On polynomially many queries to NP or QMA oracles

Sevag Gharibian Dorian Rudolph

PADERBORN UNIVERSITY

Introduction

What lies beyond NP?

- Stockmeyer [7] introduces the *polynomial hierarchy* $\mathrm{PH} = \mathrm{P}^{\mathrm{NP}^{\mathrm{NP}^{\cdots}}}$.
- On the second level of PH is $\Delta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}}$.
- P^{NP}: A poly-time Turing machine has access to an oracle for a NP-complete problem.
  - Note: $\mathrm{coNP} \subseteq \mathrm{P}^{\mathrm{NP}}$.

What if we restrict the Turing machine's queries?

- $\mathrm{P}^{\mathrm{NP[log]}}$: TM may ask at most $O(\log n)$ oracle queries.
- $\mathrm{P}^{\|\mathrm{NP}}$: TM must ask all queries at the same time.

Query graphs allow a finer restriction:

- Generally, in P^{NP} the i-th query may depend on the answers to prior queries.
- We can model these dependencies as a graph, each node representing a circuit performing a single query based on prior answers (Fig. 1).

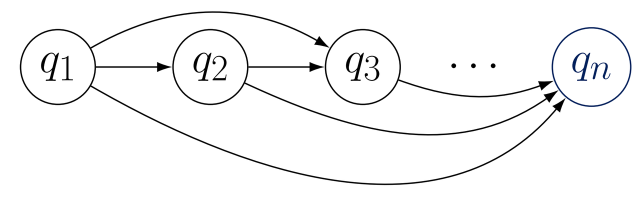

Figure 1. General (complete) query graph

- By the restricting the graph structure, we obtain classes between $\mathrm{P}^{\|\mathrm{NP}} = \mathrm{P}^{\mathrm{NP[log]}}$ [5] and P^{NP}.
- $\mathrm{P}^{\|\mathrm{NP}}$ has *non-adaptive* queries (Fig. 2).

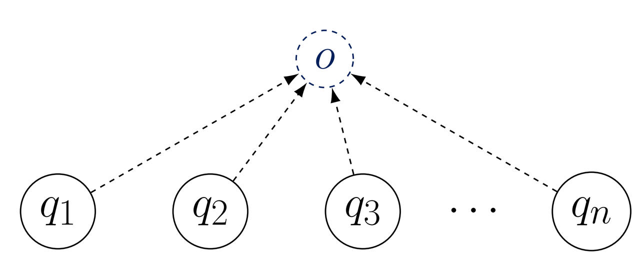

Figure 2. Query graph of adaptive queries (output node o simulates postprocessing by the TM)

- Gottlob [3] considers the restriction of query graphs to trees and shows $\mathrm{TREES(NP)} = \mathrm{P}^{\mathrm{NP[log]}}$.

Quantum: Do the $\mathrm{P}^{\mathrm{NP[log]}}$ results translate to QMA?

- Known: $\mathrm{P}^{\|\mathrm{QMA}} = \mathrm{P}^{\mathrm{QMA}}$ [2] (also for StoqMA).
- Previously open: Can QMA query trees be evaluated with $O(\log n)$ QMA-queries, i.e., $\mathrm{TREES(QMA)} = \mathrm{P}^{\mathrm{QMA[log]}}$?

Main Results

Our results hold for various complexity classes. Let $C \in \{\mathrm{NP}, \mathrm{MA}, \mathrm{QCMA}, \mathrm{QMA}, \mathrm{QMA(2)}, \mathrm{NEXP}, \mathrm{QMA_{exp}}, \mathrm{NEXP}\}$.

Definition:

- C-DAG$_s$: Query graph has *bounded separator number* s.
  - $\mathrm{s}(G)$: minimum s, such that every subgraph of G has a balanced separator of size $\leq s$.
  - $\mathrm{s}(G) \leq O(\mathrm{tw}(G)) \leq O(\log n \cdot \mathrm{s}(G))$ [4].
- C-DAG$_d$: Query graph has *bounded depth* d.

Selected Theorems:

- $C\text{-DAG}_s \subseteq \mathrm{DTIME}(2^{O(s(n)\log(n))})^{C[s(n)\log(n)]}$.
- $\Rightarrow$ $C\text{-DAG}_s = \mathrm{P}^{C[\log(n)]}$ for $s(n) = O(1)$.
  - Includes query graphs of bounded treewidth.
- $\Rightarrow$ $C\text{-DAG}_s \subseteq \mathrm{QP}^{C[\log^{k+1}(n)]}$ for $k \in \mathbb{N}$ and $s(n) = \log^k(n)$.
  - First quasi-polynomial bounds.
- Analogous results hold for C-DAG$_d$ (d = depth).

