Jumping Doesn't Help in Abstract Cubes

Ingo Schurr[⋆] and Tibor Szabó[⋆⋆]

Theoretical Computer Science, ETH Zürich
CH-8092 Zürich, Switzerland
{schurr,szabo}@inf.ethz.ch

Abstract. We construct a class of abstract objective functions on the cube, such that the algorithm BottomAntipodal takes exponentially many steps to find the maximum. A similar class of abstract objective functions is constructed for the process BottomTop, also requiring exponentially many steps.

1 Introduction

The model. Let $P \subseteq \mathbb{R}^n$ be a convex polyhedron (given as the intersection of m halfspaces) and $c : \mathbb{R}^n \to \mathbb{R}$ be a linear objective function; *linear programming* seeks a vertex of P maximizing c. While linear programming is known to have a polynomial time algorithm in the *bit size* of the input, its complexity in the so-called *unit-cost* model remains an important open question. That is, what is the smallest $f(n, m)$ such that any linear program in dimension n with m constraints can be solved in time at most $f(n, m)$ if all arithmetic operations are assumed to incur unit cost?

Numerous researchers studied this problem and still our understanding is far from satisfactory. The best known algorithms, due to Kalai [9] and Matoušek, Sharir, and Welzl [11], work in time $e^{O(\sqrt{m \log n})}$. An important aspect of both approaches is that they disregard most of the geometric content of the problem and consider only a basic combinatorial skeleton.

One of the most natural and useful combinatorial simplifications is the concept of *abstract objective functions*, which was first introduced by Adler and his coauthors [1, 2], and later by Williamson Hoke [15] and Kalai [8] under different names. Given a 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. A function $f : V \to \mathbb{R}$ is called an abstract objective function if on every face F of P there is a unique local maximum of f. That is, there is a unique vertex $v \in F$, such that v has a larger f-value than all its neighbors in $G(P)$. In particular, this unique local maximum is a global maximum on F.

[⋆] Supported by the joint Berlin/Zürich graduate program Combinatorics, Geometry, and Computation (CGC), financed by German Science Foundation (DFG) and ETH Zurich.

[⋆⋆] Research is supported by the SNF grant 200021-100316/1.

Note that a generic linear function is a special case of an abstract objective function, though most abstract objective functions of a given polytope cannot be achieved by a linear function. The model is still quite powerful: Kalai's subexponential randomized simplex algorithm for linear programming works for arbitrary abstract objective functions.

In our paper we prefer to formulate our results in an alternative but equivalent framework. An orientation of the edges of the graph $G(P)$ is called an *acyclic unique sink orientation*[1] or *AUSO* of the polytope P if the induced subgraph of $G(P)$ on the vertex set of any face of P has exactly one sink (vertex of outdegree zero). We assume that the orientation is given by an oracle: after querying a vertex the oracle returns the orientation of the edges incident to the queried vertex.

Any abstract objective function induces an AUSO: Orient every edge from the vertex with the smaller value to the one with the larger value. The optimum vertex with the largest value of the objective function becomes the (unique) sink of $G(P)$. On the other hand, given an AUSO, the vertices of $G(P)$ can be numbered according to an arbitrary linear order which extends the partial order defined by the AUSO. This numbering defines an abstract objective function on P.

The algorithm. The oldest and most natural combinatorial attack to solve a linear program is the *simplex algorithm*. Geometrically, it can be viewed as follows: We start at some initial vertex of the polytope P and at each step we move 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 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 [10] 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 increases along it. Subsequently such worst-case examples were found by various researchers for almost all known deterministic pivot rules; see Goldfarb [6] for an overview and Amenta and Ziegler [3] for a new unified view of these examples.

Motivated by the unsuccessful attempts of deterministic simplex algorithms, Kaibel [7] suggested a very non-simplex-like algorithm, something which takes not only one of the outgoing edges into account, but, in some sense all of them. Of course in order to abandon the idea of progress along an edge, one needs to know something about the structure of the polytope. For example, when our polytope is combinatorially equivalent to the n-dimensional cube, then it makes sense to

[1] For some purposes, it is also very interesting to consider unique-sink orientations of polytopes that are not necessarily acyclic (see, e.g., [12, 14, 13]).

speak about an "antipodal vertex". Each vertex is at the bottom of the face generated by its outgoing edges, as it is the source of this face. Motivated by the picture of the orthogonal cube, the algorithm BottomAntipodal jumps from the bottom vertex of the face generated by its outgoing edges to the antipodal vertex within this face.

