Final Report

D10943012 電子博一 梁峻瑋

1. Classical computer system

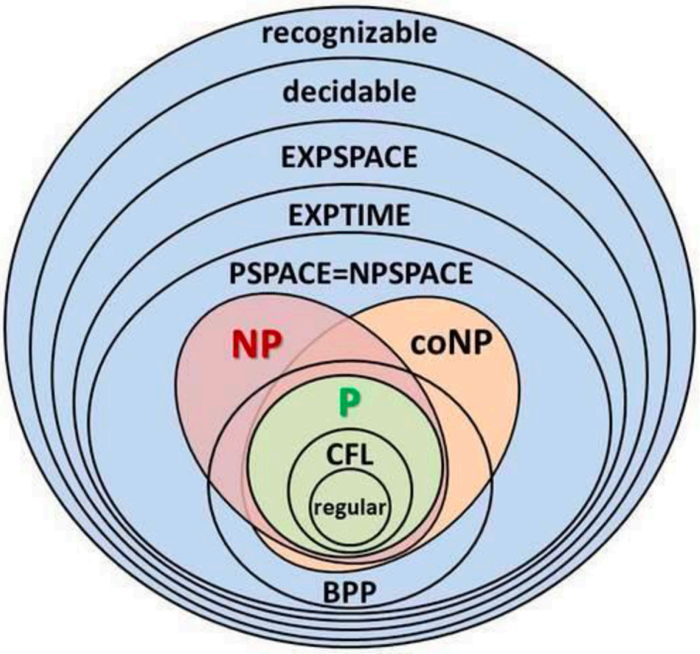
In this part, we want to introduce several typical computational complexities of the classical computer system. First of all, let's introduce the definitions and relationships of P, NP, coNP, and BPP. The follows few quotes and a figure from [1]:

➔ For P: "In complexity theory, the class P contains all decision problems solvable by a deterministic Turing Machine in polynomial time."

➔ For NP: "The Nondeterministic Polynomial time (NP) class of problems is a category of decision problems that is solvable in polynomial time by a non-deterministic Turing Machine."

➔ For coNP: "The coNP are the class of problems that have a polynomial-time algorithm mapping for "no-instance" solutions which can be used to verify that the proposed solution is valid but there is no such mapping for "yes-instances". The P class is a subset of both coNP and NP."

➔ For BPP: "The bounded-error probabilistic polynomial time (BPP) is a class of problems solved by a probabilistic Turing Machine in polynomial time with an attached probability distribution function with a given error degree. BPP can be interpreted as the complexity class P with a randomness boundary factor."



[1][Fig1. Diagram representation for the many categories of computational complexity]

Note that PSPACE is the category of problems that can solve by a Turing machine with polynomial input size. Such as the AlphaGo example mentioned in the class.

So far, we cannot provide an example that is in “NP \ P”, “coNP \ P”, or “NP \ coNP”. If such an example exists, it is equally to say: “P ≠ NP”, which is still a significant open problem in the CS field.

2. Quantum computer system

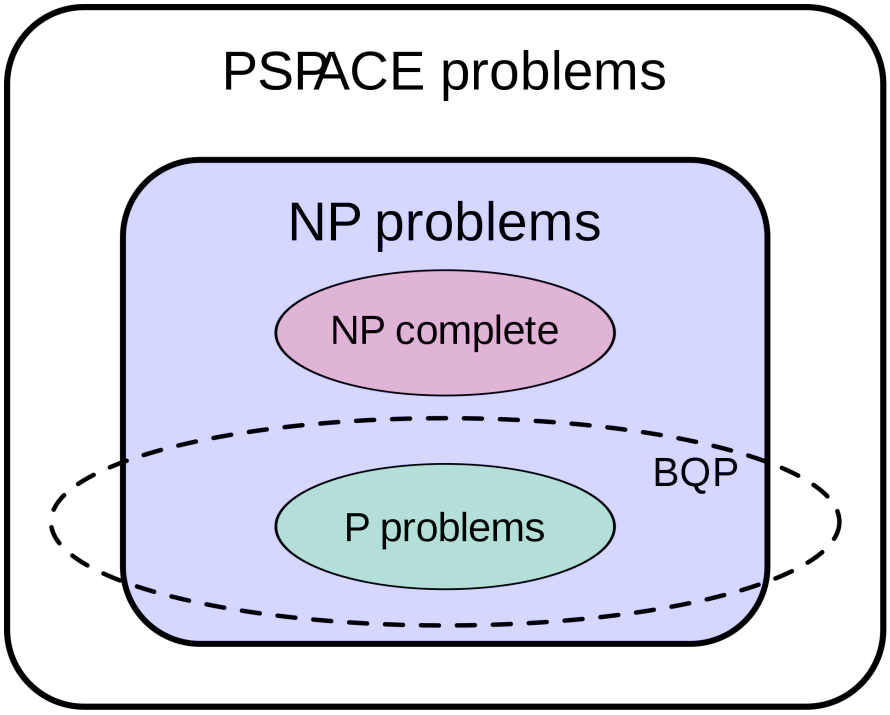
This part will locate the BQP complexity category——a set of problems solvable by the polynomial–time complexity algorithm running on a quantum computer with a particular error probability.