These results are the first progress on P^{NP} since '95.

Embedding C-DAG into APX-SIM

The Approximate Simulation Problem (APX-SIM) is $\mathrm{P}^{\mathrm{QMA[log]}}$-complete [1] (simulate measurements on Hamiltonian's ground state).

- Lifting Lemma [8] embeds $\mathrm{P}^{\|\mathrm{QMA}}$ into APX-SIM, preserving *geometric properties* (e.g. locality) of the *circuit-to-Hamiltonian* mapping.
  - *Geometric properties* model actual physical systems.
  - *Circuit-to-Hamiltonian* mappings embed QMA-queries into a Hamiltonian's ground state [6].

We extend the lifting lemma to the

Generalized Lifting Lemma: Poly-time many-one reduction from C-DAG$_s$, C-DAG$_d$ ($d, s = O(1)$), satisfying geometric properties of circuit-to-Hamiltonian mapping.

Proof sketch — $\mathrm{QMA\text{-}DAG}_s \subseteq \mathrm{P}^{\mathrm{QMA[log]}}$

Theorem: $\mathrm{QMA\text{-}DAG}_s \subseteq \mathrm{P}^{\mathrm{QMA[log]}}$ for $s = O(1)$.

Proof: similar structure to Gottlob [3], but with novel graph transformation, total solution weight function, and verification procedure to deal with QMA and promise problems.

- Model QMA-queries as QMA-circuits Q_i with proof $|\psi_i\rangle$ and input $z_i = \bigcirc_{j \to i} x_j$ (Fig. 3).
- Each node v_i implicitly defines a promise problem $\Pi^i \in \mathrm{QMA}$.
- $x_i = 1$ if $z_i \in \Pi^i_{\mathrm{yes}}$, $x_i = 0$ if $z_i \in \Pi^i_{\mathrm{no}}$.
- If $z_i \in \Pi^i_{\mathrm{inv}}$ (promise problem!), then $x_i = 0/1$ (nondeterministic).
- However, for a valid query graph $x_n = 1$ (or 0), for all nondeterministic choices.
- **Goal:** Compute x_n with $O(\log n)$ QMA-queries.

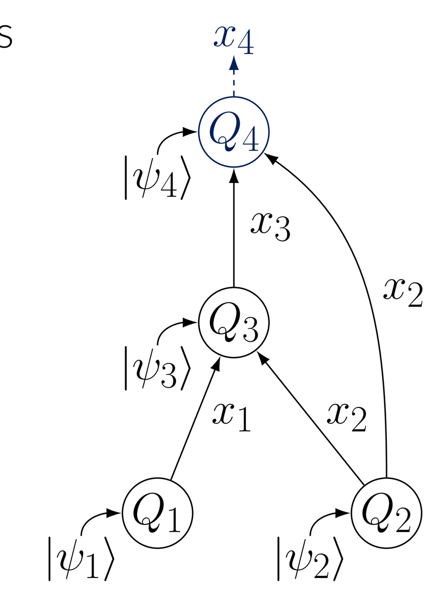

Figure 3. Query graph with proofs

- **Idea:** Total solution weight function $t(x, \psi_1, \ldots, \psi_n) = \sum_{i=1}^n f(v_i) g_i(x_i, z_i, \psi_i)$ should be maximal for *correct* $x, \psi_1, \ldots, \psi_n$.
  - g_i penalizes accepting/rejecting proof for $x_i = 0/1$.
- *Admissable weighting function* f: $f(v) \geq 1 + c \sum_{v \to w} f(w)$ for a chosen constant $c > 1$.
  - Prioritize satisfying queries with more descendants.
  - $\omega(v) := (c+1)^{|\mathrm{Desc}(v)|}$ is admissable, $\mathrm{Desc}(v)$ denoting the descendants of v in G.
- If $f(v) \leq \mathrm{poly}(n)$, we can approximate $T := \max t(\cdots)$ with $O(\log n)$ QMA-queries.
- However, f may be exponential in general!

Graph transformation: Ensure $f \leq \mathrm{poly}(n)$.

