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 $\leqslant$

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

Objective function: #queries

- Calssical 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 $\leqslant$
- a partial order $P = (V, \leqslant_P)$ compatible with $\leqslant$

Goal: Discover $\leqslant$ by making queries "is $x \leqslant y$?", 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 $\leqslant$
- a partial order $P = (V, \leqslant_P)$ compatible with $\leqslant$

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

Objective function: #queries

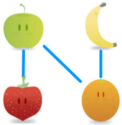
Sorting by Comparisons under Partial Information

Input:

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

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

Objective function: #queries



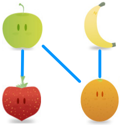
Sorting by Comparisons under Partial Information

Input:

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

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

Objective function: #queries



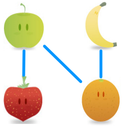
Sorting by Comparisons under Partial Information

Input:

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

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

Objective function: #queries



 OR  ?

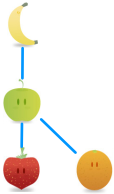
Sorting by Comparisons under Partial Information

Input:

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

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

Objective function: #queries



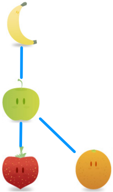
Sorting by Comparisons under Partial Information

Input:

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

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

Objective function: #queries



 OR  ?

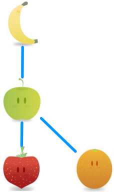
Sorting by Comparisons under Partial Information

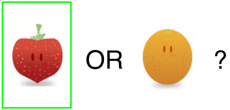
Input:

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

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

Objective function: #queries





Sorting by Comparisons under Partial Information

Input:

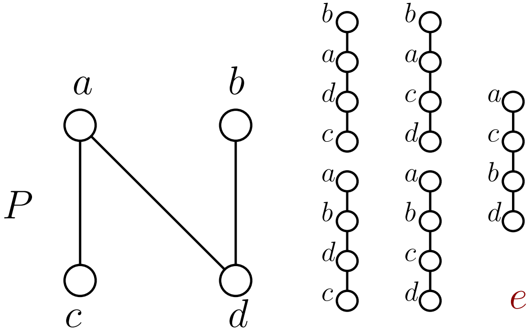
- a set V, totally ordered by an unknown order $\leqslant$
- a partial order $P = (V, \leqslant_P)$ compatible with $\leqslant$

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

Objective function: #queries

Partial Orders and Linear Extensions

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



$$e(P) = 5$$

Lower Bound

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

$$\boxed{\lg e(P)}$$

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 $\mathcal{O}(P)$ of P and how it relates to $\lg e(P)$
- The Chain Polytope $\mathcal{C}(P)$ and how it relates to $\mathcal{O}(P)$
- Approximating $\lg e(P)$ using the graph entropy and how it relates to the two polytopes
- A sorting algorithm

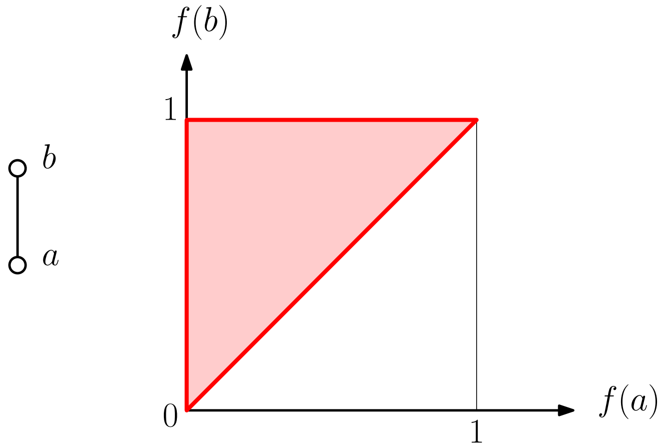
The Order Polytope

- We consider the Euclidean space $\mathbb{R}^V$ of all functions $f : V \to \mathbb{R}$
- the Order Polytope $\mathcal{O}(P)$ of P is the subset of $\mathbb{R}^V$ defined by:

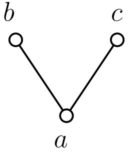
$$\begin{aligned} 0 \leq f(x) \leq 1 &\qquad \forall x \in V \\ f(x) \leq f(y) &\qquad \text{if } x \leqslant_P y \end{aligned}$$

Interpretation

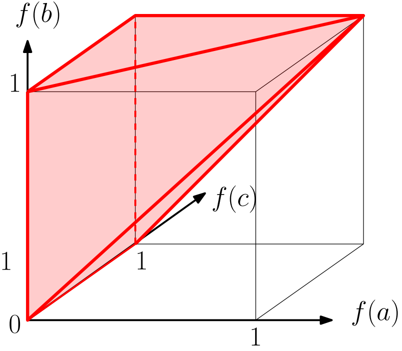
- $\mathcal{O}(P)$ is the intersection of the subsets depicted below, for each comparable pair $a \leqslant_P b$



