SORTING
AND
A TALE OF TWO POLYTOPES

Jean Cardinal
ULB, Brussels, Belgium
ALGORITHMS & PERMUTATIONS, Paris, 2012
Sorting by Comparisons

**Input:** a set $V$, totally ordered by an **unknown** order $\leq$

**Goal:** Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

**Objective function:** $\#$queries

- Classical problem in algorithms
- $\Theta(|V| \log |V|)$ queries necessary and sufficient (Heap Sort, Merge Sort)
Sorting by Comparisons under Partial Information

Input:
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: $\#\text{queries}$
Sorting by Comparisons under Partial Information

Input:

- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: $\#\text{queries}$
Sorting by Comparisons under Partial Information

Input:
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: $\#\text{queries}$
Sorting by Comparisons under Partial Information

Input:
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: #queries

![Diagram showing comparisons between elements to determine the order]
Sorting by Comparisons under Partial Information

Input:
▶ a set $V$, totally ordered by an unknown order $\leq$
▶ a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: $\#\text{queries}$
Sorting by Comparisons under Partial Information

Input:
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y?$”, for some $x, y \in V$

Objective function: $\#\text{queries}$
Sorting by Comparisons under Partial Information

**Input:**
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

**Goal:** Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

**Objective function:** #queries
Sorting by Comparisons under Partial Information

Input:
- a set $V$, totally ordered by an unknown order $\leq$
- a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries “is $x \leq y$?”, for some $x, y \in V$

Objective function: #queries
Sorting by Comparisons under Partial Information

Input:
▶ a set $V$, totally ordered by an unknown order $\leq$
▶ a partial order $P = (V, \leq_P)$ compatible with $\leq$

Goal: Discover $\leq$ by making queries "is $x \leq y$?", for some $x, y \in V$

Objective function: $\#\text{queries}$
Partial Orders and Linear Extensions

- **Hasse diagram**: maximum on top, transitive reduction
- $e(P) := \text{#linear extensions of } P$

\[ P \]
\[ \begin{array}{cccc}
  a & b & c & d \\
  \text{●} & \text{●} & \text{●} & \text{●} \\
  c & d & b & a \\
\end{array} \]

\[ e(P) = 5 \]
Lower Bound

Every comparison-based sorting algorithm can be forced to do at least

\[ \lg e(P) \]

comparisons.
Balanced Pairs

Does there always exist a comparison that "splits" the set of linear extensions into roughly equal part?

- **$1/3 - 2/3$ conjecture**: In any partial order $P$, there exists a pair of elements $a, b$ such that the fraction of linear extensions having $a < b$ is between $1/3$ and $2/3$.

- Proved for smaller values
  
  Brightwell, Felsner, and Trotter, 1995 – Brightwell, 1999
A Counting Issue

- Computing $e(P)$, or computing the fraction of linear extensions in which $a < b$ for some pair $a, b$ are \#P-Complete problems

Brightwell and Winkler, 1991

Goal of this talk:

- Insights into approximations of $\lg e(P)$ that will eventually yield efficient sorting algorithms
Plan

- The Order Polytope $O(P)$ of $P$ and how it relates to $\lg e(P)$
- The Chain Polytope $C(P)$ and how it relates to $O(P)$
- Approximating $\lg e(P)$ using the graph entropy and how it relates to the two polytopes
- A sorting algorithm
We consider the Euclidean space $\mathbb{R}^V$ of all functions $f : V \rightarrow \mathbb{R}$.

The Order Polytope $\mathcal{O}(P)$ of $P$ is the subset of $\mathbb{R}^V$ defined by:

\[
0 \leq f(x) \leq 1 \quad \forall x \in V \\
f(x) \leq f(y) \quad \text{if } x \leq_P y
\]
\( O(P) \) is the intersection of the subsets depicted below, for each comparable pair \( a \leq_P b \)
Example

\[
\begin{align*}
  f(a) &\leq f(b) \\
  f(a) &\leq f(c) \\
  0 &\leq f(a), f(b), f(c) \leq 1
\end{align*}
\]
Volume of the Order Polytope

\[ \text{vol}(\mathcal{O}(P)) = \frac{e(P)}{|V|!} \]

Stanley, 1986

A short proof:

- Every linear extension of \( P \) defines a simplex of \( \mathcal{O}(P) \)
- Every simplex has volume \( 1/|V|! \)
Volume of the Order Polytope

\[ f(a) \leq f(b) \]
\[ f(a) \leq f(c) \]
\[ 0 \leq f(a), f(b), f(c) \leq 1 \]
Volume of the Order Polytope

\[ f(a) \leq f(b) \leq f(c) \]
\[ 0 \leq f(a), f(b), f(c) \leq 1 \]
Volume of the Order Polytope

\[ f(a) \leq f(c) \leq f(b) \]
\[ 0 \leq f(a), f(b), f(c) \leq 1 \]
The Chain Polytope

- The Chain polytope $\mathcal{C}(P)$ of $P$ is the subset of $\mathbb{R}^V$ defined by
  \[
  0 \leq g(x) \quad \forall x \in V \\
  \sum_{x \in C} g(x) \leq 1 \quad \text{for every chain } C \text{ in } P
  \]

- Convex hull of the characteristic vectors of subsets of mutually incomparable elements (antichains)

