

# Hybrid Quantum Benders' Decomposition For Mixed-integer Linear Programming

Zhongqi Zhao<sup>1,2</sup>, Lei Fan<sup>2</sup>, and Zhu Han<sup>1</sup>

*Dept. of Electrical and Computer Engineering<sup>1</sup> and Dept. of Engineering Technology<sup>2</sup>  
University of Houston, TX, USA*

**Abstract**—The Benders' decomposition algorithm is a technique in mathematical programming for complex mixed-integer linear programming (MILP) problems with a particular block structure. The strategy of Benders' decomposition can be described as a strategy of divide and conquer. The Benders' decomposition algorithm has been employed in a variety of applications such as communication, networking, and machine learning. However, the master problem in Benders' decomposition is still NP-hard, which motivates us to employ quantum computing. In the paper, we propose a hybrid quantum-classical Benders' decomposition algorithm. We transfer the Benders' decomposition's master problem into the quadratic unconstrained binary optimization (QUBO) model and solve it by the state-of-the-art quantum annealer. Then, we analyze the computational results and discuss the feasibility of the proposed algorithm. Due to our reformulation in the master problem in Benders' decomposition, our hybrid algorithm, which takes advantage of both classical and quantum computers, can guarantee the solution quality for solving MILP problems.

**Index Terms**—Benders' Decomposition, Mixed-integer Linear Programming, Hybrid Quantum-Classical Computing, Quantum Computing, Communication, Networking

## I. INTRODUCTION

In recent years, mixed-integer linear programming (MILP) is widely employed in many fields includes but is not limited to planning and scheduling [1], transportation and telecommunications [2], energy and resource management [3] [4], network design [5], and vehicle routing [6]. However, because MILP is an NP-hard problem [7] [8], it is desirable to have powerful solutions to solve large-scale MILP problems.

Benders' decomposition algorithm [9] provides an efficient way to solve the MILP problem. In recent years, the Benders' decomposition method has become a popular algorithm. It exploits the structure of the MILP problem and successfully reduces the computation workload. Benders' decomposition divides a MILP problem into a master problem and a subproblem. The subproblem is a linear programming model that has strong duality. During the iterative solving process, the optimality cuts and feasibility cuts obtained from the solution of the subproblem will be added to the master problem. The iterative process will stop when the gap between the upper bound provided by the master problem and the lower bound provided by the subproblem is sufficiently small. However, the master problem in Benders' decomposition is still NP-hard, which motivates us to employ quantum computing.

Due to the characteristics of parallel computing, the quantum computer can simultaneously calculate the final result with

every possible model input. Therefore, quantum computers are often considered to have surpassed the computational speed of classical computers. The quantum advantage has been proved by quantum algorithms, including Deutsch-Jozsa Algorithm [10], Grover's Algorithm [11], Shor's Algorithm [12]. In recent years, leading quantum computer companies such as D-Wave, IBM, Google, IonQ, Honeywell have made significant progress in hardware design. Normally, their quantum computers are in one of the two groups (i.e., analog quantum model, universal quantum gate model). D-Wave currently provides the quantum computer with the largest number of qubits on the market. By deploying the Ising model, the D-Wave's quantum annealer computer can solve the problem formulated by the quadratic unconstrained binary optimization (QUBO) model. It depicts the energy state with coupling qubits interaction and externally applied fields [13].

The power of quantum computers and challenges in MILP problems inspire us to design a hybrid quantum-classical Benders' decomposition algorithm by jointly using quantum computing and classical computing techniques. However, there are several obstacles when we try to use Benders' decomposition to combine quantum and classical computing. The first difficulty is how to convert the problem into an integer linear programming (ILP) problem recognized by the quantum computer. The Second difficulty is how to convert the NP-hard ILP problem into a QUBO model as an input to a quantum computer. In addition, the third difficulty is how to reformat the output of quantum computers from the binary solution to the decimal numeral system.

To overcome the above challenges, this paper reformulates the master problem as an ILP model in Benders' decomposition under the MILP problem. We transfer the master problem, which is an ILP model, into a QUBO form. Although the master problem in the Benders' decomposition is still NP-hard, we overcome this issue by proposing a hybrid Benders' decomposition algorithm with the D-Wave hybrid quantum computer for the MILP problem. Finally, we study a simple case to evaluate the performance of the D-Wave hybrid solver in solving the MILP problem. The contributions of this paper are summarized as follows.

