Quantum Complexity Theory Notes

Xiao Geng

Prelude:

Quantum Complexity Theory is difficult but important. So we will have a survey but no proofs.

When you can't find an algorithm to solve a problem, maybe try proving that it cannot be solved (by complexity).

Classes:

Typically organize decision problems into classes with similar computational difficulty.

See later that it is usually not hard to frame problems into decision (0/1 answer) problems.

Classes that we've already seen:

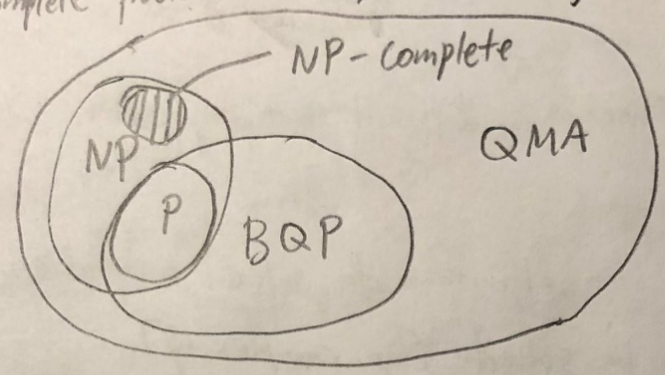
- P = solvable in polynomial time by a classical computer
- BPP = solvable in polynomial time with > 2/3 success rate by a classical computer.
- BQP = solvable in polynomial time with > 2/3 success rate by a quantum computer.

Consensus views:

First recall the definitions of NP and NP-complete:

- NP = decision problems where a solution can be verified in P.

- NP-complete = the hardest class of problems in NP, where if any NP-complete problem is in P, then all of NP must be in P.



Now, $P \neq NP$ is strongly believed by computer scientists but unproven, whereas physicists use it as a natural law. This would mean no NP-complete problems can by solved in poly time.

$P = BPP$ is strongly believed but unproven (i.e. generally believed that the use of randomness doesn't add anything)

$P \neq BQP$ is proven, <u>assuming factor is not P</u>.

- Hamiltonian Simulation is in BQP but not in NP.

<u>$NP \not\subseteq BQP$ is strongly believed but unproven</u>, whereas physicists use this as a natural law.

- Factoring is in BQP but is <u>not</u> NP-complete.
- In particular, it is strongly believed that quantum computers <u>cannot</u> solve NP-complete problems.

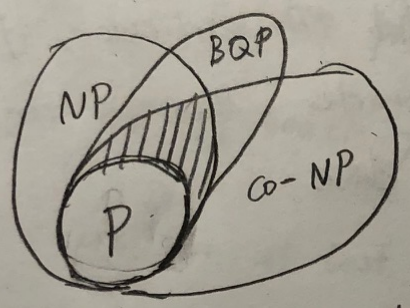
<u>QMA (Quantum-Merlin-Arthur)</u> is the quantum analog of NP.

- i.e. A problem in QMA if it is a decision problem if one can verify any solution in BQP.
- $BQP \neq QMA$ is strongly believed but unproven (follows from $P \neq NP$)

Heuristics

co-NP: For a decision problem in NP, for any input such that the output (decision) is "yes", NP requires for there to be a certificate verifiable in poly time. But the choice of "yes" above is arbitrary. If we swap it for "no", then we would get co-NP.

- It is believed that NP ≠ co-NP, and BQP includes a number of problems in NP ∩ co-NP, including factoring



- Related to "statistical zero-knowledge". Very interesting and lots of open (and difficult) problems.

Quantum Speedups

- Shor's Algorithm (factoring): exponential speedup.
- Grover's Algorithm (search): quadratic speedup.

How do we know what kind of speedup we can expect for different problems?

Quantum Query Complexity: Running time measured in terms of function calls to the Oracle (queries).

- In Deutsch-Josza the oracle is a function f that is either balanced or constant.

Query complexity is a bit problematic because each query might take exponential time, so an algorithm might have poly query complexity but takes in fact exponential time.

Also problematic: query complexity uses oracles, which are completely black boxes. But in reality we are likely able to study the circuit of the oracle.

However, query complexity is still helpful. In particular, if an algorithm does not study the circuit of the oracles, then we can translate query complexity to normal complexity classes

Now, need to restrict to decision problems.

- Grover's algorithm can be converted to a decision problem: if $f=0$ everywhere then output "yes", otherwise "no".
  - Apply QPE to $|\psi\rangle$ in the Grover iteration.
  - Grover rotates $|\psi\rangle$ if there is a solution. So if the phase is 0, then f must be constant.
- Simon's problem — similar to other problems we've gone through.

☆ Key property is whether any function (oracle) is allowed.

- Deutsch-Josza restricts to constant/balanced (exp. speedup)
- Grover allows any function (quadratic speedup)

In general, quatum speedups at most poly if any f is allowed.

Beals et. al. (1998):

Allowing any functions, if there is a quantum algorithm solving a problem with T queries, then there is a classical algorithm that solves it in $O(T^6)$ queries.

Polynomial (root 6) speedup.

Further, the bound tightens to $O(T^2)$ when answer to the problem is unchanged if the bits passed to the oracle are permuted

- e.g. replacing oracle f with $g(a,b,c) = f(b,a,c)$.

$\Rightarrow$ Hence we can at best achieve a quadratic speedup for Grover's algorithm.

- The decision problem for Grover remains the same no matter how you permute the input to f.

Aaronson et al. (2020):

Improved general bound to $O(T^4)$.

- Requires Boolean Fourier Analysis
  - Fun and not too hard, but too long a detour here.
- Relies on breakthrough result of Hao Huang (the sensitivity Conjecture)

Note: No restrictions on $f \Rightarrow$ no exponential speedup.
but Some restrictions on $f \not\Rightarrow$ exponential speedup.

Intuition: Quantum algs take advantage of <u>special structures</u> in functions. (e.g. special algebraic properties)

- See the Hidden Subgroup Problem later.
- QFT is a key ingredient.
  - There are many hidden subgroup problems that rely on different QFTs.