2/(j+3)$, as all states of type (i,j) are of outdegree $j+1$. Several reshuffles may occur in a row, but they have exactly the same effect as a single reshuffle.

If a state of weight k is reshuffled, the obtained state is of type $(k-j,j)$, where $0 \le j \le k$, with probability $a_{k,j} = \binom{m-(k-j+1)}{j} / \binom{m}{k}$. (Among the $\binom{m}{k}$ binary strings of length m and weight k, there are exactly $\binom{m-(k-j+1)}{j}$ strings that start with $1^{k-j}0$.) Clearly $\sum_{j=0}^{k} a_{k,j} = 1$.

When a reshuffle occurs, we initially consider only the type of the state obtained. We *delay* the decision as to which state exactly we are in. (All states belonging to the type are equally likely.) This simplifies the analysis, as we only need to consider $O(m^2)$ types ($\binom{m+1}{2} + 1$ to be exact) rather than 2^m possible states.

Suppose that we are now in a state of type (i,j), where $k = i + j$. One or more reshuffles are performed with a probability of at least $2/(j+3)$, and then a standard move is performed. Let (i',j'), where $k' = i' + j'$, be the type of the new state obtained. Note that $k' = k-1$ or $k' = k+1$, as a reshuffle does not change the weight and a single move either increases or decreases the weight by 1. We show that the probability of a weight increase, i.e., $k' = k+1$ is bounded by some constant $c < \frac{1}{2}$. As the weight of a random starting vertex is close to $m/2$ and as the weight of the sink is m, it would follow that an exponential number of steps are needed, with high probability, to reach the sink.

Let $\bar{p}_k$ be the probability of a weight increase from a state of weight k given that a reshuffle occurs. We have $\bar{p}_k = \sum_{j=0}^{k} \frac{a_{k,j}}{j+1}$, as the reshuffle generates a state of type $(k-j,j)$ with probability $a_{k,j}$ and the probability of a weight increase from such a state is $1/(j+1)$.

Let $p_{k,j}(s)$ be the probability of a weight increase from a state of type $(k-j,j)$, when the corresponding state in A has s outgoing edges. The probability of a reshuffle from a state of type $(k-j,j)$ is $\frac{s}{j+s+1}$, and the probability of no reshuffle is $\frac{j+1}{j+s+1}$. If there is no reshuffle, the probability of a weight increase is $\frac{1}{j+1}$. If there is a reshuffle, then the probability of a weight increase is $\bar{p}_k$. Hence,

$$p_{k,j}(s) \;=\; \frac{j+1}{j+s+1}\cdot\frac{1}{j+1}+\frac{s}{j+s+1}\cdot\bar{p}_k \;=\; \frac{1}{j+s+1}+\frac{s}{j+s+1}\cdot\bar{p}_k\;.$$

We next upper bound $\bar{p}_k$, the probability of a weight increase following a reshuffle from a state of weight k.

▶ **Lemma 14.** $\bar{p}_k \;<\; \dfrac{m-k}{(k+1)(m-k-1)}$.

Proof. A simple manipulation of binomial coefficients yields:

$$\frac{a_{k,j}}{j+1} \;=\; \frac{\binom{m-k+j-1}{j}}{\binom{m}{k}}\,\frac{1}{j+1} \;=\; \frac{\binom{m-k+j-1}{j+1}}{\binom{m}{k}}\,\frac{1}{m-k-1} \;=\; \frac{1}{m-k-1}\,\frac{\binom{m}{k+1}}{\binom{m}{k}}\,a_{k+1,j+1}\;.$$

Hence,

$$\bar{p}_k=\sum_{j=0}^{k}\frac{a_{k,j}}{j+1}=\frac{1}{m-k-1}\,\frac{\binom{m}{k+1}}{\binom{m}{k}}\sum_{j=0}^{k}a_{k+1,j+1}<\frac{1}{m-k-1}\,\frac{\binom{m}{k+1}}{\binom{m}{k}}=\frac{m-k}{(k+1)(m-k-1)}.$$

