

QAOA-Assisted Benders' Decomposition for Mixed-integer Linear Programming

Zhongqi Zhao^{*†}, Lei Fan^{*†}, Yuanxiong Guo[‡], Yu Wang[§], Zhu Han^{*}, and Lajos Hanzo[¶]

^{*}Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA

[†]Department of Engineering Technology, University of Houston, Houston, TX, USA

[‡]Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX, USA

[§]Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA

[¶]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK

{zzhao27, lfan8, zhan2}@uh.edu, yuanxiong.guo@utsa.edu, wangyu@temple.edu, lh@ecs.soton.ac.uk

Abstract—Benders' decomposition (BD) algorithm constitutes a powerful mathematical programming method of solving mixed-integer linear programming (MILP) problems with a specific block structure. Nevertheless, BD still needs to solve an NP-hard quasi-integer programming master problem (MAP), which motivates us to harness the popular variational quantum algorithm (VQA) to assist BD. More specifically, we choose the popular quantum approximate optimization algorithm (QAOA) of the VQA family. We transfer the BD's MAP into a digital quantum circuit associated with a physically tangible problem-specific *ansatz*, and then solve it with the aid of a state-of-the-art digital quantum computer. Next, we evaluate the computational results and discuss the feasibility of the proposed algorithm. The hybrid approach advocated, which utilizes both classical and digital quantum computers, is capable of tackling many practical MILP problems in communication and networking, as demonstrated by a pair of case studies.

Index Terms—Benders' Decomposition, Mixed-Integer Linear Programming, Optimization, Digital Quantum Computing, Quantum Approximate Optimization Algorithm

I. INTRODUCTION

Mixed-integer linear programming (MILP) techniques have been broadly applied in communications and networking for UAV trajectory optimization in sensing services [1], space-air-sea blue data computation [2], and beamforming for multi-user communications [3]. However, large-scale MILP problems are NP-hard in general and are not easy to solve. Specifically, Benders' decomposition (BD) [4] has been proposed to help solve these kinds of problems. Briefly, BD divides a MILP problem into a master problem (MAP) and one or more subproblems (SUBs), which are solved alternatively to find the optimal solution. Each subproblem is a linear programming problem and easy to solve. However, the MAP still constitutes a pure integer programming problem and it is generally NP-hard.

To solve MILP problems, quantum computing provides a new promising technique [5]. For example, quantum annealing (QA), as a promising quantum algorithm, has been applied for solving the vector perturbation transmit precoding problem in a multiple-input multiple-output (MIMO) system [6]. Moreover, QA was applied in [7] to solve an integer linear programming (ILP) problem in network function virtualization (NFV). Additionally, a quantum annealer computer and a

classical computer are combined for jointly solving a MILP [8]. The authors of [9] applied such a method for solving a distributed learning optimization and confirmed its advantages. However, QA is a specialized algorithm that cannot work on contemporary digital quantum computers. To address this issue, researchers proposed the quantum approximate optimization algorithm (QAOA) for digital quantum computers [10]. QAOA, as a special case of the variational quantum algorithm (VQA) [11], can reduce noise in the system and make efficient use of the available quantum resources by combining classical optimizers and parameterized quantum circuits [12]. QAOA can solve combinatorial optimization problems with potential computational advantages [13]. The basic philosophy of QAOA is to compute a Hamiltonian whose ground state represents the optimal solution of a combinatorial problem constituted by a quadratic unconstrained binary optimization problem (QUBO) [14]. Moreover, the authors of [15] have made the standard QAOA more suitable for constrained optimization problems and applied it for solving knapsack problems. According to [16], although the existing digital quantum computers have a limited number of qubits, QAOA performs way better than random sampling. Moreover, QAOA allows for the presence of higher-order terms in the Ising input, and it is scalable to a heavy-hexagonal lattice of any size.

The success of QA-assisted BD in solving MILP problems and the power of QAOA inspire us to design a new hybrid digital-quantum BD algorithm by jointly using QAOA and classical computing techniques. However, there are several challenges in integrating QAOA and classical computing algorithms. The *first challenge* is how to construct the quantum circuit for the digital quantum computer. The *second challenge* is how to convert the ILP problem and its potential inputs into the *ansatz* as an input to a digital quantum computer.

To overcome the above challenges, this paper reformulates the MAP to an ILP model and turns it into a QUBO, which can be represented by a digital model to run on a digital quantum computer. We conceive a general method to map the quadratic terms to the corresponding gate pairs. Then, we implement the proposed QAOA-assisted BD algorithm using the IBM Qiskit quantum computer. Finally, we consider a pair of simple

case studies for characterizing the performance of the IBM Qiskit quantum computer in solving the MILP problem with *ansatz*. The main contributions of this paper are summarized as follows.