Example



$$f(a) \leq f(b)$$
$$f(a) \leq f(c)$$
$$0 \leq f(a), f(b), f(c) \leq 1$$



Volume of the Order Polytope

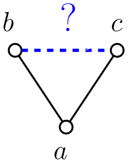
$$\boxed{vol(\mathcal{O}(P)) = e(P)/|V|!}$$

Stanley, 1986

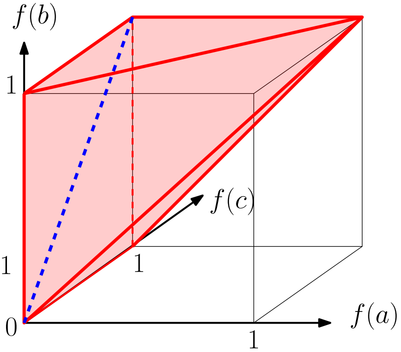
A short proof:

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

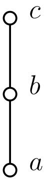
Volume of the Order Polytope



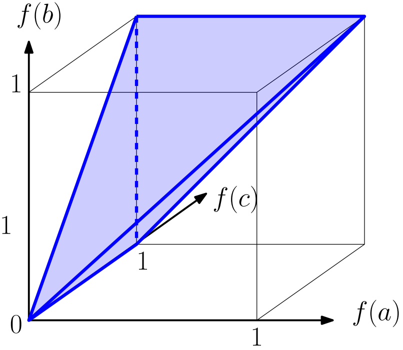
$f(a) \leq f(b)$
$f(a) \leq f(c)$
$0 \leq f(a), f(b), f(c) \leq 1$



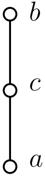
Volume of the Order Polytope



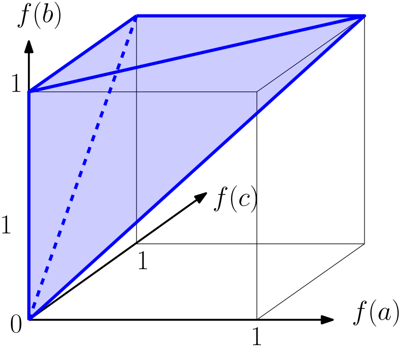
$$f(a) \leq f(b) \leq f(c)$$
$$0 \leq f(a), f(b), f(c) \leq 1$$



Volume of the Order Polytope



$f(a) \le f(c) \le f(b)$
$0 \le f(a), f(b), f(c) \le 1$



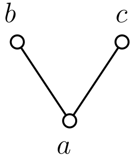
The Chain Polytope

- The Chain polytope $\mathcal{C}(P)$ of P is the subset of $\mathbb{R}^V$ defined by

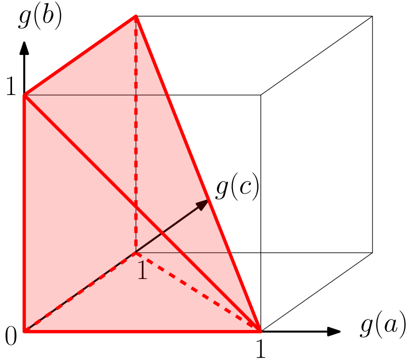
$$
\begin{aligned}
0 \leq g(x) & \qquad \forall x \in V \\
\sum_{x \in C} g(x) \leq 1 & \qquad \text{for every chain } C \text{ in } P
\end{aligned}
$$

- 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

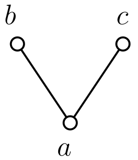
Example



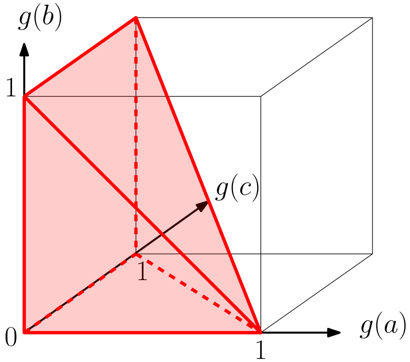
$$g(a) + g(b) \leq 1$$
$$g(a) + g(c) \leq 1$$
$$g(a), g(b), g(c) \geq 0$$