We also consider a process called BottomTop (also suggested by Kaibel [7]), which, in some sense, represents the most greedy approach one can imagine. Being at a vertex v and knowing the adjacent outgoing edges, one knows that every vertex in the face generated by these outgoing edges is *better* than the current vertex. Suppose we have access to an oracle which tells us the *best* vertex, i.e., the sink, in this subcube, and thus we are able to jump there in one step. It is then plausible to believe this to be a good idea. A step of the process BottomTop is defined by jumping from the current vertex v to the sink v' in the subcube generated by the outgoing edges incident to v. BottomTop is of course *not* an algorithm, since we need to have access to an oracle which tells us the sink of a subcube once we provide the source.

The results. In Sect. 4 of this paper we construct an acyclic unique sink orientation of the n-dimensional cube, such that BottomAntipodal takes an exponential number of queries to find the sink. In Sect. 5 we give a construction of an AUSO of the cube on which BottomTop performs an exponential number of queries.

2 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 vw stands for the concatenation of the vectors v and w. The zero vector of dimension greater than one is denoted by $\mathbf{0}$ 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 $v + e_i$, $i = 1, 2, \ldots, n$.

Formally we will identify an n-dimensional AUSO A with its *outmap* $s_A : \{0,1\}^n \to \{0,1\}^n$, 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. It is known that the outmap s_A is a bijection for any AUSO A, even if we restrict it to an arbitrary subcube. In particular, an AUSO does not only have a unique sink per face, but also, e.g., a unique source. For this and other facts about unique sink orientations of cubes see, for example, [14].

We say that two AUSOs 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.

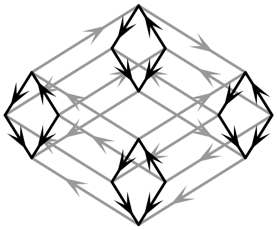

Fig. 1. The blowup-construction: Four identical 2-dimensional AUSO (drawn in black) are interconnected by 2^2 2-dimensional frames (drawn in gray).

In the following we describe two lemmas, special cases of results of [13], which allow us to construct new AUSOs from old ones. The first lemma uses the product structure of the cube.

Lemma 1 (Blowup construction, [13, Lemma 3]). *Let A be an AUSO of dimension m and for each $u \in \{0,1\}^m$ let B_u be a n-dimensional AUSO. Then the map $s_C : \{0,1\}^{m+n} \to \{0,1\}^{m+n}$ defined by $s_C(uv) = s_A(u)s_{B_u}(v)$ is the outmap of an $(m+n)$-dimensional AUSO C.*

One can imagine that we blow up each vertex of A to a n-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^n copies of A and, for each vertex u of A, interconnecting all the 2^n copies of u by an n-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 $m \leq n$). We call S a *hypersink* of A if all edges connecting vertices of S to vertices outside S are oriented towards S.

Lemma 2 (Hypersink reorientation, [13, 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.*

As a warm-up let us recall the definition of the Klee-Minty cube in the framework of our lemmas. The zero-dimensional Klee-Minty cube KM_0 consists of one vertex. To construct KM_n we take two copies K and K' of KM_{n-1} and flip the orientations of all edges in one of them, say in K'. Then we add a perfect matching between the vertices of K and K' having identical coordinates and orient these edges from K' towards K . Note that by Lemma 1 the resulting orientation is an AUSO: We interconnect 2^{n-1} copies of a 1-dimensional USO using the two frames K and K'. Originally the Klee-Minty cube was defined as a geometric object. It is a polytope combinatorially equivalent to the cube,

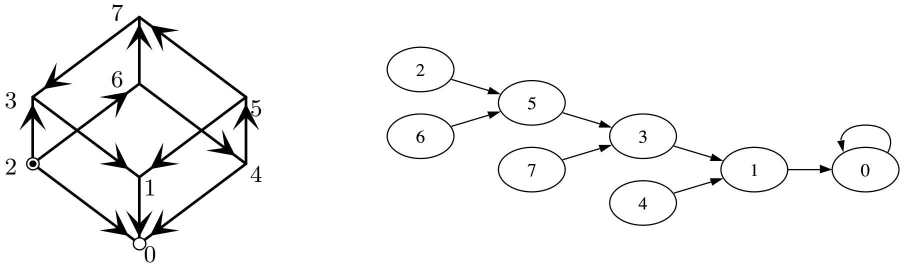

Fig. 2. An AUSO A with a graph $T_{\mathrm{ba}}(A)$ of height 4.

