I think there is a world market for maybe five *quantum* computers

Dominik Schultes

6. May 2004

In 1943, Thomas J. Watson, chairman of IBM, stated "I think there is a world market for maybe five computers." From todays point of view, we can boldly say that this statement could be assessed with a superlative of "wrong" if such a word existed. The main reason for the success story of the (conventional) computer is the fact that it is a general purpose machine. As it can solve problems from a wide range of applications, everyone needs one.

At the moment, the situation regarding quantum computers seems to be different. Despite of the fact that the idea of quantum computing was proposed already over 20 years ago by R. Feynman [Fey82], till this day, only a minor amount of applications is covered by quantum algorithms so that we cannot speak of a general purpose machine. Let us have a closer look at some of the most popular problems that should demonstrate the potential strength of the quantum computer.

1. *Deutsch's problem*: We are given a black box that represents a boolean function $f : \{0, 1\} \to \{0, 1\}$ and we want to determine the value of $f(0)$ XOR $f(1)$, but we are allowed to use the black box only once. It can be shown that it is possible to solve this problems with the given constraint on a quantum computer due to the fact that we are able to pack more than one request into a single request [Cal04]. Of course, this is an impressive example for the capability of a quantum computer because it can do something quite easily that a conventional computer cannot do. However, we have to pay attention to the fact that a conventional computer actually *can* decide whether $f(0)$ XOR $f(1)$ is true or false: it just needs two requests instead of one. And in practice, there are probably only a few situations where the artificial constraint of Deutsch's problem appears.

2. *Shor's algorithm to factorize numbers*: In 1994, P. Shor published a quantum algorithm that is able to factorize a number in polynomial time [Sho94]. As the ability of factoring a number efficiently leads to the ability of cracking common cryptography systems like RSA, Shor's algorithm has been the "running threat" of a quantum computer, which heavily contributes to the general interest in quantum computing. As a matter of fact, we cannot see this threat. Most people that mention Shor's algorithms "forget" to add that quantum mechanics also make completely new cryptography systems possible that are really secure and cannot be cracked – not even by a quantum computer. The first method of this kind was published by C. H. Bennet and G. Brassard in 1984 and is often called the BB84-Protocol [BB84]. It uses photons of different polarizations in order to transmit a shared key. The crucial fact is that it is not possible to measure the photons, i.e., to get the desired information, without perturbing them. Hence, the receiver can check if a third person has tried to spy on the transmission. If the receiver realizes that the photons have been perturbed, the key is discarded and a new one is requested; otherwise, the receiver can be sure that no third person knows the key.

   It is important to realize that it is not necessary to build a fully working quantum computer in order to realize quantum cryptography. In comparison with the complexity of a quantum computer, such a cryptography system could be implemented quite easily. This is underlined by the fact that in the last years many practical experiments were already successful. For example, in an experiment in April 2002, the BB84-Protocol was used to transmit a key from the Zugspitze to the Westliche Karwendelspitze[1] over a distance of 23,4 km [Hal02].

   Hence, we can conclude that if it was possible to built a quantum computer (that is able to crack RSA), it would be possible to install a new cryptography system much earlier so that the quantum computer would become obsolete before it would start working.

[1] Both are mountains in the European Alps.

3. *Grover's algorithm to search an item in an unsorted database*: In 1996, Grover presented a fast quantum mechanical algorithm for database search [Gro96]. The main advantage of this algorithm is the fact that it beats the lower bound of $O(n)$ for the search in an unsorted database. According to Grover, a search can be performed in $O(\sqrt{n})$, which is a significant improvement. But, who wants to search in an unsorted database in $O(\sqrt{n})$ instead of sorting the database once and then searching in $O(\log n)$, which is *much* better than $O(\sqrt{n})$? Either you have a small database and can afford linear search or you have a big database and you even cannot afford $O(\sqrt{n})$. Probably, there are only a few databases where, on the one hand, a linear search in $O(n)$ takes too much time and, on the other hand, you do not want to sort it so that, in this case, you could profit from Grover's algorithm.

4. *Simulation of quantum systems*: The simulation of a quantum system could obviously be a natural application of a quantum computer, which could be of great value for some physicists. But this possible application certainly does not contradict our claim that the quantum computer seems not to be a general purpose machine.

After looking at the problems that a quantum computer could solve, let us deal with this issue from a different point of view and let us ask which types of problems exist at all. On the one hand, there are the – roughly speaking – *easy* problems, i.e., the problems that can be solved at a conventional computer in polynomial time. A good example for this category is searching an item in an array. Generally, we doubt that it is worth it to develop a quantum computer in order to solve such "easy" problems because the potential of improvement is limited. If you can already solve a problem sufficiently fast, why should you look for a faster method ?

On the other hand, there are the *really difficult* problems that are not computable at a Turing machine. First of all, these problems are of great theoretical interest. Furthermore, many interesting applications (e.g. program verification) would arise if such problems could be solved on another machine. However, the chances to master such a problem in practice can be considered to be quite low.

Actually, the interesting things are situated between the problems that are either too easy or too difficult. Particularly, a fast and reliable solution of NP-complete problems ranks top on a computer scientist's list of wishes because of the following reasons: Firstly, the present-day solutions on conventional computers are often not satisfactory as they are too slow or too inaccurate. Secondly, there are many real world applications that can be reduced to a NP-complete problem. Thirdly, if you can solve one NP-complete problem efficiently, you can solve all of them efficiently; hence, there is a possibility to kill many birds with one stone. Since NP-complete problems are not even proven to be unsolvable on a conventional computer using only polynomial time and since we can therefore not even be sure that these problems actually are difficult, it can be assumed that there are significant chances to find a fast solution – maybe using a different model of computation. *Unfortunately, there is no quantum algorithm so far that solves a known NP-complete problem in polynomial time.* Due to Grover's algorithm we can obtain a speed-up because the brute force algorithms that perform an exhaustive search on NP-complete problems can be accelerated. However, the running time is only improved from $\Theta(2^n)$ to $\Theta(\sqrt{2^n}) = \Theta(2^{n/2})$. Thus, it is still exponentially. Shor's algorithm solves the problem of factorization in polynomial time, but it is not known if this problem is NP-complete. If it was proven that it is NP-complete, the power of quantum computers would immediately increase in an explosive way.

We can deduce therefore that the most promising way to disprove the title of this essay would be the concentration of most of the efforts on finding a quantum algorithm that solves a NP-complete problem in polynomial time.

We want to conclude by some final remarks:

- Of course, finding of a quantum algorithm that solves a NP-complete problem efficiently has nothing to do with the implementation of a quantum computer, in other words, even if our main claim is proven to be wrong and there actually is a world market for quantum computers, then it is still possible that this world market cannot be saturated.

- The considerations about the world market for quantum computers have nothing to do with the fact that the research on bounds of computability is of great interest – at least from a theoretical point of view – and should not be totally neglected.

- Furthermore, it can be very useful – from the physicist's point of view – to build a quantum computer even if there is no demand because the developed technology could turn out to be of great theoretical or practical use, maybe in unexpected fields.

References

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.

[Cal04] C. S. Calude. Unconventional models of computation. lecture notes, University of Auckland, 2004. http://www.cs.auckland.ac.nz/∼cristian/umc/umc.html.

[Fey82] R. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467, 1982.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, page 212, May 1996.

[Hal02] Matthäus Halder. Quantenkryptographie - Ein Freiraumexperiment zur Schlüsselübertragung über 23.4 km. Ludwig-Maximilian-Universität München, Germany, 2002. http://scotty.quantum.physik.uni-muenchen.de/publ/.

[Sho94] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society Press, 1994.