Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm

Utkarsh Azad, Bikash K. Behera, Emad A. Ahmed, Prasanta K. Panigrahi, and Ahmed Farouk, Member, IEEE

Abstract—Intelligent transportation systems (ITS) are a critical component of Industry 4.0 and 5.0, particularly having applications in logistic management. One of their crucial utilization is in supply-chain management and scheduling for optimally routing transportation of goods by vehicles at a given set of locations. This paper discusses the broader problem of vehicle traffic management, more popularly known as the Vehicle Routing Problem (VRP), and investigates the possible use of near-term quantum devices for solving it. For this purpose, we give the Ising formulation for VRP and some of its constrained variants. Then, we present a detailed procedure to solve VRP by minimizing its corresponding Ising Hamiltonian using a hybrid quantum-classical heuristic called Quantum Approximate Optimization Algorithm (QAOA), implemented on the IBM Qiskit platform. We compare the performance of QAOA with classical solvers such as CPLEX on problem instances of up to 15 qubits. We find that performance of QAOA has a multifaceted dependence on the classical optimization routine used, the depth of the ansatz parameterized by $p$, initialization of variational parameters, and problem instance itself.

Index Terms—Vehicle routing problem, Ising model, combinatorial optimization, quantum approximate algorithms, variational quantum algorithms.

I. INTRODUCTION

I
NTELIGENT transportation systems (ITS) aim to integrate state of the art technologies into vehicles (cargo vehicles, personal vehicles, etc.) and their infrastructure (roadways, traffic management system, etc.) to enhance the efficiency of the transport services by turning the decision-making involved within smarter, i.e., more coordinated and data-driven [1].

Manuscript received 30 December 2021; revised 24 March 2022; accepted 22 April 2022. Date of publication 13 May 2022; date of current version 7 July 2023. The Associate Editor for this article was S. Mumtaz. (Corresponding author: Ahmed Farouk.)

Utkarsh Azad is with the Center for Computational Natural Sciences and Bioinformatics and the Center for Quantum Science and Technology, International Institute of Information Technology, Hyderabad, Telangana 500032, India (e-mail: utkarsh.azad@research.iiit.ac.in).

Bikash K. Behera is with the Department of Physical Sciences, Indian Institute of Science Education and Research Kolkata, Mohanpur, West Bengal 741246, India, and also with Bikash’s Quantum (OPC) Private Ltd., Balindi, Mohanpur, Nadia, West Bengal 741246, India (e-mail: bikash@bikashsquqantum.com).

Emad A. Ahmed is with the Department of Computer Science, Faculty of Computers and Information, South Valley University, Qena 83523, Egypt (e-mail: emad@svu.edu.eg).

Prasanta K. Panigrahi is with the Department of Physical Sciences, Indian Institute of Science Education and Research Kolkata, Mohanpur, West Bengal 741246, India (e-mail: prprasanta@iiserkol.ac.in).

Ahmed Farouk is with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada 83523, Egypt (e-mail: ahmed.farouk@sci.svu.edu.eg).

Digital Object Identifier 10.1109/TITS.2022.3172241

For example, many ITS technologies can help supply-chain management and scheduling by (i) optimizing the trip routes for supply vehicles to take the shortest possible path to reduce the consumption of resources like fuel and time, and (ii) managing the vehicular traffic effectively, leading to lesser congestion [2]. Since ITS technologies possess the solution for fixing the ravaging economic impact of logistics control, it becomes a crucial component required to make the ongoing effort to transition to Industry 4.0 and 5.0 possible [1], [3].

However, the key challenge for implementing ITS is to be able to solve computationally hard optimization problems involved with them [1]. The most general strategy for realizing practical applications of ITS technologies involves mapping them to known optimization problems that can be either solved fully by an exact algorithm or solved approximately by relaxing some of its constraints. In general, it is always challenging to obtain exact solutions for these problems since an 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 exponentially, making their full solutions computationally intractable. One of the popular applications of ITS technologies in the context of planning and scheduling of supply chains that fits this description is the Vehicle Routing Problem (VRP), which is an NP-hard combinatorial optimization problem [4], [5]. It involves finding the optimal set of routes for a fleet of vehicles to deliver goods at a given set of locations.

The most commonly used techniques for solving VRP are (i) exact algorithms like Branch and Bound, which would require exponential time in case of decently big instances of VRP [6], (ii) heuristic algorithms, which use smarter search-space exploration strategies to get good quality solutions in limited time [7], (iii) two-phase algorithms, which uses a divide and conquer approach involving clustering the problem into small instances and then routing them [8], and (iv) metaheuristic algorithms such as genetic algorithms, simulated annealing, etc. [9], [10]. None of these approaches can fully solve the VRP and can only provide some approximate solutions in a limited time. This is the case with various other potential applications of ITS technologies in Industry 4.0 and 5.0 that involve solving such optimization problems. Hence, it becomes necessary for us to develop efficient tools and techniques for solving optimization problems more effectively in a real-time setting in order to realize ITS applications practically.