- We propose a hybrid quantum-classical Benders' decomposition algorithm to find the solution for the MILP problem. Our hybrid quantum-classical Benders' decomposition algorithm converges and returns the correct final result as the classical algorithm does.

- We propose an ILP model for the master problem in Benders' decomposition for the MILP problem. We reformulate the ILP model as a QUBO model recognized by the quantum annealing machine.
- We employ the quantum computer provided by D-Wave to solve the MILP problem by their Leap™ quantum cloud service. Our experiments show the possibility of using quantum computing to solve the MILP.

The rest of this paper is organized as follows. Section II introduces the basics about the MILP problem and the Benders' decomposition. Section III illustrates our hybrid quantum-classical Benders' Decomposition algorithm. Section IV validates our algorithm by showing the corresponding simulation. Finally, Section V concludes the whole paper.

## II. MILP AND BENDERS' DECOMPOSITION BASICS

### A. Mixed-integer Linear Programming

MILP has been widely adopted optimization problems that include but are not limited to communication and networks. In the field of communication, for instance, it has been employed to the resource allocation in wavelength division multiple access (WDMA) [14], multi-access edge computing (MEC) [15], offloading for fog computing [16], etc. Furthermore, it plays an essential role in the field of network research, such as minimizing the network delay [17], scheduling virtual network re-configurations [18], and improving the supply chain network [19]. Besides civilian applications, it is no doubt that MILP has been adopted for military purposes such as assisting military pilot training [20]. These demonstrate that MILP is a powerful tool for either classical problems or future popular topics such as communication networks and resource allocation. Consider a MILP model as follows:

$$\begin{aligned} \max_{x,y} \quad & c^\top x + h^\top y \\ \text{s.t.} \quad & Ax + Gy \leq b, \\ & x \in X, x \in \{0,1\}^n, y \in \mathbb{R}_+^p. \end{aligned} \quad (1)$$

Here  $x$  is a vector with binary variables and  $A$  is its corresponding coefficient matrix in constraints.  $y$  is a vector with non-negative continuous variables and  $G$  is its corresponding coefficient matrix in constraints.  $c^\top$  is the coefficient vector of variable  $x$  in the objective function and  $h^\top$  is the coefficient vector of variable  $y$  in the objective function.

### B. Benders' Decomposition for MILP

We first provide a brief introduction of the Benders' Decomposition. In the MILP problem (1), for each possible choice of  $\bar{x} \in X$ , we find the best choice for  $y$  by solving a linear program. So we regard  $y$  as a function of  $x$ . Then we replace the contribution of  $y$  to the objective with a scalar variable representing the value of the best choice for a given  $\bar{x}$ . We start with a crude approximation to the contribution of  $y$  and then generate a sequence of dual solutions to tighten up the approximation. Hence, the original MILP problem in (1) can be written equivalently to a master problem as follows.

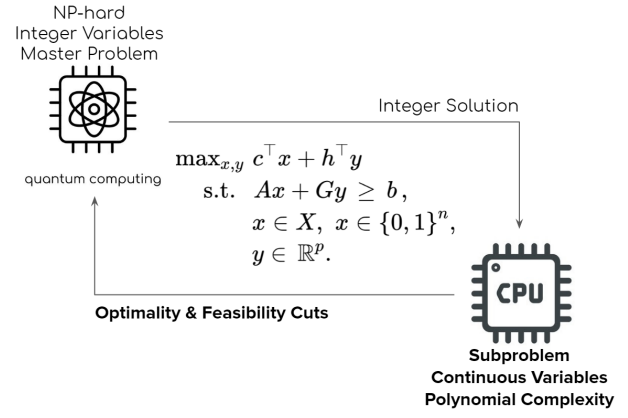


Fig. 1. The structure of hybrid Benders' decomposition algorithm for MILP using both quantum and classical computing

$$\begin{aligned} \text{Master Problem: } \max_{x,t} \quad & c^\top x + t \\ \text{s.t.} \quad & (b - Ax)^\top u^k \geq t \quad \text{for } k \in \hat{K}, \\ & (b - Ax)^\top r^j \geq 0 \quad \text{for } j \in \hat{J}, \\ & x \in X, x \in \{0,1\}^n, t \in \mathbb{R}. \end{aligned} \quad (2)$$

