№ 2-3 (2), 2020 ISSN: 2682 – 9517 (print) ISSN: 2683 – 0930 (online)
INSTITUTE OF INFORMATICS AND INNOVATIVE TECHNOLOGIES

Solved and Unsolved Problems in Quantum Computing

Petar Nikolov

FDIBA, Technical University of Sofia
8. Kliment Ohridski Blvd, blok 10, Sofia, Bulgaria
pnikolov@fdiba.tu-sofia.bg

Решени и нерешени пролеми в квантовите компютри

Петър Николов

ФаГИОПМ, Технически университет – София
бул."Кл. Охридски" 8, блок 10, София – 1000

***Abstract*—As one of the fastest growing research areas these days, quantum computing becomes more and more interesting in terms of solving real-world problems. Starting from the beginning of quantum computation – the first quantum algorithm for factorization, written by Peter Shor back in the 1990s and coming to present days where people use quantum computers for machine learning and optimization problems, this paper is going to give a quick overview of the solved and unsolved problems.**

Keywords—quantum computing, machine learning, optimization, quantum search, factorization

I. Introduction

Quantum computer was first proposed by Richard Feynman back in 1981 when he pointed out that simulating quantum-mechanical systems would be inefficient, so new type of quantum-mechanical computers must be built [1]. Many hardware manufacturers nowadays are trying to build real quantum processors, and while almost every week there is a technological breakthrough, in this paper I'm only going to make a review of the quantum problems in the algorithmic and software perspective. One of the scariest ideas is about the constantly growing scale of the problems, which solution is only possible when using quantum computers. The quantum computers give hope that the hardest problems in science might have solution (at least could be solved efficiently), and this hope is coming with a new problem – the algorithms. Since quantum and classical computations are quite different, we cannot run classical algorithms on quantum hardware and expect speedup – we need to design and develop new algorithms, especially for the quantum hardware. Quantum algorithms may be classified into those based on Quantum Fourier Transform and those based on Grover's search algorithm [2,3]. Classical computers work with bits and each bit has deterministic state – 0 or 1 at a moment of time. The fundamental quantum-carrying elements are qubits (single atom, electron, photon on superconducting or optical circuit). The huge complexity (compared with the classical systems) of quantum-mechanical systems comes when there needs to be given a full description of highly entangled quantum states [4,5].

I can define three main reasons why we need quantum computers and how these would outperform the classical ones for high complexity problems:

- Quantum algorithms for classical problems [6,7] - these are problems which are well-known to be hard to solve on classical computers, but quantum algorithms could perform significantly faster.
- Complexity theory arguments – The states of a quantum-mechanical system have super-classical properties. And if a quantum register is measured, this is sampling from a correlated probability distribution, which can't be sampled efficiently by classical methods. [8,9]
- Classical computers cannot simulate quantum computers efficiently. [1]

II. Solved and Unsolved Problems

Quantum Search

The quantum search algorithm, developed by Grover, performs a search over unordered set ot N=2n items to find a unique element which satisfies the predifined conditions [3]. The algorithm requires only O(√N) operations to perform the search (quadratic speedup over its classical opponets).

How this algorithm works?

The search problem needs to be translated into quantum-mechanical problem. The quantum systems must have N =2n states, where the representation of these states are n-bit strings. The unique state S_v must satisfy the condition $C(S_v) = 1$ and all other states C(S) = 0. The three steps of the algorithm are as follows:

- System initialization by applying the Walsh-Hadamard gate to each qubit in the quantum register, so the result is equal probability for each register value.
- Repetition of a unitary operation O(√N) times:

a. If C(S) = 1: rotate by π radians, else if C(S) = 0: leave system unaltered.

b. Apply diffusion transform D, where D=WRW, and W is the Walsh-Hadamard transformation and R is rotation matrix.

R_{ij}=0 if i=j.

R_{ii}=-1 if i≠0.

$W_{ij} = 2^{-n/2}(-1)^{i.j}$

c. Sample the resulting states. The final state S_v with probability ≥1/2 if $C(S_v)$=1.

A. Factorization

The problem of factorization is a problem in number theory. It is decomposition of large integer numbers to a product of small prime numbers. Peter Shor introduces his algorithm for quantum factorization in 1994 [6] and he shows that using quantum computers can speed-up dramatically the task for prime factorization. The fastest classical equivalent takes exponential time, while the quantum algorithm is giving a solution in polynomial time [7].

The quantum algorithm for prime factorization uses one of the fundamental properties of quantum systems – the coherence. The coherence describes the correlation between several wave packets.

