Upper Bounds in Quantum Complexity

Review: QMA is the set of problems with a succinct quantum witness that can be verified efficiently by a QC. The local Hamiltonian problem and the consistency of reduced density matrices problem are QMA-complete.

The "interesting" part of both of these results was to show that these problems are QMA-hard ("lower bounding the complexity"). In contrast, putting them in QMA ("upper bounds") was relatively easy.

Today we turn to the converse subject: upper bounding the power of quantum complexity classes (and the hardness of quantum-related tasks) in terms of classical complexity classes.

In computational physics, we want to upper bound the hardness of quantum-related tasks by showing they are in P (or BPP).

In complexity theory, we have more options because we can prove upper bounds involving powerful wizards. For example, if some restricted version of LH is in NP, then that could be an important and surprising result.

Upper Bounds in Quantum Complexity

Consider BQP, the class of problems solvable by a quantum computer with bounded error in polynomial time. How powerful does a classical computation need to be to simulate BQP?

One brute-force method would be to use an exponential amount of time and space (memory) to simulate classically simulate BQP by matrix-vector multiplication in the full Hilbert space. [proof by MATLAB]

This can be done in the classical complexity class EXP, which is analogous to P but runs for exponential time.

Definition (exponential time). $L \in \mathrm{EXP}$ if there exists a polynomial p and a verifier $\{V_r\}_{r\in\mathbb{N}}$ such that

$$x \in L \implies V_{\exp(p(|x|))}(x) = 1$$

$$x \notin L \implies V_{\exp(p(|x|))}(x) = 0$$

Upper Bounds in Quantum Complexity

Let's get an explicit worst-case upper bound on the asymptotic runtime of this classical simulation.

Suppose the quantum circuit has n qubits, starts in $|0^n\rangle$, and has m gates $U_1, ..., U_m$ that are at most 2-local. How much time and space does MATLAB use, in the worst case?

$$\text{TIME} = \mathcal{O}(m \cdot 2^n) \qquad \text{SPACE} = \mathcal{O}(2^n)$$

Note that in general we may not be able to parallelize any of the gates. (can you give an example?)

A $2^n \times 2^n$ matrix has 4^n. Therefore these time and space bounds require a sparse representation of the matrix. How many entries are in each row of the matrix (assume U is 2-local)?

$$I \otimes I \otimes \ldots \otimes I \otimes U \otimes I \otimes \ldots \otimes I$$

Upper Bounds in Quantum Complexity

This description of a MATLAB program shows that $\mathrm{BQP} \subseteq \mathrm{EXP}$. What about $\mathrm{QMA} \subseteq \mathrm{EXP}$?

We can show $\mathrm{QMA} \subseteq \mathrm{EXP}$ in two ways. One would be to write a loop that goes over all possible witnesses (precision?) and simulates the circuit evaluated on each witness. This would take:

$$\mathrm{TIME} = \mathcal{O}(m \cdot 2^n) \qquad \mathrm{SPACE} = \mathcal{O}(2^n)$$

Another method to show $\mathrm{QMA} \subseteq \mathrm{EXP}$ would be to use MATLAB to solve the local Hamiltonian problem. The Hamiltonian on n qubits is a $2^n \times 2^n$ matrix. An N x N matrix can be diagonalized in time $\mathcal{O}(N^3)$ by Gaussian elimination (or a little faster by better methods). MATLAB has “eigs” for sparse matrices.

The number of non-zero entries per row of the Hamiltonian is upper bounded by a constant times the number of local terms, so it is polynomial. Therefore the eigs-based simulation of LH requires

$$\mathrm{TIME} = \mathcal{O}(poly(n) \cdot 2^n) \qquad \mathrm{SPACE} = \mathcal{O}(poly(n) \cdot 2^n)$$

Upper Bounds in Quantum Complexity

So far these results are relatively straightforward: everything we can do with a polynomial amount of quantum space seems to be brute-force simulable using an exponential amount of classical resources.

We can obtain a tighter upper bound on BQP by changing the way we represent quantum mechanics. Instead of the Schrodinger representation we've used so far, we'll shift to the Feynman representation.