which produces the orientation defined above if we evaluate an appropriate linear objective function on its vertices. (See, e.g, [4] for a more detailed study on Klee-Minty cubes.) As we mentioned in the introduction the Klee-Minty cube is a worst-case example for many of the natural pivot rules of the simplex algorithm. As it turns out, BottomAntipodal and BottomTop are very efficient on the Klee-Minty cube, even on a much wider class of AUSOs called *decomposable* orientations (See [15] for an introduction to decomposable AUSOs).

Lemma 3. *Starting at an arbitrary vertex,* BottomTop *needs at most* $n+1$ *steps to find the sink of the* n*-dimensional Klee-Minty cube.*

Due to the construction of the Klee-Minty cubes for the coordinate n all edges are oriented towards the same facet S_0. In particular *all* subcubes have their sink in S_0, so after the first step BottomTop will be in S_0. Repeating this argument inductively, after $n+1$ steps we are in the sink. A similar argument works for BottomAntipodal (see [13, Proposition 7]).

3 The BottomAntipodal Tree

The behavior of BottomAntipodal on an AUSO A can be described by the following directed graph $T_{\mathrm{ba}}(A)$: The vertex set of $T_{\mathrm{ba}}(A)$ is the vertex set of the underlying cube of A. Two vertices v, w form an edge $v \to w$ if $v + s_A(v) = w$. In particular, every vertex v has exactly one outgoing edge $v \to v + s_A(v)$. We call $v + s_A(v)$ the *successor* of v and denote it by $\mathrm{succ}_A(v)$. The sink o of s_A is special since it is the only vertex in $T_{\mathrm{ba}}(A)$ having a loop. For an example, see Fig. 2.

The unique path starting in a vertex v in $T_{\mathrm{ba}}(A)$ will be called the *trace* of v. Obviously, the trace of a vertex v is the sequence of queries BottomAntipodal produces starting in v.

BottomAntipodal will terminate on any AUSO, i.e., $T_{\mathrm{ba}}(A)$ is a tree. The vertex v is a source in the cube spanned by v and $v + s_A(v)$. The following lemma provides us with a path from v to $v + s_A(v)$ in the AUSO A. (The proof is rather easy and omitted.)

Lemma 4. *There is a path from the source of an AUSO to any vertex of the cube.*

By the above lemma, the trace of a vertex in $T_{\mathrm{ba}}(A)$ induces a trail in A. In particular, if $T_{\mathrm{ba}}(A)$ contains a cycle, so does A. Hence for an AUSO A the graph $T_{\mathrm{ba}}(A)$ is a tree, the so-called *bottom-antipodal tree*. This tree was first introduced by Kaibel [7] in connection to randomized simplex algorithms.

Since $T_{\mathrm{ba}}(A)$ is a tree, the trace of a vertex v is a path in $T_{\mathrm{ba}}(A)$ to the sink of A. The length of this path is the *height* $h_A(v)$ of v. Obviously, $h_A(v)$ differs by one from the number of queries of BottomAntipodal starting in v. The height $h(A)$ of $T_{\mathrm{ba}}(A)$ is defined as the maximal height of a vertex in A and the average height $\tilde{h}(A)$ is the average over the heights of all vertices in A, that is, for an n-dimensional A we have

$$h(A) = \max\{h_A(v) \mid v \in \{0,1\}^n\} \qquad \tilde{h}(A) = \frac{1}{2^n} \sum_{v \in \{0,1\}^n} h_A(v).$$

The height of A corresponds to the worst-case behavior of BottomAntipodal, whereas the average height reflects the expected behavior if we randomize the starting vertex. Thus, if we can construct a family of examples for which the maximal height grows exponentially, then BottomAntipodal has exponential worst-case complexity.