The main part of the factoring algorithm is the period finding. If ther is a periodic function f, where f maps some numbers {0, 1, ..., M − 1} to some set S, such that ∀x , f (x) = f (x + r). The task is to find the period r. The number of the repetitions of the period is M/r. If this function is considered on a single period, then f is 1-1: values are never repeated. This is the first condition to the period finding: that f is 1-1 for each period. The second condition is that r divides M. In order to solve the factoring problem, the M/r >> r must be true (M >> r2).

What does the quantum algorithm do? It could be divided into two parts:

The order finding problem – classical implementation.

The second part is quantum solution to the order finding problem:

- Initialization of a quantum register of length k into 1 |0k> state. Then apply the Walsh-Hadamard transformation to all the k qubits. This will result in equial probability for each of the 2^k states for the register.
- Construct the f(x) function as quantum function and apply it to the register. The result will be superposition of k+n qubits.
- Apply the inverse Quantum Fourier transform to the input register |x>.
- Perform quantum measurement both on the input and the output register.
- Perform continued fraction expansion to find the appropriate period r. (If the solution for the period is prime factor, here is the end, otherwise obtain more candidates for r.)

B. HHL

Operation with linear equations, linear functions and matrices are part of the branch of mathematics called “Linear algebra”. The linear algebra is essential part of many machine learning algorithms and solving systems of linear equations is very common problem in science and engineering.

Approximating the solution for N linear equations in N unknown parameters takes time of order N with classical methods. A quantum algorithm in some cases can approximate the value of a function of the full solution to these N equations scaling logarithmically in N [10]. The problem for this algorithm looks like: a Hermitian matrix A = N x N and a unit vector b is given.

Find vector x, such that Ax=b.

The algorithm:

a. Represent b as quantum state [b> = Σbi

b. Use Hamiltonian simulation techniques to apply e^{iAt} to |b> for a superposition of different times [11].

c. Use phase estimation to decompose |b> [12]. The result is: Σβ|μ>|y>

d. Use a non-unitary operation to do a linear mapping.

Some of the potential applications of this algorithm are: Linear regression, Supervised classification, Support Vector Machine, Hamiltonian simulations.

C. *Quantum PCA*

The principal component analysis (PCA) is a bedrock to dimensionality reduction technique for probability and statistics, commonly used in data science and machine learning applications, where there is big dataset with statistical distribution and the low-dimensional patterns must be uncovered.

In the quantum way, this problem could be translated into revealing properties of unknown quantum state [13]. A quantum coherence can be created among multiple copies of a randomly generated quantum state to perform a quantum principal component analysis, which reveals the eigenvectors corresponding to the large eigenvalues of this quantum state. This requires O(d) operations in qRAM divided over O(log d) steps, which could be executed in parallel. The quantum tomography is a widely used tool, where a given multiple copies of an unknown quantum state in d-dimensional Hilbert space are being measured with various techniques in order to extract useful information showing some features of the state [14,15,16]. Multiple copies of the state can play active role in its own measurement and implement the unitary operator $e^{-i\rho t}$ energy operator or Hamiltonian, which generates transformations on other states.

Exponentiate density matrix - this exponentiates non-sparse matrices in O(logn), which is exponential speed-up over its classical opponents.

Using Suzuki-Trotter expansion the e^{-iXt} can be constructed for non-sparce positive matrix X:

Σek = 1 for X

Application to quantum phase estimation algorithm to find the eigenvalues and eigenvectors of the unknown density matrix.

Advantages and future applications of quantum self-tomography:

Reveals eigenvectors and eigenvalues in time O(Rlogd) compared to the compressive tomography (O(Rdlogd)). [15]

The density matrix exponentiation is time optimal. [13]

Quantum self-tomography is comparable to group representation methods, but not only the spectrum is approximated - also as a result, the eigenvectors are found. [17]

Speed-up of some machine learning problems in clustering and pattern recognition. [18]

D. *Input and Output Problem*

Loading classical data into a quantum computer is a bottleneck for some algorithms. Most quantum machine learning algorithms require exponential time procedures to load data into quantum states. One solution to this problem is using quantum Random Access Memory (qRAM), but it is a costly solution for big datasets.

Similar problem is noticeable when a readout for a quantum system is required. Also known as the 'output problem'. It is a common problem for all linear algebra-based quantum machine learning algorithms, since it is exponentially hard to estimate the classical quantities for the solution vector of the qPCA algorithm.