If the output state of the quantum circuit is $|\psi\rangle = U_m \dots U_1 |0^n\rangle$, then the acceptance probability is

$$p_{\text{accept}} = \langle\psi|A|\psi\rangle = \langle 0^n|U_1^\dagger \dots U_m^\dagger A U_m \dots U_1|0^n\rangle$$

Where $A = |1\rangle\langle 1| \otimes I$. The Feynman path integral expresses this probability as a sum over exponentially many "paths",

$$\sum_{z_1,\dots z_{2m} \in \{0,1\}^n} \langle 0^n|U_1^\dagger|z_1\rangle\langle z_1|U_2^\dagger|z_3\rangle \dots \langle z_{m-1}|U_m^\dagger|z_m\rangle\langle z_m|A|z_{m+1}\rangle\langle z_{m+1}|U_m|z_{m+2}\rangle \dots \langle z_m|U_1|0^n\rangle$$

Upper Bounds in Quantum Complexity

Therefore we can compute the acceptance probability p_{accept} of a BQP circuit as

$$\sum_{z_1,\ldots z_{2m}\in\{0,1\}^n} \langle 0^n|U_1^\dagger|z_1\rangle\langle z_1|U_2^\dagger|z_3\rangle\ldots\langle z_{m-1}|U_m^\dagger|z_m\rangle\langle z_m|A|z_{m+1}\rangle\langle z_{m+1}|U_m|z_{m+2}\rangle\ldots\langle z_m|U_1|0^n\rangle$$

Each path $(z_1, ..., z_{2m})$ is described by $2nm = \text{poly}(n)$ bits. For a given $(z_1, ..., z_{2m})$, how long does it take to compute the amplitude

$$\langle 0^n|U_1^\dagger|z_1\rangle\langle z_1|U_2^\dagger|z_3\rangle\ldots\langle z_{m-1}|U_m^\dagger|z_m\rangle\langle z_m|A|z_{m+1}\rangle\langle z_{m+1}|U_m|z_{m+2}\rangle\ldots\langle z_m|U_1|0^n\rangle$$

It takes time $\mathcal{O}(nm)$, so each contribution to this circuit path integral takes poly(n) time to compute.

Upper Bounds in Quantum Complexity

More importantly, each amplitude only takes poly(n) space to compute. We can loop through the amplitudes and keep a running total of the sum of all amplitudes we've seen so far.

Each amplitude takes poly(n) space to compute, then we can add it to the running total and erase / reset the bits that we used to compute the amplitude. By erasing as we go, we are careful to never use more than poly(n) space. This algorithm needs resources:

$$\mathrm{TIME} = \mathcal{O}(4^m) \qquad \mathrm{SPACE} = \mathcal{O}(n+m)$$

The important point is that although the algorithm still requires an exponential amount of time, it now only needs a polynomial amount of space.

The set of problems that a classical computer can solve in polynomial space, with no restriction on time, is PSPACE. Therefore we have $\mathrm{BQP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$.

Upper Bounds in Quantum Complexity

The Feynman representation allows us to see $\mathrm{BQP} \subseteq \mathrm{PSPACE}$. To get a better feel for PSPACE, we can describe a PSPACE-complete problem called OTHER END OF THIS LINE (OEOTL).

Let G = (V,E) be a directed graph where every vertex is associated with an n bit string. The edges are represented by polynomial sized circuits S and P, with an edge from u to v if

$$S(u) = v \quad , \quad P(v) = u$$

Therefore G contains vertices of degree at most 2, so it is a union of paths, cycles, and isolated vertices.

OEOTL: Given that the vertex 00…0 has an outgoing edge and no incoming edge, find the other end of this line.

Upper Bounds in Quantum Complexity

Using the path integral we can get tighter upper bounds. Next we will show $\mathrm{BQP} \subseteq \mathrm{PP}$.

Recall that PP is the class probabilistic polynomial time. It removes the restriction of bounded-error from BPP. As a result, the completeness and soundness for PP are arbitrarily close to ½. Therefore PP is the class of problems you can solve efficiently with probability better than random guessing.

Definition (Probabilistic Polynomial-time). $L \in \mathrm{PP}$ if there exists polynomials p and q, and a verifier such that $\{V_r\}_{r \in \mathbb{N}}$

$$x \in L \implies \left(2^{-q(|x|)} \left|\left\{y : |y| = q(|x|) \wedge V_{p(|x|)}(x, y) = 1\right\}\right|\right) > 1/2$$

$$x \notin L \implies \left(2^{-q(|x|)} \left|\left\{y : |y| = q(|x|) \wedge V_{p(|x|)}(x, y) = 0\right\}\right|\right) \leq 1/2$$

Can we see that $\mathrm{PP} \subseteq \mathrm{PSPACE}$? But how to imagine putting EotL in PP?

