41076: Methods in Quantum Computing

'Quantum Complexity' Module

Min-Hsiu Hsieh

Centre for Quantum Software & Information, Faculty of Engineering and Information Technology, University of Technology Sydney

Abstract

Contents to be covered in this lecture are

1. BQP
2. Local Hamiltonian problem is QMA-complete
3. QIP=PSPACE
4. IQP and Boson sampling

In this module, we will introduce fundamental results in quantum complexity theory. The purpose of this module is to understand what problems can be efficiently solved on a quantum computer, and what problems remain difficult even if we have a quantum computer. Understanding the boundary between different complexity classes will help us to better identify the power of quantum computation versus classical computation.

1 Overview of complexity theory and definitions

In the computational complexity theory, the goal is to prove *lower* bounds on the time or space resources required by the best possible algorithm for solving the problem.

1.1 Computational models

Complexity classes are defined in terms of computational models, including

- Turing machines;
- Interactive proof systems;
- Boolean circuits;
- Quantum Turing machines;
- Quantum interactive proof systems;

1.1.1 Turing machine

A Turing machine, which manipulates symbols on a strip of tape according to a set of rules, is a mathematical model that represents a general computing machine. The deterministic Turing machine (DTM) uses a fixed set of rules to determine its future actions; while the probabilistic (Nondeterministic) Turing machine is allowed to explore multiple possible future actions from a given state.

As a computational model, the Turing machine is able to compute an enormous variety of functions, such as basic arithmetical operations, searching, and simulation of all operations performed on a modern computer. Its theoretical foundation is summarized by the Church-Turing thesis:

Theorem 1 (Church-Turing thesis)**.** *The class of functions computable by a Turing machine corresponds exactly to the class of functions which we could naturally regard as being computable by an algorithm.*

Exercise 2. *Show that the Turing machine cannot solve the 'halting problem': does the machine with Turing number x halt upon input of the number y?*

A quantum Turing machine can be defined in a way similar to the classical Turing machine. It provides an abstract model that can capture the computation power of a quantum computer.

The (quantum) Turing machine will be used to define the complexity classes: **P**, **BPP**, **BQP**, **NP**, and **QMA** in the subsequent section.

1.1.2 Interactive proof systems

An interactive proof system is an abstract machine that models computation as exchange of messages between two spatially separated parties, a prover and a verifier, where a prover is assumed to have unlimited computational power but cannot be trusted, while the verifier is assumed to be always honest. All interactive proof systems have to satisfy the following two conditions:

- Completeness: If the statement is true, the honest verifier can be convinced by the untrusted prover.

- Soundness: If the statement is false, no prover can convince the honest verifier.

If the prover and the verifier are given access to quantum computation, then we have the *quantum interactive proof system.*

The (quantum) interactive proof system will be used to define complexity classes: **IP** and **QIP** in the subsequent section.

1.2 List of complexity classes

Computational problems can be formulated as *decision problems*; namely, problems with a *yes* or *no* answer.

The following is a summary of major complexity classes.

- **P**: This class contains decision problems that can be solved by a deterministic Turing machine in polynomial time.

- **BPP** (bounded-error probabilistic polynomial time): This class contains decision problems that can be solved by a probabilistic Turing machine in polynomial time with an error probability bounded away from 1/3 on every input.
- **NP** (nondeterministic polynomial time): This class contains decision problems for which the problem instances whose answer is "yes" have proofs verifiable in polynomial time by a deterministic Turing machine.
- **PP** (probabilistic polynomial time): This class contains decision problems that is solvable by a probabilistic Turing machine in polynomial time, with an error probability less than 1/2 for all instances.
- **IP** (Interactive polynomial time): The class contains problems solvable by an interactive proof system, where the verifier is a probabilistic polynomial-time machine with access to a random bit string polynomial on the input size, and a polynomial number of messages are exchanged before decision.
- **PSPACE**: This class contains all decision problems that can be solved by a Turing machine using a polynomial amount of space. Note that **PSPACE=IP**.
- **BQP** (bounded-error quantum polynomial time): This class contains decision problems that can be solved by a quantum computer in polynomial time with an error probability bounded away from 1/3 on every input. It is the quantum analogue to the complexity class **BPP**.
- **QMA** (Quantum Merlin Arthur): This class contains decision problems for which the problem instances whose answer is YES, have a polynomial-size quantum proof (a quantum state) that convinces a quantum verifier in polynomial time with high probability. Moreover, when the answer is NO, every polynomial-size quantum state is rejected by the verifier with high probability. It is is the quantum analog of the complexity class **NP**.
- **QIP** (Quantum Interactive Polynomial time): The class contains the set of problems solvable by an interactive proof system with a polynomial-time verifier and one computationally unbounded prover.

