

Hybrid Quantum-Classical Computing via Dantzig-Wolfe Decomposition for Integer Linear Programming

Xinliang Wei*, Jiyao Liu*, Lei Fan^{†‡}, Yuanxiong Guo[§], Zhu Han[‡], Yu Wang*

*Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA

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

[‡]Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA

[§]Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX, USA
{xinliang.wei, jiyao.liu, wangyu}@temple.edu, {lfan8, zhan2}@uh.edu, yuanxiong.guo@utsa.edu.

Abstract—Numerous optimization scenarios such as industrial production planning, network communication routing, and logistic scheduling can be modeled as large-scale integer linear programming problems. However, due to the NP-Hardness of these problems, it is very challenging to optimally solve these problems in a short time on classical computers. Quantum computers have emerged as a new computing platform to provide new computing paradigms to tackle these problems. However, the scalability and efficiency of current quantum computers pose significant challenges in practical implementations of quantum optimization algorithms. In this paper, we propose a novel hybrid quantum-classical approach, termed Hybrid quantum-classical Dantzig-Wolfe Decomposition (HyDWD), aimed at solving these problems. In this framework, the subproblems can be solved in parallel on quantum computers. Our results demonstrate the benefits of integrating parallel quantum computing with the proposed hybrid quantum-classical framework via Dantzig-Wolfe decomposition, paving the way for advancements in optimization and decision-making processes.

Index Terms—Quantum computing, Dantzig-Wolfe decomposition, hybrid quantum-classical method, integer linear programming, quantum network

I. INTRODUCTION

Quantum computing [1], [2] has emerged as a promising technology with the potential to revolutionize problem-solving in various domains by harnessing the principles of quantum mechanics. Quantum computing offers the capability to process vast amounts of data and solve complex problems at significantly accelerated speeds compared to classical computers [3]–[10]. However, its scalability and efficiency remain crucial challenges in practically implementing quantum algorithms on existing quantum computers. For example, most current circuit-based quantum processors can only support up to around 1,000 qubits, while annealing quantum processors can reach around 5,000 qubits.

The work is partially supported by the US National Science Foundation (under Grant No. CNS-2006604, CNS-2106761, CNS-2107216, CNS-2128368, CNS-2128378, CMMI-2222670, CMMI-2222810, and ECCS-2302469), the US Department of Transportation, Toyota, Amazon, and Japan Science and Technology Agency (JST) Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) under JPMJAP2326.

To overcome the limitation of current state of quantum computers, hybrid quantum-classical (HQC) computing framework [11]–[15] has been developed for different applications [15]–[21]. For example, several HQC approaches [12], [15], [18] leverage both quantum and classical computers to solve large-scale mixed-integer linear programming (MILP), and such methods have been used for solving many challenging scheduling or resource allocation problems in production scheduling [15], wireless networks [18], power systems [19], and distributed learning systems [20], [21]. In these methods, the original MILP problem is decomposed into the master problem and subproblems via various Benders’ decomposition techniques (BD), and then the master problem is solved by quantum annealer (QA) or quantum approximate optimization algorithm (QAOA) while the subproblems are solved by classical computers. In addition, HQC approaches via QAOA and quantum variational circuits (QVCs) have also been applied to tackle different optimization [22], [23] and machine learning [24] problems. However, all these HQC computing frameworks cannot be applied to large-scale constrained pure integer linear programming problems, which are generally NP-hard problems.

In this paper, by leveraging the parallel computing capability of QC, we focus on designing a new hybrid quantum-classical computing method to solve the large-scale constrained pure integer linear problem (ILP). We explore the synergistic integration of quantum computing and distributed computing and propose a new hybrid quantum-classical framework based on Dantzig-Wolfe (DW) decomposition for solving complex ILP problems. By leveraging quantum distributed computing with DW decomposition, we aim to enhance the scalability and efficiency of solving ILP problems, paving the way for new advancements in optimization and decision-making processes.

When the linear programming model has a specific block structure, the model can be decomposed through DW decomposition proposed by [25] to facilitate the solution of large-scale linear programs. This method is further extended to solve integer linear programming models with such structures [26].

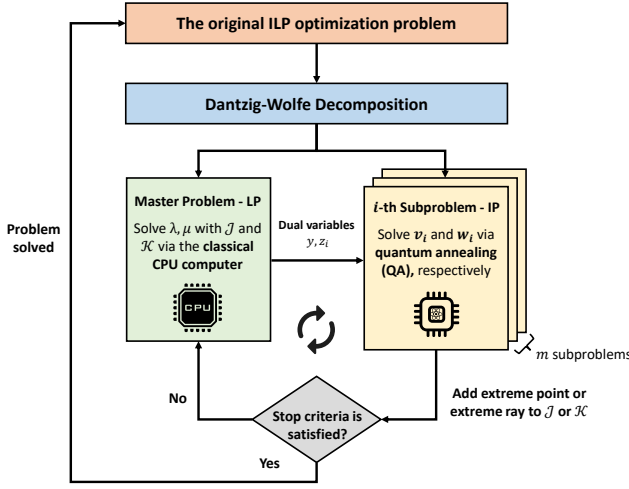


Fig. 1. Proposed HyDWD framework for ILP problem.

Particularly, we develop a novel HQC approach, termed *Hybrid quantum-classical Dantzig-Wolfe Decomposition* (HyDWD) method. As shown in Fig. 1, in HyDWD, the master problem constitutes a linear programming problem solvable on a classical computer, while the sub-problem manifests as a binary optimization problem amenable to resolution via quantum annealing (QA). Furthermore, as the subproblem can be further decomposed into multiple subproblems, each smaller subproblem can be addressed on a quantum machine leveraging QA. This framework integrates parallel quantum computing techniques, thereby offering potential improvements in computational efficiency and solution quality for the overall HQC framework. To illustrate such advances, we use an optimal entanglement distribution problem in satellite-assisted quantum networks as a case study and conduct numerical validation via various settings and a hybrid D-Wave quantum annealer. Our simulation results demonstrate the clear benefits of integrating parallel distributed quantum computing with the proposed HQC framework via DW decomposition.

The remainder of this paper is organized as follows. Section II presents an overview of the related works in HQC. In Section III, we briefly review the classical integer linear programming problem and its Dantzig-Wolfe decomposition. Section IV introduces the proposed HyDWD algorithm. In Section V, we use an optimal entanglement distribution problem as a case study to show the usage of the proposed HyDWD with performance evaluation. Finally, we conclude the paper with a discussion of possible applications and future works in Section VI.

II. RELATED WORKS

Quantum computing has demonstrated superiority in solving numerous computationally intensive problems [3], [8]–[10]. However, its application is constrained by the current state of quantum computers, including factors such as availability, scalability, robustness, and cost.