- We propose a QAOA-assisted BD algorithm for solving MILP problems. Our hybrid quantum BD algorithm converges and returns the same result as the classical algorithm.
- We reformulate the constraints and objective function (OF) of the ILP model to the corresponding QAOA digital quantum circuits.
- We harness a quantum computer provided by IBM to solve MILP problems with the aid of their Qiskit IBM quantum platform. Our experiments demonstrate the feasibility of using digital quantum circuits to efficiently solve MILP problems.

The rest of this paper is organized as follows. Section II introduces the basics of MILP problems and the BD algorithm. Section III illustrates our QAOA-assisted BD algorithm, while Section IV validates our algorithm via simulations on IBM quantum computers and quantum simulators. Finally, Section V concludes the paper.

II. BACKGROUND OF HYBRID QUANTUM CLASSICAL BENDERS' DECOMPOSITION ALGORITHM AND QAOA

A. Hybrid Quantum-classical Benders' Decomposition

We commence by a succinct overview of the hybrid quantum-classical Benders' decomposition (HQCBD) algorithm introduced by [8]. The MILP problem of

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{y}} \quad & \mathbf{c}^\top \mathbf{x} + \mathbf{h}^\top \mathbf{y} \\ \text{s.t.} \quad & \mathbf{Ax} + \mathbf{Gy} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{B}^n, \mathbf{y} \in \mathbb{R}_+^p, \end{aligned} \quad (1)$$

can be decomposed into a MAP and a SUB. First, the MAP is modified by discretizing the SUB's bound $\lambda \in \mathbb{R}$ and turning it into a set of binary bits $\mathbf{z}$ of length m . Now, we have $\lambda \simeq \hat{\lambda} \equiv \hat{\lambda}(\mathbf{z})$, and the co-domain of $\hat{\lambda}$ is $\mathbb{Q}^*$, but for further detail, please refer to [8]. Then, the modified MAP may be reformulated into an ILP suitable for the QA computer to solve, yielding,

$$\begin{aligned} (\text{MAP}) \quad & \max_{\mathbf{x}, \hat{\lambda}} \quad \mathbf{c}^\top \mathbf{x} + \hat{\lambda} \\ \text{s.t.} \quad & (\mathbf{b} - \mathbf{Ax})^\top \mathbf{u}^k \geq \hat{\lambda}, \\ & (\mathbf{b} - \mathbf{Ax})^\top \mathbf{r}^j \geq 0, \\ & \mathbf{x} \in \mathbb{B}^n, \hat{\lambda} \in \mathbb{Q}^*, \\ & \forall k \in \hat{K}, \forall j \in \hat{J}. \end{aligned} \quad (2)$$

In (2), $\hat{K}$ and $\hat{J}$ represent the known extreme points and rays. Then, as seen in Figure 1, the algorithm applies the QA-QUBO penalty conversion method to the MAP's objective function and constraints, followed by creating a QUBO for QA computers to solve. Moreover, the authors of [17] and [9] developed the concept of multi-cuts in HQCBD to accelerate the process.

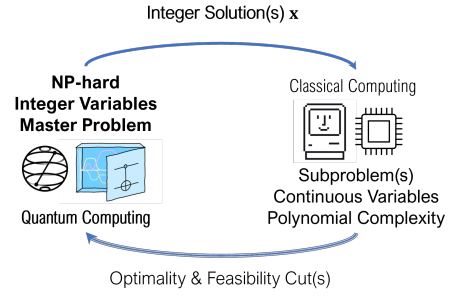


Fig. 1: The proposed hybrid quantum-classical Benders' decomposition algorithm [8]

B. Quantum Approximate Optimization Algorithm

The QAOA was proposed for solving combinatorial optimization problems in [10]. It aims to find approximate solutions to problems of the form $\min_{\mathbf{x}} C(\mathbf{x})$, where $\mathbf{x}$ represents a set of binary variables, and $C(\mathbf{x})$ is a cost function. This type of problems are prevalent in various fields, including operations research, machine learning, and finance.

The QAOA has two main components. The *first component* is a parameterized quantum circuit that manages the schedule of operations. The choice of the circuit structure and the values of its parameters play a crucial role in the algorithm's performance. The *second component* is *classical optimization* harnessed for adjusting the parameters of the quantum circuit. The goal is to find the specific parameter values that minimize the expected cost function.

