VII Международная научная конференция
«Математическое и компьютерное моделирование»
Омск, ОмГУ, 22 ноября 2019 г.

УДК

R. Alvarez, O. Galindo, and V. Kreinovich
University of Texas at El Paso,
El Paso, Texas, USA
ralvarezlo@miners.utep.edu,
ogalindomo@miners.utep.edu, vladik@utep.edu

WHY QUANTUM ALGORITHMS ONLY USE REAL-VALUED AMPLITUDES: A POSSIBLE EXPLANATION

Quantum computing: a brief reminder. It is known that if we use quantum processes for computing, we often get much better results that with the best possible non-quantum algorithms; see, e.g., [1].

For example, in the non-quantum case, to find an element with given properties in an unsorted array of n elements, we need to look into every element – if we miss one of them, we may miss the desired element. This, in the non-quantum case, we need at least n computational steps to solve this problem. In contrast, in the quantum case, we can find this element in time proportional to the square root of n (*Grover's algorithm*). This possibility comes from the fact that in quantum physics, for every states s_1, …, s_n, we can also consider their *superposition*, i.e., a state $a_1|s_1> + \ldots + a_n|s_n>$, where a_i are complex numbers (called *amplitudes*) for which the sum of squares of their absolute values is equal to 1. Because of this opportunity, in addition to asking the computer to check a single element of the array (as in the non-quantum case), we can also request a superposition of such requests. Such a request targets several elements of the array at the same time; this allows us to get the result faster than in the non-quantum case.

In general, in quantum computing, the analogue of a bit – which can be in two states 0 and 1 – is a quantum bit (qubit), a superposition of these two states: $a_0|0> + a_1|1>$.

There are other efficient quantum algorithms. The most well-known is probably *Shor's algorithm* for factoring large integers. At first glance, this may sound like an academic problem, but it is very practical: most cryptographic schemes used in communications and commerce are based on the fact that with non-quantum computers, this task is very difficult. As a result, an agent interested in receiving a secure message comes up with two large prime numbers p and q, and makes their product n = p * q known as the *public code*. Anyone can use this code to encrypt their message and send it via open channel, but to

decode it we need to know the factors p and q. If quantum computing becomes possible, we will be able to read all the encoded messages sent so far. This does not mean, of course, that there will be no more privacy or security: there is also a quantum encoding algorithm that – in contrast to the current algorithms – provides absolute security.

Interesting feature of current quantum algorithms. An interesting feature of current quantum algorithms is that while, in general, we can have arbitrary complex values of the amplitudes, all existing quantum algorithms use only real values. In this abstract, we provide a possible explanation for this phenomenon.

Our explanation. In non-quantum computing, to store the information about a single bit, we need exactly 1 bit of information. To store the information about the state of a real-valued quantum qubit $a_0|0> + a_1|1>$, we need to store a real number a_0, plus the sign of a_1 (the absolute value of a_1 is uniquely determined from the condition that the sums of squares of absolute values is 1). In modern computers, storing a real number requires 64 bits, while storing a sign requires 1 extra bit – to the total of 65 bits.

If we want to use general complex-valued amplitudes, then we need to store three real numbers: the real and imaginary parts of a_0 and the phase of a_1 (similarly to the real-valued case, the absolute value of a_1 is uniquely determined by the absolute value of a_0.) Thus, overall, we need $3 \times 64 = 192$ bits.

So, when we from non-quantum computing to quantum computing with real-valued amplitudes, we got from 1 bit to 65 bits and thus, increase our possibilities 65 times. On the other hand, when we go from quantum computing with real-valued amplitudes to general quantum computing, with complex-valued amplitudes, we only increase number of bits – and thus, our possibilities -- by a factor of 3. To cover a 65 times increase, we need at least 4 such smaller increases (indeed, with 3 smaller increases, we would only gain a $3^3 = 27$ times increase). Overall, the transition from non-quantum computing to a general quantum computing can be described as 5 such factor-of-3 increases, of which 4 falls into the transition to real-valued quantum computing and only one corresponds to the final transition to general complex-valued quantum computing.

Crudely speaking, this means that the probability that an algorithm falls into the first (major) transition is about 4/5, which is much larger than 0.5. The probability that two algorithms both falls into this category is equal to $(4/5)^2 = 0.64$, which is still larger than 0.5. So, it is not surprising that both major quantum algorithms known now – Grover's algorithm and Shor's algorithm – fall into this major transition category, i.e., use only real-valued amplitudes.

References

1. *Nielsen M.A. Chuang I.L.*, Quantum Computation and Quantum Information.-- Cambridge, U.K.: Cambridge University Press, 2000.