Abstract

We introduce unique sink orientations of grids as digraph models for many well-studied problems, including linear programming over products of simplices and generalized linear complementarity problems over P-matrices (PGLCP). We investigate the combinatorial structure of such orientations and develop randomized algorithms for finding the sink. We show that the orientations arising from PGLCP satisfy the Holt-Klee condition known to hold for polytope digraphs, and we give the first expected linear-time algorithms for solving PGLCP with a fixed number of blocks.
1 Introduction

A grid is a graph whose vertex set is the Cartesian product of \( n \) finite sets, with edges joining all pairs of vertices that differ in exactly one component, see Figure 1 (a).

If all sets have size two, we get the graph of the \( n \)-cube. A face or subgrid is any induced subgraph spanned by the Cartesian product of subsets of the original sets. An orientation \( \psi \) of the grid is called unique sink orientation (USO) if any nonempty face has a unique sink with respect to \( \psi \). Figure 1 (b) depicts a USO of the \( (3 \times 2 \times 2) \)-grid. In particular, the grid itself must have a unique sink. Grid USO may contain directed cycles, as the 3-cube in Figure 2 shows.

The significance of USO on grids comes from the fact that they form a simple combinatorial framework subsuming a number of well-studied problems. We show in this paper that the problem of solving a generalized linear complementarity problem over a P-matrix (PGLCP), as introduced by Cottle and Dantzig [8], can be recast as the problem of finding the unique sink of an implicitly given grid USO. As special cases, this includes standard linear complementarity problems over P-matrices (PLCP) [9], linear programming (LP) over products of simplices, and LP over combinatorial cubes. In the LP applications, we get acyclic unique sink orientations (AUSO).

Two major open problems motivate our research. On the one hand, it is unknown whether polynomial-time algorithms exist for PLCP or PGLCP, even though both problems are unlikely to be NP-hard. Megiddo has shown that hardness of PLCP would
imply NP = co-NP [35], and his proof extends to PGLCP easily. LP, on the other hand, is solvable in polynomial time (a celebrated result of Khachyian [30]), but a strongly polynomial algorithm is not known, even if we are dealing only with LP over combinatorial cubes. Candidates for strongly polynomial algorithms must be combinatorial in the sense that the number of arithmetic operations they perform depends only on the combinatorial structure of the LP but not on the actual numbers that encode it. The AUSO approach attempts to extract the combinatorial structure behind LP; in this paper, we generalize to the combinatorics of PGLCP. A polynomial-time algorithm for finding the sink of a grid USO (using an oracle that returns the orientation of a given edge) would solve both problems in strongly polynomial time. The USO generalization reveals some (algorithmically useful) hidden structure, leading to new results for PGLCP. Ultimately, results obtained along these lines may help to resolve the (combinatorial) complexities of LP and PGLCP.

The AUSO framework also covers a generalization of LP resulting from the replacement of linear objective functions with abstract objective functions (AOF) [1, 28], or completely unimodal numberings of vertices [46, 25]. On general polytopes, these concepts are dual to the notion of shellings [25], and they have successfully been applied to the theory of polytope (di)graphs and linear programming [29, 27].

The case of grid AUSO—equivalently, AOF on products of simplices—has been treated in detail by Björklund et al. as a combinatorial framework for the problem of computing optimal infinite game strategies [2, 3]. The games considered include parity, mean-payoff, and simple stochastic games. Whether parity games (the easiest among the three) can be solved in polynomial time is an important open question [24].

In the planar case (n = 2), the combinatorial and algorithmic properties of grid AUSO have been examined by Tschirschnitz et. al. [44, 14].

Unique sink orientations of cubes (not necessarily acyclic) were first considered by Stickney and Watson as digraph models for PLCP [40]. Most remarkably, Szabó and Welzl gave algorithms for finding the sink of an n-cube USO by looking at only $O(c^n)$ vertices and edges, for some $c$ strictly smaller than 2 [42]. This in particular yields the first combinatorial algorithms for PLCP with nontrivial runtime bounds. Cube USO are also useful as combinatorial models for geometric optimization problems, in particular the problem of computing the smallest enclosing ball of a set of points [42, 18], or a set of balls [15]. In the point case, this problem can be reduced to PLCP with positive definite matrix [18], but the ball case does not fit into any known class of cube USO. In both cases, cyclic orientations may arise.

Unique sink orientations of the graphs of general polytopes are dual to exact signings studied by Kleinschmidt and Onn [32].

**Statement of Results.**

In this paper, we generalize results known to hold in some of the above special cases, and we prove new structural and algorithmic results of particular significance for the theory of PLCP and PGLCP. Probably the most surprising insight is that all these results hold even under the presence of directed cycles in the orientation.
**Inherited USO and Refined Index.** A *hypervertex* is a face consisting of all vertices whose coordinates agree with a given vertex on some fixed subset $I$ of the $n$ grid dimensions. An $(n - |I|)$-dimensional *inherited grid* can be imposed on the set of hypervertices defined by $I$ in a natural way. Given a USO of the grid, the *inherited orientation* of the inherited grid is obtained from the orientations of the outgoing edges incident to the sinks of the hypervertices. We prove that the inherited orientation is USO again. Figure 3 shows an example.

![Figure 3](image1.png)

Figure 3: Two-dimensional inherited USO over one-dimensional hypervertices (shaded) in a three-dimensional grid. The inherited USO is defined through the arcs incident to the hypervertex sinks (white); in this case, we obtain the inherited USO of Figure 4

While this result is an easy generalization of a corresponding result for cubes [42], its implications are more interesting. Based on the inherited structure, we are able to prove that the *refined index* is a bijection for any grid USO. With respect to a given USO, the refined index maps every vertex to an $n$-tuple of natural numbers, the entry at position $i$ corresponding to the number of outgoing edges in dimension $i$ of the grid. Figure 4 depicts a USO along with all refined index values.

![Figure 4](image2.png)

Figure 4: The refined index. For each vertex, the two components correspond to the outgoing edges in horizontal and vertical direction

This bijection was previously known to hold for cubical grids (in which case it easily follows from the USO property [42]), and for acyclic grids [2], with a proof that does not extend to USO with cycles. An earlier proof for the planar case is due to Tschirschnitz.
For USO coming from LP over a product of simplices, or from a PGLCP, the bijection can be established using simple geometric arguments. In the planar setting, it is the basis for a tight analysis of the RANDOMEDGE simplex algorithm [19].

Our inductive proof of the bijection for grid USO differs fundamentally from the proofs of the special cases. The reason is that the inherited USO— which we use in our proof— will in general not come from PGLCP, even if the original grid USO does (we give a concrete example for this). Here, the generalization to USO allows for new approaches that are not necessarily viable in the concrete setting.

**Algorithms.** Given an oracle that returns the orientation of any given edge in a grid USO, it is natural to ask how many oracle calls are required in order to find the global sink of the USO. In all concrete instances, this oracle can easily be implemented in polynomial time, meaning that the number of oracle calls is a good measure of complexity. We develop two simple randomized algorithms whose expected number of calls to the oracle is of the order $f(n)N$, with $N$ being the sum of sizes of the $n$ sets whose product forms the grid, and with $f(n) \approx n!$.

The two algorithms are very similar in spirit; the first one generalizes an algorithm for the $n$-cube [21, 42], and its sequence of vertices ‘jumps around’ on the grid. The expected runtime only depends on the parameters of the grid, but not on the concrete USO. The second algorithm follows a directed path to the global sink. Compared to the first algorithm, its expected worst-case runtime is slightly higher, but it has the potential of being much faster if the starting vertex is already close to the sink, or if directed paths tend to be short in the USO under consideration.

In particular, if $n$ is fixed, we get linear-time algorithms. Specialized to PGLCP, this corresponds to the case in which we have a fixed number of blocks; in this situation, we get the first algorithms whose expected complexity is linear in the number of variables—this is optimal. Even if $n$ is not considered to be fixed, our algorithms improve over the trivial bound resulting from the evaluation of all edge orientations. To the best of our knowledge, no previously published algorithm could be proven to perform substantially better than the trivial approach, in the worst case. A reduction from PGLCP to PLCP was claimed that would yield a quadratic bound of $O(N^2 + \text{plcp}(n))$, where plcp$(n)$ is the time necessary to solve a PLCP instance of size $n$ [11]. However, the proof contains an error, and there is a counterexample to the correctness of the reduction [12].

In the acyclic case, linear-time algorithms for fixed $n$ are known. This follows from the fact that the problem of finding the sink in an AUSO can be formulated as an LP-type problem [33] in a natural way [2]. In the LP-type framework, a number of $f(n)N$ oracle calls suffices, even deterministically [5]. The currently best algorithm for the acyclic case combines two randomized algorithms [33, 7] and requires an expected number of $O(Nn + f(n))$ oracle calls, where $f(n) = \exp(O(\sqrt{n \log n}))$ is a subexponential function [20]. No subexponential methods are known for USO that contain cycles.

**The Holt-Klee Condition.** Unique sink orientations coming from LP over products of simplices satisfy an interesting geometric property: consider any subgrid along with
its unique sink and unique source.\footnote{The existence of a unique source follows from the USO axioms, see Section 2.2.} The Holt-Klee (HK) condition states that there are as many vertex-disjoint directed paths between source and sink as there are neighbors of the source (equivalently, the sink) in the subgrid [26]. This is obviously the maximum possible number of such paths. The HK condition does not hold for general AUSO; there are two nonequivalent AUSO of the 3-cube with only two vertex-disjoint directed paths from source to sink, see Figure 5.

![Figure 5: Two AUSO violating the Holt-Klee condition](image)

The HK condition is involved in a prime example of how combinatorial information can be exploited algorithmically. Matoušek has constructed a class of \( n \)-cube AUSO with the property that the RANDOMFACET simplex algorithm [28] has superpolynomial expected runtime on a randomly chosen instance from the class [34]. This runtime is of the order \( \exp(\Omega(\sqrt{n})) \) and matches the best known upper bound of \( \exp(O(\sqrt{n})) \) for the expected performance of this algorithm on general \( n \)-cube AUSO [16]. While this result settles the complexity of RANDOMFACET on \( n \)-cube AUSO, the question remained...
open whether the performance analysis can be improved when the AUSO is induced by an actual LP over a combinatorial cube. At least within Matoušek’s class, this is the case. It can be shown that RandomFacet needs an expected number of only $O(n^2)$ pivot steps on all $n$-cube AUSO in the class that satisfy the HK condition [16]. It is open whether this result extends to larger classes of AUSO.