C. Quantum Circuit Implementation and Hamiltonians

The digital quantum computer represents the Hamiltonian simulation using quantum circuits with appropriately designed quantum gates. More explicitly, the quantum gates are used to represent the pulse schedules to implement the Hamiltonian evolution in these quantum circuits. These transformations are described by the unitary time evolution operator, defined as $U(H, t) = e^{-iHt/\hbar}$. The time-evolution operator $U(H, t)$ can be fully described by a time-domain scalar λ and a Hamiltonian H . In fact, any unitary operator U can be expressed as $e^{i\gamma H}$, where H is a Hermitian operator that can be viewed as a Hamiltonian and γ is a scalar. Accordingly, the quantum circuits can simulate the Hamiltonian evolution based on quantum gates. Using the Trotter-Suzuki decomposition formula [18] of $e^{A+B} \approx (e^{A/n} e^{B/n})^n$, quantum circuits can be designed to express the Hamiltonian with the aid of many non-commuting terms. For instance, we may implement an approximate unitary time-evolution operator as a Hamiltonian of the form $H = \sum_{k=1}^K H_k$ as follows:

$$U(H, t, n) = \prod_{t=1}^n \prod_{k=1}^K e^{iH_k t/n}. \quad (3)$$

where K is the number of sub-Hamiltonians. Then $U(H, t, n)$ approaches $e^{iHt/n}$ as n becomes larger and will finally converge when $n \rightarrow \infty$. Figure 2 shows the implementation and equivalence.

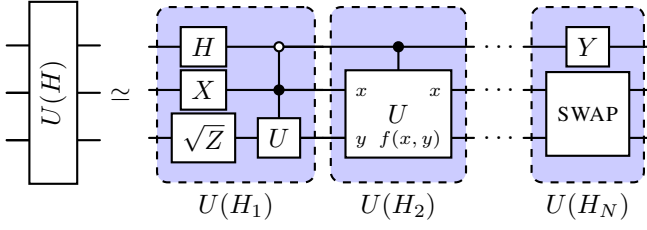


Fig. 2: Quantum circuit implementation and equivalence

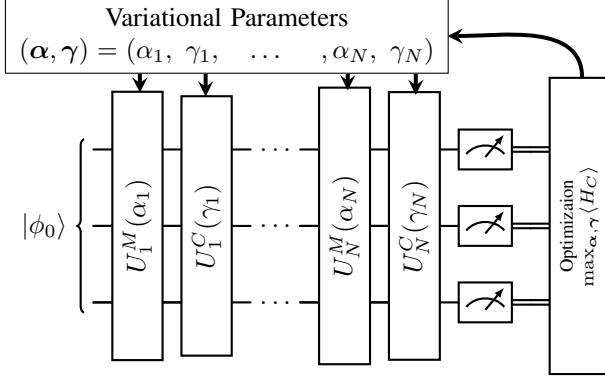


Fig. 3: QAOA quantum circuit

D. QAOA Circuit Diagrams

The QAOA first defines a cost Hamiltonian H_C , so that its ground state represents the solution to the optimization problem. It often relies on the standard mixer Hamiltonian $H_M = \sum_{i=1}^n X_i$ as the candidate. Then, the circuits $U_{H_C} = e^{-i\gamma H_C} = U^C$ and $U_{H_M} = e^{-i\alpha H_M} = U^M$ are constructed as the cost and mixer layers, respectively. Then, the QAOA repeatedly activates the cost and mixer Hamiltonians in its circuit in order to iteratively search for an optimal or near-optimal solution by evolving the quantum state through a sequence of quantum operations. Moreover, it chooses a parameter set (α, γ) having a length of $n \geq 1$ and constructs the circuit $U(\alpha, \gamma)$ as follows:

$$U(\alpha, \gamma) = e^{-i\alpha_n H_M} e^{-i\gamma_n H_C} \dots e^{-i\alpha_1 H_M} e^{-i\gamma_1 H_C}, \quad (4)$$

again, repeatedly activating the cost and mixer Hamiltonians layers. The circuit commences from an initial state, applies $U(\alpha, \gamma)$ of (4), and then uses a classical computer for optimizing the parameters (α, γ) . In the end, the measurements of the output state eventually reveal approximate answers to the optimization problem, once the circuit has been optimized. The general QAOA circuit is illustrated in Figure 3.

