The quantum query complexity of sorting under partial information

Jérémie Roland

Université libre de Bruxelles

Quantum Information & Communication

Joint work (in progress) with Jean Cardinal and Gwenaël Joret

Outline

The Sorting problem

Definition

- Let $V = \{v_1, \ldots, v_n\}$ be totally ordered by an unknown linear order $\leqslant$
- Determine $\leqslant$ by making queries of the form "is $v_i \leqslant v_j$?"

Classical query complexity (or decision tree complexity)

- $C(\text{Sorting}) =$minimum #queries to solve Sorting
- Trivial lower bound: $C(\text{Sorting}) \geq \log n! = \Omega(n \log n)$
  - One line proof: # possible orders$= n!$
- Upper bound: $C(\text{Sorting}) = O(n \log n)$
  - Many algorithms: Mergesort, Heapsort

Quantum speedup?

The classical query complexity of Sorting is $\Theta(n \log n)$

Question

Can quantum algorithms provide a speedup for the Sorting problem?

No...

Outline

The adversary bound

Theorem [Ambainis'02, Høyer Lee Špalek'07]

$$Q_\epsilon(\mathrm{Sorting}_P) = \Omega(\mathrm{Adv}(\mathrm{Sorting}_P))$$

where

$$\mathrm{Adv}(\mathrm{Sorting}_P) = \max_\Gamma \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

Notes

- Valid for any problem in the query model, not just for $\mathrm{Sorting}_P$
- For $\mathrm{Sorting}$
  - The involved matrices are $n! \times n!$
  - Lines and columns are indexed by permutations σ over $\{1, \ldots, n\}$ such that

$$v_i \leqslant v_j \qquad \Leftrightarrow \qquad \sigma(i) \leq \sigma(j)$$

  - For σ, the unknown total order is therefore such that

$$v_{\sigma^{-1}(1)} \leqslant v_{\sigma^{-1}(2)} \leqslant \ldots \leqslant v_{\sigma^{-1}(n)}$$

  - J is the all-1 matrix, and Δ^{ij} the boolean matrix such that
    - $\Delta^{ij}_{\sigma\tau} = 1$ iff the query $v_i \leq v_j$ returns the same answer for σ and τ

Quantum lower bound for Sorting

- We just need to find a good adversary matrix Γ
- Høyer, Neerbek and Shi proposed to use

$$\Gamma_{\sigma\tau} = \frac{1}{d} \qquad \text{for} \qquad \tau = (k, k+1, \ldots, k+d) \circ \sigma$$

  - $\Gamma_{\sigma\tau} = \frac{1}{d}$ when total orders are the same except for one element shifted by d positions

Theorem [Høyer Neerbek Shi'02]

$$\mathrm{Adv}(\mathrm{Sorting}) = \Omega(n \log n)$$

Conclusion

- No quantum speedup for Sorting

Outline

Definition

- Let $V = \{v_1, \ldots, v_n\}$ be totally ordered by an unknown linear order $\leqslant$
- Let $P = (V, \leqslant_P)$ denote a poset (partially ordered set) compatible with $(V, \leqslant)$
- Given P, determine $\leqslant$ by making queries of the form “is $v_i \leqslant v_j$?”

Notes

- A poset $P = (V, \leqslant_P)$ specifies a partial order between elements of V
- Since $P = (V, \leqslant_P)$ compatible with $(V, \leqslant)$

$$v_i \leqslant_P v_j \qquad \Rightarrow \qquad v_i \leqslant v_j$$

- Since P is given, some comparisons are already known

Definition

- Let $V = \{v_1, \ldots, v_n\}$ be totally ordered by an unknown linear order $\leqslant$
- Let $P = (V, \leqslant_P)$ denote a poset (partially ordered set) compatible with $(V, \leqslant)$
- Given P, determine $\leqslant$ by making queries of the form "is $v_i \leqslant v_j$?"

Classical query complexity

- Let $e(P)$ be the number of linear extensions of P
  - # total orders $(V, \leqslant)$ compatible with $(V, \leqslant_P)$
- Trivial lower bound: $C(\mathrm{Sorting}_P) \geq \log e(P)$
- Can we design an algorithm that matches this lower bound?