◀

▶ **Lemma 15.** *Let (i,j) be a type of* Random-Reshuffle$_2$ *walk on the path AUSO P_m of weight $k=i+j$ satisfying $8\le k\le m-9$. Then, the probability that the type obtained after one step of a* Random-Reshuffle$_2$ *walk, i.e., a reshuffle with a probability of at least $2/(j+3)$ followed by a standard move, is of weight $k+1$ is at most $\frac{5}{12}$.*

Proof. We need to show that $p_{k,j}(s)\le\frac{5}{12}$, for every $8\le k\le m-9$, $0\le j\le k$, and $s\ge 2$. Recall that $p_{k,j}(s)=\frac{j+1}{j+s+1}\cdot\frac{1}{j+1}+\frac{s}{j+s+1}\cdot\bar{p}_k$. For $8\le k\le m-9$ we have $\bar{p}_k\le\frac{1}{8}$, as $\frac{1}{k+1}\le\frac{1}{9}$ while $\frac{m-k}{m-k-1}\le\frac{9}{8}$. If $j\ge 2$, then $p_{k,j}(s)\le\frac{1}{3}$, for every $s\ge 1$, as $\frac{1}{j+1}\le\frac{1}{3}$ and $\bar{p}_k\le\frac{1}{3}$. If $j=1$, then $p_{k,1}(s)\le\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{8}=\frac{5}{16}$, again for every $s\ge 1$. Finally, if $j=0$, then $p_{k,0}(s)\le\frac{1}{3}+\frac{2}{3}\cdot\frac{1}{8}=\frac{5}{12}$, for every $s\ge 2$. ◀

Lemma 15 does not hold for Random-Reshuffle$_1$ as for $j=0$ we relied on the assumption $s\ge 2$. This is why we need to use 2-fold randomized products. Alternatively, we can look at *two* steps of Random-Reshuffle$_1$ on P_m. This yields a negative drift from all types, including $(i,0)$, but the proof is slightly more complicated. The details will appear in the full version of the paper.

▶ **Definition 16** (Biased random walk)**.** Let $0\le c<\frac{1}{2}$. A random process on states $\{0,1,\ldots,n\}$ is *c-bounded* if and only if whenever the process is in state i, where $i>0$, then it moves to state $i+1$ with probability at most c, and to state $i-1$ with the complementary probability of at least $1-c$. From state 0, the process moves to state 1 with probability 1.

In the above definition, the transition probabilities may depend on the history of the random process, as well as on external factors. The following claim is well known. For completeness, we include a proof that follows the presentation in Levin, Peres and Wilmer [25] (Section 17.3.1).

▶ **Lemma 17.** *Let $0\le c<\frac{1}{2}$. The probability that a c-bounded random process on $\{0,1,\ldots,n\}$ that starts at state 0 reaches state n in less than K steps is at most $K\big(\frac{c}{1-c}\big)^{n-1}$. In particular, for any $\alpha<\frac{1-c}{c}$, the probability that the random walk reaches n in less than α^{n-1} steps is at most $\big(\frac{\alpha c}{1-c}\big)^{n-1}$.*

Proof. Let us assume that the probability of moving from i to $i+1$ is exactly c. It is not difficult to show that this implies the more general claim.

Let X_t be a c-bounded random walk on the integers starting at 1, i.e., $X_0 = 1$. Let $r = (1-c)/c$. It is easy to check that r^{X_t} is a *martingale*. Consider the *stopping time* τ defined as the smallest t for which $X_t = 0$ or $X_t = n$. By the *Optional Stopping Theorem* we get that $\mathbb{E}[r^{X_\tau}] = \mathbb{E}[r^{X_0}] = r$. Let a be the probability that the random walk reaches n before it reaches 0. Then $\mathbb{E}[r^{X_\tau}] = ar^n + (1-a)$. Thus $ar^n + (1-a) = r$ and $a = \frac{r-1}{r^n-1} < (1/r)^{n-1}$.

Returning to the c-bounded walk on $\{0, 1, \ldots, n\}$ starting at 0, we note that the expected number of times that this random walk returns to 0 before visiting n is a geometric random variable with parameter a. Thus the probability that the random walk reaches n in less than K steps, and in particular less than K returns to 0, is at most $1-(1-a)^K < Ka < K\left(\frac{c}{1-c}\right)^{n-1}$. ◀

Putting everything together, we get a proof of Lemma 11.

