THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 22, Number 1/2021, pp. 93–100

QUANTUM IMPLEMENTATION OF THE MODIFIED DIJKSTRA'S ROUTING ALGORITHM

Robert BOTEZ, Iustin-Alexandru IVANCIU, Calin-Marian IURIAN, Virgil DOBROTA

Technical University of Cluj-Napoca, No. 28 Memorandumului Street, Cluj-Napoca, 400114, Romania
Corresponding author: Virgil DOBROTA, E-mail: Virgil.Dobrota@com.utcluj.ro

Abstract. Quantum computing can be used to solve certain problems faster than classical computers. One such problem is related to optimal routing in Internet of Things and sensor networks where the number of nodes is very high and multiple constraints (energy efficiency, latency, throughput) must be met. We propose a quantum implementation of the Modified Dijkstra's algorithm for finding the optimal route between a source node and a destination node in a connected graph. This is a particular case of the Travelling Salesman Problem, which is NP hard. Our contribution is two-fold: first of all, we encode both the Available Transfer Rate and the One-Way Delay in the controlled unitary matrices in order to be able to use Quantum Phase Estimation for computing the composite metric of a path. Second, we use an optimized version of the Grover quantum search algorithm to find the path with the minimum composite metric with zero failure rate.

Key words: Grover search, Modified Dijkstra's Algorithm, Qiskit, quantum routing, quantum phase estimation.

1. INTRODUCTION

Over the past few years, requirements in both data storing, processing and transmitting have increased in an exponential rate. A forecast by Cisco [1] predicts that global mobile data traffic will reach 77.5 exabytes per month by 2022 and annual traffic will reach almost one zettabyte. Nearly 80% of this traffic will be video. As such, computing devices need to work at higher speeds and with better efficiency. Quantum computing takes advantage of quantum mechanical effects in order to solve certain problems faster than classical computers. It is especially suitable for solving NP hard problems, such as optimal routing in different network topologies. Moreover, reducing transistors size means that their operation becomes highly influenced by quantum effects thus making the study of quantum computation systems a top priority. IBM for example offers both simulators and real systems in their IBM Q Experience in a gate-based architecture up to 65 qubits [16]. A 1000-qubit quantum computer is announced for delivery by 2023. D-Wave's most recent model is called Advantage and offers more than 5000 qubits [17].

The main advantage of quantum computing systems stems from their inherent parallelization capabilities. A classical bit may either have a value of 0 or 1. A qubit, on the other hand, may have the values $|0\rangle$, $|1\rangle$, or any superposition of the two [2]. The column vector of a quantum state is called the ket representation and is denoted by $|\cdot\rangle$. For two qubits, the quantum state may have the following four values simultaneously: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. In a classical n-bit register, only one of the 2^n combinations is active at any given moment. However, in an n-qubit quantum register, the quantum state is a superposition of all 2^n values simultaneously. Any quantum operation applied to such a quantum register leads to modifying all values at the same time. It is this parallel processing capability that enables quantum algorithms to require fewer computational steps or find more suitable solutions for specific problems than their classical implementations.

This paper proposes a quantum implementation of the Modified Dijkstra's Algorithm [3] in Qiskit. The solution finds the optimal route between given source and destination nodes in a graph using phase estimation [10] and a modified version of Grover's search algorithm [6]. First, we study the feasibility of adapting the quantum phase estimation algorithm used in solving the Travelling Salesman Problem in order to compute the Composite Metric (CM). Second, the CMs of the paths are encoded in a quantum database and the minimum value is computed using an optimized quantum minimum searching algorithm. Compared to other solutions [10], the novelty in our approach consists in encoding a second element, the Available Transfer Rate (ATR), in the controlled unitary matrices, without using additional qubits. Moreover, we use a modified version of the traditional Grover quantum search algorithm to find the path with the lowest CM, with zero failure rate.

The remainder of this paper is organized as follows. Section 2 discusses related work. The architecture and design of the system is presented in Section 3. We discuss and evaluate the proposed solution in Section 4 and conclude in Section 5.

