Symmetries, graph properties, and quantum speedups

Full version: arXiv:2006.12760, To appear in FOCS 2020

Supartha Podder

University of Ottawa

Shalev Ben-David
University of Waterloo

Andrew M. Childs
University of Maryland

András Gilyén
Caltech/Berkeley

William Kretschmer
UT Austin

Daochen Wang
University of Maryland

The power of quantum computers

Quantum computers can sometimes solve problems significantly faster than classical computers.

Some problems for which quantum computation exhibits exponential speedups

Quantum simulations, period finding, factoring, discrete logarithm problem, quantum linear algebra, Jones polynomial approximation etc.

While for others it does not

Unstructured search, collision problem, formula evaluation, st-connectivity, finding subgraphs, subgraph isomorphism problems, certain computational geometry problems, convex optimizations etc.

Most problems can be modelled in the query complexity model.

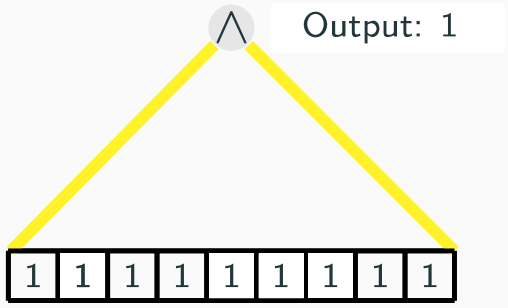
Query complexity model

Ignore the processing time: Only consider the number of times input is read:

- Given a known function $f : \{0,1\}^n \to \{0,1\}$ and
- Given oracle access to the bits of an unknown string $x \in \{0,1\}^n$, compute $f(x)$.

$$\textit{Oracle } O : i \to x_i \textit{ for any } i \in [n].$$

- Cost: Minimum number of bits needed to be queried.
  Example: $AND : \{0,1\}^n \to \{0,1\}, AND(x) = 1$ iff $\forall i \in [n], x_i = 1$.



AND function
D(AND)=n

- The deterministic (or randomized/quantum) complexity of f is the number of queries needed on the worst input x by the best deterministic (or randomized/quantum) algorithm computing f.
- For all f, $Q(f) \leq R(f) \leq D(f) \leq n$.

Relations between quantum and classical computation

No super-polynomial speedups for total functions
Beals, Buhrman, Cleve, Mosca, de Wolf (1998) showed that for all total functions classical and quantum query complexity are polynomially (power 6) related.

Aaronson, Ben-David, Kothari, Tal 2020
For all total functions classical and quantum query complexity are at most power 4 related.

So promises are required for super polynomial speedups.

- $f : P \to \{0, 1\}$.
- If $P = \{0, 1\}^n$, then function is total.
- Otherwise if $P \subset \{0, 1\}^n$, the function is a partial function.

Partial functions and speedup

Do all promised problems give super polynomial speedup?

x_1	x_2	x_3	x_4	AND
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Example: 2-to-1 collision problem
Given black-box queries to the input $[m]^n$ decide whether a sequence of numbers $(x_1, \cdots, x_n) \in [m]^n$ is one-to-one (each number appears once) or two-to-one (each number appears twice).

- $R(f) = \Theta(n^{1/2}), Q(f) = \Theta(n^{1/3})$.

Example: Simon's problem
$f : \{0,1\}^n \to \{0,1\}^n$ is defined as: $f(x) = f(y) \iff (x \oplus y) \in \{0^n, s \neq 0^n\}$. Find out if $s \neq 0^n$.

- $R(f) = \Theta(n^{1/2}), Q(f) = \Theta(\log n)$.

Need of structure for exponential speedups:
Structured basically means that we are trying to determine some property of the input, assuming that the input satisfies some global regularity.

Symmetric Functions

In this direction, it is known that we cannot have super polynomial speedups if the function is too symmetric.

Permutation invariant functions
A partial function $f : P \to \{0, 1\}$ is permutation invariant if for all permutation $\pi \in S_n$:

- $x = (x_1, x_2, \cdots, x_n) \in P \implies x \circ \pi = (x_{\pi_1}, \cdots, x_{\pi_n}) \in P$ and
- $f(x) = f(x \circ \pi)$.