From the reference [2] in 1997, we know “BQP ≦ PP”, and so “BQP ≦ PSPACE”. The idea in this paper is first to prove “$BQP_{ploy(1/\varepsilon)}$ ≦ PP ≦ $P_{\#P}$”. Then, we imply “BQP ≦ PP” by arguing that “$BQP_{ploy(1/\varepsilon)}$, BQP_{BV}, BQP_{Q}“ didn’t have too much difference. Finally, we can easily obtain “BQP ≦ PSPACE” by the previously known fact.

On the other hand, “BPP ≦ BQP”. Thanks to the universal quantum gate theorem, we can reproduce AND, OR, XOR gates by the set of “one 2–qubit entangled gate and all 1–qubit gates”. As a result, we also can produce all 0/1–circuit, all boolean algebra functions, and so to run all algorithms in the quantum computer with polynomial time complexity.

Finally, through the reference [3], BQP is low for itself, which means BQP^{BQP} = BQP. But we ignore the definition of “A^A” form here.

A figure from the wiki can describe our learning so far:



[4][Fig2. The suspected relationship of BQP to other problem spaces]

3. Suspect “BPP < BQP”——Does Quantum Advantage Exist?

So far, we have two chains of the relationships of complexity category:

$$P \subseteq NP \cap coNP \subseteq PSPACE$$
$$P \subseteq BPP \subseteq BQP \subseteq PSPACE$$

How about the “quantum advantage”? We have learned some examples in the course, such as Integer factorization or Discrete logarithm. These examples are based on the effectively finding factors of the integer (Shor’s order-finding algorithm) and the tool of quantum Fourier transform.

Te be more specific, let’s take HW3-Q2 as an example: let’s use N=21, t=9, a=2 and measurement result y = 427 to show Shor’s algorithm:

➔ Step1: $y/N = 427/512 = 1/(1+1/(5+1/(42+1/2))) = [1,5,42,2]$
➔ Step2: $[1,5]=5/6$ is the first a[i] satisfies $|y/N-l/r| \leqq 1/(2N^2)$, obtain r=6
➔ Step3: Check r=6 is the period of $f(x)=2^x \bmod N$
➔ Step4: $f(6) = 2^6-1 \bmod 21 = (2^3+1)(2^3-1) \bmod 21 = 0$
Since $\gcd(2^3+1, 21)\neq 1$, $\gcd(2^3-1,21)\neq 1$; we successfully obtain 21=3*7

In Step1&2, we use the sequence of continued fractions, a sequence of $O(n^2)$ convergence rates, to approximate the y/N. Due to its $O(n^2)$ convergence rates, it ensures the existence of answers on the sequence. Moreover, the $O(n^2)$ convergence rates and $|y/N-l/r| \leqq 1/(2N^2)$ also promise the uniqueness of the answer. Otherwise, “$|y/N-l_0/r_0| \leqq 1/(2N^2)$, $|y/N-l_1/r_1| \leqq 1/(2N^2)$ $|l_0/r_0-l_1/r_1| \leqq 1/(N^2)$” can derive the contradiction.

In Step3&4, we expect r to be an even period of f(x) and $2^{0.5r}+1$, $2^{0.5r}-1$ is not caprice with N. In this case, we successfully finish the task of factor N. Otherwise, with a certain probability, we fail the mission and do the algorithm again and again.

To conclude, we can solve the integer factorization problem on the quantum computer in $O(n^3)$ time complexity. On the other hand, the best record algorithm on the classical computer is exponentially slow (about $2^{O(n^{1/3})}$). This is a so-called example of “quantum advantage”.

However, it is still not enough to say “BPP < BQP”. Although we have found efficient algorithms on the classical computer for decades, polynomial-time algorithms still can exist on the classical computer, with relatively low probability.

4. Suspect “NP ⊀ BQP”——An example in NP, but not in BQP?

Until now, we know that “P ⊆ BPP ⊆ BQP”. Also, we cannot make sure whether “P = NP” or “P < NP”. Suppose there exists an example(X) in NP but not in BQP; this means X is an example in NP but not in P, equally to claim “P < NP”. It is a milestone in solving the P versus NP problem.

