ISSN 0278-6419, Moscow University Computational Mathematics and Cybernetics, 2019, Vol. 43, No. 1, pp. 47–51. © Allerton Press, Inc., 2019.
Russian Text © K.R. Khadiev, L.I. Safina, 2019, published in Vestnik Moskovskogo Universiteta, Seriya 15: Vychislitel'naya Matematika i Kibernetika, 2019, No. 1, pp. 76–83.

Quantum Algorithm for Shortest Path Search in Directed Acyclic Graph[#]

K. R. Khadiev[1,2*] **and L. I. Safina**[2**]

[1]OOO Kvantovye Intellektual'nye Tekhnologii, Kazan, 420111 Russia

[2]Kazan (Volga Region) Federal University, Kazan, 420008 Russia

Received July 26, 2018

Abstract—In this work, we consider a well-known "Single Source Shortest Path Search" problems for weighted directed acyclic graphs (DAGs). We suggest a quantum algorithm with time complexity $O(\sqrt{nm}\log n)$ and $O(1/n)$ error probability, where n is a number of vertexes, m is the number of edges. We use quantum dynamic programming approach (Khadiev, 2018) and Dürr and Høyer minimum search algorithm to speed up our search. Our algorithm is better than C. Dürr and co-authors' quantum algorithm for an undirected graph. The time complexity of C. Dürr's algorithm is $O(\sqrt{nm}(\log n)^2)$. The best known deterministic algorithm for the problem is based on a dynamic programming approach and its time complexity is $O(n+m)$. At the same time, if we use algorithms for general graphs, then we do not get better results. The time complexity of best implementations of Dijkstra's algorithm with Fibonacci heap is $O(m+n\log n)$.

Keywords: quantum algorithms, graph theory, shortest path search, query model, algorithms.

DOI: 10.3103/S0278641919010023

1. INTRODUCTION

Quantum computing [1–4] is one of the hot topics in computer science for last decades. Researchers have found many problems for which quantum algorithms outperform the best known classical algorithms [5].

One of such problems is the "Single Source Shortest Path Search" problem. Here we present a new quantum algorithm that shows better distance for quantum and classical complexities.

We consider directed weighted topologically sorted acyclic graph with N vertexes and M edges. The problem is searching lengths of paths from vertex s to all other vertices.

If we want to apply algorithms that work for undirected graphs, then they have the following time complexities. Floyd–Warshall algorithm [6] has $O(N^3)$, Bellman-Ford algorithm [6] has $O(NM)$, Dijkstra's algorithm [6] with Fibonacci heap [6, 7] or Brodal queue [8] has $O(M+N\log N)$. For directed acyclic graph (DAG), we can apply dynamic programming approach [6]. The time complexity of such algorithm is $O(N+M)$.

Researchers [9, 10] have suggested the quantum algorithm for "Single Source Shortest Path Search" problem for undirected graphs. The algorithm's complexity is $O(\sqrt{NM}\log^2 N)$, and it is based on ideas of Dijkstra's algorithm and C. Dürr, P. Høyer [11] minimum search algorithm.

In this paper, we present the quantum algorithm that is based on ideas of dynamic programming approach for DAG [12] and C. Dürr, P. Høyer [11] minimum search quantum algorithm. This combination allows us to have time complexity $O(\sqrt{NM}\log N)$ and error probability $O(1/N)$. Similar approaches were used in [9, 10,13–15] for different graph problems.

[*] E-mail: kamilhadi@gmail.com.

[**] E-mail: liliasafina94@gmail.com.

[#] The article was translated by the authors.

Note, that time complexity depends on the way of storing the graph in the memory. We consider two graph storing models: "neighbor list" and "adjacency matrix". Above we present complexities for "neighbors list". For "adjacency matrix", Dynamic programming approach has $O(N^2)$, quantum algorithm of Dürr et al. has $O(N^{1.5}(\log N)^2)$. Our algorithm has $O(N^{1.5}\log N)$ time complexity. So, it has better time complexity comparing with known quantum and classical algorithms.

The paper has the following structure. Section 2 contains the main definitions. There is the description of the algorithm in Section 3. The conclusion is in Section 4.

