Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm

3 authors:

Utkarsh Azad
International Institute of Information Technology, Hyderabad
6 PUBLICATIONS   **0** CITATIONS

SEE PROFILE

Bikash K. Behera
Bikash's Quantum (OPC) Pvt. Ltd.
132 PUBLICATIONS   **440** CITATIONS

SEE PROFILE

Prasanta K. Panigrahi
Indian Institute of Science Education and Research Kolkata
577 PUBLICATIONS   **4,548** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Image Enhancement View project

Quantum Games View project

Solving Vehicle Routing Problem
Using Quantum Approximate Optimization Algorithm

Utkarsh,[1, *] Bikash K. Behera,[2, 3, †] and Prasanta K. Panigrahi[3, ‡]

[1]*Center for Computational Natural Sciences and Bioinformatics,*
International Institute of Information Technology Hyderabad, Hyderabad 500032, Telangana, India
[2]*Bikash's Quantum (OPC) Pvt. Ltd.,*
Balindi, Mohanpur 741246, Nadia, West Bengal, India
[3]*Department of Physical Sciences,*
Indian Institute of Science Education and Research Kolkata, Mohanpur 741246, West Bengal, India

In this paper, we describe the usage of Quantum Approximate Optimization Algorithm (QAOA), which is a quantum-classical heuristic, to solve a combinatorial optimization and integer programming task known as Vehicle Routing Problem (VRP). We outline the Ising formulation for VRP and present a detailed procedure to solve VRP by minimizing its simulated Ising Hamiltonian using IBM Qiskit platform. Here, we attempt to find solutions for the VRP problems: $(4, 2)$, $(5, 2)$, and $(5, 3)$, where each (n, k) represents a VRP problem with n locations and k vehicles. We find that the performance of QAOA is largely limited by the problem instance.

I. INTRODUCTION

A majority of the real world applications involve mapping the task at hand to an optimization problem, whose solutions are either fully known or can be approximated by relaxing some of its constraints. In general, the increment in problem size or augmentation of additional constraints can increase the hardness of the problem. This means that the computational resources required in solving them scale-up exponentially, making them computationally intractable. Hence, researchers worldwide have been working on developing efficient tools and techniques for solving these problems efficiently with the available computing power.

With the recent development of quantum processors by IBM[1], Rigetti[2], Google[3] etc, various proposals have been made to use them to find solutions for optimization problems. In general, quantum computing devices are supposed to have a computational advantage over classical processors by using quantum resources such as superposition and entanglement. However, the computational capabilities of these current generation quantum processors also known as Noisy Intermediate-Scale Quantum (NISQ)[4] devices, are considerably restricted due to their intermediate size (in terms of qubits count), limited connectivity, imperfect qubit-control, short coherence time and minimal error-correction. Hence, they are only able to run algorithms with limited circuit depth. The Quantum Approximate Optimization Algorithm (QAOA)[5] is one of such algorithm which belongs to the class of quantum-classical hybrid variational algorithms. It can be thought of as a coarsely trotterized adiabatic time evolution in p steps to $\left| \psi_{GS}^{H_c} \right\rangle$ i.e., the ground state of a Hamiltonian H_c which encodes the problem from $\left| \psi_{GS}^{H_m} \right\rangle$ i.e., the ground state of the Hamiltonian H_m which is known and easier to prepare. Thus, it can be used to solve combinatorial optimization problems mapped to the minimization of an Ising Hamiltonian on near-term devices. As the Ising problem is itself NP-hard, QAOA is an expected candidate for demonstrating quantum supremacy. Here, in this paper, we use QAOA to solve the Vehicle Routing Problem (VRP), which is an NP-hard combinatorial optimization problem[6].

Structure – In Section II, we discuss VRP and present its Ising formulation in Section III. Then, in Section IV, we explain QAOA in detail and present our simulation results in Section V. Finally, in Section VI, we discuss the performance and limitations of using QAOA to solve combinatorial optimizations problem in general.

II. VEHICLE ROUTING PROBLEM

Vehicle Routing Problem is an NP-hard combinatorial optimization problem. Any problem instance (n, k) of VRP involves k vehicles, and $n - 1$ locations (other than the depot D). Its solution is the set of routes in which all of the k vehicles begin and end in the D, such that each location is visited exactly once. The optimal route is the one in which the total distance travelled by k vehicles is least. In a way, this problem is a generalization of the classic Travelling Salesman Problem[7], where now a group of k salesmen have to collectively serve $n - 1$ locations such that each location is served exactly once.

In most real world applications, the VRP problem[8] is generally augmented by constraints, such as vehicle capacity or limited coverage-time. However, here we only focus on showing how to solve the most basic version of VRP without any of these additional constraints.