There are natural inclusions of these complexity classes as follows (see also Figure 1).

$$\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{BQP} \subseteq \mathbf{QMA} \subseteq \mathbf{PP} \subseteq \mathbf{PSPACE}. \tag{1}$$

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{QMA} \subseteq \mathbf{PSPACE}. \tag{2}$$

One of the fundamental results is **QIP=PSPACE** [4].

2 BQP: Efficient Quantum Computation

By definition, the class of **BQP** contains problems that can be efficiently solved if we have a quantum computer. In order to show that a specific problem is **BQP**, the solution involves an efficient quantum algorithm. Therefore it is important to know which kind of problems they are.

The following gives a formal definition of **BQP**. Denote by Σ^* the collection of all (finite) strings of symbols from the alphabet Σ. A language L over the alphabet Σ is a subset of the set Σ^*.

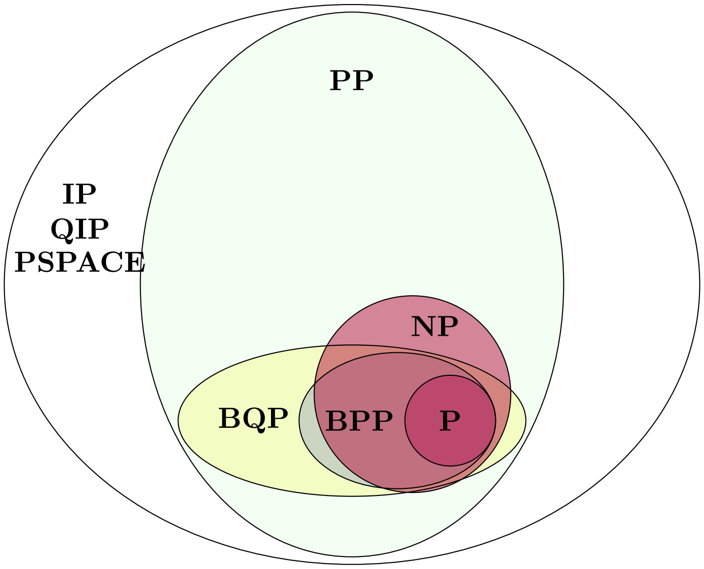

Figure 1: The relations between complexity classes. Note that (i) the relation between **BPP** and **NP** is not known; (ii) the relation between **BQP** and **NP** is also unknown.

Definition 3 (BQP)**.** *A language $L = \{L_{\text{yes}}, L_{\text{no}}\}$ is in $\boldsymbol{BQP}(c, s)$ if and only if there exists a polynomial-time uniform family of quantum circuits $\{Q_n : n \in \mathbb{N}\}$, where Q_n takes n qubits as input and outputs 1 bit, such that*

- $\forall x \in L_{\text{yes}}$,
$$\Pr\{Q_{|x|}(x) = 1\} \geq c;$$
- $\forall x \in L_{\text{no}}$,
$$\Pr\{Q_{|x|}(x) = 1\} \leq s.$$

The class $\boldsymbol{BQP}$ is defined as $\boldsymbol{BQP}$=$\boldsymbol{BQP}(2/3, 1/3)$.

In general, c and s in Definition 3 are polynomial functions: $\mathbb{N} \to [0, 1]$.