A general grid USO in dimension $n = 2$ is known to be HK if and only if no subgrid orientation is isomorphic to the orientation of Figure 4 which we call the double twist [44, 14]. We extend this result to $n = 3$, proving that in order to satisfy the HK condition, it is necessary and sufficient to avoid the double twist as well as the two cubes of Figure 5. For $n = 4$, we give an example of a minimal grid that violates HK but is not a cube. We do not know if the set of minimal non-HK grids is finite in grid dimension $n \geq 4$.

The Planar Case. We show that no directed cycles can occur in a grid USO of dimension $n = 2$, meaning that the cube of Figure 2 is in fact the smallest possible cyclic example. If the orientation is HK, this follows from the interpretation in terms of a pseudoline arrangement [14]. Specialized to PGLCP with two blocks, this shows that cycling is impossible in Bard-type pivoting algorithms.

2 Basics

Throughout this paper, we fix two natural numbers $N \geq n \geq 1$ and an ordered partition

$$\Pi = (\Pi_1, \ldots, \Pi_n)$$

of the set $[N] := \{1, \ldots, N\}$ into $n$ nonempty subsets. We also refer to $\Pi_i$ as the block $i$.

A subset $J \subseteq [N]$ is called a $\Pi$-vertex (or simply vertex) if $|J \cap \Pi_i| = 1$ for all $i$. We also set

$$J_I = \bigcup_{i \in I} (J \cap \Pi_i), \quad I \subseteq [n].$$

Let $V = V(\Pi)$ be the set of vertices.

The grid spanned by $S \subseteq [N]$ is the undirected graph

$$G(S) = (V(S), E(S)),$$

with

$$V(S) := \{J \in V \mid J \subseteq S\}$$

$$E(S) := \{\{J, J'\} \subseteq V(S) \mid |J \oplus J'| = 2\},$$

where $\oplus$ denotes symmetric difference. The vertices of $G(S)$ canonically correspond to the elements of the Cartesian product

$$\prod_{i=1}^n S_i, \quad S_i := S \cap \Pi_i.$$
(Note that $J_i = J_{i+1}$.) Edges join pairs of vertices $J, J'$ that differ in exactly one coordinate.

If $|S_i| \leq 2$ for all $i$, $G(S)$ is the graph of some $k$-cube, $k \leq n$. If $S_i = \emptyset$ for some $i$, $G(S)$ is the empty graph.

A face or subgrid of $G(S)$ is any graph of the form $G(S')$, for $S' \subseteq S$. Throughout, we abbreviate $G([N])$ as $G$ and $E([N])$ as $E$.

Definition 2.1 Let $\psi$ be an orientation of $G$. $\psi$ is called unique sink orientation (USO) if all subgraphs of $\psi$ induced by nonempty faces of $G$ have unique sinks. If $\psi$ contains the arc $(J, J')$, we also write $J \xrightarrow{\psi} J'$.

In the remainder of this section, we introduce PGLCP as a generalization of LP over products of simplices, and we prove that any generic PGLCP instance gives rise to a USO. Subsequently, we prove further properties of general grid USO.

2.1 Grid LP and Generalized LCP

Consider a linear program in the variables $x = (x_1, \ldots, x_N)^T$, of the form

$$\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b, \\
& \quad x \geq 0,
\end{align*}$$

(1)

where $A \in \mathbb{R}^{n \times N}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^N$. For $S \subseteq [N]$, let $A_S$ denote the submatrix of columns indexed by $S$. Furthermore assume that every vertex $J \in V$ is a nondegenerate basis in (1), meaning that $A_J^{-1}b > 0$. We say that the LP is $\Pi$-compatible, and we call $A_J$ a representative submatrix. We will assume that the ordering of the columns in $A_J$ is compatible with $\Pi$, meaning that the $i$-th column of $A_J$ comes from block $\Pi_i, i \in [n]$.

Using Cramer’s rule, the following is then easily established.

Observation 2.2 Consider a $\Pi$-compatible LP of the form (1). Then

(i) all determinants $\det(A_J)$ of representative submatrices have the same (nonzero) sign, and

(ii) the vertices $J$ are the only bases of the LP (1).

For any $c \in \mathbb{R}^N$, a canonical $\Pi$-compatible LP is obtained by setting $b_i = 1$ and

$$A_{ij} := \begin{cases} 
1, & j \in \Pi_i \\
0, & \text{otherwise}
\end{cases}, \quad i \in [n], j \in [N].$$

In this case, the feasible region is the product of $n$ simplices, where the $i$-th simplex is defined in the space of variables $x_j, j \in \Pi_i$, via the constraints

$$\sum_{j \in \Pi_i} x_j = 1, \quad x_j \geq 0.$$
In general, the feasible region of a \( \Pi \)-compatible LP is combinatorially equivalent to the product of \( n \) simplices with a total of \( N \) facets: as a consequence of Observation 2.2(ii), the vertex-edge graph of the feasible region is a grid. By nondegeneracy, the feasible region is a therefore a simple polytope, sharing its graph with the product of simplices defined in (2). Because the combinatorial structure of a simple polytope is determined by its graph [4, 27], the statement follows.

A unique sink orientation of the grid \( G \) is obtained from a \( \Pi \)-compatible LP with \textit{generic} objective function vector \( c \), meaning that \( c^T x \) is not constant on any edge. In this case, an edge can be directed towards its vertex of lower objective function value. The resulting orientation is a USO, with the unique sink of the face \( G(S) \) corresponding to the unique optimal basis of the linear program

\[
\begin{align*}
\text{minimize} & \quad c^T x_S \\
\text{subject to} & \quad A_S x_S = b, \\
& \quad x_S \geq 0
\end{align*}
\]

in the \( |S| \) variables \( x_S := (x_j)_{j \in S} \).

The simplex algorithm (Chvátal’s book [6] contains an excellent introduction) determines the edge orientations in terms of \textit{reduced cost coefficients}. More precisely, if \( J \) is some basis, the row vector

\[
\bar{c}(J) := c^T - c_j A_j^{-1} A
\]

is the \textit{reduced cost vector} associated with \( J \). Note that \( \bar{c}(J)_j = 0 \), and that \( \bar{c}(J)_j \neq 0 \) for \( j \notin J \), if \( c \) is generic. Then, if \( J' \) is adjacent to \( J \), with \( j \) being the unique index in \( J' \setminus J \), we have

\[
J \rightarrow J' \iff \bar{c}(J)_j < 0.
\]

The existence of a unique sink \( K \) therefore provides us with a vector \( y^T := c_K A_K^{-1} \) such that

(i) \( c^T \geq y^T A \), and

(ii) for every \( i \in [n] \), there is some \( j \in \Pi_i \) with \( c_j = (y^T A)_j \).

Note that the second set of constraints forces \( y \) to be of the form \( c_j A_j^{-1} \) for some vertex \( J \). If \( c \) is generic, \( y \) is then obviously unique. Nevertheless, uniqueness also holds in the nongeneric case, but possibly with several generating bases \( K \). In fact, \( y \) is an optimal solution to the LP \textit{dual} to (1), and it is the unique solution by nondegeneracy of the primal.

If \( A \) and \( c \) do not come from a \( \Pi \)-compatible LP as in (1), a vector \( y \) satisfying properties (i) and (ii) need not exist. However, restricting \( A \) to be a matrix that—in a suitable sense—respects the partition \( \Pi \), we arrive at the following generalization of \( \Pi \)-compatible LP.

\textbf{Definition 2.3} Let \( A \in \mathbb{R}^{n \times N} \) and \( c \in \mathbb{R}^N \) such that \( A \) has property \( P \), meaning that condition (i) of Observation 2.2 holds. The \( P \)-generalized linear complementarity problem (PGLCP) is the problem of finding a vector \( y \in \mathbb{R}^n \) such that

\[
c^T \geq y^T A,
\]

in the \( |S| \) variables \( x_S := (x_j)_{j \in S} \).
and with the property that for every $i \in [n]$, there is some $j \in \Pi_i$ satisfying
\[
c_j = (y^T A)_j.
\] (6)

Intuitively, PGLCP is LP of the form (1) ’without a right-hand side’. It turns out that under this generalization, uniqueness of solution as well as the USO formulation persist. Because already the cyclic cube USO of Figure 2 arises from a PGLCP (Stickney & Watson [40] give an example in an equivalent dual setting, see also Section 4.3), the generalization is proper: given a PGLCP instance $(A, c)$, it is not always possible to find a right-hand side $b$ such that $A, b, c$ form a $\Pi$-compatible LP.

**Theorem 2.4** Let $A \in \mathbb{R}^{n \times N}$ and $c \in \mathbb{R}^N$ define a PGLCP instance.

(i) There exists a unique vector $y \in \mathbb{R}^n$ satisfying conditions (5) and (6).

(ii) If $c$ is generic (which means that $\bar{c}(J)_j \neq 0$ for $j \notin J \in V$), the edge orientations given by (4) define a unique sink orientation of the grid $G$.