2. DEFINITIONS

Let us consider a formal description of the problem. We have a weighted directed acyclic graph (DAG) $G=(V,E)$, where V is the set of vertices and E is the sent of directed edges. It is known that for any edge $e=(v_i,v_j)$, $e\in E$, $v_i,v_j\in V$, the condition $i<j$ holds. Let $N=|V|$ be number of vertexes and $M=|E|$ be number of edges. Let $\mathcal{D}_i=\{v_j:\exists e=(v_j,v_i)\in E\}$ where $i\in\{1,\dots N\}$ and $d_i=|\mathcal{D}_i|$. The $w(j,i)$ is weight of the edge $e=(v_j,v_i)\in E$, where $i,j\in\{1,\dots N\}$. If $e\notin E$, then $w(j,i)=+\infty$; $w(i,i)=0$ for $i\in\{1,\dots N\}$.

For solving P_s problem, we should find lengths of shortest paths from vertex v_s to all other reachable vertices, for $s\in\{1,\dots,N\}$.

For the "neighbors list" model of graph storing, we have L array such that $L[i]$ is list of (v,w) pairs, where $v\in\mathcal{D}_i$ and w is weight of the $e=(v,v_i)$ edge. The list $L[i]$ is sorted by v. If j-th element of the list is (v,w), then we denote them $L[i][j].v$ and $L[i][j].w$ respectively.

For the "adjacency matrix" model of graph storing, we have B matrix such that $B[i][j]=w(i,j)$.

2.1. The Computation Model

Let us describe the basic concepts of quantum computing. The quantum bit (qubit) is complex-value vector $|\psi\rangle=(a_0,a_1)$ such that $|a_0|^2+|a_1|^2=1$. Dirac notation [1, 17] is used quantum computing. The $|\psi\rangle$ vector from Hilbert space $\mathcal{H}^2$. a_i is amplitude of i-th state. We can apply two transformation to qubit: unitary transformation and measurement. The unitary transformation is a multiplication the qubit to unitary 2×2-matrix. The measurement operation allows us to obtain information from qubit. We get 0-result and $|0\rangle=(1,0)$ state with probability $|a_0|^2$; and 1-result and $|1\rangle=(0,1)$ state with probability $|a_1|^2$.

The register of t qubit is complex value 2^t-vector $|\psi\rangle=(a_0,\dots,a_{2^t})\in\mathcal{H}^{2^t}$, such that $\sum_{i=0}^{2^t-1}|a_i|^2=1$. The unitary transformation is a multiplication the qubit to unitary $2^t\times 2^t$-matrix. The measurement gets i-result with probability $|a_i|^2$.

One of the popular computation models for quantum algorithms is query model [1, 2]. In this model, we do a query to the oracle that has access to graph storing model (the "neighbors list" or the "adjacency matrix"). Query is unitary transformation with diagonal matrix Q such that $q_{ii}=1$ for $f(i)=0$ and $q_{ii}=-1$ for $f(i)=1$, where f is some function that can be computed by the oracle. The Oracle can compute only functions that have constant or logarithmic time complexities.

3. THE QUANTUM ALGORITHM FOR THE "SINGLE SOURCE SHORTEST PATH SEARCH" PROBLEM

The quantum algorithm is based on ideas of dynamic programming approach for DAG [6,12] and C. Dürr, P. Høyer minimum search quantum algorithm [11]. The minimum search quantum algorithm is based on Grover's search algorithm [16] and has the following property [11].

Property 1. The C. Dürr, P. Høyer [11] minimum search quantum algorithm finds index of maximal or minimal element in unordered sequence of length K. It's expected time complexity is $O(\sqrt{K})$ and error probability is $\frac{1}{2}$.

Let us use the following notation that is used in our algorithm:

- s is the index of the start vertex.

- $h[i]$ is the length of the shortes path from v_s to v_i. If v_i is not reachable from v_s, then $h[i] = +\infty$.

The quantum algorithm is following:

```
Algorithm 1.
$h = \{+\infty, \dots, +\infty\}$;
$h[s] = 0$;
for ($i$ from $s+1$ to $N$)
    $h[i] = \min(h[j_1] + w(j_1, j), \dots, h[j_u] + w(j_u, i))$,   where   $u = d_i$,   $\mathcal{D}_i = (v_{j_1}, \dots, v_{j_u})$.
```

The $\min$ is the quantum subroutine that returns the minimal value from its arguments, and its implementation is based on C. Dürr, P. Høyer [11] minimum search quantum algorithm.

