Quantum Computing for Cable-Routing Problem in Solar Power Plants

Zhongqi Zhao[1,2], Lei Fan[2], Honghao Zheng[3], and Zhu Han[1]
Dept. of Electrical and Computer Engineering[1] and Dept. of Engineering Technology[2]
University of Houston, TX, USA
Grid Strategy & Analytic[3]
ComEd, Chicago, IL, USA

2023 North American Power Symposium (NAPS) | 979-8-3503-1509-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/NAPS58826.2023.10318615

***Abstract*—The solar power plant is a large-scale photovoltaic (PV) system aimed to generate solar power for the electricity grid. It includes PV arrays (PVAs), cables, and other electrical accessories. Moreover, the solar power plant builder must consider various parameters and design regulations. The cable routing problem (CRP) is critical in large-scale solar power plant design. The objective of our CRP is to minimize the installation cost of the cable by determining the partition of the photovoltaic array and the cable routing. In this study, we use the quantum computer to solve the CRP, an NP-hard integer linear programming (ILP) problem, and show its advantages over classic computers. We transfer the ILP CRP into the quadratic unconstrained binary optimization (QUBO) model and solve it by the advanced quantum annealer. Finally, we analyze the computational results and discuss the advantages of our approach to solving the CRP.**

***Index Terms*—Cable Routing, Integer Linear Programming, Optimization, Quadratic Unconstrained Binary Optimization, Quantum Computing, Solar Power Plant**

I. Introduction

The massive consumption of fossil fuels such as oil, natural gas, and coal has caused various environmental problems, such as the rise of greenhouse gas (GHG) emissions. As a promising source of clean energy to decline the GHG, the adoption of photovoltaic (PV) power worldwide has been rocketing, increasing for more than two decades from a global capacity of 40 gigawatts (GW) in 2010 to at least 760.4 GW in 2020, which takes up about 3.7% of global electricity demand [1]. Among all countries, the US market increases to 19.2 GW. It is a new record with utility-scale installations accounting for about 73% of the new additions [2]. By the end of 2050, solar power is projected to become the world's second-largest source of electricity, exceeding 8,000 GW in cumulative installed capacity [3].

In order to advance the growth of renewable energy, it is crucial to minimize investment expenses, particularly construction costs. An effective blueprint layout can potentially lower the budget by 5%–10% [4]. For example, the cable-routing problem (CRP) as an essential component in the blueprint layout can minimize the building cost in the initial construction period. Some researchers have investigated CRP applications in wind farms and solar plants. [5] proposed a basic mixed integer linear programming (MILP) model for a real-world offshore wind park CRP and compared solution quality versus computing time with their heuristic algorithms. [6] proposed a genetic algorithm to optimize CRP that could beat the Dijkstra method. [4] proposed a general integer linear programming (ILP) model for solar plants. They proved the efficiency of their proposed Branch-and-Price-and-Cut algorithm by analyzing the computing time and solution quality. Previous research papers have proven that CRP is a time-consuming task when the problem size becomes larger, and finding a better way to reduce the computing time is worthwhile.

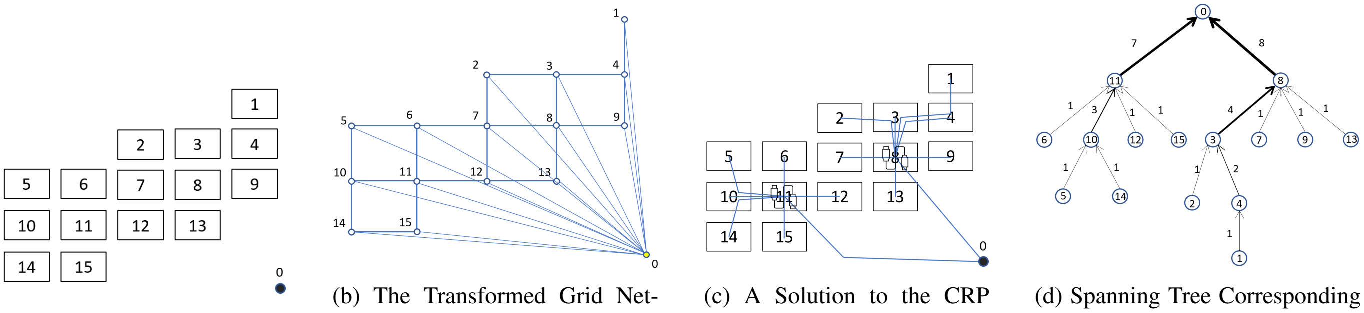
In this paper, we adopt [4]'s concept and focus on a solar plant construction project of a constrained CRP in which the PV arrays (PVAs) location is given. We address the CRP and get a feasible solution for a given zone by routing the cables to connect the necessary components. Then, the CRP aims to minimize the total cable cost while satisfying several constraints regularized by the safety requirements, the physical characteristics of components, and other special needs. There are two basic constraints required as follows:

1) Capacity Rule: The number of PVAs allocated to each combiner box is related to the capacity of the combiner box. We usually connect as many PVAs as possible to each combiner box to save costs.
2) Connectivity Rule: Each PVA must be connected by a cable to its corresponding combiner box for safety considerations. Cable intersection is forbidden in the PV system [7]. As a result, the distance between each PVA and the corresponding combiner box is defined by the Manhattan distance.

Thanks to parallel computing capability, quantum computers are reckoned to have better computational time than classical computers. Quantum Monte Carlo simulation is a well-known example [8].

With the technique of the Ising model, the quantum annealer can solve the quadratic unconstrained binary optimization (QUBO) problem. Researchers in [9] have shown that quantum computing is able to solve ILP optimization problems in Benders' decomposition. [10] discusses the challenges and opportunities in the fields of energy systems, and [11] points out the quantum computing speed-up potential in renewable energy topic.

Since the CRP belongs to the NP-hard ILP problem under



(a) A CRP Instance

(b) The Transformed Grid Network

(c) A Solution to the CRP Instance

(d) Spanning Tree Corresponding to the Solution

Fig. 1: A CRP Instance of a Solar Plant with PVAs

the renewable energy topic, the power of quantum computers in the energy field and challenges in ILP problems inspire us to use quantum computing techniques to solve the CRP. However, there is an obstacle when we try to use quantum computing. The difficulty is converting the ILP problem into a QUBO recognized by the quantum computer and solving it quickly. This paper converts the CRP into a QUBO to overcome these challenges and validates the solution.

Contributions of this paper are summarized as follows:

- We propose a quantum computing approach to find the solution for the CRP, which is an ILP problem. Our quantum computing approach employs the quantum computer provided by D-Wave to solve the CRP with their Leap™ quantum cloud service.
- We set up several cases with different grid sizes to evaluate the performance of our quantum computing approach. Our quantum computing approach returns the optimal solution faster and more robust than the classical computer solver. It proves quantum supremacy in solving ILP problems

The rest of this paper is organized as follows. Section II introduces the basics of the CRP problem and the model setup. Section III illustrates our quantum formulation and QUBO setup. Section IV validates our algorithm using experiments. Finally, Section V concludes the whole paper.

II. Problem Definition and Formulations

In this section, we first introduce the problem setup and notations in our research. Then, we provide a classical arc-flow formulation inspired by [4] and explain the objective function and each constraint.

A. Problem Setup and Notations

The problem setup is shown in Figure 1. In our problem setup, we assume the location of every solar panel is given and fixed. Then, the realistic layout of solar panels is transferred to a grid-like map Figure 1a.

Therefore, the CRP grid-like map can be described by a directed graph $G = (V, E)$ in Figure 1b, where $V = \{0, 1, ..., n\}$ represents the whole set of vertex including the energy collection spot (V_0). Then, the rest of the grid nodes (denoted as $V' = \{1, 2, ..., n\}$) represent the rest of the PVAs. If the V_i and V_j are adjacent to each other, there is a pair of edges (i, j) and (j, i) that connect two vertices. The edge set E includes every edge in the network. Moreover, each PVA vertex in V' has a directed edge connecting the corresponding vertex to V_0 (energy collection spot), namely $(i, 0) \in E \backslash E'$, $\forall\, i \in V'$. These edges are called the bus edges of the network.

Figure 1c shows an example of a solution of a CRP with two bus edges dividing PVAs into two sub-groups. V_{11} and V_8 collect all PVAs' energy from their subset to itself and send the energy to V_0 (energy collection spot).

Figure 1d provides a spanning treemap visualization of Figure 1c. Assuming a unit of supply needs to be sent from each $i \in V'$ to V_0 (energy collection spot).