III. ISING FORMULATION OF VRP

To solve a problem instance (n, k) of VRP using QAOA, we first need to map it to the minimization of an Ising Hamiltonian H_c[9]. We do this by first finding the H_c, which encodes the given problem instance.

Let x_{ij} be the binary decision variable which has the value 1 if there exists an edge from i to j with weight $w_{ij} > 0$, else it is 0. To represent a solution to VRP problem, there will be $n \times (n-1)$ decision variables. Next, we for every edge $i \rightarrow j$ we define two sets $source[i]$ and $target[j]$. The set $source[i]$ will contain the nodes js to which node i sends an edge. Similarly, the set $target[j]$ will contain all the nodes i which send an edge to node j. Hence, the VRP can be forumalted as

$$VRP(n,k) = min_{\{x_{ij}\}_{i \rightarrow j} \in \{0,1\}} \sum_{i \rightarrow j} w_{ij} x_{ij} \qquad (1)$$

Subjected to the following constraints:

$$\sum_{j \in source[i]} x_{ij} = 1, \forall i \in \{1, \ldots, n-1\} \qquad (2)$$

$$\sum_{j \in target[i]} x_{ji} = 1, \forall i \in \{1, \ldots, n-1\} \qquad (3)$$

$$\sum_{j \in source[0]} x_{0j} = k \qquad (4)$$

$$\sum_{j \in target[0]} x_{j0} = k \qquad (5)$$

Here, Eqs. (2), (3) impose the node-visiting constraint so that each node is visited exactly once. Also, the Eqs. (4), (5) impose the constraint to enforce that all the vehicles begin from and return back to depot D, i.e., the node 0. Now, using Eqs. (1-5), the energy functional H_{VRP} of the above problem can be written as:

$$H_{VRP} = H_A + H_B + H_C + H_D + H_E \qquad (6)$$

$$H_A = \sum_{i \rightarrow j} w_{ij} x_{ij} \qquad (7)$$

$$H_B = A \sum_{i \in 1, \ldots, n-1} \left(1 - \sum_{j \in source[i]} x_{ij}\right)^2 \qquad (8)$$

$$H_C = A \sum_{i \in 1, \ldots, n-1} \left(1 - \sum_{j \in target[i]} x_{ji}\right)^2 \qquad (9)$$

$$H_D = A\left(k - \sum_{j \in source[0]} x_{0j}\right)^2 \qquad (10)$$

$$H_E = A\left(k - \sum_{j \in target[0]} x_{j0}\right)^2 \qquad (11)$$

Here, $A > 0$ is a constant. Next, for $VRP(n,k)$, we can represent all the decision variables x_{ij} using the following vector $\vec{x}$:

$$\vec{\mathbf{x}} = [x_{(0,1)}, x_{(0,2)}, \ldots x_{(1,0)}, x_{(1,2)}, \ldots x_{(n-1,n-2)}]^{\mathbf{T}} \qquad (12)$$

Using $\vec{x}$, we also define the following vectors for every node i: $\vec{z}_{S[i]}$ and $\vec{z}_{T[i]}$. The first vector $\vec{z}_{S[i]}$ is $\vec{x}$ with $x_{ij} = 1$, $x_{kj} = 0$ if $k \neq j$, $\forall j, k \in \{0, \ldots, n-1\}$. Similarly, the second vector $\vec{z}_{T[i]}$ is $\vec{x}$ with $x_{ji} = 1$, $x_{jk} = 0$ if $k \neq i$, $\forall j, k \in \{0, \ldots, n-1\}$.

From Eq. (12), we can represent minimization of H_{VRP} as a quadratic unconstrained binary optimization (QUBO) problem which is isomorphic to the ising problem. In general, a QUBO problem for a graph $G = (N, V)$ may be defined as:

$$f(x)_{QUBO} = min_{x \in \{0,1\}^{(N \times V)}} \ x^T Q x + g^T x + c \qquad (13)$$

Here the quadratic coefficient Q represents the edge weight i.e., coupling or interaction between two nodes, the linear coefficient g represents the node weight i.e., contribution from individual nodes, and the term c is a constant offset. Therefore, in the QUBO formulation[10] of H_{VRP} given in Eq. (6), these coefficients will be:

$$Q = A\Big[[z_{T[0]}, \ldots, z_{T[n-1]}]^T [z_{T[0]}, \ldots, z_{T[n-1]}]$$
$$+ (\mathbb{I}_n \otimes \mathbb{J}_{(n-1,n-1)})\Big] \qquad (14)$$

$$g = W - 2Ak\big((e_0 \otimes \mathbb{J}_{n-1}) + [z_{T[0]}]^T\big)$$
$$+ 2A(\mathbb{J}_n \otimes \mathbb{J}_{n-1}) \qquad (15)$$

$$c = 2A(n-1) + 2Ak^2 \qquad (16)$$

Here, $\mathbb{J}$ is all-ones matrix, $\mathbb{I}$ is identity matrix, W is the weight matrix, and the e_0 is the basis vector $[1, 0, \ldots, 0]^T$. From this, to construct Ising Hamiltonian for VRP we expand Eq. (13) by using Eqs. (14-16) and rewrite all the binary variables $x_{ij} \in \{0, 1\}$ using spin variables $s_{ij} \in \{-1, 1\}$[10].