Exercise 4. *Let $r : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function and $r(n) \geq 2$ for all n. Then*

$$\boldsymbol{BQP} = \boldsymbol{BQP}(1 - 2^{-r}, 2^{-r}).$$

Exercise 5 (BQP subroutine theorem)**.** *Show that $\boldsymbol{BQP}^{\boldsymbol{BQP}} = \boldsymbol{BQP}$.*

We collect a list of **BQP** problems:

1. Integer factorization and discrete logarithm.
2. Approximating the Jones polynomial at certain roots of unity.
3. Quadratically signed weight enumerator problem.
4. Local Hamiltonian Eigenvalue Sampling.
5. Phase Estimation Sampling.

2.1 Integer factorization and discrete logarithm

Integer factorization and discrete logarithm both belong to special cases of the hidden subgroup problems for finite Abelian groups, and share some common properties. The difficult of finding solutions to these problems classically has been used to construct various cryptographic applications, including RSA, Diffie-Hellman key agreement, ElGamal encryption, the ElGamal signature scheme, the Digital Signature Algorithm, and the elliptic curve cryptography.

2.1.1 Integer factorization

The task of integer factorization is to decompose a composite number into a product of smaller integers. If the factors are further restricted to prime numbers, it is known as prime factorization. This problem is believed to be hard (outside **P**) because the best known classical algorithms are sub-exponential in the problem size. Specifically, the best classical algorithm for a b-bit number n requires running time

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}\right). \tag{3}$$

It is generally believed that the integer factorization problem is not in class **P** because there are no known efficient classical algorithms that can factor all integers in polynomial time. The problem is clearly in class **NP** but has not been proved to be or not be **NP**-complete. However, this problem is generally suspected not to be **NP**-complete.

2.1.2 Discrete logarithm

The task of discrete logarithm is to find the solution x to the equation $b^x = a$ given elements a and b of a finite group G.

2.1.3 Shor's algorithm

Peter Shor developed a quantum algorithm (now called Shor's algorithm) that can solve integer factorization and discrete logarithm in polynomial time [8]; hence placing this problem in **BQP**.

2.2 Jones polynomial

The problem of approximating the Jones polynomial of the plat closure of a braid at $2^{i2\pi/k}$ for constant k is **BQP**-complete [2].

2.3 Signed weight enumerator problem

Let $A \in \mathbb{Z}_2^{m\times n}$ and $B \in \mathbb{Z}_2^{n\times n}$. The quadratically signed weigh enumerator is defined as

$$S(A,B,x,y) = \sum_{\boldsymbol{b}:A\boldsymbol{b}=0} (-1)^{\boldsymbol{b}^T B\boldsymbol{b}} x^{|\boldsymbol{b}|} y^{n-|\boldsymbol{b}|}, \tag{4}$$

where $\boldsymbol{b}^T$ is the transpose of the column vector $\boldsymbol{b} \in \mathbb{Z}_2^n$, $|\boldsymbol{b}|$ is the Hamming weight of $\boldsymbol{b}$. We know the following facts about computation of $S(A,B,x,y)$.

- Given positive numbers k and ℓ, computing $S(A,B,k,\ell)$ is in the class #**P**.

- Let $\Gamma(A)$ be the lower triangular part of A, while the diagonal and the upper triangular elements are set to zero. Given positive integers k and ℓ and the promise $|S(A,\Gamma(A),k,\ell)| \geq \frac{(k^2+\ell^2)^{n/2}}{2}$, determining the sign of $S(A,\Gamma(A),k,\ell)$ is **BQP**-complete [6].

2.4 Linear algebraic problems

Definition 6. *Given are (1) a Hamiltonian $H = \sum_i H_i$ on n qubits, where i ranges over a set of size polynomial in n and each H_i acts on a constant number of qubits; (2) an estimation precision $\epsilon = \Omega(1/poly(n))$; (3) a sampling error probability $\delta = \Omega(1/poly(n))$; and (4) a classical n-bit string $b \in \mathbb{Z}_2^n$. Suppose the eigenvalues and eigenvetors of H are $\{\lambda_k, |\nu_k\rangle\}$ satisfying $|\lambda_k| < poly(n)$ for each k. The local Hamiltonian eigenvalue sampling problem asks to output an estimation of λ_k up to ϵ with probability at least $(1-\delta)|\langle b|\nu_k\rangle|^2$.*