2. RELATED WORK

The Travelling Salesman Problem (TSP) represents a classical optimization problem in computer science. It involves a salesman who must travel through N cities but can only visit each city once. Moreover, the salesman must start and end his travel in the same city. Cities are connected by links with different costs. The end goal is to minimize the total cost of the travel in the conditions described above. This problem belongs to the class of NP hard problems and the simplest way to solve it is by brute force. However, as the number of cities N increases, so does the number of iterations needed to find the solution, which is $(N-1)!$. As such, different strategies to tackle this problem must be found. A possible approach involves the use of quantum computing. Moylett et al. [8] proved a quadratic quantum speedup can be achieved by applying a quantum backtracking algorithm to a classical approach. Markevich and Trushechkin [9] propose a quantum branch-and-bound algorithm which uses the quantum nested searching algorithm. They prove that in most cases, a classical approach is quicker than the quantum algorithm due to greater adaptability. However, the operation time of the quantum algorithm is the same for all problems, while the classical algorithm may run very slowly for certain cases. This makes the quantum branch-and-bound algorithm more efficient as the number of cities increases. The authors of [10] use the phase estimation technique and the Durr-Hoyer search algorithm [7] on four cities, using IBM's quantum simulator. Paper [11] presents a Quantum Approximate Optimization Algorithm implemented in pyQuil which uses mixer operators based on hard and soft constraints. Warren [12] analyzes four software programs that solve the TSP problem on a quantum annealer and provides a pertinent list of strength and weaknesses for each of them. An entirely different framework is the adiabatic quantum computation AQC. Kieu [13] provides such an algorithm for the TSP which may be implemented in quantum optics or quantum field theory. However, this algorithm is not meant for quantum computers using unitary gates acting on qubits.

3. ARCHITECTURE AND DESIGN

The graph on which we apply Modified Dijkstra's Algorithm is illustrated in Fig. 1. Considering node 1 as the source and node 4 as the destination, there are three possible paths between these nodes: $1->4$, $1->2->4$ and $1->3->4$. As described in [3], the ATR and OWD are computed globally for each path. Measuring the ATR on each link can be done by using the active measurement technique and Kalman Filtering described in [14]. For estimating the OWD several cyclic measurements are performed, as described in [15]. Note that in our implementation we disregard the BER for the reasons mentioned in [3].

Our goal is to find the path with the minimum composite metric/cost between nodes 1 and 4. This is a two-step process: first, we use quantum phase estimation to compute the cost of each path. As explained in [10], instead of encoding the cost of each link as an element of a matrix, we represent them as phases. This

way the tensor products used in quantum computing will lead to the summation of these costs, ultimately providing the composite metric for an entire path. The second step involves employing Quantum Maximum or Minimum Searching Algorithm (QUMMSA) [6], which enables us to find the path with the minimum cost. The following paragraphs provide a thorough description of the phase estimation and QUMMSA algorithms. Both algorithms were implemented in Python and executed on simulators using the Qiskit framework.

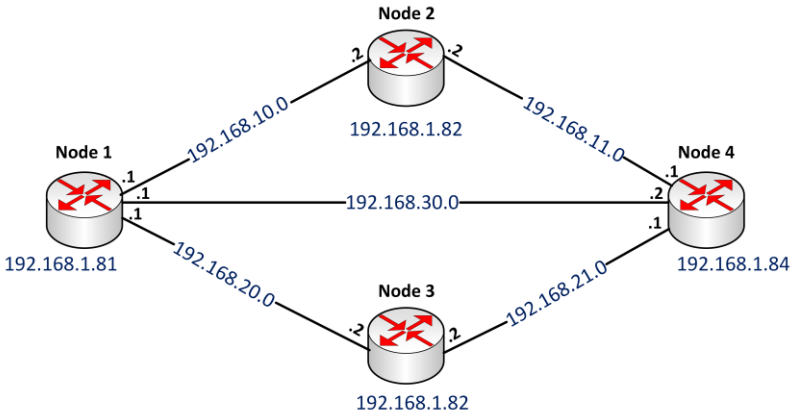

Fig. 1 – Graph for Modified Dijkstra's Algorithm.

