The Complexity of Recognizing Unique Sink Orientations

Bernd Gärtner and Antonis Thomas

Department of Computer Science,
Institute of Theoretical Computer Science, ETH Zürich
8092 Zürich, Switzerland
{gaertner,athomas}@inf.ethz.ch

Abstract

Given a Boolean Circuit with n inputs and n outputs, we want to decide if it represents a Unique Sink Orientation (USO). USOs are useful combinatorial objects that serve as abstraction of many relevant optimization problems. We prove that recognizing a USO is **coNP**-complete. However, the situation appears to be more complicated for recognizing acyclic USOs. Firstly, we give a construction to prove that there exist cyclic USOs where the smallest cycle is of superpolynomial size. This implies that the straightforward representation of a cycle (i.e. by a list of vertices) does not make up for a **coNP** certificate. Inspired by this fact, we investigate the connection of recognizing an acyclic USO to **PSPACE** and we prove that the problem is **PSPACE**-complete.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics

Keywords and phrases complexity, recognizing, unique sink orientations, coNP, PSPACE

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.341

1 Introduction

Over the past 15 years, *unique sink orientations* (USO) have intensively been studied as simple and appealing combinatorial models for many concrete optimization problems. After their introduction by Stickney and Watson in the context of mathematical programming [16], USOs had been forgotten for more than 20 years, before Szabó and Welzl rediscovered them from a computational geometry angle [17]. Subsequently, the structural, algorithmic, and combinatorial aspects of USOs were investigated; new applications were found, in particular in the area of mathematical programming where the concept originally comes from. We refer the interested reader to Foniok at al. [4] and the references therein.

A USO is an orientation of the n-dimensional hypercube graph, with the property that every face of dimension $d \in \{0, 1, \dots, n\}$ induces a subgraph with a unique sink. In particular, there is a unique global sink, and the algorithmic problem is to find it.

In all known applications, the USO is given in *succinct representation*, i.e. there is an oracle that returns for a given vertex the orientations of the incident edges, and the question is how many oracle calls are necessary in order to find the global sink. The oracle itself can typically be implemented by a polynomial-time algorithm.

For a concrete such application, consider the problem of finding the smallest enclosing ball $B(P)$ of a set P of n affinely independent points in $\mathbb{R}^{n-1}$. Every subset $Q \subseteq P$ naturally corresponds to a vertex of the hypercube, and we have a directed edge from Q to $Q \cup \{p\}, p \notin Q$, if and only if p is outside of $b(Q)$, the smallest ball that has all points of Q *on its boundary*. This ball $b(Q)$ is easy to compute by solving a system of linear equations, so we have a polynomial-time oracle at our disposal. Moreover, the global sink S has the

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 341–353
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

property that $b(S) = B(P)$, hence finding the global sink solves our geometric problem. The USO approach also works for the more general problem of finding the smallest enclosing ball of a set of *balls* [3], for general linear programs (LP) [8], and—this is the original application by Stickney and Watson— for P-matrix linear complementarity problems (PLCP) [16].

None of these problems have known *strongly* polynomial-time algorithms (i.e. polynomial in the real RAM model of computation); for PLCP, even weakly polynomial-time algorithms are not known. This would change if we could find the sink of an n-dimensional USO with a number of oracle calls that is polynomial in n.

Currently, we cannot, and it even seems somewhat stupid to further generalize problems that are already difficult. On the other hand, for some of the above concrete problems, the USO approach *does* yield the currently best known algorithms. Most notably, this is the case for a relevant class of LP in the RAM model [8], and for PLCP in general [17]. This clearly shows the usefulness of the USO abstraction, and the elegant combinatorial algorithms obtained in this abstraction [17].

Our contribution

In this paper, we study USO from a novel angle. While previous research mostly addresses the algorithmic problem of finding the global sink in a USO, we deal with the more fundamental problem of *recognizing* a USO, given in succinct representation. Concretely, we are interested in the computational complexity of deciding whether a succinct oracle indeed specifies a USO. In order to fit this problem into standard complexity theory, we assume that the oracle is implemented by a succinct Boolean circuit that in turn forms the input for the decision problem. Such a circuit has n input and n outputs, where n is the dimension of the USO; it is customary to use Boolean circuits in describing graphs succinctly (cf. [5]). By *succinct*, we mean that the size of the circuit is polynomial in n. We prove the following two main results.

1. It is **coNP**-complete to recognize USO, given in succinct Boolean circuit representation.
2. It is **PSPACE**-complete to recognize *acyclic* USO (AUSO), given in succinct Boolean circuit representation.

Here, an AUSO is a USO without any directed cycles. These results may come as a surprise, given that the *algorithmic* problem seems to be easier in the acyclic case: the best known (randomized) algorithm for finding the sink in an AUSO requires only a *subexponential* number of $\exp(2\sqrt{n})$ oracle calls [6]. For general USO, the best randomized bound is $O(1.438^n)$ [18].

Our results in particular show that there are simple certificates for non-USOs, but probably not for non-AUSOs. We explicitly show with a family of examples that the list of vertices on a directed cycle is not an efficient certificate for a non-AUSO, because such a list may have to be superpolynomially long. The construction works over an interesting and easy-to-analyze subclass of USOs (*flip matching orientations*) and is of independent interest.

The applications we advertise above reduce to digraphs that are guaranteed to be (A)USOs. Still, the complexity of recognizing an (A)USO from a succinct description is interesting from a theoretical viewpoint. In fact, similar theoretical results from the past include the recognizability of a P-Matrix, which is proved **coNP**-complete in [2]. Even though from applications (e.g. solving simple stochastic games [7]) we do get P-Matrices, the question of recognizability is still relevant.

The study of computational problems on graphs that are represented in an exponentially succinct way, through Boolean circuits, has been initiated by Galperin and Wigderson [5]. They proved that a number of trivial graph properties become **NP**-hard when the input of the graph is given in such a succinct way. Subsequently, Papadimitriou and Yannakakis

[13] proved that, under the same representation, problems that are **NP**-complete when the graph is given explicitly become **NEXP**-hard. Finally, in [1], Balcázar *et al.* prove that it is **PSPACE**-hard to decide several fundamental properties in succinct graphs, such as the existence of an Eulerian circuit and of a path connecting two given nodes.