No exponential speedups for symmetric functions (Ambainis-Aaronson 2014, Chailloux 2018)

For all permutation invariant partial functions f,

$$R(f) \leq Q(f)^3.$$

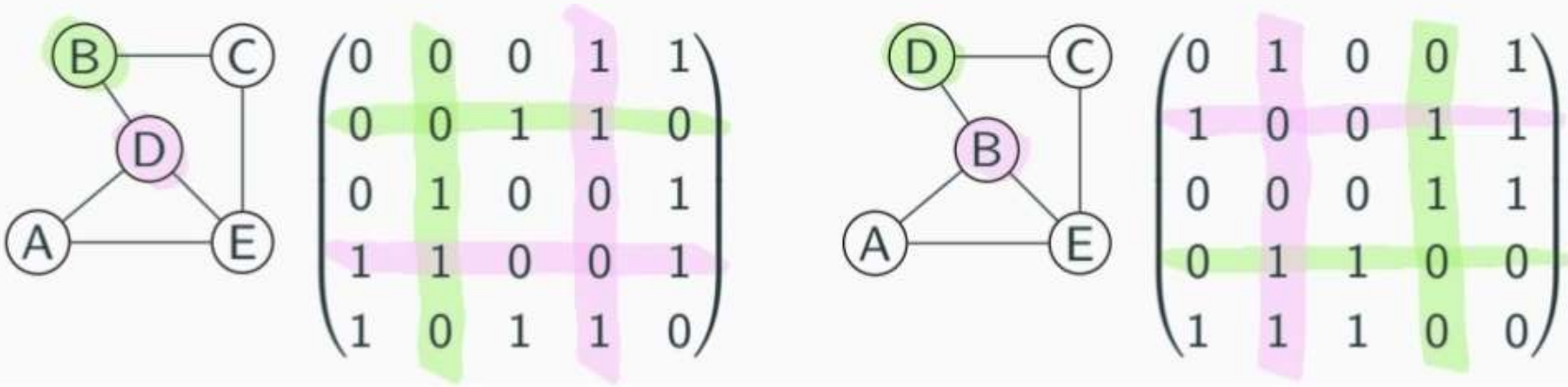
Other Symmetric functions?

Question:
What about other kinds of symmetry?

Symmetric under group action G
Let $f : P \to \{0,1\}$ be a function and $G \subseteq S_n$ be a permutation group. We say f is symmetric under G for every $\pi \in G$:

- $x = (x_1, x_2, \cdots, x_n) \in P \implies x \circ \pi = (x_{\pi_1}, \cdots, x_{\pi_n}) \in P$ and
- $f(x) = f(x \circ \pi)$.

Graph properties



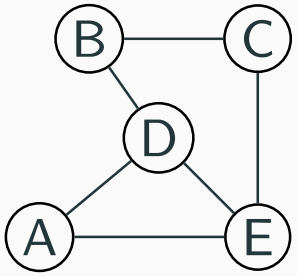
Graph properties: less symmetry

- Graph properties are invariant under vertex renaming, i.e., property depends only on the isomorphism class.
- A graph $G \in P$, then for any $\pi \in S_n$, $\pi(G) \in P$ and $f(G) = f(\pi(G))$.
- An n-vertex graph is given as an oracle access to its edges (in the adjacency matrix representation).

$$O : (i, j) \to A_{ij}.$$

- Vertex permutation induces edge permutation implies much less symmetry: $(n! \ll (n^2)!)$.

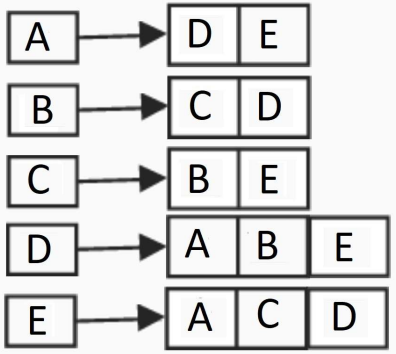
Graph properties



Graph

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency matrix



Adjacency list

Question: (Ambainis, Childs, and Liu 2010 and Montanaro and de Wolf 2013)

Can we show that exponential speedups are not possible for any graph properties (in both adjacency matrix and adjacency list model)?

In fact, they asked the question about property testing problems.

Our Results

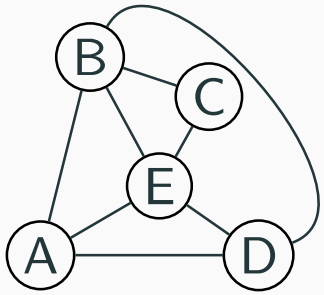
Our results: Graph properties

No super-polynomial speedups is possible for graph properties in the adjacency matrix model.
For all k-uniform hypergraph properties f:

$$R(f) = O(Q(f)^{3k}).$$

Corollary
For all graph properties f in the adjacency matrix model:

$$R(f) = O(Q(f)^{6}).$$



Same holds for directed graph symmetries, bipartite graph symmetries etc.