We denote  $K$  and  $J$  as the extreme points  $u^k$  and extreme rays  $r^k$  of the dual polyhedron  $Q = \{u \in \mathbb{R}_+^m \mid G^\top u \geq h\}$  generated by the linear programming's duality, which is called the subproblem. The subproblem is as follows.

$$\begin{aligned} \text{Subproblem: } z_{LP}(x) = \min_u \quad & (b - Ax)^\top u \\ \text{s.t.} \quad & G^\top u \geq h, \\ & u \in \mathbb{R}_+^m. \end{aligned} \quad (3)$$

In the subproblem, if the inner product between  $(b - Ax)$  and any dual ray  $r^{j'}$  is negative then  $z_{LP}(x) = -\infty$ . Equivalently, in this situation, the dual problem of Problem (3) is infeasible. Then,  $x$  does not allow a feasible solution to the original mixed-integer problem (1). Thus, we have a new feasibility cut, i.e.,

$$(b - Ax)^\top r^{j'} \geq 0, \quad j' \in J. \quad (4)$$

If  $x$  satisfies (4), then we yield an extreme point  $u^{k'}$  and the value of  $z_{LP}(x)$  is given by

$$z_{LP}(x) = \min_{k' \in K} (b - Ax)^\top u^{k'}. \quad (5)$$

Thus, problem (1) is written equivalently as problem (2) by denoting  $z_{LP}(x)$  as a continuous real number variable  $t$ . Equation (5) posts a new optimality cut in master problem. In addition to the extreme points and extreme rays, we use  $\hat{K}$  and  $\hat{J}$  to denote the current known extreme points and extreme rays of  $Q$ , respectively, where  $\hat{K} \subseteq K$  and  $\hat{J} \subseteq J$ .

## III. HYBRID QUANTUM-CLASSICAL BENDERS' DECOMPOSITION

Quantum computer is a powerful tool to solve NP-hard integer problems. Accordingly, we introduce the quantum

TABLE I  
TABLE OF COMMON CONSTRAINT-PENALTY PAIRS

Constraint	Equivalent Penalty
$x_1 + x_2 = 1$	$P(x_1 + x_2 - 1)^2$
$x_1 + x_2 \geq 1$	$P(1 - x_1 - x_2 + x_1 x_2)^2$
$x_1 + x_2 \leq 1$	$P(x_1 x_2)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1 x_2 + x_1 x_3 + x_2 x_3)$

computer to solve the master problem in (2) since it has a special formulation that can convert to an ILP problem. The subproblem is a linear programming model which can be solved well on the classical computer. As Fig. 1 shows, we design a hybrid algorithm in which the master problem is solved by the quantum computing techniques and the subproblem is solved by the classical computing techniques. The two different computers are communicating with each other. The subproblem returns new optimality and feasibility cuts to tighten the bounds of the master problem. The integer solution returned from the master problem gives the subproblem a direction to find new optimality and feasibility cuts.

#### A. Quantum Formulation

Quantum annealers are able to solve the optimization problem in a QUBO formulation. To leverage state-of-art quantum annealers provided by D-Wave, the ILP problem has to be converted to the corresponding QUBO formulation. The definition of QUBO are (6) and (7). Let  $f: \{0, 1\}^n \rightarrow \mathbb{R}$  be a quadratic polynomial over binary variables  $x$  of length  $n$  and  $q_{ij}$  is corresponding cost coefficient of  $x_i x_j$ . We get equation (6)

$$f_Q(\mathbf{x}) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} q_{ij} x_i x_j = \mathbf{x}^T \mathbf{Q} \mathbf{x}. \quad (6)$$

The QUBO problem consists of finding a binary vector  $\mathbf{x}^*$  that is minimal with respect to  $f$  among all other binary vectors, namely,

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x}). \quad (7)$$

Here  $\mathbf{x}$  is a vector of binary variables where the length of  $n$ , and  $\mathbf{Q}$  is either an upper-diagonal matrix or a symmetric matrix. Since (6) is an unconstrained optimization model, we need to reformulate our constrained ILP as unconstrained QUBO by using penalties. Next, we get the optimal solution by finding the best penalty coefficients of the constraints. The principles of transforming classical constraints to their equivalent penalties are displayed in the TABLE I. Here  $x_1$ ,  $x_2$  and  $x_3$  are binary variables.  $s_i$  is a binary slack variable.  $a_l$  is the coefficient for the corresponding slack variable.  $b$  is a constant.  $P$  is a user-defined penalty coefficient.