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Recently, with the development of quantum processors by both industries and academia [11]–[13], various proposals have been made to use them to solve these optimization problems. One main reason for this is that these devices are speculated to have a computational advantage over classical ones because they use quantum resources such as superposition and entanglement. Hence, this makes them an ideal candidate to realize the computational resources required to implement ITS technologies. However, the computational capabilities of these current generation quantum processors, also known as Noisy Intermediate-Scale Quantum (NISQ) devices, are considerably restricted due to their intermediate size in terms of qubits count, limited connectivity between qubits, imperfect qubit-control, short coherence times, and minimal error-correction. Hence, they are only able to run algorithms with limited circuit depth [14].

The Quantum Approximate Optimization Algorithm (QAOA) [15] is one such algorithm that belongs to the class of hybrid quantum-classical (HQC) algorithms. These are called such because they use a workflow involving both quantum and classical computers to solve a given task by solving only the computational bottleneck part of the problem on a quantum device and thus alleviating the need for high depth quantum circuits. In this context, in QAOA, we attempt to perform a coarse trotterized adiabatic evolution of the known ground state of a mixer Hamiltonian $H_{\text{mixer}}$ to the unknown ground state of a cost Hamiltonian $H_{\text{cost}}$ which encodes the problem using a parameterized unitary $U(\hat{\theta})$ on a quantum processor, where parameters $\hat{\theta}$ are tuned by a classical processor. It is based on the fact that the problem of solving NP-hard combinatorial optimization problems can be mapped to the problem of minimizing an Ising Hamiltonian, which can be done on near-term devices using variational principle [16].

In this work, we investigate the use of QAOA to solve the Vehicle Routing Problem (VRP). We first present the complete QUBO formalism and the corresponding Ising formulation for the VRP for near near-term quantum hardware. We also discuss strategies to include time windows or capacity constraints to these formulations. We then explored the performance of QAOA on small problem instances of VRP requiring up to 15 qubits and conducted a comparative analysis with CPLEX [17], which is one of the best known classical solvers for optimization problems. This analysis also looked at the effect of utilizing different classical optimizers routines and increasing the depth of parameterized quantum circuits in the performance of QAOA. Finally, we conclude by discussing the insights for QAOA to be advantageous and discuss strategies to improve its performance.

Structure – In section II, we discuss VRP and present its Ising formulation. Then, in section III, we present a didactic introduction to QAOA and then show the results of our simulations in section IV. Finally, in section V, we discuss the performance and limitations of using QAOA to solve combinatorial optimizations problems in general.

II. VEHICLE ROUTING PROBLEM

Vehicle Routing Problem is an NP-hard combinatorial optimization problem [4], [5]. 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 [18], where now a group of $k$ salesmen have to collectively serve $n - 1$ locations such that each location is served exactly once.

In the majority of real world applications, the VRP problem [19] 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 them while providing a brief methodology to introduce any required additional constraints in the Sec. II-D. But before coming to that, we first describe solving the problem instance $(n,k)$ of VRP using QAOA. We discuss this in the following subsequent subsections, where we discuss steps for first finding the QUBO formulation of VRP and then mapping it to the minimization of energy of the corresponding Ising Hamiltonian $\hat{H}_{\text{Ising}}$ [20], an equivalent NP-Hard problem.

A. Energy Functional for VRP

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

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

Subject to the following constraints:

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

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

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

$$\sum_{j \in \text{target}[0]} x_{j0} = k$$  (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 with index 0. Now, using Eqs. (1-5), the energy functional $H_{\text{VRP}}$ of the above problem can be written as:

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

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

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

Adiabatic evolution (Fig. is adapted from [21]). This adiabatic time evolution path. Simulated annealing (middle) follows this path (black) in the state space. Conceptually, quantum annealing (top) follows this adiabatic time evolution path (black) in the state space. Conceptually, quantum annealing (top) 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 (Fig. is adapted from [21]).