Split graph recursively using balanced separators to build a separator tree (Fig. 4).

- Supernodes (dashed) are separators in subgraph induced by descendants.
- Edges of G need not follow separator tree structure: output node is y_2.
- Height of separator tree is $O(\log n)$, separator size $\mathrm{s}(G)$.

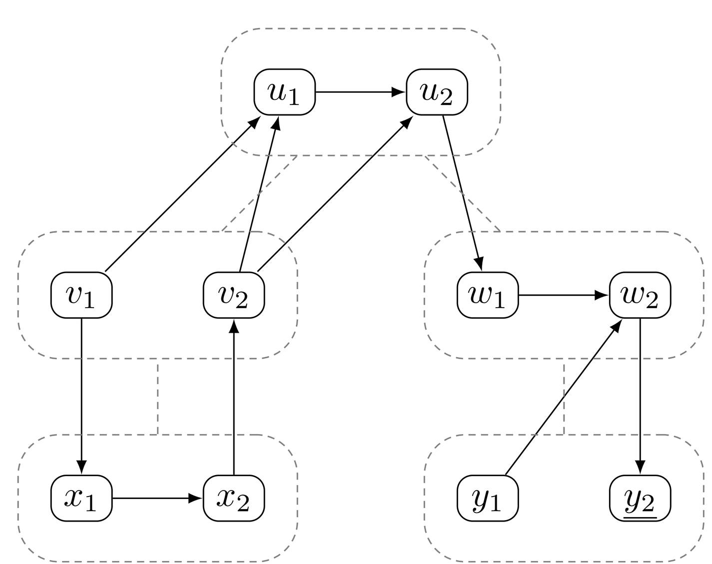

Figure 4. G with separator tree (dashed)

To transform G to G', create copies of every node conditioned on nodes above and in the same separator (Fig. 5).

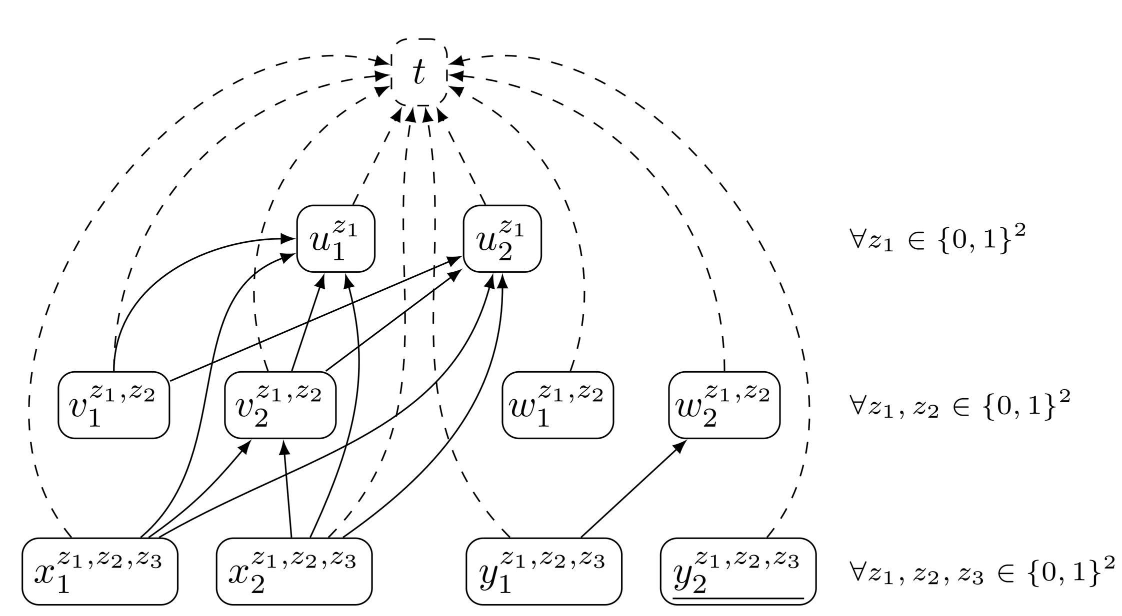

Figure 5. Transformed graph G' with output node (dashed)

- E.g., $v_2^{z_1,z_2}$ assumes outputs z_1, z_2 for u_1, u_2, v_1, v_2.
  - Chooses copy of x_2 conditionally, using output of x_1.
  - Hence, v_2 no longer depends on v_1, and x_2 on x_1.
- Output node t computes correct z_1, z_2, z_3 and returns output of $y_2^{z_1,z_2,z_3}$ (copy of G's output node).
- G' has only edges to ancestors in the separator tree.
  - Nodes have $O(s\log(n))$ descendants and $\omega(v) \leq \mathrm{poly}(n)$.