Balanced pairs

- Suppose we start with a poset $P(V, \leqslant_P)$ with $e(P)$
- After performing a query " is $v \leqslant w$?", we can update P
  - If yes: $P_\leqslant = P(v \leqslant w)$, with $e(P_\leqslant) \leq e(P)$
  - If no: $P_\geqslant = P(v \geqslant w)$, with $e(P_\geqslant) \leq e(P)$
- Observation: $e(P_\leqslant) + e(P_\geqslant) = e(P)$
- Ideal case: $e(P_\leqslant) \approx e(P_\geqslant) \approx e(P)/2$

Theorem(s)

If P is not a chain, then $\exists$ incomparable pair $v, w \in V$ s.t.

$$\delta \cdot e(P) \leq e(P(v \leqslant w)) \leq (1-\delta) \cdot e(P)$$

for some absolute constant $\delta > 0$

- $\delta = \frac{3}{11} \simeq 0.2727$ [Kahn Saks'84]
- $\delta = \frac{5-\sqrt{5}}{10} \simeq 0.2764$ [Brightwell Felsner Trotter'95]
- 1/3–2/3 conjecture: $\delta = \frac{1}{3}$

Algorithm

Algorithm for Sorting under partial information

1. Given P, find a δ-balanced pair v, w
2. Query " is $v \leqslant w$?"
3. Update P according to result
4. Repeat until P is a total order

Discussion

- The algorithm uses $\leq \log_{1/(1-\delta)} e(P) = \Theta(\log e(P))$ queries
  - Good!
- Computing $e(P)$ is a $\#P$-complete problem
  - Bad...
- Can we approximate $e(P)$?

Outline

The Order Polytope

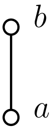
- In all that follows, $P = (V, \leqslant_P)$ is a poset on a set V of n elements

Definition

The Order Polytope $\mathcal{O}(P)$ of P is the subset of points $x \in \mathbb{R}^V$ satisfying:

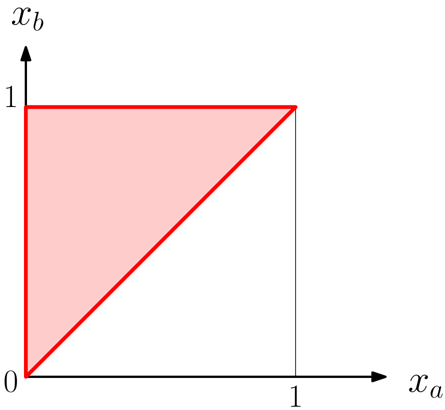
$$\begin{aligned} \mathbf{0} \leq \mathbf{x_v} \leq \mathbf{1} & \qquad \forall v \in V \\ \mathbf{x_v} \leq \mathbf{x_w} & \qquad \forall v, w \in V \text{ such that } v \leqslant_P w \end{aligned}$$

Examples

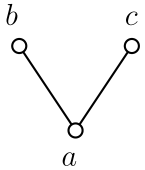


$0 \le x_a, x_b \le 1$

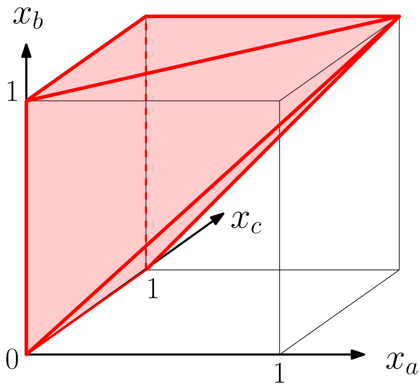
$x_a \le x_b$



Examples



$$0 \le x_a, x_b, x_c \le 1$$
$$x_a \le x_b$$
$$x_a \le x_c$$



Volume of the Order Polytope

- Recall
  - $n := |V|$
  - $e(P) :=$ #linear extensions of P

Theorem [Stanley'86]

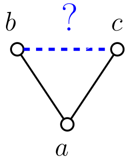
$$vol(\mathcal{O}(P)) = \frac{e(P)}{n!}$$

Proof (sketch)

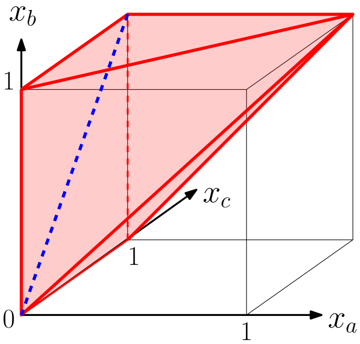
- Every linear extension of P defines a simplex of $\mathcal{O}(P)$
- Every simplex has volume $1/n!$
  - One simplex for each of the $n!$ possible total orders