The paper is organized as follows. Firstly, we introduce the concepts and the notation we use, in Section 2, together with three lemmas, from the work of Schurr and Szabó [15], that we use in our constructions. In Section 3 we prove **coNP**-completeness of the USO recognition problem. The **coNP** membership is implicit already in the work of Szabó and Welzl [17], and hardness will follow by a simple reduction from SAT. Section 4 shows that the canonical NO-certificate for the AUSO recognition problem—an explicit list of vertices on a directed cycle—cannot be used to establish **coNP** membership. To this end, we explicitly construct an n-dimensional USO with a unique directed cycle of length $\Omega(2^{n/3})$. Section 5 reveals the deeper reason for the failure of the cycle certificate, namely that the AUSO recognition problem is **PSPACE**-complete. For **PSPACE** membership, we use standard results from complexity theory and the theory of succinct graphs; our main contribution is **PSPACE** hardness, proved via a reduction from satisfiability of quantified Boolean formulas.

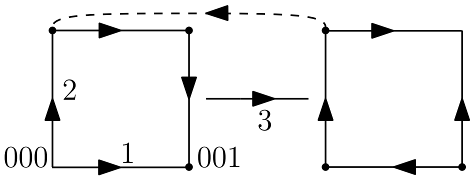
2 Preliminaries

We use the notation $[n] = \{1, \dots n\}$, and $[i : j]$, $i < j$, for $[i : j] = \{i, \dots, j\}$. Let $Q_n = \{0,1\}^n$, which we also interpret as the set of vertices of the n-dimensional hypercube. Let $\psi : Q_n \to Q_n$ be a Boolean function. Moreover, let C_ψ be a Boolean circuit with n inputs and n outputs that represents ψ. There is an explicit ordering on the coordinates and with x_i and $\psi(x)_i$ we denote the ith coordinate of the corresponding bitstring, where the first one is the **rightmost**. Here we use the term bitstring to refer to a ordered string of binary bits. When the subscript is a set, e.g. $x_{[k]}$ or $\psi(x)_{\{2,3\}}$, we mean the bitstring resulting from taking only the coordinates that appear in the subscript. Moreover, we use superscripts to differentiate different functions or different bitstrings, e.g. x^1, x^2 represent two different bitstrings and ψ^1, ψ^2 two different Boolean functions. With the notation $\overline{x_i}$ we mean $1 - x_i$. Given two bitstrings x, y, with $x \cdot y$ we denote their concatenation. Given $x \in Q_n$ we define the neighborhood of x as $\mathcal{N}(x) = \{y \in Q_n | x \text{ and } y \text{ are at Hamming distance } 1\}$.

Let $I \in 2^{[n]}$ and $v \in Q_n$. A face of the hypercube, $F_{I,v}$, is defined as the set of vertices that are reached from v by flipping the coordinates defined by any subset of I, i.e. $F_{I,v} = \{u \in Q_n | u_i = v_i, \forall i \notin I\}$. The dimension of the face is $|I|$. We call edges the faces of dimension 1, e.g. $F_{\{i\},x}$, and vertices the faces of dimension 0.

We say that $\psi : Q_n \to Q_n$ represents an orientation when $\forall i \in [n]$ and $\forall x \in Q_n$ we have that $\psi(x)_i \neq \psi(x')_i$, where $F_{\{i\},x} = \{x, x'\}$, i.e. the orientation of every edge is consistent from both sides. Given an orientation ψ and a vertex $x \in Q_n$, we call $\psi(x)$ the *outmap* of x and we call ψ, simply, the outmap. With $G_\psi = (Q_n, E_n)$ we mean the digraph of the hypercube where the edges are oriented according to the outmap. That means that the edge on coordinate i is outgoing for vertex x if and only if $\psi(x)_i = 1$ (cf. Figure 1). Given an orientation ψ of Q_n, a vertex $x \in Q_n$ and an incident edge $F_{\{i\},x}$ we say that the edge is oriented backwards if $\psi(x)_i = x_i$ and is oriented forwards if $\psi(x)_i = \overline{x_i}$. For example, consider vertex 001 in Figure 1: the incident edges on coordinates 1 and 3 are forwards while the one on coordinate 2 is backwards. With $\psi_{\mathbb{UF}}$ we denote the orientation such that $\psi_{\mathbb{UF}}(x) = \overline{x}$, for all $x \in Q_n$. This is orientation is called uniform forwards (all edges are forwards); similarly, the orientation defined by $\psi_{\mathbb{UB}}(x) = x$, for all $x \in Q_n$, is called uniform backwards.

▶ **Definition 1.** A *unique sink orientation* (USO) is an orientation of the hypercube where



vertex	outmap
000	111
001	100
010	001
011	110

vertex	outmap
100	010
101	011
110	101
111	000

Figure 1 An example of a cyclic USO (the vertices participating in the cycle are highlighted as discs). In the left part we give an illustration of the USO graph G_ψ (we explicitly indicate the vertices 000 and 001) and in the right part we give explicitly the Boolean function $\psi : Q_3 \to Q_3$. In this paper we draw USOs by depicting a number of faces (in this case the 2-dimensional faces) and show the orientation of the edges that connect those. The numbers show the ordering of the coordinates. An arc like the one with label 3 means that all edges on the 3rd coordinate are directed likewise. The dashed arc (also on coordinate 3) means that the specific edge (in this case $F_{\{3\},010}$) is reversed w.r.t. to the the orientation suggested by the (non-dashed) arc labeled 3.

the subgraph induced by every non-empty face has a unique sink.

The existence of a unique sink implies the analogous unique source [17]. As the whole hypercube is a face of itself that means that there is a unique sink (and source) for the whole hypercube, which we call global. We say that ψ or C_ψ represents a USO if the output corresponds to the outmap of a USO and thus G_ψ is a USO. The outmap of a USO is a bijection [17]. If, in addition G_ψ is acyclic, then we call it an acyclic USO (AUSO).

Finally, we give three lemmas from the work of Schurr and Szabó [15] that we use for our constructions. We rephrase the lemmas to use the notation used in the current paper. The first gives us tools to expand a USO to one with more coordinates. The interpretation we use in this paper is that we can take a k_1-dimensional USO and embed in every vertex an k_2-dimensional USO; the end-product is a $(k_1 + k_2)$-dimensional USO, which is acyclic if all involved USOs are acyclic. Note that Lemma 2, as presented in [15], is slightly more general than here; we direct the reader to [15] for full generality.

▶ **Lemma 2** ([15], Lemma 3)**.** *Let $k_1, k_2 \in \mathbb{N}$ and let $\psi : Q_{k_1} \to Q_{k_1}$ represent a USO. Let the Boolean functions $\psi^u : Q_{k_2} \to Q_{k_2}$, for each $u \in Q_{k_1}$, also represent USOs. Consider the Boolean function $\psi' : Q_n \to Q_n$, where $n = k_1 + k_2$. Let $x \in Q_n$; we define ψ' by*

$$\psi'(x) = \psi(x_{[k_2+1:n]}) \cdot \psi^{x_{[k_2+1:n]}}(x_{[k_2]}).$$

ψ' represents a USO. Furthermore, if ψ and all ψ_u are acyclic, then so is ψ'.