3.1. Quantum Phase Estimation

We propose an original version of the phase estimation algorithm used to solve the Travelling Salesman Problem [10]. The input is represented as a matrix A where $A_{ij} = e^{j\phi_{ij}}$ and ϕ_{ij} is the cost of travelling from node i to j. Next, we construct unitaries U_j from matrix A:

$$\left[U_j\right]_{kk} = \frac{1}{\sqrt{N}}\left[A\right]_{jk}. \tag{1}$$

N represents the number of nodes and $j,k \in \left[1,N\right]$. The main difference from [10] is the fact that we use a composite metric which involves both the ATR and the OWD. Moreover, the ATR of a path is not additive like the OWD. Consequently, we must find a way of also encoding it into matrix A. The solution we proposed consists in using the elements on the main diagonal of this matrix. This is possible because the cost of going from node i to node i is 0 and therefore not used in computing the cost of the path. Note however that in this approach we can only encode N additional elements. This number increases to $N \times \left(N+1\right)/2$ if we consider symmetrical links. Matrices U_j are diagonal unitary matrices, with all the other elements initialized to 0. Matrix U can now be computed as the tensor products of all unitaries U_j: $U = U_1 \otimes U_2 \otimes U_3 \otimes ... \otimes U_N$. Matrix U is also diagonal and its eigenvalues represent the costs of the paths from source and destination.

These eigenvalues can be estimated using the quantum phase estimation algorithm [4]. It is important to normalize the phases in unitaries U_j such that the eigenvalues are bound between 0 and 2π. Out of the N^N diagonal elements of matrix U we are only interested in the ones corresponding to the possible paths between the source and destination nodes, in our case the three paths presented earlier. We extract these elements by carefully preparing computational basis eigenstates corresponding to the desired locations and performing phase estimation as proposed in [10]. The results are obtained in binary form on the n control qubits.

A Qiskit quantum circuit used for phase estimation, such as the one presented in Fig. 2, must be designed for each eigenstate corresponding to a certain path.

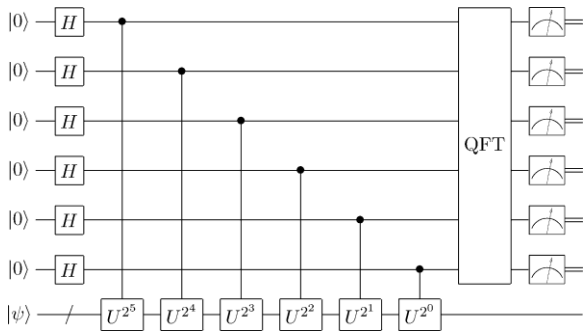

Fig. 2 – Phase estimation quantum circuit implemented in Qiskit.

3.2. Optimized Quantum Maximum or Minimum Searching Algorithm

The Optimized Quantum Maximum or Minimum Searching Algorithm (QUMMSA), introduced in [6] allows finding the maximum or minimum element from an unsorted database with N items. It is based on the Grover-Long algorithm which initially finds $M \geq 1$ solutions: a random value d_0 is taken as threshold and all $M = d_0+1$ solutions will be less than or equal to this value. Out of the M solutions one is selected as the new threshold and the algorithm is repeated until $M = 1$. All M solutions are given with the same probability and, after each step, the number of solutions reduces by half, on average. Therefore, the number of steps needed to find the minimum is, in theory, $\log_2 N$. The first step in the Grover-Long algorithm is preparing the initial state by using the W operator:

$$\left|\psi\right\rangle = W\left|0^{\otimes n}\right\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\left|i\right\rangle = \sqrt{\frac{M}{N}}\left|\psi_{good}\right\rangle + \sqrt{\frac{N-M}{N}}\left|\psi_{bad}\right\rangle. \tag{2}$$

$\left|\psi_{good}\right\rangle$ stores the desired solutions while $\left|\psi_{bad}\right\rangle$ stores other values. When $N = 2^n$, n – the number of qubits, the initial state is a uniform superposition state and W becomes $H^{\otimes n}$, which is the Walsh-Hadamard transformation. Next, each iteration of Grover is divided into four operators:

$$G = WI_0W^{-1}O. \tag{3}$$

O is the oracle which performs a phase inversion on $\left|\psi_{good}\right\rangle$ and I_0 is a conditional phase shift operator used to perform a phase inversion on $\left|0\right\rangle$.

