Pivoting in Linear Complementarity: Two Polynomial-Time Cases

Jan Foniok[*†] Komei Fukuda[*♯♭] Bernd Gärtner[♯⊛]
Hans-Jakob Lüthi[*◇]

March 28, 2009

Abstract

We study the behavior of simple principal pivoting methods for the P-matrix linear complementarity problem (P-LCP). We solve an open problem of Morris by showing that Murty's least-index pivot rule (under any fixed index order) leads to a quadratic number of iterations on Morris's highly cyclic P-LCP examples. We then show that on K-matrix LCP instances, *all* pivot rules require only a linear number of iterations. As the main tool, we employ *unique-sink orientations* of cubes, a useful combinatorial abstraction of the P-LCP.

1 Introduction

The third author of this paper still vividly recollects his visit to Victor Klee at the University of Washington (Seattle) in August 2000.

> We had a memorable drive in Vic's car from the hotel to the math department. Vic skipped all the small talk and immediately asked: "Do you think that the simplex method is polynomial?" I still remember how awkward I felt to be asked this question by the very person who had provided the first and seminal insights into it. My (then as now quite irrelevant) opinion shall remain between Vic and me forever.

The seminal work referred to above is of course the 1972 paper of Klee and Minty entitled "How good is the simplex algorithm?" [19]. It deals with the number of iterations that may be required in the worst case to solve a linear program (LP) by Dantzig's

[*]ETH Zurich, Institute for Operations Research, 8092 Zurich, Switzerland
[†]foniok@math.ethz.ch
[♯]ETH Zurich, Institute of Theoretical Computer Science, 8092 Zurich, Switzerland
[♭]fukuda@ifor.math.ethz.ch
[⊛]gaertner@inf.ethz.ch
[◇]luethi@ifor.math.ethz.ch

simplex method [9], which was at that time the only available practical method. Klee and Minty exhibited a family of LPs (now known as the *Klee-Minty cubes*) for which the number of iterations is *exponential* in the number of variables and constraints.

Dantzig's specific *pivot rule* for selecting the next step is just one of many conceivable rules. The question left open by the work of Klee and Minty is whether there is some *other* rule that provably leads to a small number of iterations. This question is as open today as it was in 1972.

Nowadays, the simplex method is no longer the only available method to solve LPs. In particular, there *are* proven polynomial-time algorithms for solving LPs—Khachiyan's *ellipsoid method* [18] and Karmarkar's *interior point method* [17]—that are based on techniques developed originally for nonlinear optimization. It is still unknown, though, whether there is a *strongly* polynomial-time algorithm for solving LPs, that is, an algorithm for which the number of arithmetic operations does not depend on the bit lengths of the input numbers.

The P-matrix linear complementarity problem. In this paper we are concerned with pivoting methods for the *P-matrix linear complementarity problem* (P-LCP), a prominent problem for which neither polynomial-time algorithms nor hardness results are available.

In general, an LCP is given by a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$, and the problem is to find vectors $w, z \in \mathbb{R}^n$ such that

$$w - Mz = q, \quad w, z \geq 0, \quad w^T z = 0. \tag{1}$$

It is NP-complete to decide whether such vectors exist [7, 20]. However, in a P-LCP, M is a P-matrix (meaning that all principal minors are positive), and then there are unique solution vectors w, z for every right-hand side q [30]. The problem of efficiently finding them is unsolved, though.

Megiddo has shown that the P-LCP is unlikely to be NP-hard. For this, he considers the following (more general) variant: given M and q, either find w, z that satisfy (1), or exhibit a nonpositive principal minor of M. NP-hardness of the latter variant would imply that NP = coNP [23].

There is a different notion of hardness that might apply to the P-LCP as a member of the complexity class PPAD [28]. This class has complete problems, and no polynomial-time algorithm is known for any of them. It has recently been shown that the famous problem of finding a Nash equilibrium in a bimatrix game is PPAD-complete [6]. Some researchers consider this to be an explanation why (despite many efforts) no polynomial-time algorithm has been found so far. Incidentally, this is the *second* problem for which Megiddo showed in his technical report [23] that it is unlikely to be NP-hard. However, it is currently not known whether the P-LCP is also PPAD-complete.

There are various algorithms for solving P-LCPs, among them the classic method by Lemke [22] as well as interior point approaches [21, 20]. Still, there is no known polynomial-time algorithm for P-LCPs. For example, the complexity of interior point algorithms depends linearly on a matrix parameter κ [20] that is not bounded for P-matrices.

In this work, we focus on combinatorial methods for P-LCPs and in particular on *simple principal pivoting methods* that share their essential idea with the simplex method.

Simple principal pivoting methods. Let $B \subseteq \{1, 2, \ldots, n\}$, and let A_B be the $n \times n$ matrix whose ith column is the ith column of $-M$ if $i \in B$, and the ith column of the $n \times n$ identity matrix I_n otherwise. If M is a P-matrix, then A_B is invertible for every set B. We call B a *basis*. If $A_B^{-1}q \geq 0$, we have discovered the solution: let

$$w_i := \begin{cases} 0 & \text{if } i \in B \\ (A_B^{-1}q)_i & \text{if } i \notin B \end{cases}, \qquad z_i := \begin{cases} (A_B^{-1}q)_i & \text{if } i \in B \\ 0 & \text{if } i \notin B \end{cases}. \tag{2}$$

If on the other hand $A_B^{-1}q \not\geq 0$, then w and z defined above satisfy $w - Mz = q$ and $w^T z = 0$, but $w, z \geq 0$ fails. In *principal pivoting*, one tries to improve the situation by replacing the basis B with the symmetric difference $B \oplus C$, where C is some nonempty subset of the "bad indices" $\{i : (A_B^{-1}q)_i < 0\}$. The greedy choice is to let C be the set of all bad indices in every step. For some P-LCPs, however, this algorithm cycles and never terminates [27].[1] In *simple principal pivoting*, C is of size one, and a *pivot rule* is employed to select the bad index. Simple principal pivoting methods are sometimes called *Bard-type* methods and were first studied by Zoutendijk [34] and Bard [2]. For a detailed exposition on simple principal pivoting methods see, for instance, [8, Section 4.2] or [27, Chapter 4].

The digraph model and unique sink orientations. There is a natural digraph model behind principal pivoting, first studied by Stickney and Watson [31]. The graph's vertices are all bases B, with a directed edge going from B to B' if $B \oplus B' = \{i\}$ and $(A_B^{-1}q)_i < 0$.[2] The underlying undirected graph is the graph of the n-dimensional cube. A principal pivot step considers the cube that is spanned by the outgoing edges at the current vertex, and it jumps to the antipodal vertex of some subcube of it. A simple principal pivot step can be interpreted as the traversal of an outgoing edge, and this is the analogy with the simplex method for linear programming.

The above digraph has the following specific property: every nonempty subcube (including the whole cube) has a unique sink, and the global sink corresponds to the solution of the P-LCP [31]. In the terminology of Szabó and Welzl [32], we are dealing with a *unique-sink orientation* (USO).