III. QAOA-ASSISTED BENDERS' DECOMPOSITION

In the QAOA-assisted BD algorithm, we have to use QAOA to solve the master problem, which is nontrivial. In this section, we describe how to map QUBO to QAOA. *First*, we convert the associated binary variables to the corresponding Pauli matrix in Hamiltonian. Then, we convert different types of linear and quadratic terms from QUBO to digital quantum gates by applying the Pauli-Z measurement operator. *Thirdly*, we introduce the QAOA *ansatz*, and finally, we use those gates to construct the final circuit and get the MAP QAOA *ansatz*.

$$|\phi\rangle \rightarrow R_Z(2ct) = \begin{bmatrix} e^{-ict} & 0 \\ 0 & e^{ict} \end{bmatrix}$$

Fig. 4: QAOA Pauli-Z operator circuit

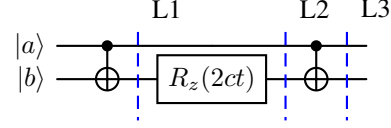


Fig. 5: QAOA $Z \otimes Z$ operator circuit. L_1 , L_2 , and L_3 are 3 probes that detect the state of each layer

A. Mapping QUBO to QAOA

Consider the QUBO problem f_{QUBO} representing the MAP in every iteration [8]:

$$f_{\text{QUBO}}(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i x_j = \mathbf{x}^T \mathbf{C} \mathbf{x}. \quad (5)$$

To solve the problem using QAOA, we can associate the variable x_i with the i th input qubit. If the variable assumes the value of 0, this corresponds to the qubit being in state $|0\rangle$; likewise, value 1 corresponds to the qubit being in state $|1\rangle$. On a digital quantum computer, we measure the state of a qubit in the $\{|0\rangle, |1\rangle\}$ basis by applying the Pauli-rotation- Z operator. However, the eigenvalues of this operator are $+1$ for the state $|0\rangle$, and -1 for the state $|1\rangle$. In order to map this to the $\{0, 1\}$ values of binary variables, we have to modify the measurement operator to be

$$x_i \leftrightarrow \frac{I - Z_i}{2}, \quad (6)$$

where I is the identity operator. With these considerations in mind, the cost Hamiltonian H_C encoding the objective f becomes:

$$\begin{aligned} H_C^{\text{BD}} &= \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n c_{ij} (I - Z_i) (I - Z_j) \\ &= \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n c_{ij} (I - Z_i - Z_j + Z_i Z_j) \end{aligned} \quad (7)$$

$$\xrightarrow[\text{Constant}]{\text{Ignore}} \sum_{i=1}^n \sum_{j=1}^n c_{ij} (Z_i Z_j - Z_i - Z_j),$$

$$Z = R_Z(\pi) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad R_Z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}.$$

In (7), Z is the Pauli- Z gate, R_Z is the Pauli-rotation- Z gate, and θ is a rotation angle about the x-axis of the Bloch sphere. For the problem-driver, we use the mixer Hamiltonian:

$$\begin{aligned} H_M^{\text{BD}} &= \sum_{i=1}^n X_i, \quad X = R_X(\pi) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ R_X(\theta) &= \begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \end{aligned} \quad (8)$$

where X is the Pauli- X gate, $R_X(\theta)$ is the Pauli-rotation- X gate, and θ is a rotation angle about the x-axis of the Bloch sphere.

B. Quantum Gate Formulation

According to Subsection II-D, the QAOA circuit $U(\alpha, \gamma)$ exponentiates the cost Hamiltonians and mixer Hamiltonians, which results in:

$$U^{\text{BD}}(\alpha, \gamma) = e^{-i\alpha_n H_M^{\text{BD}}} e^{-i\gamma_n H_C^{\text{BD}}} \dots e^{-i\alpha_1 H_M^{\text{BD}}} e^{-i\gamma_1 H_C^{\text{BD}}}. \quad (9)$$

Thus, there are 3 types of exponents that we have to convert. The single Pauli-Z operator is the easiest one, since we have $e^{-icZt} = R_Z(2ct)$. Therefore, the transformation is as seen in Figure 4.

1) *Cascade Z Operator*: The quadratic exponential term $e^{-icZZt} = e^{-icZ \otimes Zt}$ in the exponential transformation has to be converted to quantum gate combinations. Observe that the Pauli-Z operator possesses the eigenvectors $|0\rangle, |1\rangle$, with eigenvalues of 1, -1. The induction proceeds as in

$$e^{-icZ \otimes Zt} |ab\rangle = e^{-ci-1^{a \oplus b}t} |ab\rangle. \quad (10)$$

We have $e^A |v\rangle = e^\lambda |v\rangle$ if $A|v\rangle = \lambda|v\rangle$, as proven in [18]. The details are shown in Figure 5, where we have $L_1 = |a\rangle|a \oplus b\rangle$, $L_2 = |a\rangle|e^{-ci-1^{a \oplus b}t}a \oplus b\rangle$, and $L_3 = |a\rangle|e^{-ci-1^{a \oplus b}t}a \oplus b \oplus a\rangle$. Note that the third layer may also be written as $L_3 = |a\rangle|e^{-ci-1^{a \oplus b}t}a \oplus b \oplus a\rangle = e^{-ci-1^{a \oplus b}t} |a\rangle|b\rangle$. We can draw an equivalence between the circuit in Figure 5 and $Z \otimes Z$.