Let us show the correctness of the algorithm and compute the time complexity and the error probability.

Lemma 1. *Algorithm 1 computes lengths of the shortest paths to all vertexes that are reachable from v_s. The success probability is $O\left(\frac{1}{2^N}\right)$.*

Proof. Because of the topological sorting of the graph, when we computing $h[i]$, we already have computed $h[j_1], \dots, h[j_u]$.

Let us proof correctness by induction. Assume that $h[j_1], \dots, h[j_u]$ are computed with no error.

Suppose that the minimum is $h[r] + w(r, i)$, but the length of shortest path is $z < h[r] + w(r, i)$. Then, there is p such that v_p belongs to the shortest path and the path contains path from v_i to v_p and the (v_p, v_i) edge. So, $z = h[p] + w(p, i) < h[r] + w(r, i)$, but it is impossible because $h[r] + w(r, i)$ is minimum.

Let us compute the probability of the right answer. Due to Property 1, the proability of computing $h[i]$ with no error is $O(\frac{1}{2})$, for $i \in \{s+1, \dots, N\}$. Therefore, the probability of computing all $h[i]$ is $O(\frac{1}{2^N})$ because the events are independent.

Lemma 2. *The expected time complexity of Algorithm 1 is $O(\sqrt{NM})$ for the "neighbors list" and $O(N^{1.5})$ the "adjacency matrix".*

Proof. Let us compute the time complexity for the neighbors list. Due to Property 1, the expected time of v_i vertex processing is $O(\sqrt{d_i})$. Therefore, the time complexity of processing of all vertices is $O\left(\sum\limits_{i=1}^{N} \sqrt{d_i}\right)$. Due to Cauchy-Bunyakovsky-Schwarz inequality, we have the following time complexity

$$O\left(\sum_{i=1}^{N} \sqrt{d_i}\right) \leqslant O\left(\sqrt{N \sum_{i=1}^{N} d_i}\right) = O\left(\sqrt{NM}\right).$$

The last equality is right because M is the number of edges.

Let us compute the time complexity for the "adjacency matrix". Due to Property 1, the expected time of v_i vertex processing is $O(\sqrt{N})$. Therefore, the time complexity of processing of all vertices is $O\left(N\sqrt{N}\right) = O\left(N^{1.5}\right)$.

We cannot apply algorithm for big N because $\lim\limits_{N \to \infty} \frac{1}{2^N} = 0$.

Table 1. Algorithms for the Single Source Shortest Path Search Problem

Name	Time Complexity	
	neighbors list	adjacency matrix
Brute force	$O(NM2^M)$	$O(N^3 2^M)$
Dijkstra Algorithm, naive implementation	$O(N^2+M)$	$O(N^2)$
Dijkstra Algorithm with binary heap	$O(M\log N)$	$O(N^2)$
Dijkstra Algorithm with Fibonacci heap	$O(N\log N+M)$	$O(N^2)$
Bellman–Ford Algorithm	$O(NM)$	
Floyd–Warshall algorithm	–	$O(N^3)$
Dinamic Programming	$O(N+M)$	$O(N^2)$
Quantum algorithm of C. Dürr et al.	$O(\sqrt{NM}\log^2 N)$	$O(N^{1.5}\log^2 N)$
Our quantum algorithm	$O(\sqrt{NM}\log N)$	$O(N^{1.5}\log N)$

3.1. Improvement of the Quantum Algorithm

We invoke Dürr's and Høyer's algorithm $K = 2\log_2 N$ times for better success probability. Then we choose the best value from these K results.

So, we obtain the following algorithm.