To tackle these challenges, researchers have developed a hybrid quantum-classical computing framework aimed at

addressing complex optimization tasks by harnessing the strengths of both quantum and classical computers. This hybrid quantum-classical optimization approach has found applications in various domains, such as machine learning, distributed computing, wireless communication, and task scheduling [12]–[18], [20]. For example, Tran et al. [16] introduced an HQC approach to solving complex scheduling problems (such as graph-coloring-type scheduling, Mars Lander task scheduling, and airport runway scheduling), decomposing them into the master problem and subproblem, with a quantum annealer solving the master problem while the subproblem and the global search tree was solved and managed by a classical computer. Ajagekar et al. [15] also proposed HQC-based optimization techniques for solving scheduling problems in manufacturing systems which can be modeled as MILP and mixed-integer fractional programming. Similarly, Zhao et al. [12], [18] devised an HQC algorithm using a different Benders' decomposition technique to solve the MILP problem, with applications in distributed learning [20], [21]. In these methods, the master problem is tackled by a quantum annealer, while Zhao et al. [13] employed a quantum approximate optimization algorithm (QAOA) to assist in solving the master problem within the Benders' decomposition framework. Paterakis [17] also provided an HQC method for unit commitment problems, presenting a method to control the size of the master problem using various cut selection criteria. Our proposed HyDWD method also operates within an HQC framework but adopts a distinct decomposition technique, DW decomposition, tailored specifically for ILP problems.

One of the advantages of our proposed HyDWD framework is allowing the subproblem to be decomposed into multiple smaller subproblems, each can be addressed by a quantum machine in parallel. This makes the potential usage of distributed quantum computing, offer better efficiency and scalability. Distributed quantum computing [27], [28] represents a paradigm shift in quantum computing, where the processing power of multiple quantum computing nodes is harnessed to solve complex computational problems collectively. By leveraging a network of interconnected quantum computing nodes, distributed quantum computing may overcome the inherent limitations of a single centralized quantum computer (such as the susceptibility to errors, resource constraints, and scalability challenges), provide better usage of the distributed quantum computing resources and improved fault tolerance, and support large-scale collaborative computing tasks.

III. DANTZIG-WOLFE DECOMPOSITION FOR ILP

In this section, we briefly review the Dantzi-Wolf decomposition algorithm for ILP problems.

A. Integer Linear Programming

Due to the flexibility of the integer linear programming modeling techniques, they have been broadly applied to many industrial applications such as vehicle routing, manufacture planning, and network communication. To apply the Dantzi-

wolf decomposition, we consider an integer programming model with a special structure as

$$\min_{\mathbf{x}} \sum_{i \in \mathcal{I}} \mathbf{c}_i^\top \mathbf{x}_i \quad (1)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} \mathbf{A}_{0,i} \mathbf{x}_i = \mathbf{b}_0, \quad (1a)$$

$$\mathbf{x}_i \in \mathbf{X}_i, \quad \forall i \in \mathcal{I}, \quad (1b)$$

$$\mathbf{x}_i \geq \mathbf{0}, \quad \forall i \in \mathcal{I}. \quad (1c)$$

Here, $\mathcal{I} = \{1, 2, \dots, m\}$. $\mathbf{x}_i$ is a nongative vector with integer variables and $\mathbf{X}_i = \{\mathbf{A}_{i,i} \mathbf{x}_i = \mathbf{b}_i\}$. $\mathbf{c}_i^\top$ is its corresponding coefficient vector in the objective function. $\mathbf{A}_{0,i}$ and $\mathbf{A}_{i,i}$ are the coefficient matrix of variable $\mathbf{x}$ in constraints, while $\mathbf{b}_0$ and $\mathbf{b}_i$ are the constant vector in constraints.

In the model in (1), there are m subgroups of decision variables. The linking constraint (1a) combined all the decision variables together. The Dantzig-wolf decomposition algorithm will utilize this characteristic to decouple this model into m subproblems.

B. Dantzig-Wolfe (DW) Decomposition

The DW decomposition method first reformulates the model of (1) into a model of (4) using the convex combination of the extreme points and extreme rays of all the subsets (i.e., P_i).

1) *Master Problem*: Let $P_i = \{\mathbf{x}_i \geq \mathbf{0} \mid \mathbf{A}_{i,i} \mathbf{x}_i = \mathbf{b}_i\}$. If P_i is bounded, then $\mathbf{x}_i$ can be expressed as a convex combination of the vertices of P_i . Let $\mathcal{P}_x = \{\mathbf{v}_i^{(j)}, \forall j \in \mathcal{J}\}$ be the *extreme point* set of P_i , then there exists $\lambda_{i,j} \geq 0$ and $\sum_{j \in \mathcal{J}} \lambda_{i,j} = 1$, we have

$$\mathbf{x}_i = \sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{v}_i^{(j)}. \quad (2)$$

In general, if P_i is bounded or unbounded, $\mathbf{x}_i$ can also be represented by a linear combination of vertices and polar rays of P_i . Let $\mathcal{Q}_x = \{\mathbf{w}_i^{(k)}, \forall k \in \mathcal{K}\}$ be the *extreme ray* set of P_i , and then there exists $\lambda_{i,j} \geq 0$ and $\sum_{j \in \mathcal{J}} \lambda_{i,j} = 1$, $\mu_{i,k} \geq 0$. Consequently, we have

$$\mathbf{x}_i = \sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{v}_i^{(j)} + \sum_{k \in \mathcal{K}} \mu_{i,k} \mathbf{w}_i^{(k)}. \quad (3)$$

Therefore, the original ILP problem of (1) can be written equivalently to a master problem as

$$\min_{\lambda, \mu} \sum_{i \in \mathcal{I}} \left(\sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{c}_i^\top \mathbf{v}_i^{(j)} + \sum_{k \in \mathcal{K}} \mu_{i,k} \mathbf{c}_i^\top \mathbf{w}_i^{(k)} \right) \quad (4)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} \left(\sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{A}_{0,i} \mathbf{v}_i^{(j)} + \sum_{k \in \mathcal{K}} \mu_{i,k} \mathbf{A}_{0,i} \mathbf{w}_i^{(k)} \right) = \mathbf{b}_0, \quad (4a)$$

$$\sum_{j \in \mathcal{J}} \lambda_{i,j} = 1, \quad \forall i, \quad (4b)$$

$$\lambda_{i,j}, \mu_{i,k} \geq 0, \quad \forall i, j. \quad (4c)$$

Here, $\lambda_{i,j}$ and $\mu_{i,k}$ are decision variables with positive value. If P_i is bounded, problem (4) can be reduced as follows

$$\min_{\lambda} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{c}_i^\top \mathbf{v}_i^{(j)} \quad (5)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \lambda_{i,j} \mathbf{A}_{0,i} \mathbf{v}_i^{(j)} = \mathbf{b}_0, \quad (5a)$$

$$\sum_{j \in \mathcal{J}} \lambda_{i,j} = 1, \quad \forall i, \quad (5b)$$

$$\lambda_{i,j} \geq 0, \quad \forall i, j. \quad (5c)$$

However, it is very challenging to directly obtain all the extreme points to formulate the model of (4), since there are a large number of extreme points. Instead of getting all the extreme points in one shot, we start with a limited number of the extreme points. Then, we gradually bring these extreme points back to the model of (4) by solving subproblems (8).