$$x_{ij} = \frac{s_{ij} + 1}{2} \qquad (17)$$

By grouping together similar terms from the expansion, one can express $\hat{H}_{ISING}$ for $VRP(n,k)$ as:

$$H_{ISING} = -\sum_i \sum_{j<i} J_{ij} s_i s_j - \sum_i h_i s_i + d \qquad (18)$$

$$J_{ij} = -\frac{Q_{ij}}{2}, \quad \forall i < j \qquad (19)$$

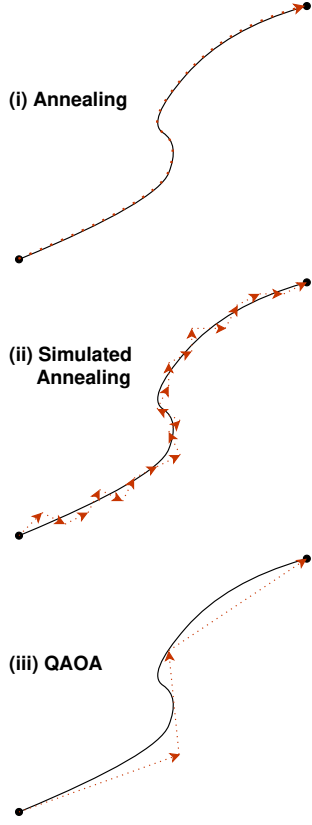$$h_i = \frac{g_i}{2} + \sum_j \frac{Q_{ij}}{4} + \sum_j \frac{Q_{ji}}{4} \qquad (20)$$

FIG. 1. Consider an adiabatic time evolution path (black) in the state space. Conceptually, quantum annealing (top) follows this adiabatic time evolution path. Simulated annealing (middle) follows this path in discrete steps i.e. it follows a trotterized adiabatic time evolution path. QAOA follows this path in p steps, i.e. it follows a coarsely trotterized adiabatic time evolution.[11]

$$d = c + \sum_i \frac{g_i}{2} + \sum_i \sum_j \frac{Q_{ij}}{4} \qquad (21)$$

Replacing s_i with σ_i^z i.e. the Pauli-Z operator acting on i^{th} qubit, gives the quantum mechanical description of $\hat{H}_{ISING}$ which.

IV. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

Adiabatic quantum computation (AQC)[12] was the first quantum computation model to be used for solving hard combinatorial optimization problems. Unlike the gate-based quantum computation model, it was based on adiabatic theorem from quantum mechanics. In this model, to perform any computation we need two Hamiltonians called $\hat{H}_{mixer}$ and $\hat{H}_{cost}$. Amongst them, the ground state of $\hat{H}_{mixer}$ should be an easily preparable state such as $|+\rangle^{\otimes N}$ and ground state of $\hat{H}_{cost}$ encode the solution to our problem. Both Hamiltonians $\hat{H}_{mixer}$ and $\hat{H}_{cost}$, should be local, i.e. they only involve terms for interac-

tions between a constant number of particles. The instantaneous Hamiltonian $\hat{H}(t)$ for the system is:

$$\hat{H}(t) = (1 - t)\hat{H}_{mixer} + t\hat{H}_{cost} \qquad (22)$$

In AQC, ΔE, i.e., the difference between the ground state and first excited state energy of $\hat{H}(t)$ bounds the step size one can take to follow adiabatic pathway[13]. Hence, the computation time to solve any problem rises exponentially as ΔE becomes infinitesimally small. This limits its capability to solve a certain instance of hard optimization problems.

In Quantum Approximate Optimization Algorithm (QAOA), we eliminate this restriction on the step size. Instead, whole of the adiabatic pathway is discretized in some p steps, where p represents precision (Fig. 1). To do this, we trotterize the unitary into p steps using the parameters $\{\beta, \gamma\}$ as follows:

$$\begin{aligned} U = U(\hat{H}_{mixer}, \beta_0)U(\hat{H}_{cost}, \gamma_0) \dots \\ U(\hat{H}_{mixer}, \beta_p)U(\hat{H}_{cost}, \gamma_p) \end{aligned} \qquad (23)$$

In gate-model quantum computation this means[13] that starting from some initial product state $|\psi_{GS}^{HM}\rangle$, we apply a parameterized gate sequence to produce the state $|\psi^{HC}\rangle$. For some optimal value of the parameters: $\{\beta^*, \gamma^*\}$, this is the ground state of $\hat{H}_{cost}$. The parameters $\{\beta, \gamma\}$ are provided by a classical processor, and also optimized by a classical optimization routine based on the result of energy measurement for final state $|\psi^{HC}\rangle$. Hence, QAOA belongs to the class of hybrid quantum-classical variational algorithms.