Proof of Lemma 11. A Random-Reshuffle$_2$ walk on the path AUSO P_m induces a random walk on *types* (i,j). The random walk on types induces a random walk on *weights*. Lemma 15 claims that when $8 \le k \le m-9$, the random walk on the weights is $\frac{5}{12}$-bounded. The weight of a random starting point of Random-Reshuffle$_2$ is binomially distributed. By Chernoff bound, the probability that the starting state of the bounded walk on the weights starts at a weight larger than $(\frac{1}{2}+\delta)m$, for some $\delta > 0$, is exponentially small. By Lemma 17 the probability that the random walk moves from the random starting point to $m-9$ in at most an exponential number of steps is exponentially small. The existence of appropriate constants α, β follows easily. ◀

6 An Improved Lower Bound

The improved lower bound of $2^{\Omega(\sqrt{n \log n})}$ is obtained using essentially the same construction but with a strengthened analysis. We again let $A_0 = P_m$ and $A_i = A_{i-1} \times_R^2 P_m$, for $i = 1, \ldots, \ell$, but this time we choose $\ell = \gamma m \log m$, for some $\gamma > 0$. We can still show that with high probability Random-Edge performs at least 2^ℓ steps on A_ℓ. Thus A_ℓ is an n-AUSO, where $n = 2\gamma m^2 \log m$, on which Random-Edge performs with high probability at least 2^ℓ steps, where $\ell = \gamma m \log m = \Omega(\sqrt{n \log n})$. This establishes the following theorem, which is the main result of this paper.

▶ **Theorem 18.** *There exist n-AUSOs and appropriate starting points from which* Random-Edge *performs $2^{\Omega(\sqrt{n \log n})}$ steps with probability at least $1 - 2^{-\delta\sqrt{n}}$, for some $\delta > 0$.*

To prove this result, we need a strengthened version Lemma 11. Lemma 11 states that there are constants $\alpha, \beta > 0$ such that the probability that Random-Reshuffle$_2$ on P_m, starting from a random vertex, performs less than $2^{\alpha m}$ steps before reaching the sink is at most $2^{-\beta m}$. The proof of Lemma 11 relied on the fact that from a state of type (i,j), the probability of a reshuffle is at least $2/(j+3)$. If the reshuffle probabilities are always $2/(j+3)$, then Lemma 11 is essentially tight. However, the reshuffle probability is $2/(j+3)$ only if the vertex $\mathbf{x} \in A$ has only two outgoing edges. By Lemma 7 this can happen at most $\binom{n}{2}$ times. More generally, let $s = \sqrt{n}$. The number of vertices in A of outdegree less than s is at most $\sum_{i=0}^{s} \binom{n}{i} \le n^s$.

▶ **Definition 19** (Random-Reshuffle$_k^{(N,s)}$)**.** A Random-Reshuffle$_k^{(N,s)}$ walk on an m-AUSO B is a random walk with reshuffles on B in which the reshuffle probability from a vertex $\mathbf{y} \in \{0,1\}^m$ of outdegree t is at least $s/(s+t)$, except for at most N times in which the reshuffle probability is only guaranteed to be at least $k/(k+t)$.

By the above discussion, the random process on B induced by a Random-Edge walk on $A \times_R^2 B$ is a Random-Reshuffle$_k^{(N,s)}$ process, with $N \leq n^s$. The following lemma is a strengthening of Lemma 11. The proof will appear in the full version of the paper.

▶ **Lemma 20.** *Let P_m be the path m-AUSO. There are constants $\alpha, \beta > 0$ such that the probability that* Random-Reshuffle$_2^{(N,s)}$ *on P_m, where $s = m^{1/2}$, $N \leq m^{3s}$, starting from a random vertex, performs less than $2^{\alpha m \log m}$ steps before reaching the sink is at most $2^{-\beta m}$.*

We note that Lemma 20 is essentially best possible, which is one of the reasons we believe that improving our $2^{\Omega(\sqrt{n \log n})}$ lower bound will require new techniques. (See also Section 7.)