The second lemma says that we are allowed to orient the edge that connects two neighboring vertices x^1, x^2 any way we like and still have a USO, as long as the outmaps of the two vertices are exactly the same in every coordinate that is not the one of the incident edge.

▶ **Lemma 3** ([15], Corollary 6)**.** *Let $\psi : Q_n \to Q_n$ represent a USO. Let $x^1, x^2 \in F_{\{i\},x^1} \subseteq Q_n$, such that $\psi(x^1)_{[1:n]\setminus\{i\}} = \psi(x^2)_{[1:n]\setminus\{i\}}$. Then, $\psi' : Q_n \to Q_n$ with $\psi'(x) = \psi(x)$, for all $x \in Q_n$ except $\psi'(x^1)_i = \overline{\psi(x^1)_i}$ and $\psi'(x^2)_i = \overline{\psi(x^2)_i}$ also represents a USO.*

The third gives lemma describe a constructive process to get an acyclic USO where we can choose which vertex is the global sink and which vertex is the global source.

▶ **Lemma 4** ([15], Corollary 4)**.** *For any two distinct $x, y \in Q_n$, there exists a $\psi : Q_n \to Q_n$, with $\psi(x) = 0^n$ and $\psi(y) = 1^n$, such that ψ represents an acyclic USO.*

3 Recognizing USOs

In this section we prove that recognizing a USO is **coNP**-complete. The computational problem is USO-recognizability: We are given a Boolean circuit C_ψ such that $\psi : Q_n \to Q_n$ and the question is if ψ represents a USO. Note that a **coNP** upper bound for this problem is already known by [17]: A pair of vertices $x, y \in Q_n$ such that $\psi_i(x) = \psi_i(y)$, $\forall i \in I$, where $I = \{i \in [n] | x_i \neq y_i\}$, constitutes a short NO certificate.

▶ **Theorem 5.** *USO-recognizability is **coNP**-complete.*

Proof. We describe a reduction from SAT to the complement of our problem. Let ϕ denote a SAT formula with n variables. By $\phi(x)$ we mean the evaluation of ϕ on $x \in \{0,1\}^n$, which returns 0 for false and 1 for true. Based on ϕ we construct the Boolean circuit C_ψ, with $\psi : Q_{n+1} \to Q_{n+1}$. The function is such that on input $x \in Q_n$ we have $\psi(x \cdot 0) = x \cdot 0$ and $\psi(x \cdot 1) = x \cdot \overline{\phi(x)}$. It is easy to see that $x \in Q_n$ is satisfying for ϕ if and only if the pair $x \cdot 0, x \cdot 1$ violates the USO property. ◀

Note that the proof above really is about whether ψ represents a valid orientation. Furthermore, we observe that the hardness proof above also works for *completely unimodal numberings* (CUN). We define these, in the spirit of [19], as bijective functions of the form $\chi : Q_n \to [0 : 2^n - 1]$, such that every face F of the hypercube has a unique local minimum vertex $x_F \in F$ (which means that x_F attains the minimum value of χ over $\mathcal{N}(x_F) \cap F$). The search problem with CUNs is to find the vertex that attains the value 0. These numberings have been extensively studied, see e.g. [9, 19]. Of course, we can represent χ by a succinct Boolean circuit C_χ with n input and n output bits, such that the output is the binary representation of an integer number. Then, the computational problem of deciding if a given circuit represents a CUN can be proved **coNP**-hard by slightly modifying the reduction above. Moreover, CUNs have short NO certificates (i.e. two vertices that are both local minima of the same face) and thus recognizing if a given circuit represents a CUN is **coNP**-complete. Note that CUNs induce AUSOs by directing every edge from the larger to smaller values [19]. However, as we will see in Section 5, recognizing AUSOs is **PSPACE**-complete.

4 Long Cycles in USO

In this section, we present the construction of a cyclic USO that has a unique cycle of superpolynomial size (number of involved vertices). This demonstrates that we cannot expect a **coNP** upper bound for cyclicity in USOs by listing the set of vertices that participate in a cycle. This intuition is verified in the next section with Theorem 12, where we prove that it is actually **PSPACE**-hard to decide the cyclicity of a USO. At first, we introduce a special class of USOs.

▶ **Definition 6.** Consider the family of orientations that arises when we start with $G_{\psi_{\mathbb{UF}}}$, choose a matching, and reverse the orientation of the edges of the matching. Call this *flip-matching* orientations (FMO).

Note that when we talk about FMOs in the construction below we mean the graph of the hypercube with the edges directed according to an FMO. Such orientations can be seen to be USOs, as a corollary of Lemma 3 [15]. This fact has also been shown by Matoušek, in [12], who used FMOs to provide $\left(\frac{n}{e}\right)^{2^n-1}$ as a lower bound on the number of distinct USOs.

In the following, we explain some notation regarding cycles in orientations of the hypercube. Let $x \in Q_n$. With $|x|$ we denote the Hamming weight of x, i.e. the number of ones in the bitstring x. Note that a forward (backward) edge increases (decreases) Hamming weight by 1. Let $\psi : Q_n \to Q_n$ be an orientation and consider $G_\psi = (Q_n, E_n)$. Let $c = \{v_1, \dots, v_k\} \subseteq Q_n$ be a k-cycle in G_ψ, that is a cycle over k vertices. Cycles are represented by the set of participating vertices, which we present in order of appearance; the last vertex in the sequence c has an outgoing edge to the first one.

Next, we observe that in an FMO every vertex that participates in a cycle must have an incident backward edge. Let $c \subseteq Q_n$ be a cycle in an FMO and let $v \in c$ be a vertex on the cycle. Assume that v has no backward edge attached. Let v' be the next vertex on c; we have that $|v'| = |v| + 1$ because the edge $v \to v'$ is forwards. The vertices that follow v' on the cycle have Hamming weight at least $|v|$, because a lower Hamming weight would imply that there are two consecutive backward edges, which is not allowed by our graph being an FMO. Then we conclude that v is reached with a forward edge from a vertex of Hamming weight at least $|v|$, which is of course not possible. We have proved the following.

