The Subset Sum Problem

Reducing Time Complexity of NP-Completeness with Quantum Search

Bo Moon
bo.moon@ymail.com

University of South Florida

Manoug Manougian
Mathematics and Statistics

Jing Wang
Computer Science & Engineering

Abstract

The Subset Sum Problem is a member of the NP-complete class, so no known polynomial time algorithm exists for it. Although there are polynomial time approximations and heuristics, these are not always acceptable, yet exact-solution algorithms are unfeasible for large input. This paper discusses the physical and conceptual feasibility of quantum computation and demonstrates the utility of quantum search by analyzing the time complexities of the classical dynamic programming algorithm and Grover's algorithm in solving the Subset Sum Problem.

Problem Statement

The paper consists of analysis of how a quantum search algorithm can improve the efficiency of currently known solutions to a class of intractable problems in computer science.

The Subset Sum Problem:

Given a set S positive integers and a positive target integer T, determine whether there exists a subset of S whose elements sum to T.

Example:

Given the set $S = \{1, 2, 4, 8, 16\}$, is there a subset whose elements sum to 21?

Yes: $\{1, 4, 16\}$

It is easy for humans to "see" the solution just by looking at the set, but a computer is strictly procedural. As such, the most obvious algorithm (i.e. generate all possible subsets) is extremely slow, typical of the entire NP-complete class.

Mathematical Approach

Dynamic Programming:

DP is a technique for developing efficient algorithms for problems that exhibit two properties: overlapping subproblems (the problem can be defined recursively) and optimal substructure (solutions rely on solutions to smaller subproblems). Through recursion, the problem statement is simplified and can be solved simply by combining solutions to smaller versions of the same problem.

In short, the Subset Sum Problem exhibits both properties, and DP can be applied by identifying an underlying recursive nature. The subproblems consist of smaller sets and smaller target sums, so each recursive call will break the current subproblem into two cases: both will have one less element in its subset, but one case will have the same target sum as before while the other will have a smaller one. In this way, all possible combinations of target sums and subsets will be created, and the solutions to smaller cases can be put together to recreate the original problem statement.

Let $M(i, j)$ return a Boolean value representing if it is possible to reach the target sum i using a subset consisting of the first j elements in the set. The function call $M(T, n)$ then solves the original problem statement.

Time complexity: $O(n2^n)$

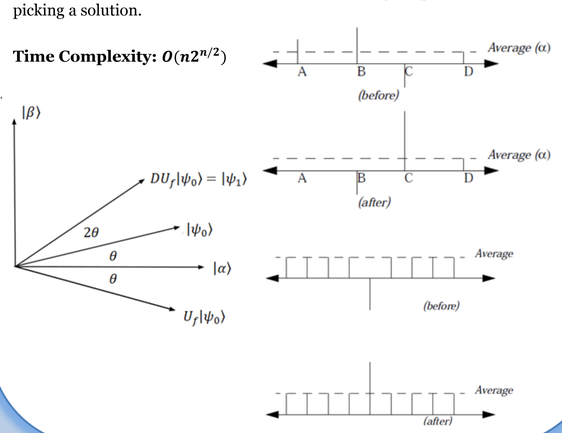
$$M(i,j) = \begin{cases} 1 & if\ i = 0 \\ 0 & if\ i < 0\ or\ j = 0 \\ M(i - x_j, j-1) \vee M(i, j-1) & otherwise \end{cases}$$

Grover's Algorithm:

As a quantum algorithm, its operations are represented by linear transformations acting upon a state vector in Hilbert space. Geometrically, it iteratively applies a product of two reflections across $|\alpha\rangle$ and then $|\psi_0\rangle$, resulting in a net rotation of 2θ towards $|\beta\rangle$. The goal of Grover's algorithm is to make the state vector $|\psi\rangle$ as close to $|\beta\rangle$ as possible in order to maximize the probability that the mechanical wave function of the quantum system collapses into a desirable state, namely a solution to the Subset Sum Problem.

Pictorially, imagine that all possible subsets of the original set are teeth on a comb, and we must pick a tooth at random with probability proportional to length. Grover's algorithm inverts the direction of the tooth that solves the Subset Sum Problem, and then applies an "inversion about the mean"—each short tooth is increased so that it is above the average length by the same amount it was previously below the average, and vice versa for long teeth. After repeating this several times, the length of the solution teeth are very long while the incorrect teeth are very short, thereby maximizing the probability of picking a solution.

Time Complexity: $O(n2^{n/2})$



Discussion

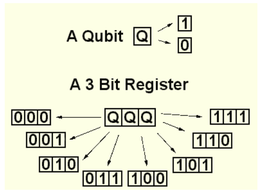
The NP-Complete Class:

Problems within the NP-complete class have no efficient solutions; that is, the only known algorithms involve testing every possible answer. Since NP-complete problems are so difficult, computer scientists must study them so that they know not to waste time searching for an efficient solution. Interestingly, it has been shown that any NP-complete problem can be rewritten as any other NP-complete problem, so a solution to one problem can resolve the entire class. As such, this paper focuses only on a specific problem because any results can be generalized for the entire NP-complete class.