▶ **Lemma 21.** *Let B be any m-AUSO. Then, with high probability* Random-Reshuffle$_k$ *on B makes at most $O((km)^m) = 2^{O(m \log(km))}$ steps before reaching its sink.*

Proof. Each vertex of B is of distance at most m from the sink (see, e.g., [26, 16]). Thus, if there are m consecutive steps with no reshuffles and the right edge in B is followed, then the sink is reached. This happens with a probability of at least $((k+1)m)^{-m}$. ◀

The proof of Lemma 20 relies on the following strengthening of Lemma 17. The proof will appear in the full version of the paper.

▶ **Lemma 22.** *Let $0 \leq c_1 < c_2 < \frac{1}{2}$. Consider a random walk on $\{0,1,\ldots,n\}$ in which in each step a controller is allowed to flip a coin with success probability at most c_1, or a coin with success probability at most c_2. The controller is allowed to make the second choice at most N times. According to the outcome of the coin, the process moves from i to either $i-1$ or $i+1$. (From 0 the move is always to 1.) Then, the probability that the walk reaches n in less than K steps is at most $K\left(\frac{c_1}{1-c_1}\right)^{n/2-1} + N\left(\frac{c_2}{1-c_2}\right)^{n/2-1}$.*

Relying on these strengthened lemmas, Theorem 18 follows using essentially the same arguments used to prove Theorem 13.

7 Decomposable and Layered AUSOs

The AUSOs used above to obtain the $2^{\Omega(\sqrt{n \log n})}$ lower bound, as well as the AUSOs used by Matoušek and Szabó [30, 31] and by Friedmann et al. [9] are *decomposable*.

▶ **Definition 23** ((k,ℓ)-decomposable AUSO)**.** A $(k\ell)$-AUSO is said to be (k,ℓ)-decomposable if its coordinates can be partitioned into k blocks $B_1,\ldots,B_k$ each of size ℓ such that if $\mathbf{x}$ is the sink of A and $\mathbf{y}$ agrees with $\mathbf{x}$ in $B_i, B_{i+1},\ldots,B_k$, for some $1 \leq i \leq k$, then all the edges in these coordinates are directed towards $\mathbf{y}$.

Decomposable AUSOs belong to a much wider class of *layered AUSOs*. A layered AUSO is an AUSO whose vertices can be partitioned into a relatively small number of layers such that from each non-sink vertex there is a relatively short directed path to a vertex of a lower layer, and such that no directed path may lead from a vertex to a vertex of a higher layer. More formally:

▶ **Definition 24** ((k,ℓ)-Layered AUSOs)**.** An n-AUSO A is said to be (k,ℓ)-*layered* if its vertices can be partitioned into disjoint *layers* $L_0, L_1, \ldots, L_k$ such that: *(i)* L_0 only contains the sink. *(ii)* If $\mathbf{x} \in L_i$, where $i > 0$, then there is a directed path of length at most ℓ in A from $\mathbf{x}$ to some vertex $\mathbf{y} \in L_j$, for $j < i$. *(iii)* If $\mathbf{x} \in L_i$, then there is no directed path in A from $\mathbf{x}$ to any vertex $\mathbf{y} \in L_j$, for $j > i$.

▶ **Lemma 25.** *If A is a (k,ℓ)-decomposable AUSO then it is also (k,ℓ)-layered.*

Proof. Let $B_1, B_2, \ldots, B_k$ be the partition of the coordinates of A and let $\mathbf{x}$ be the sink of A. For $i = 0, 1, \ldots k$, let L_i be the set of vertices that agree with $\mathbf{x}$ in all the coordinates of $B_{i+1}, B_{i+2}, \ldots, B_k$ and differ from $\mathbf{x}$ in some coordinate of B_i. It is easy to check that all conditions are met by using the fact that every vertex in a subcube of dimension ℓ has a path of length at most ℓ to the sink of that subcube (see, e.g., [26, 16]). ◀

We next show that Random-Edge is fairly quick on layered AUSOs.