Our results - graph properties in the adjacency list model

Exponential speedups is possible for a property testing problem in adjacency list model

There is a graph *property testing* problem in the adjacency list model that can be solved with $poly(n)$ quantum queries, but requires $2^{\Omega(n)}$ classical queries.

Hence, for graph properties whether or not we can get large quantum speedups depends on what model the graph is represented in.

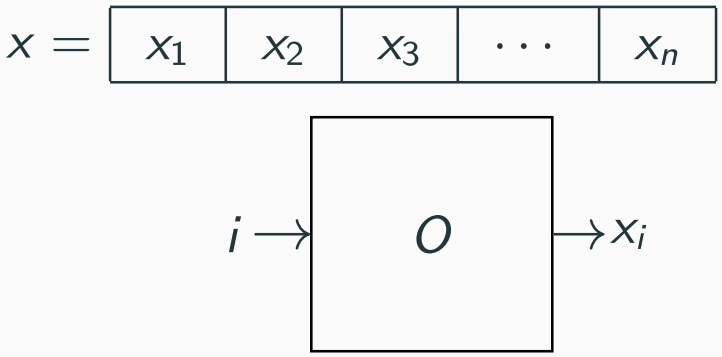
Our results - other types of symmetries

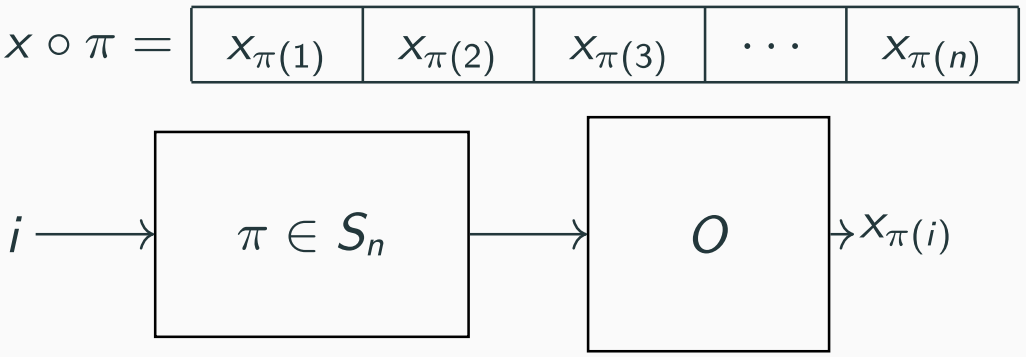
Question:
Can we classify all permutation groups G such that functions f symmetric under G have $R(f) = O(Q(f)^{O(1)})$?

Our result (Informal)
Permutation groups constructed out of hypergraph symmetries are essentially the only permutation groups that prevent super-polynomial quantum speedups.

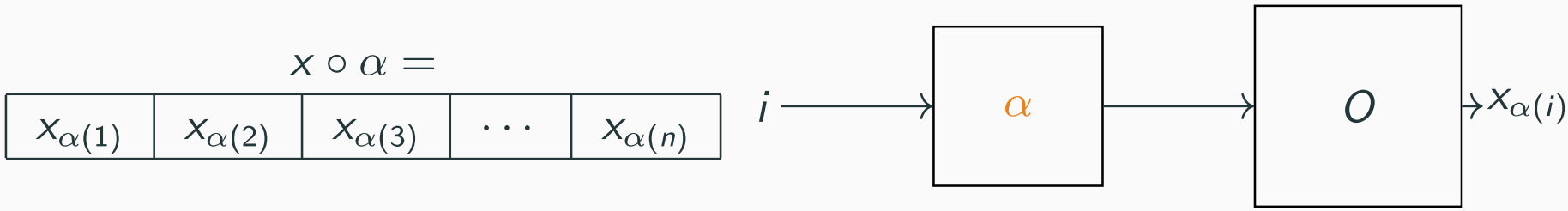
No super-polynomial speedups for graph symmetries

Chailloux's proof and extension to graph symmetries





Consider a function with small range, $\alpha : [n] \to [n]$ with $|\alpha([n])| = r$.