Upper Bounds in Quantum Complexity

How do we use the circuit path integral to show that $\mathrm{BQP} \subseteq \mathrm{PP}$?

Hint: recall that {Hadamard, Toffoli} is a universal set of quantum gates in which every amplitude is real in the computational basis.

To simulate a BQP circuit, we want to decide whether the circuit outputs 1 at least 2/3s of the time, or no more than 1/3 of the time, promised that one of these is the case.

The standard tactic for a PP machine is to put in the tiniest possible amount of effort, and then try to guess the answer. Since it only has to do the tiniest bit better than random guessing.

Therefore the PP machine will select a single random path that contributes to the path integral. If that path contributes a positive amplitude then it accepts, and if it contributes a negative amplitude it rejects.

Upper Bounds in Quantum Complexity

We've established $\mathrm{BQP} \subseteq \mathrm{PP}$. What about $\mathrm{QMA} \subseteq \mathrm{PP}$?

We could imagine looping through some discretized description of the set of all possible quantum witness states. Such a proof could work, but needs to be explicit about how to search over witness states.

Another way to show $\mathrm{QMA} \subseteq \mathrm{PP}$ would be to show that we can solve the local Hamiltonian problem in PP. This is the route we will take, using a Feynman path integral.

Given a local Hamiltonian $H = H_1 + H_2 + \ldots + H_m$ and two real numbers (a,b) with $b - a \geq 1/\mathrm{poly}(n)$ we seek to decide between two cases:

YES: $E_0(H) \leq a$

NO: $E_0(H) \geq b$

Upper Bounds in Quantum Complexity

To distinguish between these cases, define $G = I - \epsilon H$, where $\epsilon \leq \|H\|^{-1}$. The eigenvalues of G are contained in the interval [0,1], and the largest eigenvalue is $\mu = 1 - \epsilon E_0(H)$.

We will consider a "partition function" $Z = \mathrm{tr}\left(G^L\right)$.

In a YES instance we have $\mu \geq \mu^+$, and in a NO instance $\mu \leq \mu_-$, with $\mu_+ - \mu_- > 1/\mathrm{poly}(n)$

Can we upper bound Z for NO instances in terms of μ_- , and lower bound Z for YES instances with μ_+ ?

YES: $Z \geq \mu_+^L$

NO: $Z \leq 2^n \mu_-^L$

$$\Longrightarrow \qquad \frac{Z_{\mathrm{yes}}}{Z_{\mathrm{no}}} \geq 2^{\frac{L}{\mathrm{poly}(n)} - n}$$

So L = poly(n) suffices to make the ratio of partition functions between the two cases exponentially large.

Upper Bounds in Quantum Complexity

Expanding Z as a path integral:

$$Z = \mathrm{tr}\left(G^L\right) = \sum_{z_1 \in \{0,1\}^n} \langle z_1 | G^L | z_1 \rangle = \sum_{z_1,\ldots,z_L \in \{0,1\}^n} \langle z_1|G|z_2\rangle\langle z_2|G|z_3\rangle \ldots \langle z_{L-1}|G|z_L\rangle\langle z_L|G|z_1\rangle$$

More compactly: $Z = \mathrm{tr}\left(G^L\right) = \sum_{z_1,\ldots,z_L \in \{0,1\}^n} \prod_{i=1}^{L} \langle z_i|G|z_{i+1}\rangle$

We can compute the amplitudes on at a time using polynomial space. We can put all the positive amplitudes on one side of the ledger, and the negative amplitudes on the other side.

Equivalently, since the ratio $Z_{\mathrm{yes}}/Z_{\mathrm{no}}$ is exponentially large, we can decide between the two cases with probability better than random guessing, just by looking at the amplitude of a single random path.

Upper Bounds in Quantum Complexity

From the perspective of the Feynman path integral, the power of quantum computing (BQP, or even QMA) comes from the ability to approximate exponentially large alternating sums.

These path integrals add up exp many amplitudes, which may be positive or negative and thus undergo many cancellations, and the result depends on what is left over. This is **interference**.