Quantum Computation:

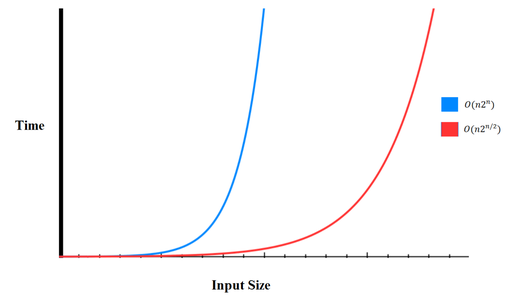
All quantum algorithms require a quantum computer, which is a computer that manipulates the states of quantum systems as part of computation. Information is represented by the discrete states of a quantum bit, or qubit, with a quantum system: for instance, using a hydrogen atom, a "1" corresponds to an electron in an excited state while "0" corresponds to an electron in the ground state. There are two main advantages to quantum computation:

- The superposition principle: A quantum system exists in a linear combination of all possible states. As such, a single qubit can represent two states at once whereas a normal bit on a computer can only be one, so quantum computers have exponentially more memory.
- Quantum parallelism: An operation on a quantum system is equal to applying the same operation to each of its components. For instance, if a quantum system has 10,000 possible states, then one operation on that system is equivalent to 10,000 operations on the individual states. This allows for an exponential number of operations to be carried out instantaneously.



The main problem with this is that measurement, or the act of observing, forces the quantum system to collapse from the superposition of an exponential number of states to a single state, so all other states cannot be retrieved. The goal of quantum algorithms is to apply linear transformations on the state vector of a quantum system to maximize the amplitude, and thus probability, of desirable states while minimizing amplitudes of undesirable ones so that the system produces useful information upon collapsing.

Comparison of Time Complexities



Conclusions

One of the most promising applications of quantum search is to resolve the exponential time complexity that is characteristic of NP-complete problems. Since Grover's algorithm provides a quadratic improvement in speed, one may conjecture an even faster quantum algorithm, perhaps a polynomial one, may exist. Unfortunately, it has been proven that Grover's algorithm is optimal, so the fastest time complexity of any search algorithm, classical or quantum, is $O(2^{n/2})$ oracle calls. Moreover, the NP-complete class is defined to be a set of search problems that are equivalent to each other, yet the fastest possible search algorithm is still exponential, so this severely diminishes hope that NP-complete problems can be efficiently solved. However, there exist several alternatives, though not promising, to consider.

It is widely believed that NP-complete problems are intractable because there is no exploitable organization in the underlying structure of the search space. In fact, it is precisely this property that makes certain classical deterministic algorithms so efficient. As such, it may be possible to expedite solutions to the NP-complete class by reexamining the search space in novel ways. For example, the dynamic programming algorithm for the Subset Sum Problem presented earlier runs in pseudopolynomial time by utilizing optimal substructure and overlapping subproblems. Though this approach was not sufficiently fast, it demonstrates that there could be some property of the Subset Sum Problem, and possibly the entire NP-complete class, not yet observed that can be used as the basis of an even faster algorithm.

Another possibility to consider is the application of reductions. Finding the prime factorization of an integer is a famously hard problem (though not NP-complete) that has no known polynomial time solution in the classical realm, but quantum computation offers one. Shor's quantum algorithm reduces this problem into one of identifying periodicity, which can be solved with some mathematical insights. Similarly, reductions may help bring known polynomial time solutions to the NP-complete problems. Of course, the largest and most overwhelming obstacle to this consideration by far is that for an NP-complete problem to reduce to P, it would require the proof of $P = NP$, a puzzle that has eluded generations of computer scientists. Still, much of computational complexity theory analyzes problems with respect to classical algorithms, so perhaps reductions of the NP-complete class may be possible with advances in quantum computation.

References

[1] R. E. Neapolitan and K. Naimipour, in Foundations of Algorithms: Using C++ Pseudocode, Sudbury, Jones and Bartlett Publ., 1998, p. 194.

[2] S. Dasgupta, C. H. Papadimitriou and U. Vazirani, in Algorithms, Boston, McGraw-Hill Higher Education, 2008, pp. 243, 297-298, 301, 307.

[3] D. N. Mermin, in Quantum Computer Science: An Introduction, Cambridge, Cambridge University Press, 2007, pp. 1, 17, 24, 37-38, 89-94.

[4] K. H. Rosen, in Discrete Mathematics and Its Applications, Boston, McGraw-Hill, 2003, p. 781.

[5] G. F. Viamontes, I. L. Markov and J. P. Hayes, "Is Quantum Search Practical?," Computing in Science and Engineering, vol. 7, no. 3, pp. 62-70, 2005.

[6] G. P. Berman, G. D. Doolen, R. Mainieri and V. I. Tsifrinovich, in Introduction to Quantum Computers, Singapore, World Scientific, 1999, pp. 38, 85-86.

[7] M. A. Nielson and I. L. Chuang, in Quantum Computation and Quantum Information, Cambridge, Cambridge University Press, 2000, pp. 22, 29, 62, 153, 249-253.

[8] O. Morsch, in Quantum Bits and Quantum Secrets: How Quantum Physics is Revolutionizing Codes and Computers, Weinheim, Wiley-VCH, 2008, pp. 96, 109.

[9] L. K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack," Physical Review Letters, vol. 79, no. 2, pp. 325-328, 1997.

[10] U. Vazirani, "Chem/CS/Phys191: Qubits, Quantum Mechanics, and Computers," [Online]. Available: http://www-inst.eecs.berkeley.edu/~cs191/sp12/. [Accessed 24 June 2012].

[11] C. Zalka, "Grover's quantum searching algorithm is optimal," Physical Review A, vol. 60, no. 4, pp. 2746-2751, 1999.