From Eq. (12), we can represent minimization of $H_{\text{VRP}}$ 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:

$$f(x)_{\text{QUBO}} = \min_{x \in \{0,1\}^{n \times n}} \mathbf{x}^T H \mathbf{x} + g^T \mathbf{x} + c$$  \hfill (15)

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. In order to find these coefficients in the QUBO formulations [22] of $H_{\text{VRP}}$ given in Eq. (6) we first put in Eqs. (13-14) in Eqs. (8-9) respectively, then expand and regroup Eq. (6) according to Eq. (15).

$$H_{\text{VRP}} = A \sum_{i=0}^{n-1} (z_{S[i]}^T z_{S[i]}^T + z_{T[i]}^T z_{T[i]}^T) \mathbf{x}^2$$

$$+ 2A \sum_{i=1}^{n-1} (z_{S[i]}^T z_{T[i]}^T) z_{S[i]}^T \mathbf{x}$$

$$- 2Ak(z_{S[0]}^T + z_{T[0]}^T) z_{S[0]}^T + 2A(n-1) + 2Ak^2$$  \hfill (16)

From Eq. (16), we get the coefficients $Q$ $(n(n-1) \times n(n-1))$, $\mathbf{g}$ $(n(n-1) \times 1)$ and $c$:

$$Q = A \left( [z_{S[1]}^T, z_{T[2]}^T, \ldots, z_{S[n-1]}^T, z_{T[n-2]}^T, z_{T[0]}^T, z_{T[2]}^T, \ldots, z_{T[n-1]}^T]^T, \ldots, z_{T[0]}^T]^T \right)$$

$$\frac{[z_{S[1]}^T]^{(n-1)} [z_{S[2]}^T]^{(n-1)} \ldots [z_{S[n-1]}^T]^{(n-1)}}{1}$$  \hfill (17)

$$\mathbf{g} = \mathbf{w} - 2A(\mathbf{J} + \mathbf{K}) - 2Ak(z_{S[0]}^T + z_{T[0]}^T)$$

$$c = 2A(n-1) + 2Ak^2$$  \hfill (18)

Here, $\mathbf{J}$ is a $n \times (n-1)$ vector with first $n-1$ elements 0 and rest $(n-1)^2$ elements 1, vector $K$ is $\mathbf{x}$ with $x_{ij} = 1$ if $j \neq 0, \forall i \in [0, \ldots, n-1]$, else 0 and $\mathbf{w}$ is a weight vector.

C. Ising Formulation for VRP

To construct complete Ising Hamiltonian for $H_{\text{VRP}}$ we expand Eq. (15) by using Eqs. (17-19) and rewrite all the binary variables $x_{ij} \in \{0,1\}$ using spin variables $s_{ij} \in \{-1,1\}$.

$$x_{ij} = \frac{s_{ij} + 1}{2}$$  \hfill (20)

By regrouping [22] together similar terms from the expansion, one can express $H_{\text{Ising}}$ for VRP $(n, k)$ as:

$$H_{\text{Ising}} = - \sum_{i < j} I_{ij} s_i s_j + \sum_i h_i s_i + d$$  \hfill (21)

$$I_{ij} = \frac{Q_{ij}}{4} \quad \forall i < j, \quad I_{ii} = 0 \quad \forall i$$  \hfill (22)

$$h_i = \frac{g_i}{2} + \sum_j Q_{ij}/4 + \sum_j Q_{ji}/4$$  \hfill (23)

$$d = c + \sum_i \frac{g_i}{2} + \sum_i \frac{Q_{ii}}{4} + \sum_i \sum_j \frac{Q_{ij}}{4}$$  \hfill (24)

Replacing $s_i$ with $\sigma_i^z$ i.e. the Pauli-Z operator acting on $i^{th}$ qubit, gives the quantum mechanical description of $H_{\text{Ising}}$ which is implementable on a quantum computer.
D. Other Variants of VRP

In the previous subsections, we provide the Ising formulation for the general purpose VRP without explicitly enforcing any additional constraints. However, in practice, several variations for the VRP exists both in the literature and are utilized in practice. For example, in vehicle routing problem with time windows (VRPTW), the goal is to make vehicles visit the location within specific time windows during which visit must be made. Similarly, in capacitated vehicle routing problem (CVRP), the vehicles have a limit on number of visits they can make. In these cases, one may then follow the same approach as our current one to obtain a QUBO formulation for the decision variables $x_{ij}$ and then map that to the Ising one.

E. Related Work

Traditionally, many classical optimization methods have been used for solving the Vehicle Routing Problem (VRP) and its various variants in the past. These include exact algorithms such as branch and bound [6], construction heuristics like Clarke and Wright saving algorithm [23], improvement heuristics [24], and meta-heuristic algorithms such as genetic algorithms [9], simulated annealing [10], Tabu search [25], and quantum approximate optimization algorithm [27], etc.

Lately, in addition to these existing methods, many quantum-based optimization strategies are also being explored. One of the first few techniques that were used for solving capacitated vehicle routing problems was based on quantum annealing [28], and quantum walks [29]. However, these quantum-based techniques are unsuitable for the current generation of noisy quantum devices as their realization would require large circuit depths for their realization. Hence, there has also been a significant effort in using hybrid quantum-classical optimization algorithms for this purpose [30]. In this regard, we investigate using the quantum approximate optimization algorithm (QAOA) to solve the general purpose variant of VRP.

III. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

Adiabatic quantum computation (AQC) [31] 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 $H_{\text{mixer}}$ and $H_{\text{cost}}$. Amongst them, the ground state of $H_{\text{mixer}}$ should be an easily preparable state such as $|+\rangle^\otimes N$ and ground state of $H_{\text{cost}}$ encode the solution to our problem. Both Hamiltonians $H_{\text{mixer}}$ and $H_{\text{cost}}$, should be local, i.e. they only involve terms for interactions between a constant number of particles. The
instantaneous Hamiltonian $\hat{H}(t)$ for the system is:

$$\hat{H}(t) = (1 - t)\hat{H}_{\text{mixer}} + t\hat{H}_{\text{cost}}$$  \hspace{1cm} (25)$$

In AQC, the spectral gap $\Delta E = E_1 - E_0$, i.e., the difference between the ground state and first excited state energy of $\hat{H}(t)$ bounds the step size $dt$ one can take to follow adiabatic pathway as $dt \sim O(1/(\Delta E)^2)$ [32]. Hence, the computation time to solve any problem rises exponentially as $\Delta E$ becomes infinitesimally small. This limits its ability to solve a certain instance of hard optimization problems where spectral gaps are in general small.

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:

$$U = U(\hat{H}_{\text{mixer}}, \beta_0)U(\hat{H}_{\text{cost}}, \gamma_0) \ldots$$

$$U(\hat{H}_{\text{mixer}}, \beta_{p-1})U(\hat{H}_{\text{cost}}, \gamma_{p-1})$$  \hspace{1cm} (26)$$

In gate-model quantum computation this means [32] that starting from some initial product state $|\psi_{GS}^0\rangle$, we apply a parameterized gate sequence to produce the state $|\psi_{\text{cost}}\rangle$. For some optimal value of the parameters: $[\beta^*, \gamma^*]$, this is the ground state of $\hat{H}_{\text{cost}}$. 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 $|\psi_{\text{cost}}^*\rangle$. Hence, QAOA belongs to the class of hybrid quantum-classical variational algorithms.

IV. SIMULATIONS AND RESULTS

To showcase the performance of QAOA for solving VRP, we prepare various problem instances $(n, k)$, describing a VRP with $n$ locations and $k$ vehicles with a distance matrix $D$ representing the euclidean distances between locations. We attempt to solve these instances using the QAOA algorithm implemented in IBM Qiskit and subsequently compare the obtained results against those obtained from classical algorithms like CPLEX. For performing QAOA simulation using our formulation presented in Sec. II-C, one needs $N = n \times (n - 1)$ qubits to encode the problem instance, i.e., the state of each qubit represents the possibility of an edge between two nodes.

The mixing Hamiltonian $\hat{H}_{\text{mixer}}$ and the cost Hamiltonian $\hat{H}_{\text{cost}}$ for this problem are:

$$\hat{H}_{\text{mixer}} = \sum_{i=0}^{n \times (n-1)-1} \sigma_i^z$$  \hspace{1cm} (27)$$

$$\hat{H}_{\text{cost}} = - \sum_{j=0}^{n \times (n-1)-1} l_{ij} \sigma_i^z \sigma_j^z - \sum_{i} h_i \sigma_i^z - d$$  \hspace{1cm} (28)$$

In $\hat{H}_{\text{cost}}$, the terms of $l_{ij}$ determine the coupling between spin $\sigma_i^z$ and $\sigma_j^z$, the terms $h_i$ determine the contribution of each individual spin $\sigma_i^z$, and $d$ is an offset determined by the problem instance. For implementing unitary for $\hat{H}_{\text{cost}}$, the $\sigma_i^z \sigma_j^z$-interaction is realized using RZ gate operation on qubit $j$ sandwiched between two CNOT gates with $i$ and $j$ as control and target qubits, respectively. Similarly, for implementing $\hat{H}_{\text{mixer}}$, the $\sigma_i^z$-interaction is realized with an RX gate operation.

In each of the following cases, we begin with the state $|+\rangle ^{\otimes (n-1)}$, which is the ground state of $\hat{H}_{\text{mixer}}$ given in Eq. (27). This state is prepared by applying Hadamard on all qubits initialized to $|0\rangle$. From Eq. (26), this state is evolved as:

$$|\vec{\gamma}, \vec{\beta}\rangle = e^{-i\hat{H}_{\text{mixer}}\beta_0}e^{-i\hat{H}_{\text{cost}}\gamma_0} \ldots$$

$$e^{-i\hat{H}_{\text{mixer}}\beta_{p-1}}e^{-i\hat{H}_{\text{cost}}\gamma_{p-1}} |+\rangle ^{\otimes (n-1)}$$  \hspace{1cm} (29)$$