It's a bit like the year 2000 US presidential election: +50 million votes for Bush, +50 million votes for Gore, and Bush won because he led by 537 votes in the Florida recount.

Aaronson: "as far as I'm concerned, Feynman got the Nobel Prize for showing $\mathrm{BQP} \subseteq PP$." …

The close ties between quantum complexity and alternating sums raises a question. What is the complexity of quantum systems described by path integrals with all positive amplitudes?

Upper Bounds in Quantum Complexity

Returning to the "partition function": $Z = \mathrm{tr}\left(G^L\right) = \sum_{z_1,\ldots,z_L \in \{0,1\}^n} \prod_{i=1}^{L} \langle z_i|G|z_{i+1}\rangle$

We could put aside H for a moment, and rephrase the local Hamiltonian problem as "deciding the largest eigenvalue of a Hermitian matrix."

What happens to the complexity of estimating this sum if all the terms have a nonnegative amplitudes?

$$\prod_{i=1}^{L} \langle z_i|G|z_{i+1}\rangle \geq 0 \quad \forall z_1, \ldots, z_L$$

A sufficient condition for the amplitudes to all be nonnegative would be for G to be a matrix with nonnegative entries (in the basis of the $|z_i\rangle$).

In what follows we will (1) motivate the case of nonnegative G in terms of the Hamiltonian, and (2) establish a protocol with a classical prover and classical verifier for solving this restricted version of LH.

Upper Bounds in Quantum Complexity

An example of a Hamiltonian giving rise to a nonnegative matrix G, consider the Ising model:

$$H = -\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} Z_i Z_{i+1}$$

(one may specify *this* Ising model as the “ferromagnetic transverse Ising spin chain”)

We want to understand H as a matrix in the computational basis, and show that it can shifted and rescaled into a nonnegative matrix G.

The ZZ terms are on the diagonal, so we can control the sign of those (make them all positive or negative) by shifting the energy by a constant. $H \mapsto H - E_0 I$

What about the off-diagonal matrix elements of H in the computational basis?

Upper Bounds in Quantum Complexity

The off-diagonal elements of $\sum_i X_i$ are (real and) nonnegative, so the off-diagonal elements of H are real and non-positive.

Therefore, after shifting the ground energy to zero, $G = I - H/\|H\|$ is a nonnegative matrix.

Any H with all real and non-positive matrix elements in a basis B can be rescaled and shifted into a nonnegative matrix in that basis. Physicist say Hamiltonians with this property "do not have a sign problem", and in quantum information we call these Hamiltonians **stoquastic**.

"Stoquastic" = "Quantum" + "Stochastic." These Hamiltonians have some properties in common with stochastic matrices. The key point is that their ground state path integrals are nonnegative sums, instead of alternating sums, and this will limit the complexity of the stoquastic local Hamiltonian problem.

Upper Bounds in Quantum Complexity

Before using path integral to solve the stoquastic local Hamiltonian problem using a classical prover and classical verifier, it's worth noting that stoquastic Hamiltonians are ubiquitous in nature.

Besides the ferromagnetic transverse Ising chain, all generalized transverse Ising models are stoquastic:

$$H = -\sum_{i=1}^{n} X_i - \sum_{i,j} \alpha_{ij} Z_i Z_j$$

The ferromagnetic Heisenberg model is also stoquastic. $H = -\sum_{i\ j} \vec{S}_i \cdot \vec{S}_j$

In physics, any H = T + U (kinetic + potential) for distinguishable particles (or bosons) is stoquastic e.g. particles hopping on a graph under the influence of a potential:

$$H = -\sum_{(u,v)\in E} (|u\rangle\langle v| + |v\rangle\langle u|) + \sum_{v\in V} U_v |v\rangle\langle v|$$

Upper Bounds in Quantum Complexity

In contrast with the general local Hamiltonian problem which is QMA-complete, the stoquastic LH problem is in the classical class AM. This is proven in "The complexity of stoquastic local Hamiltonian problems". Bravyi, Divincenzo, Oliviera, Terhal, 2006.

We have already defined the class MA: Merlin sends a poly sized classical witness to Arthur, who verifies it using a BPP machine. AM is similar, but now Arthur goes first: he can flip some coins and ask some questions before Merlin sends the witness.