#### B. Variable Representation

Now consider problem (2), the initial binary decision variables make up the vector  $\mathbf{x} \in X$  with length of  $n$ . In order to reformulate the master problem into the QUBO formulation, we need to represent the continuous variable  $t$  using binary bits. We use a binary vector  $\mathbf{w}$  with length of  $M$  bits to replace continuous variable  $t$  and denote it as a new discrete number  $\bar{t} \in \mathbb{Q}$ . In general,  $\bar{t}$  requires the binary numeral system assigning  $M$  bits to replace continuous variable  $t$ . Then we can recover the  $\bar{t}$  by

$$\begin{aligned} \bar{t} &= \sum_{i=-\underline{m}}^{\bar{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\bar{m}_-} 2^j w_{j+(1+\underline{m}+\bar{m}_+)} \\ &= \bar{t}(\mathbf{w}). \end{aligned} \quad (8)$$

In (8),  $\bar{m}_+ + 1$  is the number of bits that assigned to represent positive integer part,  $\underline{m}$  is the number of bits assigned to represent the positive decimal part, and  $\bar{m}_- + 1$  is the number of bits that represent the negative integer part.

#### C. QUBO Setup

Let's reconsider problem (2) and replace  $\bar{t}$  in (8). The new problem is only depend on binary decision variables  $\mathbf{x}$  and  $\mathbf{w}$ . Then, the new master problem is

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{w}} \quad & c^T \mathbf{x} + \sum_{i=-\underline{m}}^{\bar{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\bar{m}_-} 2^j w_{j+(1+\underline{m}+\bar{m}_+)} \\ \text{s.t.} \quad & (b - A\mathbf{x})^T \mathbf{u}^k \geq \bar{t}(\mathbf{w}), \quad \text{for } k \in \hat{K}, \\ & (b - A\mathbf{x})^T \mathbf{r}^j \geq 0, \quad \text{for } j \in \hat{J}, \\ & \mathbf{x} \in X, \quad \mathbf{x} \in \{0, 1\}^n, \\ & \mathbf{w} \in W, \quad \mathbf{w} \in \{0, 1\}^M. \end{aligned} \quad (9)$$

Then we create a new binary decision variable collection  $\mathbf{x}' = \{\mathbf{w}, \mathbf{x}\}$  to set up our QUBO matrix, where the length of the binary vector  $\mathbf{x}'$  is  $n + M$ . According to the rule of setting up QUBO, we will transfer our objective and constraints one by one to be a symmetric matrix.

1) *Objective Function:* Because the quantum computer only accepts a quadratic polynomial over binary variables. Following the principle of the QUBO formulation, we convert the objective function to be a QUBO matrix as follows.

$$\begin{aligned} & c^T \mathbf{x} + \sum_{i=-\underline{m}}^{\bar{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\bar{m}_-} 2^j w_{j+(1+\underline{m}+\bar{m}_+)} \\ \Rightarrow \mathbf{Q}_{obj} &= \mathbf{x}'^T \text{diag}(c) \mathbf{x}' + \sum_{i=-\underline{m}}^{\bar{m}_+} w_{i+\underline{m}} 2^i w_{i+\underline{m}} \\ & - \sum_{j=0}^{\bar{m}_-} w_{j+(1+\underline{m}+\bar{m}_+)} 2^j w_{j+(1+\underline{m}+\bar{m}_+)}. \end{aligned} \quad (10)$$

2) *Optimality Cuts*: Similarly, following the principle of constraint-penalty pairs in TABLE I, we not only introduce the penalty but also convert the optimality cuts constraint to be a QUBO matrix as follow.