The suspect example is the Graph Isomorphism Problem[5]: "check to see if two graphs look differently are actually the same"? First of all, it is clearly in NP. Given a solution to Graph Isomorphism Problem, a claimed isomorphism mapping relation Then, we can check whether each pair of vertices has an edge or not in C(n,2) time to verify the correctness. Hence it is in NP.

Furthermore, from [6] in 1988, we know that the Graph Isomorphism Problem is a particular case of the Subgraph Isomorphism Problem, a special case of an NPC problem. Some specific case of the Graph Isomorphism Problem is in P, such as the case of the trees and the case of Interval graphs. People also guess the Graph isomorphism problem is in the hierarchy between NPC and P.

So, why do we suspect the Graph Isomorphism Problem is not in BQP? Referring to [7] in 2016, they proposed a more robust system than BQP, named naCQP. From the article, it mentions:
➔ “The class is defined by imagining that quantum computers can perform (non-adaptive) measurements that do not collapse the wave-function.”

➔ “This non-physical model of computation can efficiently solve problems such as Graph Isomorphism and Approximate Shortest Vector which are believed to be intractable for quantum computers.”

➔ “This is surprising as most modifications of BQP increase the power of quantum computation to NP or beyond”

We cannot solve the NP-hard problem in polynomial time with the block box method, roughly between $N^{1/4}$ to $N^{1/3}$ time complexity. But, this robust model still can solve the “lower NPC hierarchy” problem, such as the Graph Isomorphism Problem.

To conclude, the Graph Isomorphism Problem is in NP and naCQP, but may not be in BQP. As a result, it could be an example in “NP \ BQP”, also an essential example of “P≠NP”.

5. Other – news about quantum & NPC problems

Following the ending of 4., the studying the mainstream of quantum & NPC problems assumes the quantum computer is not enough to polynomially solve NPC problems, such as the Graph Isomorphism Problem is not in BQP. Then, they propose a robust model and try to solve NPC problems polynomially.

First of all, [8] in 1993 proposed a special quantum computer to solve some NPC problems polynomially, but it also consumes a lot of energy. Later in 2000, a popular robust quantum model was published [9]: “quantum adiabatic behavior/evolution.” The adiabatic evolution relies on the Adiabatic Theorem and is a kind of quantum annealing. Initially, [10] in 2001 randomly generated small NPC instances and successfully solved them polynomially. This work is a big step toward solving NPC problems and quantum computers!

Secondly, [11] in 2008 proposed a standard quantum algorithm with the chaotic dynamical system to solve SAT problem polynomially. More specifically, “chaotic behavior in a classical system is usually considered an exponential sensitivity to initial conditions.”

Finally, some funny idea also connects the quantum computer and NP–complete problems. [12] in 2014 proved that “computing quantum discord is NP–complete”. As a result, with the idea that the quantum computer is not enough to solve NP–complete problems, quantum discord is “intractable”. This article is an interesting but helpful way to say something is not achievable on the quantum computer.

6. Reference

[1] Lima, Matheus Sant'Ana. "Information theory inspired optimization algorithm for efficient service orchestration in distributed systems." Plos one 16.1 (2021): e0242285.

[2] Adleman, Leonard M., Jonathan DeMarrais, and Ming-Deh A. Huang. "Quantum computability." SIAM Journal on Computing 26.5 (1997): 1524-1540.

[3] Bernstein, Ethan, and Umesh Vazirani. "Quantum complexity theory." SIAM Journal on computing 26.5 (1997): 1411-1473.

[4] https://en.wikipedia.org/wiki/BQP#/media/File:BQP_complexity_class_diagram.svg.

[5] Fortin, Scott. "The graph isomorphism problem." (1996).

[6] Schöning, Uwe. "Graph isomorphism is in the low hierarchy." Journal of Computer and System Sciences 37.3 (1988): 312-323.

[7] Aaronson, Scott, et al. "The space" just above" BQP." Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science. 2016.

[8] Černý, Vladimír. "Quantum computers and intractable (NP-complete) computing problems." Physical Review A 48.1 (1993): 116.

[9] Farhi, Edward, et al. "Quantum computation by adiabatic evolution." arXiv preprint quant-ph/0001106 (2000).

[10] Farhi, Edward, et al. "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem." Science 292.5516 (2001): 472-475.

[11] Ohya, Masanori, and Igor V. Volovich. "New quantum algorithm for studying NP-complete problems." Selected Papers Of M Ohya. 2008. 83-90.

[12] Huang, Yichen. "Computing quantum discord is NP-complete." New journal of physics 16.3 (2014): 033027.