Definition 7. *Given are (1) the description of a $poly(n)$-size quantum circuit U on n qubits; (2) an estimation precision $\epsilon = \Omega(1/poly(n))$; (3) a sampling error probability $\delta = \Omega(1/poly(n))$; and (4) a classical n-bit string $b \in \mathbb{Z}_2^n$. Suppose the eigenvalues and eigenvetors of H are $\{\lambda_k, |\nu_k\rangle\}$, where $\lambda_k = e^{i2\pi\varphi_k}$ for each k. The phase estimation sampling problem asks to output an estimation of φ_k up to ϵ with probability at least $(1-\delta)|\langle b|\nu_k\rangle|^2$.*

The local Hamiltonian eigenvalue sampling problem and the phase estimation sampling problem are shown to be **BQP**-complete in Ref. [9].

3 QMA: Quantum NP

The quantum Merlin Arthur (**QMA**) complexity class is a quantum generalization of the **NP**. Problems in **QMA** are considered difficult to solve even with a quantum computer.

Definition 8 (QMA)**.** *A language $L = \{L_{\text{yes}}, L_{\text{no}}\}$ is in **QMA**(c,s) if there exists a polynomial time quantum verifier V and a polynomial $p(x)$ such that*

- *$\forall x \in L_{\text{yes}}$, there exists a quantum state $|\psi\rangle \in \mathcal{H}_2^{\otimes p(x)}$ such that the probability that V accepts the input $(|x\rangle, |\psi\rangle)$ is greater than c, i.e.,*

$$\Pr\{V(|x\rangle, |\psi\rangle) = 1\} \geq c;$$

- *$\forall x \in L_{\text{no}}$, for all quantum states $|\psi\rangle \in \mathcal{H}_2^{\otimes p(x)}$, the probability that V accepts the input $(|x\rangle, |\psi\rangle)$ is less than s, i.e.,*

$$\Pr\{V(|x\rangle, |\psi\rangle) = 1\} \leq s.$$

*The class **QMA** is defined as **QMA**=**QMA**$(2/3, 1/3)$.*

A problem is said to be **QMA**-hard if every problem in **QMA** can be reduced to it, and a problem is said to be **QMA**-complete if it is **QMA**-hard and is in **QMA**.

Exercise 9. *Show that*

$$\boldsymbol{QMA}\left(\frac{2}{3}, \frac{1}{3}\right) = \boldsymbol{QMA}\left(\frac{1}{2} + \frac{1}{q(x)}, \frac{1}{2} - \frac{1}{q(x)}\right) = \boldsymbol{QMA}\left(1 - 2^{-r(x)}, 2^{-r(x)}\right) \tag{5}$$

where $q(x)$ and $r(x)$ are polynomials in the input size x.

3.1 k-local Hamiltonian problem

The k-local Hamiltonian problem is a natural generalization of k-SAT (Boolean satisfiability problem) to the quantum domain. To see this, consider the following formula on n variables $\{X_1, \cdots, X_n\}$.

$$\Gamma = C_1 \wedge C_2 \wedge \cdots \wedge C_r, \tag{6}$$

where each clause C_i is an OR over three variables (or their negations). Take for example $n = 5$, and $C_i = X_2 \vee \neg X_3 \vee X_5$. Hence $(X_2, X_3, X_5) = (0, 1, 0)$ is the only unsatisfying assignment for the clause C_i. We can now embed the problem to quantum domain by defining a projector H_i onto the unsatisfying assignment of C_i, and in this case,

$$H_i = |00100\rangle\langle 00100|. \tag{7}$$

In other words, for any satisfying assignment $\boldsymbol{v} = (v_1, \cdots, v_5) \in \mathbb{Z}^5$ of C_i, we have $H_i|\boldsymbol{v}\rangle = 0$, where $|\boldsymbol{v}\rangle = |v_1\rangle \otimes \cdots \otimes |v_5\rangle$; while $H_i|\boldsymbol{u}\rangle = +1|\boldsymbol{u}\rangle$ if $|\boldsymbol{u}\rangle$ is the unsatisfying assignment. Equivalently, we say that the matrix H_i penalizes the unsatisfying assignment by giving it one unit of energy. Finally let $H = \sum_{i=1}^r H_i$ where each projector H_i corresponds to each clause C_i. We have $H|\boldsymbol{z}\rangle = \mu|\boldsymbol{z}\rangle$, where μ is the number of unsatisfying assignments of the sequence $\boldsymbol{z} \in \mathbb{Z}^n$ to Γ in Eq. (6). Thus the 3-SAT problem is equivalent to whether the smallest eigenvalue of H 0 or is it at least 1?