The goal of this paper is to deepen the understanding of the USO description of simple principal pivoting methods for the P-LCP. On a general level, we want to understand whether and how algebraic properties of a P-LCP (which are well studied) translate to combinatorial properties of the corresponding USO (which are much less studied), and what the algorithmic implications of such translations are. The principal motivation

[1] For $n = 3$, one such example is the P-LCP (5) whose digraph model (see next paragraph) appears in Figure 1. Cycling occurs for any of the three start bases just below the top vertex in Figure 1.

[2] We require q to be nondegenerate, meaning that $(A_B^{-1}q)_i \neq 0$ for all B and i. This is no serious restriction since a nondegenerate problem can always be obtained from a degenerate one by a symbolic perturbation of q.

behind this research is the continuing quest for a polynomial-time P-LCP algorithm. In this paper, the concept of a USO serves as the main tool to obtain two more positive results for simple principal pivoting.

We believe that this combinatorial approach has some untapped potential for the theory of (simple) principal pivoting methods for LCPs.

The (randomized) method of Murty and the Morris orientations. The simple principal pivoting method of Murty (also known as the *least-index* rule) [25] works for all P-LCP instances, but it may take exponentially many iterations in the worst case [26]. As a possible remedy, researchers have considered *randomized* methods. The two most prominent ones are RandomizedMurty (which is just the least-index rule, after randomly reshuffling the indices at the beginning), and RandomEdge (which performs a purely random walk in the USO). However, Morris found a family of P-LCP instances (we call their underlying digraphs the *Morris orientations*) on which RandomEdge spends a long time running in cycles and thus performs much worse than even the exhaustive search algorithm [24].

For RandomizedMurty there are as yet no such negative results on P-LCP instances. In particular, on Murty's worst-case example, this algorithm takes an expected *linear* number of steps [11]; the expected number of steps becomes quadratic if we allow arbitrary start vertices [1].

On general P-LCP instances, the expected number of iterations taken by RandomizedMurty is *subexponential* if the underlying digraph is acyclic [14], but it is unknown whether this also holds under the presence of directed cycles (as they appear in the Morris orientations, for example).

We prove that not only the randomized variant, but actually *any* variant of the least-index rule obtained by some initial reshuffling of indices leads to a *quadratic* number of iterations on the Morris orientations. In particular, this "kills" another family of potentially bad instances for the RandomizedMurty rule.

K-matrix linear complementarity problems. A *K-matrix* is a P-matrix so that all off-diagonal elements are non-positive. LCPs with K-matrices (K-LCPs) appear for example in free boundary problems [8, Section 5.1] and in American put option pricing [3, 4]. It is known that every K-LCP instance can be solved in polynomial time (even by a simple principal pivoting method [5, 29], see also [8, Section 4.7]), but we prove something stronger: *every* simple principal pivoting method takes only a linear number of iterations. We obtain this result by showing that in K-LCP-induced USOs, *all* directed paths are short. Our approach is to extract combinatorial structure from the algebraic properties of K-matrices. Subsequently, we use the distilled combinatorial information to get our structural results.

LCPs with sufficient matrices and the criss-cross method. Let us step back and take a broader view that brings together LP, the P-LCP, pivoting methods, and computational complexity: linear complementarity problems with *sufficient* matrices generalize both

LP and the P-LCP while still being amenable to pivoting approaches. The *criss-cross method* [12, 13] may be considered an extension of Murty's algorithm. In fact, the results of [12] make LCPs with sufficient matrices the largest known class of LCPs for which Megiddo's techniques of [23] still apply, so that NP-hardness is unlikely. Determining the complexity of sufficient matrix LCP remains a tough challenge for the future.

2 Unique-sink orientations

Now we formally introduce the digraph model behind P-LCPs and show some of its basic properties. The model was first described by Stickney and Watson in 1978 [31].

We use the following notation. Let $[n] := \{1, 2, \ldots, n\}$. For a bit vector $v \in \{0,1\}^n$ and $I \subseteq [n]$, let $v \oplus I$ be the element of $\{0,1\}^n$ defined by

$$(v \oplus I)_j := \begin{cases} 1 - v_j & \text{if } j \in I, \\ v_j & \text{if } j \notin I. \end{cases}$$

This means that $v \oplus I$ is obtained from v by flipping all entries with coordinates in I. Instead of $v \oplus \{i\}$ we simply write $v \oplus i$.

Under this notation, the (undirected) *n-cube* is the graph $G = (V, E)$ with

$$V := \{0,1\}^n, \quad E := \{\{v, v \oplus i\} : v \in V,\ i \in [n]\}.$$

A *subcube* of G is a subgraph $G' = (V', E')$ of G where $V' = \{v \oplus I : I \subseteq C\}$ for some vertex v and some set $C \subseteq [n]$, and $E' = E \cap \binom{V'}{2}$.

2.1 Definition. Let ϕ be an orientation of the n-cube (a digraph with underlying undirected graph G). We call ϕ a *unique-sink orientation* (USO) if every nonempty subcube has a unique sink in ϕ.

Figure 1 depicts a USO of the 3-cube. The conditions in Definition 2.1 require the orientation to have a unique global sink, but in addition, all proper nonempty subcubes must have unique sinks as well. In Figure 1 there are six 2-dimensional subcubes, twelve 1-dimensional subcubes (edges) and eight 0-dimensional subcubes (vertices). For edges and vertices, the unique-sink condition is trivial. The figure also shows that USOs may contain directed cycles.

If V' is the vertex set of a subcube (as defined above), then the directed subgraph of ϕ induced by V' is denoted by $\phi[V']$.

It is convenient to view the orientation of the n-cube G as a mapping $\phi : \{0,1\}^n \to \{-,+\}^n$, where $\phi(v)_i = -$ if the edge between v and $v \oplus i$ is oriented towards $v \oplus i$, and $\phi(v)_i = +$ if it is oriented towards v. To encode a vertex v along with the orientations of its incident edges, we can then use a *configuration* of n bits and n signs, where the ith sign is $\phi(v)_i$ (see Figure 1). If an orientation ϕ of G contains the directed edge $(v, v \oplus i)$, we write $v \xrightarrow{\phi} v \oplus i$, or simply $v \to v \oplus i$ if ϕ is clear from the context.

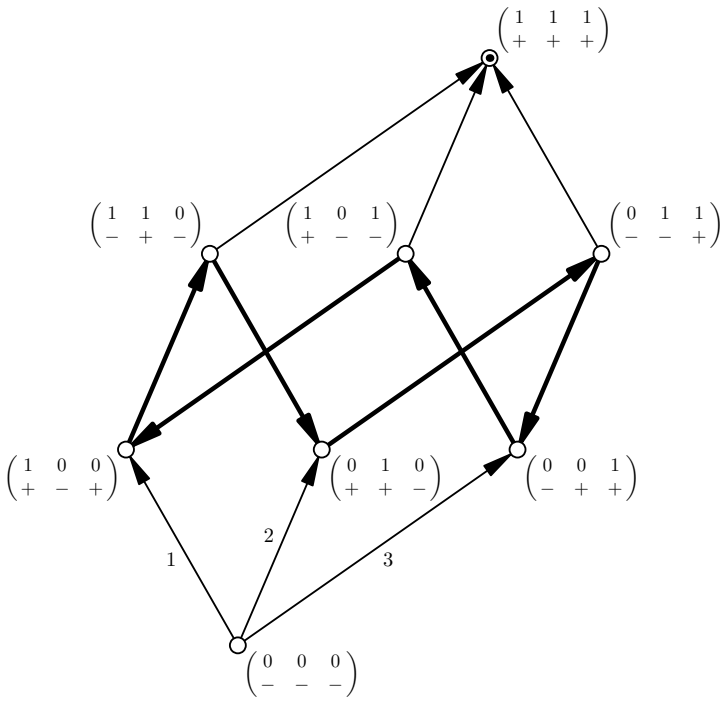

Figure 1: A USO of the 3-cube; for each vertex, the orientation of incident edges is encoded with a sign vector in $\{-,+\}^n$ ($-$ for an outgoing edge; $+$ for an incoming edge). The unique global sink and a directed cycle are highlighted.