4 The Construction

Theorem 1. *Let A be an arbitrary n-dimensional AUSO. Then there exists a $(n+2)$-dimensional AUSO D, such that the height of D is at least $2h(A)$ and the average height of D is at least $\frac{3}{4}\tilde{h}(A) + \frac{1}{2}h(A)$.*

Starting, e.g., with $A_2 = KM_2$ and the orientation A_3 from Fig. 2, Theorem 1 yields a sequence of AUSOs with exponential height.

Corollary 1. *In every dimension $n \geq 2$ there is an AUSO, such that the height of the corresponding bottom-antipodal tree is at least $\sqrt{2}^n$ and the average height is at least $\frac{2}{5}\sqrt{2}^n$.*

Proof (of Theorem 1).
We can assume without loss of generality that the sink of A is in 0. As an intermediate step we first construct an $(n+2)$-dimensional AUSO C, which is the blow-up of A. For each vertex $u \in \{0,1\}^n$ we select a 2-dimensional AUSO B_u which is isomorphic to KM_2. We do not, however, select identical copies. Let $K_0 = KM_2$ be the 2-dimensional Klee-Minty cube, while K_1 be KM_2 with its two coordinates permuted. (Fig. 3)

Now let $B_u = K_0$ if the height $h_A(u)$ is even and let $B_u = K_1$ if $h_A(u)$ is odd. We let C be the blowup of A by these B_u. So, according to our preferred view of the blowup construction, we take 4 copies of A and interconnect them by the 2-dimensional frames B_u, each is the Klee–Minty cube with possibly permuted

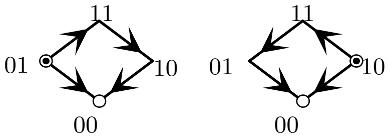

Fig. 3. The two orientations K_0 and K_1.

coordinates. All B_u have their sink at $u00$, that is, in the same n-dimensional face S_0, which is a copy of A. Consequently, this copy of A is a hypersink in C. Let v_{max} be a vertex of A with $h_A(v_{max}) = h(A)$. We now reorient the hypersink S_0 by rotating v_{max} to 0 (which used to be the sink in S_0). More formally, we obtain a new orientation A' isomorphic to A, where the outmap of a vertex z is obtained by $s_{A'}(z) = s_A(z+w)$. Then we orient the hypersink S_0 of C according to A' and we denote the resulting $(n+2)$-dimensional AUSO by D.

We can also express the outmap of D formally. For an $(n+2)$-dimensional vector v denote by v' the projection of v to the first n coordinate, while v'' is the projection of v to the last 2 coordinates. Then $s_D(v)' = s_A(v')$ unless $v'' = 0$ in which case $s_D(v)' = s_A(v' + v_{max})$. The last two coordinates $s_D(v)''$ of the outmap $s_D(v)$ only depend on the last two coordinates of v and the parity of $h_A(v')$. See Table 1 for the details.

Table 1. The outmap of the constructed AUSO D in the last two coordinates and the color-coding of the vertices of D.

v''	00	11		10		01	
$h_A(v')$		even	odd	even	odd	even	odd
$s_D(v)''$	00	01	10	10	11	11	01
color	blue	green		red	yellow	yellow	red

We introduce a color-coding of the vertices depending on where they lie in the 2-dimensional Klee-Minty frame they belong to. The hypersink S_0, having vertices with last two coordinates 00, is composed of blue vertices only. The green vertices occupy the n-dimensional subcube S_3 with last two coordinates 11. The remaining two subcubes, S_1 and S_2 are occupied by yellow and red vertices, depending on the parity of the height of the projection of the particular vertex in A. Crucially, $v \in S_1 \cup S_2$ and $v + \mathbf{0}11$, being in the same Klee-Minty frame, have different colors; one of them is red, the other is yellow. See Table 1 on how the vertices are colored. Fig. 4 aims to illustrate the construction and color-coding.

In the following we make a few observations about the successor $\mathrm{succ}_D(v)$ of a vertex $v \in \{0,1\}^{n+2}$ in the bottom antipodal tree of D.