Volume of the Order Polytope

Illustration of the proof

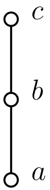


$$0 \leq x_a, x_b, x_c \leq 1$$
$$x_a \leq x_b$$
$$x_a \leq x_c$$

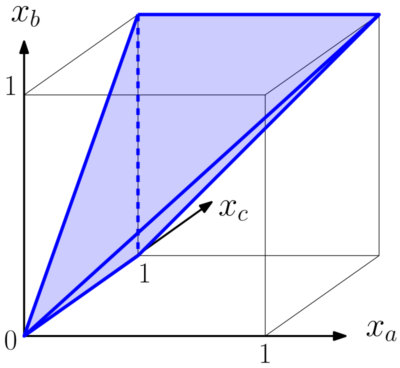


Volume of the Order Polytope

A first simplex:

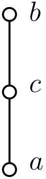


$x_a \leq x_b \leq x_c$
$0 \leq x_a, x_b, x_c \leq 1$

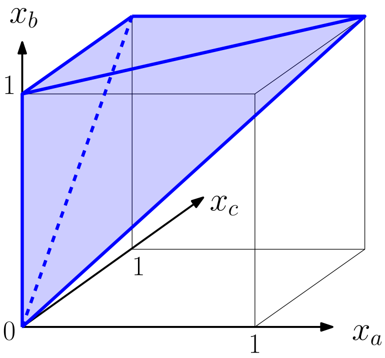


Volume of the Order Polytope

A second simplex:



$x_a \leq x_c \leq x_b$
$0 \leq x_a, x_b, x_c \leq 1$



The Chain Polytope

Notion of chain

- Given a poset P, a chain C is a subset of elements such that

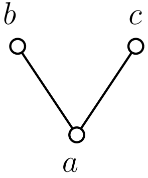
$$v_1 \leqslant_P v_2 \leqslant_P \ldots \leqslant_P v_k$$

Definition

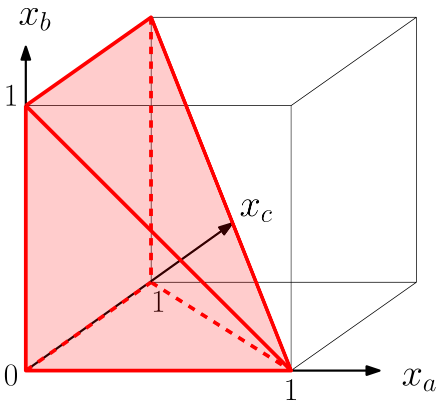
The Chain Polytope $\mathcal{C}(P)$ of P is the subset of points $x \in \mathbb{R}^V$ satisfying:

$$\begin{aligned} \mathbf{x_v} &\geq \mathbf{0} && \forall v \in V \\ \textstyle\sum_{\mathbf{v} \in \mathbf{C}} \mathbf{x_v} &\leq \mathbf{1} && \text{for every chain } C \text{ in } P \end{aligned}$$

Example



$$x_a, x_b, x_c \geq 0$$
$$x_a + x_b \leq 1$$
$$x_a + x_c \leq 1$$



Definition

Let $\phi : \mathcal{O}(P) \mapsto \mathcal{C}(P) : x \to y$ where, for each $v \in V$

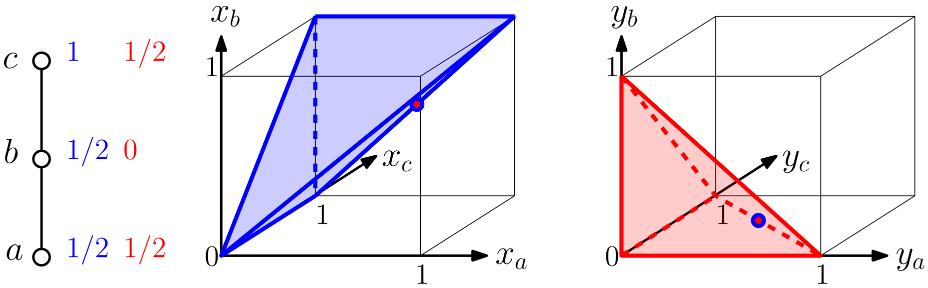
$$y_v = \begin{cases} x_v & \text{if } v \text{ minimal element} \\ \min\{x_v - x_w : w <_P v\} & \text{otherwise.} \end{cases}$$

Properties of ϕ

- ϕ is a continuous, piecewise-linear bijection from $\mathcal{O}(P)$ onto $\mathcal{C}(P)$