Let ϕ be an orientation of the n-cube and let $F \subseteq [n]$. Then $\phi^{(F)}$ is the orientation of the n-cube obtained by reversing all edges in coordinates contained in F; formally

$$v \xrightarrow{\phi^{(F)}} v \oplus i \quad :\Leftrightarrow \quad \begin{cases} v \xrightarrow{\phi} v \oplus i & \text{if } i \notin F, \\ v \oplus i \xrightarrow{\phi} v & \text{if } i \in F. \end{cases}$$

2.2 Proposition ([32])**.** *If ϕ is a USO, then $\phi^{(F)}$ is also a USO for an arbitrary subset $F \subseteq [n]$.* □

This proposition can be proved by induction on the size of F; it essentially suffices to show that under $\phi^{(\{i\})}$, the whole cube still has a unique sink.

Now we can derive the *unique-completion property*. Its meaning is that we can prescribe the bits for some coordinates and the signs for the remaining coordinates, and in a USO there will always be a unique vertex that satisfies the prescription. In particular, it follows that any USO (and any of its subcubes) also has a *unique source*.

In fact, the unique-completion property characterizes unique-sink orientations.

2.3 Lemma. *An orientation ϕ of the n-cube G is a unique-sink orientation if and only if for every partition $[n] = A \cup B$ and every pair of maps $\alpha : A \to \{0,1\}$, $\beta : B \to \{-,+\}$*

there exists a unique vertex v such that

$$v_i = \alpha(i) \quad \textit{for all } i \in A, \tag{3a}$$
$$\phi(v)_i = \beta(i) \quad \textit{for all } i \in B. \tag{3b}$$

Proof. First suppose that ϕ is a USO. Let $U := \{u : u_i = \alpha(i) \text{ for all } i \in A\}$ and consider the subcube $\phi[U]$. Let $F := \{i \in B : \beta(i) = -\}$. As we have just noted, $\phi^{(F)}$ is a unique-sink orientation, and thus the subcube $\phi[U]$ has a unique sink v with respect to $\phi^{(F)}$. This vertex v is the only vertex that satisfies (3a) and (3b).

Conversely, let ϕ be an orientation of the n-cube which satisfies the unique-completion property and let $U := \{u \oplus I : I \subseteq C\}$. Set $B := C$ and $A := [n] \setminus C$, and let $\alpha(i) := u_i$ for $i \in A$ and $\beta(i) := +$ for $i \in B$. Then the unique vertex v satisfying (3a) and (3b) is the unique sink of the subcube $\phi[U]$. □

P-LCP-induced USOs. We now formally define the USO that is induced by a nondegenerate P-LCP instance P-LCP(M, q) [31]. For $v \in \{0,1\}^n$, let $B(v) := \{j \in [n] : v_j = 1\}$ be the canonical "set representation" of v. Then the unique-sink orientation ϕ induced by P-LCP(M, q) is

$$v \xrightarrow{\phi} v \oplus i \quad :\Leftrightarrow \quad (A^{-1}_{B(v)} q)_i < 0. \tag{4}$$

Recall that A_B is the (invertible) $n \times n$ matrix whose ith column is the ith column of $-M$ if $i \in B$, and the ith column of the $n \times n$ identity matrix I_n otherwise.

In this way, we have reduced the P-LCP to the problem of finding the sink in an implicitly given USO. Access to the USO is gained through a *vertex evaluation oracle* that for a given vertex $v \in \{0,1\}^n$ returns the orientations $\phi(v) \in \{-,+\}^n$ of all incident edges. In this setting, the aim is to find the sink of the USO so that the number of queries to the oracle would be bounded by a polynomial in the dimension n. In the case of a P-LCP-induced USO, a (strongly) polynomial-time implementation of the vertex evaluation oracle is immediate from (4). From the sink of ϕ, we can reconstruct solution vectors w and z as in (2). The fact that the n-cube orientation defined in this way is indeed a unique-sink orientation was proved by Stickney and Watson [31].

Not every USO is P-LCP-induced; for $n = 3$, the P-LCP-induced USOs are characterized in [31], but for $n \geq 4$, we have only a necessary condition. This condition was originally proved by Holt and Klee for polytope graphs oriented by linear functions [16].

2.4 Theorem ([15])**.** *Every P-LCP-induced USO of the n-cube satisfies the* Holt-Klee condition*, meaning that there are n directed paths from the source to the sink with pairwise disjoint sets of interior vertices.* □

3 Pivot Rules

In the USO setting resulting from P-LCPs, simple principal pivoting can be interpreted as follows: start from any vertex, and then proceed along outgoing edges until the global sink (and thus the solution to the P-LCP) is reached. A *pivot rule* R determines which

outgoing edge to choose if there is more than one option. Here is the generic algorithm, parameterized with the rule R. It outputs the unique sink of the given USO ϕ.

3.1 Algorithm.

SimplePrincipalPivoting$_{\mathrm{R}}(\phi)$
  Choose $v \in \{0,1\}^n$
  while $\mathcal{O} := \{j \in [n] : v \xrightarrow{\phi} v \oplus j\} \neq \emptyset$ **do**
    choose $i \in \mathcal{O}$ according to the pivot rule R
    $v := v \oplus i$
  end while
  return v

Note that when ϕ contains directed cycles as in Figure 1, this algorithm may enter an infinite loop for certain rules R, but we consider only rules for which this does not happen. In the USO setting we are restricted to *combinatorial* rules. These are rules that may access only the orientations of edges $\{v, v \oplus j\}$ (as given by the vertex evaluation oracle) but not any numerical values such as $(A^{-1}_{B(v)}q)_j$ that define these orientations.

Let us now introduce the combinatorial pivot rules that are of interest to us. We write them as functions of the set $\mathcal{O}$ of candidate indices. The first one is Murty's least-index rule Murty that simply chooses the smallest candidate. In the cube, this may be interpreted as a "greedy" approach that always traverses the first outgoing edge that it finds (in the order of the cube dimensions).