Theorem (Zhandry 2015)

Distinguishing a random small range function $\alpha : [n] \to [n]$ with $|\alpha([n])| = r$ from a random permutation $\pi \in S_n$ requires $\Omega(r^{1/3})$ quantum queries.

- Thus if $r \approx Q(f)^3$, then the quantum algo does not distinguish $x \circ \pi$ from $x \circ \alpha$.
- Quantum algorithm on $x \circ \alpha$ can be simulated with r classical queries.

Graph symmetries

- More generally, if distinguishing a random function with range r from a random permutation $\pi \in G$ requires $\Omega(r^{1/c})$ quantum queries, then $Q(f) = O(R(f)^c)$ for every f symmetric under G.
- We show this by reductions to S_n.
- Furthermore, we show that a version of Zhandry's result that is sufficient for our purposes follows easily from the collision lower bound.

Characterization of speedups for other more general symmetries

Primitive groups

Existence of large quantum speedups for any sufficiently "small" permutation group

Let G be a permutation group on $[n]$ and f be a partial function. Then we can design a partial function g symmetric under G such that:

- $Q(g) \leq Q(f) + b(G)$.
- $R(g) \geq R(f)$.

Theorem (Liebeck 1984)

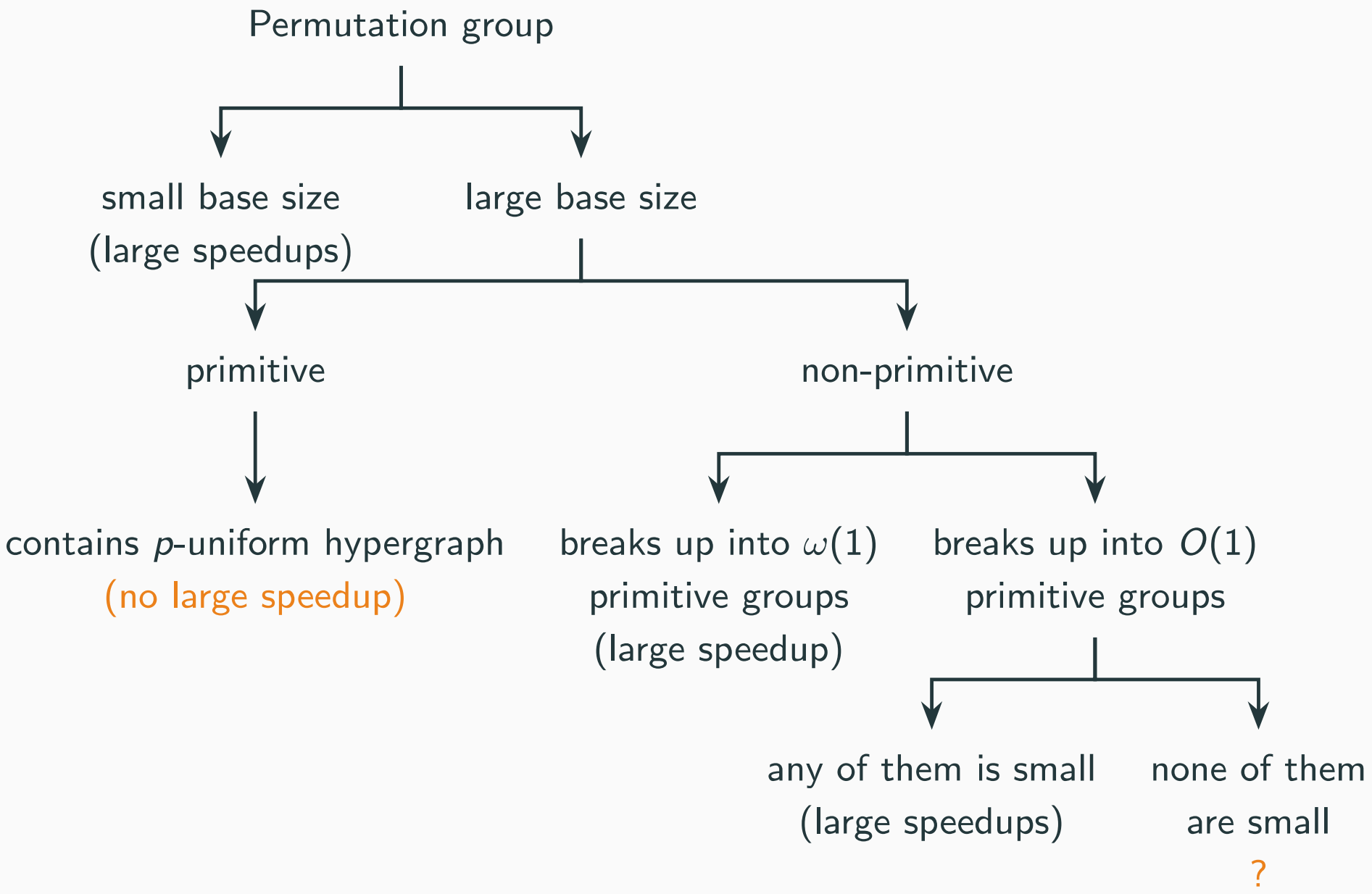
- For primitive groups either $b(G)$ is really small, or
- G contains the symmetries of a p-uniform hypergraph.