Instead of using phase inversion, Grover-Long [5] rotates the phase with an adjustable angle ϕ:

$$\phi = 2\arcsin\left(\frac{\sin\frac{\pi}{4J+2}}{\sin\beta}\right), \tag{4}$$

where $\sin\beta = \sqrt{\frac{M}{N}}$. In the J-th iteration, one of the marked states is obtained with zero failure rate as explained and demonstrated in [5]:

$$J \geq \text{floor}\left(\frac{\frac{\pi}{2}-\beta}{\beta}\right)+1. \tag{5}$$

The exact values for β, ϕ and J can be computed using the number of solutions M and the database size N. The Oracle can recognize the solutions of the searching problem and apply a phase rotation to them. If only one solution is needed, the Oracle can be described as the diagonal matrix having only one $e^{i\Phi}$:

$$O = e^{i\phi}|\upsilon\rangle\langle\upsilon| + \sum_{\tau=0;\tau\neq\upsilon}^{2^n-1} |\tau\rangle\langle\tau|, \tag{6}$$

where ν is the position of $e^{i\phi}$ in the diagonal matrix. If M solutions are needed, the Oracle becomes:

$$O = e^{i\phi}\sum_{\tau=1}^{M}|\upsilon_\tau\rangle\langle\upsilon_\tau| + \sum_{\tau=0;\tau\neq\upsilon}^{2^n-1} |\tau\rangle\langle\tau|, \tag{7}$$

where the number of $e^{i\phi}$ is M and the set of $e^{i\phi}$ positions is $V=\{v_1, v_2, \ldots, v_M\}$. In conclusion, an Oracle marking multiple quantum states is ultimately composed of several Oracles marking just one quantum state. The Qiskit quantum circuit used for the implementation of QUMMSA is presented in Fig. 3.

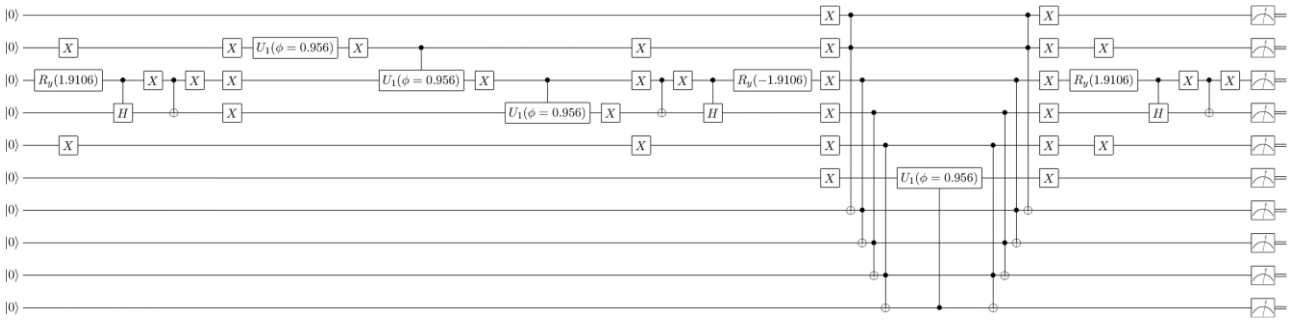

Fig. 3 – QUMMSA circuit implemented in Qiskit,

4. EVALUATION

We start by measuring the ATR and OWD for each link of the graph. In order to simplify the experiment, we assume that the links are symmetrical, i.e. the ATR and OWD are equal in both ways of the link. The measured values are presented in Table 1. Note that they were obtained in our previous work and we use them to validate the quantum solution by comparing the results to the ones obtained in the legacy approach. The composite metrics corresponding to the three different paths between node 1 and node 4 are computed using $K_0 = 10^9$ bps and $K_1 = 10^{-5}$ s. Since the value of the ATR is the same for all links, it is very easy to compute the first term of the CM: 12.18778. If the ATR for the links is different, we can compute the ATR of each path, using either QUMMSA, in a quantum approach, or offline, using legacy methods. For the second term however, we need to add the OWDs for all the links composing a certain path. Moreover, to compute the CM we also need to add the term corresponding to the ATR, computed earlier.