V. SIMULATIONS AND RESULTS

We have executed QAOA using IBM Qiskit to solve VRP for three problem instances: $(4, 2)$, $(5, 2)$, and $(5, 3)$, where each (n, k) represents a problem with n locations and k vehicles with a distance matrix D representing the squared euclidean distances between locations. One needs $N = n \times (n - 1)$ qubits to encode the problem instance, i.e., state of each qubit represents the possibility of an edge between two nodes. The mixing Hamiltonian $\hat{H}_{mixer}$ and the cost Hamiltonian $\hat{H}_{cost}$ for this problem are:

$$\hat{H}_{mixer} = -\sum_{i=0}^{n \times (n-1)-1} \sigma_i^x \qquad (24)$$

$$\hat{H}_{cost} = -\sum_i \sum_{j<i} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z - d \qquad (25)$$
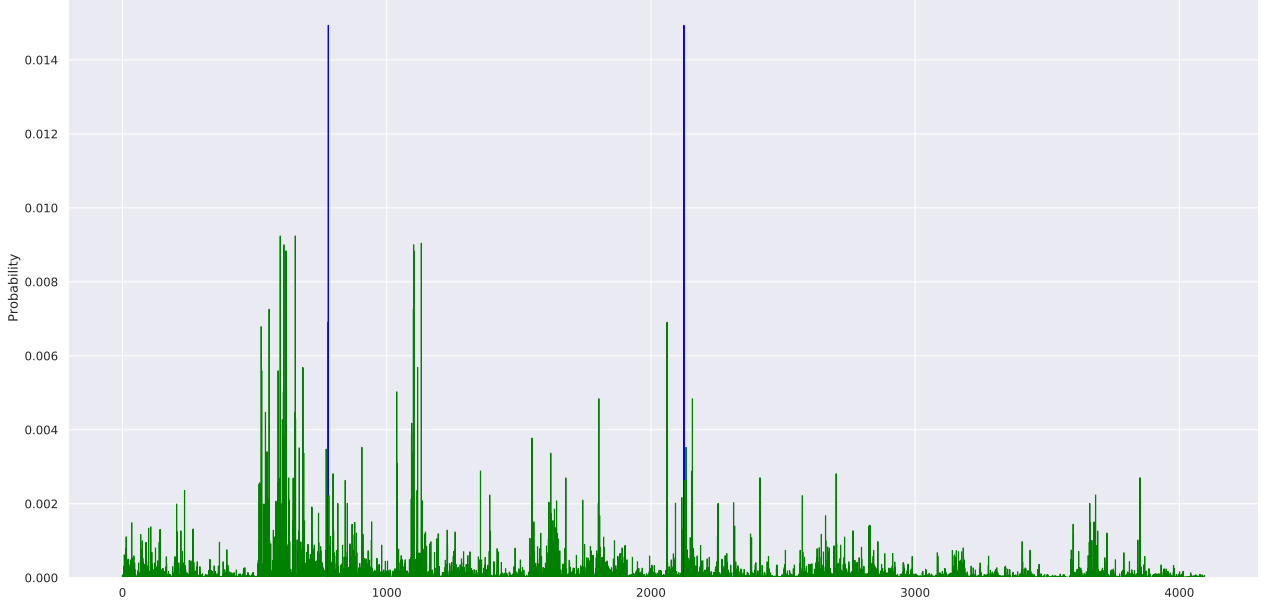
FIG. 2. Probability distribution for the problem instance $(4, 2)$ for $p = 12$. Indices of optimal states (blue) are: 779 and 2125.

