Quantum Algorithms: Promise and Perspective

Shelby Kimmel, Middlebury College

Introduction

Much of the excitement surrounding quantum computers is driven by the possibility of using quantum algorithms to solve problems that are intractable using standard, or "classical," computers. Several prominent examples of quantum speed-ups have been discovered, in particular with applications to cryptography and chemistry. Despite these exciting applications, quantum computers are not a magic bullet for computation, and there are also examples of problems where quantum algorithms do not give any advantage, or only give a small advantage. However, because quantum computers are so difficult to simulate and analyze, for many problems and algorithms, we simply do not know how quantum algorithms will perform. As larger quantum devices and computers come on-line in the coming years, we will be able to test quantum algorithms on these devices, and hopefully discover new problems where quantum computers give an advantage.

Examples of Quantum Algorithms

Perhaps the most famous quantum algorithm is for factoring [Shor, 1994]. There is a quantum factoring algorithm whose run-time scales roughly like the number of digits in the integer, whereas the best classical algorithm requires nearly exponential time. The assumption that factoring is difficult is critical for the functioning of our current cryptosystems – the systems that allow for much of e-commerce. In these cryptosystems, you encode information (like your credit card in order to purchase something online), and the only way we know that eavesdroppers can only decode the information is by factoring a very large number. As long as there are no fast

methods for factoring, this strategy is good for keeping information private. However, once large quantum computers are developed, they will easily break these codes.

A second well known application of quantum computers is modeling chemical and physical interactions. Quantum mechanics is often very difficult to simulate using a regular computer because the amount of space needed to keep track of the quantum system scales exponentially in the number of particles. This fact has made it challenging for scientists and engineers to understand the workings of many important systems and interactions, from high-TC superconductors to photosynthesis. However, a quantum computer by its very nature can simulate these many-body quantum systems [Buluta, and Nori 2009, Kassal et al 2011]. For example, one of the hopes of quantum computers is that they will improve and speed-up the development of drugs and industrial chemistry processes by allowing testing of chemical synthesis and performance in silico.

While quantum algorithms have impressive performance for problems like factoring and quantum simulation, there are also many problems where quantum computers provably do not have a large advantage over regular computers. The most famous such example is the problem PARITY, which involves determining whether a string of 0's and 1's has an even or odd number of 1's. The best possible quantum algorithm for PARITY has the same scaling as the best possible classical algorithm [Beals et al 2001].

Another example of a problem with only a modest quantum speed-up is SEARCH. While a classical computer can search through the elements of a list to find a certain item in time that scales like the number of items in the list, there is a quantum algorithm that can do this in time that scales at best like the square root of the number of elements [Grover 1996, Boyer et al

1998]. While this is certainly an improvement, it is not the kind of speed-up that would likely warrant the huge investment that will be required to make quantum computers a reality.

Why Quantum Computers Are Useful for Some Problems and Not Others

In order to get perspective on the potential of quantum algorithms, it is important to understand why quantum algorithms have such an advantage for some types of problems, and little to no advantage for other problems. There are three properties of quantum mechanics that are different from standard computation, and which are required for quantum speed-ups: superposition, interference, and entanglement. We will focus on the first two; while entanglement is necessary for a quantum speed-up [Josza and Linden 2003], the presence of superposition and interference often imply entanglement.

Superposition is a property of quantum systems that allows them to be in multiple states at the same time. The most famous example of this is Schrodinger's cat, in which a cat is put in a situation in which it is both alive and dead at the same time. While superposition sounds incredibly powerful, as it seems to imply unlimited parallel computation, there is a catch. While a quantum computer can be in a superposition of many states, when you try to get an answer from the computer, it system collapses to one of the states at random. Thus superposition on its own is essentially as powerful as probabilistic computation, where a state of the system is chosen at random.

In order for quantum algorithms to get an advantage out of superposition, superposition needs to be combined with interreference. Whereas in probabilistic computation, each state of the computer would be associated with a positive probability, in quantum computing, each state of the system can be associated with a complex "probability" or weight. The advantage of having

states with complex weighting is that when you combine a positive and a negative weight, you get cancellation. In probabilistic computation, where all of the weightings are positive, you can never get this cancellation.

Therefore, a successful quantum algorithm involves creating a large superposition of carefully weighted states, such that when the states interfere with each other, the states that don't correspond to solutions cancel out, and the states that give the correct solution are constructively reinforced. Certain problems have structure that allow this cancellation to happen quickly, while others require time to amass a proper distribution of weightings. The more structure a problem has, the more likely it is to admit a large speed-up. PARITY and SEARCH don't have a lot of structure, because for both of these problems, flipping a single bit of the input can change the value of the outcome. Problems like factoring or simulate chemistry are not so sensitive to small changes in the input and instead extract larger scale structures.

The Future of Quantum Algorithms

Quantum computers behave in ways that we don't know how to simulate easily or efficiently with classical computers. While this fact is why quantum computation is exciting, it also hampers the development of quantum algorithms. There are whole classes of quantum algorithms that seem promising, but because we can't simulate their behavior on large problems of interest, and we can't analyze them by hand, we don't know how they will perform in practice.

Quantum algorithm designers have developed a toolkit of quantum algorithmic paradigms. Only for some of these approaches, and for only certain problems, have we been able to analyze their performance. As we develop small scale quantum computers in the coming

years, we will have the exciting opportunity to test these algorithms, and search for new paradigms.

As we make the transition from a field that has been purely theoretical to a field that also is "computational" there is much work that needs to be done. Quantum programming languages have just begun to be developed, and for many of the existing algorithms in the toolkit, there is not an easy way to go from the theoretical description of the algorithm to a programming language description. Furthermore, even when an algorithm is described by a quantum programming language, there are still many challenges in converting those instructions to quantum machine code, as different physical quantum computers tend to have different sets of basic operations.

Conclusion

Quantum computers are not all-powerful – they do not provide significant speed-ups for all problems. However, for problems with appropriate structure, we can take advantage of quantum properties like superposition and interference to achieve significantly better performance than is possible with standard computers. While these applications are exciting, our understanding of quantum algorithms is truly quite limited; as physical quantum computers are developed and we can start testing quantum algorithms on these devices, we will have an unprecedented new tool for the development and exploration of quantum algorithms.

References

Beals, R., Buhrman, H., Cleve, R., Mosca, M. and De Wolf, R., 2001. Quantum lower bounds by polynomials. *Journal of the ACM (JACM), 48*(4), pp.778-797.

Boyer, M., Brassard, G., Høyer, P. and Tapp, A., 1998. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, *46*(4-5), pp.493-505.

Buluta, I. and Nori, F., 2009. Quantum simulators. *Science*, *326*(5949), pp.108-111.

Grover, L.K., 1996, July. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219). ACM.

Jozsa, R. and Linden, N., 2003, August. On the role of entanglement in quantum-computational speed-up. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (Vol. 459, No. 2036, pp. 2011-2032). The Royal Society.

Kassal, I., Whitfield, J.D., Perdomo-Ortiz, A., Yung, M.H. and Aspuru-Guzik, A., 2011. Simulating chemistry using quantum computers. *Annual review of physical chemistry*, *62*, pp.185-207.

Shor, P.W., 1994, November. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on* (pp. 124-134). Ieee.