**Proof.** Fix some vertex $J$, define
\[
M^T := A_J^{-1} A, \\
q^T := \bar{c}(J) = c^T - c^T_J A_J^{-1} A,
\]
and consider the problem of finding $z \in \mathbb{R}^n, w \in \mathbb{R}^N$ such that
\[
w - Mz = q,
\] (7)
\[
\prod_{j \in \Pi_i} w_j = 0, \quad i \in [n],
\] (8)
and
\[
w, z \geq 0.
\] (9)

By definition, every solution must satisfy
\[
z_i = w_j, \quad i \in [n], j \in J_i,
\] (10)
meaning that (8) is equivalent to
\[
z_i \prod_{j \in \Pi_i} w_j = 0.
\] (11)

Equations (7), (9) and (11) define the P-generalized linear complementarity problem according to Cottle & Dantzig who show that a feasible solution $(w, z)$ exists if every representative submatrix of $M^T$ is a P-matrix [8]. A matrix is a P-matrix if all determinants of principal minors are positive [9]. In our case, $M^T$ satisfies this, which easily
follows from the fact that $M^T$ has property P and contains a representative identity matrix. Note that the $n$ variables $w_j, j \in J$ of system (7) are redundant by (10) and can be deleted from the problem, along with their corresponding rows.

To prove the existence of $y$ in part (i), it remains to observe that $y$ solves (5) and (6) if and only if $z^T = c^T - y^T A_j$ and $w^T = c^T - y^T A$ solve (7), (8) and (9). The uniqueness of $y$ follows from the known uniqueness proofs in the setup of Cottle & Dantzig [43, 23].

Statement (ii) is a corollary of (i), because $c$ being generic implies that the vector $y$ from (i) can be expressed in the form $y^T = c^T K A_1$ for exactly one $K$. This set $K$ is the unique sink of $G$ in the orientation defined by (4). The fact that this orientation defines a USO easily follows: applying the above arguments to the PGLCP instance $(A_S, c_S)$, we can prove the existence of a unique sink in the subgrid $G(S)$.

Given a PGLCP instance $(A, c)$ and $i \in [n]$, the replacement of submatrix $A_{i\Pi}$ with $-A_{i\Pi}$ preserves property P. Let $A'$ be the resulting matrix, and let $c'$ be the vector resulting from $c$ by replacing the subvector $c_{i\Pi}$ with its negative. It is easily seen from (3) and (4) that the USO induced by $(A, c)$ and $(A', c')$ differ exactly in the orientations of the edges along dimension $i$. This can be summarized in the following

**Theorem 2.5** Let $\psi$ be a grid orientation, choose $i \in [n]$, and let $\phi$ be the orientation resulting from $\psi$ by flipping all arcs of the form

$$(J, J'), \quad J \oplus J' \subseteq \Pi_i.$$  

Then $\psi$ is a PGLCP-induced USO if and only if $\phi$ is a PGLCP-induced USO.

Theorem 2.12 proves that the class of general grid USO is closed under this flipping operation as well.

### 2.2 USO Properties

In this subsection, we study general grid USO according to Definition 2.1 and develop some basic theory. The material will also be used in later sections.

**Outmap and $h$-vector.** Any USO can be specified by associating each vertex $J$ with its outgoing edges. Given $J$ and $j \in [N] \setminus J$, we define $J \rhd j$ to be the unique vertex $J' \subseteq J \cup \{j\}$ which is different from $J$, and we call $J'$ the neighbor of $J$ in direction $j$. Note that $J$ is a neighbor of $J'$ in some direction different from $j$.

Given an orientation $\psi$, the function $s_\psi : V \to 2^{[N]}$, defined via

$$s_\psi(J) := \{j \in [N] \setminus J \mid J \xrightarrow{\psi} J \rhd j\},$$

is called the outmap of $\psi$. Björklund et al. consider the outmap for acyclic grid USO and call it VID-function (vector of improving directions) [2]. Szabó and Welzl [42] deal with outmaps for cube USO; formally, these are different from ours, even when we specialize to the cube case, because Szabó and Welzl identify the $n$-cube vertices with the subsets of some $n$-element set, while we identify them with representative $n$-subsets of some $2n$-element set. Still, we can generalize the characterization of USO outmaps by Szabó and Welzl, using one more ingredient.
Definition 2.6 Let \( s : V \to 2^{[N]} \). The vector \( h(s) = (h_0(s), \ldots, h_{N-n}(s)) \), defined via
\[
h_k(s) = \#\{J \in V \mid |s(J)| = k\}
\]
is called the h-vector of \( s \). If \( s = s_\psi \) for some orientation \( \psi \) on \( G \), we also refer to \( h(s) \) as \( h(\psi) \).

The following result is well-known for the acyclic case \([25, 2]\) but going through its proof, one realizes that only the unique sink property is used \([44]\).

Theorem 2.7 \( h(\psi) = h(\psi') \) for any two USO \( \psi, \psi' \) of \( G \).

As all USO on a grid have the same h-vector, we denote it by \( h() \), emphasizing the fact that it depends on the parameters of the grid only. An immediate corollary of this theorem is that the h-vector is symmetric, meaning that \( h_k(s) = h_{N-n-k}(s) \) for all \( k \in \{0, \ldots, N-n\} \). To see this, choose \( \psi \) as some PGLCP-induced USO; repeated application of Theorem 2.5 guarantees that the orientation \( \psi' \) in which all arcs are flipped is again PGLCP-induced, so that \( h_k(\psi) = h_{N-n-k}(\psi') \) holds. In particular, w.r.t. any given USO, all nonempty subgrids of \( G \) also have unique sources.

Here is the characterization of functions \( s \) that are of the form \( s_\psi \), for \( s \) being USO.

Lemma 2.8 Let \( s : V \to 2^{[N]} \) satisfy \( s(J) \cap J = \emptyset \) for all \( J \in V \). \( s \) is the outmap of a USO of \( G \) if and only if

(i) \( (s(J) \oplus s(J')) \cap (J \oplus J') \neq \emptyset \), for all \( J \neq J' \), and

(ii) \( h(s) = h(\Pi) \).

Proof. If \( s \) is the outmap of a USO, (ii) follows from Theorem 2.7. To see (i), consider the cube \( G(J \cup J') \) spanned by \( J \) and \( J' \), and assume w.l.o.g. that \( J \) is not the sink of that cube. Then there is \( j \in J' \setminus J \) such that \( j \in s(J) \), and because \( j \notin s(J') \), (i) follows.

For the other direction, assume that \( s \) is a function satisfying (i) and (ii). \( s \) defines a set of arcs
\[
D = \{(J, J \triangleright j) \mid J \in V, j \in s(J)\}.
\]

We first show that \( D \) contains exactly one arc for each grid edge \( \{J, J \triangleright j\} \), meaning that \( D \) defines a proper orientation \( \psi \). From (i), it follows that each edge generates at least one arc. On the other hand, we have
\[
|D| = \sum_k k h_k(s) \overset{(ii)}{=} \sum_k k h_k(\Pi) = |E|,
\]
because \( h(\Pi) \) is defined via a proper orientation. It follows that each edge gets a unique orientation.

It remains to prove that \( \psi \) is USO. (i) implies that no face has more than one sink w.r.t. \( \psi \): assuming there are sinks \( J \neq J' \) in some face \( G(S) \), it follows that \( s(J) \cap S = s(J') \cap S = \emptyset \), and together with \( J, J' \subseteq S \), we get a violation of (i).
Condition (ii) then guarantees that every face has exactly one sink. To see this, note that every $J$ which is counted for $h_k(s)$ is the sink in exactly $2^{N-n-k}$ faces, spanned by $J$ and some subset of its incoming edges. In other words, the expression

$$\sum_k 2^{N-n-k} h_k(s) = \sum_k 2^{N-n-k} h_k(\Pi)$$

counts the total number of vertex-face pairs $(J, F)$ such that $J$ is a sink in $F$ w.r.t. $\psi$. By the previous equation, this number equals the total number of nonempty faces, because in any USO, all nonempty faces have unique sinks. This again means that in $\psi$, each nonempty face has on average one sink, which—together with the previously derived fact that all faces have at most one sink—implies that $\psi$ is USO.

Inherited USO. Let $I \subseteq [n]$. Two vertices $J, J' \in V$ are called $I$-equivalent if and only if $J_I = J'_I$. This relation is an equivalence relation, and we call the equivalence classes of it the hypervertices generated by $I$. For example, the hypervertices generated by $I = [n]$ correspond to the original vertices, while $I = \emptyset$ induces one hypervertex containing all the vertices. Figure 3 (page 4) depicts a partition of a grid into 1-dimensional hypervertices.

**Definition 2.9** For $I \subseteq [n]$ and $J \in V$,

$$J^I := \{ J' \in V \mid J_I = J'_I \}$$

is the hypervertex generated by $I$ and $J$.

$$V^I := \{ J^I \mid J \in V \}$$
denotes the set of all these hypervertices.

$I$ defines for any subset $S \subseteq \bigcup_{i \in I} \Pi_i$ an inherited grid $G^I(S) = (V^I(S), E^I(S))$ on the hypervertices, via

$$V^I(S) := \{ J^I \in V^I \mid J_I \subseteq S \}$$

$$E^I(S) := \{ \{ J^I, J'^I \} \subseteq V^I(S) \mid |J_I \oplus J'_I| = 2 \}.$$  

By definition of a hypervertex, this graph is well-defined and isomorphic to the grid defined by the ordered partition $\Pi^I(S) = (S_i)_{i \in I}$. All edges in $E^I(S)$ are of the form

$$\{ J^I, (J \gtrdot j)^I \}, \quad j \in \bigcup_{i \in I} S_i \setminus J.$$  

To obtain $G^I(S)$, one starts with the face $G(S \cup \bigcup_{i \in [n] \setminus I} \Pi_i)$ of the original grid and collapses all dimensions $i \in [n] \setminus I$. Vertices of the original grid that only differ w.r.t. collapsed dimensions are identified with each other, resulting in a single hypervertex

$$J^I = V(J_I \cup \bigcup_{i \in [n] \setminus I} \Pi_i).$$
In particular, any hypervertex corresponds to a face of the original grid. As before, if \( S = \bigcup_{i \in I} \Pi_i \), we may omit the parameter \( S \) from \( G^I, V^I \) and \( E^I \). We also abuse notation and identify a hypervertex with the face of the original grid that it corresponds to. We are now prepared to introduce the important concept of an inherited orientation.