In $\hat{H}_{cost}$, we have $n \times (n - 1) \times (n - 2)$ terms of J_{ij}, $n \times (n-1)$ terms of h_i, and d is an offset. In each of the following cases, we begin with the state $|+\rangle^{n \otimes (n-1)}$, which is the ground state of $\hat{H}_{mixer}$ given in Eq. (24). This state is prepared by applying Hadamard on all qubits initialized to $|0\rangle$. From Eq. (23), this state is evolved as:

$$|\beta, \gamma\rangle = e^{-i\hat{H}_{mixer}\beta_p} e^{-i\hat{H}_{cost}\gamma_p} \cdots$$
$$e^{-i\hat{H}_{mixer}\beta_0} e^{-i\hat{H}_{cost}\gamma_0} |+\rangle^{n \otimes (n-1)} \tag{26}$$

For the evolved state $|\beta, \gamma\rangle$, we calculate energy E by measuring the expectation value of $\hat{H}_{cost}$ as:

$$E = \langle \beta, \gamma | \hat{H}_{cost} | \beta, \gamma \rangle \tag{27}$$

Running a classical optimization routine on Eq. (27), we get the optimal value of $\{\beta, \gamma\}$ as $\{\beta^*, \gamma^*\}$. To get the final result we measure the state $|\beta^*, \gamma^*\rangle$. As shown in Fig. (2), the state can collapse to any of the $2^{n \times (n-1)}$ possibilities. To visualize the solution, we represent the index of the collapsed state as $2^{n \times (n-1)}$ length bit string which represents flattened version of the adjacency matrix of the graph.

A. Experiment 1

In the first experiment, we have solved the problem instance $(4, 2)$, described by the following distance matrix:
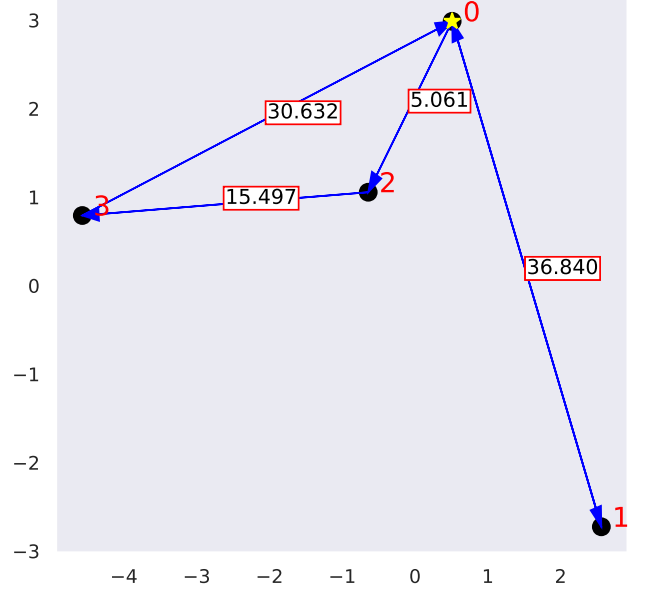

FIG. 3. Visualization of the solution state indexed $779 =$ "110100001100". The cost is: $C_1 = 30.632 + 15.497 + 5.061 + 2 \times 36.840 = 124.871$. Here, the node with yellow star denotes the depot, or the origin.

$$D_1 = \begin{bmatrix} 0. & 36.84 & 5.06 & 30.63 \\ 36.84 & 0. & 24.55 & 63.22 \\ 5.06 & 24.55 & 0. & 15.50 \\ 30.63 & 63.22 & 15.50 & 0 \end{bmatrix} \tag{28}$$

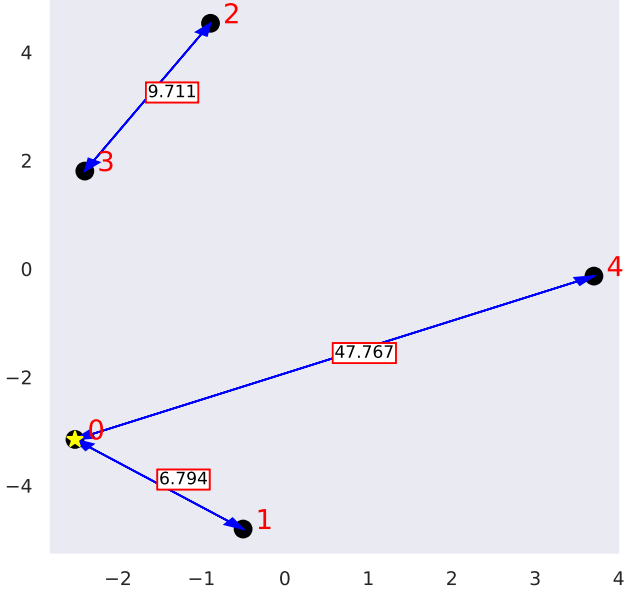To encode the problem, we used $N = 4 \times 3 = 12$

FIG. 4. Visualization of the QAOA state indexed $623144 =$ "10011000001000101000". The cost is: $C_2 = 2 \times 9.711 + 2 \times 47.767 + 2 \times 6.794 = 128.545$. Here, the node with yellow star denotes the depot, or the origin.