Here, the evolution performed by $e^{-i\hat{H}_{\text{mixer}}\beta}$ entangles the qubit by separating the relative phase between them while keeping the amplitude unchanged. In contrast, the subsequent evolution performed by $e^{-i\hat{H}_{\text{cost}}\gamma}$ mixes the amplitude of different computational basis states, ensuring that the subspace with more probable feasible solutions gets expressed more. For the final evolved state $|\vec{\gamma}, \vec{\beta}\rangle$, we calculate energy $E$ by measuring the expectation value of $\hat{H}_{\text{cost}}$ as:

$$E = \langle \vec{\gamma}, \vec{\beta} | \hat{H}_{\text{cost}} | \vec{\gamma}, \vec{\beta} \rangle$$  \hspace{1cm} (30)$$

Running a classical optimization routine on Eq. (30), we get the optimal value of $|\vec{\gamma}, \vec{\beta}\rangle$ as $|\vec{\gamma}^*, \vec{\beta}^*\rangle$. To get the final result we measure the state $|\vec{\gamma}^*, \vec{\beta}^*\rangle$. As shown in Fig. (3a), 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 a $2^{n \times (n-1)}$ length bit string $b$ which represents flattened version of the adjacency matrix of the graph. The total cost
the correct solution with sufficient probability. In Fig. (3a),

Using COBYLA optimizer, for $p$ ranging from 6 to 40. Fig. (4b) represents the visualization of the state indexed:

Fig. 4. Visualization of QAOA solutions for VRP instances: Each state is used to get the edges between the nodes in the solution and their weights have been labelled. Furthermore, the node with yellow star denotes the depot, or the origin. (a) state indexed 779 = 110100001100 with cost $C_1 = 30.632 + 15.497 + 5.061 + 2 \times 36.840 = 124.871$ for (4, 2), (b) state indexed 623144 = 10011000001000101000 with cost $C_2 = 2 \times 9.711 + 2 \times 47.767 + 2 \times 6.794 = 128.545$ for (5, 2), (c) state indexed 688424 = 101010000010101000 with cost $C_3 = 24.557 + 9.711 + 42.887 + 47.767 + 2 \times 6.794 = 138.511$ for (5, 2), and (d) state indexed 69963 = 1000100010010111 with cost $C_4 = 12.138 + 5.3 + 7.2 + 2 \times 2.626 + 2 \times 0.320 = 30.53$ for (5, 3).

$C$ associated with each bit string is the summation of matrix elements $D_{ij}$ whose corresponding binary variable $b_k$ has non-zero value. In more general terms, each pair $(b_k, D_{ij})$ is representative of $x_{ij}$ and existence of edge $i \rightarrow j$ as shown in Fig. (4) and is further shown in Eqs. (32), (34) and (35). In the subsequent subsections, we first present solutions obtained from QAOA routine for individual problem instances (4, 2), (5, 2), and (5, 3) and then give a comparative analysis of the experimental results and cost $C$ obtained from solving several of these instances against CPLEX.

A. QAOA Experiments

1) Experiment 1: In the first experiment, we have solved the problem instance (4, 2), described by the following distance matrix:

$$D_1 = \begin{bmatrix} 0 & 36.84 & 5.06 & 30.63 \\ 36.84 & 0 & 24.55 & 63.22 \\ 5.06 & 24.55 & 0 & 15.50 \\ 30.63 & 63.22 & 15.50 & 0 \end{bmatrix}$$  \hfill (31)

To encode the problem, we used $N = 4 \times 3 = 12$ qubits. Using COBYLA optimizer, for $p \geq 12$ we were able to get the correct solution with sufficient probability. In Fig. (3a), 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, 1, 1, 0, 0] in Fig. (4a) using the following adjacency matrix $A_1$:

$$A_1 = \begin{bmatrix} X & 1 & 1 & 0 \\ 1 & X & 0 & 0 \\ 0 & 0 & X & 1 \\ 1 & 0 & 0 & X \end{bmatrix}$$  \hfill (32)

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

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

$$D_2 = \begin{bmatrix} 0 & 6.794 & 61.653 & 24.557 & 47.767 \\ 6.794 & 0 & 87.312 & 47.262 & 39.477 \\ 61.653 & 87.312 & 0 & 9.711 & 42.887 \\ 24.557 & 47.262 & 9.711 & 0 & 40.98 \\ 47.767 & 39.477 & 42.887 & 40.98 & 0 \end{bmatrix}$$  \hfill (33)

To encode the problem, we used $N = 5 \times 4 = 20$ qubits. We tried COBYLA for different values of $p$ ranging from 6 to 40. Fig. (4b) represents the visualization of the state indexed:
10 while convergence not met