We use the following notation throughout the paper:

- $G = (V, E)$: The CRP grid-like graph; V is the whole set of vertex including the energy collection spot (V_0); E is the set of edges between every vertex including V_0.
- $E' = \{(i, j) \mid (i, j) \in E,\ i, j \in V'\}$: the set of edges between panel vertices. $E \backslash E'$ are bus edges $(i, 0)$, $i \in V'$;
- $\delta_i^+ = \{(i, j) \mid (i, j) \in E\}$: the set of outgoing edges from the vertex i;
- $\delta_i^- = \{(j, i) \mid (j, i) \in E\}$: the set of incoming edges of the vertex i;
- c_e: the cost of edge $e \in E$;
- T: the set of capacity levels (types), $T \subseteq \{1, 2, ..., Q\}$;
- $\overline{m}$: the total bus edges amount of sub-trees with a capacity equal to $t \in T$;
- m_t^l: the minimum number of sub-trees with a capacity equal to $t \in T$;
- m_t^u: the maximum number of sub-trees with a capacity equal to $t \in T$;
- $Q = |V'|$: the maximum capacity of the solar grid;
- p: the coefficient controlling the flow cost;
- $(\cdot)_d$, $(\cdot)_t$: a sub-notation to show its capacity level; d is used when we choose every capacity level $\{1, 2, ..., Q\}$. t is used when we choose the capacity level in T;
- $x_{e,d/t}$: binary decision variable that equals 1 if d/t unit(s) of flow traverse the edge $e \in E$, and 0 otherwise.

B. *Classical Arc-flow Formulation*

The formulation that is inspired by [12] and as follows:

$$\min \sum_{e\in E} c_e \sum_{d=1}^{Q} x_{e,d} + p \sum_{e\in E'} \sum_{d=1}^{Q} (d-1) c_e x_{e,d}, \tag{1a}$$

$$\text{s.t.} \sum_{e\in\delta_i^+} \sum_{d=1}^{Q} x_{e,d} = 1, \quad \forall i \in V', \tag{1b}$$

$$\sum_{e\in\delta_i^+} \sum_{d=1}^{Q} d x_{e,d} - \sum_{\hat{e}\in\delta_i^-} \sum_{d=1}^{Q} d x_{\hat{e},d} = 1,\ \forall i \in V', \tag{1c}$$

$$m_t^l \le \sum_{e\in E\backslash E'} x_{e,t} \le m_t^u, \quad \forall t \in T, \tag{1d}$$

$$\sum_{e\in E\backslash E'} \sum_{d=1}^{Q} x_{e,d} = \overline{m}, \tag{1e}$$

$$x_{e,t} = 0, \quad \forall e \in E\backslash E',\ \forall t \notin T, \tag{1f}$$

$$x_{e,t} \in \{0,1\}, \quad \forall e \in E,\ t \in \{1,2,...,Q\}.$$

The objective function (1a) aims to minimize the cable connection's total cost. Constraints (1b) guarantee that each vertex has exactly one outgoing edge with a certain flow level. Constraints (1c) represent that the flow difference between the outgoing and incoming edges of each vertex must equal a single flow unit. Constraints (1d) ensure that the solution has certain sub-trees within a range from $\left[m_t^l, m_t^u\right]$ in level $t \in T$. Constraint (1e) ensures that the solution has $\overline{m}$ bus edges. Constraints (1f) remove the certain bus edges we do not use. Note that the problem is still valid if it discards these decision variables. However, we decided to keep this constraint to generalize the proposed model. Since the model is an ILP, the set of decision variables is $\{x_{e,t}\}$ with a length of $E{\cdot}Q$. The rest of the symbols in the problem are the known parameters.

III. Quantum Formulation

Quantum annealers can solve optimization problems with QUBO formulation [13]. The QUBO problem aims to find a corresponding binary vector $\mathbf{x}^*$ that minimizes $\mathbf{x}^\intercal \mathbf{Q}\mathbf{x}$. The optimal solution can be written as follows.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{Q}\mathbf{x}, \tag{2}$$