▶ **Lemma 7.** *Let $G = (Q_n, E_n)$ be an FMO. Let $c \subseteq Q_n$ be a cycle in G. Then, every vertex in c has an incident backward edge. It follows that edges on c alternate between forwards and backwards and that reversing a backward edge cannot create any new cycles.*

Following, we describe our lower bound construction. It is an inductive construction that builds an FMO of dimension n from an FMO of dimension $n-3$. Note that in the resulting FMO we want exactly one cycle. For this we use Lemma 7 on the FMOs of dimension $n-3$, in order to turn their unique cycles into paths and construct an FMO of dimension n that contains a unique cycle. The base cases are FMOs that contain a unique cycle of size $2n$ for $n = 3, 4, 5$; those are easy to construct, as an example see the 3-dimensional cyclic FMO in Figure 1 (at least 3 dimensions are needed for a USO to be cyclic and 6 is the smallest size for a cycle in a USO).

Let $G^{n-3} = (Q_{n-3}, E_{n-3})$ be the resulting graph after the lth induction step. Let $c = \{v_1, v_2, \dots, v_k\}$ be the unique cycle, of size $6 \le |c| = k$, that is contained in G^{n-3}. We name the vertices $v_1, \dots, v_k$ in order of appearance on the cycle. We assume that the first edge of the cycle $(v_1, v_2) \in E_{n-3}$ is forwards. In our construction we will use three variants of G^{n-3} w.r.t. Lemma 7 (thus turning the cycle into a path):

1. G_1^{n-3} is derived from G^{n-3} by reversing the backward edge $(v_k, v_1) \in E_{n-3}$;
2. G_2^{n-3} is derived from G^{n-3} by reversing the backward edge $(v_{k-2}, v_{k-1}) \in E_{n-3}$;
3. G_3^{n-3} is derived from G^{n-3} by reversing all the backward edges except $(v_{k-2}, v_{k-1}), (v_k, v_1) \in E_{n-3}$.

Note that the three graphs above are all FMOs. We obtain each of these graphs by reversing edges that were backwards in G^{n-3} to forwards. The fact that the edges described in the first two items are backwards can be seen by Lemma 7.

Now we describe how to proceed with the induction at the $l+1$th step and eventually construct G^n. Consider the set of faces $\mathcal{F} = \{F_{[n-3], x \cdot 0^{n-3}} | x \in Q_3\}$. These are the faces that appear as ellipsoids in Figure 2. We embed the orientation $G_{\psi_{\mathrm{UF}}}^{n-3}$ in all the faces of $\mathcal{F}$, with three exceptions: In face $F_{[n-3],110\cdot 0^{n-3}}$ we embed G_1^{n-3}, in face $F_{[n-3],101\cdot 0^{n-3}}$ we embed G_2^{n-3} and in face $F_{[n-3],011\cdot 0^{n-3}}$ we embed G_3^{n-3}. The edges at the extra 3 coordinates follow the forward uniform orientation, except the following three edges that we orient backwards: $F_{\{n-2\},010\cdot v_2}$, $F_{\{n-1\},100\cdot v_k}$, $F_{\{n\},001\cdot v_{k-2}}$. See Figure 2 for an illustration of the cycle in G^n.

▶ **Theorem 8.** *There exist cyclic n-dimensional FMOs that contain a unique cycle of size $\Omega(2^{\frac{n}{3}})$.*

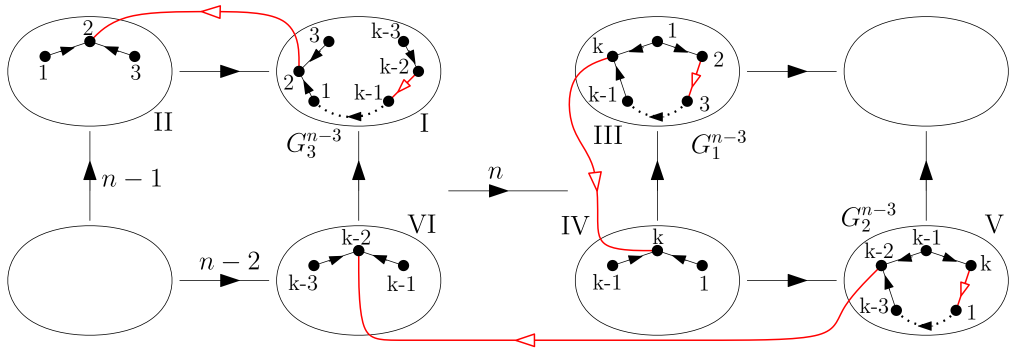


Figure 2 The ellipsoids represent the $(n-3)$-dimensional faces in $\mathcal{F}$. The black edges (with filled arrows) are forwards and the red edges (with non-filled arrows) are backwards. The dotted arcs represent a sequence of edges that starts with a forward one and ends with a backward one. Finally, we show only some of the vertices to illustrate the construction. A vertex with label i denotes v_i, the ith vertex on the cycle. The faces that contain the graphs $G_1^{n-3}, G_2^{n-3}, G_3^{n-3}$ are labeled. To see the cycle one can follow the Latin numbers I,...,VI.

Proof. Our construction, as we presented it above, satisfies the claimed theorem. Consider $G^n = (Q_n, E_n)$ and $c \subseteq Q_n$ the cycle in G^n. By induction, the faces where we embed the variants of G^{n-3} are FMOs. The other $(n-3)$-dimensional faces in $\mathcal{F}$ contain the uniform orientation. The three new coordinates also obey the uniform orientation except the three edges that we reversed. All of the reversed edges are incident to vertices that do not have other backward edges incident. Thus, G^n is an FMO.

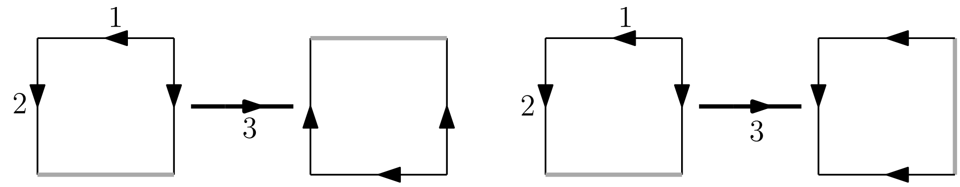
The existence of cycle c in G^n can be witnessed in Figure 2. Furthermore, there are no backward edges other than the ones that are incident to c. Thus, by Lemma 7, there is no other cycle and c is the unique cycle in G^n. Finally, let $C(n)$ denote the size of the cycle in our construction at dimension n. Then, we have the following recursive formula: $C(n) = 2C(n-3) + 6 = \Omega(2^{\frac{n}{3}})$ ◀

5 Recognizing Acyclic USOs

We start the section with the formal definitions of the two computational problems of interest:

- *AUSO-Accessibility*: The input is a Boolean circuit C_ψ, such that $\psi : Q_m \to Q_m$, and two vertices $s, t \in Q_m$. The answer to an instance is YES if and only if ψ represents an acyclic USO such that there is a directed path from s to t in G_ψ.
- *USO-Cyclicity*: The input is a Boolean circuit C_ψ, such that $\psi : Q_m \to Q_m$. The answer to an instance is YES if and only if ψ represents a USO such that there is a directed cycle in G_ψ.