▶ **Lemma 26.** *The expected number of steps that* Random-Edge *performs on a (k,ℓ)-layered n-AUSO, from any starting vertex, is at most $k\ell n^\ell$.*

Proof. We show that the expected number of steps per layer is at most ℓn^ℓ, which proves the lemma. From every vertex in the current layer there is at least one path of length ℓ to a lower layer. Random-Edge follows this path with a probability of at least $1/n^\ell$. Therefore the expected number of trials before successfully following such a path is at most n^ℓ, and since each trial uses at most ℓ steps, the total expected number of steps is at most ℓn^ℓ. ◀

We note that the n-AUSOs used to obtain our $2^{\Omega(\sqrt{n \log n})}$ lower bound are (k,ℓ)-layered for $k = O(\sqrt{n \log n})$ and $\ell = O(\sqrt{n/\log n})$. Lemma 26 therefore shows that the expected number of steps performed by Random-Edge on our AUSOs is at most $2^{O(\sqrt{n \log n})}$, which means that our analysis of these AUSOs is tight up to a constant factor in the exponent. Moreover, to improve the lower bound it is necessary to construct n-AUSOs that are not $(2^{O(\sqrt{n \log n})}, O(\sqrt{n/\log n}))$-layered, or, more generally, not (k,ℓ)-layered for any choice of k and ℓ such that $k\ell n^\ell = 2^{O(\sqrt{n \log n})}$. No such AUSOs are currently known to exist. Similarly, to improve on the 1.8^n upper bound of Hansen et al. [16] it is enough to prove that every n-AUSO is $(2^{cn}, dn/\log n)$-layered, for $c + d < 0.847 < \log_2 1.8$.

The lower bound of Friedmann et al. [9] uses AUSOs that simulate a binary counter. A t-bit counter is simulated by $(t, n/t)$-decomposable n-AUSO. The lower bound obtained using such AUSOs is roughly 2^t. As $(t, n/t)$-decomposable n-AUSO are also $(t, n/t)$-layered, we get using Lemma 26 that the best lower bound that can be obtained using such AUSOs is $\min\{2^t, n^{n/t+1}\}$, which attains a maximum value of about $2^{\Theta(\sqrt{n \log n})}$ when $t = \sqrt{n \log n}$.

A notion of the *niceness* of AUSOs and USOs, which is related to the notions of decomposable and layered AUSOs was defined by Welzl [35]. Gärtner and Thomas [14] have independently observed that the AUSOs used to obtain our lower bound are $\sqrt{n \log n}$-nice and therefore cannot be used to obtain further improved lower bounds.

8 Concluding remarks and open problems

We proved a $2^{\Omega(\sqrt{n \log n})}$ lower bound on the number of steps performed by Random-Edge on some AUSOs. The two obvious open problems are proving a $2^{o(n)}$ upper bound and an improved lower bound of $2^{\omega(\sqrt{n \log n})}$. We believe that radically new techniques would be needed to prove a lower bound of $2^{\omega(\sqrt{n \log n})}$. Another interesting open problem is improving the $2^{\Omega(n^{1/4})}$ lower bound of [9] for AUSOs that correspond to actual linear programs.

Acknowledgments. We would like to thank Mikkel Thorup for encouraging us to improve our lower bound from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(\sqrt{n \log n})}$, to Yuval Peres for pointing us to [25] for a proof of Lemma 17, and to Bernd Gärtner and Antonis Thomas for bringing the concept of niceness of AUSOs to our attention.

References

1 N. Amenta and G.M. Ziegler. Deformed products and maximal shadows of polytopes. In *Advances in Discrete and Computational Geometry*, pages 57–90, Providence, 1996. Amer. Math. Soc. Contemporary Mathematics 223.

2 D. Avis and V. Chvátal. Notes on Bland's pivoting rule. In *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 24–34. Springer, 1978.

3 V. Chvátal. *Linear programming*. W. H. Freeman and Company, 1983.

4 G.B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963.

5 J. Fearnley. Exponential lower bounds for policy iteration. In *Proc. of 37th ICALP*, pages 551–562, 2010. `doi:10.1007/978-3-642-14162-1_46`.

6 O. Friedmann. An exponential lower bound for the latest deterministic strategy iteration algorithms. *Logical Methods in Computer Science*, 7(3), 2011. `doi:10.2168/LMCS-7(3:23)2011`.

7 O. Friedmann. A subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games. In *Proc. of 15th IPCO*, pages 192–206, 2011. `doi:10.1007/978-3-642-20807-2_16`.

8 O. Friedmann, T.D. Hansen, and U. Zwick. A subexponential lower bound for the Random Facet algorithm for parity games. In *Proc. of 22nd SODA*, pages 202–216, 2011.