where $\mathbf{x}$ is a binary variable vector of a length n. $\mathbf{Q}$ is a symmetric matrix. It is required to reformulate the constrained ILP to a QUBO with penalties since (2) is an unconstrained optimization model. The detailed process is introduced in III-A. Researchers in [13] proposed some rules for transforming classical constraints to their equivalent penalty pairs, as shown in TABLE I. Here x_i are binary variables. A is either a matrix or a horizontal vector. $\mathbf{b}$ is a constant vector. For the general inequality in 1, we derive its corresponding QUBO form according to the guidance of [9]. P is a user-defined penalty coefficient. For now, there is no specific route explained in the literature. It often requires a combination of theoretical understanding, experimentation, and intuition. Adjusting and refining penalties based on results and feedback is a crucial part of the process. In general, the quantum annealer computer is designed to solve (2) and provide the user with a list of answers and their corresponding energy (objective function values). We reckon the answer with the lowest objective function value is the optimal solution.

TABLE I: Commonly Used Constraint-Penalty Pairs

Constraint	**Equivalent Penalty**
$x_1 + x_2 \ge 1$	$P(1 - x_1 - x_2 + x_1x_2)^2$
$x_1 + x_2 + x_3 \le 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$
$A\mathbf{x} = \mathbf{b}$	$P(A\mathbf{x} - \mathbf{b})^2$

A. *Quantum Formulation*

Now consider the CRP problem (1). We deal with the objective function and constraints separately.

1) Objective Function: Following the principle of the QUBO formulation. Objective Function (1a) is reformulated as a quadratic term (3), which is the initial objective function.

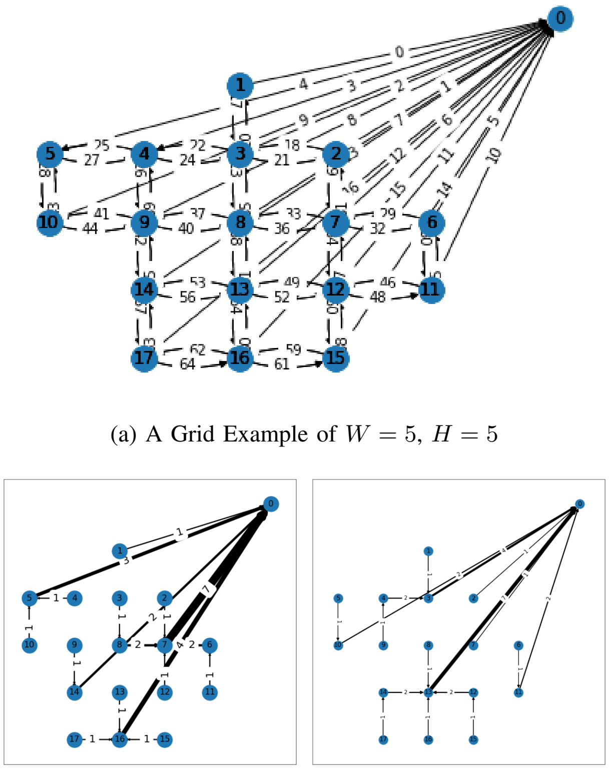
$$\begin{aligned} &\sum_{e\in E} c_e \sum_{d=1}^{Q} x_{e,d} + p \sum_{e\in E'} \sum_{d=1}^{Q} (d-1) c_e x_{e,d} \\ \Rightarrow \mathbf{Q}_{\text{obj}} = &\sum_{d=1}^{Q} \left(\sum_{e\in E} x_{e,d} c_e x_{e,d} + \sum_{e\in E'} x_{e,d} (pd - p) c_e x_{e,d} \right). \end{aligned} \tag{3}$$

2) Single Outgoing Edge: The single outgoing edge constraints (1b) can be reformulated as a quadratic penalty term (4) and added to the objective function.

$$\begin{aligned} &\sum_{e\in\delta_i^+} \sum_{d=1}^{Q} x_{e,d} = 1, \quad \forall i \in V' \\ \Rightarrow &P_i \left(\sum_{e\in\delta_i^+} \sum_{d=1}^{Q} x_{e,d} - 1 \right)^2, \quad \forall i \in V'. \end{aligned} \tag{4}$$

3) Unit Flow Difference: The flow difference constraints (1c) can be reformulated as quadratic penalty terms (5) and added to the objective function.