Example

A point $x \in \mathcal{O}(P)$ and its image $y = \phi(x) \in \mathcal{C}(P)$



Consequence

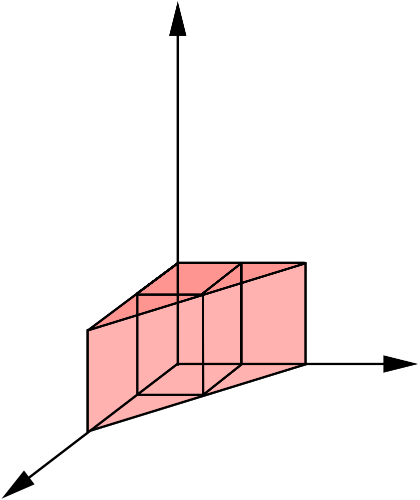
Corollary [Stanley'86]

$$vol(\mathcal{C}(P)) = vol(\mathcal{O}(P)) = \frac{e(P)}{n!}$$

We may thus work with either polytope to approximate $e(P)$

Outline

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



Maximizing the box volume inside the Chain Polytope

Observation

- For each $x \in \mathcal{C}(P)$, the box with the origin and x as opposite corners is fully contained in $\mathcal{C}(P)$
- Let us consider the included box with the largest volume
  - Maximum included box program:

$$\max \prod_{v \in V} x_v \qquad \text{s.t.} \qquad x \in \mathcal{C}(P)$$

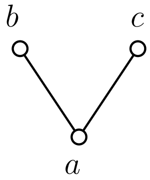
- Taking the log, normalizing by n, and changing sign, we get

Definition

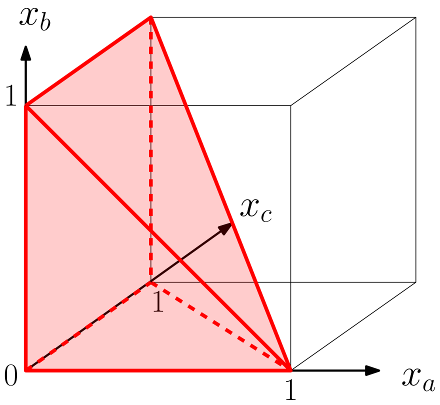
The entropy of P is

$$H(P) := \quad \min \quad -\frac{1}{n} \sum_{v \in V} \log x_v \qquad \text{s.t.} \quad x \in \mathcal{C}(P)$$

- Special case of Graph entropy [Körner'73]
  - For the comparability graph of P



$$x_a, x_b, x_c \geq 0$$
$$x_a + x_b \leq 1$$
$$x_a + x_c \leq 1$$



Maximizing the box volume inside the Chain Polytope

Observation

- For each $x \in \mathcal{C}(P)$, the box with the origin and x as opposite corners is fully contained in $\mathcal{C}(P)$
- Let us consider the included box with the largest volume
  - Maximum included box program:

$$\max \prod_{v \in V} x_v \qquad \text{s.t.} \qquad x \in \mathcal{C}(P)$$

- Taking the log, normalizing by n, and changing sign, we get

Definition

The entropy of P is

$$H(P) := \quad \min \quad -\frac{1}{n} \sum_{v \in V} \log x_v \qquad \text{s.t.} \quad x \in \mathcal{C}(P)$$

- Special case of Graph entropy [Körner'73]
  - For the comparability graph of P

Approximating $\log e(P)$

Main idea

- The volume of the Chain Polytope is $vol(\mathcal{C}(P)) = vol(\mathcal{O}(P)) = \frac{e(P)}{n!}$
- Taking the log, and changing sign, we get
  - $-\log vol(\mathcal{C}(P)) = n \log n - \log e(P) + O(n)$
- Let $\mathcal{V}$ be the volume of the maximum included box
  - $-\log \mathcal{V} = nH(P)$ is used as an approximation for $n \log n - \log e(P)$
- Introducing the dual entropy $H(\overline{P}) = \log n - H(P)$
  - $nH(\overline{P})$ is used as an approximation for $\log e(P)$

Theorem(s)

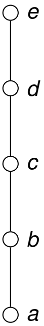
$$\log e(p) \leq nH(\overline{P}) \leq c \log e(P)$$

- $c = 1 + 7 \log e \approx 11.1$ [Kahn Kim'95]
- $c = 2$ (tight) [Cardinal Fiorini Joret Jungers Munro'10]

Definition

$$H(P) := \min\{f(x) : x \in \mathcal{C}(P)\} \qquad \text{where} \qquad f(x) := -\frac{1}{n}\sum_{v\in V} \log x_v$$