**Definition 2.10** Let \( \psi \) be an orientation of the grid \( G, I \subseteq [n] \). The orientation \( \psi^I \) is defined on \( G^I \) via

\[
J^I \xrightarrow{\psi^I} (J \triangleright j)^I \iff K \xrightarrow{\psi} K \triangleright j,
\]

where \( K \) is the sink (w.r.t. \( \psi \)) of the face \( J^I \).

In Figure 3, the hypervertex sinks \( K \) correspond to the vertices in white.

**Lemma 2.11** If \( \psi \) is USO, then \( \psi^I \) is USO.

**Proof.** Any nonempty face of \( G^I \) corresponds to some inherited grid \( G^I(S) \), obtained by collapsing dimensions in the face \( F = G(S \cup \bigcup_{i \in [n]\setminus I} \Pi_i) \) of the grid \( G \). The hypervertex \( J^I \) that contains the sink of \( F \) w.r.t. \( \psi \) is a sink in \( \psi^I \). Conversely, any sink w.r.t. \( \psi^I \) contains a vertex that is a sink in \( F \) w.r.t. \( \psi \). This implies the USO property.

Using the concept of inherited orientations, the ’edge flipping’ Theorem 2.5 is easily established for general USO.

**Theorem 2.12** Let \( \psi \) be a grid orientation, choose \( i \in [n] \), and let \( \phi \) be the orientation resulting from \( \psi \) by flipping all arcs of the form

\[
(J, J'), \quad J \oplus J' \subseteq \Pi_i.
\]

Then \( \psi \) is USO if and only if \( \phi \) is USO.

**Proof.** Let \( I = \{i\} \). By construction, \( \phi^I \) arises from \( \psi^I \) by flipping the orientations of all edges of the 1-dimensional inherited grid \( G^I \). Assuming that \( \psi \) is USO, \( \psi^I \) is USO, and from the existence of unique sources w.r.t. \( \psi^I \) (see the discussion after Theorem 2.7), it follows that \( \phi^I \) is USO as well. The sinks of \( G \) w.r.t. \( \phi \) are exactly the sinks (w.r.t. \( \psi \)) of exactly the hypervertices \( J^I \) that are sinks of \( G^I \) w.r.t. \( \phi^I \). Because both orientations involved in this argument are USO, there is a unique sink of \( G \) w.r.t. \( \phi \). The same can be shown for any face of \( G \), so the USO property of \( \phi \) follows.

**Refined Index.** The outmap value \( s_\psi(J) \subseteq [N] \) of a vertex \( J \) w.r.t. some USO \( \psi \) is partitioned according to the dimensions of the grid (the sets \( \Pi_i \)). Accordingly, the outdegree of \( J \) can be refined to an \( n \)-vector of dimensional outdegrees, as follows.

**Definition 2.13** Let \( \psi \) be a grid USO. The function

\[
r_\psi : V \rightarrow \prod_{i=1}^{n} \{0, \ldots, |\Pi_i| - 1\},
\]

with

\[
r_\psi(J) = (|s_\psi(J) \cap \Pi_1|, \ldots, |s_\psi(J) \cap \Pi_n|), \quad J \in V
\]

is called the **refined index** of \( \psi \).
Unlike the outmap, the refined index is a mapping between two sets of the same size, so it is natural to ask whether this mapping is a bijection. This is true in the cube case (where the refined index is just a different way of writing the outmap) [42], and it also holds for grid AUSO [2], where Björklund et al. use the term *signature* (SIG). The proof of the latter result does not generalize to the case of general USO. We show that the refined index is a bijection in general, using the concept of inherited orientations.

**Theorem 2.14** Let $\psi$ be a grid USO. The refined index $r_\psi$ is a bijection.

**Proof.** We proceed by induction on $n$. For $n = 1$, Theorem 2.7 implies $h_k(\psi) = 1$ for all $k \in \{0, \ldots, |\Pi_1| - 1\}$. In other words, for any $k$ in this range, there is exactly one vertex whose outmap has size $k$. The result follows.

For $n > 1$, we set $I = [n] \setminus \{i\}$, for $i$ arbitrarily chosen from $[n]$ and consider the $(n-1)$-dimensional grid $G^I$ over the 1-dimensional hypervertices $J^I, J \in V$. Now fix some value $k \in \{0, \ldots, |\Pi_i| - 1\}$. By induction (or rather, Theorem 2.7 again), we may w.l.o.g. choose the generating vertex $J$ of any hypervertex $J^I$ to be the unique vertex in $J^I$ such that $|s_\psi(J) \cap \Pi_i| = k$.

**Claim.** The orientation $\phi$ of $G^I$, defined via

$$J^I \xrightarrow{\phi} (J \triangleright j)^I \iff J \xrightarrow{\psi} J \triangleright j$$

is USO, see Figure 6.

![Figure 6](image)

Figure 6: A USO $\psi$ on the left and its 'inherited' USO $\phi$ generated by vertices (white) having refined index 1 w.r.t. their hypervertices (shaded)

Note that for $k = 0$, the claim is Lemma 2.11 applied to the special case $I = [n] \setminus \{i\}$. For $k > 0$, we prove that $G^I$ contains exactly one sink w.r.t. $\phi$; the argument for a face $G^I(S)$ is the same, after restricting to the partition $\Pi' = \Pi^I(S)$.

First, let us assume that $G^I$ contains two distinct sinks $J^I, J'^I$ w.r.t. $\phi$. If $J_i = J'_i$, then $G(([N] \setminus \Pi_i) \cup J_i)$ contains two distinct sinks $J, J'$ w.r.t. $\psi$ which is impossible. If $J_i \neq J'_i$, we consider the 1-dimensional grid $G^{(i)}$, in which $J$ and $J'$ are the sinks (w.r.t. $\psi$) of their respective $(n - 1)$-dimensional hypervertices. However, both hypervertices
have outdegree $k$ in the inherited USO $\psi^{(i)}$, contradicting $h_k(\psi^{(i)}) = 1$. The case in which $G^I$ has no sink w.r.t. $\phi$ leads to $h_k(\psi^{(i)}) = 0$, a contradiction as well.

Having established the claim, we can resume the main proof. It is sufficient to show that no two distinct vertices $J, J' \in V$ have the same refined index value. If $r_\psi(J)_i \neq r_\psi(J')_i$, we are done. Otherwise, $r_\psi(J)_i = r_\psi(J')_i = k \in \{0, \ldots, \Pi_i - 1\}$, and the claim provides us with a USO $\phi$ on $G^I$ for which $r_\phi(J^I) \neq r_\phi(J'^I)$ holds by induction (note that $J^I \neq J'^I$ as a consequence of $r_\psi(J)_i = r_\psi(J')_i$). By definition of $\phi$,

$$
r_\phi(J^I)_i = r_\psi(J)_i, \quad r_\phi(J'^I)_i = r_\psi(J')_i \quad i' \neq i,
$$

and $r_\psi(J) \neq r_\psi(J')$ follows.

With this result, vertex $K$ in Definition 2.10 of the inherited orientation is the unique vertex in $J^I$ satisfying $r_\psi(K)_i = 0, i \in [n] \setminus I$. The refined index bijection more generally implies that this definition also yields a USO if in all $J^I$, $K$ is chosen as the unique vertex with $r_\psi(K)_i = k_i, i \in [n] \setminus I$, where the $k_i \in \{0, \ldots, \Pi_i - 1\}$ are any fixed numbers. The orientation $\phi$ derived in the proof of Theorem 2.14 is an instance of such a generalized inherited USO.

## 3 Algorithms

In this section, we develop algorithms for finding the sink of a given grid USO, implicitly specified by either an edge evaluation or a vertex evaluation oracle. The edge evaluation oracle must be able to return the orientation of any given grid edge, while the vertex evaluation oracle returns the outmap value of any given vertex. Our complexity measure will be the maximum (expected) number of oracle calls needed to find the sink in the worst case. In concrete instances, both oracles can typically be implemented to run in polynomial time. In the case of PGLCP, for example, an edge evaluation must return the sign of a single coefficient of the reduced cost vector (3), while a vertex evaluation computes all signs. Obviously, the necessary number of edge evaluations cannot exceed the number of vertex evaluations by a factor of more than $N - n$, but better bounds may be possible.

Under both oracles, the additional computations performed by our (very simple) algorithms do not generate asymptotic overhead, and this justifies our measure of complexity. In particular, any USO algorithm which calls the oracle only a polynomial number of times is actually a strongly polynomial algorithm for LP and PGLCP. Moreover, the complexity of a single edge evaluation typically only depends on $n$ but not on $N$ (in the PGLCP case, this complexity is $O(n^3)$). Therefore, if $n$ is considered to be a constant, any bound on the number of edge evaluations determines the complexity of the algorithm up to a constant factor. Therefore, counting edge evaluations will be our primary concern.
3.1 The Product Algorithm

This algorithm generalizes the product algorithm of Szabó and Welzl from cubes [42] to grids, with a slight twist: while in the $n$-cube, all dimensions are equivalent with respect to their size (which is two), a general grid may have 'heavy' dimensions (with large $\Pi_i$) and 'light' dimensions. The analysis will reveal why our algorithm gives priority to the heavy dimensions. The basic idea is the following.

Choose $i \in [n]$ such that $|\Pi_i| > 1$ and recursively compute the sink $K$ of the subgrid $G([N] \setminus \{j\})$, where $j \in \Pi_i$. If the edge incident to $K$ in direction $j$ is incoming, we have already found the global sink, otherwise we need to search the lower-dimensional 'facet' $G((|N| \setminus \Pi_i) \cup \{j\})$ recursively. A generic call to PRODUCT finds the sink of a nonempty face $G(S)$. The correctness proof of the algorithm is a straightforward induction, using the USO properties. By the end of this subsection, it will also become clear why the algorithm is called PRODUCT.