$$\begin{aligned} &\sum_{e\in\delta_j^+} \sum_{d=1}^{Q} d x_{e,d} - \sum_{\hat{e}\in\delta_j^-} \sum_{d=1}^{Q} d x_{\hat{e},d} = 1,\ \forall j \in V' \\ \Rightarrow &P_j \left(\sum_{e\in\delta_j^+} \sum_{d=1}^{Q} d x_{e,d} - \sum_{\hat{e}\in\delta_j^-} \sum_{d=1}^{Q} d x_{\hat{e},d} - 1 \right)^2,\ \forall j \in V'. \end{aligned} \tag{5}$$



(a) A Grid Example of $W = 5$, $H = 5$

(b) A Gurobi's Solution

(c) A QC's Solution

Fig. 2: A CRP Instance for Numerical Validation

4) Sub-tree Type Regulation: Now let us discuss how to reformulate the constraints (1d). We deal separately with the upper and lower bounds in constraints (1d).

a) Upper Bound Constraints:

$$\sum_{e\in E\setminus E'} x_{e,t} \leq m_t^u, \quad \forall t \in T$$

$$\Rightarrow P_t\Big(\sum_{e\in E\setminus E'} x_{e,t} + \sum_{i=0}^{\bar{S}_t^u} 2^i x_{i,t} - m_t^u\Big)^2, \quad \forall t \in T, \tag{6}$$

$$\text{where } \bar{S}_t^u = \left\lceil \log_2 \left(m_t^u - \min_{\mathbf{x}} \sum_{e\in E\setminus E'} x_{e,t} \right) \right\rceil.$$

According to [13], we introduce slack variables $\bar{S}_t^u$ to help us convert the inequality constraints into equality constraints. Then, the upper bound of the sub-tree type regulation (1d) can be reformulated as a quadratic penalty term (6) and added to the objective function.

TABLE II: Solver Access Time Comparison

Detail **Model**	**Var.** #: $E \cdot Q$	**Average Executing Time** *Unit: millisecond(s)*
W5H5 Gurobi	{969,1105}	51.9280
W5H5 Quantum		32.0485

b) Lower Bound Constraints:

$$\sum_{e\in E\setminus E'} x_{e,t} \geq m_t^l, \quad \forall t \in T$$

$$\Rightarrow P_t\Big(\sum_{e\in E\setminus E'} x_{e,t} - \sum_{i=0}^{\bar{S}_t^l} 2^i x_{i,t} - m_t^l\Big)^2, \quad \forall t \in T, \tag{7}$$

$$\text{where } \bar{S}_t^l = \left\lceil \log_2 \left(\max_{\mathbf{x}} \sum_{e\in E\setminus E'} x_{e,t} - m_t^l \right) \right\rceil.$$

Similarly, here we introduce slack variables $\bar{S}_t^l$ to help us convert the inequality constraints into equality constraints. Then, the lower bound of the sub-tree type regulation (1d) can be reformulated as a quadratic penalty term (7) and added to the objective function.

5) Bus Edge Regulation: The bus edge regulation constraint (1e) can be reformulated as a quadratic penalty terms (9) and added to the objective function, i.e.,

$$\begin{aligned} &\sum_{e\in E\setminus E'} \sum_{d=1}^{Q} x_{e,d} = \overline{m} \\ \Rightarrow &P \left(\sum_{e\in E\setminus E'} \sum_{d=1}^{Q} x_{e,d} - \overline{m} \right)^2. \end{aligned} \tag{8}$$

6) Special Regulation: The special regulation constraint (1f) can be reformulated as a quadratic penalty term (9) at every certain bus edge we do not use. Then, it is added to the objective function, i.e.,

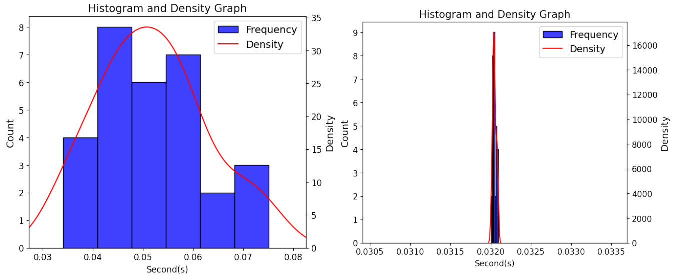
$$\begin{aligned} &x_{e,t} = 0, \quad \forall e \in E\setminus E', \ \forall t \notin T \\ \Rightarrow &P_{e,t}\left(x_{e,t}\right)^2 \ \ \forall e \in E\setminus E', \ \forall t \notin T. \end{aligned} \tag{9}$$