The quantum information is very different from its classical counterpart, because it exists in superposition and it is hard to measure it – every observation made on the quantum register leads to collapse of the superposition. One of the main differences between quantum and classical computing is the representation of the information. The fundamental limits on operations with a quantum state are:

No-cloning theorem – no unknown quantum state can be cloned perfectly, unless it is known to belong to a set of pairwise orthogonal states. [19,20]

It is not possible to extract more than n bits of classical information from n qubits (Holevo's theorem) [21]. For n qubits, all possible amplitudes are 2^n, so only a small amount of the information can be extracted and classically represented.

E. *Benchmark Problem*

The benchmark problem is a general problem not only for the quantum algorithms, but also a huge research area in classical computer science. In the quantum world the benchmark problem is connected not only with need for probing performance of

quantum computers against their classical counterparts for identical (similar) problems, but also for a comparison between various quantum hardware back-ends. The stud of time/space performance trade-off is done via a family of rectangular quantum circuits [22].

The quantum benchmark is a set of quantum circuits and instructions, analysis procedure and interpretation rules [22], and there exists few families of benchmarks, each of them measuring different metrics:

- Quantum Volume
- Randomized Benchmark
- Long-sequence gate set tomography
- Volumetric bechmarks

III. CONCLUSION

The quantum information theory has a great advantage in handling hard and complex scientific problems coming from the fundamental properties of quantum-mechanical systems. The development of new quantum algorithms, however, is not an easy task, and many considerations in the process must be taken into account – implementation and hardware limitations, methods for validation of the results, etc. This paper gives an overview of the quantum solutions with some basic analysis up to 2020.

REFERENCES

[1] Feynman, R. P. (1982). Simulating physics with computers. *Interna*tional Journal of Theoretical Physics, 21(6-7):467–488.

[2] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journalon Computing, 26(5):1484–1509.

[3] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, pages 212–219, New York, NY, USA. Association for Computing Machinery.

[4] Jordan, S. (2019), Quantum Algorithm Zoo, https://quantumalgorithmzoo.org/, [online]

[5] Montanaro, A. (2016). Quantum algorithms: an overview, npj Quantum Information.

[6] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5):1484–1509.

[7] Chuang, I. L., Laflamme, R., Shor, P. W., and Zurek, W. H. (1995). Quantum computers quantum computers, factoring, and decoherence. Science, 270(5242):1633–1635.

[8] Lund, A. P., Bremner, M. J., and Ralph, T. C. (2017). Quantum sampling problems, BosonSampling and quantum supremacy. Npj Quantum Information.

[9] Harrow, A. W. and Montanaro, A. (2017). Quantum computational supremacy. Nature, 549(7671):203–209.

[10] Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. Physical Review Letters, 103(15).

[11] Berry, D. W., Ahokas, G., Cleve, R., and Sanders, B. C. (2006). Efficient quantum algorithms for simulating sparse hamiltonians. Communications in Mathematical Physics, 270(2):359–371.

[12] Luis, A. and Peřina, J. (1996). Optimum phase-shift estimation and the quantum description of the phase difference. Phys. Rev. A, 54:4564–4570.

[13] Lloyd, S., Mohseni, M., and Rebentrost, P. (2014). Quantum principal component analysis. Nature Physics, 10(9):631–633.

[14] Nielsen, M. A. and Chuang, I. L. (2011). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, USA, 10th edition.

[15] Gross, D., Liu, Y.-K., Flammia, S. T., Becker, S., and Eisert, J. (2010). Quantum state tomography via compressed sensing. Physical Review Letters, 105(15).

[16] Shabani, A., Kosut, R. L., Mohseni, M., Rabitz, H., Broome, M. A., Almeida, M. P., Fedrizzi, A., and White, A. G. (2011a). Efficient measurement of quantum dynamics via compressive sensing. Phys. Rev. Lett., 106:100401.

[17] Keyl, M. and Werner, R. F. (2001). Estimating the spectrum of a density operator. Phys. Rev. A, 64:052311.

[18] Rebentrost, P., Mohseni, M., and Lloyd, S. (2014). Quantum support vector machine for big data classification. Physical Review Letters, 113(13).

[19] Wootters, W. K. and Zurek, W. H. (1982). A single quantum cannot be cloned. Nature, 299(5886):802–803.

[20] E., Brassard, G., and Gambs, S. (2006). Machine learning in a quantum world. In Advances in Artificial Intelligence, pages 431–442. Springer Berlin Heidelberg.

[21] Holevo, A. (1973). Bounds for the quantity of information transmitted by a quantum mechanical channel. Problems of Information Transmissions, 9:177–183.

[22] Blume-Kohout, R. and Young, K. C. (2019). A volumetric framework for quantum computer benchmarks.