Both these problems can be seen to be in **PSPACE**. Firstly, as we argued in Section 3, we can check if ψ represents a USO in **coNP**(the relationship **coNP**$\subseteq$ **PSPACE** is well-known). Then, the standard argument, that has been used to show that accessibility and cyclicity in directed graphs given by succinct representations, are in **PSPACE** (see e.g. [1, 13]) suffices in our case too. By this argument, we decide the existence of a cycle in the following way: we fix a vertex (non-deterministically) and pick the next vertex from the set of neighbors that can be accessed by an outgoing edge (also non-deterministically). If we reach the same vertex then we conclude that there is a directed cycle (formally here what we need to decide is the non-existence of a cycle; we are using the fact that all deterministic classes are closed



(a) Base case for q_1 is $\exists$. **(b)** Base case for q_1 is $\forall$.

Figure 3 An illustration of the two base cases of the inductive construction.

under complement). Similarly, to decide the existence of an $s - t$ path, we fix vertex s and perform the same process; if we reach t then we conclude that there is an $s - t$ path. These processes use only polynomial space (actually linear, only one vertex needs to be stored in memory) and they give non-deterministic **PSPACE** upper bounds, which is the same as deterministic **PSPACE** by Savitch's Theorem [14].

We are ready to present our first theorem which shows that it is **PSPACE**-hard, and thus by the above argument **PSPACE**-complete, to decide the problem AUSO-Accessibility.

▶ **Theorem 9.** *AUSO-Accessibility is **PSPACE**-complete.*

The proof is by reduction from the problem of deciding the satisfiability of a Quantified Boolean Formula (QBF) which is the standard **PSPACE**-complete problem. The input to the latter is a CNF formula Φ with n variables $v_1, \ldots, v_n$ and a set of n quantifiers $q_1, \ldots, q_n$ that can be either $\exists$ or $\forall$. The construction is presented in an inductive fashion, where the induction is on the number of variables of the QBF formula. The base case is a 3-dimensional acyclic USO and then for each variable we add 3 coordinates when the next quantifier is existential and 4 coordinates when it is universal. All in all, the result of the construction is $\psi : Q_m \to Q_m$ which represents an acyclic USO and such that $m \leq 4(n-1) + 3 = 4n - 1$. For this purpose, we describe the construction of G_ψ; then, the question to be decided is if there exists a directed path from 0^m to 1^m.

We have a set of vertices, called *active* and denoted with $\mathcal{AV} \subset Q_m$. We call an edge $F_{\{i\},x}$ active when $F_{\{i\},x} \subset \mathcal{AV}$. We denote with gray color the active edges in the illustrations for the base case (cf. Figure 3) and the faces that contain active edges in the illustrations for the inductive steps (cf. Figure 4). With $\mathcal{AV}^l$ we denote the set of active vertices after the lth inductive step. The size of $\mathcal{AV}$ is 4 for the base case and it triples at each induction step ($|\mathcal{AV}^l| = 3|\mathcal{AV}^{l-1}|$). The orientations of the active edges depend on an evaluation of Φ for a given assignment that can be obtained by the coordinates of the active vertices. This process will be explained at a later step. We are ready now to describe our construction. The 3-dimensional base cases are presented in Figure 3.

Let $G^l = (Q_k, E_k)$, with $k < 4l$, be the graph after the lth induction step and let q_{l+1} be $\exists$. We introduce three extra coordinates. At coordinate $(k+1)$ and $(k+2)$ all edges are forwards. At coordinate $(k+3)$ all edges are backwards except the edges $F_{\{k+3\},000\cdot 1^k}$ and $F_{\{k+3\},010\cdot 0^k}$ which are reversed. Then, we embed G^l in the faces $\mathcal{F}_0 = F_{[k],0^{k+3}}$, $\mathcal{F}'_0 = F_{[k],1\cdot 0^{k+2}}$ and $\mathcal{F}_1 = F_{[k],111\cdot 0^k}$. The rest of the faces in $\mathcal{F}^\exists = \{F_{[k-3],y\cdot 0^{k-3}} | y \in Q_3\}$ are all oriented according to $\psi_{\mathbb{UB}}$ (backwards uniform) in the first k coordinates (cf. Figure 4a).

For the other case, let q_{l+1} be $\forall$. Introduce four extra coordinates. At coordinate $(k+1)$ we have that the edges in face $F_{[k+2],0^{k+4}}$ are backwards and every other edge is forwards. At coordinate $(k+2)$ all edges are backwards except edge $F_{\{k+2\},0000\cdot 1^k}$ which is reversed. At coordinate $(k+3)$ all edges are forwards. At coordinate $(k+4)$ all edges are backwards except the edge $F_{\{k+4\},0110\cdot 0^k}$ which is reversed. Then, we embed G^l in the faces $\mathcal{F}_0 = F_{[k],0^{k+4}}$, $\mathcal{F}'_0 = F_{[k],0010\cdot 0^k}$ and $\mathcal{F}_1 = F_{[k],1111\cdot 0^k}$. The rest of the faces in $\mathcal{F}^\forall = \{F_{[k-4],y\cdot 0^{k-4}} | y \in Q_4\}$ are all oriented according to $\psi_{\mathbb{UB}}$ in the first k coordinates (cf. Figure 4b).

The graph $G^n = (Q_m, E_m)$ is the end product of our reduction (after the nth induction step). Note that G^n is not an FMO and neither will be the graph we construct in the proof of the next theorem. We still have to describe the orientation of the active edges in G^n. Let $v \in \mathcal{AV}$. The orientation of the active edge adjacent to v, say $e \in E_m$, is decided by the following simple algorithm:

- Let $x \in Q_n$ be the assignment for the variables of the input QBF which we build based on the coordinates of v. Initialize $j = 3$ and $x_1 = v_j$.
- For $i = 2$ to n repeat:
  - If q_i is $\exists$ then set $j \leftarrow j + 3$ and $x_i \leftarrow v_{j-1}$.
  - If q_i is $\forall$ then set $j \leftarrow j + 4$ and $x_i \leftarrow v_j$.
- If $\Phi(x) = 1$ then e is forwards, otherwise it is backwards.