7) Final QUBO: We sum all QUBO matrices to the final QUBO as (10). Following the (2), we take $\mathbf{x}^\top \mathbf{Q}_{\text{sum}} \mathbf{x}$ as the input to the quantum computer. The $\mathbf{x}$ contains the original decision variables and the slack variables we introduced in (6) and (7). The quantum annealer computer will give us a list of answers, and we pick the one with the lowest energy as the final solution. The detailed analysis is in the next section.

$$\mathbf{Q}_{\text{sum}} = (3) + (4) + (5) + (6) + (7) + (8) + (9). \tag{10}$$

TABLE III: Standard Deviation Comparison

Detail Model	Average Standard Deviation Unit: 10^{-3}	Gain
W5H5 Gurobi	10.2621	99.79%
W5H5 Quantum	0.0214	



(a) Gurobi

(b) D-Wave

Fig. 3: Histogram and Density Graph of 30 Execution Attempts of Gurobi & D-Wave's Quantum computer in the Case of $W = 5$, $H = 5$ Grid

IV. Numerical Validation

We validate our proposed approach on a hybrid D-Wave quantum processing unit. The reason to choose a hybrid quantum computer is that the hybrid system provides a larger input size than the direct quantum annealing system. [14] has more details of the hybrid system. We access the D-Wave system through their Leap™ quantum cloud service project. We compare our quantum computing approach with the commercial solver Gurobi. The hardware platform is AMD™ 3900X, a high-end CPU on the current market.

A. *Experiments Setup*

In our simulation, we set up different scenarios to evaluate our approach. Figure 2a is an example in our test bench. "W" represents the maximum width of the grid. "H" describes the maximum height of the grid. Since the qubit of the quantum computer is limited. We choose the grid with 5 units of width and 5 units of height maximum. We prepared 10 random scenarios to prevent creating bias, and each graph guarantees $W = 5$, $H = 5$. In addition, the decision variable amount of every problem is 969 or 1105. Moreover, we execute both quantum and classical approaches 3 times just in case of unknown randomness. For the sub-tree type regulation constraint (1d), we set up a rule that the solution must have at least a level-7 and a level-4 sub-tree in bus edges ($m^l_4 = m^l_7 = 1$). The maximum number of sub-trees in every level is 2 ($m^u_{t \in \{1,2,\ldots,Q\}} = 2$). For the bus edge regulation constraint (1e), we set up a rule that we only want 5 bus edges ($\overline{m} = 5$). For other parameters, p and c_e are set to one unit.

B. *Simulation Result*

Figure 2b and Figure 2c show the optimal solution derived from the quantum computing and classical computing approaches. Their solutions have at least a level-4 and a level-7 bus edges. Both are optimal solutions. The success rate of the quantum computing and classical computing approach is promising. Although both approaches achieve finding the optimal solution for every problem, there is a gap between the two approaches' performance in terms of solver access time and robustness.

1) *Average solver access time:* As shown in Table II, we test 10 random scenarios of the $W = 5$, $H = 5$ grid with a decision variable size of either 969 or 1105. Compared to the classical computing approach, The quantum computing approach has a better average solver access time with a gain of 38.28%. In addition to that, quantum computing beat the classical computing approach in almost every scenario and test. In general, quantum computing has shown its computing supremacy over the classical computing approach to solving the CRP.

2) *Robustness:* In the histogram Figure 3, both sub-graphs share an identical axis layout. The x-axis is the bins with a certain time interval. The interval is automatically chosen by Seaborn, which is a library for making statistical graphics in Python. The left y-axis is the frequency of the solver access time within the corresponding time interval. The right y-axis is the density probability after fitting the result to bins of certain time intervals. The curve is the density curve of the solver access time's frequency.

The solver access time of the classical computing approach has more samples of outliers than the quantum computing approach in both test cases, which means the classical computing approach's distribution of solver access time scatters wider than the quantum computing approach. As Table III states, the quantum computing approach has a gain of 99.79% (500 times more robust) over the classical computing approach, implying that the quantum computing approach is more robust. What's more, it means that the quantum computing approach's computation performance is not sensitive to the disturbance of the parameters in the optimization model. Therefore, histograms, density graphs, and the standard deviation table support that the quantum computing approach is more robust than the classical computing approach.