2) *Subproblem*: Let $y \in \mathbb{R}$ and $z_i \in \mathbb{R}$ be the dual variables of constraints (4a) and (4b), respectively. Then the reduced cost of $\lambda_{i,j}$ and $\mu_{i,k}$ can be expressed as

$$\alpha_{i,j} = \mathbf{c}_i^\top \mathbf{v}_i^{(j)} - y^\top \mathbf{A}_{0,i} \mathbf{v}_i^{(j)} - z_i, \quad (6)$$

$$\beta_{i,k} = \mathbf{c}_i^\top \mathbf{w}_i^{(k)} - y^\top \mathbf{A}_{0,i} \mathbf{w}_i^{(k)}, \quad (7)$$

where $\alpha_{i,j}$ and $\beta_{i,k}$ are the *reduced cost* related to y and z_i . When $\alpha_{i,j}, \beta_{i,k} \geq 0$, an optimal solution is found for the original problem. Therefore, in the subproblem, we need to calculate $\mathbf{v}_i^{(j)}$ (or $\mathbf{w}_i^{(k)}$), so that the values of $\alpha_{i,j}$ and $\beta_{i,k}$ are as small as possible. The i -th subproblem is defined as

$$i\text{-th subproblem:} \min_{\mathbf{x}_i} f_i := (\mathbf{c}_i^\top - \mathbf{A}_{0,i}^\top y)^\top \mathbf{x}_i \quad (8)$$

$$\text{s.t.} \mathbf{A}_{i,i} \mathbf{x}_i = \mathbf{b}_i, \quad (8a)$$

$$\mathbf{x}_i \geq \mathbf{0}. \quad (8b)$$

Here, we consider three cases when solving the subproblem.

- 1) The optimal target value is $-\infty$. In this case, $\alpha_{i,j}, \beta_{i,k} < 0$ and the solution of the subproblem is the extreme ray $\mathbf{w}_i^{(k)}$, and then we add $\mathbf{A}_{0,i} \mathbf{w}_i^{(k)}$ to the master problem for next round calculation.
- 2) The optimal target value is bounded and $f_i < z_i$. In this case, $\alpha_{i,j} < 0$ and the solution of the subproblem is the extreme point $\mathbf{v}_i^{(j)}$, and then we add $\mathbf{A}_{0,i} \mathbf{v}_i^{(j)}$ to the master problem for next round calculation.
- 3) The optimal target value is bounded and $f_i \geq z_i$. In this case, $0 \leq \alpha_{i,j} \leq \beta_{i,k}$, and $z_i \geq 0$, and therefore we find the optimal solution.

IV. HYBRID QUANTUM-CLASSICAL SOLUTION: HYDWD

In the last section, the original problem in (1) is decomposed into a master problem of (4) and several subproblems (8) by the DW decomposition method. The master problem is a classical LP problem with continuous variables and can be solved by a classical solver, such as Gurobi and Scipy, etc. The subproblem is an IP problem and we will solve it via the quantum annealing technique. We also consider solving the subproblem in a distributed manner to save the quantum

resource and handle the large-scale scenario, as shown in Fig. 1. In the following subsections, we will discuss the quadratic unconstrained binary optimization (QUBO) formulation for the subproblem and propose the hybrid quantum-classical algorithm.

A. QUBO Formulation

In a QUBO problem, there's typically a set of binary variables represented by vector $\mathbf{x}$ and an upper-diagonal matrix denoted as $\mathbf{Q}$, which is an $N' \times N'$ matrix with upper-triangular properties. The objective of QUBO is to minimize the following function:

$$\min_{\mathbf{x} \in \{0,1\}^{N'}} \mathbf{x}^\top \mathbf{Q} \mathbf{x}. \quad (9)$$

This problem can be solved by quantum computers such as quantum annealer. The quantum annealer embeds this problem into the energy spectrum of the quantum system. After quantum evolution, the quantum machine can return the ground energy and its eigenvector. The ground energy value is the optimal objective value of the model of (9) and its eigenvector is the optimization solution of the model of (9).

It is noted that the subproblem in (8) is an ILP problem with constraint (8a). Therefore, we must rewrite the subproblem and formulate it into QUBO format. To do so, we introduce the *penalty* for all constraints by following the principle of constraint-penalty pairs in [29]. The constraint (8a) can be converted as follows

$$(8a) \Rightarrow \xi_i : P_i(\mathbf{A}_{i,i}\mathbf{x}_i - \mathbf{b}_i)^2.$$

Here, P_i is the predefined penalty coefficient when the corresponding constraint is violated. Then the reformulated unconstrained subproblem is defined below

$$i\text{-th subproblem} : \min_{\mathbf{x}_i} f_i := (\mathbf{c}_i^\top - \mathbf{A}_{0,i}^\top \mathbf{y})^\top \mathbf{x}_i + \xi_i. \quad (10)$$

Now, the subproblem in (10) is a pure unconstrained linear integer problem that can be conveniently mapped into the QUBO form as discussed before. In addition, when these penalty coefficients are bounded, we can find the best penalty coefficients for each constraint using binary search.

B. Proposed Algorithm - HyDWD

We now introduce the proposed *Hybrid quantum-classical Dantzig-Wolfe Decomposition* (HyDWD) algorithm, which iteratively solves the master problem in (4) and the subproblem in (10) until convergence is reached. Moreover, all subproblems can be solved in a distributed manner through distributed quantum computing rather than a centralized manner as illustrated in Fig. 1.

Algorithm 1 gives the detailed HyDWD algorithm. Given input parameters including a coefficient vector $\mathbf{c}$, a coefficient matrix $\mathbf{A}$, a constant vector $\mathbf{b}$, and an upper bound m of i , the HyDWD algorithm aims to determine the optimal solution $\mathbf{x}$. The algorithm initializes the extreme points and rays, along with the dual values, and proceeds through a series of iterations (Lines 1-4). During each iteration, distributed

Algorithm 1 Hybrid quantum-classical Dantzig-Wolfe Decomposition (HyDWD) algorithm

Input: coefficient vector $\mathbf{c}$, coefficient matrix $\mathbf{A}$, constant vector $\mathbf{b}$, the upper bound m of i .