Murty$(\mathcal{O})$
  return $\min(\mathcal{O})$

Murty$_\pi(\mathcal{O})$
  return $\pi(\min(\pi^{-1}(\mathcal{O})))$

It is easy to prove by induction on n that this rule leads to a finite number of iterations, even if the USO contains directed cycles. The rule has a straightforward generalization, using a fixed permutation $\pi \in S_n$: choose the index $\pi(i)$ with the smallest i such that $v \to v \oplus \pi(i)$. This just reshuffles the cube dimensions by applying π, and with $\pi = \mathrm{id}$, we recover Murty's least-index rule.

The natural randomized variant of Murty$_\pi$ is RandomizedMurty: at the beginning, choose the permutation π uniformly at random from the set S_n of all permutations, and then use Murty$_\pi$ throughout.

Finally, RandomEdge stands for the plain random walk: in each iteration, simply choose a random candidate.

RandomizedMurty$(\mathcal{O})$
  if called for the first time **then**
    choose $\pi \in S_n$ uniformly at random
  end if
  return Murty$_\pi(\mathcal{O})$

RandomEdge$(\mathcal{O})$
  choose $i \in \mathcal{O}$ uniformly at random
  return i

Unlike the previous rules, RandomEdge may lead to cycling in SimplePrincipalPivoting$_{\mathrm{R}}$, but the algorithm still terminates with probability 1, and with an expected

number of at most $(n+1)n^n$ iterations. This is a simple consequence of the fact that there is always a short directed path to the sink.

3.2 Lemma ([31, Proposition 5]). *Let $t \in \{0,1\}^n$ be the global sink of an n-cube USO ϕ, and let $v \in \{0,1\}^n$ be any vertex. If k is the Hamming distance between v and t, then ϕ contains a directed path of length k from v to t.* □

4 The Morris orientations

Morris proved that under R = RandomEdge, Algorithm 3.1 may be forced to perform an expected *exponential* number of iterations [24]. More precisely, at least $((n-1)/2)!$ iterations are required on average to find the sink of the n-cube USO generated (as described in Section 2) by the LCP(M,q), with

$$M = \begin{pmatrix} 1 & 2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 2 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 2 \\ 2 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}, \qquad q = \begin{pmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \\ -1 \\ -1 \end{pmatrix}. \tag{5}$$

Here, n must be an odd integer in order for M to be a P-matrix.

To prove this result Morris first extracted the relevant structure of the underlying USO, and then showed that on this USO, RandomEdge runs in cycles for a long time before it finally reaches the global sink. It was left as an open problem to determine the expected performance of RandomizedMurty, the other natural randomized rule, on this particular USO.

We solve this open problem by showing that for *any* permutation π, Murty$_\pi$ incurs only $O(n^2)$ iterations on the n-dimensional Morris orientation. In particular, this bound then also holds for RandomizedMurty.

4.1 Theorem. *For any $\pi \in S_n$, Algorithm 3.1 with* R = Murty$_\pi$ *finds the sink of the n-dimensional Morris orientation after at most $2n^2 - (5n-3)/2$ iterations, from any start vertex.*

Computational experiments suggest that the worst running time is attained for the identity permutation if the algorithm starts from 0 (the global source). Therefore we analyze this case thoroughly and compute the precise number of iterations.

4.2 Theorem. *Algorithm 3.1 with* R = Murty$_{\mathrm{id}}$ *finds the sink of the n-dimensional Morris orientation after $(n^2+1)/2$ iterations, starting from 0.*

Experimental data suggest that this is an upper bound on the number of iterations for any permutation π and start vertex 0. Allowing start vertices different from 0, we

get slightly higher iteration numbers, but $\pi = \mathrm{id}$ continues to be the worst permutation. In view of this, we conjecture that $n^2/2 + O(n)$ iterations actually suffice for all π and all start vertices. This would mean that the bound in Theorem 4.1 is still off by a factor of 4.

The combinatorial structure. The n-cube USO resulting from (5) can be described in purely combinatorial terms [24, Lemma 1]. Here we make its structure even more transparent by exhibiting a *finite state transducer* (finite automaton with output) with just two states that can be used to describe the USO, see Figure 2. For the remainder of this section, we fix n to be an odd integer and we are considering the n-dimensional Morris orientation ϕ.

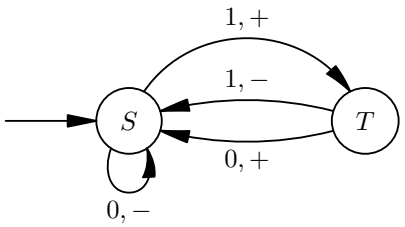

Figure 2: The finite state transducer that generates the Morris orientations. Each transition is labeled with an input symbol (bit) and an output symbol (sign)

The orientation is then determined for each vertex $v = (v_1, v_2, \ldots, v_n) \in \{0,1\}^n$ as follows. If $v_1 = v_2 = \cdots = v_n = 1$, then v is the global sink. Otherwise choose some i in $[n]$ so that $v_i = 0$ and consider the bit string $v_{i-1}v_{i-2}\ldots v_1 v_n v_{n-1}\ldots v_i$ as the input string to the transducer. In other words, the transducer is reading the input from right to left, starting immediately to the left of some 0. The output string $\phi(v)_{i-1}\phi(v)_{i-2}\ldots\phi(v)_1\phi(v)_n\phi(v)_{n-1}\ldots\phi(v)_i$ then determines the orientation of each edge incident to v. The choice of i does not matter, because after reading any zero, the transducer is in the state S.

For instance, let $n = 5$ and let $v = (1,0,1,1,0)$. Then the transducer takes $v_4 = 1$, outputs $+$ and switches to state T; reads $v_3 = 1$, outputs $-$ and switches to S; reads $v_2 = 0$, outputs $-$ and stays in S; reads $v_1 = 1$, outputs $+$ and switches to T; and finally reads $v_5 = 0$, outputs $+$ and switches to S. The resulting configuration for v is $\left(\begin{smallmatrix} 1 & 0 & 1 & 1 & 0 \\ + & - & - & + & + \end{smallmatrix}\right)$.

Figure 1 depicts the Morris orientation for $n = 3$. It can easily be checked that the above procedural definition of the orientation is equivalent to the one given in [24, Lemma 1]. In particular, we obtain a USO. It has the remarkable symmetry that the set of configurations is closed under cyclic shifts.

Pivoting. From the transducer we can easily derive the sign changes in the configuration $\left(\begin{smallmatrix} v \\ \phi(v) \end{smallmatrix}\right)$ that are caused by a pivot step $v \to u := v \oplus \{i\}$. We always have $\phi(v)_i = -$ and $\phi(u)_i = +$.