3) *Analysis:* We believe the quantum computing approach can replace the classical computing approach to solve CRPs for the following reasons. First, while both achieve the 100% success rate, the quantum computing approach is better than the classical computing approach regarding average solver access time with a 38.28% gain. Second, the classical computing approach is eminently less robust in dealing with extreme problems, while the quantum computing approach is surprisingly way more robust, which yields a gain of 99.79%.

Therefore, compared with the classical computing approach, we believe the quantum computing approach has more advantages in solving the ILP CRPs.

V. Conclusion

In this paper, we first propose a QUBO model for the CRP problem based on the classical computing model. Then, we solve the QUBO-based CRP model by using the quantum annealer. The simulation study demonstrates that quantum computing performs better than the classical computing approach. From the solution quality perspective, firstly, our quantum computing approach converges and returns the correct optimal result as the classical computing algorithm. Secondly, Although limited by the total qubit amount, the quantum computing approach has a faster solver access time in solving medium-size CRPs. In addition, our approach guarantees robustness when it meets extreme CRP cases. Therefore, we can conclude that the proposed quantum computing approach is a new tool for solving CRP with great potential.

Acknowledgement

This work is partially supported by US NSF EECS 2045978, CNS-2107216, CNS-2128368, CMMI-2222810, and ECCS-2302469.

References

[1] M. J. Roney, “Solar PV breaks records in 2010: Earth policy institute,” Oct 2011. [Online]. Available: http://www.earth-policy.org/mobile/releases/solar_power_2011

[2] “Snapshot 2021 - iea-pvps,” Apr 2021. [Online]. Available: https://iea-pvps.org/snapshot-reports/snapshot-2021/

[3] “Future of photovoltaic,” Nov 2019. [Online]. Available: https://www.irena.org/publications/2019/Nov/Future-of-Solar-Photovoltaic

[4] Z. Luo, H. Qin, T. Cheng, Q. Wu, and A. Lim, “A branch-and-price-and-cut algorithm for the cable-routing problem in solar power plants,” *INFORMS Journal on Computing*, vol. 33, no. 2, pp. 452–476, Sep. 2021.

[5] M. Fischetti and D. Pisinger, “Optimizing wind farm cable routing considering power losses,” *European Journal of Operational Research*, vol. 270, no. 3, pp. 917–930, Nov. 2018.

[6] X. Ma, K. Iida, M. Xie, J. Nishino, T. Odaka, and H. Ogura, “A genetic algorithm for the optimization of cable routing,” *Systems and Computers in Japan*, vol. 37, no. 7, pp. 61–71, Apr. 2006.

[7] M. Bischoff, H. Ewe, K. Plociennik, and I. Schüle, “Multi-objective planning of large-scale photovoltaic power plants,” in *Operations Research Proceedings*, ser. Operations Research Proceedings. Cham: Springer International Publishing, Nov. 2013, pp. 333–338.

[8] M. Troyer and U.-J. Wiese, “Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations,” *Physical review letters*, vol. 94, no. 17, pp. 170 201.1–170 201.4, May 2005.

[9] Z. Zhao, L. Fan, and Z. Han, “Hybrid quantum benders’ decomposition for mixed-integer linear programming,” *arXiv preprint arXiv:2112.07109*, Dec. 2021.

[10] A. Ajagekar and F. You, “Quantum computing for energy systems optimization: Challenges and opportunities,” *Energy*, vol. 179, pp. 76–89, Apr. 2019.

[11] A. Giani and Z. Eldredge, “Quantum computing opportunities in renewable energy,” *SN Computer Science*, vol. 2, no. 5, p. 393, Jul. 2021.

[12] E. Uchoa, R. Fukasawa, J. Lysgaard, A. Pessoa, M. P. De Aragao, and D. Andrade, “Robust branch-cut-and-price for the capacitated minimum spanning tree problem over a large extended formulation,” *Mathematical Programming*, vol. 112, no. 2, pp. 443–472, Apr. 2008.

[13] F. Glover, G. Kochenberger, and Y. Du, “Quantum bridge analytics i: a tutorial on formulating and using qubo models,” *4OR*, vol. 17, no. 4, pp. 335–371, Dec. 2019.

[14] “D-Wave Hybrid Solver Service: An Overview.” [Online]. Available: https://www.dwavesys.com/resources/white-paper/d-wave-hybrid-solver-service-an-overview/