FIG. 5. Visualization of the solution state indexed $688424 =$ "10101000000100101000". The cost is: $C_3 = 24.557 + 9.711 + 42.887 + 47.767 + 2 \times 6.794 = 138.511$. Here, the node with yellow star denotes the depot, or the origin.

qubits. Using COBYLA optimizer, for $p \geq 12$ we were able to get the correct solution with sufficient probability. In Fig. (2), we have shown the probability distribution of our result for $p = 12$. States corresponding to indexes 779 and 2125 are equiprobable and solution states. We present the visualization of the state indexed $779 = [1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0]$ in Fig. (3) using the following adjacency matrix A_1:

$$A_1 = \begin{bmatrix} X & 1 & 1 & 0 \\ 1 & X & 0 & 0 \\ 0 & 0 & X & 1 \\ 1 & 0 & 0 & X \end{bmatrix} \quad (29)$$

The cost in both cases come out to be $C_1 = 30.632 + 15.497 + 5.061 + 2 \times 36.840 = 124.871$.

B. Experiment 2

In the second experiment, we have solved the problem instance $(5, 2)$, described by the following distance matrix:

$$D_2 = \begin{bmatrix} 0 & 6.794 & 61.653 & 24.557 & 47.767 \\ 6.794 & 0 & 87.312 & 47.262 & 39.477 \\ 61.653 & 87.312 & 0. & 9.711 & 42.887 \\ 24.557 & 47.262 & 9.711 & 0 & 40.98 \\ 47.767 & 39.477 & 42.887 & 40.98 & 0 \end{bmatrix} \quad (30)$$
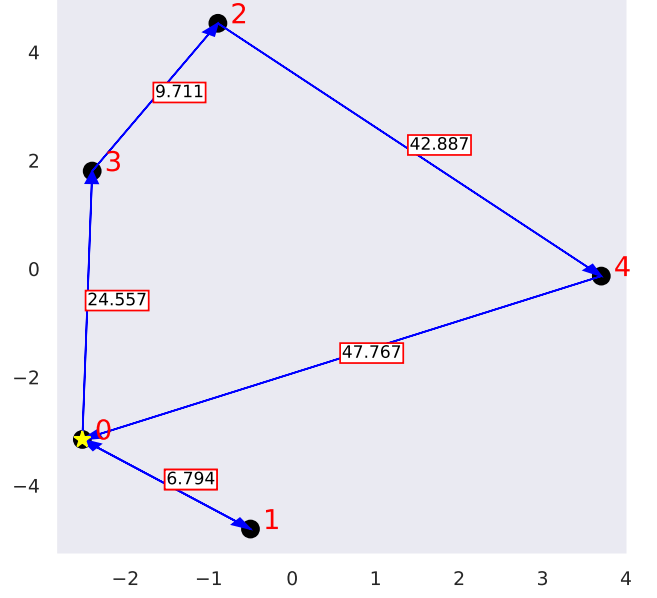
To encode the problem, we used $N = 5 \times 4 = 20$ qubits. We tried COBYLA, NELDER MEAD, and L-BFGS-B optimizers, for different values of p ranging from 6 to 40. Fig. (4) represents the visualization of the state indexed: $623144 = [1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0]$, i.e., the state we got, using the following adjacency matrix A_2:

$$A_2 = \begin{bmatrix} X & 1 & 0 & 0 & 1 \\ 1 & X & 0 & 0 & 0 \\ 0 & 0 & X & 1 & 0 \\ 0 & 0 & 1 & X & 0 \\ 1 & 0 & 0 & 0 & X \end{bmatrix} \quad (31)$$

The cost in this case comes out to be $C_2 = 2 \times 9.711 + 2 \times 47.767 + 2 \times 6.794 = 128.545$.

Where as, Fig. (5) represents the visualization of the state indexed: $688424 = [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0]$, i.e., the optimal state we should have got, using the following adjacency matrix A_3:

$$A_3 = \begin{bmatrix} X & 1 & 0 & 1 & 0 \\ 1 & X & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 1 \\ 0 & 0 & 1 & X & 0 \\ 1 & 0 & 0 & 0 & X \end{bmatrix} \quad (32)$$