This is where we take advantage of the Quantum Phase Estimation Algorithm described in the previous paragraphs. First, we need to map the scaled OWDs of each link and the ATR of the entire path into phases, making sure that the CM doesn't exceed 2π. For example, the CM for path 1–>2–>4 translates to:

$$CM_{1\to2\to4} = \frac{10^9}{82,049,400} + \frac{0.000030}{10^{-5}} + \frac{0.000039}{10^{-5}} = 12.18778 + 3 + 3.9\,. \tag{8}$$

First, each term is rounded to the nearest integer and then scaled by multiplying with a factor of π/32. This value was chosen taking into account that we represent the phases on 6 qubits. Thus, the maximum value that can be represented is 63, corresponding to 2π. As such, the least-significant qubit has a value of π/32. Note that both rounding and quantization steps introduce errors. A solution is increasing the number of qubits for a more precise phase representation. In our case, the three terms in (8) round to 12, 3 and 4 and then are scaled into 12π/32, 3π/32 and 4π/32. Similarly, all other ATRs and OWDs are obtained:

Table 1

Values of the ATR, OWD and corresponding phases for each link

Link	ATR [kbps]	OWD [s]	ATR [rad]	OWD [rad]
1–>2	82,049.4	0.000030	12π/32	3π/32
2–>4	82,049.4	0.000039	12π/32	4π/32
1–>4	82,049.4	0.000144	12π/32	14π/32
1–>3	82,049.4	0.000131	12π/32	13π/32
3–>4	82,049.4	0.000044	12π/32	4π/32

In order to obtain the CMs for all three paths, we construct four unitary matrices U_1, U_2, U_3 and U_4. These matrices are diagonal and the four elements of the main diagonal correspond to the phases associated to the OWDs of the links. The difference from other approaches [10] is that we also have to include the ATR in these matrices. There are two options here: we can either use element $[U_j]_{jj}$ since there is no link from node j to itself or we can use the other elements of the matrix which are associated to non-existing links e.g. 2–>3 and 3–>2. Herein, we used the second approach. Next, we computed $U = U_1 \otimes U_2 \otimes U_3 \otimes U_4$. U is also a diagonal matrix which has 256 eigenstates on the main diagonal. Out of these 256 elements, we are only interested in the 3 associated to the 3 desired paths. We constructed 3 different circuits in Qiskit, one for each eigenstate corresponding to a path. The 8 bits of an eigenstate are grouped into four 2-bit groups and each group represents the position of an element in U_j : the first group corresponds to U_1, the second to U_2 and so on. By carefully setting the eigenstate, we can select the combinations of OWD and ATR involved in certain path and thus compute the CM, as seen in Fig. 4.

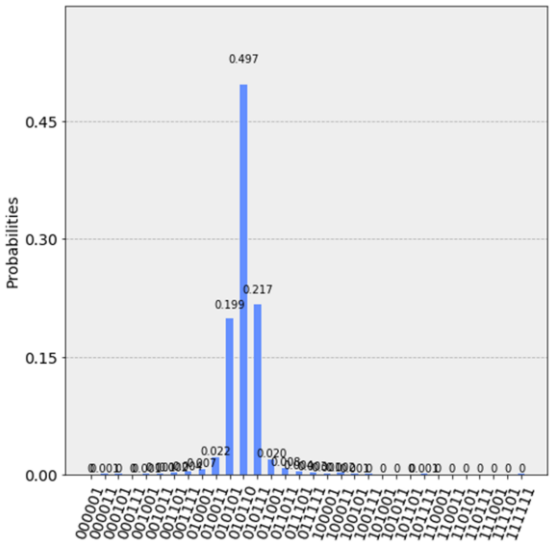

Fig. 4 – Quantum phase estimation for Composite Metric of path 1–>2–>4.

Out of 4 096 simulations, for the Composite Metric of path 1–>2–>4 state 010110 was obtained with the highest probability. The eigenstates, the theoretical and experimental values of the CMs for the 3 paths are presented in Table 2.