Algorithm 3.1

\textbf{PRODUCT}(S):

(* Input: USO $\psi$ on $G(S)$, given by edge evaluation oracle *)

(* Output: unique sink of $G(S)$ w.r.t. $\psi$ *)

(* Invariant: $S$ contains a vertex of $G$ *)

\begin{verbatim}
IF is_vertex(S) THEN
  RETURN S
ELSE
  choose a heaviest $S_i \in S$
  choose $j \in S_i$ uniformly at random
  $K := \text{PRODUCT}(S \setminus \{j\})$
  $K' := K \triangleright j$
  IF $K' \xrightarrow{\psi} K$ THEN
    RETURN $K$
  ELSE
    RETURN $\text{PRODUCT}((S \setminus S_i) \cup \{j\})$
  END
END
\end{verbatim}

To prepare the analysis of the algorithm, we need one more

\textbf{Definition 3.2} Let $S \subseteq [N]$ be a set containing a vertex and let $z = \max_{i=1}^{n} |S_i|$ be the size of a heaviest dimension. The $(z - 1)$-vector $(a_2, \ldots, a_z)$, defined through

\[a_t = |\{i \in [n] \mid |S_i| = t\}|\]

is called the characteristic of $S$. If $z = 1$ (meaning that $S$ is a vertex itself), the characteristic is the empty vector $()$. 
For example, the \( n \)-cube has characteristic \( (n) \), and the grid in our first Figure 1 has characteristic \( (2, 1) \). It can easily be shown by induction that the expected number of edge evaluations in PRODUCT\( (S) \) only depends on the characteristic of \( S \) but (maybe surprisingly) not on the input USO \( \psi \). We can even compute the exact expectation.

**Theorem 3.3** For \( z \geq 1 \), let \( T_e(a_2, \ldots, a_z) \) denote the expected number of edge evaluations in a call to PRODUCT\( (S) \), where \( S \) has characteristic \( (a_2, \ldots, a_z) \). Then

\[
T_e(a_2, \ldots, a_z) = T_e(a_2, \ldots, a_{z-1}+1, a_z-1) + 1 + \frac{1}{z} T_e(a_2, \ldots, a_{z-1}, a_z-1), \quad z > 1, \tag{13}
\]

with

\[
T_e(a_2, \ldots, a_{z-1}, 0) := T_e(a_2, \ldots, a_{z-1}) \text{ and } T_e() = 0.
\]

The solution to this recurrence is

\[
T_e(a_2, \ldots, a_z) = \prod_{k=2}^{z} H_k^{a_k} + \sum_{k=2}^{z} \prod_{t=k}^{z} (H_t - H_{k-1} + 1)^{a_t} - z. \tag{14}
\]

Here, \( H_k = \sum_{t=1}^{k} 1/t \) is the \( k \)-th Harmonic number.

**Proof.** It is clear that \( (a_2, \ldots, a_{z-1}+1, a_z-1) \) and \( (a_2, \ldots, a_{z-1}, a_z-1) \) are the characteristics of the grids handled in the recursive calls. Moreover, the second recursive call is executed if and only if the global sink contains the chosen element \( j \). This happens with probability \( 1/z \). The recurrence follows. The closed form (14) can be checked by induction.

As an example, consider \( T_e(a_2) \) (the cube case). The theorem yields

\[
T_e(a_2) = 2 \left( \frac{3}{2} \right)^{a_2} - 2,
\]

which is still exponential but much smaller than the total number \( a_2 2^{a_2-1} \) of cube edges.

For fixed \( z \), (14) is maximized if \( a_z = n \). This corresponds to the characteristic \( (0, \ldots, 0, n) \) of the \( (z \times \cdots \times z) \)-grid. It follows that

\[
T_e(a_2, \ldots, a_z) \leq H_n^z + \sum_{k=2}^{z} (H_z - H_{k-1} + 1)^n - z. \tag{15}
\]

The middle term

\[
f(n, z) := \sum_{k=2}^{z} (H_z - H_{k-1} + 1)^n \tag{16}
\]

asymptotically dominates the bound in (15), and an estimate of \( f(n, z) \leq (z - 1) H_n^z \approx z \ln^n z \) immediately follows. The next result shows that the bound is actually linear in \( z \).

**Lemma 3.4**

\[
T_e(a_2, \ldots, a_z) \leq ([en!] - 1) z + H_n^z.
\]

18
Proof. Using the estimate
\[ \sum_{t=1}^{n} g(t) \leq \int_{t-1}^{n} g(x) \, dx \]  
for any decreasing function \( g \) such that the integral exists, we can bound (16) as follows.

\[
f(n, z) = \sum_{k=2}^{n} (H_z - H_{k-1} + 1)^n \leq \sum_{k=2}^{n} (\ln z - \ln(k - 1) + 1)^n.
\]

\[
f(n, z) = \sum_{k=1}^{z-1} \left(1 - \ln \frac{k}{z}\right)^n < \sum_{k=1}^{z} \left(1 - \ln \frac{k}{z}\right)^n.
\]

\[
\leq \int_{0}^{z} \left(1 - \ln \frac{k}{z}\right)^n \, dk = z \int_{0}^{1} (1 - \ln x)^n \, dx := zI_n.
\]

Integration by parts yields the recurrence relation
\[
I_n = 1 + nI_{n-1},
\]
with \( I_0 = 1 \). This solves to \( I_n = [en!] \) for \( n > 0 \) \cite{39}. The statement follows. \( \square \)

This means, algorithm PRODUCT solves any PGLCP instance with a fixed number \( n \) of blocks in expected time \( O(z) = O(N) \) which is asymptotically optimal.

Algorithm 3.1 can easily be rewritten to work under a vertex evaluation oracle. For this, we simply need to call the oracle for the set \( S \) immediately before we RETURN \( S \). This has the effect that all face sinks computed by recursive calls of the algorithm get evaluated before they are returned, meaning that all tests ‘\( K' \not\rightarrow K \)’ come for free.

The resulting expected number \( T_v(a_2, \ldots, a_z) \) of vertex evaluations satisfies the recurrence
\[
T_v(a_2, \ldots, a_z) = T_v(a_2, \ldots, a_{z-1} + 1, a_z - 1) + \frac{1}{z} T_v(a_2, \ldots, a_{z-1}, a_z - 1),
\]
with
\[
T_v(a_2, \ldots, a_{z-1}, 0) := T_v(a_2, \ldots, a_{z-1}) \text{ and } T_v() = 1.
\]

Its solution is
\[
T_v(a_2, \ldots, a_z) = \prod_{k=2}^{z} H_k^{a_k}. \tag{19}
\]

For the cube with characteristic \( (a_2) \), we get
\[
T_v(a_2) = H_2^{a_2} = \left(\frac{3}{2}\right)^{a_2}.
\]

\( T_v(a_2, \ldots, a_z) \) is maximized if \( a_z = n \), from which we obtain the upper bound
\[
T_v(a_2, \ldots, a_z) \leq H_z^n.
\]

For fixed \( n \), the expected number of vertex evaluations required to find the sink is therefore only polylogarithmic in \( z \), equivalently in \( N \). In order to turn this bound into
an actual runtime bound, an $O(N)$ factor has to be included, though, which makes the algorithm superlinear. The reason is that under the vertex evaluation oracle, all edges incident to a vertex are evaluated, regardless of whether their orientation is needed later or not. Our previous analysis in terms of edge evaluations avoids this.

Algorithm 3.1 is a close relative of algorithms due to Seidel (for linear programming with $n$ variables and $N$ constraints) [38] and Welzl (for finding the smallest enclosing ball of a set of $N$ points in dimension $n$) [45]. If $N \leq n+1$, the smallest enclosing ball problem is known to induce a cube USO [42, 18], but for larger $N$, the problem has in general no apparent grid structure. The complexity bounds for Welzl’s algorithm are very similar to ours and hold as upper bounds. In our scenario, the additional grid structure makes it possible to determine the complexity exactly.

Instead of choosing a heaviest dimension $S_i$ in the algorithm, other rules could be applied. For example, we might choose a lightest dimension, or even a random one. It turns out that the expected number of vertex evaluations does not depend on the particular rule and always equals the expression in (19). In order to explain this, let us adopt a somewhat different view of the algorithm under the vertex evaluation oracle.

Let $i$ be the heaviest (or any other) dimension and set $I = \{i\}$. We can consider the grid $G$ as being the product of the inherited grids $G^I$ and $G^{[n]\setminus I}$. Algorithm PRODUCT can then be considered as an algorithm that evaluates the sink of the one-dimensional inherited USO $\psi^I$ of $G^I$. For this, it chooses a random hypervertex $J^I$ in $G^I$ (more precisely, the one where $J_i = j$) and recursively evaluates the sink $K^I$ among the remaining $|\Pi_i| - 1$ hypervertices.\(^3\) If $K^I$ is not the sink w.r.t. $\psi^I$ yet, one more evaluation of the true sink $(K \triangleright j)^I = J^I$ is necessary. For each vertex evaluation w.r.t. $\psi^I$, the algorithm evaluates the sink of a hypervertex $J^I$ (which has the structure of $G^{[n]\setminus I}$) w.r.t. the original USO $\psi$. The total (expected) number of vertex evaluations is therefore the product of the respective numbers for the grids $G^I$ and $G^{[n]\setminus I}$. For $G^I$ (a one-dimensional grid), this number is easily seen to be $H_{|\Pi_i|}$ from which (19) follows by induction.

Interestingly, the expected number of edge evaluations does depend on the choice of $i$, and a bad choice can lead to a complexity which is asymptotically worse than what we found in Theorem 3.3. For instance, always choosing $i$ to be the lightest dimension yields an expected number of

$$T_e(a_2, \ldots, a_z) = \sum_{k=1}^{z-1} k \left( \prod_{\ell=2}^{k} H_{\ell}^{a_{\ell}} \right) \sum_{m=0}^{a_{k+1}-1} H_{k+1}^{m}$$

edge evaluations, which is again largest for $a_z = n$. In this case we get

$$(z - 1) \sum_{m=0}^{a_z-1} H_z^m = \frac{z - 1}{H_z - 1} (H_z^n - 1),$$

which is superlinear in $z$ and asymptotically not better than the obvious upper bound obtained from multiplying (19) by $N - n$. We believe (although we cannot prove it

\(^3\)In this recursive call, the algorithm might proceed according to some dimension different from $i$, but inductively, we can argue that this does not influence the expected number of vertex evaluations, so we might as well stick to dimension $i$.\)
formally) that our choice of \( i \) in Algorithm 3.1 leads to the smallest possible expected number of edge evaluations.