The successor of a blue vertex is blue. Moreover if v is blue then $\mathrm{succ}_D(v)' = \mathrm{succ}_{A'}(v')$. In other words once BottomAntipodal is in the hypersink S_0,

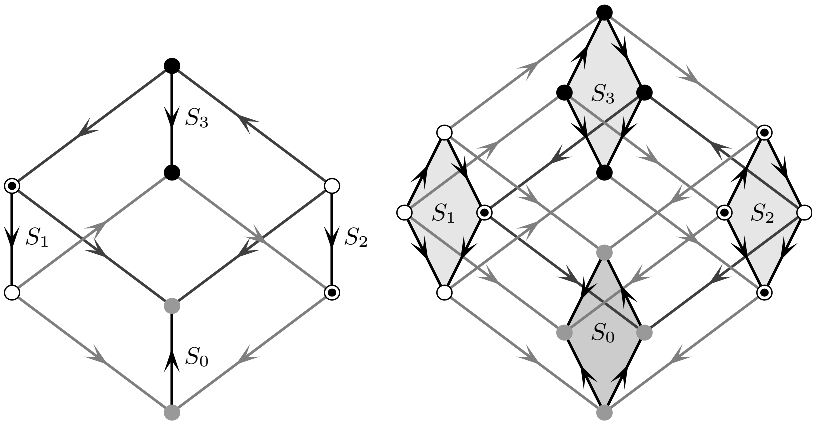

Fig. 4. The orientations A_3 and A_4. The orientation A_3 to the left is constructed from the 1-dimensional AUSO A_1 with sink in $\mathbf{0}$. The orientation A_4 to the right is based on A_2. The subcubes S_1, S_2, and S_3 contain A_1 and A_2, respectively, whereas S_0 contains a flipped variant. The colors are encoded the following way: yellow ○, green ●, blue ●, red ⊙.

it never leaves S_0 and follows the trace of A' (which is isomorphic to A). In particular $h_D(v) = h_{A'}(v') = h_A(v + v_{max})$.

Next we check the height of the four vertices with $v' = \mathbf{0}$. By construction, the vertices $\mathbf{0}01$, $\mathbf{0}10$, and $\mathbf{0}11$ are the sinks of the copies of A in S_1, S_2, S_3, respectively. In the subcube S_0 we rotated A, such that the sink and $v_{\max}$ change position. Hence, $\mathbf{0}00$ has height $h_D(\mathbf{0}00) = h_A(v_{max}) = h(A)$. The vertex $\mathbf{0}10$ has only its $(n+1)$-edge outgoing. Thus, its successor is $\mathbf{0}00$ and $h_D(\mathbf{0}10) = 1 + h_D(\mathbf{0}00) = 1 + h(A)$. The vertex $\mathbf{0}01$ has outgoing edges along coordinate $(n+1)$ and $(n+2)$ and its successor is $\mathbf{0}10$. Therefore $h_D(\mathbf{0}01) = 1 + h_D(\mathbf{0}10) = 2 + h(A)$. Finally, the vertex $\mathbf{0}11$ has only its $(n+2)$-edge outgoing and $h_D(\mathbf{0}11) = 1 + h_D(\mathbf{0}10) = 2 + h(A)$.

The remaining vertices are considered by their color. Assume now that v is not blue and $v' \neq \mathbf{0}$.

If v is red then its successor is blue.

If v is yellow then its successor is also yellow. Moreover $\mathrm{succ}_D(v)' = \mathrm{succ}_A(v')$. A yellow vertex is either in S_1 or S_2. Since $s_D(v)'' = 11$, the successor of v is in the antipodal n-face. Moreover $\mathrm{succ}_D(v)' = \mathrm{succ}_A(v')$, since the copies of A in S_1 and S_2 are translates of each other. Now $\mathrm{succ}_D(v)$ is yellow because the height of $\mathrm{succ}_D(v)'$ is one less, thus has a different parity than the height of v'. Thus, the trace of v reaches the sink of S_2 through yellow vertices while the first n coordinates are going through the trace of v' in A. Hence, we have that $h_D(v) = h_A(v') + 1 + h_D(\mathbf{0}10) = h_A(v') + h(A) + 2$.

If v is green then its successor is yellow. Moreover $\mathrm{succ}_D(v)' = \mathrm{succ}_A(v')$.