Table 2

Theoretical and experimental values for the composite metrics of the paths

Path	Eigenstate	Theoretical	Experimental
1–>4	00010100	011010	011010
1–>2–>4	01110111	010011	010110
1–>3–>4	00100010	011101	011110

As described in [10], in order to accurately obtain the phase up to n bits with a probability of success at least 1–ε, we need to use t qubits:

$$t = n + \log\left(2 + \frac{1}{2\varepsilon}\right). \tag{9}$$

In our case, we used $t = 6$ qubits. From Table 2 it is easy to observe that for the second and for the third path 2 bits are erroneous. One way of increasing the accuracy is by increasing the number of phase estimation qubits t. This however does not eliminate the error entirely and even one single wrong bit (e.g. the MSB) is enough to compromise the result. Next, we need to find the minimum of the three CMs. For this, we constructed another Qiskit circuit implementing QUMMSA. The first step was to encode the three values into a 6-qubit unsorted database. The rest of the 61 values of the database are initialized to 0. We then randomly selected a reference value from the database: $011100_2 = 28$. In the first iteration of the algorithm we should obtain $M = 28+1$ solutions less than or equal to this reference value. However, since there are only 3 elements of interests in the database from now on we will only refer to them and ignore all other solutions equal to zero. Using $M = 29$ and $N = 64$ from (4) and (5) we obtain $\beta = 0.738$, $J \geq 2$ and $\phi = 0.956$. The two elements less than 28 are marked having the highest probability (Fig. 5a). For the second iteration we chose another reference value from the two solutions obtained earlier: $010110_2 = 22$. M becomes 23, N is still 64 while $\beta = 0.643$, $J \geq 2$ and $\phi = 1.086$. Using these parameters, only one solution (the minimum value in the database) was obtained, $010110_2 = 22$, which is marked correctly as having the highest probability (see Fig. 5b).

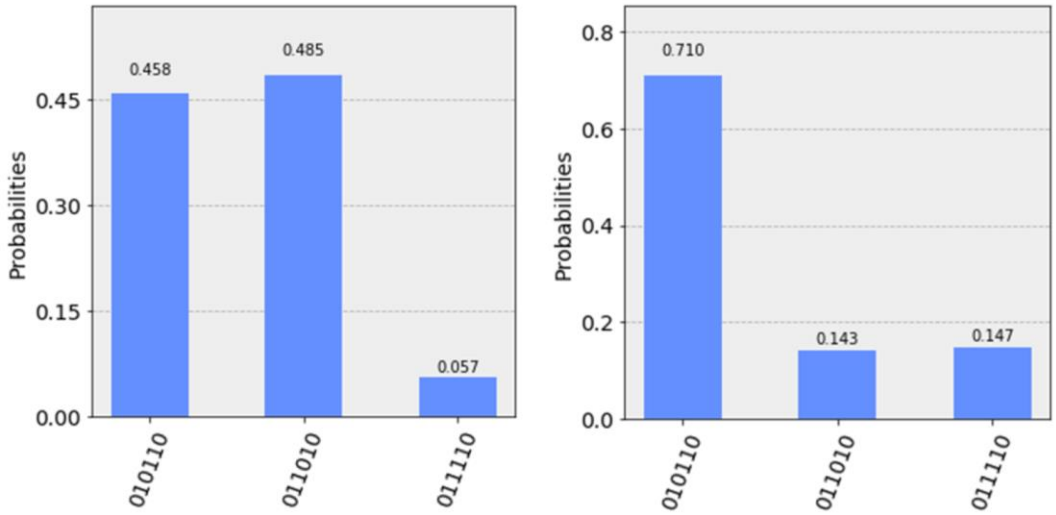

Fig. 5 – Results: a) after the first iteration, respectively (b) after the second and last iteration of QUMMSA.

The algorithm now stops and the best path is the one corresponding to eigenstate 01110111, namely path 1–>2–>4. This result is coherent with the one obtained using the classical non-quantum method. Even though our example is admittedly simplistic, the same approach can be used for a larger database, with the same performance, as shown in [6]. Moreover, we designed our approach as a proof-of-concept. As such, we only executed the algorithms on the local simulator and we were not particularly interested in optimizing simulation time, number of executions, etc. The code for this project is available at [18].