Similarly, if the Hamiltonian commutes, we can concatenate the corresponding circuits as seen in Figure 6 and formulated as

$$e^{-iaZ_i Z_j t - ibZ_j Z_k t} = e^{-iaZ_i Z_j t} e^{-ibZ_j Z_k t}. \quad (11)$$

Now we can simulate the operator corresponding to the cost Hamiltonian in (7) as follows:

$$\begin{aligned} U_C^{\text{BD}}(\gamma_m) &= e^{-i\gamma_m H_C^{\text{BD}}} \\ &= \exp \left\{ -i\gamma_m \sum_{i=1}^n \sum_{j=1}^n c_{ij} (Z_i Z_j - Z_i - Z_j) \right\}. \end{aligned} \quad (12)$$

2) *Mixer Operator*: Besides the cost operator, $U(\alpha, \gamma)$ also exponentiates the mixer Hamilton H_M as $e^{-i\alpha H_M^{\text{BD}}}$. The final representation is calculated as

$$U_M^{\text{BD}}(\alpha) = e^{-i\alpha H_M^{\text{BD}}} = e^{-i\alpha \sum_{i=1}^n X_i} \xrightarrow{\text{one term}} e^{-i\alpha X}. \quad (13)$$

Since we use the Pauli-rotation-Z operator as our measure, we have $X = HZH$, and $HH = I$, H is the Hadamard operator and I is the identity. For the mixer operator, we have

$$\begin{aligned} U_M^{\text{BD}}(\alpha) &= e^{-i\alpha X t} = \sum_{j=0}^{\infty} \frac{(-i\alpha X t)^j}{j!} = \sum_{j=0}^{\infty} \frac{(-i\alpha HZHt)^j}{j!} \\ &= \sum_{j=0}^{\infty} \frac{H(-i\alpha Zt)^j H}{j!} = H \left(\sum_{j=0}^{\infty} \frac{(-i\alpha Zt)^j}{j!} \right) H \\ &= H e^{-i\alpha Zt} H, H \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \end{aligned} \quad (14)$$

Therefore, the circuit of the mixer operator, also known as the Pauli-X operator, becomes

$$U_M^{\text{BD}}(\alpha_m) = e^{-i\alpha_m X} = H e^{-i\alpha_m Z} H. \quad (15)$$

The equivalent circuit is shown in Figure 7.

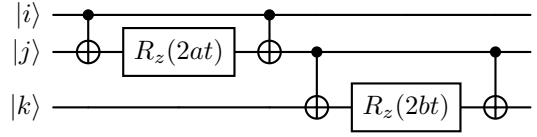


Fig. 6: QAOA commuted Hamiltonian circuit

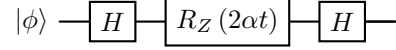


Fig. 7: QAOA mixer (Pauli-X) operator circuit

3) *Ansatz of QAOA*: The QAOA *ansatz*, or the trial state, is constructed through the application of a sequence of unitary operations commencing from a chosen initial state, which is typically the uniform superposition of all computational basis states:

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=1}^{2^n} |x\rangle. \quad (16)$$

At its heart, QAOA uses a variational approach wherein an *ansatz* state is prepared using a sequence of parameterized quantum gates. The *ansatz* for QAOA is given by:

$$|\psi(\alpha, \gamma)\rangle = \prod_{p=1}^P e^{-i\gamma_p H_C} e^{-i\alpha_p H_M} |\phi\rangle, \quad (17)$$

where $|\phi\rangle$ is an initial state, that is typically the equi-probable superposition of all possible bit-strings. Here, H_M is the mixing Hamiltonian, which promotes exploration of the solution space, and the parameters α and γ are continuously varied for minimizing the expectation value of the cost Hamiltonian H_C with respect to the *ansatz*. The depth of the QAOA circuit is represented by p , and a higher p typically provides a more accurate approximation to the problem solution, but also requires a deeper quantum circuit. Therefore, an optional array of parameter values, as the initial point, may be provided as the starting α and γ parameters for the QAOA *p-ansatz*. After preparing the *p-ansatz* state on a quantum computer, the expectation value is evaluated, and classical computers are utilized for optimizing the parameters for subsequent iterations, aiming for finding the optimum.