- If P is a total order then
$$\mathcal{C}(P) = \{x \in \mathbb{R}^V : \quad x_v \geq 0 \quad \forall v \in V \quad \& \quad \sum_{v\in V} x_v \leq 1\}$$
  - setting $x_v = \frac{1}{n} \ \forall v \in V$ minimizes $f(x)$, thus $H(P) = \log n$
- If P is an empty order then
$$\mathcal{C}(P) = \{x \in \mathbb{R}^V : 0 \leq x_v \leq 1 \quad \forall v \in V\}$$
  - setting $x_v = 1 \ \forall v \in V$ minimizes $f(x)$, thus $H(P) = 0$
- If Q is a poset on V extending P, then $H(Q) \geq H(P)$
  - Thus in general $\boxed{0 \leq H(P) \leq \log n}$



Outline

Kahn & Kim's approach

Lemma [Kahn Kim'95]

If P is not a chain, then $\exists$ incomparable pair $v, w \in V$ s.t.

$$\max\left\{nH(\overline{P(v \leqslant w)}), nH(\overline{P(v \geqslant w)})\right\} \leq nH(\overline{P}) - c$$

where $c = \log(1 + 17/112) \simeq 0.2$

Discussion

- Similar to δ-unbalanced pairs
  - Using entropy $H(\overline{P})$ instead of $e(P)$
- $H(\overline{P})$ can be computed efficiently (ellipsoid method)

Kahn & Kim's algorithm

Algorithm for Sorting under partial information

1. Given P, find an incomparable pair v, w as in previous lemma
2. Query " is $v \leqslant w$?"
3. Update P according to result
4. Repeat until P is a total order

Discussion

- The algorithm uses $O(nH(\overline{P})) = O(\log e(P))$ queries
  - Good!
- It is polynomial and deterministic
  - Good!

Outline

Recall: the adversary bound

Theorem [Ambainis'02, Høyer Lee Špalek'07]

$$Q_\epsilon(\text{Sorting}_P) = \Omega(\text{Adv}(\text{Sorting}_P))$$

where

$$\text{Adv}(\text{Sorting}_P) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

Notes

- Valid for any problem in the query model, not just for Sorting_P
- For Sorting_P
  - The involved matrices are $e(P) \times e(P)$
  - Lines and columns are indexed by permutations σ consistent with P

Yao's quantum lower bound for $\mathrm{Sorting}_P$

- Using the same adversary matrix as [Høyer Neerbek Shi'02]

$$\Gamma_{\sigma\tau} = \frac{1}{d} \qquad \text{for} \qquad \tau = (k, k+1, \ldots, k+d) \circ \sigma$$

  - Restricted to lines/columns for $\sigma \in \Delta(P)$ (those consistent with P)
- Yao proved the following lower bound

Theorem [Yao'04]

For any poset P,

$$\mathrm{Adv}(\mathrm{Sorting}_P) = \mathrm{QLB}(P) := \mathbf{E}_{\sigma\in\Delta(P)}\left[\sum_v H_{d_v(\sigma)-1}\right]$$

where H_k is the k-th Harmonic number and

$$d_i(\sigma) := \begin{cases} \sigma(i) & \text{if } v_i \text{ minimal element in } P \\ \min\{\sigma(i)-\sigma(j) : v_j <_P v_j\} & \text{otherwise.} \end{cases}$$

Conjecture: no quantum speedup

- Yao conjectured that this bound is tight and matches the classical complexity

Conjecture [Yao'04]

For any poset P

$$\mathrm{QLB}(P) \geq c \log e(P) \qquad \text{for some constant } c > 0$$

- Using connections with graph entropy, he was able to prove

Theorem [Yao'04]

For any poset P

$$\mathrm{QLB}(P) \geq c \log e(P) - c'n \qquad \text{for some constant } c, c' > 0$$

- Due to the linear term, this gives a trivial bound if $\log e(P) = o(n)$

Yao's approach

- First, let's switch to natural logarithms:

$$H(\overline{P}) = \max_{x \in \mathcal{C}(P)} [\ln n - f(x)] \qquad \text{where} \qquad f(x) = -\frac{1}{n} \sum_{v \in V} \ln x_v$$

- Therefore $nH(\overline{P}) \geq \ln e(P)$

Lemma [Yao'04]

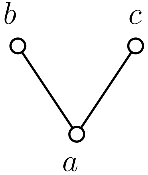
For any poset P

$$\mathrm{QLB}(P) \geq n\mathbf{E}_{x \in \mathcal{C}(P)} [\ln n - f(x)]$$

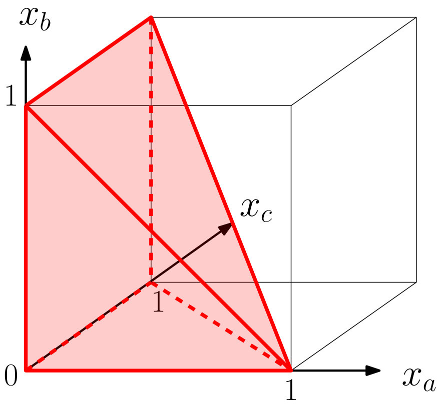
$$\mathbf{E}_{x \in \mathcal{C}(P)} [\ln n - f(x)] \geq H(\overline{P}) - 200$$