Algorithm 2.

```
$h = \{+\infty, \ldots, +\infty\}$;
$h[s] = 0$;
for($i$ from $s+1$ to $N$)
  for ($j$ from $1$ to $\lceil 2\log_2 N\rceil$)
      $h[i] = \min\big(h[i], \min(h[j_1] + w(i, j_1), \ldots, h[j_u] + w(i, j_u))\big)$,
where $u = d_i$, $\mathcal{D}_i = (v_{j_1}, \ldots, v_{j_u})$.
```

The new version of the algorithm has the following property:

Theorem. *Algorithm 2 computes lengths of the shortest paths to all vertexes that are reachable from v_s. The success probability is $O(1-\frac{1}{N})$. The expected time complexity of Algorithm 2 is $O(\sqrt{NM}\log N)$ for the ”neighbors list” and $O(N^{1.5}\log N)$ the “adjacency matrix”.*

Proof. The proof of correctness is similar to the proof from Lemma 1. Let us compute the error probability for one vertex. The probability of the event that there are no minimal element among results of $K = 2\log_2 N$ invocations of min subroutine is $Pr_{err}(i) = \frac{1}{2^K} = \frac{1}{2^{2\log_2 N}} = \frac{1}{N^2}$. Therefore, the right result probability for vertex i is

$$Pr_{corr}(i) = 1 - Pr_{err}(i) = 1 - \frac{1}{N^2}.$$

The right result probability for the whole graph is

$$Pr_{corr} = \left(1 - \frac{1}{N^2}\right)^N = O\left(1 - \frac{1}{N}\right)$$

Note that $Pr_{corr} \to 1$ for $N \to \infty$.

The time complexity can be computed like in the proof of the Lemma 2.

4. CONCLUSION

We show properties of different algorithms for the problem in the table.

The suggested quantum algorithm works faster than existing classical and quantum algorithms for the Single Source Shortest Path Search Problem for DAG in a case when $M > N(\log_2 N)^2$. The algorithm shows the best efficiency for $M = O(N^2)$.

The work is supported by Russian Science Foundation Grant 17-71-10152.

REFERENCES

1. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge Univ. Press, Cambridge, 2010).
2. A. Ambainis, "Understanding quantum algorithms via query complexity," arXiv.2017.N 1712.06349.
3. A. C. C. Say and A. Yakarylmaz, "Quantum finite automata: a modern introduction," Lect. Notec Comput. Sci. **8808**, 208–222 (2014).
4. R. de Wolf, "Quantum computing and communication complexity," PhD Thesis (Univ. Amsterdam, Amsterdam, 2001).
5. S. Jordan, Quantum Algorithms Zoo. https://math.nist.gov/quantum/zoo.
6. A. C. C. Say and A. Yakarylmaz, "Quantum finite automata: a modern introduction," Lect. Notec Comput. Sci. **8808**, 208–222 (2014).
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms,* 2nd ed. (McGraw-Hill, New York, 2001).
8. G. S. Brodal, G. Lagogiannis, and R. E. Tarjan, "Strict Fibonacci heaps," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing STOC '12* (ACM, New York, 2012), pp. 1177–1184.
9. G. S. Brodal, "Worst-case efficient priority queues," in *Proceeding of 7th Annual ACM-SIAM Symposium SODA'96 on Discrete Algorithms* (ACM–SIAM, New York–Philadelphia, 1996), pp. 52–58.
10. C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla, "Quantum query complexity of some graph problems," in *Proceeding of International Colloquium on Automata, Languages, and Programming* (Springer, 2004), pp. 481–493.
11. C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla, "Quantum query complexity of some graph problems," SIAM J. Comput. **35**, 1310–1328 (2006).
12. C. Dürr and P. Høyer, "A quantum algorithm for finding the minimum," quant-ph/9607014 (1996).
13. K. Khadiev, "Quantum dynamic programming algorithm for DAGs. Applications for AND-OR DAG evaluation and DAG's diameter search," arXiv:1804.09950 (2018).
14. A. Ambainis and R. Špalek, "Quantum algorithms for matching and network flows," in *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science* (Springer, 2006), pp. 172–183.
15. S. Dürr, "Quantum algorithms for matching problems", Theory Comput. Syst. **45**, 613–629 (2009).
16. S. Dürr, "Quantum complexity of graph and algebraic problems," PhD Thesis (Univ. Ulm, Ulm, 2008).
17. L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th annual ACM Symposium on Theory of Computing* (ACM, New York, 1996) pp. 212–219.
18. R. Shankar, *Principles of Quantum Mechanics* (Springer, London, 2012).