4) *Complete Circuit*: For the final circuit, we combine H_C and H_M as shown in (17) for *p-ansatz*, and H_M is as shown in Figure 7. For H_C , all non-zero linear terms of the coefficients of the QUBO matrix in (5) are turned into quantum gates as shown in Figure 4. For all non-zero quadratic exponential terms, we will use the quantum gates of Figures 5 and 6 for implementing them. The complete circuit is shown in Figure 8.

C. Proposed Algorithm

In our proposed method, we use the IBM Qiskit digital quantum computer to solve the ILP. For a good QAOA model, we also have to carefully adjust the parameters and penalties. With a relatively high likelihood, the digital quantum solution will provide the correct answer, if the parameters and penalties are adjusted appropriately. Therefore, **Algorithm 1** shows the detail of our proposed QAOA-assisted BD approach. Figure 9 displays the flow chart of **Algorithm 1**.

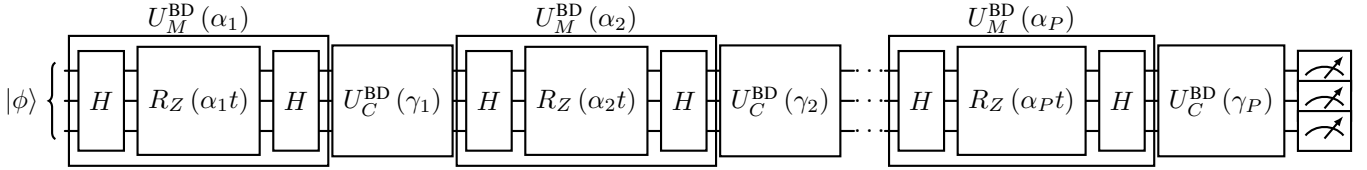


Fig. 8: QAOA complete quantum circuit

Algorithm 1 QAOA-assisted BD Algorithm

Initial $\hat{K}$, $\hat{J}$ of extreme points and rays, and $[\bar{\lambda}, \underline{\lambda}]$ [8].
while $|\bar{\lambda} - \underline{\lambda}| \geq \epsilon$ **do**
 $[\mathbf{P}, \mathbf{Q}] \leftarrow$ Appropriate penalties numbers or arrays and the QUBO formulation by using corresponding rules in [8]
 Build the corresponding QAOA circuit (Figure 8) according to III-A and III-B.
 $[\mathbf{x}', \hat{\lambda}, \bar{\lambda}] \leftarrow$ Solve MAP by digital quantum computers. The multi-cut strategy introduced by [9], [17] can be applied here to accelerate the iteration.
 $[\underline{\lambda}, \mathbf{y}, j \text{ or } k] \leftarrow$ Calculate SUB(s) by classical computer(s), and update the extreme ray set $\hat{J}$ and extreme point set $\hat{K}$ according to [8].
end while
return $[\bar{\lambda}, \underline{\lambda}, \mathbf{x}^*, \mathbf{y}^*]$

D. Applications to Communications and Networking

As we introduced in Section I, MILP has wide applications in communication and networking. For example, [19] constructs a MILP system model for multi-user mobile edge computing (MEC) in the face of interference-infested channels, where the optimization problems are MILPs. There are also other MILP applications in UAV communication scenarios [1], space-air-sea data computation [2], and beam-forming in multi-user communications [3]. Our proposed method can be used for all these applications. In [20], the authors have shown the potential of QAOA in maximum likelihood detection problems. Based on recent QAOA research (reviewed in Section I) QAOA has shown promise in solving optimization problems. As the authors of [21] state, a MILP problem can be solved using BD. Inspired by this, with the cooperation of both classical and digital quantum computers, the QAOA-assisted BD framework proposed can be used for solving optimization problems in the applications mentioned above. Therefore, we believe that the QAOA-assisted BD algorithm has great potential in solving MILPs in communication and networking.

IV. NUMERICAL VALIDATION

We validate the proposed algorithm by running it on IBM's digital quantum processing units (QPUs) [22]. It is worth noting that the field of quantum computing is rapidly evolving, and the maximum number of qubits is increasing.