The argument is similar to the above. As v is in S_3 and $s_D(v)'' = 01$ or 10, the successor of v is either in S_1 or S_2 Moreover $\mathrm{succ}_D(v)' = \mathrm{succ}_A(v')$, since the copies of A in S_3, S_1, and S_2 are translates of each other. Now $\mathrm{succ}_D(v)$ is yellow: The height of $\mathrm{succ}_D(v)'$ has a different parity than the height of v'. Hence, $\mathrm{succ}_D(v)$ is in a different frame than v and $s_D(\mathrm{succ}_D(v))'' = 11$. In fact the successor of v is identical to the successor of the yellow vertex w in the 2-dimensional Klee-Minty frame v belongs to. Quantitatively, $h_D(v) = h_D(w) = h_A(w') + h(A) + 2 = h_A(v') + h(A) + 2$.

Now consider the green vertex u with $u' = v_{\max}$. By the above we obtain that

$$h(D) \geq h_D(u) = h_A(u') + h(A) + 2 = 2h(A) + 2.$$

For the average height $\tilde{h}(D)$ we forget about the red vertices (as we don't know their height) and get (with B being the set of blue vertices, Y the set of yellow vertices and G the set of green vertices)

$$\begin{aligned}\tilde{h}(D) &= \frac{1}{2^{n+2}} \sum_{v} h_D(v) \geq \frac{1}{2^{n+2}} \left(\sum_{v \in B} h_D(v) + \sum_{v \in Y} h_D(v) + \sum_{v \in G} h_D(v) \right) \\ &= \frac{1}{4} \cdot \frac{1}{2^n} \sum_{v \in B} h_{A'}(v') + \frac{1}{4} \cdot \frac{1}{2^n} \sum_{v \in Y} h_A(v') + \frac{1}{4} \cdot \frac{1}{2^n} \sum_{v \in G} h_A(v') + \frac{h(A)}{2} + 1 \\ &= \frac{3}{4} \tilde{h}(A) + \frac{1}{2} h(A) + 1.\end{aligned}$$

□

5 The BottomTop Tree

Recall that the process BottomTop in each step jumps to the sink of the subcube spanned by the outgoing edges of the current vertex. We define a directed graph $T_{\mathrm{bt}}(A)$, similar to $T_{\mathrm{ba}}(A)$. The vertex set of $T_{\mathrm{bt}}(A)$ is the vertex set of A. Two vertices u and v are connected by an edge $u \to v$, if v is the sink in the subcube spanned by u and the outgoing edges incident to u. Again, we call v the successor of u. Since A has a unique sink in every subcube, each vertex u has exactly one successor v. Also, since A is acyclic, by Lemma 4 again $T_{\mathrm{bt}}(A)$ is a tree, the *bottom top tree*. Let $t_A(v)$ be the height of v in $T_{\mathrm{bt}}(A)$, $t(A)$ be the height of $T_{\mathrm{bt}}(A)$ and $\tilde{t}(A)$ be the average height of $T_{\mathrm{bt}}(A)$.

Theorem 2. *Let A be an arbitrary n-dimensional AUSO. Then there exists a $(n+2)$-dimensional AUSO D, such that the height of the bottom top tree of D is at least $2t(A)$ and the average height is at least $\frac{1}{2}\tilde{t}(A) + \frac{1}{4}t(A)$.*

Proof. We repeat the construction in Theorem 1, but now we use t instead of h. That is, v_{max} now is of maximal height with respect to t_A and the copies of K_0 and K_1 for the frame are chosen according to the parity of $t_A(u)$.

With the same color coding as in Theorem 1 we now have the following picture. As before, blue vertices have blue successors and $\mathrm{succ}_D(v)' = \mathrm{succ}_{A'}(v')$ provided v is a blue vertex. The vertex $\mathbf{0}00$ corresponds to v_{max} in A, hence

$t_D(\mathbf{0}00) = t(A)$. The vertices $\mathbf{0}01$ and $\mathbf{0}10$ have successor $\mathbf{0}00$ and thus $t_D(\mathbf{0}01) = 1 + t(A) = t_D(\mathbf{0}10)$. Finally, $\mathbf{0}11$ has successor $\mathbf{0}10$.