Discussion

- Almost what we want, except that we have an average version of the entropy instead of a max
- Still OK if both versions are close
- Since it is multiplied by n in the lower bound, the -200 terms causes the linear loss



$$x_a, x_b, x_c \geq 0$$
$$x_a + x_b \leq 1$$
$$x_a + x_c \leq 1$$



Outline

Observations

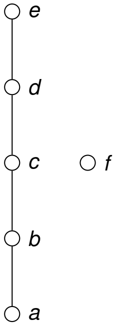
- The bound $n\mathbf{E}_{x\in\mathcal{C}(P)}\left[\ln n - f(x)\right]$ cannot be tight
- If P is a total order

$$n\mathbf{E}_{x\in\mathcal{C}(P)}\left[\ln n - f(x)\right] = -\gamma n + O(1)$$

  - where γ Euler-Mascheroni constant
- If P is the 'ordered insertion' poset

$$n\mathbf{E}_{x\in\mathcal{C}(P)}\left[\ln n - f(x)\right] = \ln(n-1) - \gamma n + O(1)$$

- For all examples considered: loss of $-\gamma n$
  - Maybe not a coïncidence?
  - Recall that $\gamma = \lim_{n\to\infty}[H_n - \ln n]$
- With a finer analysis of QLB we proved the following



Theorem [Cardinal Joret R.'17]

$$\mathrm{QLB}(P) = n\mathbf{E}_{x\in\mathcal{C}(P)}\left[H_n - f(x)\right]$$

Proof idea and consequences

- Proof based on the following main technical lemma, together with Stanley's map $\phi : \mathcal{O}(P) \mapsto \mathcal{C}(P)$

Lemma

For any poset P, for all $\sigma \in \Delta(P)$ and for all $1 \leq i \leq n$, we have

$$H_{d_i(\sigma)-1} = H_n + \mathbf{E}_{y \in \mathcal{O}_\sigma(P)} [\ln d_i(y)]$$

- We conjecture that the strengthened lower bound is tight, which reduces to the following conjecture

Conjecture

For any poset P

$$\mathbf{E}_{x \in \mathcal{C}(P)} [H_n - f(x)] \geq c \max_{x \in \mathcal{C}(P)} [\ln n - f(x)] \qquad \text{for some constant } c > 0$$

Towards proving Yao's conjecture

- We are able to prove the conjecture for an extended class of posets

Theorem [Cardinal Joret R.'17]

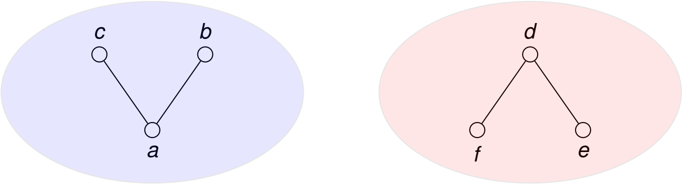
For any *series-parallel* poset P

$$\mathbf{E}_{x\in\mathcal{C}(P)}\left[H_n - f(x)\right] \geq c \max_{x\in\mathcal{C}(P)} [\ln n - f(x)]$$

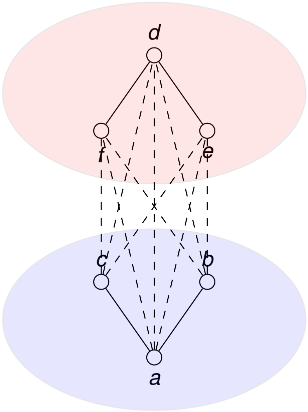
for $c = \frac{1}{2\ln 2} \simeq 0.72$

- Series-parallel posets can be obtained by composing iteratively smaller posets using
  - Parallel composition
  - Series composition

Parallel composition



Series composition



Towards proving Yao's conjecture

Theorem [Cardinal Joret R.'17]

For any *series-parallel* poset P

$$\mathbf{E}_{x\in\mathcal{C}(P)}\left[H_n - f(x)\right] \geq c \max_{x\in\mathcal{C}(P)}\left[\ln n - f(x)\right]$$

for $c = \frac{1}{2\ln 2} \simeq 0.72$

Proof idea

- We show that the average and the max-entropy behave the same
  - under series composition
  - under parallel composition
- The main theorem is then proved by induction on the size of P

Towards proving Yao's conjecture

Conjecture

For *any* poset P

$$\mathbf{E}_{x \in \mathcal{C}(P)} \left[H_n - f(x) \right] \geq c \max_{x \in \mathcal{C}(P)} \left[\ln n - f(x) \right]$$

for some constant c

Observation

- The function f is Schur concave

$$f(x) = -\frac{1}{n} \sum_{v \in V} \ln x_v$$

- Use majorization?