The cost in this case comes out to be $C_3 = 24.557 + 9.711 + 42.887 + 47.767 + 2 \times 6.794 = 138.511$. Therefore, as shown in Fig. (4), the solution we got was not the optimal one i.e., given in Fig. (5).
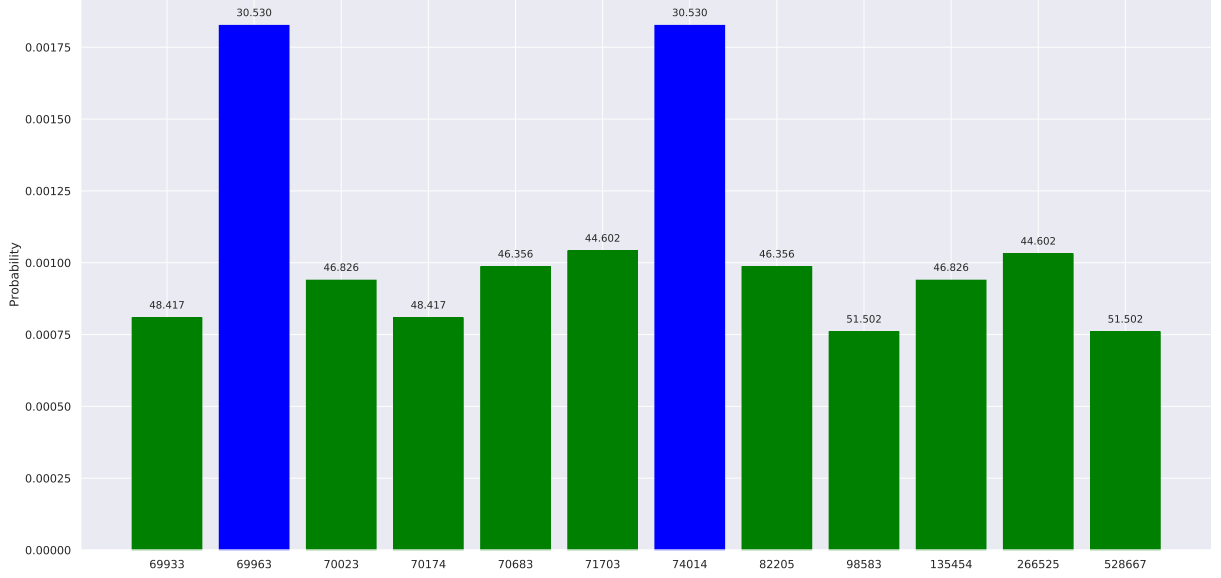
FIG. 6. Probability distribution for top-12 feasible solutions of the problem instance $(5, 3)$ for $p = 24$. Indices of optimal states (blue) are: 69963 and 74014. Costs corresponding to each state are written at the top of the bars.
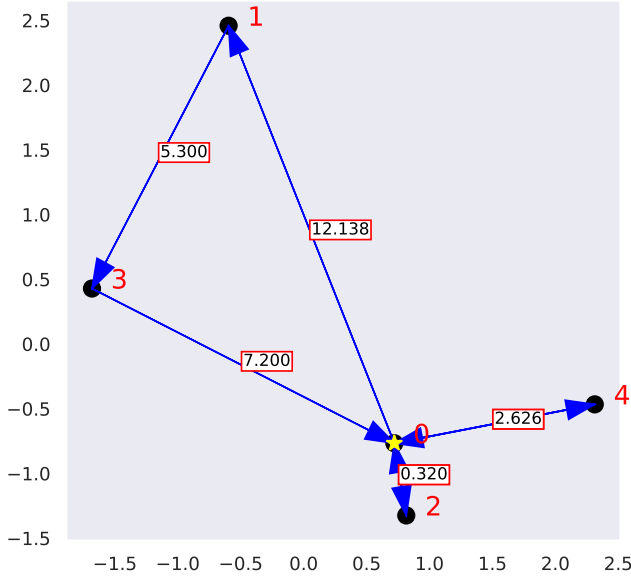

FIG. 7. Visualization of the solution state indexed $69963 =$ '10001000101001011'. The cost is: $C_4 = 12.138 + 5.300 + 7.200 + 2 \times 2.626 + 2 \times 0.320 = 30.530$. Here, the node with yellow star denotes the depot, or the origin.

C.   Experiment 3

In the third experiment, we have solved the problem instance $(5, 3)$, described by the following distance matrix:

$$D_4 = \begin{bmatrix} 0. & 12.138 & 0.32 & 7.2 & 2.626 \\ 12.138 & 0. & 16.307 & 5.3 & 17.021 \\ 0.32 & 16.307 & 0. & 9.309 & 2.98 \\ 7.2 & 5.3 & 9.309 & 0. & 16.759 \\ 2.626 & 17.021 & 2.98 & 16.759 & 0. \end{bmatrix} \quad (33)$$

To encode the problem, we used $N = 5 \times 4 = 20$ qubits. Using COBYLA optimizer, for $p \geq 24$ we were able to get the correct solution with sufficient probability. In Fig. (6), we have shown the probability distribution of our result for $p = 24$ for top 12 feasible solutions. States corresponding to indexes 69963 and 74014 are equiprobable and solution states. We present the visualization of the state $69963 = [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1]$ in Fig. (7) using the following adjacency matrix A_4:

$$A_4 = \begin{bmatrix} X & 1 & 1 & 0 & 1 \\ 0 & X & 0 & 1 & 0 \\ 1 & 0 & X & 0 & 0 \\ 1 & 0 & 0 & X & 0 \\ 1 & 0 & 0 & 0 & X \end{bmatrix} \quad (34)$$

The cost in both cases come out to be $C_4 = 12.138 + 5.300 + 7.200 + 2 \times 2.626 + 2 \times 0.320 = 30.530$.