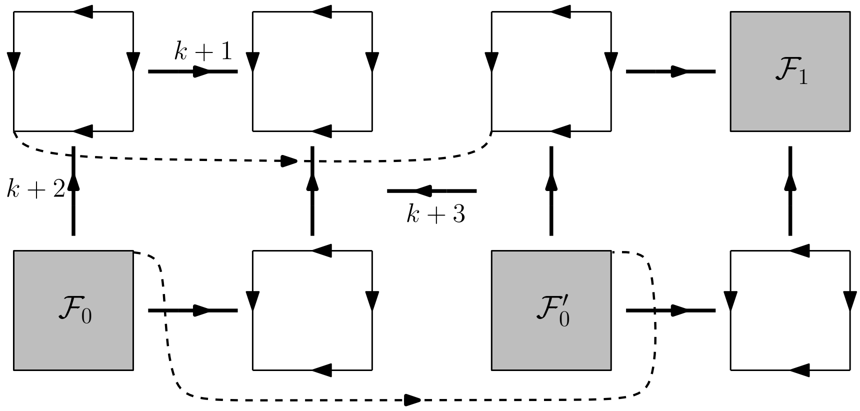
For example, consider the following simple QBF: $\Phi = (v_1 \vee v_2 \vee v_3)$, $q_1 = q_3 = \exists$ and $q_2 = \forall$. This gives rise to a USO over Q_{10}. We give the vertex $v = 1\mathbf{0}0\mathbf{1}111\mathbf{0}00$ as input to the algorithm above (the bold bits are the ones that the algorithm will extract). This is translated to the 3-length bitstring $x = 010$ which means that variable v_2 is set to true and the other two to false and thus $\Phi(x) = 1$ and the corresponding active edge is forwards.

▶ Claim 10. G^n is an acyclic USO.

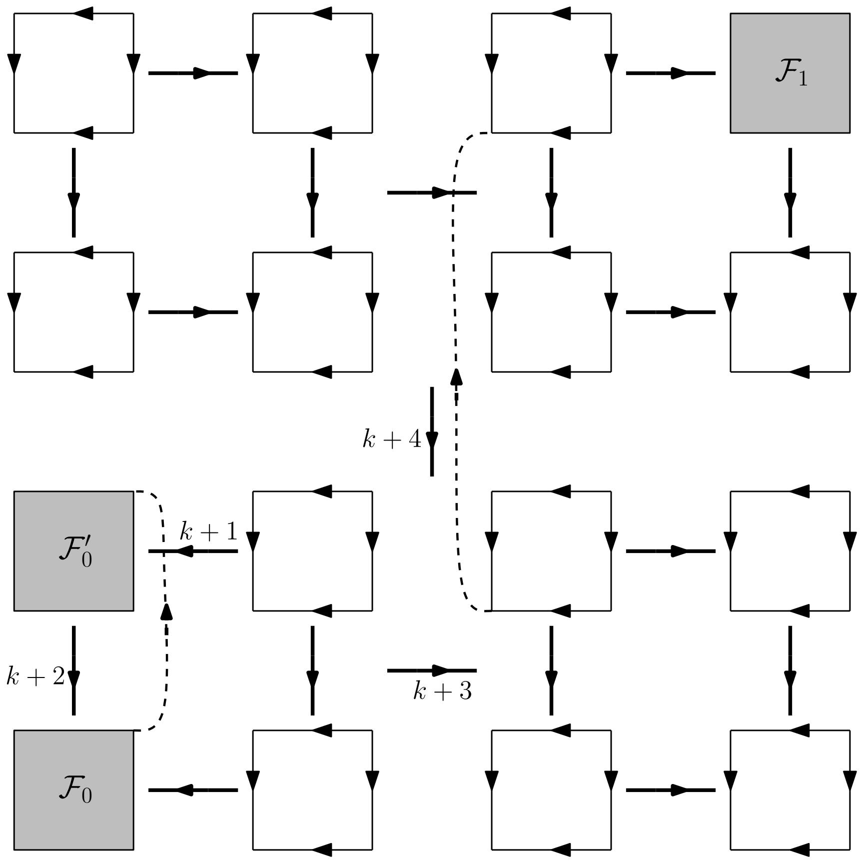
Proof. It can be seen by Lemma 3 that both base cases of the construction are 3-dimensional USOs, regardless of the orientation of the active edges. In addition, they are acyclic because at coordinate 3 every edge is forwards. Then, we argue that for every step of the induction the graph remains an acyclic USO. Consider the $l+1$th step of the induction and let q_{l+1} be $\exists$. Moreover, let $G^{l+1} = (Q_{k+3}, E_{k+3})$ and consider $\mathcal{F}^\exists$. It holds that every face in $\mathcal{F}^\exists$ is an acyclic USO: for $\mathcal{F}_0, \mathcal{F}'_0, \mathcal{F}_1$ it holds by induction and in every other face we have the backwards uniform orientation.

We interpret the construction in two steps: First, the faces in $F^\exists$ are put on a 3-dimensional acyclic USO whose orientation is defined above by the orientation of the extra coordinates $(k+1, k+2, k+3)$, before reversing the edges $F_{\{k+3\},000\cdot 1^k}$ and $F_{\{k+3\},010\cdot 0^k}$. The result is an acyclic USO by Lemma 2. In the next step we reverse the aforementioned edges. The result is a USO by Lemma 3. Moreover, reversing these edges does not create any cycles. The orientation in the face $F_{[k]\cup\{k+3\},0^{k+3}}$ remains acyclic after reversing $F_{\{k+3\},000\cdot 1^k}$ because the orientations in faces $\mathcal{F}_0$ and $\mathcal{F}'_0$ are identical (and a cycle in the former face would imply that the latter faces are cyclic; this is the reason we orient $\mathcal{F}'_0$ this way). A similar argument applies to reversing the edge $F_{\{k+3\},010\cdot 0^k}$ within the face $F_{[k]\cup\{k+3\},010\cdot 0^k}$. All the edges at coordinates $k+1$ and $k+2$ are forwards and thus a cycle can only involve the $k+3$th coordinate, which is not possible by the arguments above.