Output: $\mathbf{x}$

```

1:  $\mathbf{v}_i^{(0)}, \mathbf{w}_i^{(0)} \leftarrow$  Initialize the extreme points and rays
2:  $\mathcal{J}, \mathcal{K} \leftarrow \mathbf{v}_i^{(0)}, \mathbf{w}_i^{(0)}$ 
3:  $\mathbf{y}^{(0)}, \mathbf{z}_i^{(0)} \leftarrow$  Initialize the dual values
4:  $max\_itr \leftarrow 100, t \leftarrow 0$ 
5: while  $t < max\_itr$  do
6:   for  $i = 1$  to  $m$  do in parallel
7:      $\mathbf{v}_i^{(t)}, \mathbf{w}_i^{(t)} \leftarrow$  solve (10) with  $\mathbf{y}^{(t)}, \mathbf{z}_i^{(t)}$  using quantum annealing
8:     if  $\alpha_{i,j} < 0$  and  $\beta_{i,k} < 0$  then
9:        $\mathcal{K} \leftarrow \mathbf{w}_i^{(t)}$   $\triangleright$  adding the extreme ray to  $\mathcal{K}$ 
10:    end if
11:    if  $\alpha_{i,j} < 0$  and  $\beta_{i,k} \geq 0$  then
12:       $\mathcal{J} \leftarrow \mathbf{v}_i^{(t)}$   $\triangleright$  adding the extreme point to  $\mathcal{J}$ 
13:    end if
14:  end for
15:  if all  $0 \leq \alpha_{i,j} \leq \beta_{i,k}$  then
16:    Stop the loop.
17:  end if
18:   $\lambda_{i,j}, \mu_{i,k} \leftarrow$  solve (4) with  $\mathbf{v}_i^{(j)}, \mathbf{w}_i^{(k)}$  using classical CPU computer
19:   $t \leftarrow t + 1$ 
20:   $\mathbf{y}^{(t)}, \mathbf{z}_i^{(t)} \leftarrow$  get dual solution from (4)
21: end while
22:  $\mathbf{x} \leftarrow$  calculate from  $\mathbf{v}^{(j)}, \mathbf{w}^{(k)}$ 
23: return  $\mathbf{x}$ 

```

quantum computing techniques are employed to update the extreme points and rays based on the dual values. The algorithm then evaluates specific conditions to adjust the extreme points and rays as necessary (Lines 5-17). Specifically, if reduced costs $\alpha_{i,j} < 0$ and $\beta_{i,k} < 0$, then the extreme ray is added to the master problem (Lines 8-10). If the reduced cost $\alpha_{i,j} < 0$ and $\beta_{i,k} \geq 0$, an extreme point is added to the master problem (Lines 11-13). If all $\alpha_{i,j}$ and $\beta_{i,k}$ are greater or equal to 0, the optimal solution is found and we stop the loop (Lines 15-17). Subsequently, a classical computing step solves a master problem to derive updated dual values (Lines 18). This iterative process continues until a predefined stopping criterion is satisfied. Finally, the algorithm computes the optimal solution $\mathbf{x}$ using the obtained extreme points and rays (Lines 20).

V. CASE STUDY: SATELLITE-BASED ENTANGLEMENT DISTRIBUTION FOR QUANTUM NETWORKS

We now present a specific case study, i.e., the optimal satellite-based entanglement distribution problem in quantum networks, to show the advantage of the proposed HyDWD algorithm.

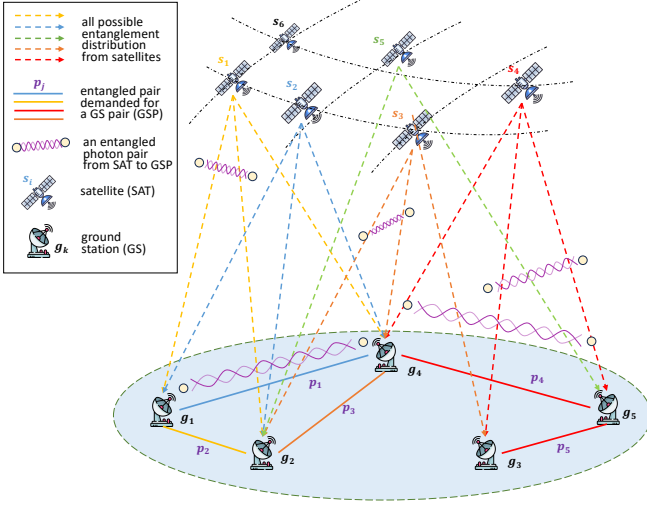


Fig. 2. The overall satellite-based entanglement distribution architecture.

A. The Optimization Problem

In satellite-assisted quantum networks [30], [31], satellites equipped with entanglement-generating photon sources can create pairs of entangled photons and distribute them between a pair of ground stations, as illustrated in Fig. 2. With these pre-distributed pairs, ground stations can then quickly generate new entangled pairs for their communication, thus significantly reducing the communication latency even over long-distance links [32], [33].

With a large constellation of satellites, we now can generate and distribute sufficient entangled pairs for the larger quantum network, however, deciding the optimal assignment of these satellites to meet entanglement distribution demands at various ground stations becomes a challenging optimization problem [30], [34], [35].

We consider a space-terrestrial network with $|S|$ satellites (SAT) and $|G|$ ground stations (GS). The sets $S = \{s_i\}$ and $G = \{g_k\}$ represent the satellites and GSes. We define $P = \{p_j\}$ as the set of demanded ground station pairs (GSPs) requiring pre-distributed entangled pairs, where $p_j = \{g_{k1}, g_{k2}\}$ and g_{k1}/g_{k2} are the two GSes in this pair. Let P_{g_k} represent the set of pairs which includes the station g_k . Each satellite s_i has an entangled photon source and T_i transmitters to send entangled photon pairs to multiple GSPs. Each GS g_k has R_k receivers to receive photons and create entanglement for quantum applications. Moreover, both SATs and GSes have a quantum memory to store entangled photons, represented by sm_i and gm_k , respectively. Let $sm_{max} = \max_i \{sm_i\}$, which is the maximum size of memory among all SATs.

We define an *optimal entanglement distribution problem* in this satellite-assisted quantum network, which aims to assign satellites to cover demanded GSPs and allocate appropriate entangled photon resources to each assignment. Let $x_{i,j}$ be the binary variables indicating whether SAT s_i is assigned to GSP p_j , $y_{i,j}$ be the integer variable range from 0 to sm_{max} indicating the entangled photon pair allocation between SAT

s_i and GSP p_j , and $w_{i,j}$ be the associated utility representing entanglement generation rate or arrival rate of requests. Then the optimization problem is to assign satellites to cover demanded GSPs with allocated entangled photon resources, in order to maximize the weighted utility. The optimization problem is formulated as follows

$$\max_{x,y} \sum_{p_j \in P} \sum_{s_i \in S} w_{i,j} x_{i,j} y_{i,j} \quad (11)$$

$$\text{s.t.} \sum_{s_i \in S} x_{i,j} \leq L_j, \quad \forall p_j \in P, \quad (11a)$$

$$\sum_{s_i \in S} \sum_{p_j \in P_{g_k}} x_{i,j} \leq R_k, \quad \forall g_k \in G, \quad (11b)$$

$$\sum_{p_j \in P} x_{i,j} \leq T_i, \quad \forall s_i \in S, \quad (11c)$$

$$\sum_{s_i \in S} \sum_{p_j \in P_{g_k}} x_{i,j} y_{i,j} \leq gm_k, \quad \forall g_k \in G, \quad (11d)$$

$$\sum_{p_j \in P} x_{i,j} y_{i,j} \leq sm_i, \quad \forall s_i \in S, \quad (11e)$$

$$x_{i,j} \in \{0, 1\}, \quad y_{i,j} \in \{0, \dots, sm_{max}\}. \quad (11f)$$

Here, Constraint (11a) ensures that each GSP $p_j \in P$ can only connect to L_j satellites simultaneously. Constraint (11b) means that a GS g_k can be part of multiple GSPs and thus is not allowed to be allocated to more than R_k satellites due to its limited number of receivers. Constraint (11c) ensures that SAT s_i does not get allocated to more than T_i GSPs due to its limited number of transmitters on board. Constraint (11d) makes sure that the total entangled photon received from different satellites cannot exceed the maximal quantum memory of GS g_k . Constraint (11e) guarantees that the total entangled photon pair allocation fraction of SAT s_i cannot exceed sm_i .