Definition 10 (k-local Hamiltonian)**.** *The operator $H : \mathcal{L}(\mathcal{H}_2^{\otimes n}) \to \mathcal{L}(\mathcal{H}_2^{\otimes n})$ on n qubits is a k-local Hamiltonian if $H = \sum_{i=1}^r H_i$, where each term H_i is a Hermitian operator acting on at most k qubits.*

Given a k-local Hamiltonian on n qubits $H = \sum_{i=1}^r H_i$ with $r = \text{poly}n$ and each $\|H_i\| \leq \text{poly}(n)$, and let $a < b$ be two constants, the k-local Hamiltonian problem is to decide

- **YES** instance: $\lambda_{\min}(H) \leq a$;
- **NO** instance: $\lambda_{\min}(H) > b$.

Theorem 11 ([5])**.** *The k-local Hamiltonian is QMA-complete for $k \geq 2$.*

4 Boson Sampling, IQP and NISQ

We are currently in the era of having noisy intermediate-scale quantum technologies (NISQ), i.e., the quantum devices with unavoidable noise could only perform specially designed tasks. One of the fundamental questions is whether the NISQ devices could already outperform classical computations in certain tasks. If the answer is no, it means that we must have fault-tolerant quantum computers to see quantum advantages. If the answer is yes, what tasks are they?

In this section, we will explore two proposals; namely the Boson sampling problem [1] and the instantaneous quantum circuits (IQP) [3], that are believed to have potential to answer the aforementioned question.

4.1 Boson Sampling

The Boson sampling problem considers sampling the output of a multimode linear optical circuit of n modes that is injected with m indistinguishable single photons ($n > m$). Suppose that the linear optical circuit is described by the $n \times n$ unitary U, which performs a linear transformation of the creation operator $a_i^\dagger$ of the i-th input mode as follows:

$$b_j^\dagger = \sum_{i=1}^{m} U_{j,i} a_i^\dagger, \tag{8}$$

where $U_{j,i}$ is the (j,i)-th element of U and $b_j^\dagger$ is the creation operator of the j-th output mode. Without loss of generality, we can consider the initial input state of m single photons in n modes as follows:

$$\begin{aligned} |\psi_{\text{in}}\rangle &= |1_1, \cdots, 1_m, 0_{m+1}, \cdots, 0_n\rangle, & (9) \\ &= a_1^\dagger \cdots a_m^\dagger |0_1, \cdots, 0_n\rangle. & (10) \end{aligned}$$

The output state after U is a superposition of different configurations of how the m photons could have arrived in n modes:

$$|\psi_{\text{out}}\rangle := U|\psi_{\text{in}}\rangle = \sum_S \gamma_S |t_1^S, t_2^S, \cdots, t_N^S\rangle, \tag{11}$$

where S is a configuration, t_i^S is the number of photons in the i-th output mode associated with the S configuration, and γ_S is the amplitude. The number of configurations of m single photons in n modes is

$$|S| = \begin{pmatrix} n+m-1 \\ m \end{pmatrix}. \tag{12}$$

The probability of measuring the configuration S is

$$\begin{aligned} p(S) &= |\gamma_S|^2 & (13) \\ &= \frac{\text{Perm }(U_S)^2}{t_1^S! \cdots t_N^S!}, & (14) \end{aligned}$$

where U_S is an $m \times m$ sub-matrix of U and Perm(U_S) is the permanent of the matrix U_S. The expression (14) was first noted in [7]. Hence the computation of γ_S in Eq. (13) is related to computing the permanents of an $m \times m$ matrix, which is known to be #**P**-complete. Moreover, approximation to within multiplicative error of matrix permanents is also #**P**-hard. The concludes that classical simulation of the Boson sampling is extremely hard.