If $v_i = 1$, we get $u_i = 0$. In processing v and u, the transducer then performs exactly the same steps, except that at the ith coordinate different transitions are used to get from T to S. But this implies $\phi(u)_j = \phi(v)_j$ for $j \neq i$. Here is an example for such a pivot step (with $i = 3$; affected signs are indicated):

$$\left(\begin{smallmatrix} 1 & 0 & 1 & 1 & 0 \\ + & - & - & + & + \end{smallmatrix}\right) \to \left(\begin{smallmatrix} 1 & 0 & 0 & 1 & 0 \\ + & - & \oplus & + & + \end{smallmatrix}\right) \tag{6}$$

If $v_i = 0$, we get $u_i = 1$, meaning that at the ith coordinate the transducer stays in S for v but switches to T for u (assuming that u is not the sink). This implies that signs change at all coordinates $i-1$, $i-2$ down to (and including) the next coordinate k (wrap around possible) for which $v_k = u_k = 0$. In our example we have such a step for $i = 2$:

$$\left(\begin{smallmatrix} 1 & 0 & 1 & 1 & 0 \\ + & - & - & + & + \end{smallmatrix}\right) \to \left(\begin{smallmatrix} 1 & 1 & 1 & 1 & 0 \\ \ominus & \oplus & - & + & \ominus \end{smallmatrix}\right) \tag{7}$$

In both v and u, the signs in the block of 1's at indices $k+1, \ldots, i-1$ alternate.

Levels. It will be useful to define the following function on the vertices of the cube. Let $\ell(v)$ be the number of coordinates where $\binom{0}{-}$ appears. Formally,

$$\ell(v) := |\{i : v_i = 0 \text{ and } \phi(v)_i = -\}|.$$

The number $\ell(v)$ is called the *level* of v. From the two types of pivots, it is easy to see that the value of ℓ does not increase along any directed edge [33, Lemma 3]. Indeed, if u is an out-neighbor of v, then either $\ell(u) = \ell(v)$ (this happens in every pivot of type (6), and in pivots of type (7) with an odd block of 1's at indices $k+1, \ldots, i-1$), or ℓ decreases to the next possible value: $\ell(u) = \ell(v) - 2$, or $\ell(u) = 0$ if $\ell(v) = 1$. In particular, the values of ℓ lie in the set $\{n, n-2, \ldots, 3, 1, 0\}$. The sink is the only vertex at level 0.

Let us make a small digression and briefly explain why RandomEdge is slow on ϕ. Let $L(v) := |\{i : v_i = 1 \text{ and } \phi(v)_i = -\}|$. So the outdegree of v is $\ell(v) + L(v)$. Now let v be a vertex at level 1. In pivots of type (6), L decreases by one, whereas in pivots of type (7) that do not reach the sink, L increases by one. The latter occurs if and only if RandomEdge pivots at the unique $\binom{0}{-}$, which happens with probability $1/(L(v)+1)$.

At level 1, we can thus interpret RandomEdge as a random walk on the integers, where $k \to k+1$ with probability $1/(k+1)$, and $k \to k-1$ with probability $k/(k+1)$. Therefore the walk is strongly biased towards 0, but in order to reach a neighbor u of the sink, we need to get to $k = L(u) = (n-1)/2$. This takes exponentially long in expectation [24].

A quadratic bound for Murty$_\pi$. To prove Theorem 4.1 we first derive a (somewhat surprising) more general result: for *any* pivot rule, starting from any vertex v with $v_k = 0$ for some k, it takes $O(n^2)$ iterations to reach a vertex u with $u_k = 1$. This can be viewed as a statement about the USO induced by the n-dimensional Morris orientation on the cube facet $\{v : v_k = 0\}$. The statement is that in this induced USO, all directed paths are short (in particular, the induced USO is acyclic).

4.3 Lemma. *Let $v \in \{0,1\}^n$ be a vertex, and let k be any index for which $v_k = 0$. Starting from v, Algorithm 3.1 (with an arbitrary pivot rule* R*) reaches a vertex u satisfying $u_k = 1$ after at most $\binom{n+1}{2}$ iterations.*

Proof. We employ a nonnegative potential that decreases with every pivot step. By a cyclic shift of coordinates we may assume that $k = 1$, and thus $v_1 = 0$.

Let us define

$$N_0(v) := \{j \in [n] : v_j = 0 \text{ and } \phi(v)_j = -\},$$
$$N_1(v) := \{j \in [n] : v_j = 1 \text{ and } \phi(v)_j = -\}.$$

The *potential* of v is the number

$$p(v) = |N_1(v)| + \sum_{j \in N_0(v)} j.$$

Now we will observe how the potential changes during a pivot step $v \to u = v \oplus \{i\}$. We always have $\phi(v)_i = -$, $\phi(u)_i = +$. There are two cases. If $v_i = 1$, as in (6), the configurations of v and u differ only at index i. Hence $N_0(u) = N_0(v)$ and $N_1(u) = N_1(v) \setminus \{i\}$. Therefore $p(u) = p(v) - 1$.

In the other case we have $v_i = 0$, as in (7). If $i = 1$, we are done, since then $u_1 = 1$. Otherwise, v is of the form

$$v = (v_1, \ldots, \underbrace{v_j}_{0}, \underbrace{v_{j+1}, \ldots, v_{i-1}}_{1,\ldots,1}, \underbrace{v_i}_{0}, v_{i+1}, \ldots, v_n),$$

with $1 \le j < i$. If $i-j$ is odd, there is an even number of 1's between v_j and v_i, implying that $|N_1(u)| = |N_1(v)|$ and $N_0(u) = N_0(v) \setminus \{j, i\}$. We then have $p(u) < p(v)$. If $i - j$ is even, there is an odd number of 1's between v_j and v_i, implying that $|N_1(u)| = |N_1(v)|+1$ and $N_0(u) = N_0(v) \cup \{j\} \setminus \{i\}$. Since $i - j \ge 2$ in this case, we get $p(u) \le p(v) - 1$.

To summarize, the potential decreases by at least 1 in every pivot step $v := v \oplus i$, and as long as v is not the sink, we have $p(v) \ge 1$. The highest potential is attained by the source, $p(0) = \binom{n+1}{2}$. Therefore after at most $\binom{n+1}{2}$ steps we have $v_1 = 1$. □

If v happens to be at level 1 already, we obtain a better bound, since $|N_0(v)| = 1$ in this case. In fact, the vertex with largest potential in level 1 such that $v_1 = 0$ is $v = \left(\begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 0 \\ + & + & - & + & - & \cdots & + & - & + & - \end{smallmatrix}\right)$, and its potential is $p(v) = 3(n-1)/2$.

4.4 Corollary. *Let $v \in \{0,1\}^n$ be a vertex at level 1, and let k be any index for which $v_k = 0$. Starting from v, Algorithm 3.1 (with arbitrary pivot rule* R*) reaches a vertex u satisfying $u_k = 1$ after at most $3(n-1)/2$ iterations.* □

Now MURTY$_\pi$ comes in: we will apply either Lemma 4.3 or Corollary 4.4 to vertices v with $v_{\pi(i)} = 0$ and $v_{\pi(i+1)} = \cdots = v_{\pi(n)} = 1$, for some i. The next lemma shows that the 1's at coordinates $\pi(i+1), \ldots, \pi(n)$ will stay.