623144 = [1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
i.e., the state we got, using the following adjacency matrix $A_2$:

\[
A_2 = \begin{bmatrix}
X & 1 & 0 & 0 & 1 \\
1 & X & 0 & 0 & 0 \\
0 & 0 & X & 1 & 0 \\
0 & 0 & 1 & X & 0 \\
1 & 0 & 0 & 0 & X
\end{bmatrix}
\] (34)

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

Whereas, Fig. (4c) represents the visualization of the state indexed: 688424 = [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0], i.e., the optimal state we should have got, using the following adjacency matrix $A_3$:

\[
A_3 = \begin{bmatrix}
X & 1 & 0 & 1 & 0 \\
1 & X & 0 & 0 & 0 \\
0 & 0 & X & 0 & 1 \\
0 & 0 & 1 & X & 0 \\
1 & 0 & 0 & 0 & X
\end{bmatrix}
\] (35)

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. (4b), the solution we got was not the optimal one i.e., given in Fig. (4c).

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

\[
D_4 = \begin{bmatrix}
0. & 12.138 & 0.32 & 7.2 & 2.626 \\
12.138 & 0. & 16.307 & 5.3 & 17.021 \\
0.32 & 16.307 & 0. & 9.309 & 2.98 \\
7.2 & 5.3 & 9.309 & 0. & 16.759 \\
2.626 & 17.021 & 2.98 & 16.759 & 0.
\end{bmatrix}
\] (36)

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. (3a), 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.

We present the visualization of the state 69963 = [1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0] in Fig. (4a) using the following adjacency matrix $A_4$:

\[
A_4 = \begin{bmatrix}
X & 1 & 1 & 0 & 1 \\
0 & X & 0 & 1 & 0 \\
1 & 0 & X & 0 & 0 \\
1 & 0 & 0 & 0 & X
\end{bmatrix}
\] (37)

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

B. Comparative Analysis

For the next set of experiments, we made 150 random problem instances each for (a) (4, 2), (b) (5, 2) and (c) (5, 3) VRP instances. To do this, we sampled the distance matrices $D$ randomly and used them to define VRPs, and then solved

Fig. 5. Comparative Analysis of QAOA and CPLEX: We compare the performance of QAOA with $p \in \{1, 10\}$ and CPLEX, which is a classical optimization solver. We use 150 problem instances of VRP: (a) (4, 2), (b) (5, 2) and (c) (5, 3) and plot the average cost obtained for them using both QAOA and CPLEX (red). For QAOA, we present the results for four different classical optimizers with $p = 1$ (green) and $p = 10$ (blue): (i) SPSA, (ii) COBYLA, (iii) Powell, and (iv) BFGS, choosing the best results over the trial of 5 runs.
them using both QAOA and CPLEX. We average out the costs over each of the VRP instances (choosing the best results from 5 trial runs) and present them in Fig. (5). We see that for \( p = 1 \), QAOA performs relatively much more poorly than CPLEX for all three VRP instances with all four tested classical optimizers (with maximum iterations set to 5000): SPSA [33], COBYLA [34], Powell [35], and BFGS [36]. However, as we increase the value of \( p \) to 10, we can see a considerable increase in the performance as the average output for each VRP instance starts approaching the values given by the CPLEX solver. Especially, SPSA and COBYLA optimizers perform much better than the rest of the classical optimizers for higher values of \( p \). In contrast, for lower values of \( p \), no such preference for particular optimizers is seen, and they all perform poorly on average. For a few problem instances, both \( p = 1 \) and \( p = 10 \) gave the same result. However, increment \( p \) positively affects the underlying probability of the possible feasible output states. This effect is also seen while choosing some intelligent guesses for choosing the initial values of the parameters \( \gamma \) and \( \beta \) depending upon the loss landscape of the given problem instance, which becomes a bit trickier as the number of parameters increases with increase in \( p \).

V. DISCUSSION AND CONCLUSION

In the past, quantum approximate optimization algorithm (QAOA) has been widely used in solving various combinatorial optimization problems [37], [38], and here we use it for solving the vehicle routing problem (VRP) on near-term quantum devices. For this purpose, we first devise the Ising formulation for VRP with relevant constraints, which can also be utilized in a quantum annealing routine on a quantum annealer. In our formulation, the number of qubits required for solving a VRP problem instance \((n, k)\) scale quadratically with \( n \), the number of locations. In case of additional constraints such as capacity constraints, it also scales linearly with \( k \), the number of vehicles.

We studied three distinct problems instances \((5, 2), (4, 3)\) and \((5, 3)\), requiring 10, 12, and 15 qubits, respectively. From the results listed in Sec. IV, we see that the probability of obtaining a feasible answer increases with \( p \). Despite this, 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 [39]. This is because, in QAOA, instead of following the adiabatic time evolution path, we try to guess it using \( p \) steps. So, the first straightforward reason could be that the chosen value \( p \) does not produce a good enough guess. Another reason which could explain the failure of QAOA at larger values of \( p \) could be the emergence of new local minimums in our solution energy landscape, which traps classical optimizer routines and make them converge prematurely.

