The Effective Solving of the Tasks from NP by a Quantum Computer

Sergey Sysoev
Dep. of Mathematics and Mechanics
St.Petersburg State University
St.Petersburg, Russia
Email: sysoev@petroms.ru

Abstract—The new model of quantum computation is proposed, for which an effective algorithm of solving any task in NP is described. The work is based and inspired by the Grover’s algorithm for solving NP-tasks with quadratic speedup compared to the classical computation model. The provided model and algorithm exhibit the exponential speedup over that described by Grover.

Index Terms—Quantum Computing, Class NP, Quantum Algorithm, Grover’s Algorithm

I. INTRODUCTION

The mathematical model of quantum computation has appeared in the 1980-th with the works of David Deutsch [1] and others [2-4]. In 1994 Peter Shor [5] proposed an effective algorithm for solving an NPI-candidate task – factorisation of big composite numbers. This breakthrough allowed to expect, that any task from NP could have an effective solution on a quantum computer. This hope was weakened after the Lov Grover’s work [6] in 1996, which proposed a general quantum computation algorithm for solving any task from NP. The algorithm was designed to search a particular point in the unsorted database of size N with only √N oracle queries. Grover has shown that under the considered assumptions the algorithm is optimal, so only quadratic speedup can be achieved by the quantum computer compared to the classical one, if we don’t have additional information about the oracle functioning.

This paper describes the improvement of the quantum database search algorithm, exhibiting the exponential speedup over the classical search, by changing some preliminary assumptions about the model of computation.

II. THE GROVER’S ALGORITHM

First, let us recall the initial algorithm proposed by Lov Grover. Let’s consider the function f:

\[ f : \{0, 1\}^n \rightarrow 0, 1 \]

\[ \exists \omega : f(x) = 1 \iff x = \omega, \]

which is some decision function over the set of n-bit strings. \( N = 2^n \) – number of all possible n-bit strings, thus the number of all possible inputs to f. Function f is implemented as a black box, and our purpose is to find n-bit string \( \omega \) on which f returns 1. Grover defines the quantum oracle \( U_\omega \):

\[ U_\omega : |x\rangle |y\rangle \rightarrow |x\rangle |y \oplus f(x)\rangle, \]

where x is n-bit string, and y is one bit. On the first n qubits \( U_\omega \) acts as follows:

\[ U_\omega = 1 - 2 |\omega\rangle \langle \omega|, \]

\( U_\omega \) acts identically on any vector orthogonal to \( |\omega\rangle \), while changes the sign of \( |\omega\rangle \) itself. This can be treated like reflection of any vector over the hyperplane orthogonal to \( |\omega\rangle \).

Then the vector \( |s\rangle \) and operator \( U_s \) are introduced as follows:

\[ s = H|0\rangle^n, \]

\[ U_s = 2 |s\rangle \langle s| - 1, \]

where H is n-qubit Hadamard transform. Operator \( U_s \) reflects any vector over the vector \( |s\rangle \). The Grover Iteration is

\[ U_{\text{grov}} = U_s U_\omega. \]

The \( U_{\text{grov}} \) operator rotates the initial vector \( |s\rangle \) towards the desired vector \( |\omega\rangle \) by the angle \( 2\theta \), where \( \sin \theta = \frac{1}{\sqrt{N}} \). The action of the Grover iteration can be seen on the figures 1 and 2.
III. THE ALGORITHM

Following Grover, we will denote:
\[
f : \{0, 1\}^n \rightarrow 0, 1,
\]
\[
U_\omega : |x⟩|y⟩ \rightarrow |x⟩|y \oplus f(x)⟩,
\]
\[
|s⟩ = H |0⟩^n.
\]

In Grover’s work it was supposed that there is the only input \(ω\), for which \(f\) returns 1. Here we will consider the task of determining if there exist such a point (points) for which \(f\) returns 1, assuming that the number of such points is relatively small or zero.

Our task is to determine if there exist such input \(x\): \(f(x) = 1\).