The Arthur-Merlin protocol for deciding stoquastic LH is based on approximating the partition function:

$$Z = \mathrm{tr}\left(G^L\right) = \sum_{z_1,\ldots,z_L \in \{0,1\}^n} \prod_{i=1}^{L} \langle z_i | G | z_{i+1} \rangle$$

The proof illustrates a general theme in approximation algorithms called "the connection between sampling and counting."

Upper Bounds in Quantum Complexity

The first step is to reduce our sum over nonnegative weights to an unweighted counting problem.

$$Z = \mathrm{tr}\left(G^L\right) = \sum_{z_1,\ldots,z_L \in \{0,1\}^n} \prod_{i=1}^{L} \langle z_i | G | z_{i+1} \rangle$$

Since the matrix elements of G are specified by m = poly(n) many bits, we can decompose G as an average:

$$G = \frac{1}{2^m} \sum_{t \in \{0,1\}^m} G(t)$$

where each G(t) is a binary matrix (a matrix with entries 0 and 1). This decomposition is highly nonunique, there are many explicit choices one could make (it's also inessential, but used in Bravyi et al.'s proof).

Upper Bounds in Quantum Complexity

Now the trace is re-expressed as:

$$\mathrm{tr}\left(G^L\right) = \frac{1}{2^{mL}} \sum_{t_1,\ldots,t_L \in \{0,1\}^m} \mathrm{tr}\left(G(t_1)\ldots G(t_L)\right) = \frac{1}{2^{mL}} \sum_{s \in \{0,1\}^{mL}} F(s)$$

Where $s = (t_1, ..., t_L, z_1, ..., z_L)$ is a binary string of length (n + m)L, and F is a Boolean function:

$$F(s) = \langle z_1|G(t_1)|z_2\rangle\langle z_2|G(t_2)|z_3\rangle \ldots \langle z_L|G(t_L)|z_1\rangle \in \{0,1\}$$

Therefore the original problem is reduced to counting the cardinality of the support of F,

$$Z = \mathrm{tr}\left(G^L\right) = \frac{1}{2^{mL}} \sum_{s \in \{0,1\}^{mL}} F(s) = \mathrm{supp}(F) = 2^{-mL}|\Omega| \quad , \quad \Omega = \{x \in \{0,1\}^{(n+m)L} : F(x) \neq 0\}$$

Upper Bounds in Quantum Complexity

Therefore the problem of deciding whether the ground state energy is large or small is now reduced to deciding whether Ω is large of small. As Bravyi et al. phrase it:

$$\begin{aligned} \mu(G) \geq \mu_+ \quad &\Longrightarrow \quad |\Omega| \geq \text{LARGE} \\ \mu(G) \leq \mu_- \quad &\Longrightarrow \quad |\Omega| < \text{SMALL}, \end{aligned}$$

where

$$\text{LARGE} = 2^{L(p_2(n)+\log \mu_+)} \quad \text{and} \quad \text{SMALL} = 2^{L(p_2(n)+\log \mu_- + \frac{n}{L})},$$

such that

$$\text{LARGE} = 2^n \cdot \text{SMALL} \quad \text{if} \quad L = 2np_1(n).$$

Thus it suffices for Merlin to prove a lower bound $|\Omega| \geq \text{LARGE}$.

Immediately after this line, the proof is concluded with the following argument.

Upper Bounds in Quantum Complexity

We can now invoke the Goldwasser and Sipser approximate counting protocol [20] based on Carter-Wegman universal hashing functions [21]. Recall that Ω is a set of k-bit strings, where $k = L(n + p_2(n))$. The main idea of [20] is that Arthur can compress k-bit strings to shorter b-bit strings using randomly chosen linear hash functions. One can choose parameters of the hashing such that the image $h(\Omega) \subseteq \Sigma^b$ is sufficiently dense (for positive instances). Arthur estimates the volume of $h(\Omega)$ using the standard Monte-Carlo method: he generates a large list of random b-bit strings and estimates the fraction of strings that belong to $h(\Omega)$. At this stage he needs Merlin's help, since a membership in the set $h(\Omega)$ is no longer efficiently verifiable because each string in Σ^b may have exponentially large number of pre-images. On the other hand, Merlin can prove a membership in the set $h(\Omega)$ by sending Arthur any of pre-images. In Appendix A we give some details of the parameters of the hash functions.