The situation is symmetrical when q^{l+1} is $\forall$ and $G^{l+1} = (Q_{k+4}, E_{k+4})$. First, we argue about the set of faces $\mathcal{F}^\forall$. Then, why reversing the edge $F_{\{k+2\},0000\cdot 1^k}$ does not create any cycles within the face $F_{[k]\cup\{k+2\},0^{k+4}}$ and reversing the edge $F_{\{k+4\},0110\cdot 0^k}$ does not create any cycles within the face $F_{[k]\cup\{k+4\},0110\cdot 0^k}$. The argument is exactly the same as above. Remember that at the $k+2$th coordinate all edges are backwards and at the $k+3$th all edges are forwards. Then, we conclude that reversing the edge $F_{\{k+2\},0000\cdot 1^k}$ does not create any cycle in G^{l+1}, since all the edges at the $k+1$th and the $k+4$th coordinate that are incident to the face $F_{[k],0^{k+4}}$ are backwards. Furthermore, we conclude that reversing the



(a) G^{l+1} with q_{l+1} is $\exists$. The k-dimensional faces in $\mathcal{F}^{\exists}$ appear as 2-faces here.



(b) G^{l+1} with q_{l+1} is $\forall$. The k-dimensional faces in $\mathcal{F}^{\forall}$ appear as 2-faces here.

Figure 4 An illustration of the steps of the inductive construction. The active faces (faces that contain active edges) are filled with gray color. The reversed edges are depicted as dashed.

edge $F_{\{k+4\},0110\cdot 0^k}$ does not create any cycle in G^{l+1}, since in the face $F_{[k+4]\setminus\{k+3\},0100\cdot 0^k}$ all edges at the $k+1$th coordinate are forwards and at the $k+2$th are backwards. ◀

▶ Claim 11. 0^m is connected to 1^m in G^n if and only if the input QBF is satisfiable.

Proof. First, note that at the base case 0^3 is connected to 1^3 if and only if there is at least one forward active edge in the case q_1 is $\exists$ and if and only if both active edges are forwards in the case q_1 is $\forall$.

Then, consider the $l+1$th step of the induction and let the quantifier q_{l+1} be $\exists$ and $G^l = (Q_{k+3}, E_{k+3})$. There is a directed path from 0^{k+3} to 1^{k+3} if and only if at least one of the following is true: Either there is a directed path from 0^{k+3} to $000 \cdot 1^k$ or there is a directed path from $111 \cdot 0^k$ to 1^{k+3}. This is because the $k+3$th coordinate is directed backwards for all edges except the two we reversed during the construction ($F_{\{k+3\},000\cdot 1^k}$ and $F_{\{k+3\},010\cdot 0^k}$). If there is a directed path from 0^{k+3} to $000 \cdot 1^k$, then there is one from 0^{k+3} to 1^{k+3} through the edge $F_{\{k+3\},000\cdot 1^k}$. Otherwise, there is an edge from any vertex $x^1 \in \mathcal{F}_0$ to a vertex $x^2 \in F_{[k],010\cdot 0^k} \cap \mathcal{N}(x^1)$, from there to the vertex $010 \cdot 0^k$ and finally to $111 \cdot 0^k$ through edge $F_{\{k+3\},010\cdot 0^k}$. Thus, if there is a path from $111 \cdot 0^k$ to 1^{k+3} then there is a path from 0^{k+3} to 1^{k+3}.

Following, we consider the case that q_{l+1} is $\forall$ and $G^l = (Q_{k+4}, E_{k+4})$. Then, there is a directed path from 0^{k+4} to 1^{k+4} if and only if there is a directed path from 0^{k+4} to $0000 \cdot 1^k$ and one from $1111 \cdot 0^k$ to 1^{k+4}. Note at the $k+4$th coordinate all edges are backwards, except $F_{\{k+4\},0110\cdot 0^k}$ and thus a path from 0^{k+4} to 1^{k+4} has to go through vertex $0110 \cdot 0^k$. The only way this is possible is if there is a path from 0^{k+4} to $0000 \cdot 1^k$ and from there through the edge $F_{\{k+2\},0000\cdot 1^k}$ to face $\mathcal{F}_0'$ and, finally, from there a path to face $F_{[k],0110\cdot 0^k}$. From the latter the vertex $1110 \cdot 0^k$ is accessible and finally the vertex $1111 \cdot 0^k$. A directed path from $1111 \cdot 0^k$ to 1^{k+4} completes the path from 0^{k+4} to 1^{k+4}.

Thus, we have shown the existence of which paths is mandatory, for the existence of a directed path from the all-zero to the all-one vertex in both cases. It remains to explain that these paths exist if and only if the input QBF is satisfiable.

For the forward case of the claim assume that the input QBF is satisfiable. Then, there exists an assignment of the variables of Φ whose quantifier is existential such that, for any assignment of the rest of the variables, Φ is satisfiable. This means that for every step of the induction that corresponds to an existential quantifier there exists a directed path either from 0^{k+3} to $000 \cdot 1^k$ in $\mathcal{F}_0$ or from $111 \cdot 0^k$ to 1^{k+3} in $\mathcal{F}_1$ (since the corresponding active edges are forwards). For an inductive step that corresponds to a universal quantifier we have that there are directed paths both from 0^{k+4} to $0000 \cdot 1^k$ in $\mathcal{F}_0$ and from $1111 \cdot 0^k$ to 1^{k+4} in $\mathcal{F}_1$. By the inductive construction this means there is a directed path in G^n from 0^m to 1^m.

Reversely, assume that there is a directed path from the vertex 0^m to the vertex 1^m in G^n. Again, by inductive reasoning. If q_{l+1} is $\exists$ then there is a path in at least one of $\mathcal{F}_0$ and $\mathcal{F}_1$; this means that Φ is satisfiable for at least one of the two possible assignments. If q_{l+1} is $\forall$ then there are both the paths in $\mathcal{F}_0$ and $\mathcal{F}_1$; this means that Φ is satisfiable for both possible assignments. ◀

In the next step we prove that USO-Cyclicity is **PSPACE**-hard based on the **PSPACE**-hardness of AUSO-Accessibility. This implies **PSPACE**-completeness by the arguments we gave in the beginning of this section.