Full characterization of primitive permutation groups

Let G be a primitive permutation group. Then:

- If base size of G is $n^{o(1)}$, then there exists a class of partial functions f symmetric under G, $R(f) = Q(f)^{\omega(1)}$.
- If base size of G is $n^{\Omega(1)}$, then for all partial functions f symmetric under G, $R(f) = Q(f)^{O(1)}$.

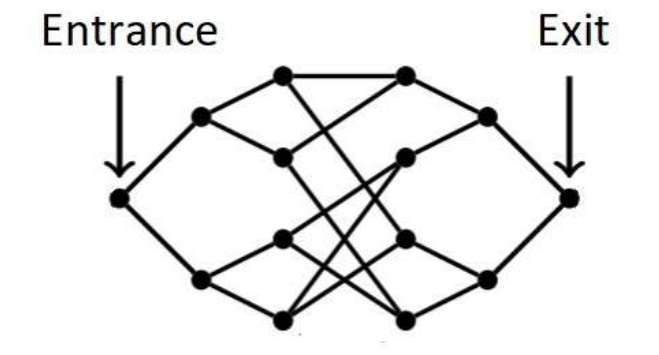
General symmetries



Here "large" denotes super-polynomial speedups.

Exponential speedups of graph property testing problem in the adjacency list model

