

Quantum Computation, NP-Completeness and physical reality [1] [2] [3]

Compiled by Saman Zarandioon
samanz@rutgers.edu *

1 Introduction

The NP versus P question is one of the most fundamental questions in the computer science and mathematics that has challenged many scientists and has resisted to any absolute answer. There are some evidences that make most of scientists believe that it is not solvable by Turing equivalent machines. However, machines that are equivalent to Turing machine are machines that operate based on principles of classical physics. Even if one proves that $P \neq NP$ for Turing machine, we may still have hope to solve an NP-complete problem efficiently on a machine that is built based on a non-classical theories that describe the universe more precisely. This essay focuses on this question: "can NP-complete problems be solved in polynomial time using the resources of the physical universe?" Studying this question will help us to:

- maybe find a way to solve our NP-complete problems efficiently.
- gain insight on NP-complete problems. (eg. studying quantum computing helped us to develop some tools and solve some open problems in classical computational complexity.)
- better understand physics and universe.

*This essay is mainly based on "NP-complete Problems and Physical Reality, Scott Aaronson", "Computational Complexity: A Modern Approach, S. Arora and B. Barak", and Vazirani's lecture notes; some of sentences are directly extracted from these resources.

One may ask how this question can help us to understand physics and universe. In physics some principles such as impossibility of superluminal signaling, and Second Law of Thermodynamics serve as a constraint in search for new physical theories. In computer science, especially in cryptography, assumption of $P \neq NP$ has been base for many interesting theatrical results. Therefore, maybe physicist can also use this assumption to evaluate new theories and reduce the search space for new theories use it to understand what constraints in universe leads to $P \neq NP$.

In this short essay we review some models based on classical physics and also more recent theories and see if they can help us to solve NP-complete problems and what the obstacles are.

2 Soap Bubbles

Recently Bringsjord and Taylor [4] posted a paper entitled "P=NP". They argued that since (1) since finding a Steiner tree in NP-hard (2) soap bubbles find a Steiner tree in polynomial time (3) soap bubbles are classical objects, and (4) classical physics can be simulated by a Turing machine with polynomial slowdown, it follows that $P = NP$.

However, Scott Aaronson conducted an experiment on this and his results were against this claim. He observed that soap bubbles fail to form Steiner tree when number of pegs are large. Sometimes they got stuck in local minima and sometimes they formed a suboptimal solution and then slowly "relaxed" toward an optimal structure. Also he observed that results are highly non-deterministic.

Most of other similar suggestions are subject to the same pitfalls of local minima and potentially long relaxation times.

3 Protein Folding

It is also known that protein folding problem is also NP-complete but it seems that nature can solve this problem efficiently without falling into local minima. However, this observation cannot convince us that NP-complete problems can be solved efficiently in classical physics (and consequently Turing machines). The reason is that maybe nature is good only in solving

this problems for natural proteins (proteins that exist in the nature) ; in other words, only proteins that nature can efficiently fold can survive. So, there is no reason if you convert an instance of 3SAT problem to a protein folding problem then nature can solve it defiantly. Solution to these artificial proteins may fall into local minima or suffer from long relaxation time.

4 Quantum Computing

If we cannot solve NP-complete problems in classic physics that is equivalent to Turing machine maybe we can use phenomena in the nature that classical physic is not able to describe and escape from this limitation and solve our problem. Quantum computers proposed based on principles of quantum mechanics have some interesting features that have made some scientists believe that they can give us more power over classical model.

4.1 What is Quantum Computation?

Quantum computers are proposed based on the following three basic principles in quantum mechanics:

- The superposition principle: An n-bit quantum register that has $k = 2^n$ classical state, has a quantum state that is a linear combination (superposition) of all classical states with complex coefficients. Therefore the quantum state of this register can be represented by a k-dimensional complex vector. Using ket notation we can represent its states as:

$$|\Psi\rangle = \sum_{i=0}^{i=k-1} \alpha_i |i\rangle$$

Where α_i is the complex amplitude corresponding to classical state $|i\rangle$ which is a vector that only its i^{th} row is one and the rest is zero.

- The measurement principle: The quantum state of a quantum register is hidden to us the only thing we can do is measuring it and we will get classical state $|i\rangle$ with probability $|\alpha_i|^2$ (So, length of quantum state vector should be unit, Hilbert space)

The weird thing is that measurements following the first measurement will result the same value that was obtained in the first measurement. (This property forms the basis of quantum cryptography.)

- Unitary evolution: Every operation on the quantum state vector is a unitary operation. Intuitively, a unitary operator is a rotation or reflection of the Hilbert space.

4.2 Brute-force quantum algorithm for 3SAT

It is easy to prove that availability of constant or polynomial parallel units will not help us to use a brute-force algorithm and solve 3SAT efficiently. However, in quantum computation model there is a built-in exponential parallelism. Is there any way to exploit these parallel units and solve 3SAT problems? In other words, we know that a quantum computer keeps some information (coefficients) about all potential answers of a 3SAT instance; is there any way to manipulate the register so that the setting corresponding to the satisfying assignment will stand out?

In 1994 Bennett, Bernstein, Brassard, and Vazirani [5] showed that any quantum algorithm that searches an unordered database of N items (2^n assignments of 3SAT) for a single "marked" item (satisfying assignment) must query the database about \sqrt{N} times. It follows that any brute-force algorithm needs at least $2^{n/2}$ steps. Therefore, any potential efficient algorithm should exploit the structure of 3SAT formula.

Why we cannot exploit the inherent exponential parallelism in the quantum mechanics? All proofs for this lower-bound exploit linearity of quantum operators which is the crucial property of quantum mechanics. Intuitively, if we think of the components of a superposition as parallel universes, then linearity is what prevents the universe containing the marked item from simply telling all the other universes about it.