A. Step1. Initial superposition

At first we prepare the superposition vector \(|s⟩\). Then we use oracle:
\[
|s⟩_1 = U_\omega |s⟩
\]
Let’s denote \(|x⟩_0\) – the last qubit in \(|s⟩_1\).
\[
|x⟩_0 = \alpha |1⟩ + \beta |0⟩.
\]
If \(\forall x f(x) = 0\), then \(\alpha = 0\), otherwise \(\alpha > 0\). All further operations are aimed to determine, if \(\alpha = 0\) or not.

B. Step 2. The iteration

Let’s introduce the operator \(X_i\):
\[
X_i = 1 - 2 |x⟩_i ⟨x|,
\]
where \(|x⟩_i\) is the the current system state. We consider the iteration:
\[
|x⟩_{i+1} = X_i |0⟩.
\]

Note, that \(X_i\) is operator derived from the current state of the system, and it acts on the zero-initialized qubit of a different system. The result of such action delivers us the next operator \(X_{i+1}\), which again will act on a new quantum system of one qubit initialized by \(|0⟩\). Thus, on each step we need a new quantum system of one qubit. In fact, we can reuse these systems, so only two of them are necessary. If \(\alpha = 0\), then \(\forall i X_i = 1\), and the measurement of the last qubit will give us 0 independently of the number of iterations. But, if \(\alpha > 0\), then the last qubit of \(|x⟩_0\) has non-zero angle \(θ_0\) with \(|0⟩\).

\(θ_1\) - the angle of \(|x⟩_1\) with \(|0⟩\) equals \(2θ_0\), since \(X_0\) reflects the vector \(|0⟩\) over \(|x⟩_0\). Each iteration doubles the previous state angle with \(|0⟩\), increasing the probability of getting 1 after the state measurement (see figure 3).

If the number of solutions to \(f(x) = 1\) is 1, then \(θ_0 = \frac{1}{\sqrt{N}}\), and we will need
\[
T_1 = \log \sqrt{N}
\]
iterations to get 1 after the measurement with the probability close to 1. With two solutions we need only \(T_2 = T_1 - 1\) iterations for the same measurement probability. And for \(k\) solutions we need
\[
T_k = T_1 - \log k,
\]
iterations.

IV. THE ITERATION IMPLEMENTATION

The implementing of \(X_i\) operator, which depends on some quantum state \(x_i\) and acts on another quantum system is an open question in the scope of this work. We can imagine some setting where the operator system is implemented as a state of a photon, while the operated system is an atom, excited by that photon and vice versa. But the actual implementation is yet to be done, however there are no obvious obstacles we could found on this matter.
V. Conclusion

The proposed quantum algorithm allows us to determine, if there exist the specified number of solutions to \( f(x) = 1 \). The number of iterations \( T \) of the algorithm is always less then \( \lg \sqrt{N} \). To check all possible situations one could try the algorithm with number of iterations varying from 1 to \( T \). This procedure will take \( (\lg \sqrt{N})^2 \) iterations with only one oracle \( U_\omega \) query.

The Grover’s algorithm provides us not only with the yes/no answer, but also with the solution \( x : f(x) = 1 \). The algorithm proposed here gives only the answer if such \( x \) exists. But this information can effectively help us to find \( x \) by the standard classical procedure.

Let’s consider, for example, the Zero Sum Subset problem. We have set of \( n \) elements and the question if there exists a subset of it with zero sum. We can sequentially remove elements from the initial set and ask our quantum circuit if there still exists a solution. If after removing some element we start to get the ‘no’ answer, that would mean that the removed element belongs to the desired solution set.

The above means that the described here quantum circuit can effectively solve any task from NP, if using quantum states as operators over other quantum systems will be proved to be possible.

References

[1] D. Deutsch, The Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A 400: 97. 1985
[2] D. Deutsch, R. Jozsa Rapid solutions of problems by quantum computation. Proceedings of the Royal Society of London A 439: 553. 1992.
[3] E. Bernstein, U. Vazirani Quantum complexity theory. Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing (STOC ’93), pp. 1120. 1993.
[4] D. R. Simon, On the power of quantum computation. Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on: 116123. 1994.
[5] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring. Proceedings of the 35th IEEE Symposium on Foundations of Computer Science. pp. 124 134. 1994.
[6] L. K. Grover, A fast quantum mechanical algorithm for database search. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212 219. 1996.