Thank you!

Setup

- Let us consider a quantum algorithm for Sorting
- We denote by σ the permutation over $\{1, \dots, n\}$ such that

$$v_i \leqslant v_j \qquad \Leftrightarrow \qquad \sigma(i) \leq \sigma(j)$$

  - The unknown total order is therefore such that

$$v_{\sigma^{-1}(1)} \leqslant v_{\sigma^{-1}(2)} \leqslant \dots \leqslant v_{\sigma^{-1}(n)}$$

- The algorithm should work for any $\sigma \in S_n$
- Let $|\psi_\sigma^t\rangle$ be the state of the quantum computer after t queries for permutation σ
- Let ρ^t be the Gram matrix of those states for all permutations $\sigma, \tau \in S_n$

$$\rho_{\sigma\tau}^t = \langle \psi_\sigma^t | \psi_\tau^t \rangle$$

Properties of the Gram matrix

- ρ^t is the matrix with entries

$$\rho^t_{\sigma\tau} = \langle \psi^t_\sigma | \psi^t_\tau \rangle$$

  - where $|\psi^t_\sigma\rangle$ is the state after t queries for permutation σ
- Initially (at $t = 0$)
  - Before any queries, the state $|\psi^0_\sigma\rangle$ is independent of σ

$$\rho^0 = J \qquad \text{(the all-1 matrix)}$$

- Unitaries independent of σ do not affect the Gram matrix

$$\langle \psi^t_\sigma | U^\dagger U | \psi^t_\tau \rangle = \langle \psi^t_\sigma | \psi^t_\tau \rangle$$

- At the end of the algorithm (at $t = T$, assuming T queries in total)
  - The algorithm must discriminate between all permutations so $\langle \psi^T_\sigma | \psi^T_\tau \rangle \approx \delta_{\sigma\tau}$

$$\rho^T \approx I \qquad \text{(the identity matrix)}$$

Progress function and effect of queries

Idea

In order to track the progress of the algorithm from $\rho^0 = J$ to $\rho^T \approx I$, we introduce a progress function

$$W(\rho^t) = \mathrm{Tr}[\Gamma(\rho^t \circ |\delta\rangle\langle\delta|)]$$

where Γ is a so-called adversary matrix and $|\delta\rangle$ its principal eigenvector

Effect of queries

- Let Δ^{ij} be the boolean matrix such that
  - $\Delta^{ij}_{\sigma\tau} = 1$ iff the query $v_i \leq v_j$ returns the same answer for σ and τ
- We can show that for each query

$$\left| W(\rho^{t+1}) - W(\rho^t) \right| \leq \max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|$$

Properties of the progress function

$$W(\rho^t) = \mathrm{Tr}[\Gamma(\rho^t \circ |\delta\rangle\langle\delta|)]$$

- Initially: $W(\rho^0) = W(J) = \mathrm{Tr}[\Gamma\,|\delta\rangle\langle\delta|] = \|\Gamma\|$
- For each query

$$\left|W(\rho^{t+1}) - W(\rho^t)\right| \leq \max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|$$

- After T queries: $\left|W(\rho^T) - W(\rho^0)\right| \leq T \max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|$
- At the end: $W(\rho^T) \approx W(I) = \mathrm{Tr}[(\Gamma \circ I)\,|\delta\rangle\langle\delta|] = 0$
  - assuming $\Gamma_{\sigma\sigma} = 0$

Conclusion

$$T \gtrsim \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

The adversary bound

Theorem [Ambainis'02, Høyer Lee Špalek'07]

$$Q_\epsilon(\text{Sorting}) = \Omega(\text{Adv}(\text{Sorting}))$$

where

$$\text{Adv}(\text{Sorting}) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

Notes

- Valid for any problem in the query model, not just for Sorting
- This bound is tight!