VI.   DISCUSSIONS

Here, we have used Quantum Approximate Optimization Algorithm to solve the Vehicle Routing Problem. In past, QAOA has been widely used in solving various combinatorial hard optimization problems[14–16]. However, looking at the results from Exp. ($V\,A$), ($V\,B$), and

$(V\,C)$, we conclude that in general, for a finite value of p, there is no guarantee that the solution achieved by QAOA corresponds to the most optimal solution of the original combinatorial optimization problem[17]. This is because, in QAOA instead of the following the adiabatic time evolution path, we try to guess it using p steps.

Moreover, in the previous studies[17–19], it has been shown that while running QAOA on near term quantum processors, noise-based errors affect both the fidelity of state: $|\beta, \gamma\rangle$, prepared by a quantum routine, and the minimized expectation value of $\hat{H}_{cost}$, i.e., $\langle \hat{H}_{cost}\rangle$. Characterizing the behaviour of noisy quantum hardware is essential in developing error correction code, and noise-resilient algorithms. Therefore, in future, we would like to do a noise analysis of QAOA for solving the problem of VRP, or any other combinatorial optimization problem.

DATA AVAILABILITY

The code created to run these simulations and related supplementary data could be made available to any reader upon reasonable request.

ACKNOWLEDGMENTS

[*] utkarsh.azad@research.iiit.ac.in

[†] bikas.riki@gmail.com, bkb18rs025@iiserkol.ac.in

[‡] pprasanta@iiserkol.ac.in

[1] Qiskit: An Open-source Framework for Quantum Computing, 10.5281/zenodo.2562110 (2019).

[2] R. Smith, M. J. Curtis, and W. J. Zeng, A Practical Quantum Instruction Set Architecture, arXiv:1608.03355 [quant-ph] (2016).

[3] Google AI Quantum Lab. Retrieved December 16, 2019, from https://research.google/teams/applied-science/quantum/.

[4] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[5] E. Farhi, J. Goldstone, and S. Gutmann, A Quantum Approximate Optimization Algorithm, arXiv:1411.4028 [quant-ph] (2014).

[6] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Mauerer, and C. Linnhoff-Popien, A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer, Front. ICT **6**, 13 (2019).

[7] K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience, arXiv:1805.10928 [quant-ph] (2018).

[8] A. Nowbagh, B. K. Behera, and P. K. Panigrah, A Quantum Approach for Solving Vehicle Routing Problem: An IBM Quantum Experience, DOI:10.13140/RG.2.2.18975.30887 (2019).

[9] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[10] D-Wave Systems: Difference between BQM, Ising, and QUBO problems? Retrieved December 16, 2019, from https://support.dwavesys.com/hc/en-us/community/posts/360017439853-Difference-between-BQM-Ising-and-QUBO-problems.

[11] G. Verdon, M. Broughton, and J. Biamonte, A quantum algorithm to train neural networks using low-depth circuits, arxiv:1712.05304 [quant-ph] (2019)

[12] T. Albash, and D. Lidar, Adiabatic Quantum Computing, Rev. Mod. Phys. **90**, 015002 (2018).

[13] D. Aharonov, W. V. Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. 45th Annual IEEE Symposium on Foundations of Computer Science (2004).

[14] Z. Wang, S. Hadfield, Z. Jiang, and G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, Phys. Rev. A **97**, 1-13 (2018)

[15] E. Farhi, J. Goldstone, and S. Gutmann, A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem, arXiv:1411.4028 [quant-ph] (2014).

[16] C. Lin, and Y. Zhu, Performance of QAOA on Typical Instances of Constraint Satisfaction Problems with Bounded Degree. arXiv:1601.01744 [quant-ph]. (2016).

[17] M. Willsch, D. Willsch, F. Jin, H. D. Raedt, and K. Michielsen, Benchmarking the Quantum Approximate Optimization Algorithm, arxiv:1907.02359 [quant-ph] (2019).

[18] M. Alam, A. Ash-Saki, and S. Ghosh, Analysis of Quantum Approximate Optimization Algorithm under Realistic Noise in Superconducting Qubits, arxiv:1907.09631 [quant-ph] (2019).

[19] G. Guerreschi, and A. Y. Matsuura, QAOA for Max-Cut requires hundreds of qubits for quantum speed-up, Sci. Rep. **9**, 6903 (2019).