Assume that $v' \neq \mathbf{0}$. Red and yellow vertices all have blue successors. The major difference, compared to the proof of Theorem 1 is that now the green vertices have green successors. A green vertex v has exactly one of the edges along coordinates $n+1$ and $n+2$, say $n+1$, outgoing. Thus, the subcube B spanned by v and its outgoing edges consists of two facets, each is a translate of the subcube B', spanned by v' and its outgoing edges in A. The sink of B is either in the vertex $w'01$ or $w'11$, where w' is the sink of B'. Since $w' = \mathrm{succ}_A(v')$, the value $t_A(w')$ is one less, thus has a different parity than $t_A(v')$. Thus the edge between $w'01$ and $w'11$ is oriented towards $w'11$, which is then the sink of B and the successor of v. Note, that $w'11$ is a green vertex and $\mathrm{succ}_D(v)' = \mathrm{succ}_A(v')$.

For $u = v_{max}11$ we now obtain $t_D(u) = t_A(v_{max}) + 2 + t(A) = 2t(A) + 2$.

The average height of D we estimate by forgetting about all the red and yellow vertices. Summing over the green and blue vertices we get an estimate of

$$\tilde{t}(D) \geq \frac{1}{2}\tilde{t}(A) + \frac{1}{4}t(A).$$

□

Easy calculation shows that $t(A_2) = 2$, $\tilde{t}(A_2) = 1$, $t(A_3) = 4$, and $\tilde{t}(A_3) = 7/4$. This yields the following corollary.

Corollary 2. *In every dimension $n \geq 2$ there is an AUSO, such that the height of the corresponding* BottomTop *tree is at least $\sqrt{2}^n$ and the average height is at least $\frac{1}{6}\sqrt{2}^n$.*

6 Remarks and Open Problems

1. In [14] unique sink orientations of cubes were investigated, which are not necessarily acyclic. As it turns out this more general model is quite useful, for example it contains linear programming in its whole generality, not just on polytopes combinatorially equivalent to the cube. This connection is more abstract than the obvious relation to AUSOs, more details about it are found in [5]. BottomAntipodal is also possible to perform on this more general model, except that it is even less useful than for AUSOs since there BottomAntipodal can cycle.

2. It is an intriguing open question to decide what happens when BottomAntipodal or BottomTop is performed on a realizable AUSO, that is on a polytope combinatorially equivalent to the cube. Note that for the Klee-Minty-cube, and in fact for the much larger class of decomposable orientations, both processes have a worst case running time of $n+1$.

Also, we don't know whether it is possible to extend our construction to AUSOs which satisfy the Holt-Klee condition. (The Holt-Klee condition, satisfied by all realizable polytopes, requires that there are d edge-disjoint source-to-sink paths in any d-dimensional face of the polytope.)

Acknowledgment

We would like to thank Emo Welzl for stimulating discussions.

References

1. I. Adler, G. B. Dantzig. Maximum diameter of abstract polytopes. *Math. Programming Studies*, 1:20–40, 1974.
2. L. Adler, R. Saigal, Long monotone paths in abstract polytopes *Math. Oper. Res.*, 1:89–95, 1976.
3. N. Amenta and G. M. Ziegler. Shadows and slices of polytopes. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1996.
4. B. Gärtner, M. Henk, and G. M. Ziegler. Randomized simplex algorithms on Klee-Minty cubes. *Combinatorica*, 18(3):349–372, 1998.
5. B. Gärtner and I. Schurr. Linear Programming and Geometric Unique Sink Orientations. Manuscript in preparation, 2004.
6. D. Goldfarb. On the complexity of the simplex algorithm. In *Advances in optimization and numerical analysis*, pages 25–38, Dordrecht, 1994. Kluwer.
7. V. Kaibel. *The La Claustra lecture.*
8. G. Kalai. A simple way to tell a simple polytope from its graph. *J. Combin. Theory, Ser. A*, 49(2):381–383, 1988.
9. G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.
10. V. Klee, G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, 1972.
11. J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
12. W. D. Morris. Randomized principal pivot algorithms for P-matrix linear complementarity problems. *Mathematical Programming*, Ser. A, 92:285–296, 2002.
13. 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.
14. T. Szabó, E. Welzl. Unique sink orientations of cubes. In *Proc.* 42^{nd} *IEEE Symp. on Foundations of Comput. Sci.*, pages 547–555, 2001.
15. K. Williamson Hoke. Completely unimodal numberings of a simple polytope. *Discrete Appl. Math.*, 20:69–81, 1988.