A. Simulation Setup

In our real-world QPU experiments, we consider a MILP based on (1) to test our proposed quantum algorithm, where we have $\mathbf{x} \in \mathbb{B}^2$, $\mathbf{y} \in [0, 1]^4$. Since the number of qubits is limited and the slack variable will also take up some qubit space, two

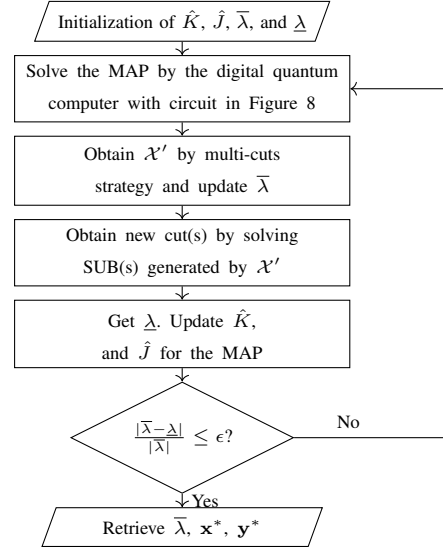


Fig. 9: QAOA-assisted Benders' decomposition flow chart
qubits are assigned to $\mathbf{x}$, three qubits are assigned to the integer part of $\hat{\lambda}$ and one qubit is assigned to the decimal part of $\hat{\lambda}$. The rest of the qubits are reserved for slack variables. We also test a more sizeable scenario associated with h_s and c_s on the IBM Qiskit quantum simulator, which is a classical computer.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ -1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{h}_q^T = [1 \quad 2 \quad 1 \quad 2], \quad \mathbf{c}_q^T = [-1 \quad -2],$$

$$\mathbf{h}_s^T = [1.5 \quad 1.5 \quad 1 \quad 1], \quad \mathbf{c}_s^T = [-1.5 \quad -1].$$

Based on (1), $\mathbf{h}$ and $\mathbf{c}$ represent the coefficients of the continuous and integer decision variables, respectively, in the objective function. Similarly, $\mathbf{G}$ and $\mathbf{A}$ denote the coefficients of continuous and integer decision variables on the left-hand side of the constraints. The right-hand side of the constraints is associated with $\mathbf{b}$.

B. Simulation Results

Both the experiments and simulation results indicate that our proposed QAOA-assisted BD algorithm performs well and obtains the right cuts each time. Hence, the MAP's constraints are increased by the optimality and feasibility cuts in each iteration. Figure 10 shows the test case run on the IBM digital quantum machine and Figure 11 shows the test case run on the IBM quantum simulator, which is a classical computer. In

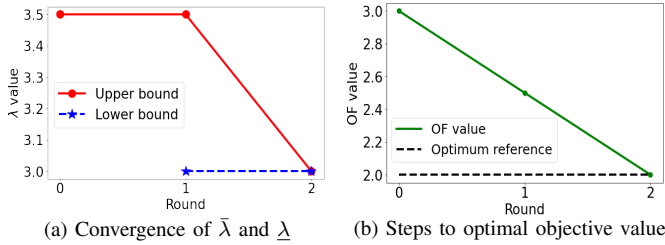


Fig. 10: The test case on the IBM digital quantum machine.

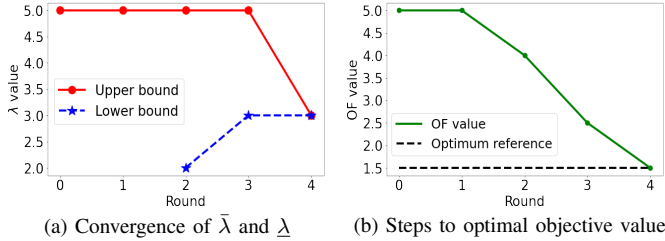


Fig. 11: The test case on the IBM quantum simulator.

Figure 10a, the test case requires 3 rounds for λ to converge. By contrast, as shown in Figure 11a, the simulator needs 5 rounds to converge. The feasible region of both test cases is tightened by adding cut(s) iteratively in the space. Each round's ideal solution can be found by the algorithm. As a result, our QAOA-assisted BD algorithm is dependable and effective. The parts of the dashed lines in Figures 10a and 11a are invisible, because the lower bound in the respective round is negative infinity. These graphs illustrate how the upper and lower bounds converge. The non-negative lower bound can be found by our algorithm in just one round. Meanwhile, Figures 10b and 11b indicate how the objective function settles on its optimal value for both cases. These outcomes demonstrate the mathematical consistency between our suggested algorithm and the classical BD algorithm. Put differently, our algorithm can accomplish the same goal as the classical BD algorithm.