### 3.2 The Algorithm RandomFacet

It is certainly pleasing from a mathematical point of view that the complexity of the PRODUCT algorithm does not depend on the USO \( \psi \) and can be evaluated exactly (Theorem 3.3). On the other hand, this means that the algorithm fails to deal with easy orientations \( \psi \) in the appropriate way. Consider for example the uniform USO \( \psi \), given by

\[
J \mapsto J \cup \{j\} \setminus \{j'\} \iff j < j'.
\]

The sink is the vertex \( J \) which collects from each block \( \Pi_i \) the smallest element. Moreover, any directed path in \( \psi \) has length at most \( N - n \). It follows that any path-following algorithm will find the sink quickly while the algorithm PRODUCT will be comparatively slow. In this subsection, we present a path-following algorithm whose expected worst-case complexity is not much larger than that of algorithm PRODUCT. In particular, its runtime is again \( O(N) \) for fixed \( n \).

The RANDOMFACET algorithm shares its basic idea with the PRODUCT algorithm. However, in addition to the current set \( S \), it maintains a current vertex \( J \in V(S) \) which may be replaced at some point by a neighbor of \( J \) along an outgoing edge.

In order to guarantee the invariant \( J \in V(S) \), the element \( j \) which gets removed from \( S \) for the first recursive call must not be in \( J \). This restricts \( j \) to the set \( S_i \setminus J_i \), where the PRODUCT algorithm was free to choose \( j \in S_i \). As before, the correctness of the algorithm is easily established through the USO axioms.

**Algorithm 3.5**

**RANDOMFACET**(*J, S*):

(* Input: USO \( \psi \) on \( G(S) \), given by edge evaluation oracle *)

(* Output: unique sink of \( G(S) \) w.r.t. \( \psi \) *)

(* Invariant: \( V \ni J \subseteq S \) *)

**IF** \( S = J \**

**RETURN** \( J \)

**ELSE**

choose a heaviest \( S_i \in S \)
choose \( j \in S_i \setminus J_i \) uniformly at random

\( K := \text{RANDOMFACET}(J, S \setminus \{j\}) \)

\( K' := K \triangleright j \)

**IF** \( K' \overset{\psi}{\rightarrow} K \**

**RETURN** \( K \)

**ELSE**

**RETURN** \( \text{RANDOMFACET}(K', (S \setminus S_i) \cup \{j\}) \)

**END**

**END**
The following recurrence relation is derived as in the case of the PRODUCT algorithm, where the probability of the second recursive call is now at most $1/(z-1)$. The closed formula for the upper bound can again be proved inductively.

**Theorem 3.6** For $z \geq 1$, let $T_e'(a_2, \ldots, a_z)$ denote the expected number of edge evaluations in a call to $\text{RANDOMFACET}(S)$, where $S$ has characteristic $(a_2, \ldots, a_z)$. Then

$$T_e'(a_2, \ldots, a_z) \leq T_e'(a_2, \ldots, a_{z-1} + 1, a_z - 1) + 1 + \frac{1}{z-1}T_e'(a_2, \ldots, a_{z-1}, a_z - 1), \quad z > 1,$$

with

$$T_e'(a_2, \ldots, a_{z-1}, 0) := T_e'(a_2, \ldots, a_{z-1}) \text{ and } T_e'() = 0.$$

An upper bound is

$$T_e'(a_2, \ldots, a_z) \leq \sum_{k=1}^{z-1} \prod_{\ell=k}^{z-1} (H_\ell - H_{k-1} + 1)^{a_{\ell+1}} - z + 1.$$

As an example, let us consider the cube case again. The theorem yields

$$T_e'(a_2) \leq 2^{a_2} - 1.$$

which is not better than the trivial upper bound that holds for any path-following algorithm. The reason is that $j$ is ‘randomly chosen’ from a set of size one, so there is no randomness left. In fact, if we break ties in the choice of $i$ by choosing $i$ for example as the largest index for which $|S_i| = 2$, the bound in (22) is tight. A worst-case input is the Klee-Minty cube, an $n$-cube AUSO with a directed Hamiltonian path through all vertices. This AUSO is actually generated by an LP and was the first example to show that the simplex method with Dantzig’s pivot rule may require an exponential number of pivot steps [31].

If we break ties randomly, the situation changes. For all cube AUSO, a subexponential upper bound of

$$e^{2\sqrt{a_2}} - 1$$

on the expected number of edge evaluations can then be shown [16]. It is open whether this bound (or any bound of the form $2^{o(a_2)}$) holds for general cube USO.

As in the case of the PRODUCT algorithm, the complexity is maximized if $a_z = n$, and an upper bound of

$$\sum_{k=1}^{z-1} (H_{z-1} - H_{k-1} + 1)^n - z + 1 = (H_{z-1} + 1)^n + \sum_{k=2}^{z} (H_{z-1} - H_{k-1} + 1)^n - z$$

on $T_e'(a_2, \ldots, a_z)$ holds. Comparing this with the expected number of edge evaluations (15) of PRODUCT, we see that for large $z$, both algorithms have approximately the same
expected worst-case complexity which we have shown to be linear in $z$. However, only \textsc{RandomFacet} has the potential of being faster than the upper bound in practice.

The algorithm \textsc{RandomFacet} is a close relative of an algorithm by Matoušek, Sharir and Welzl for \textit{LP-type problems} [33]. This combinatorial framework covers grid AUSO but also general LP and other (acyclic) problems which have no grid structure. While the \textsc{Product} algorithm is insensitive to directed cycles, \textsc{RandomFacet} (more precisely, its present analysis) suffers to some extent: the exponential behavior in $n$ (see Lemma 3.4 which also applies to \textsc{RandomFacet} with minor modifications) is much worse than the subexponential bounds one gets for LP-type problems [33]. In other words, not only for the cube ($z = 2$) but for any grid with small $z$, cycles make a huge difference. On the other hand, if $z$ (and $N$) are large compared to $n$, our analysis of \textsc{RandomFacet} shows that cycles are much less problematic.

All complexity bounds we have developed in this section are of the form $f(n)N$, where $f(n)$ is exponential in $n$. We note that one can reduce these complexities to $O(Nn) + f(n)$, extending techniques by Clarkson [7] to deal with cyclic orientations [17].

4 The Holt-Klee Condition

A grid USO is said to be \textit{Holt-Klee} (HK), if the number of vertex-disjoint paths from source to sink is $N-n$, and if in addition, every nonempty subgrid is HK. In this section, we prove a number of HK-related results for general USO as well as for specific USO coming from PGLCP.

4.1 Three-dimensional Grids

It is easy to prove [44, 14] that a 2-dimensional grid-USO is HK if and only if the double twist of Figure 7 (A) does not occur as a subgrid. We derive a similar statement for $n = 3$.

\textbf{Theorem 4.1} For $n = 3$, a grid USO $\psi$ is HK if and only if no subgrid orientation is isomorphic to any of the three forbidden subgrids depicted in Figure 7.
Proof. Assume the grid has dimensions $d_1 \times d_2 \times d_3$. It is clear that no forbidden subgrid can occur if the orientation is HK. For the other direction, it suffices to show that $N-n$ paths can be found in the full grid $G$; the same arguments then apply to the subgrids.

We distinguish three cases, depending on the positions of the global source $\overline{J}$ and the global sink $\overline{J}$ relative to each other.

Case 1. Source and sink span a 1-dimensional face. One can easily check that in this case, any grid USO has $N-n$ paths from source to sink. The proof only requires the $2 \times 2$ subgrids to be properly oriented.

Case 2. Source and sink span a 2-dimensional face. Without loss of generality, we can assume that source and sink agree in their third coordinate, see Figure 8. The edge labeled 1 in Figure 8 (left) must be oriented as indicated, otherwise the forbidden subgrid (A) occurs. In total, there are $(d_1 - 1) + (d_2 - 1)$ vertex-disjoint ways of drawing the figure, one for each neighbor of $\overline{J}$ in the face spanned by $\overline{J}$ and $\overline{J}$. The remaining $d_3 - 1$ paths are obtained by going through the third dimension, see Figure 8 (right) where two of them are shown. Because the 3-cube spanned by $\overline{J}, \overline{J}, \overline{J}, \overline{J}$, must be HK due to the absence of the forbidden cube (B) in Figure 7, there is always a path from $\overline{J}$ to $\overline{J}$ in the upper facet of that cube (indicated by the dotted arrow).

Case 3. Source and sink span a 3-dimensional face. For $i = 1, 2, 3$, $\overline{J}$ has $d_i - 2$ neighbors in dimension $i$ which do not share a coordinate with $\overline{J}$. For any such neighbor $\overline{J}$, we prove that there is a path from $\overline{J}$ to $\overline{J}$ through $\overline{J}$, with the property that all intermediate vertices on the path share the $i$-th coordinate with $\overline{J}$, see Figure 9. This gives $N-6$ vertex-disjoint paths, and as the forbidden subgrid (C) is not allowed to occur, we get three more paths in the cube spanned by $\overline{J}$ and $\overline{J}$.

Let us consider the layer spanned by the vertices $\overline{J}$ and $\overline{J}$ in Figure 9. If there is no path from $\overline{J}$ to $\overline{J}$ in that layer, we may assume w.l.o.g. that the edges of the layer are oriented as in Figure 10, case (i) or (ii). We will derive a contradiction in both cases.

The contradiction in case (i) is obtained as follows: edges 1 and 2 must be as given in order to avoid bad 2-cubes. Edge 3 prevents the right 3-cube from having two sinks and edge 4 circumvents the forbidden subgrid (A). Edges 5 and 6 again avoid bad 2-cubes and finally edge 7, avoiding a cyclic row, yields the forbidden subgrid (A).
Figure 9: Possible paths from $\vec{J}$ to $\vec{J}'$ and the layer spanned by $J$ and $J'$, dividing the grid into two subgrids.