As we examined the performance of QAOA for different types of optimizers, we see that in the case of gradient-based optimizers, there exist possibilities for training getting stalled due to the majority of gradients for variational parameters becoming zero [40]. This could be one possible reason for gradient-free optimizers like SPSA and COBYLA performing much better than the other optimizers in our experiments with higher values of \( p \). Furthermore, Figs. (4b) and (4c) show that sometimes QAOA can choose a lower-energy solution that fails certain constraints instead of a higher-energy solution that satisfies all constraints. This failure could be circumvented by identifying such constraints and increasing the associated penalty.

Overall, we see that training time to find optimal angles for higher \( p \) values becomes exponentially longer for all optimizers. However, good initialization of parameters for QAOA can help reduce this significantly and also improve its success probability [41]. For example, in IV-B, we used the optimized parameters from the QAOA circuit with \( p - 1 \) layers as our initial guess for the parameters in the QAOA circuit with \( p > 1 \) layers to significantly speed up our optimization procedure. Similarly, another technique known as parameter concentration could also be used for this purpose, which uses optimal parameter values for some smaller problem instances to infer initial guesses for a bigger problem instance [42].

Furthermore, to further improve the optimization procedure, we can restrict the solution subspace for QAOA by choosing a different mixer Hamiltonian other than the standard one used in this work which explores all possible solution states [43]. Moreover, in the previous studies [39], [44], 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 \( \langle H_{\text{cost}} \rangle \), \( \langle H_{\text{cost}} \rangle \). Characterizing the behavior of noisy QAOA is crucial for improving its noise resilience and developing error correction codes [45].

Therefore, we conclude that QAOA can possibly be used to solve combinatorial optimization problems like VRP, which lies at the heart of realizing intelligent transportation systems (ITS) for both industry 4.0 and 5.0. However, to do this, one needs to follow a multi-faceted approach to improving the performance of QAOA by working on the research directions mentioned above. We also note that while QAOA can solve VRP for smaller problem instances, it soon becomes challenging for larger problem instances. Moreover, the qubits required to encode problem instances of practical usability for which we can showcase a possible quantum advantage are in order of millions. We could not note any speed-ups in finding optimal solution for even the realizable smaller problem instances due to the classical optimization of variational parameters being challenging for higher values of \( p \).

REFERENCES

[1] S. A. Shaheen and R. Finson, “Intelligent transportation systems,” in Encyclopedia of Energy. Amsterdam, The Netherlands: Elsevier, 2004, pp. 487–496.

[2] M. Juma and K. Shaalan, “Cyberphysical systems in the smart city: challenges and future trends for strategic research,” in Swarm Intelligence for Resource Management in Internet of Things. Amsterdam, The Netherlands: Elsevier, 2020, pp. 65–85.

[3] L. Abualigah, A. Diabat, P. Sumari, and A. H. Gandomi, “Applications, deployments, and integration of Internet of Drones (IoD): A review,” IEEE Sensors J., vol. 21, no. 22, pp. 25532–25546, Nov. 2021.

[4] J. K. Lenstra and A. H. G. R. Kan, “Complexity of vehicle routing and scheduling problems,” Networks, vol. 11, no. 2, pp. 221–227, 1981.

[5] G. Laporte and Y. Nobert, “Exact algorithms for the vehicle routing problem,” in Surveys in Combinatorial Optimization, Amsterdam, The Netherlands: Elsevier, 1987, pp. 147–184.
We refer to the original source for detailed information on the contributions and methodologies in the field of quantum computing, optimization algorithms, and vehicle routing problems. The works cited here represent a selection of significant research that has advanced these areas.

[31] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, “Quantum approximate optimization algorithm,” Nov. 2014, arXiv:1412.6062.

[32] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, “Adiabatic quantum computation is equivalent to standard quantum computation,” in Proc. 45th Annu. IEEE Symp. Found. Comput. Sci., May 2004, pp. 42–51.

[33] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 3, no. 7, pp. 1–48, Jun. 2019.

[34] S. H. Sack and M. Serbyn, “Quantum annealing initialization of the quantum approximate optimization algorithm,” Quantum Inf. Process., vol. 19, no. 7, pp. 1–24, Jun. 2020.

[35] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, “XY mixtures: Analytical and numerical results for the quantum alternating operator ansatz,” Phys. Rev. A, Gen. Phys., vol. 104, no. 1, Jul. 2021, Art. no. L010401.

[36] A. Ash-Saki, and S. Ghosh, “Analysis of quantum approximate optimization algorithm under realistic noise in superconducting qubits,” 2019, arXiv:1907.09631.