Let us look at two examples that could help us better understand Eq. (14). Consider $m = 2$, and the final configuration $S = \{2, 4\}$, where one photon at mode 2 and one photon at mode 4. In this example, there are two ways that the photons can reach the desired configuration, namely, (i) the first photon moves from mode 1 to mode 2 and the second photon moves from mode 2 to mode 4; (ii) the first photon moves from mode 1 to mode 4 and the second photon stays at mode 2. Therefore,

$$\begin{aligned} \gamma_{\{2,4\}} &= U_{1,2}U_{2,4} + U_{1,4}U_{2,2} & (15) \\ &= \text{perm}(U_S) & (16) \end{aligned}$$

where the 2 × 2 sub-matrix U_S is

$$U_S = \begin{pmatrix} U_{1,2} & U_{2,2} \\ U_{1,4} & U_{2,4} \end{pmatrix}.$$

The next example considers $m = 3$ and the final configuration $S = \{345\}$. Note that there are $3! = 6$ different ways in which three photons can reach this configuration.

$$\begin{aligned} \gamma_{\{3,4,5\}} &= U_{1,3}U_{2,4}U_{3,5} + U_{1,3}U_{2,5}U_{3,4} + U_{1,4}U_{2,3}U_{3,5} && (17) \\ &+ U_{1,4}U_{2,5}U_{3,3} + U_{1,5}U_{2,3}U_{3,4} + U_{1,5}U_{2,4}U_{3,3} && (18) \\ &= \operatorname{perm}(U_S) && (19) \end{aligned}$$

where the 3 × 3 sub-matrix U_S is

$$U_S = \begin{pmatrix} U_{1,3} & U_{2,3} & U_{3,3} \\ U_{1,4} & U_{2,4} & U_{3,4} \\ U_{1,5} & U_{2,5} & U_{3,5} \end{pmatrix}.$$

The boson sampling can be performed using

- Input: Vacuum and single photon states;
- Evolution: A passive linear interferometer;
- Measurement: An on-off photodetector;

which are very easy to implement. Note that these components alone are not universal.

4.2 Instantaneous Quantum Computation (IQP)

An IQP circuit on n qubits is the quantum circuit that only contains commuting quantum gates, say in the X basis, while the initial state is $|0\rangle^{\otimes n}$.

It has been shown that the output distribution generated by uniform families of IQP circuit cannot be classically simulated (otherwise it will violate certain complexity assumptions) [3]. The class of IQP circuits is not universal, and can be easily implemented on the NISQ devices.

References

[1] Scott Aaronson and Alex Arkhipov, *The computational complexity of linear optics*, Proceedings of the forty-third annual acm symposium on theory of computing, 2011, pp. 333?342.

[2] Dorit Aharonov, Vaughan Jones, and Zeph Landau, *A polynomial quantum algorithm for approximating the jones polynomial*, Proceedings of the thirty-eighth annual acm symposium on theory of computing, 2006, pp. 427436.

[3] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd, *Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **467** (2011), no. 2126, 459–472, available at `https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2010.0301`.

[4] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous, *Qip = pspace*, Commun. ACM **53** (December 2010), no. 12, 102109.

[5] Julia Kempe, Alexei Kitaev, and Oded Regev, *The complexity of the local hamiltonian problem*, SIAM Journal on Computing **35** (2006), no. 5, 1070–1097, available at `https://doi.org/10.1137/S0097539704445226`.

[6] E. Knill and R. Laflamme, *Quantum Computation and Quadratically Signed Weight Enumerators*, arXiv e-prints (September 1999), quant–ph/9909094, available at `quant-ph/9909094`.

[7] Stefan Scheel, *Permanents in linear optical networks* (2004), available at `arXiv:quant-ph/0406127`.

[8] Peter W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing **26** (1997), no. 5, 1484–1509, available at `https://doi.org/10.1137/S0097539795293172`.

[9] Pawel Wocjan and Shengyu Zhang, *Several natural bqp-complete problems*, 2006.