Exponential speedups by welded trees



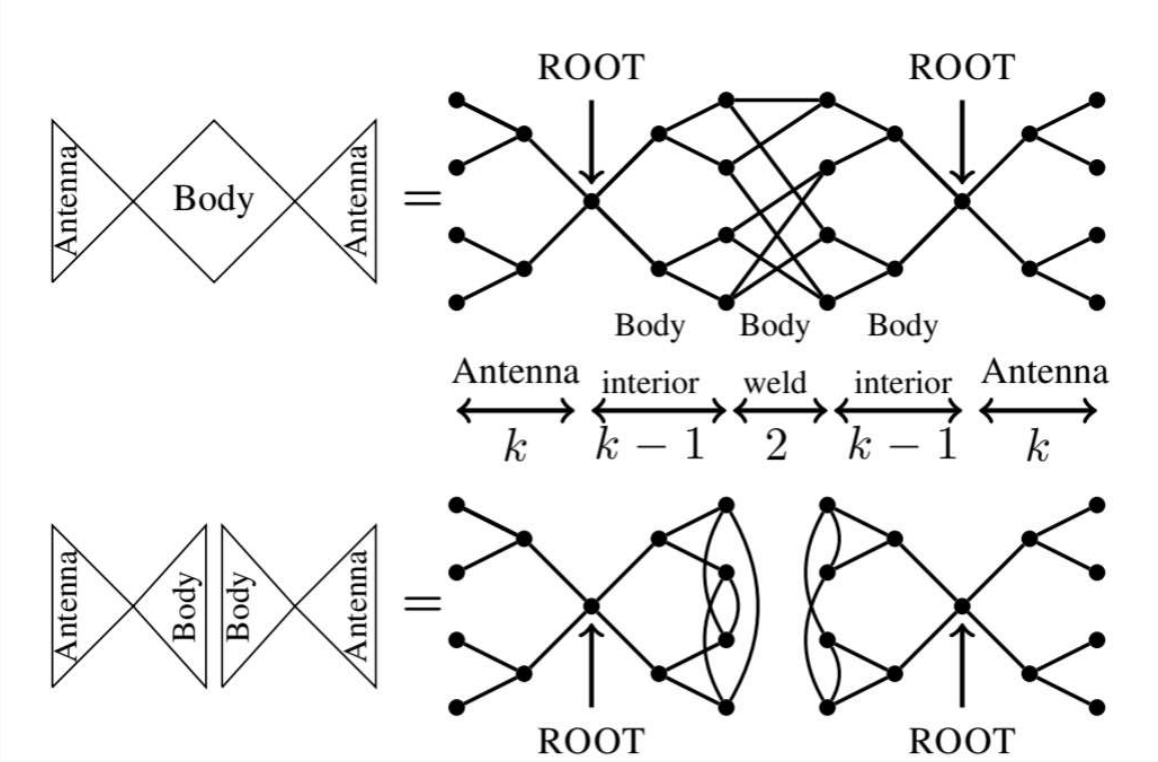
Welded trees

Welded trees problem
Given the adjacency list representation of two welded trees and given the entrance vertex, find the exit vertex.

Theorem: Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 2003

Given a welded trees problem, a quantum walk algorithm finds the *exit* vertex in time $poly(n)$, but a randomized classical algo takes $2^{\Omega(n)}$ time.

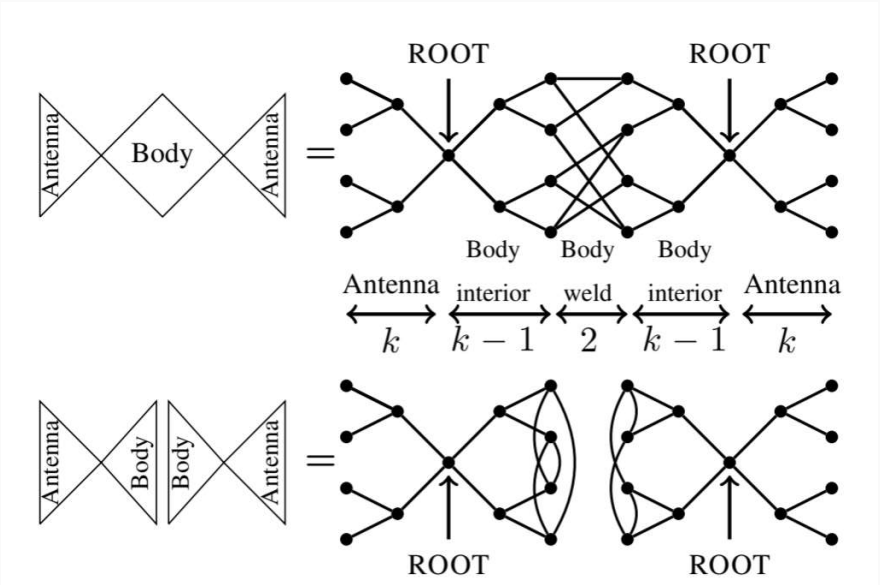
Graph property speedups

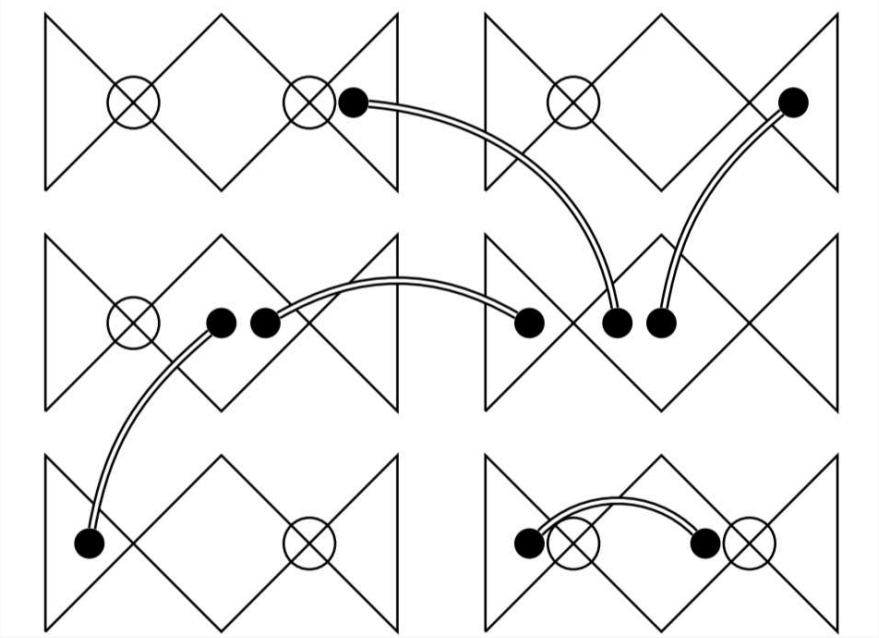


Candy graph

- Connect two large binary trees ("antenna") to the entrance and exit vertex.
- A random vertex is a leaf of the antenna with about $1/2$ probability.
- From such a vertex it is easy to find the entrance
- And then quantum algorithm can quickly find the exit vertex, whereas classical algorithm still needs exponential time to do so with high probability.