Example



independent sets: $\{a\}, \{b\}, \{c\}, \{b, c\}$



From the Order Polytope to the Chain Polytope

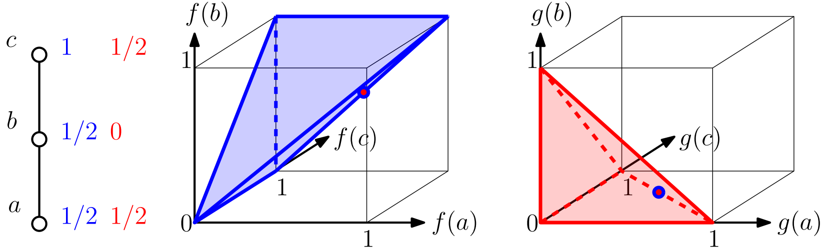
- Define the transfer map $\phi : \mathcal{O}(P) \to \mathcal{C}(P)$ as follows: if $f \in \mathcal{O}(P)$ and $x \in V$, then

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

- It can be checked that ϕ 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

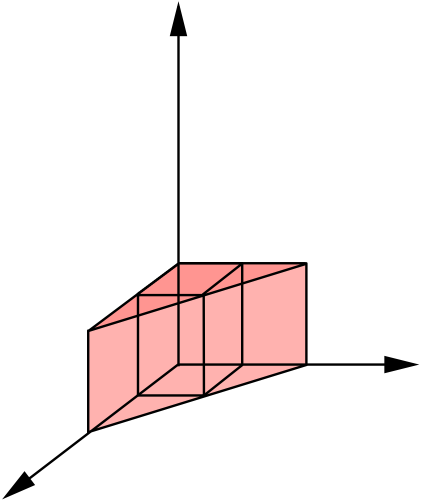
$$\boxed{vol(\mathcal{C}(P)) = vol(\mathcal{O}(P)) = e(P)/|V|!}$$

Stanley, 1986

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

Approximation of (log)$e(P)$

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



Maximizing the Box Volume

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

$$\begin{array}{ll} \max & \prod_{v \in V} x_v \\ \text{s.t.} & x \in \mathcal{C}(P) \end{array}$$

Entropy

Taking the log, normalizing by $n := |V|$, and changing sign:

$$\begin{aligned} \min \quad & -\frac{1}{n} \sum_{v \in V} \lg x_v \\ \text{s.t.} \quad & x \in \mathcal{C}(P), x > 0 \end{aligned}$$

Entropy

- Let us give it a name:

$$H(P) = \begin{array}{ll} \min & -\frac{1}{n}\sum_{v\in V} \lg x_v \\ \text{s.t.} & x \in \mathcal{C}(P), x > 0 \end{array}$$

- Special case of the Graph Entropy — Körner, 1973
- Applications to data compression, Boolean formulas, optimization

Entropy

- For P a total order, $\mathcal{C}(P)$ 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, $\mathcal{C}(P)$ 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:

$$
\begin{aligned}
\prod_{v \in V} x_v &\leq e(P)/n! \\
H(P) &= -\frac{1}{n} \log\left(\prod_{v \in V} x_v\right) \\
n \log n - nH(P) &\leq \lg e(P) + 1.443n
\end{aligned}
$$

Approximation

- In fact, we can show that

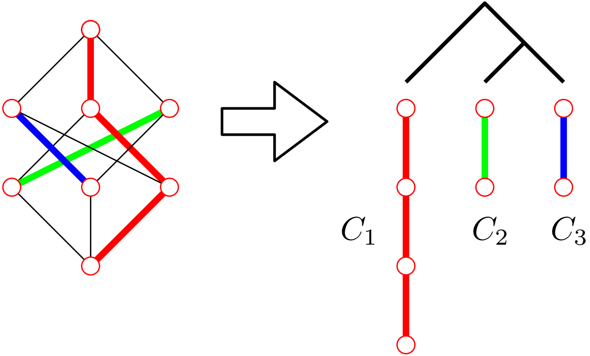
$$\boxed{\lg e(P) = \Theta(n \log n - nH(P))}$$

Kahn and Kim, 1992

- Furthermore, computing $H(P)$ is a 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 $\lg e(P)$

