Unique-Sink Orientations of Hypercubes and Linear Complementarity Problems

Jan Foniok
with Komei Fukuda, Bernd Gärtner, Lorenz Klaus, Hans-Jakob Lüthi

ODSA 2010
Unique-sink orientation — USO

an oriented graph with

- \( V = \{0, 1\}^n \)
- \( u \sim v \) iff in Hamming distance 1
Unique-sink orientation — USO

an oriented graph with

- $V = \{0, 1\}^n$
- $u \sim v$ iff in Hamming distance 1
- oriented so that every subcube has a unique sink

So the whole cube must have a unique sink
Unique-sink orientation — USO

an oriented graph with

- $V = \{0, 1\}^n$
- $u \sim v$ iff in Hamming distance 1
- oriented so that every subcube has a unique sink

So the whole cube must have a unique sink, but also proper subcubes, like this square.
Unique-sink orientation — USO

an oriented graph with

- $V = \{0, 1\}^n$
- $u \sim v$ iff in Hamming distance 1
- oriented so that every subcube has a unique sink

So the whole cube must have a unique sink, but also proper subcubes, like this square. And not two sinks.
Unique-sink orientation — USO

an oriented graph with

- $V = \{0, 1\}^n$
- $u \sim v$ iff in Hamming distance 1
- oriented so that every subcube has a unique sink

So the whole cube must have a unique sink, but also proper subcubes, like this square. And not two sinks. And not none.
Unique-sink orientation — USO

an oriented graph with
- \( V = \{0, 1\}^n \)
- \( u \sim v \) iff in Hamming distance 1
- oriented so that every subcube has a unique sink

So the whole cube must have a unique sink, but also proper subcubes, like this square. And not two sinks. And not none. Cycles may occur.
Goal: Find the sink

**Input representation:** by the vertex enumeration oracle: ask for the orientation of edges incident with a given vertex

**Algorithm efficiency:** number of oracle calls as function of dimension

**Algorithms**

**Naive algorithm:** check all vertices \((2^n \text{ queries})\)

**Path-following algorithms:** selection rule ??

**“Random access” algorithms:** seesaw
<table>
<thead>
<tr>
<th></th>
<th>deterministic</th>
<th>randomized</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>general USOs</strong></td>
<td>$1.609^n$</td>
<td>$1.438^n$</td>
</tr>
<tr>
<td>Szabó, Welzl</td>
<td>Szabó, Welzl, Rote</td>
<td></td>
</tr>
<tr>
<td><strong>acyclic USOs</strong></td>
<td>$\exp(2\sqrt{n})$</td>
<td>Matoušek, Sharir, Welzl, Gärtner</td>
</tr>
</tbody>
</table>
Why care?

Reductions from

- computing smallest enclosing ball of balls
- solving general linear and convex quadratic programs
- finding optimal strategies in simple stochastic games
- linear complementarity problems

Linear Complementarity Problem (LCP)

Given an $n \times n$ real matrix $M$, a real $n$-dimensional vector $q$, find two non-negative real $n$-dimensional vectors $w$, $z$ such that

$$w - Mz = q \quad w^Tz = 0$$

5 / 15
Why care?

Reductions from
- linear complementarity problems

Linear Complementarity Problem (LCP)

Given
- an $n \times n$ real matrix $M$,  
- a real $n$-dimensional vector $q$,

find
- two non-negative real $n$-dimensional vectors $w, z$ such that

\[
\begin{align*}
  w - Mz &= q \\
  w^T z &= 0
\end{align*}
\]
\[ w - Mz = q \]
\[ w^T z = 0 \]
\[ w - Mz = q \]
\[ w^T z = 0 \]
Complexity

NP-complete to decide whether a solution exists

Unsolved case: P-matrices (P-LCP)

A P-matrix is a matrix whose principal minors are all positive.

Is there a polynomial-time algorithm for solving P-LCP?

Theorem (Samelson, Thrall, Wesler 1958; Ingleton 1966)

A matrix $M$ is a P-matrix if and only if $\text{LCP}(M, q)$ has a unique solution for every vector $q$. 
### LCP Equations

\[
q = w - Mz \\
w^Tz = 0 \rightarrow w_i = 0 \text{ or } z_i = 0 \text{ for each } i
\]

### Problem reduction

The hard part: determine whether \( w_i = 0 \) or \( z_i = 0 \) for each \( i \).
The rest is a system of linear equations.
**LCP Equations**

\[
q = w - Mz \\
w^T z = 0 \\
\rightarrow w_i = 0 \text{ or } z_i = 0 \text{ for each } i
\]

**Problem reduction**

The hard part: determine whether \( w_i = 0 \) or \( z_i = 0 \) for each \( i \).

The rest is a system of linear equations.

**Inducing a USO**

- a choice of \( w_i = 0 \) or \( z_i = 0 \) corresponds to a 0-1-vector
- solve equations: negative values \( \Rightarrow \) outgoing edges
- *for a P-matrix, this is a USO* [Stickney, Watson, 1978]
## Some matrix classes

- **P-matrix**: all principal minors positive
- **K-matrix**: P-matrix and all off-diagonal elements $\leq 0$

Not all USOs can arise from a P-matrix LCP...
**Some matrix classes**

**P-matrix:** all principal minors positive

**K-matrix:** P-matrix and all off-diagonal elements $\leq 0$

Not all USOs can arise from a P-matrix LCP...