4.5 Lemma. *Let $i \in [n]$. After Algorithm 3.1 with* R = MURTY$_\pi$ *visits a vertex v with $v_{\pi(i+1)} = \cdots = v_{\pi(n)} = 1$, it will visit only vertices u satisfying $u_{\pi(i+1)} = \cdots = u_{\pi(n)} = 1$.*

Proof. At every iteration, the algorithm replaces the current vertex v with $v \oplus \pi(j)$, where j is the smallest index such that $v \to v \oplus \pi(j)$. It follows that coordinates $\pi(i+1), \dots, \pi(n)$ will be touched only if $v \oplus \pi(j) \to v$ for all $j \le i$. When this holds for the first time, we have

$$\begin{aligned} v_{\pi(j)} &= 1, \quad \text{for } j > i, \\ \phi(v)_{\pi(j)} &= +, \quad \text{for } j \le i. \end{aligned}$$

The unique-completion property (Lemma 2.3) then implies that we have already reached the sink. □

Now we can put it all together.

Proof of Theorem 4.1. Let v be the start vertex, and let i be the largest index such that $v_{\pi(i)} = 0$ (if no such index exists, v is the sink, and we are done). Lemmas 4.3 and 4.5 show that after at most $\binom{n+1}{2} - 1$ iterations, we additionally have $\phi(v)_{\pi(1)} = \phi(v)_{\pi(2)} = \dots = \phi(v)_{\pi(i-1)} = +$ (since the next pivot step flips $v_{\pi(i)}$). At this point v is at level 1, since there is a unique $\binom{0}{-}$: at coordinate $\pi(i)$. After the flip at coordinate $\pi(i)$, we repeatedly apply Corollary 4.4 together with Lemma 4.5 to also produce 1's at coordinates $\pi(i-1), \dots, \pi(i-2), \dots, \pi(1)$, at which point we have reached the sink. This takes at most $3(i-1)(n-1)/2$ more iterations, summing up to a total of

$$\binom{n+1}{2} + 3(i-1)(n-1)/2 \le \binom{n+1}{2} + 3(n-1)(n-1)/2 = 2n^2 - (5n-3)/2.$$

□

A better bound in the case of the identity permutation can be achieved by thoroughly examining the run of the algorithm, as we do next.

The identity permutation. Let us now go on to prove Theorem 4.2. For this, we first identify certain "milestone" vertices that are visited by Algorithm 3.1 with $\text{R} = \text{MURTY}_{\text{id}}$, and then we count the number of iterations to get from one milestone to the next. Let us define

$$v^i := (\underbrace{0,1,0,1\dots,0,1}_{2i},0,0,\dots,0), \quad i = 0,\dots,(n-1)/2.$$

4.6 Lemma. *If Algorithm 3.1 with* $\text{R} = \text{MURTY}_{\text{id}}$ *starts at the vertex* v^i, $0 \le i < (n-1)/2$, *it gets to the vertex* v^{i+1} *in* $4i+3$ *iterations.*

Proof. The algorithm starts at $\left(\begin{smallmatrix} 0 & 1 & 0 & 1 & \cdots & 0 & 1 & \quad 0 & 0 & \cdots \\ + & + & + & + & \cdots & + & + & \quad - & - & \cdots \end{smallmatrix}\right)$, and then proceeds by the rules of pivoting (6) and (7) in four phases (recall that MURTY_{id} always pivots on the left-most $-$):

- It pivots on the coordinates $2i+1$, $2i-1$, …, 1 and thus reaches the vertex $\left(\begin{smallmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & \quad 1 & 0 & \cdots \\ + & - & + & - & \cdots & + & - & \quad + & - & \cdots \end{smallmatrix}\right)$ after $i+1$ iterations.

- It pivots on the coordinates 2, 4, 6, ..., $2i$, $2i+2$ and thus reaches the vertex $\left(\begin{smallmatrix}1&0&1&0&\cdots&1&0&&1&1&\cdots\\+&+&+&+&\cdots&+&-&&-&+&\cdots\end{smallmatrix}\right)$ after $i+1$ iterations;
- It pivots on the coordinates $2i$, $2i-2$, ..., 2 and thus reaches the vertex $\left(\begin{smallmatrix}1&1&1&1&\cdots&1&1&&1&1&\cdots\\-&+&-&+&\cdots&-&+&&-&+&\cdots\end{smallmatrix}\right)$ after i iterations;
- It pivots on the coordinates 1, 3, 5, ..., $2i+1$ and thus reaches the vertex $\left(\begin{smallmatrix}0&1&0&1&\cdots&0&1&&0&1&\cdots\\+&+&+&+&\cdots&+&+&&+&+&\cdots\end{smallmatrix}\right) = v^{i+1}$ after $i+1$ iterations.

Therefore it takes $4i+3$ iterations to get from v^i to v^{i+1}. □

The previous lemma allows us to count the number of iterations from $0 = v^0$ to $v^{(n-1)/2}$. The following lemma takes care of the remaining iterations.

4.7 Lemma. *If Algorithm 3.1 with* $\mathrm{R} = \textsc{Murty}_{\mathrm{id}}$ *starts at the vertex* $v^{(n-1)/2}$*, it gets to the global sink in* $(n+1)/2$ *iterations.*

Proof. The proof is similar to the proof of Lemma 4.6, except that here only the first phase takes place: if the algorithm starts at the vertex $\left(\begin{smallmatrix}0&1&0&1&\cdots&0&1&&0\\+&+&+&+&\cdots&+&+&&-\end{smallmatrix}\right)$, it pivots on coordinates n, $n-2$, ..., 1 and thus reaches the sink after $(n+1)/2$ iterations. □

Proof of Theorem 4.2. By Lemmas 4.6 and 4.7, the overall number of iterations from the source to the sink is

$$\sum_{i=0}^{(n-3)/2} (4i+3) + \frac{n+1}{2} = \frac{n^2+1}{2}.$$ □

5 K-matrix LCP

In this section we examine *K-LCPs*, the linear complementarity problems $\mathrm{LCP}(M,q)$ where M is a K-matrix.

5.1 Definition. A *K-matrix* is a P-matrix so that all off-diagonal entries are non-positive.[3]

We introduce two simple combinatorial conditions on unique-sink orientations and prove that one of them implies that all directed paths from 0 are short, and the other one implies that all directed paths in the orientation are short. We show that K-LCP orientations satisfy these conditions and conclude that a simple principal pivoting method with an arbitrary pivot rule solves a K-LCP in linearly many iterations, starting from any vertex.

First, the *uniform orientation* is the USO in which $v \to v \oplus i$ if and only if $v_i = 0$ (in other words, all edges are oriented "from 0 to 1").

5.2 Definition. A unique-sink orientation ϕ is *2-up-uniform* if whenever $U = \{u \oplus I : I \subseteq \{i,j\}\}$ is such that $u_i = u_j = 0$ and u is the source of U, then the orientation of the subcube $\phi[U]$ is uniform (see Figure 3).

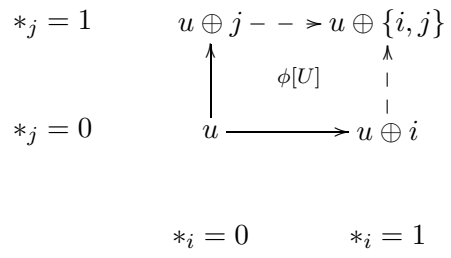

Figure 3: 2-up-uniformity: The orientations of the solid edges imply the orientations of the dashed edges. The top row contains vertices with jth entry equal to 1 and the bottom row vertices with jth entry equal to 0; the left column contains vertices with ith entry equal to 0, and the right column contains vertices with ith entry equal to 1.