The adversary bound (2)

Theorem [Reichardt'11, LMRŠS'11]

$$Q_\epsilon(\mathrm{Sorting}) = \Theta(\mathrm{Adv}(\mathrm{Sorting}))$$

where

$$\mathrm{Adv}(\mathrm{Sorting}) = \max_\Gamma \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

Notes

- Valid for any problem in the query model, not just for $\mathrm{Sorting}$
- Now we just need to find a good adversary matrix Γ

Quantum lower bound for Sorting

Theorem [Høyer Neerbek Shi'02]

$$\mathrm{Adv}(\mathrm{Sorting}) = \Omega(n \log n)$$

Proof (sketch)

- Use the adversary matrix

$$\Gamma = \sum_{\sigma} \sum_{k=1}^{n-1} \sum_{d=1}^{n-k} \frac{1}{d} \left|\sigma\right\rangle \left\langle\sigma^{(k,d)}\right|,$$

  - where the permutation $\sigma^{(k,d)}$ is defined as $(k, k+1, \ldots, k+d) \circ \sigma$.

- Step 1 (skipped)

$$\|\Gamma \circ (J - \Delta^{ij})\| \leq \pi$$

- Step 2

$$\|\Gamma\| \geq nH_n - n$$

  - where $H_n = \Theta(\log n)$ is the n-th Harmonic number

Quantum lower bound for Sorting

Theorem [Høyer Neerbek Shi'02]

$$\mathrm{Adv}(\mathrm{Sorting}) = \Omega(n \log n)$$

Proof (sketch - continued)

- For

$$\Gamma = \sum_{\sigma} \sum_{k=1}^{n-1} \sum_{d=1}^{n-k} \frac{1}{d} |\sigma\rangle \langle \sigma^{(k,d)}|$$

$$|v\rangle = \frac{1}{\sqrt{n!}} \sum_{\sigma \in S_n} |\sigma\rangle$$

- We have

$$\|\Gamma\| \geq \langle v| \Gamma |v\rangle = \sum_{k=1}^{n-1} \sum_{d=1}^{n-k} \frac{1}{d} = \sum_{k=1}^{n-1} H_{n-k} = \sum_{l=1}^{n-1} H_l = nH_n - n$$

  - by the properties of the Harmonic numbers

Outline

4. Quantum lower bounds
   - Sorting
   - Sorting under Partial Information

Recall: the adversary bound

Theorem [Ambainis'02, Høyer Lee Špalek'07]

$$Q_\epsilon(\mathrm{Sorting}_P) = \Omega(\mathrm{Adv}(\mathrm{Sorting}_P))$$

where

$$\mathrm{Adv}(\mathrm{Sorting}_P) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{i,j} \|\Gamma \circ (J - \Delta^{ij})\|}$$

Notes

- Valid for any problem in the query model, not just for $\mathrm{Sorting}_P$
- For $\mathrm{Sorting}_P$
  - The involved matrices are $e(P) \times e(P)$
  - Lines and columns are indexed by permutations σ over $\{1, \ldots, n\}$ such that

    $$v_i \leqslant v_j \qquad \Leftrightarrow \qquad \sigma(i) \leq \sigma(j)$$

  - For σ, the unknown total order is therefore such that

    $$v_{\sigma^{-1}(1)} \leqslant v_{\sigma^{-1}(2)} \leqslant \ldots \leqslant v_{\sigma^{-1}(n)}$$

  - J is the all-1 matrix, and Δ^{ij} the boolean matrix such that
    - $\Delta^{ij}_{\sigma\tau} = 1$ iff the query $v_i \leq v_j$ returns the same answer for σ and τ

Yao's quantum lower bound for $\mathrm{Sorting}_P$

- Using the same adversary matrix as [Høyer Neerbek Shi'02]

$$\Gamma = \sum_{\sigma \in \Delta(P)} \sum_{k=1}^{n-1} \sum_{d=1}^{n-k} \frac{1}{d} \left|\sigma\right\rangle \left\langle \sigma^{(k,d)} \right|,$$

  - Restricted to lines/columns for $\sigma \in \Delta(P)$ (those consistent with P)
- Yao proved the following lower bound

Theorem [Yao'04]

For any poset P,

$$\mathrm{Adv}(\mathrm{Sorting}_P) = \mathrm{QLB}(P) := \mathbf{E}_{\sigma \in \Delta(P)} \left[\sum_v H_{d_v(\sigma)-1} \right]$$

where H_k is the k-th Harmonic number and

$$d_i(\sigma) := \begin{cases} \sigma(i) & \text{if } v_i \text{ minimal element in } P \\ \min\{\sigma(i) - \sigma(j) : v_j <_P v_j\} & \text{otherwise.} \end{cases}$$