- Convex corner: contains the convex hull of the origin and the basis vectors
Example

\[ g(a) + g(b) \leq 1 \]
\[ g(a) + g(c) \leq 1 \]
\[ g(a), g(b), g(c) \geq 0 \]
Example

independent sets: \{a\}, \{b\}, \{c\}, \{b, c\}
Define the transfer map $\phi : \mathcal{O}(P) \to \mathcal{C}(P)$ as follows: if $f \in \mathcal{O}(P)$ and $x \in V$, then

$$
(\phi f)(x) = \min\{f(x) - f(y) : y <_P x\}
$$

It can be checked that $\phi$ is a continuous, piecewise-linear bijection from $\mathcal{O}(P)$ onto $\mathcal{C}(P)$.
Example

- the function $f$ is increasing along the chain
- the function $g$ has sum at most one along the chain
Consequence

\[ \text{vol}(\mathcal{C}(P)) = \text{vol}(\mathcal{O}(P)) = \frac{e(P)}{|V|!} \]

- We may work with either polytope
- Considering the Chain polytope allows us to borrow ideas from graph theory

Stanley, 1986
Approximation of $(\log)e(P)$

Approximating the volume of a convex corner by an enclosed box:
Maximizing the Box Volume

- For any $x \in C(P)$, the box with the origin and $x$ as opposite corners is fully contained in $C(P)$
- Let us define the following maximum included box program:

$$\max_{v \in V} \prod_{v \in V} x_v$$

s.t. $x \in C(P)$
Entropy

Taking the log, normalizing by \( n := |V| \), and changing sign:

\[
\min \left\{ \frac{1}{n} \sum_{v \in V} \log x_v \right\} \\
\text{s.t.} \quad x \in C(P), \ x > 0
\]
Entropy

- Let us give it a name:

\[ H(P) = \min \frac{1}{n} \sum_{v \in V} \log x_v \]

s.t. \( x \in C(P), x > 0 \)

- Special case of the Graph Entropy

- Applications to data compression, Boolean formulas, optimization

Körner, 1973
For $P$ a total order, $C(P)$ is the convex hull of the basis vectors only
- We set $x_v = 1/n \ \forall v \in V$, and obtain $H(P) = \log n$

For $P$ an empty order, $C(P)$ is the unit cube
- We set $x_v = 1 \ \forall v \in V$, and obtain $H(P) = 0$

Intuitively, $H(P)$ measures the quantity of information contained in $P$
Approximation

- Let $x$ define the optimal box
- The volume of the box is bounded by that of the polytope:

\[
\prod_{v \in V} x_v \leq \frac{e(P)}{n!}
\]

\[
H(P) = -\frac{1}{n} \log \left( \prod_{v \in V} x_v \right)
\]

\[
n \log n - nH(P) \leq \log e(P) + 1.443n
\]
Approximation

- In fact, we can show that

\[ \lg e(P) = \Theta(n \log n - nH(P)) \]

Kahn and Kim, 1992

- Furthermore, computing \( H(P) \) is a \textit{convex programming problem}, that can be solved in polynomial time

- Hence we get a polynomial time constant-factor approximation algorithm for the sorting lower bound \( \lg e(P) \)
A Sorting Algorithm

1. Compute greedy chain decomposition of $P$
2. Iteratively merge two smallest chains
A Sorting Algorithm

1. Compute greedy chain decomposition of $P$
2. Iteratively merge two smallest chains
A Sorting Algorithm

1. Compute greedy chain decomposition of $P$
2. Iteratively merge two smallest chains
A Sorting Algorithm

1. Compute greedy chain decomposition of $P$
2. Iteratively merge two smallest chains
A Sorting Algorithm

1. Compute greedy chain decomposition of $P$
2. Iteratively merge two smallest chains
A Sorting Algorithm

ETC.
Analysis (Outline)

This algorithm performs \((1 + \varepsilon) \log e(P) + O_\varepsilon(n)\) comparisons

C., Fiorini, Joret, Jungers, Munro, 2010

Proof outline:

- The tree of merges is a **Huffman tree**
- Hence the number of comparisons is at most \(g + O(n)\), where:
  \[
g := \sum_{i=1}^{k} |C_i| \log \frac{n}{|C_i|}
\]
- We could prove that \(g\) is a good approximation of \(n \log n - nH(P)\), hence of \(\log e(P)\)
Summary

1. Volume of the **Order Polytope** $\mathcal{O}(P)$ of $P$ proportional to $e(P)$

2. Chain Polytope $C(P) = \phi \mathcal{O}(P)$

3. Maximum included box in $C(P) \rightarrow$ entropy $\rightarrow$ approximates $\lg e(P)$

4. Cost of **Greedy Merge Sort** approximates entropy
Extending the Scope

- The graph entropy framework provides efficient algorithms for approximating the number of linear extensions of a partial order.
- Can it be used for other applications?
- Can it be used to approximate other quantities?
Thank You!
Pointers: Stanley’s map

R. P. Stanley.
Two poset polytopes.
Pointers: Sorting

M. L. Fredman.
How good is the information theory bound in sorting?

N. Linial.
The information-theoretic bound is good for merging.

J. Kahn and M. E. Saks.
Balancing poset extensions.

J. Kahn and N. Linial.
Balancing extensions via Brunn-Minkowski.

G. R. Brightwell.
Balanced pairs in partial orders.
Pointers: Entropy


Pointers: Sorting and Entropy

J. Kahn and J. H. Kim.
Entropy and sorting.

A. C.-C. Yao.
Graph entropy and quantum sorting problems.

J. C., S. Fiorini, G. Joret, R. Jungers, and J. I. Munro.
An efficient algorithm for partial order production.

J. C., S. Fiorini, G. Joret, R. Jungers, and J. I. Munro.
Sorting under Partial Information (without the Ellipsoid Algorithm).