5. CONCLUSIONS

This paper presents a quantum implementation of the Modified Dijkstra's algorithm used for finding the best path between a source and a destination node in a connected graph. The composite metric for each path is computed using the ATRs and OWDs of the links. First, quantum phase estimation is used to compute this metric. The novelty herein consists in the encoding of both ATR and OWD in the controlled

unitary matrices. Next, an optimized version of Grover quantum search algorithm (QUMMSA) is used to find the path with the lowest composite metric. Experimental results obtained from simulations in Qiskit validate our approach: the best path is the same as the one computed using the legacy approach described in [3] for the same input data. However, improvements are still possible, by increasing the number of qubits used for phase estimation and extending the solution for all non-symmetrical graphs, regardless of the number of nodes/links. For future work, we plan on adding multi-path routing algorithms and performing experiments on physical quantum platforms.

REFERENCES

1. *Cisco Annual Internet Report (2018–2023) White Paper*, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
2. P. BOTSINIS, D. ALANIS, Z. BABAR, H.V. NGUYEN, D. CHANDRA, S.X. NG, L. HANZO, *Quantum search algorithms for wireless communications*, IEEE Communications Surveys & Tutorials, **21**, *2*, pp. 1209–1242, 2019.
3. A.B. RUS, V. DOBROTA, A. VEDINAS, G. BOANEA, M. BARABAS, *Modified Dijkstra's algorithm with cross-layer QoS*, Acta Technica Napocensis, Electronics and Telecommunications, **51**, *3*, pp. 75–80, 2010.
4. P. COLES et al., *Quantum algorithm implementations for beginners*, arXiv: 1804.03719, 2018.
5. G.L. LONG, *Grover Algorithm with zero theoretical failure rate*, arXiv: quant-ph/0106071, 2001.
6. Y. CHEN, S. WEI, X. GAO, C. WANG, J. WU, H. GUO, *An optimized quantum maximum or minimum searching algorithm and its circuits*, arXiv: 1908.07943, 2019.
7. C. DURR, P. HOYER, *A quantum algorithm for finding the minimum*, arXiv: quant-ph/9607014, 1996.
8. D.J. MOYLETT, N. LINDEN, A. MONTANARO, *Quantum speedup of the Travelling Salesman Problem for bounded-degree graphs*, arXiv: 1612.06203, 2016, Physical Review A, **95**, *3*, p. 032323, 2017.
9. E.A. MARKEVICH, A.S. TRUSHECHKIN, *Quantum branch-and-bound algorithm and its application to the travelling salesman problem*, J Math Sci, **241**, pp. 168–184, 2019, doi: 10.1007/s10958-019-04415-6.
10. K. SRINIVASAN, S. SATYAJIT, B.K. BEHERA, P.K. PANIGRAHI, *Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience*, arXiv: 1805.10928, 2018.
11. M. RADZIHOVSKY, J. MURPHY, M. SWOFFORD, *A QAOA solution to the traveling salesman problem using pyQuil*, Report, Stanford University, 2019.
12. R.H. WARREN, *Solving the traveling salesman problem on a quantum annealer*, SN Appl. Sci. 2, **75**, 2020, doi: 10.1007/s42452-019-1829-x.
13. T.D. KIEU, *The travelling salesman problem and adiabatic quantum computation: an algorithm*, arXiv:1801.07859, 2018, Quantum Information Processing, **18**, *3*, p. 90, 2019.
14. I.A. IVANCIU, *Active measurements for routing in cloud-based networks*, PhD Thesis, Technical University of Cluj-Napoca, Romania, 2016, https://rei.gov.ro//teza-doctorat, Cod dosar: F-CA-4044/03.01.2017.
15. A. TAUT, I.A. IVANCIU, E. LUCHIAN, V. DOBROTA, *Active measurement of the latency in cloud-based networks*, Acta Technica Napocensis, Electronics and Telecommunications, **58**, *1*, pp. 22–30, 2017.
16. *IBM Quantum Experience Systems and Simulators*, https://quantum-computing.ibm.com/systems
17. *ADVANTAGE The first and only quantum computer built for business*, https://www.dwavesys.com/d-wave-two™-system
18. *Quantum-Implementation-of-the-Modified-Dijkstra-s-Routing-Algorithm*, https://github.com/robertbotez/Quantum-Implementation-of-the-Modified-Dijkstra-s-Routing-Algorithm

Received September 20, 2020