$$\bar{t}(\mathbf{w}) + (u^k)^\top \mathbf{A}\mathbf{x} \leq b^\top u^k, \text{ for } k \in \hat{K}.$$

$$\Rightarrow P_k \left( \bar{t}(\mathbf{w}) + (u^k)^\top \mathbf{A}\mathbf{x} + \sum_{l=0}^{\bar{l}^K} 2^l s_{kl}^K - b^\top u^k \right)^2,$$

$$\text{where } \bar{l}^K = \left\lceil \log_2 \left( b^\top u^k - \min_{\mathbf{w}, \mathbf{x}} \left( \bar{t}(\mathbf{w}) + (u^k)^\top \mathbf{A}\mathbf{x} \right) \right) \right\rceil.$$

3) *Feasibility Cuts*: Similarly, following the principle of constraint-penalty pairs in TABLE I, we convert the feasibility cuts constraint to be a QUBO matrix with penalties as follows:

$$(r^j)^\top \mathbf{A}\mathbf{x} \leq b^\top r^j, \text{ for } j \in \hat{J}.$$

$$\Rightarrow P_j \left( (r^j)^\top \mathbf{A}\mathbf{x} + \sum_{l=0}^{\bar{l}^J} 2^l s_{kl}^J - b^\top r^j \right)^2,$$

$$\text{where } \bar{l}^J = \left\lceil \log_2 \left( b^\top r^j - \min_{\mathbf{x}} \left( (r^j)^\top \mathbf{A}\mathbf{x} \right) \right) \right\rceil.$$

#### D. Proposed Algorithm

We deploy the D-Wave solver in our proposed heuristic algorithm to solve the model. In addition, we need to carefully tune the penalties for a decent QUBO model. An extremely large penalty may cause quantum annealer malfunctioning since it will explode the coefficients. Similarly, a soft penalty may make quantum annealer ignore the corresponding constraints. As long as the penalty is well-tuned, the quantum solver will give the right answer with a relatively high probability. Therefore, we summarize our proposed hybrid quantum-classical Benders' decomposition algorithm as **Algorithm 1**.

#### IV. NUMERICAL VALIDATION

We validate the proposed algorithm by running on a hybrid D-Wave quantum processing unit (QPU). Because the input of interest to practice is too large to fit onto current-model QPUs and be solved directly by quantum annealing. We choose to use the hybrid model instead of the pure QPU. The reason for using the hybrid model is that the hybrid solver overcomes a lot of input size barriers and allows the QPU to accept a large input. More details can be found in [21]. We accessed the D-Wave system by Leap<sup>TM</sup> quantum cloud service.

##### A. Example Setup

In our simulation, we consider a simple problem to test our proposed quantum algorithm, where  $\mathbf{x} \in X$ ,  $\mathbf{x} \in \{0, 1\}^2$ ,  $\mathbf{y} \in Y$ ,  $\mathbf{y} \geq 0$

#### Algorithm 1 Hybrid Quantum-Classical Benders' Decomposition Algorithm

---

**Require:** Initial sets  $\hat{K}$  of extreme points and  $\hat{J}$  of extreme rays of  $Q$

$\bar{t} \leftarrow +\infty$

$\underline{t} \leftarrow -\infty$

**while**  $|\bar{t} - \underline{t}| \geq \epsilon$  **do**

$\mathbf{P} \leftarrow$  Appropriate penalties numbers or arrays

$\mathbf{Q} \leftarrow$  Reformulate both objective and constraints in (2) and construct the QUBO formulation by using corresponding rules

$\mathbf{x}' \leftarrow$  Solve problem (6) by quantum computer.

$\bar{t} \leftarrow$  Extract  $\mathbf{w}$  and replace the  $\bar{t}$  with  $\bar{t}(\mathbf{w})$  (8)

$z_{LP}(\mathbf{x}) \leftarrow$  Solve the problem (3)

$\underline{t} \leftarrow z_{LP}(\mathbf{x})$

**if**  $z_{LP}(\mathbf{x}) = -\infty$  **then**

        An extreme ray  $j$  of  $Q$  has been found.

$\hat{J} = \hat{J} \cup \{j\}$

**else if**  $z_{LP}(\mathbf{x}) < \bar{t}$  **and**  $\bar{t} \neq +\infty$  **then**

        An extreme point  $k$  of  $Q$  has been found.