V. CONCLUSION

In this paper, we developed a method that converts the MAP of BD to a problem-specific QAOA *ansatz*, which is a kind of digital quantum circuit. We constructed the circuit of QAOA-assisted BD for digital quantum computers. As the circuit, we harness cascaded Z operators and a mixer operator relying on the measurement of the Pauli-rotation- Z operator. We successfully demonstrated that our algorithm succeeds in converging to the correct final result, as the classical algorithm does. In our performance evaluation, we used both IBM Qiskit's digital quantum computers and quantum simulators to solve MILP problems. In general, our algorithm is suitable for solving MILP problems in communication and networking.

ACKNOWLEDGEMENT

This work is partially supported by NSF CNS-2107216, CNS-2128368, CMMI-2222810, ECCS-2302469, US Department of Transportation, Toyota, Amazon, the Engineering and Physical Sciences Research Council projects EP/W016605/1, EP/X01228X/1, and EP/Y026721/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

REFERENCES

- [1] J. Yao and N. Ansari, "QoS-aware rechargeable UAV trajectory optimization for sensing service," in *IEEE International Conference on Communications (ICC)*, Shanghai, China, Jul. 2019.
- [2] S. S. Hassan, Y. K. Tun, W. Saad, Z. Han, and C. S. Hong, "Blue data computation maximization in 6G space-air-sea non-terrestrial networks," in *IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, Feb. 2021.
- [3] B. Di, H. Zhang, L. Li, L. Song, Y. Li, and Z. Han, "Practical hybrid beamforming with finite-resolution phase shifters for reconfigurable intelligent surface based multi-user communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4565–4570, Feb. 2020.
- [4] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, Oct. 1972.
- [5] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
- [6] S. Kasi, A. K. Singh, D. Venturelli, and K. Jamieson, "Quantum Annealing for large MIMO downlink vector perturbation precoding," in *IEEE International Conference on Communications (ICC)*, Montreal, Canada, Jun. 2021.
- [7] W. Xuan, Z. Zhao, L. Fan, and Z. Han, "Minimizing delay in network function visualization with quantum computing," in *IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, Los Alamitos, CA, Oct. 2021, pp. 108–116.
- [8] Z. Zhao, L. Fan, and Z. Han, "Hybrid quantum Benders' decomposition for mixed-integer linear programming," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, TX, May 2022, pp. 2536–2540.
- [9] X. Wei, L. Fan, Y. Guo, Y. Gong, Z. Han, and Y. Wang, "Quantum assisted scheduling algorithm for federated learning in distributed networks," in *32nd International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, Sep. 2023.
- [10] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," *arXiv preprint arXiv:1411.4028*, Nov. 2014, arXiv: 1411.4028.
- [11] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, Y. Xiao, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, Aug. 2021.
- [12] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 43001, Nov. 2019.
- [13] E. Farhi and A. W. Harrow, "Quantum Supremacy through the Quantum Approximate Optimization Algorithm," *arXiv preprint arXiv:1602.07674*, 2016.
- [14] A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, vol. 2, pp. 1–14, Jan. 2014.
- [15] W. van Dam, K. Eldefrawy, N. Genise, and N. Parham, "Quantum optimization heuristics with an application to knapsack problems," in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, Broomfield, CO, Nov. 2021, pp. 160–170.
- [16] E. Pelofske, A. Bärttschi, and S. Eidenbenz, "Quantum Annealing vs. QAOA: 127 qubit higher-order Ising problems on NISQ computers," in *International Conference on High Performance Computing*, Cham, Switzerland, May 2023, pp. 240–258.
- [17] Z. Zhao, L. Fan, and Z. Han, "Optimal data center energy management with hybrid quantum-classical multi-cuts Benders' decomposition method," *IEEE Transactions on Sustainable Energy*, Aug. 2023, Early access.
- [18] W. Scherer, *Mathematics of quantum computing*. Springer, 2019, vol. 11.
- [19] X. Chen, Y. Cai, L. Li, M. Zhao, B. Champagne, and L. Hanzo, "Energy-efficient resource allocation for latency-sensitive mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2246–2262, Dec. 2019.
- [20] J. Cui, G. L. Long, and L. Hanzo, "General Hamiltonian representation of ML detection relying on the Quantum Approximate Optimization Algorithm," *arXiv preprint arXiv:2204.05126*, Apr. 2022.
- [21] L. Fan and Z. Han, "Hybrid quantum-classical computing for future network optimization," *IEEE Network*, vol. 36, no. 5, pp. 72–76, Nov. 2022.
- [22] "Qiskit." [Online]. Available: <https://qiskit.org>