It is difficult to obtain the optimal solution to this optimization problem since it is a quadratic constrained quadratic discrete optimization problem, which is NP-hard and challenging to solve with classical computing when the problem scale increases. Therefore, DW decomposition is applied to solve the optimization problem due to the specific structure of the problem.

B. Solving the Problem with HyDWD Method

In this subsection, we describe how to use the proposed HyDWD framework to solve the optimal entanglement distribution problem. We first need to linearize the original problem, and then apply the proposed HyDWD method.

1) *Linearization*: Note that problem (11) is an integer non-linear problem due to the quadratic terms in the objective function and constraints. Since DW decomposition is available to solve the ILP problem, we have to linearize the original problem first. We introduce an auxiliary integer variable $\phi_{i,j}$ to linearize quadratic term $x_{i,j} y_{i,j}$. Therefore, the original problem is reformulated as follows

$$\max_{x,y,\phi} \sum_{p_j \in P} \sum_{s_i \in S} w_{i,j} \phi_{i,j} \quad (12)$$

$$\text{s.t.} \sum_{s_i \in S} x_{i,j} \leq L_j, \quad \forall p_j \in P, \quad (12a)$$

$$\sum_{s_i \in S} \sum_{p_j \in P_{g_k}} x_{i,j} \leq R_k, \quad \forall g_k \in G, \quad (12b)$$

$$\sum_{p_j \in P} x_{i,j} \leq T_i, \quad \forall s_i \in S, \quad (12c)$$

$$\sum_{s_i \in S} \sum_{p_j \in P_{g_k}} \phi_{i,j} \leq g m_k, \quad \forall g_k \in G, \quad (12d)$$

$$\sum_{p_j \in P} \phi_{i,j} \leq s m_i, \quad \forall s_i \in S, \quad (12e)$$

$$\phi_{i,j} \leq y_{i,j}, \quad \forall s_i \in S, p_j \in P, \quad (12f)$$

$$\phi_{i,j} \leq x_{i,j} s m_{\max}, \quad \forall s_i \in S, p_j \in P, \quad (12g)$$

$$x_{i,j} + y_{i,j} - s m_{\max} \leq \phi_{i,j}, \quad \forall s_i \in S, p_j \in P, \quad (12h)$$

$$x_{i,j} \in \{0, 1\}, y_{i,j}, \phi_{i,j} \in \{0, \dots, s m_{\max}\}. \quad (12i)$$

Here, Constraints (12f)-(12h) are the linearization of non-linear term $x_{i,j} y_{i,j}$. Given that problem (12) and its constraints are linear in nature, it can be reformulated into a more general form as follows

$$\max_{\mathbf{X}, \mathbf{Y}} \mathbf{h}^\top \mathbf{Y} \quad (13)$$

$$\text{s.t.} \mathbf{A}\mathbf{X} + \mathbf{G}\mathbf{Y} \leq \mathbf{b}_0, \quad (13a)$$

$$\mathbf{B}\mathbf{X} \leq \mathbf{b}_1, \quad (13b)$$

$$\mathbf{C}\mathbf{Y} \leq \mathbf{b}_2, \quad (13c)$$

$$\mathbf{X} = [x]^\top, \quad \mathbf{X} \in \mathcal{X}, \quad (13d)$$

$$\mathbf{Y} = [y, \phi]^\top, \quad \mathbf{Y} \in \mathcal{Y}, \quad (13e)$$

where $\mathbf{h}^\top$ represents the coefficient for integer variables in the objective function. $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{G}$ denote coefficients in the constraints while $\mathbf{b}_0, \mathbf{b}_1$, and $\mathbf{b}_2$ are constant vectors. Additionally, $\mathcal{X}$ and $\mathcal{Y}$ denote the solution sets of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

2) *HyDWD solution*: Given the general form of the optimization problem, we can apply the HyDWD algorithm to solve it. Let $\dim_{\mathbf{X}} = |S| \times |P|$ and $\dim_{\mathbf{Y}} = |S| \times |P| \times 2$. We further reformulate problem (13) by increasing variable constraints to reduce the number of inequality constraints.

By defining $\mathcal{X}$ and $\mathcal{Y}$ as

$$\mathcal{X} = \{\mathbf{X} \in \{0, 1\}^{\dim_{\mathbf{X}}} : \mathbf{B}\mathbf{X} \leq \mathbf{b}_1\}, \quad (14)$$

$$\mathcal{Y} = \{\mathbf{Y} \in \{0, \dots, s m_{\max}\}^{\dim_{\mathbf{Y}}} : \mathbf{C}\mathbf{Y} \leq \mathbf{b}_2\}, \quad (15)$$

the problem (13) can be reformulated as

$$\max_{\mathbf{X}, \mathbf{Y}} \mathbf{h}^\top \mathbf{Y} \quad (16)$$

$$\text{s.t.} \mathbf{A}\mathbf{X} + \mathbf{G}\mathbf{Y} \leq \mathbf{b}_0, \quad (16a)$$

$$\mathbf{X} \in \mathcal{X}, \quad (16d)$$

$$\mathbf{Y} \in \mathcal{Y}. \quad (16e)$$

Let $\mathcal{U}$ be the feasible region of (16), i.e., $\mathcal{U}$ is the set of all possible points of (16) that satisfy the problem's constraints:

$$\mathcal{U} = \{\mathbf{X} \in \mathcal{X}, \mathbf{Y} \in \mathcal{Y} : \mathbf{A}\mathbf{X} + \mathbf{G}\mathbf{Y} \leq \mathbf{b}_0\}. \quad (17)$$