$\hat{K} = \hat{K} \cup \{k\}$

**end if**

**end while**

**return**  $\bar{t}, \mathbf{x}$

---

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

$$h^\top = [8 \quad 9 \quad 5 \quad 6], \quad c^\top = [-15 \quad -10].$$

##### B. Result

Fig. 2 shows how our proposed hybrid quantum-classical Benders' decomposition algorithm adds optimality and feasibility cuts to get the solution and corresponding  $\bar{t}$ . In the example, the algorithm takes 4 rounds to let  $|\bar{t} - \underline{t}|$  converge. Four cuts are added iteratively in the space and tighten the feasible region. The algorithm can find the optimal solution in each round. Therefore, Fig. 2 demonstrates that our algorithm is reliable and efficient. In Fig. 3, the hidden dashed line denotes that the lower bound in the corresponding round is negative infinity. As we can see in the graph, the upper bound and lower bound converge. Our algorithm only takes two rounds to find the non-negative infinity lower bound. This result proves that our proposed algorithm is mathematically consistent with the classical Benders' decomposition algorithm. In other words, as long as the classical Benders' decomposition algorithm can solve it, our algorithm can at least achieve the same.

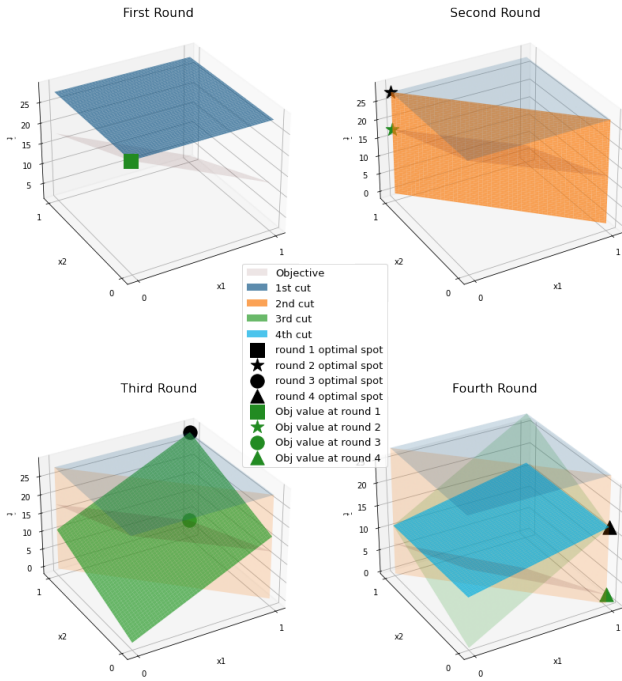


Fig. 2. Benders' Decomposition Process

## V. CONCLUSION

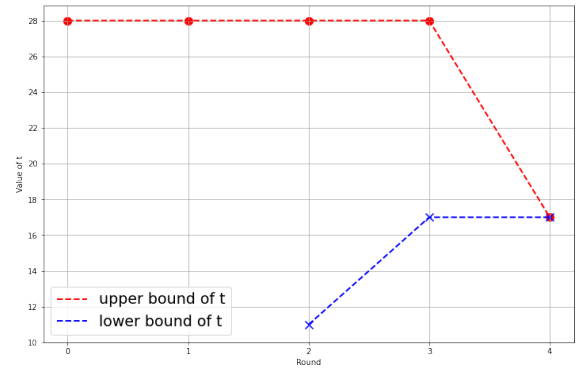
In this paper, we not only reformulate the master problem of Benders' decomposition for the MILP problem as an ILP model but also successfully convert it to a QUBO model. Our hybrid quantum-classical Benders' decomposition algorithm converges and returns the correct final result as the classical algorithm does. In addition to that, our algorithm also guarantees the solution quality for solving the MILP problem. In our simulation, we solved an NP-hard MILP problem by using the hybrid quantum computer provided by D-Wave. Therefore, we can conclude that the quantum computer can potentially replace the classical computer in solving the NP-hard master problem of Benders' decomposition algorithm in MILP problems.

## ACKNOWLEDGEMENT

This work is partially supported by US NSF CNS-2107216, CNS-2128368, EPCN-2045978.