4.3 Grover's algorithm

In 1996, Grover [6] found a quantum algorithm with running time of $O(2^{n/2})$ for finding a solution to a 3SAT problem and this proves that $2^{n/2}$ is a tight lower-bound. Since $2^{n/2}$ is much smaller than 2^n this algorithm is much more efficient than classical brute-force algorithms.

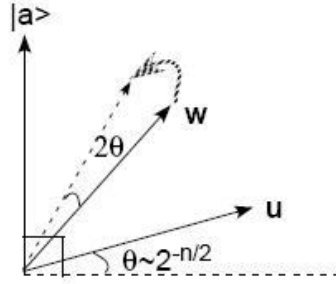


Figure 1: Get 2θ closer to $|a\rangle$ at each step

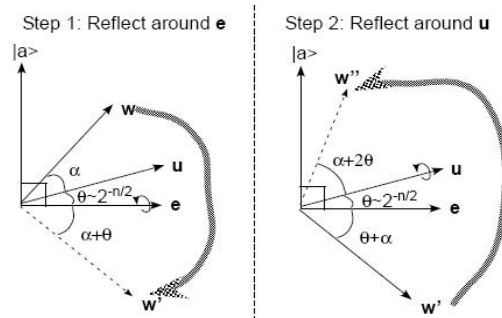


Figure 2: How to get 2θ closer to $|a\rangle$

Grover's algorithm is best described geometrically. Without loss of generality, assume function f has a single satisfying assignment \mathbf{a} . Consider an n -qbit register and let \mathbf{u} denote the uniform state vector of this register (i.e. $\mathbf{u} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$).

Using property of inner product, $\langle \mathbf{u}, |a\rangle \rangle = \frac{1}{\sqrt{2^n}} = \cos(\pi/2 - \theta)$, therefore $\sin\theta = \frac{1}{\sqrt{2^n}}$ and assuming n is sufficiently large, $\theta \geq \frac{1}{2 \cdot \sqrt{2^n}}$ (since for small θ , $\sin\theta \sim \theta$).

Start from state \mathbf{u} at each step transform it to a state that is 2θ closer to $|a\rangle$. So, in $O(1/\theta) = O(\sqrt{2^n})$ steps it gets close enough to vector \mathbf{a} so our measurement will yield \mathbf{a} with high probability.

Prove by picture: Figure 2 illustrates how by reflection around the vector \mathbf{a} and the vector $\mathbf{e} = \sum_{x \neq \mathbf{a}} |x\rangle$ (\mathbf{e} is the vector orthogonal to $|a\rangle$ in the plane

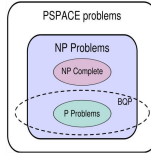


Figure 3: The suspected relationship of BQP to other problem spaces (Wikipedia)

spanned by \mathbf{u} and $|a\rangle$) we can get 2θ closer to $|a\rangle$.

4.4 Quantum computers and classical complexity classes

In 1997, P. Shor [7] introduced a polynomial time quantum algorithm for factoring problem. Still we do not know any deterministic or probabilistic efficient algorithm for factoring in classical model. Also there is an efficient quantum algorithm for Discrete Log problem. These results make scientist conjecture that class of problems that we can solve efficiently in quantum computers is different from those that we can efficiently solve in classical computers.

Lets call the class of problems that can be solved by applying polynomial number of quantum operators BQP (Bounded error, Quantum, Polynomial time). The following theorem shows what we know about power of BQP:

Theorem: $P \subseteq BPP \subseteq BQP \subseteq P^{\#P} \subseteq PSPACE$.

The fact that operators in quantum computers are linear makes them less powerful; otherwise, they could solve PSPACE problems in polynomial time!

5 Relativity, Analog and Anthropic Computing

There are some other thoughts and models for solving NP-complete problems that even though do not provide any practical solution, knowing them can give us some insight on this subject.

5.1 Relativity Computing

The idea is as follows: Run a program that tries to find a 3SAT problem in a brute force way and then board a spaceship and accelerate to a speed that is exponentially close to the speed of light. When you come back you can find the answer to your problem on the monitor.

But the problem with this solution is we need exponential energy and we are trading time with energy but energy is not less valuable than time.

5.2 Analog Computing

In 1979 Schonhag showed how to solve NP-complete and even PSPACE-complete problems in polynomial time, given the ability to compute $x + y$, $x - y$, xy , x/y , and $\lfloor x \rfloor$ in a single time step for any two real numbers x and $y \neq 0$. However, "foaminess" of space and time on the Planck scale rules out the possibility to build such an analog computer.

5.3 Anthropic Computing

These are models of computation in which one's own existence might depend on a computer's output; for example the following algorithm: "Given a formula ϕ guess a random assignment x , then kill yourself if x does not satisfy ϕ ". This way if such a question exists then you also have its answer; if your guess is incorrect you will not be there and consequently you will not have any question to answer! This is more like a philosophical approach that can help us to analyze these kinds of questions similar to the way that entropic principles in cosmology tries to answer difficult questions about the nature of scientific explanations.

References

- [1] NP-complete Problems and Physical Reality, Scott Aaronson
- [2] Computational Complexity: A Modern Approach, S. Arora and B. Barak, Book

- [3] Vazirani's lecture notes located at
<http://www.cs.berkeley.edu/~vazirani/quantum.html>
- [4] S. Bringsjord and J. Taylor. P=NP. 2004. cs. CC0406056.
- [5] C. Beckett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weakness of quantum computing. SIAM J. Comput.
- [6] L. K. Grover. A fast quantum mechanical algorithm for database search . In Proc. ACM STOC, pages 212-219, 1996
- [7] P. Shor . Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput., 1997.