A *K-USO* is a unique-sink orientation that arises (as described in Section 2) from the LCP(M, q) with some K-matrix M and some right-hand side q.

5.3 Proposition. *Every K-USO is 2-up-uniform.*

Proof. The essential fact we use is that a subcube of a K-USO is also a K-USO, as was observed already by Stickney and Watson [31, Lemma 1]. Thus it suffices to prove 2-up-uniformity for K-LCPs of dimension 2. Hence suppose that $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is a 2×2 K-matrix, so $a, d > 0$ and $b, c \leq 0$. Then $M^{-1} = \left(\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right) / (ad - bc) \geq 0$ and nonsingular. Now 0 is the source of the orientation induced by the LCP(M, q) if and only if $q < 0$. Hence $-M^{-1}q > 0$, which proves uniformity. □

One particular nice property of a 2-up-uniform orientation is that all paths from the vertex 0 to the sink t have the shortest possible length, equal to the Hamming distance $|t|$ of t from 0. To prove this statement, we use the following lemma.

5.4 Lemma. *Let ϕ be a 2-up-uniform USO and let v be a vertex such that $v_i = 1$ and $v \oplus i \xrightarrow{\phi} v$. If $v_j = 0$ and $u = v \oplus j$ is an out-neighbor of v, then $u \oplus i \xrightarrow{\phi} u$.*

Proof. Suppose for the sake of contradiction that $v_i = 1$, $v_j = 0$ and $v \oplus i \to v$, and that $u = v \oplus j$, $v \to u$ and $u \to u \oplus i$. Let $U := \{u \oplus I : I \subseteq \{i, j\}\}$ and consider the subcube $\phi[U]$ (see Figure 4, left): we observe that $v \oplus i \to u \oplus i$ because otherwise the subcube would not contain a sink. But then it follows from 2-up-uniformity of ϕ that $u \oplus i \to u$, a contradiction. □

5.5 Proposition. *Let ϕ be a 2-up-uniform USO and let t be its sink. Then any directed path from 0 to t has length $|t|$.*

[3] Another common name in the literature for a K-matrix is *Minkowski matrix*.

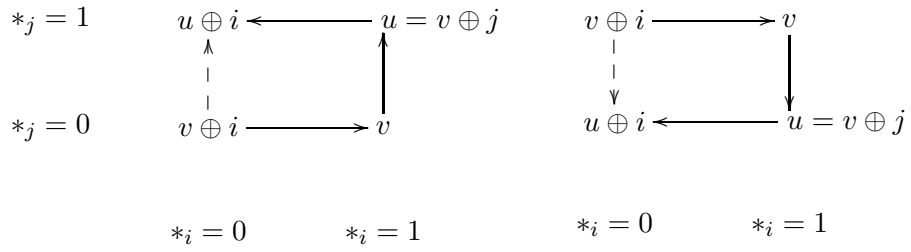

Figure 4: Illustrating the proof of Lemma 5.4 (left) and Lemma 5.11 (right).

Proof. It follows from Lemma 5.4 by induction that if u is a vertex on any directed path starting from 0, and if $u_i = 1$, then $u \oplus i \to u$. Hence all edges on a directed path from 0 to t are oriented from 0 to 1, and therefore the length of a directed path from 0 to t is $|t|$. □

As a direct consequence of Propositions 5.3 and 5.5 we get the following theorem.

5.6 Theorem. *Every directed path from 0 to the sink t of a K-USO has length $|t| \leq n$.* □

What is the actual strength of this theorem? We know that from any vertex of a unique-sink orientation there *exists* a path to the sink of length at most n (Lemma 3.2). It has also long been known how to find such a path from the vertex 0 in the case of a K-USO [5, 29]. The novelty is that *any* directed path starting from 0 reaches the sink after the least possible number of iterations. Hence a simple principal pivoting method with an *arbitrary* pivot rule finds the solution to a K-LCP in at most n iterations.

In view of this result, it is natural to ask whether it also holds if the path starts in some vertex different from 0. Imposing 2-up-uniformity is insufficient, as we can observe by looking at the Morris orientations. If we swap 0's and 1's in any Morris orientation, we get a 2-up-uniform orientation. Hence all directed paths from 0 to the sink are short; indeed, 0 is the sink of the orientation after swapping, so all such paths have length 0. Nevertheless, long directed paths (and even directed cycles) do exist in this orientation. Therefore we introduce the following stronger combinatorial property of USOs, which turns out to be satisfied by K-USOs as well.

5.7 Definition. A unique-sink orientation ϕ is *2-uniform* if it is 2-up-uniform and the orientation $\phi^{([n])}$ constructed from ϕ by reversing all edges is also 2-up-uniform.

5.8 Proposition. *Every K-USO is 2-uniform.*

Proof. If ϕ is induced by the LCP(M, q) for some K-matrix M, then $\phi^{([n])}$ is induced by the LCP$(M, -q)$, hence it is also a K-USO and therefore it is 2-up-uniform. □

Surprisingly, the simple property of being 2-uniform enforces a lot of structure on the whole orientation, as we show next.

5.9 Theorem. *The length of every directed path in a 2-uniform USO is at most $2n$.*

Before we prove the theorem, let us point out an important corollary.

5.10 Corollary. *Algorithm 3.1 with an arbitrary pivot rule* R*, starting at an arbitrary vertex of the corresponding USO, finds the solution to an n-dimensional K-LCP in at most $2n$ iterations.* □

As a consequence, we get a result for a larger class of LCPs. If M is a *principal pivotal transform* [8, Section 2.3] of a K-matrix M', then M is a P-matrix [8, Theorem 4.1.3], and the USO ϕ induced by the LCP(M, q) is isomorphic to the USO ϕ' induced by the LCP(M', q') for suitable q'. It follows that Corollary 5.10 also applies to the LCP(M, q), even though M is not necessarily a K-matrix itself.

Our proof of Theorem 5.9 is based on the following crucial fact, which extends Lemma 5.4. Informally, it asserts that once we have a $\binom{1}{+}$ in some coordinate, we will always have a $\binom{1}{+}$.

5.11 Lemma. *Let ϕ be a 2-uniform USO and let v be a vertex such that $v_i = 1$ and $v \oplus i \xrightarrow{\phi} v$. If $u = v \oplus j$ is an out-neighbor of v, then $u \oplus i \xrightarrow{\phi} u$.*

Proof. If $v_j = 0$, the claim follows from Lemma 5.4. So let us suppose that we have $v_i = 1$, $v_j = 1$ and $v \oplus i \to v$, and that $u = v \oplus j$, $v \to u$ and $u \to u \oplus i$. Let $U := \{u \oplus I : I \subseteq \{i, j\}\}$. We observe that $v \oplus i \to u \oplus i$ so that the subcube $\phi[U]$ contains a sink. But then it follows from 2-up-uniformity of $\phi^{([n])}$ that $v \to v \oplus i$, a contradiction (see Figure 4, right). □

Proof of Theorem 5.9. Let i be a fixed coordinate. Suppose in the considered directed path there is an edge $v \oplus i \to v$ such that $v_i = 1$. As a consequence of Lemma 5.11, all vertices u on a directed path starting at v satisfy $u_i = 1$. It follows that in any directed path no more than two edges appear of the form $v \oplus i \to v$ for any fixed i: possibly one with $v_i = 0$ and one with $v_i = 1$. This fact then implies that the length of any directed path is at most $2n$. □

The strength of 2-uniformity can perhaps be explained by its being equivalent to what we call *local uniformity*. A USO ϕ is *locally up-uniform* if for every $U = \{u \oplus I : I \subseteq J\}$ such that $u_J = 0$ and $u \xrightarrow{\phi} u \oplus j$ for all $j \in J$, the orientation of the subcube $\phi[U]$ is uniform. A USO ϕ is *locally uniform* if both ϕ and $\phi^{([n])}$ are locally up-uniform.