Note that each polyhedron $\mathcal{U}$ can be written as the sum of finitely many extreme points. Therefore, we represent its sets of extreme points with $\mathcal{P}_{\mathcal{X}} = \{\mathbf{X}^{(i)}, \forall i \in \mathcal{I}\}$ and $\mathcal{P}_{\mathcal{Y}} = \{\mathbf{Y}^{(j)}, \forall j \in \mathcal{J}\}$. Note that the index i and j here are different from the index i and j used in the problem (11). Since $\mathcal{I}$ and $\mathcal{J}$ contain an exponential number of extreme points, we further consider a restricted version of extreme points set by progressively putting each new extreme point to the subsets $\mathcal{I}' \subseteq \mathcal{I}$ and $\mathcal{J}' \subseteq \mathcal{J}$. Then we can express the problem (16) as the linear combination of its restricted extreme points:

Restricted Master Problem

$$\max_{\lambda_i, \forall i \in \mathcal{I}', \mu_j, \forall j \in \mathcal{J}'} \sum_{j \in \mathcal{J}} (\mathbf{h}^\top \mathbf{Y}^{(j)}) \mu_j \quad (18)$$

$$\text{s.t.} \sum_{i \in \mathcal{I}'} (\mathbf{A}\mathbf{X}^{(i)}) \lambda_i + \sum_{j \in \mathcal{J}'} (\mathbf{G}\mathbf{Y}^{(j)}) \mu_j \leq \mathbf{b}_0, \quad (18a)$$

$$\sum_{i \in \mathcal{I}'} \lambda_i = 1, \quad (18b)$$

$$\sum_{j \in \mathcal{J}'} \mu_j = 1, \quad (18c)$$

$$\lambda_i \in [0, 1], \quad \forall i \in \mathcal{I}', \quad (18d)$$

$$\mu_j \in [0, 1], \quad \forall j \in \mathcal{J}', \quad (18e)$$

where $\lambda_i, \mu_j \in \mathbb{R}$ represent the weights of each extreme point for binary and integer variables, respectively.

Next, we introduce the Lagrangian relaxation and the problem can be represented as:

$$\begin{aligned} \max_{\lambda_i, \forall i \in \mathcal{I}', \mu_j, \forall j \in \mathcal{J}'} & \alpha \mathbf{b}_0 + \sum_{i \in \mathcal{I}'} (-\alpha \mathbf{A}\mathbf{X}^{(i)} - \xi) \lambda_i + \xi \\ & + \sum_{j \in \mathcal{J}'} ((\mathbf{h}^\top - \alpha \mathbf{G}) \mathbf{Y}^{(j)} - \zeta) \mu_j + \zeta, \end{aligned} \quad (19)$$

where $\alpha \in \mathbb{R}_+$ is the row vector (dual variable) of Constraint (18a) and $\xi, \zeta \in \mathbb{R}$ are Lagrangian multipliers for Constraints (18b) and (18c), respectively. Then the Lagrangian dual problem is as follows

$$\min_{\alpha, \xi, \zeta} \alpha \mathbf{b}_0 + \xi + \zeta \quad (20)$$

$$\text{s.t.} \quad -\alpha \mathbf{A}\mathbf{X}^{(i)} - \xi \geq 0, \quad \forall i \in \mathcal{I}', \quad (20a)$$

$$(\mathbf{h}^\top - \alpha \mathbf{G}) \mathbf{Y}^{(j)} - \zeta \geq 0, \quad \forall j \in \mathcal{J}'. \quad (20b)$$

At each step t , we compute extreme points $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$. These extreme points are incorporated into the master problem, necessitating the addition of new λ_i and μ_j columns. Constraints (20a) and (20b) are called *reduced cost*. Then the two subproblems are given as:

$$\text{Subproblem-}\mathbf{X}: \max_{\mathbf{X} \in \mathcal{X}} -\alpha^{(t)} \mathbf{A}\mathbf{X}, \quad (21)$$

$$\text{Subproblem-}\mathbf{Y}: \max_{\mathbf{Y} \in \mathcal{Y}} (\mathbf{h}^\top - \alpha^{(t)} \mathbf{G}) \mathbf{Y}, \quad (22)$$

where $\alpha^{(t)}$ is the dual variables of Constraint (18a). If the solution of (21) is larger than $\xi^{(t)}$, then we set $\mathcal{I}' \leftarrow \mathcal{X}^{(t)}$. Similarly, if the solution of (22) is larger than $\zeta^{(t)}$, then we let $\mathcal{J}' \leftarrow \mathcal{Y}^{(t)}$.

These two subproblems can be solved in a centralized manner to obtain the optimal value. However, to fully leverage distributed quantum computing, Subproblem- $\mathbf{X}$ and Subproblem- $\mathbf{Y}$ can be decomposed into $|P|$ parts, enabling each GSP to compute the subproblem independently. Therefore, these two subproblems can be written as follows

$$q\text{-th Subproblem-}\mathbf{X}: \max_{\mathbf{X}_q} -\alpha_q^{(t)} \mathbf{A}_q \mathbf{X}_q, \quad (23)$$

$$q\text{-th Subproblem-}\mathbf{Y}: \max_{\mathbf{Y}_q} (\mathbf{h}_q^T - \alpha_q^{(t)} \mathbf{G}_q) \mathbf{Y}_q, \quad (24)$$

where $q \in \{1, \dots, |P|\}$. In doing so, quantum annealing can solve a series of smaller subproblems in a distributed manner to handle larger-scale problems.

C. Numerical Validation

In this subsection, we validate the proposed HyDWD algorithm via extensive simulations. The algorithms were tested using a hybrid D-Wave quantum annealer through the Leap quantum cloud service [36]. To manage the high cost of QPU usage and the time constraints of the developers, we performed multiple test cases that could be completed within 100 iterations. For the classical computing aspect, we utilized the LP solver on a classical CPU machine powered by an Intel(R) Core(TM) i9-10900K CPU operating at 3.70 GHz with 64GB of RAM.

1) *Simulation Setup*: To simulate the satellite-based quantum network, we consider a polar satellite constellation as outlined in [30], [34]. In this context, we have 10 rings of satellites in polar orbits, with each ring containing 10 satellites positioned at altitudes ranging from 2,000 km to 10,000 km. In a static scenario, only a subset of satellites is visible to a specific ground station within a fixed time frame, allowing for the potential application of other satellite constellations. Each satellite is equipped with quantum memory and a varying number of transmitters selected randomly from the ranges of [20, 100] and [6, 10], respectively.

Moreover, we designate 9 major cities as ground stations, including New York, London, Rio de Janeiro, Mumbai, Cape Town, Beijing, Sydney, Singapore, and Vancouver, with their locations determined by real GPS coordinates. With a total of 36 potential ground station pairs (GSPs), a subset of these points is randomly selected as traffic GSPs. For each GS, the quantum memory and the number of receivers are randomly chosen from the ranges of [10, 20] and [2, 6], respectively. We also assume that each GSP can connect to at most 1 satellite concurrently. In terms of the loss and noise parameters in the free-space optical transmission, we set them according to previous studies [30], [34].

2) *Compared Methods*: We compare the performance of our proposed HyDWD approach against various benchmarks:

- *QUBO* [36] solves the linearized problem (12) by transforming the problem into QUBO form.