Inconsistency problem: Construction so far works for NP, but not *promise problems*.

- *Invalid* queries have nondeterministic outputs ($z_i \in \Pi^i_{\mathrm{inv}}$).
- Creating multiple copies of an invalid query can lead to a wrong output (Fig. 6).
- **Solution:** Transform G' to G'' by merging redundant copies (Fig. 7).
- Copies $v^z, v^{z'}$ are redundant if bits of z, z' corresponding to actual dependencies of v are equal.

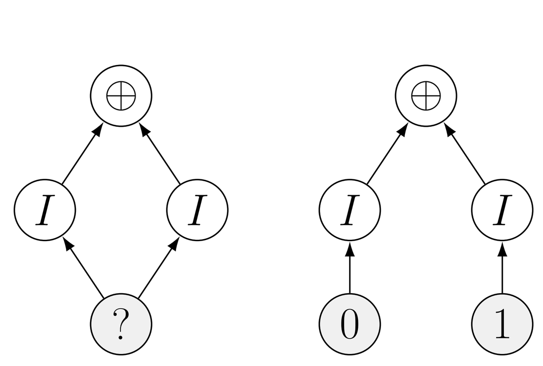

Figure 6. Copies of invalid queries can cause wrong output.

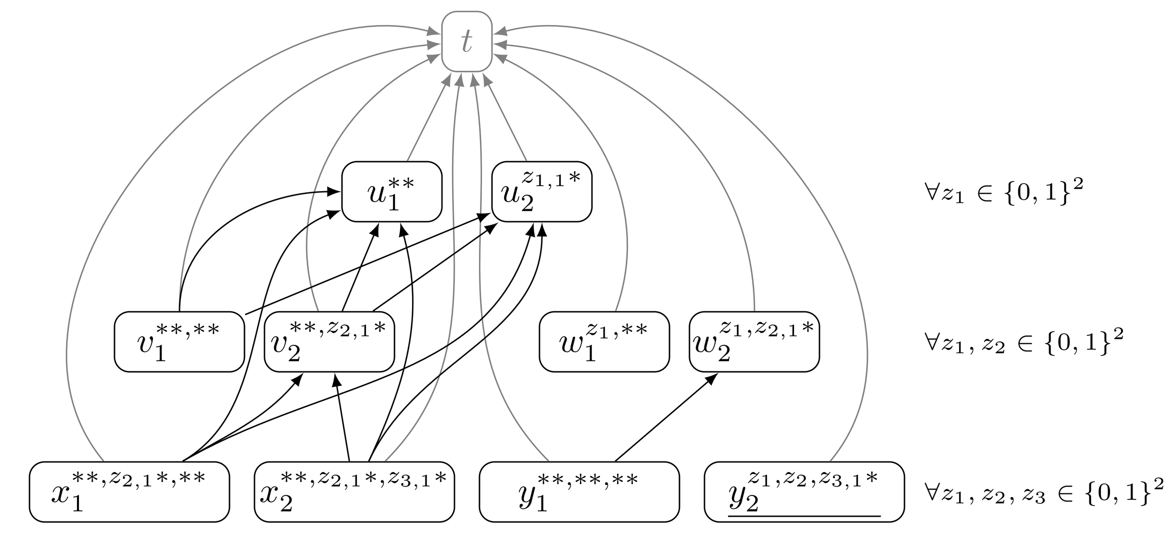

Figure 7. Graph G'' with merged redundant nodes (indicated by $*$)

Total solution weight function:

$$t(x, \psi_1, \ldots, \psi_n) = \sum_{i=1}^n f(v_i) \underbrace{\left(x_i \Pr[Q_i(z_i, \psi_i) = 1] + (1 - x_i)\frac{1}{2} \right)}_{g_i(x_i, z_i, \psi_i)},$$

- g_i penalizes *incorrect* x_i:
  - An acc. proof increases weight by $\geq 1/6$ over non-acc. (assuming $2/3, 1/3$ completeness/soundness parameters)

Lemmas:

- Can determine whether $T \geq s$ or $T \leq s - \varepsilon$ with QMA-queries ($T = \max t(\cdots)$).
- If $t(x, \psi_1, \ldots, \psi_n) \geq T - \varepsilon$, then x is correct.