5.12 Proposition. *Let ϕ be a USO.*

(i) ϕ is locally up-uniform if and only if it is 2-up-uniform.

(ii) ϕ is locally uniform if and only if it is 2-uniform.

Proof. (ii) is easily implied by (i); and one implication of (i) is trivial. The remaining implication is not difficult to prove by induction on the dimension of the considered subcube. □

Local up-uniformity allows for a slight improvement in solving a K-LCP. When following a directed path starting in 0, instead of traversing one edge at a time we can perform a "greedy" principal pivot: jump straight to the sink of the subcube induced by outgoing edges. Unlike the general P-matrix case, it cannot lead to cycling. Due to local up-uniformity, it will never increase the number of iterations and may sometimes reduce it.

Locally uniform USOs vs. K-USOs. We have shown (Propositions 5.8 and 5.12) that every K-USO is 2-uniform (equivalently, locally uniform). It is natural to ask whether the converse is also true.

The answer to this question is negative. It can be shown that the number of n-dimensional locally uniform USOs is asymptotically much larger than the number of K-USOs. A lower bound of $2^{\binom{n-1}{\lfloor (n-1)/2 \rfloor}}$ on the number of locally uniform USOs can be derived by an adaptation of Develin's construction [10] of many orientations satisfying the Holt-Klee condition. An upper bound of $2^{O(n^3)}$ on the number of P-USOs (and thus also K-USOs) follows from Develin's proof of his upper bound on the number of LP-realizable cube orientations.

At present, however, we do not know whether locally uniform P-USOs form a proper superclass of the class of K-USOs.

References

[1] J. Balogh and R. Pemantle. The Klee-Minty random edge chain moves with linear speed. *Random Structures Algorithms*, 30(4):464–483, 2007.

[2] Y. Bard. An eclectic approach to nonlinear programming. In *Optimization (Proc. Sem., Austral. Nat. Univ., Canberra, 1971)*, pages 116–128, St. Lucia, 1972. Univ. Queensland Press.

[3] A. Borici and H.-J. Lüthi. Pricing American put options by fast solutions of the linear complementarity problem. In E. J. Kontoghiorghes, B. Rustem, and S. Siokos, editors, *Computational Methods in Decision-Making, Economics and Finance*, volume 74 of *Applied Optimization*. Kluwer Acad. Publ., Dordrecht, 2002.

[4] A. Borici and H.-J. Lüthi. Fast solutions of complementarity formulations in American put pricing. *J. Comput. Finance*, 9:63–82, 2005.

[5] R. Chandrasekaran. A special case of the complementary pivot problem. *Opsearch*, 7:263–268, 1970.

[6] X. Chen and X. Deng. Settling the complexity of two-player Nash-equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, 2006.

[7] S.-J. Chung. NP-completeness of the linear complementarity problem. *J. Optim. Theory Appl.*, 60(3):393–399, 1989.

[8] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Computer science and scientific computing. Academic Press, 1992.

[9] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[10] M. Develin. LP-orientations of cubes and crosspolytopes. *Adv. Geom.*, 4(4):459–468, 2004.

[11] K. Fukuda and M. Namiki. On extremal behaviors of Murty's least index method. *Math. Programming, Ser. A*, 64(3):365–370, 1994.

[12] K. Fukuda, M. Namiki, and A. Tamura. EP theorems and linear complementarity problems. *Discrete Appl. Math.*, 84(1–3):107–119, 1998.

[13] K. Fukuda and T. Terlaky. Linear complementarity and oriented matroids. *J. Oper. Res. Soc. Japan*, 35(1):45–61, 1992.

[14] B. Gärtner. The random-facet simplex algorithm on combinatorial cubes. *Random Structures Algorithms*, 20(3):353–381, 2002.

[15] B. Gärtner, W. D. Morris, Jr., and L. Rüst. Unique sink orientations of grids. *Algorithmica*, 51(2):200–235, 2008.

[16] F. Holt and V. Klee. A proof of the strict monotone 4-step conjecture. In *Advances in discrete and computational geometry*, volume 223 of *Contemp. Math.*, pages 201–216. Amer. Math. Soc., 1999.

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

[18] L. G. Khachiyan. Polynomial algorithms in linear programming. *U.S.S.R. Comput. Math. and Math. Phys.*, 20:53–72, 1980.

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

[20] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A unified approach to interior point algorithms for linear complementarity problems*, volume 538 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1991.

[21] M. Kojima, N. Megiddo, and Y. Ye. An interior point potential reduction algorithm for the linear complementarity problem. *Math. Programming, Ser. A*, 54(3):267–279, 1992.

[22] C. E. Lemke. Recent results on complementarity problems. In *Nonlinear Programming*, pages 349–384. Academic Press, New York, 1970.

[23] N. Megiddo. A note on the complexity of P-matrix LCP and computing an equilibrium. RJ 6439, IBM Research, Almaden Research Center, 650 Harry Road, San Jose, California, 1988.

[24] W. D. Morris, Jr. Randomized pivot algorithms for P-matrix linear complementarity problems. *Math. Program., Ser. A*, 92(2):285–296, 2002.

[25] K. G. Murty. Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, 11(2–3):123–130, 1974.

[26] K. G. Murty. Computational complexity of complementary pivot methods. *Math. Programming Stud.*, 7:61–73, 1978.

[27] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*, volume 3 of *Sigma Series in Applied Mathematics*. Heldermann, Berlin, 1988.

[28] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.*, 48(3):498–532, 1994.

[29] R. Saigal. A note on a special linear complementarity problem. *Opsearch*, 7:175–183, 1970.

[30] H. Samelson, R. M. Thrall, and O. Wesler. A partition theorem for Euclidean n-space. *Proc. Amer. Math. Soc.*, 9:805–807, 1958.

[31] A. Stickney and L. Watson. Digraph models of Bard-type algorithms for the linear complementarity problem. *Math. Oper. Res.*, 3(4):322–333, 1978.

[32] T. Szabó and E. Welzl. Unique sink orientations of cubes. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 547–555, 2001.

[33] F. Wessendorp. Notes on Morris' cube orientation. Diploma thesis, ETH, Zürich, March 2001.

[34] G. Zoutendijk. *Methods of feasible directions: A study in linear and non-linear programming*. Elsevier Publishing Co., Amsterdam, 1960.