Figure 10: Contradictions if there is no path from $\vec{J}$ to $\vec{J}'$ through the layer of $J, J'$.

The contradiction in case (ii) requires slightly more work: edges 1 and 2 avoid bad 2-cubes. Assume as in case (iii) of Figure 10 that edge 3 is oriented upwards. Then edges $a$ and $b$ must have the same orientation. We must orient them to the left, as otherwise we get the forbidden subgrid (A). Choosing $c$ and $d$ such that bad 2-cubes are avoided results in the right 3-cube being the forbidden subgrid (B). So edge 3 must be oriented downwards.

Assume as in case (iv) that edge 4 is oriented upwards. Again, edges $a$ and $b$ must have the same orientation. In order to avoid (A), they are oriented to the left. Edge $c$ prevents the left 3-cube from having two sources and $d$ avoids a bad 2-cube. This forces the left 3-cube to be (B), which means that edge 4 must be oriented downwards.

Knowing the orientations of the edges 3 and 4 lets us fix edges 5-8 in case (v) (avoiding bad 2-cubes), resulting in the forbidden subgrid (C) for the 3-cube spanned by $\vec{J}$ and $\vec{J}'$. 

\[ \square \]
4.2 Higher-dimensional Grids

While we have shown in the previous subsection that a 3-dimensional grid is HK if we exclude the double twist as well as the non-HK cubes, this does not hold anymore in dimension 4.

The smallest counterexample is the orientation of the $(3 \times 2 \times 2 \times 2)$-grid depicted in Figure 11 (the big arrow indicates the common orientation of all edges between the left and right $3 \times 2 \times 2$ faces). The orientation itself is not HK (try to build two vertex-disjoint paths from source to sink, starting with edges 1 and 2), but all its faces are. In other words, this orientation is a minimal, non-cubical obstruction for the Holt-Klee condition in dimension $n = 4$. The example can be extended to yield a minimal non-cubical obstruction in any dimension $n \geq 4$.

An open question is whether there is a finite family of forbidden subgrids for given $n$ whose absence makes any $n$-dimensional grid-USO HK.

4.3 PGLCP-induced Orientations

We show in this section that the grid USO coming from PGLCP have the Holt-Klee property. We actually prove this for a larger class of digraphs, those defined by complete pointed fans. We will first show how the PGLCP case fits into this larger class, yielding interesting examples of non-polyhedral fans.

Following Section 2.1, we are given an $n \times N$ matrix $A$, an $N$-vector $c$ and a partition $\Pi$ of $[N]$ into $n$ nonempty subsets. We seek a $y \in \mathbb{R}^n$ so that

$$c^T \geq y^T A$$

(24)
and

\begin{equation}
\text{for every } i \in [n], \text{ there is some } j \in \Pi_i \text{ with } c_j = (y^T A)_j.
\end{equation}

The matrix \( A \) is assumed to have property P. This implies that there is a unique \( y \) satisfying conditions (24) and (25).

Let \( \hat{A} \) be an \((N-n) \times N\) matrix for which the row space is the orthogonal complement in \( \mathbb{R}^N \) of the row space of \( A \). A set of columns of \( \hat{A} \) indexed by a set \( K \) will be called \textit{complementary} if \(|\Pi_i \setminus K| \geq 1 \) for all \( i \in [n] \). A cone generated by a complementary set of columns of \( \hat{A} \) will be called a \textit{complementary cone}.

Note that the index set of a complementary set of \((N-n)\) columns is always the complement in \([N]\) of a \( \Pi \)-vertex. From property P and the construction of \( \hat{A} \) it follows that such a complementary set of columns of \( \hat{A} \) is linearly independent, so the complementary cones are all simplicial.

**Claim.** Fix \( A \) and \( c \). A vector \( y \in \mathbb{R}^n \) satisfies condition (25) if and only if \( q = \hat{A} c \) is a linear combination of the complementary set of columns of \( \hat{A} \) corresponding to the nonzero components of \( c - A^T y \). In particular, \( y \) satisfies (24) and (25) (i.e. \( y \) solves the PGLCP instance \((A,c)\)) if and only if \( q \) is in the complementary cone generated by the specific columns.

To see the claim, first note that \( \hat{A}(c - A^T y) = \hat{A} c = q \). Therefore, if \( y \) satisfies (25), the nonzero entries of \( c - A^T y \) correspond to a complementary set of columns. For the other direction, let \( x \) satisfy \( \hat{A} x = q \). We get \( \hat{A}(x - c) = q - q = 0 \), so \((x - c)^T\) is in the row space of \( A \) and can be expressed as \( y^T A \) for some \( y \). If in addition, the nonzero components of \( x \) correspond to a complementary set of columns of \( \hat{A} \), \( y \) satisfies (25).

This means that the PGLCP with \( A \) and \( c \) can equivalently be expressed dually as the problem of finding \( x \in \mathbb{R}^N \) such that \( \hat{A} x = q, x \geq 0 \), and with the complementarity property that for every \( i \in [n] \), there is some \( j \in \Pi_i \) with \( x_j = 0 \). We remark that the PGLCP in the version of Cottle & Dantzig [8] from Section 2.1 is of the latter form, with

\[ \hat{A} = (I, -A_{J}^{-1} A_{N \setminus J})^T, \]

where we assume that the last \( n \) columns of \( A \) are indexed by the distinguished set \( J \).

The existence of the PGLCP solution implies that the complementary cones cover \( \mathbb{R}^{N-n} \), and the uniqueness implies that the intersection of two complementary cones is a complementary cone. The complementary cones therefore form a \textit{complete pointed simplicial fan}. In proving our result for fans, we will replace, for the time being, the dimension \( N - n \) by \( d \). We follow the notation of Ziegler [47].

**Definition 4.2** A \textit{fan} in \( \mathbb{R}^d \) is a family

\[ \mathcal{F} = \{C_1, C_2, \ldots, C_t\} \]

of nonempty polyhedral cones, so that

(i) Every nonempty face of a cone in \( \mathcal{F} \) is also a cone in \( \mathcal{F} \).
(ii) The intersection of any two cones in $\mathcal{F}$ is a face of both.

A fan $\mathcal{F}$ is complete if the union of its cones is $\mathbb{R}^d$. It is pointed if the zero vector is one of its cones. From now on, we will use the term fan to denote a complete pointed fan. A $d \times N$ matrix $\hat{A}$ is said to realize a fan $\mathcal{F}$ if the columns of $\hat{A}$ are a set of generators for the 1-dimensional faces of $\mathcal{F}$.

The dual graph $G_{\mathcal{F}}$ of a fan has as its vertex set the set of $d$-dimensional cones of $\mathcal{F}$, with two cones joined by an edge if their intersection is a $(d-1)$-dimensional face of $\mathcal{F}$.

For our PGLCP application, note that two $(N-n)$-dimensional complementary cones meet in an $(N-n-1)$-dimensional complementary cone if and only if the complements of their index sets are adjacent in the graph $G$ defined in Section 2. Therefore this dual graph is isomorphic to the product of simplices introduced in Section 2.1.

A vector $q \in \mathbb{R}^d$ is said to be in general position with respect to $\mathcal{F}$ if it is not contained in any hyperplane that contains a $(d-1)$-dimensional cone of $\mathcal{F}$. If a vector $q$ is in general position with respect to a fan $\mathcal{F}$, we can define an orientation $\Gamma_{q,\mathcal{F}}$, in which an edge joining cones $C$ and $C'$ is oriented from $C$ to $C'$ if $C \cap C'$ and $q$ are on opposite sides of the hyperplane containing $C \cap C'$. The digraph $\Gamma_{q,\mathcal{F}}$ has a unique sink and source, which are the faces of $\mathcal{F}$ that contain $q$ and $-q$. The interior of a directed path in a digraph is the set of vertices in the path other than the first and the last vertex.

A generic vector $c$ from our PGLCP application yields a vector $q \in \mathbb{R}^d$ in general position by the formula $q = \hat{A}c$. It is an immediate consequence of the claim above that the grid orientation defined here is the same as that defined in Section 2.1 through the reduced costs in Equation (4). Here is the main theorem.

**Theorem 4.3** There is a set of $d$ directed paths in $\Gamma_{q,\mathcal{F}}$ from the source to the sink that have pairwise disjoint interiors.

The previous discussion immediately implies

**Corollary 4.4** Any PGLCP-induced grid USO $\psi$ satisfies the Holt-Klee condition.

The special case of the theorem in which $\mathcal{F}$ is the normal fan of a polytope was proved by Holt and Klee [26]. Our proof uses Menger’s theorem, which was employed by Holt and Klee, but does not use the geometry of a polytope. Our proof may be seen as an alternate way to prove the Holt-Klee theorem. The graph $\Gamma_{q,\mathcal{F}}$ has been used by Kleinschmidt and Onn to prove that fans are signable [32]. Restricted to PGLCP-induced fans, their result simply says that the undirected grid graph $\Gamma_{\mathcal{F}}$ underlying $\Gamma_{q,\mathcal{F}}$ has a unique sink orientation, namely $\Gamma_{q,\mathcal{F}}$. Theorem 4.3 strengthens the result of Kleinschmidt and Onn by showing that the signings they produce have an interesting additional property.

In our PGLCP application, the complementary cones form the normal fan of a polytope when there is a vector $b$ so that the LP: max $c^Ty$ subject to $Ax = b, x \geq 0$ is $\Pi$-compatible. In this case, the associated digraph $\Gamma_{q,\mathcal{F}}$ is acyclic. The following example, taken from Stickney and Watson [40], shows that the orientations associated with PGLCP properly generalize those from $\Pi$-compatible linear programs, because its digraph $\Gamma_{q,\mathcal{F}}$ contains a cycle.
Example 4.5 Let
\[ \tilde{A} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -2 \\ 0 & 1 & 0 & -2 & -1 & 0 \\ 0 & 0 & 1 & 0 & -2 & -1 \end{bmatrix}. \]
Set \( n = 3 \), choose \( \Pi_i = \{ i, i + 3 \} \) for \( i \in [n] \), and let \( \mathcal{F} \) be the collection of complementary cones of \( \tilde{A} \). Then \( \mathcal{F} \) is a fan in which the cones fit together like the faces of an octahedron. Let \( q = (1, 1, 1)^T \). The digraph \( \Gamma_{q,\mathcal{F}} \) is isomorphic to the one in Figure 2.