Therefore, we can determine output x_n using binary search, proving $\mathrm{QMA\text{-}DAG}_s \subseteq \mathrm{P}^{\mathrm{QMA[log]}}$.

Reduction sketch to APX-SIM

Goal: Given QMA-DAG G with $\mathrm{s}(G) = O(1)$, construct an equivalent APX-SIM-instance with a given circuit-to-Hamiltonian mapping H_{w}.

Use QMA-verifier V (Fig. 8) estimating T, operating on registers $\mathcal{X} \otimes \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_n$ (query string and proofs), output q_{flag}:

- Measure $\mathcal{X}$ in standard basis to obtain query string x.
- Select i with probability $p_i := f(v_i)/W$.
- If $x_i = 1$, return output of $Q_i(z_i, \mathcal{Y}_i)$.
- Else, output 1 with probability $1/2$.

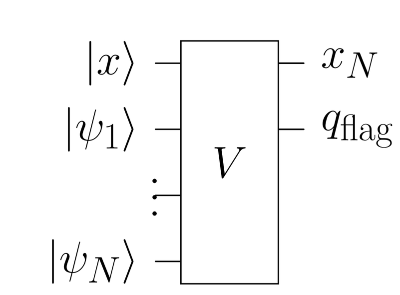

Figure 8. Verifier estimating T

Then,

$$\max_{|\psi\rangle} \Pr[V(|\psi\rangle) = 1] = T/W$$

- Ground state of Hamiltonian $H_{\mathrm{w}}(V)$ is the *history state* $|\phi\rangle = \sum_{i=0}^L (V_i \cdots V_1 |\phi_0\rangle)|i\rangle$.
- Define APX-SIM instance:
  - Hamiltonian $H := \alpha H_{\mathrm{w}}(V) + |0\rangle\langle 0|_{q_{\mathrm{flag}}} \otimes |L\rangle\langle L|$
    - $\alpha H_{\mathrm{w}}(V)$ ensures ground state is a history state.
    - $|0\rangle\langle 0|_{q_{\mathrm{flag}}} \otimes |L\rangle\langle L|$ ensures $|\phi_0\rangle$ maximizes acc. prob.
  - Measurement $A := |0\rangle\langle 0|_{x_N}$ checks output of G.

Hence, the ground state of H is a history state with $|\phi_0\rangle = |x\rangle|\psi_1\rangle \cdots |\psi_N\rangle$ maximizing $t(x, \psi_1, \ldots, \psi_N)$. x is a correct query string with output x_N.

Open Problems

- Extend results to other classes such as QMA_1, UQMA, or StoqMA ($d = O(1)$ result also holds for StoqMA).
- Does $\mathrm{tw}(G) = O(\mathrm{s}(G))$ hold?
- Compute or approximate separator trees efficiently.
  - For C-DAG$_s$ brute force is good enough.
  - However, there exist graphs G with $O(1)$-size separator tree, but $\mathrm{s}(G) = \Theta(\log(n))$.
- We show for $s(n) = \log^k(n)$, $C\text{-DAG}_s(C) \subseteq \mathrm{QP}^{C[\log^{k+1}(n)]}$. Can we improve the base to P?

References

[1] A. Ambainis. In: *CCC*. 2014, pp. 32–43.
[2] S. Gharibian, S. Piddock, and J. Yirka. In: *STACS*. Vol. 154. 2020, 20:1–20:37.
[3] G. Gottlob. In: *J. ACM* 42.2 (Mar. 1995), pp. 421–457.
[4] H. Gruber. In: *J. Comb.* 3.4 (2012), pp. 669–681.
[5] L. A. Hemachandra. In: *J. Comput. Syst. Sci.* 39.3 (Dec. 1989), pp. 299–322.
[6] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi. 2002.
[7] L. J. Stockmeyer. In: *Theor. Comput. Sci.* 3.1 (Oct. 1976), pp. 1–22.
[8] J. D. Watson, J. Bausch, and S. Gharibian. 2020. arXiv: `2012.12717 [quant-ph]`.

  {sevag.gharibian,dorian.rudolph}@upb.de