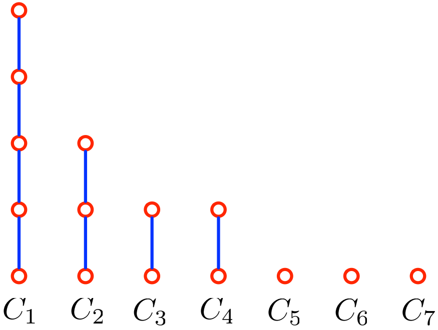
A Sorting Algorithm

1. Compute greedy chain decomposition of P
2. Iteratively merge two smallest chains



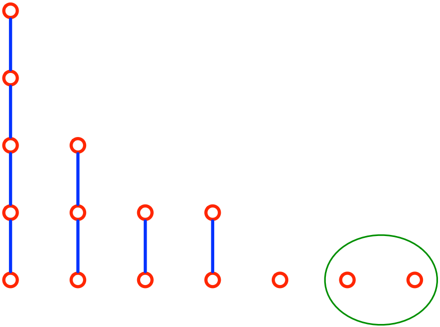
A Sorting Algorithm

1. Compute greedy chain decomposition of P
2. Iteratively merge two smallest chains



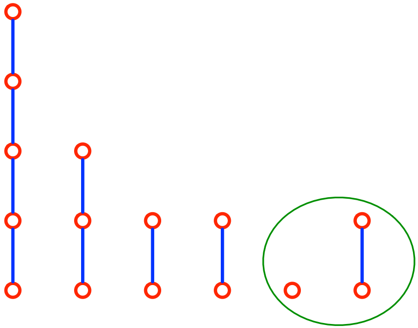
A Sorting Algorithm

1. Compute greedy chain decomposition of P
2. Iteratively merge two smallest chains



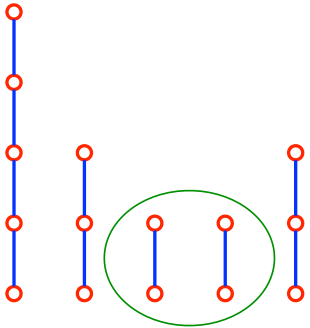
A Sorting Algorithm

1. Compute greedy chain decomposition of P
2. Iteratively merge two smallest chains

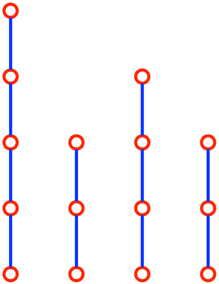


A Sorting Algorithm

1. Compute greedy chain decomposition of P
2. Iteratively merge two smallest chains



A Sorting Algorithm



ETC.

Analysis (Outline)

This algorithm performs $(1+\varepsilon)\lg e(P) + O_\varepsilon(n)$ 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 :

$$g := \sum_{i=1}^{k} |C_i| \lg \frac{n}{|C_i|}$$

- We could prove that g is a good approximation of $n \log n - nH(P)$, hence of $\lg e(P)$

Summary

1. Volume of the Order Polytope $\mathcal{O}(P)$ of P proportional to $e(P)$
2. Chain Polytope $\mathcal{C}(P) = \phi\mathcal{O}(P)$
3. Maximum included box in $\mathcal{C}(P)$ $\rightarrow$ 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.
Discrete Comput. Geom., 1:9–23, 1986.

Pointers: Counting Linear Extensions

G. R. Brightwell and P. Winkler.
Counting linear extensions.
Order, 8(3):225–242, 1991.

Pointers: Sorting

M. L. Fredman.
How good is the information theory bound in sorting?
Theor. Comput. Sci., 1(4):355–361, 1976.

N. Linial.
The information-theoretic bound is good for merging.
SIAM J. Comput., 13(4):795–801, 1984.

J. Kahn and M. E. Saks.
Balancing poset extensions.
Order, 1:113–126, 1984.

J. Kahn and N. Linial.
Balancing extensions via Brunn-Minkowski.
Combinatorica, 11:363–368, 1991.

G. R. Brightwell.
Balanced pairs in partial orders.
Discrete Mathematics, 201(1–3):25–52, 1999.

Pointers: Entropy

J. Körner.
Coding of an information source having ambiguous alphabet and the entropy of graphs.
In *Transactions of the 6th Prague Conference on Information Theory*, pages 411–425, 1973.

G. Simonyi.
Graph entropy: a survey.
In *Combinatorial optimization (New Brunswick, NJ, 1992–1993)*, volume 20 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 399–441. Amer. Math. Soc., Providence, RI, 1995.

Pointers: Sorting and Entropy

J. Kahn and J. H. Kim.
Entropy and sorting.
J. Comput. Syst. Sci., 51(3):390–399, 1995.

A. C.-C. Yao.
Graph entropy and quantum sorting problems.
In *STOC'04: 36th Annual ACM Symposium on Theory of Computing*, pages 112–117, 2004.

J. C., S. Fiorini, G. Joret, R. Jungers, and J. I. Munro.
An efficient algorithm for partial order production.
In *STOC '09: 41st ACM Symposium on Theory of Computing*, 2009.

J. C., S. Fiorini, G. Joret, R. Jungers, and J. I. Munro.
Sorting under Partial Information (without the Ellipsoid Algorithm).
In *STOC '10: 42nd ACM Symposium on Theory of Computing*, 2010.