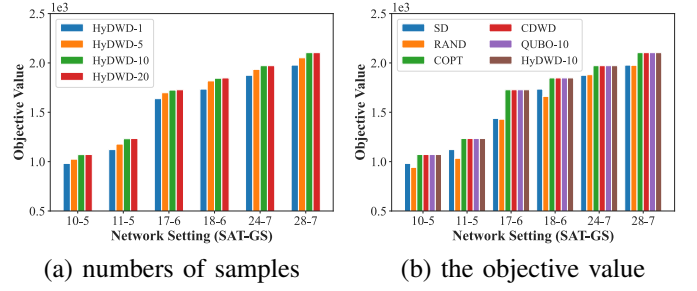


Fig. 3. (a) The objective value under for different numbers of QPU samples. (b) The objective value under different network settings.

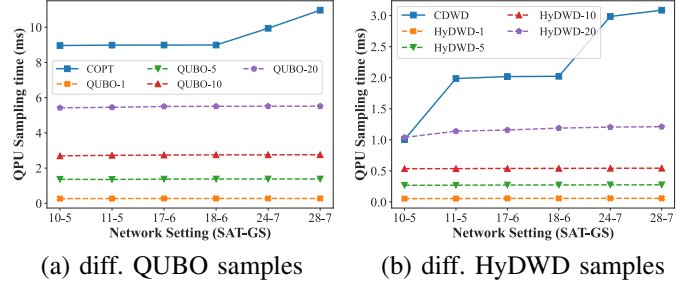


Fig. 4. (a) The QPU sampling time under various network settings for QUBO. (b) The QPU sampling time under various network settings for HyDWD.

- *SD* [36] (Steepest Descent) is a solver for binary quadratic models provided by D-Wave Systems and the best move is computed using a local minimization.
- *RAND* [36] is the uniform random sampling algorithm provided by the D-Wave Systems.
- *COPT* (Classical Optimizer) solves the linearized problem (12) by using a classical optimizer (e.g., Gurobi, Scipy) in a classical CPU computer.
- *CDWD* (Classical DW Decomposition) decouples the original problem by using DW decomposition and solves the master problem as well as all subproblems by leveraging a classical optimizer.

3) *Simulation Results*: We now report the results of the comparison between the proposed method and the benchmarks above regarding the objective value, solver accessing time, and qubit usage.

Performance in Objective Value. We begin by examining the impact of the number of samples utilized in the quantum annealing. We explore various HyDWD variants with different sample sizes (1, 5, 10, 20). As depicted in Fig. 3(a), an improvement in the objective value is observed with an increase in the number of samples. Notably, HyDWD-10 and HyDWD-20 yield identical results, followed by HyDWD-5 and HyDWD-1. Given the comparable performance of 10 and 20 samples, we opt to use 10 samples in subsequent simulations to reduce quantum annealing time.

Subsequently, we present a comprehensive analysis of the overall performance of all methods, as demonstrated in Fig. 3(b). Initially, we examine the objective value across various network settings denoted by the number of SATs and GSs. For example, 28-7 signifies a scenario with 28 SATs and

TABLE I
TOTAL QUBITS USAGE UNDER DIFFERENT NETWORK SETTINGS FOR DW SUBPROBLEMS.

Settings	Non-DW				DW							
	Quadratic		Linear		SUB X		SUB Y		q -th SUB X		q -th SUB Y	
	LQ	RQ	LQ	RQ	LQ	RQ	LQ	RQ	LQ	RQ	LQ	RQ
10-5	700	700	1000	1000	100	100	900	900	20	20	180	180
11-5	770	770	1100	1100	110	110	990	990	22	22	198	198
17-6	1750	1751	2394	2394	252	252	2142	2142	42	42	357	357
18-6	1855	1856	2538	2538	270	270	2268	2268	45	45	378	378
24-7	3374	3375	4680	4680	504	504	4176	4176	63	63	522	522
28-7	3962	3963	5464	5464	592	592	4872	4872	74	74	609	609
30-8	5628	-	7506	-	846	846	6660	-	94	94	740	740
34-8	-	-	-	-	-	-	-	-	106	106	839	839
38-8	-	-	-	-	-	-	-	-	118	118	938	938
40-9	-	-	-	-	-	-	-	-	144	144	1104	1104
44-9	-	-	-	-	-	-	-	-	158	158	1214	1214
48-9	-	-	-	-	-	-	-	-	172	172	1324	1324

LQ: Logic Qubit, RQ: Real Qubit, “-”: Not available.

7 GSs. Firstly, SD and RASD exhibit the poorest performance across all network settings compared to the other four methods, possibly due to the SD method becoming trapped in local optima. Secondly, COPT, CDWD, QUBO-10, and HyDWD-10 achieve identical results across all network settings, indicating the mathematical consistency of our proposed algorithm with the classical DW algorithm. Notably, our algorithm achieves these results in less time, as elaborated in the subsequent analysis.

Performance in Solver Accessing Time. We now investigate the performance of our proposed methods regarding the solver accessing time. The solver accessing time specifically denotes the actual time taken by the QPU solver and local solver, excluding other overheads such as variable setting time and parameter transmission time. Initially, we compare the QPU sampling time under various network settings for COPT and different numbers of QUBO sampling, as depicted in Fig. 4(a). We can find that the QPU sampling time of all QUBO variants remains stable while the computation time of COPT keeps steady before setting 18-6 and starts to increase linearly. In Fig. 4(b), we present the QPU sampling times of CDWD and all HyDWD variants. The result is similar to that in Fig. 4(a), but with reduced QPU sampling time consumption owing to the DW decomposition. In a word, all QUBO and HyDWD variants exhibit quicker convergence to similar optimization values compared to COPT and CDWD, underscoring the advantage of using quantum annealing.

Qubit Usage Comparison. We proceed to analyze the total qubit usage under different network settings for DW subproblems. Table I provides a detailed comparison of qubit usage, with each row corresponding to different settings such as 10-5, 11-5, 17-6, and so on. The table is structured into two main columns: Non-DW and DW. In the Non-DW category, the original problem remains intact, while in the DW category, the problem is decomposed using the DW technique. Additionally, the Non-DW section is further divided into Quadratic (original problem) and Linear (linearized problem), whereas the DW section compares qubit usage between centralized (SUB X

and SUB Y) and distributed (q -th SUB X and q -th SUB Y) manners. Furthermore, qubit usage is detailed within these categories as LQ (Logic Qubit for problem representation) and RQ (Real Qubit used in quantum annealer). We can see that the Non-DW and centralized DW categories can solve optimization problems up to a scale of 28-7, limited by current qubit capabilities. In contrast, the distributed DW category can tackle networks of scale 48-9 and even larger scenarios. This is made possible by leveraging distributed computing to split the problem into smaller subproblems, maximizing qubit utilization. These results demonstrate the benefits of integrating distributed quantum computing with the DW decomposition technique.