For more exotic examples of fans, see Chapter III of Ewald’s book [13]. We now return to the proof of the theorem. In the following, we assume that \( \mathcal{F} \) is a complete pointed fan in \( \mathbb{R}^d \) generated by the columns of a matrix \( \tilde{A} \), and that \( q \) is in general position with respect to \( \mathcal{F} \). The following is the key lemma.

Lemma 4.6 Let \( \mathcal{K} = \{ K_1, K_2, \ldots, K_{d-1} \} \) be \( d \)-dimensional cones of \( \mathcal{F} \), and suppose that none of these cones contains \( q \) or \(-q\). Then there exists a vector \( w \) orthogonal to \( q \) so that the segments from \( q \) to \( w \) and from \(-q\) to \( w \) both have empty intersection with each of the cones \( \mathcal{K} \).

Proof. We assume without loss of generality that \( q \) is the \( d \)-th unit vector \((0, \ldots, 0, 1)\). Let \( K \in \mathcal{K} \) and let \( \tilde{A}_K \) be the submatrix of \( \tilde{A} \) containing the generators of \( K \). Now let \( \overline{A}_K \) be the matrix obtained from \( \tilde{A}_K \) by deleting the last row. The columns of \( \overline{A}_K \) are the projections of the columns of \( \tilde{A}_K \) onto the hyperplane \( H_q \) orthogonal to \( q \). Because neither \( q \) nor \(-q\) is in \( K \), the systems \( \tilde{A}_K x = q, x \geq 0 \) and \( \tilde{A}_K x = -q, x \geq 0 \) have no solution. It follows that the system \( \overline{A}_K x = 0, x \geq 0, x \neq 0 \) has no solution. By Gordan’s Theorem [22] there exists a vector \( z_K \) so that \( z_K^T \overline{A}_K > 0 \). For such a \( z_K \), any sufficiently small perturbation of it will also satisfy the inequality. Therefore there we can find a linearly independent set \( \{ z_{K_1}, z_{K_2}, \ldots, z_{K_{d-1}} \} \) so that for each \( i \), \( z_{K_i}^T \overline{A}_K > 0 \). Let \( Z \) be a \((d-1) \times (d-1)\) matrix that has as its rows the vectors \( z_{K_i}^T, i = 1, \ldots, d-1 \). Stiemke’s Theorem [41] says that the system \( Z w \leq 0, Z w \neq 0 \) has a solution if and only if the system \( y^T Z = 0, y > 0 \) has no solution. But the matrix \( Z \) is nonsingular, so the second system has no solution. Therefore there must be a vector \( w \in H_q \) that is not in any of the cones \( \overline{A}_K \) for \( K \in \mathcal{K} \). This \( w \) is nonzero, because \( Z w \) is nonzero. The vector \( w \) satisfies the requirements of the lemma, because any intersection of a cone of \( \mathcal{K} \) with the segment from \( q \) to \( w \) or the segment from \(-q\) to \( w \) would project to the cone generated by \( w \). It should also be noted that since the cones \( \overline{A}_K \) are closed, there is an open set of such \( w \).

Now we are ready to prove the main theorem.

Proof (of Theorem 4.3). The directed vertex version of Menger’s theorem states that there will be \( d \) disjoint directed paths from the source of \( \Gamma_{q,\mathcal{F}} \) to the sink if and only if there do not exist \( d-1 \) vertices of the graph other than the source and the sink that cover all directed paths from the source to the sink. A set of \( d-1 \) vertices of \( \Gamma_{q,\mathcal{F}} \) other than the source and the sink corresponds to a set \( \mathcal{K} = \{ K_1, K_2, \ldots, K_{d-1} \} \) as in the Lemma. Because the set of \( w \) satisfying the conditions of the Lemma is open, we can choose a \( w \) for which the segments from \( w \) to \( q \) and from \( w \) to \(-q\) do not meet any cones of \( \mathcal{F} \) of
We claim that the sequence of \( d \)-dimensional cones met by the directed segment from \(-q\) to \(w\), followed by the sequence of \( d \)-dimensional cones met by the directed segment from \(w\) to \(q\), corresponds to a directed path from the source to the sink of \( \Gamma_{q,x} \). Suppose \( C_i \) and \( C_j \) are two \( d \)-dimensional cones of \( \mathcal{F} \), and that the directed segment from \(-q\) to \(w\) crosses, in order, \((C_i, C_i \cap C_j, C_j)\). Then \( C_i \setminus (C_i \cap C_j) \) and \( q \) are on opposite sides of the hyperplane spanned by \( C_i \setminus C_j \), so the edge of \( \Gamma_{q,x} \) connecting \( C_i \) and \( C_j \) is oriented from \( C_i \) to \( C_j \). Similarly, the edges connecting cones met by the directed segment from \(w\) to \(q\) are oriented consistently with the direction of the segment.

4.4 Inherited USO and Edge Flipping

In this part, we present two examples showing that in general, the Holt-Klee property is lost under the operations of taking inherited USO (Definition 2.9 and Lemma 2.11) and flipping edges (Theorem 2.12). For the inherited USO, consider the LP

\[
\begin{align*}
\max \quad & z = \frac{101}{100}x_1 + \frac{99}{100}x_3 + x_5 + \frac{1}{2}x_6 \\
\text{subject to} \quad & x_1 + x_2 = 1 \\
& 2x_1 + x_3 + x_4 = 3 \\
& 2x_1 + 2x_3 + x_5 + x_6 + x_7 = 9 \\
& x_1, \ldots, x_7 \geq 0.
\end{align*}
\]

Let \( n = 3 \), \( N = 7 \) and \( \Pi_1 = \{1, 2\}, \Pi_2 = \{3, 4\}, \Pi_3 = \{5, 6, 7\} \). Then the LP is seen to be \( \Pi \)-compatible and yields the unique sink orientation of Figure 3 which can be verified to satisfy the HK condition.

We have already seen that the choice of the hypervertices as in the figure generates an inherited orientation isomorphic to the double twist of Figure 7(1), which is not HK. Consequently, Corollary 4.4 tells us that inherited USO coming from PGLCP-induced USO are not necessarily PGLCP-induced.

For the edge flipping operation, consider the 4-cube orientation of Figure 12. Flipping all edges between the two 3-cubes in the figure leads to a USO that is not HK. Therefore, Corollary 4.4 together with Theorem 2.5 let us conclude that the HK-orientation of Figure 12 is not PGLCP-induced. We therefore have a new and purely combinatorial necessary condition for PGLCP-induced orientations: all \( 2^n \) reorientations arising from edge flips according to Theorem 2.5 must be HK. This condition is not sufficient: there is an earlier example of a 4-cube USO which is HK but not PGLCP-induced, although all its reorientations are HK as well [36].

5 Two-dimensional Grids

In this section we prove that no directed cycles can occur in a grid USO of dimension \( n = 2 \). Previously, this was only known to hold for USO of 2-dimensional grids that satisfy the Holt-Klee condition [44, 14].
Figure 12: An AUSO of the 4-cube satisfying the Holt-Klee condition. The arcs between the two depicted 3-cube facets go from the dark to the light vertices. Flipping the latter arcs leads to a USO not satisfying HK

Assuming that cycles are possible, let \( C \) be a shortest one. \( C \) will contain at most two vertices from each grid row and column, otherwise, a shortcut is possible. This implies that \( C \) alternates between vertical and horizontal edges, and that it has even length. This length must be at least 6, because we would get a cyclic 2-cube orientation otherwise, in direct contradiction to the USO property. After shuffling rows and columns, we may therefore assume that \( C \) has the form of a staircase, see Figure 13. In the following, we restrict the grid to the rows and columns (at least three each) touched by \( C \).

From the earlier refined index result (Theorem 2.14), we can deduce the existence of a vertex \( J \) with all horizontal incident edges incoming, and all vertical incident edges outgoing. \( J \) must be a vertex of \( C \), otherwise we would immediately find a shorter cycle by connecting \( C \) to its two closest staircase vertices in vertical and horizontal direction. By swapping rows and columns again, we may assume that \( J \) is the vertex formed by the second row and the third column of the grid, and subsequently we focus on the \( 3 \times 3 \)-subgrid with ‘center’ \( J \), see Figure 13.

Figure 13: The shortest cycle in a 2-dimensional grid and the subgrid defined by the special vertex \( J \)

By definition of \( J \), we already know the orientations of two off-staircase edges, see Figure 14 (left). We now claim that this partial orientation cannot be completed to a USO, yielding the desired contradiction. For this, we show that the USO properties
together with the assumption that $C$ is a shortest cycle determine a partial orientation that cannot be completed to a USO. Figure 14 (right) depicts the order in which we ‘fix’ the orientations.

![Diagram showing orientations](image)

**Figure 14: Contradiction in the subgrid**

In order to avoid bad 2-cubes, the edges labeled 1 and 2 must be oriented as in the figure. The orientations of edges 3 and 4 prevent shortcuts of the cycle that bypass $J$. Then, having edge 5 the other way around would (prematurely) fix edge 7 through a suitable 2-cube, resulting in a shortcut of $C$ along the boundary of the grid. Edge 6 keeps the first column cycle-free. The orientation of edge 7 then avoids a bad 2-cube again, and in order for the last row to remain acyclic, edge 8 must be as in the figure. The contradiction occurs in form of two sinks $K$ and $K'$ in the 2-cube spanned by them.

We have proved the following

**Theorem 5.1** *Any unique sink orientation of a 2-dimensional grid is acyclic.*

**Acknowledgment**

We thank Falk Tschirschnitz for many helpful discussions and David Loeffler for suggesting the simple proof of Lemma 3.4.

**References**


33