[37] A. Lucas, “Ising formulations of many NP problems,” Frontiers Phys., vol. 2, 2014, doi: 10.3389/fphy.2014.00005.

[38] G. Verdon, M. Broughton, and J. Biamonte, “A quantum algorithm to train neural networks using low-depth circuits,” 2017, arXiv:1712.05304.

[39] (2018), D-Wave Systems: Difference between BQM, Ising, and QUBO Problems. Accessed: Dec. 16, 2019. [Online]. Available: https://support.dwavesys.com/hc/en-us/community/posts/360071439583-Difference-between-BQM-Ising-and-QUBO-problems

[40] T. Pichpibul and R. Kawtummachai, “A heuristic approach based on Clarke-Wright algorithm for open vehicle routing problem,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 1, 2021, doi: 10.1109/TNNLS.2021.3066828.

[41] S. Hadfield, Z. Jiang, and E. G. Rieffel, “Concentrations in quantum approximate optimization,” Quantum, vol. 9, no. 1, pp. 1–6, Nov. 2018.

[42] A. Lucas, “Ising formulations of many NP problems,” Frontiers Phys., vol. 2, 2014, doi: 10.3389/fphy.2014.00005.

[43] G. Verdon, M. Broughton, and J. Biamonte, “A quantum algorithm to train neural networks using low-depth circuits,” 2017, arXiv:1712.05304.

[44] (2018), D-Wave Systems: Difference between BQM, Ising, and QUBO Problems. Accessed: Dec. 16, 2019. [Online]. Available: https://support.dwavesys.com/hc/en-us/community/posts/360071439583-Difference-between-BQM-Ising-and-QUBO-problems

[45] T. Pichpibul and R. Kawtummachai, “A heuristic approach based on Clarke-Wright algorithm for open vehicle routing problem,” Sci. World J., vol. 2013, pp. 1–11, Jan. 2013.

[46] Y. Wu, W. Song, Z. Cao, J. Zhang, and A. Lim, “Learning improvement heuristics for solving routing problems,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 1, 2021, doi: 10.1109/TNNLS.2021.3066828.

[47] S. Hadfield, Z. Jiang, and E. G. Rieffel, “Concentrations in quantum approximate optimization,” Quantum, vol. 9, no. 1, pp. 1–6, Nov. 2018.
Emad A. Ahmed received the B.Sc., M.Sc., and Ph.D. degrees in physics from the Faculty of Science, South Valley University, Qena, Egypt, in 1999, 2005, and 2010, respectively. He is currently an Associate Professor with the Computer Science Department, Faculty of Computer and Information, South Valley University. He has published many papers in reputed journals and served as a reviewer for many journals. His research interests are quantum communications and cryptography, quantum secured blockchain and the IoT, quantum information and computation, and information security and privacy.

Prasanta K. Panigrahi received the B.Sc. (Hons.) and M.Sc. degrees from Utkal University, Orissa, India, in 1978 and 1980, respectively, and the Ph.D. degree from the University of Rochester, Rochester, NY, USA, in 1988, all in physics. He is currently a Professor of physics with the Indian Institute of Science Education and Research Kolkata, Kolkata, India, and also a Faculty Member with the Physical Research Laboratory, Navrangpura, Ahmedabad. He is the author of more than 400 peer-reviewed international publications. His research interests include the area of quantum mechanics, quantum theories, and application of wavelet transform to data analysis.

Ahmed Farouk (Member, IEEE) received the M.Sc. and Ph.D. degrees from Mansoura University, Egypt. He is currently an Assistant Professor, before that he was a Post-Doctoral Research Fellow at Wilfrid Laurier University and Ryerson University, Canada. He is one of the Top 20 technical co-founders of the Quantum Machine Learning Program by Creative Destruction Laboratory, University of Toronto. Furthermore, he is selected as the Top 25 of InnovateTO150 Canada to showcase the best of Toronto’s next generation of change-makers, innovators, and entrepreneurs. He is exceptionally well-known for his seminal contributions to theories of quantum information, communication, and cryptography. He has published 62 papers in reputed and high impact journals like Scientific Reports (Nature) and Physical Review A. The exceptional quality of his research is recognized nationally and internationally. He was selected by the scientific review panel of the Council for the Lindau Nobel Laureate Meetings to participate in the 70th Lindau Nobel Laureate Meeting. His volunteering work is apparent since he appointed as the Chair of the IEEE Computer Chapter for the Waterloo-Kitchener area and editorial board for many reputed journals like Scientific Reports (Nature), IET Quantum Communication, and IEEE ACCESS. Also, he was selected for IEEE and IET Young Professional Ambassador as a moderator for the new IEEE TechRxiv, and as an Associate Editor of the IEEE Canadian Review (ICR).