9 O. Friedmann, T.D. Hansen, and U. Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In *Proc. of 43th STOC*, pages 283–292, 2011. `doi:10.1145/1993636.1993675`.

10 O. Friedmann, T.D. Hansen, and U. Zwick. Errata for: A subexponential lower bound for the random facet algorithm for parity games. *CoRR*, abs/1410.7871, 2014. URL: `http://arxiv.org/abs/1410.7871`.

11 O. Friedmann, T.D. Hansen, and U. Zwick. Random-facet and random-bland require subexponential time even for shortest paths. *CoRR*, abs/1410.7530, 2014. URL: `http://arxiv.org/abs/1410.7530`.

12 B. Gärtner. The Random-Facet simplex algorithm on combinatorial cubes. *Random Structures and Algorithms*, 20(3):353–381, 2002. `doi:10.1002/rsa.10034`.

13 B. Gärtner and V. Kaibel. Two new bounds for the Random-Edge simplex-algorithm. *SIAM J. Discrete Math.*, 21(1):178–190, 2007. `doi:10.1137/05062370X`.

14 B. Gärtner and A. Thomas. The niceness of unique sink orientations. Unpublished manuscript, 2016. URL: `https://sites.google.com/site/antonisthomas/research/niceness.pdf`.

15 D. Goldfarb and W.Y. Sit. Worst case behavior of the steepest edge simplex method. *Discrete Applied Mathematics*, 1(4):277–285, 1979.

16 T.D. Hansen, M. Paterson, and U. Zwick. Improved upper bounds for Random-Edge and Random-Jump on abstract cubes. In *Proc. of 25th SODA*, pages 874–881, 2014.

17 T.D. Hansen and U. Zwick. An improved version of the Random-Facet pivoting rule for the simplex algorithm. In *Proc. of 47th STOC*, pages 209–218, 2015.

18 R. G. Jeroslow. The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Mathematics*, 4(4):367–377, 1973.

19 G. Kalai. A simple way to tell a simple polytope from its graph. *Journal of combinatorial theory, Series A*, 49(2):381–383, 1988.

20 G. Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proc. of 24th STOC*, pages 475–482, 1992.

21 G. Kalai. Linear programming, the simplex algorithm and simple polytopes. *Mathematical Programming*, 79:217–233, 1997.

22 N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

23 L.G. Khachiyan. A polynomial time algorithm in linear programming. *Soviet Math. Dokl.*, 20:191–194, 1979.

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

25 D.A. Levin, Y. Peres, and E.L. Wilmer. *Markov chains and mixing times.* American Mathematical Soc., 2009.

26 Y. Mansour and S.P. Singh. On the complexity of policy iteration. In *Proc. of the 15th UAI*, pages 401–408, 1999.

27 J. Matoušek. Lower bounds for a subexponential optimization algorithm. *Random Structures and Algorithms*, 5(4):591–608, 1994.

28 J. Matoušek and B. Gärtner. *Understanding and using linear programming.* Springer, 2007.

29 J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

30 J. Matoušek and T. Szabó. Random Edge can be exponential on abstract cubes. In *Proc. of 45th FOCS*, pages 92–100, 2004.

31 J. Matoušek and T. Szabó. RANDOM EDGE can be exponential on abstract cubes. *Advances in Mathematics*, 1(204):262–277, 2006.

32 A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, 1986.

33 I. Schurr and T. Szabó. Finding the sink takes some time: An almost quadratic lower bound for finding the sink of unique sink oriented cubes. *Discrete & Computational Geometry*, 31(4):627–642, 2004. doi:10.1007/s00454-003-0813-8.

34 T. Szabó and E. Welzl. Unique sink orientations of cubes. In *Proc. of 42th FOCS*, pages 547–555, 2001.

35 E. Welzl. Towards the peak – problem section. Unpublished manuscript, 2001. URL: http://www.ti.inf.ethz.ch/ew/workshops/01-lc/problems/node7.html.

36 K. Williamson Hoke. Completely unimodal numberings of a simple polytope. *Discrete Applied Mathematics*, 20(1):69–81, 1988. doi:10.1016/0166-218X(88)90042-X.