Graph property testing speedups



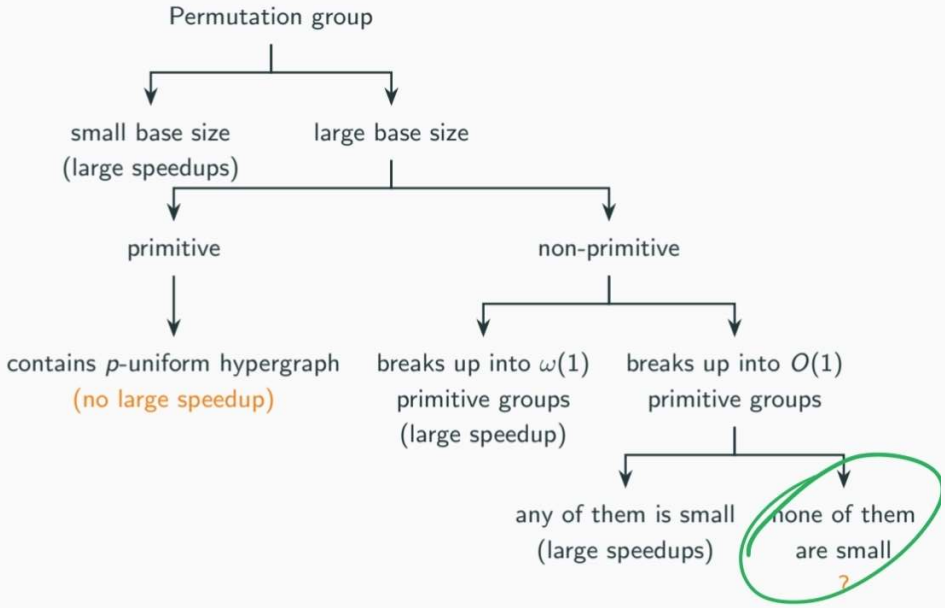


Property testing

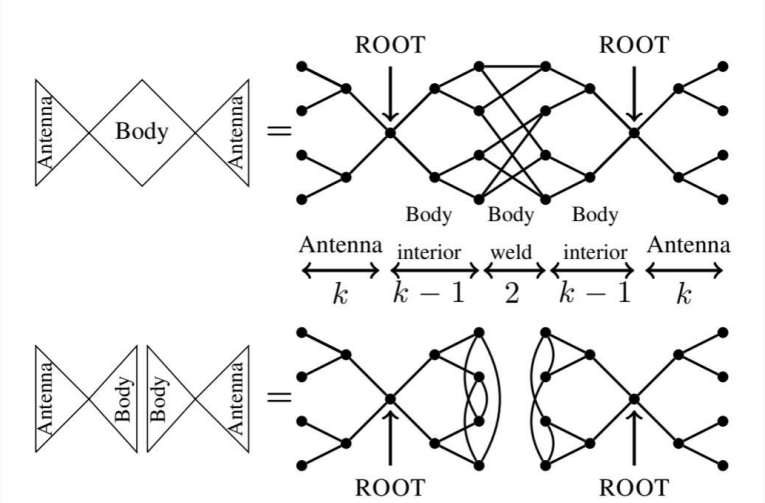
- Goal: Design a structure that is quantumly efficient, but classically hard to distinguish even from a ϵ-far graph.
- Main idea: use many copies of the candy graph, add "advice edges" that let quantum algorithm compute if a vertex is in the weld.
- Given this "advice", the problem essentially reduces to testing a binary tree which can be done efficiently even classically.
- Yet, computing the advice require traversing the welded tree which is hard for any classical algorithm.

Open questions

Can we fully characterize super-polynomial speedups for all permutation groups?



Can we find exponential speed up for a practical graph problem in the adjacency list model?



Thank you so much for listening!!

Full version available at arXiv:2006:12760