VI. CONCLUSION

In this study, we have introduced a novel hybrid quantum-classical method, HyDWD, employing Dantzig-Wolfe decomposition to tackle integer linear programming problems. Our approach strategically decomposes the ILP into a master problem and multiple smaller subproblems, leveraging classical computers and parallel quantum annealers to solve them respectively. By integrating parallel quantum computing techniques, our proposed framework exhibits the potential for significant enhancements in computational efficiency and solution quality. Through a case study focusing on the optimal entanglement distribution problem within satellite-assisted quantum networks, conducted via simulations on the D-Wave quantum annealing machine, our results confirm the efficiency and robustness of the proposed HQC framework, with a stable solver accessing time and consistent results compared to classical optimizers. These promising outcomes suggest that our novel HQC optimization approach holds substantial promise for a wide range of future applications.

REFERENCES

- [1] J. Preskill, “Quantum computing in the NISQ era and beyond,” *Quantum*, vol. 2, pp. 79, Aug. 2018.
- [2] M. Horowitz and E. Grumbling, “Quantum computing: progress and prospects,” Washington, DC: The National Academies Press. 2019. [Online]. Available: <https://doi.org/10.17226/25196>.

- [3] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, Dec. 1992.
- [4] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96, New York, NY, May 1996.
- [5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.
- [6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019.
- [7] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, "Quantum computational advantage using photons," *Science*, vol. 370, no. 6523, pp. 1460–1463, Dec. 2020.
- [8] S. Niu and A. Todri-Sanial, "Effects of dynamical decoupling and pulse-level optimizations on IBM quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 3, no. 3102510, pp. 1–10, Aug. 2022.
- [9] Ö. Salehi, A. Glos, and J. A. Miszczak, "Unconstrained binary models of the travelling salesman problem variants for quantum optimization," *Quantum Information Processing*, vol. 21, no. 2, pp. 67, Jan. 2022.
- [10] D. An and L. Lin, "Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm," *ACM Transactions on Quantum Computing*, vol. 3, no. 2, pp. 1–28, Jun. 2022.
- [11] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," *Journal of the Physical Society of Japan*, vol. 90, no. 3, pp. 032001, Feb. 2021.
- [12] Z. Zhao, L. Fan, and Z. Han, "Hybrid quantum benders' decomposition for mixed-integer linear programming," in *IEEE Wireless Communications and Networking Conference*, Austin, TX, Apr. 2022.
- [13] Z. Zhao, L. Fan, Y. Guo, Y. Wang, Z. Han, and L. Hanzo, "QAOA-assisted benders' decomposition for mixed-integer linear programming," in *Proceedings of 2024 IEEE International Conference on Communications (ICC)*, Denver, CO, June 2024.
- [14] A. Ajagekar, T. Humble, and F. You, "Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems," *Computers & Chemical Engineering*, vol. 132, no. 106630, pp. 1–19, Jan. 2020.
- [15] A. Ajagekar, K. Al Hamoud, and F. You, "Hybrid classical-quantum optimization techniques for solving mixed-integer programming problems in production scheduling," *IEEE Transactions on Quantum Engineering*, vol. 3, no. 3102216, pp. 1–16, Jun. 2022.
- [16] T. Tran, M. Do, E. Rieffel, J. Frank, Z. Wang, B. O'Gorman, D. Venturelli, and J. Beck, "A hybrid quantum-classical approach to solving scheduling problems," in *Proceedings of the International Symposium on Combinatorial Search*, New York, NY, Jul. 2016.
- [17] N. G. Paterakis, "Hybrid quantum-classical multi-cut benders approach with a power system application," *Computers & Chemical Engineering*, vol. 172, no. 108161, pp. 1–17, Apr. 2023.
- [18] L. Fan and Z. Han, "Hybrid quantum-classical computing for future network optimization," *IEEE Network*, vol. 36, no. 5, pp. 72–76, Nov. 2022.
- [19] 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*, vol. 15, no. 2, pp. 847–858, Apr. 2024.
- [20] 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 IEEE International Conference on Computer Communications and Networks*, Honolulu, HI, Jul. 2023.
- [21] X. Wei, L. Fan, Y. Guo, Y. Gong, Z. Han, and Y. Wang, "Hybrid quantum-classical Benders' decomposition for federated learning scheduling in distributed networks," *IEEE Transactions on Network Science and Engineering (TNSE)*, to appear.
- [22] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Physical Review A*, vol. 97, no. 2, pp. 022304, Feb. 2018.
- [23] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to optimize variational quantum circuits to solve combinatorial problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2367–2375, Jun. 2020.
- [24] O. Lockwood and M. Si, "Reinforcement learning with quantum variational circuit," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, pp. 245–251, Oct. 2020.
- [25] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, Feb. 1960.
- [26] L. Wolsey, *Integer Programming*. Wiley, 2020. [Online]. Available: <https://books.google.com/books?id=knH8DwAAQBAJ>
- [27] M. Caleffi, M. Amoretti, D. Ferrari, D. Cuomo, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: a survey," *arXiv preprint arXiv:2212.10609*, Dec. 2022.
- [28] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, "Towards a distributed quantum computing ecosystem," *IET Quantum Communication*, vol. 1, no. 1, pp. 3–8, Jul. 2020.
- [29] F. Glover, G. Kochenberger, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using QUBO models," *Annals of Operations Research*, vol. 17, no. 4, pp. 335–371, Apr. 2022.
- [30] S. Khatri, A. J. Brady, R. A. Desporte, M. P. Bart, and J. P. Dowling, "Spooky action at a global distance: analysis of space-based entanglement distribution for the quantum Internet," *npj Quantum Information*, vol. 7, no. 4, pp. 1–15, Jan. 2021.
- [31] T. Gonzalez-Raya, S. Pirandola, and M. Sanz, "Satellite-based entanglement distribution and quantum teleportation with continuous variables," *arXiv preprint arXiv:2303.17224*, Mar. 2023.
- [32] M. Ghaderibaneh, H. Gupta, C. Ramakrishnan, and E. Luo, "Pre-distribution of entanglements in quantum networks," in *2022 IEEE International Conference on Quantum Computing and Engineering*, Broomfield, CO, Sep. 2022.
- [33] S. Pouryousef, N. K. Panigrahy, and D. Towsley, "A quantum overlay network for efficient entanglement distribution," in *IEEE Conference on Computer Communications*, New York, May 2023.
- [34] N. K. Panigrahy, P. Dhara, D. Towsley, S. Guha, and L. Tassiulas, "Optimal entanglement distribution using satellite based quantum networks," in *IEEE Conference on Computer Communications Workshops*, Virtual, May 2022.
- [35] X. Wei, J. Liu, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Optimal entanglement distribution problem in satellite-based quantum networks," *IEEE Network Magazine*, to appear.
- [36] D-Wave Systems Inc., "Ocean software documentation: dwave-samplers, steepestdescentsolver, randomnessampler," Oct. 2023. [Online]. Available: https://docs.ocean.dwavesys.com/en/stable/docs_samplers/reference.html