**Some USO classes**

**P-USO:** coming from a P-matrix LCP

**K-USO:** coming from a K-matrix LCP

... in fact, very few of them do.
**Theorem (F., Fukuda, Gärtner, Lüthi, 2009)**

Any path-following algorithm with any starting vertex finds the sink of any K-USO after at most $2n + 1$ oracle queries.

<table>
<thead>
<tr>
<th>00</th>
<th>00</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

The proof uses a K-matrix characterization of [Fiedler, Pták, 1962], but can also be done purely combinatorially (oriented matroids) [F., Fukuda, Klaus, 2010].
**Theorem (F., Fukuda, Gärtner, Lüthi, 2009)**

Any path-following algorithm with any starting vertex finds the sink of any K-USO after at most $2n + 1$ oracle queries.

**Lemma**

In any K-USO:

The proof uses a K-matrix characterization of [Fiedler, Pták, 1962], but can also be done purely combinatorially (oriented matroids) [F., Fukuda, Klaus, 2010].
Does the “Lemma” characterize K-USOs?

No. Because:

There are at least $2^{2^n / \text{poly}(n)}$ n-dimensional USOs satisfying the “Lemma”, but at most $2^{O(n^3)}$ P-USOs.
Does the “Lemma” characterize K-USOs?

No. Because:

There are at least $2^{2n/\text{poly}(n)}$ $n$-dimensional USOs satisfying the “Lemma”, but at most $2^{O(n^3)}$ P-USOs.

Proof of the upper bound (F., Gärtner, Klaus).

The orientation is determined by the signs of $2^n \cdot n$ values of polynomials in the entries of $M$ and $q$. Each of the polynomials has degree at most $n$.

Theorem (Warren, 1968)

The number of distinct (nowhere-zero) sign patterns of $s$ real polynomials in $k$ variables, each of degree at most $d$, is at most $(4ed^s/k)^k$. 
## Counting USOs

<table>
<thead>
<tr>
<th>Class</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>all USOs [Matoušek]</td>
<td>( n^{\Omega(2^n)} )</td>
<td>( n^{O(2^n)} )</td>
</tr>
<tr>
<td>acyclic USOs [Matoušek]</td>
<td>( 2^{2^n-1} )</td>
<td>( (n + 1)^{2^n} )</td>
</tr>
<tr>
<td>satisfying “Lemma”</td>
<td>( 2^{2^n}/\sqrt{n} )</td>
<td></td>
</tr>
<tr>
<td>Holt–Klee USOs [Develin]</td>
<td>( 2^{2^n}/\text{poly}(n) )</td>
<td></td>
</tr>
<tr>
<td>P-USOs</td>
<td>( 2^{\Omega(n^2)} )</td>
<td>( 2^{O(n^3)} )</td>
</tr>
<tr>
<td>K-USOs</td>
<td>( 2^{\Omega(n)} )</td>
<td></td>
</tr>
</tbody>
</table>
Counting USOs

<table>
<thead>
<tr>
<th>class</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>all USOs [Matoušek]</td>
<td>(n^{\Omega(2^n)})</td>
<td>(n^{O(2^n)})</td>
</tr>
<tr>
<td>acyclic USOs [Matoušek]</td>
<td>(2^{2^n-1})</td>
<td>((n + 1)^{2^n})</td>
</tr>
<tr>
<td>satisfying “Lemma”</td>
<td>(2^{2^n/\sqrt{n}})</td>
<td></td>
</tr>
<tr>
<td>Holt–Klee USOs [Develin]</td>
<td>(2^{2^n/\text{poly}(n)})</td>
<td></td>
</tr>
<tr>
<td>P-USOs</td>
<td>(2^{\Omega(n^2)})</td>
<td>(2^{O(n^3)})</td>
</tr>
<tr>
<td>K-USOs</td>
<td>(2^{\Omega(n)})</td>
<td></td>
</tr>
</tbody>
</table>

Holt–Klee USOs

In every subcube of dimension \(d\) there are \(d\) vertex-disjoint directed paths from the (unique) source to the (unique) sink.

*Every P-USO is Holt–Klee.* [Gärtner, Morris, Rüst, 2008]
Counting USOs

<table>
<thead>
<tr>
<th>class satisfying “Lemma”</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^{\left\lfloor \frac{n-1}{2} \right\rfloor}$</td>
<td></td>
</tr>
</tbody>
</table>

Lemma

*In any K-USO:*

![Diagram showing transitions in USOs]
P-USOs: A recursive construction

The P-matrix:

\[ M = \begin{pmatrix} M' & 0 \\ m^T & 1 \end{pmatrix} \]

\[ m_j = \begin{cases} (q_n + 1)M_{jj}'/q_j' & \text{if } s_j = -, \\ (q_n - 1)M_{jj}'/q_j' & \text{if } s_j = +, \end{cases} \]

\[ q_n = \begin{cases} -1 & \text{if } s_n = -, \\ 1 & \text{if } s_n = +; \end{cases} \]

Every choice \( s \in \{-, +\}^n \) yields a different P-USO.
Conclusion: Why do I think it’s interesting?

- interplay of several areas of mathematics
  - linear algebra & (continuous) geometry
  - discrete geometry
  - algebraic geometry
  - combinatorics & order theory

- embarrassingly open complexity status
- strongly polynomial algorithm for linear programming ?!?

Some open problems

- complexity: Is P-matrix LCP PPAD-complete?
- counting: The number of USOs for a single P-matrix: $\Omega(2^{n^2})$?