▶ **Theorem 12.** *USO-Cyclicity is **PSPACE**-complete.*

Proof. The reduction is from QBF. In a first step, we reduce to an instance of AUSO-Accessibility as in the proof of Theorem 9. Then we have G_ψ which is an AUSO (we use this

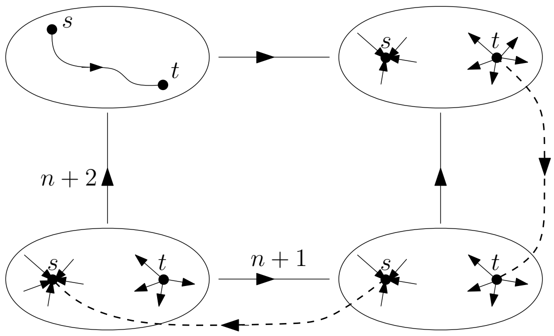


Figure 5 An illustration of the construction. The ellipsoids represent n-dimensional USOs. The face $F_{[n],10\cdot 0^n}$ contains the orientation of the AUSO-Accessibility instance.

trick since the formal definition of AUSO-Accessibility does not guarantee that the input to the problem represents an AUSO). Based on G_ψ we define $G_{\psi'}$, where $\psi' : Q_{n+2} \to Q_{n+2}$. All the edges at coordinates $n+1$ and $n+2$ are forwards except $F_{\{n+1\},00\cdot s}$ and $F_{\{n+2\},01\cdot t}$ which are reversed. We have now defined the orientation of the edges at the two extra coordinates and we turn our attention to the first n ones. In face $F_{[n],10\cdot 0^n}$ we embed G_ψ which is an AUSO. Let $G_{\psi''}$ be the AUSO graph that results by applying Lemma 4 with $\psi''(s) = 0^n$ and $\psi''(t) = 1^n$. We embed $G_{\psi''}$ in faces $F_{[n],00\cdot 0^n}$, $F_{[n],01\cdot 0^n}$ and $F_{[n],11\cdot 0^n}$. It follows that $G_{\psi'}$ is a USO from the above argument and the fact that reversing the edges is safe by Lemma 3. An illustration of the construction can be found in Figure 5.

▶ Claim 13. There is a cycle in $G_{\psi'}$ if and only if there is a directed path from s to t in G_ψ.

By construction, there is a path from vertex $10 \cdot t$ to $01 \cdot t$ through $11 \cdot t$. Note that since $11 \cdot t$ is the source of the face $F_{[n],11\cdot 0^n}$ a path from any vertex of the face $F_{[n],10\cdot 0^n}$ to $11 \cdot t$ has to go through vertex $10 \cdot t$. In $F_{[n],01\cdot 0^n}$ there is path from $01 \cdot t$ (which is the source of the face) to $01 \cdot s$ (which is the sink of the face). From the latter there is a path to vertex $00 \cdot s$ (which is the sink of the face $F_{[n],00\cdot 0^n}$ and thus no other vertex of the same face is accessible from it) and finally to $10 \cdot s$. In addition, note that the desired cycle is the only one that will use both coordinates $n+1$ and $n+2$ (if it exists). There is no other cycle that involves only one of the two extra coordinates. This is because the existence of such a cycle would imply that the orientations embedded in $F_{[n],00\cdot 0^n}$, $F_{[n],01\cdot 0^n}$ and $F_{[n],11\cdot 0^n}$ are cyclic (we have also seen this argument in the proof of Claim 10). The claim follows. ◀

The reductions described in this section give as a result a directed graph. However, the graph of the hypercube is obviously of exponential size and we are interested in a Boolean circuit that succinctly describes it. As we have already argued, this is done by actually describing the outmap of the USO. The size of such a circuit depends only on n, the number of variables of the QBF. It is discussed in [1] that the techniques used by Ladner in [10] can be used to construct such a circuit in polynomial time. Nonetheless, in our case, and because the graph is very structured, it is not too hard to explicitly describe the construction of the actual circuits for Theorems 9 and 12. The description is a bit tedious and, due to the lack of space, we postpone it to the full version of the current paper. We remark that for the proof of Theorem 9 the circuit contains internally another circuit that, given an assignment x of the n variables of Φ, returns the evaluation $\Phi(x)$. The latter is used in the algorithm described in the proof of Theorem 9 to decide the orientation of the active edges. It is known that such evaluations can be performed in polynomial time (see e.g. [11]) and thus such a circuit is easy to obtain.

References

1 José L. Balcázar, Antoni Lozano, and Jacobo Torán. The complexity of algorithmic problems on succinct instances. In Ricardo Baeza-Yates and Udi Manber, editors, *Computer Science*, pages 351–377. Springer US, 1992.

2 Gregory E. Coxson. The P-matrix problem is co-NP-complete. *Mathematical Programming*, 64(1-3):173–178, 1994.

3 Kaspar Fischer and Bernd Gärtner. The smallest enclosing ball of balls: combinatorial structure and algorithms. *Internat. J. Comput. Geom. Appl.*, 14(4-5):341–378, 2004.

4 Jan Foniok, Bernd Gärtner, Lorenz Klaus, and Markus Sprecher. Counting Unique-Sink Orientations. *Discrete Applied Mathematics*, 163, Part 2:155–164, 2014.

5 Hana Galperin and Avi Wigderson. Succinct representations of graphs. *Information and Control*, 56(3):183–198, 1983.

6 Bernd Gärtner. The Random-Facet simplex algorithm on combinatorial cubes. *Random Structures & Algorithms*, 20(3):353–381, 2002.

7 Bernd Gärtner and Leo Rüst. Simple stochastic games and P-matrix generalized linear complementarity problems. In Maciej Liskiewicz and Rüdiger Reischuk, editors, *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory (FCT'05)*, volume 3623 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2005.

8 Bernd Gärtner and Ingo Schurr. Linear Programming and Unique Sink Orientations. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 749–757, 2006.

9 Peter L. Hammer, Bruno Simeone, Thomas M. Liebling, and Dominique de Werra. From linear separability to unimodality: A hierarchy of pseudo-Boolean functions. *SIAM J. Discrete Math.*, 1(2):174–184, 1988.

10 Richard E. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7(1):18–20, 1975.

11 Nancy Lynch. Log space recognition and translation of parenthesis languages. *J. ACM*, 24(4):583–590, 1977.

12 Jiří Matoušek. The number of Unique-Sink Orientations of the hypercube*. *Combinatorica*, 26(1):91–99, 2006.

13 Christos H. Papadimitriou and Mihalis Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, 1986.

14 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970.

15 Ingo Schurr and Tibor 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.

16 Alan Stickney and Layne Watson. Digraph models of Bard-type algorithms for the linear complementarity problem. *Math. Oper. Res.*, 3(4):322–333, 1978.

17 Tibor Szabó and Emo Welzl. Unique Sink Orientations of cubes. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 547–555, 2001.

18 Stefano Tessaro. Randomized algorithms to locate the sink in low dimensional Unique Sink Orientations of cubes. Semester thesis, Computer Science Department, ETH Zürich, 2004.

19 Kathy Williamson-Hoke. Completely unimodal numberings of a simple polytope. *Discrete Applied Mathematics*, 20(1):69–81, 1988.