## REFERENCES

- [1] S. P. Canto, "Application of Benders' decomposition to power plant preventive maintenance scheduling," *European Journal of Operational Research*, vol. 184, no. 2, pp. 759–777, Jan. 2008.
- [2] A. M. Costa, "A survey on benders decomposition applied to fixed-charge network design problems," *Computers & Operations Research*, vol. 32, no. 6, pp. 1429–1450, Jun. 2005.
- [3] X. Cai, D. C. McKinney, L. S. Lasdon, and D. W. Watkins, "Solving large nonconvex water resources management models using generalized benders decomposition," *Operations Research*, vol. 49, no. 2, pp. 235–245, Apr. 2001.
- [4] J. L. Zhang and K. Ponnambalam, "Hydro energy management optimization in a deregulated electricity market," *Optimization and Engineering*, vol. 7, no. 1, pp. 47–61, Mar. 2006.
- [5] B. Fortz and M. Poss, "An improved Benders decomposition applied to a multi-layer network design problem," *Operations Research Letters*, vol. 37, no. 5, pp. 359–364, Sep. 2009.
- [6] A. I. Corr  a, A. Langevin, and L.-M. Rousseau, "Scheduling and routing of automated guided vehicles: A hybrid approach," *Computers & Operations Research*, vol. 34, no. 6, pp. 1688–1707, Jun. 2007.
- [7] A. Bulut and T. K. Ralphs, "On the complexity of inverse mixed integer linear optimization," *arXiv:2104.09002 [cs, math]*, Apr. 2021, arXiv: 2104.09002. [Online]. Available: <http://arxiv.org/abs/2104.09002>
- [8] C. H. Papadimitriou and M. Yannakakis, "The complexity of facets (and some facets of complexity)," in *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, ser. STOC '82. New York, NY: ACM, May 1982, pp. 255–260.
- [9] A. M. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, Oct. 1972.
- [10] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, Dec. 1992.
- [11] 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: ACM, Jul. 1996, p. 212–219.
- [12] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.
- [13] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift f  r Physik*, vol. 31, no. 1, pp. 253–258, Feb. 1925.
- [14] A. S. Elgamal, O. Z. Alsulami, A. A. Qidan, T. E. H. El-Gorashi, and J. M. H. Elmoghani, "Q-learning algorithm for resource allocation in WDMA-based optical wireless communication networks," *arXiv:2105.10729 [cs, eess]*, May 2021, arXiv: 2105.10729. [Online]. Available: <http://arxiv.org/abs/2105.10729>
- [15] Y. Yu, X. Bu, K. Yang, H. Yang, X. Gao, and Z. Han, "UAV-Aided Low Latency Multi-Access Edge Computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4955–4967, May 2021.
- [16] Z. Wu, B. Li, Z. Fei, Z. Zheng, B. Li, and Z. Han, "Energy-Efficient Robust Computation Offloading for Fog-IoT Systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4417–4425, Apr. 2020.
- [17] W. Xuan, Z. Zhao, L. Fan, and Z. Han, "Minimizing delay in network function visualization with quantum computing," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. Los Alamitos, CA: IEEE Computer Society, oct 2021, pp. 108–116.
- [18] X. Pan, S. Zhao, H. Yang, S. Tang, and Z. Zhu, "Scheduling Virtual Network Reconfigurations in Parallel in Hybrid Optical/Electrical Datacenter Networks," *Journal of Lightwave Technology*, vol. 39, no. 17, pp. 5371–5382, Sep. 2021.
- [19] P. Santander, F. A. Cruz Sanchez, H. Boudaoud, and M. Camargo, "Closed loop supply chain network for local and distributed plastic recycling for 3D printing: a MILP-based optimization approach," *Resources, Conservation and Recycling*, vol. 154, p. 104531, Mar. 2020.
- [20] V. Mak-Hau, B. Hill, D. Kirszenblat, B. Moran, V. Nguyen, and A. Novak, "A simultaneous sequencing and allocation problem for military pilot training: Integer programming approaches," *Computers & Industrial Engineering*, vol. 154, p. 107161, Apr. 2021.
- [21] "D-Wave Hybrid Solver Service: An Overview." [Online]. Available: <https://www.dwavesys.com/resources/white-paper/d-wave-hybrid-solver-service-an-overview/>

Fig. 3. Convergence of  $\bar{t}$  and  $\tilde{t}$