How to Solve Combinatorial Optimization Problems Using Real Quantum Machines: A Recent Survey

SOVANMONYNUTH HENG, DONGMIN KIM, TAEKYUNG KIM, AND YOUNGSUN HAN, (Member, IEEE)

1Department of AI Convergence, Pukyong National University, Nam-gu, Busan 48513, South Korea
2Department of Computer and Information Technology, Incheon Jaeneung University, Dong-gu, Incheon 22573, South Korea

Corresponding author: Youngsun Han (younsun@pknu.ac.kr)

This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00014, A Technology Development of Quantum OS for Fault-tolerant Logical Qubit Computing Environment).

ABSTRACT A combinatorial optimization problem (COP) is the problem of finding the optimal solution in a finite set. When the size of the feasible solution set is large, the complexity of the problem increases, and it is not easy to solve in a reasonable time with the current classical computer technology. Quantum annealing (QA) is a method that replaces classical simulated annealing (SA) methods that do not solve these cases. Therefore, several attempts have been made to solve this problem using a special-purpose quantum annealer to which the QA method is applied. In this survey, we analyze recent studies that solve real-scale COPs using quantum annealers. Through this, we discuss how to reduce the size of the COP to be input to overcome the hardware limitations of the existing quantum annealer. Additionally, we demonstrated the applicability of quantum annealer to COP on a practical scale by comparing and analyzing the results of the classical simulated annealing (SA) and quantum annealing (QA) method from each study.

INDEX TERMS Combinatorial optimization problem (COP), simulated annealing (SA), quantum annealing (QA), real quantum machine.

I. INTRODUCTION

Combinatorial optimization problem (COP) is the problem of finding the optimal solution from a set of feasible solutions [1], [2], [3], [4], [5]. It is closely related to various fields such as computer science, software engineering, and applied mathematics [6], [7]. Due to the nature of COP, the complexity of the problem increases rapidly as the size of the feasible solution space increases, making it difficult to find an optimal solution [8]. Therefore, as the complexity of the COP increases, it becomes more challenging to find the optimal solution in a reasonable time on a classic computer. To alleviate this problem, various studies have been actively made to effectively solve COP. Representatively, methods such as meta-heuristic [9], hybrid algorithm [10], [11], [12], and parallel implementation [13], have been proposed and evaluated. The feasibility of COP solution for each study has also been demonstrated. On top of that, the future research work of these studies was to expand their research to large-scale practical problems. However, a new concept computer with the potential to solve the COP on a practical scale has recently appeared. Quantum computing has been suggested as an efficient way to solve many important types of COP problems [14], [15], [16]. For instance, a quantum annealer can be used to derive the minimum value for a given function using a special optimization process called quantum annealing (QA) [17], [18]. There is a similar concept to quantum annealing (QA) called classical simulated annealing (SA) that can be performed on a classical computer. However, quantum annealing uses special quantum mechanics...
called quantum fluctuation [6], [18], [19], [20], [21], which makes it more efficient. Quantum fluctuation refers to the annealing process of quickly finding the optimal solutions to problems. The term annealing is one of the properties which make quantum computing special, as it is faster than SA. In this survey, we discuss studies that have solved real-world COPs using an actual quantum annealer that outperforms the CA methods. Real-world COPs cannot be directly solved on quantum annealers because of hardware limitations of the existing quantum annealer. Therefore, to overcome this, we show several methods for reducing the size of COPs to a scale that can be operated in a quantum annealer directly. Finally, we compare and analyze the performance of QA and SA methods to COPs.

In summary, the contributions of this paper are summarized as follows.

- We show the availability of an actual QA for solving COPs by discussing studies that solve COPs using a real quantum annealer.
- We classify various types of COPs into several groups and analyze each COP. This provides guidelines for solutions to COPs with similar characteristics.
- We show that the hardware limitations of the current quantum annealer can be overcome by introducing the decomposition were used in each study for the COPs.
- We compare the performance of COPs on the QA method and the SA methods. This shows extendable quantum supremacy in terms of solving various kinds of COPs.

The remainder of this paper is organized as follows. In Section II, we explain the background. Section III analyzes the COPs solved using the quantum annealer. Section IV describes the size reduction procedures for submitting the COPs to the quantum annealer. In Section V, we compare and analyze the performance of COPs on the QA method and the SA methods. Finally, the conclusion and discussion of our survey is made in Section VI.

II. BACKGROUND

This section explains the concept of COP, QA, real QA machine, and decomposition which is a key principle in solving COP. We also present various existing quantum annealers using quantum annealing (QA) methods.

A. COMBINATORIAL OPTIMIZATION PROBLEM

Combinatorial optimization problem (COP) is an optimization problem that determines the optimal solution from a finite set of solutions. Figure 1 shows the process of finding the optimal solution among a significantly large set of discrete possible solutions using an optimization algorithm after problem modeling. As the size of the COP increases, the size of the set of feasible solutions increases, and the complexity of finding the optimal solution also increases. This is a common case in real-world problems. In this case, it is not easy to solve it in a reasonable time with current generation classical computers. To alleviate this problem, various studies on software are in progress such as exact algorithms, meta-heuristics, and hybrid algorithms [10], [22], [23], [24], [25]. Moreover, various studies have been made to efficiently solve COP with strong hardware performance. Typically, meta-heuristics can effectively solve a wide variety of COPs. They are used in various fields such as finance [8], [10], [26], management system [10], [26], [27], and computing applications [6], [8], [10], which are naturally NP-hard problems. However, the current meta-heuristic trend has limitations in solving complex formulas including many constraints and objectives, which are characteristic of most practical-scale COPs. Accordingly, researchers in the field of meta-heuristics are actively conducting research to develop more effective optimization algorithms for very large sizes of COPs [9], [10], [11], [12].

B. QUANTUM ANNEALING

Figure 2 shows the process of simulated annealing (SA) and quantum annealing (QA). Both annealing processes aim to find the lowest energy of the objective function [6], [17]. The difference between the two annealings is how to find the lowest energy state of the problem. QA uses quantum fluctuation [6], [20], [21], one of quantum mechanics, to escape from local minimum, which is not the target solution. By avoiding falling into local minimum, it can find global minimum which is the lowest energy state among possible local minimum. In the case of SA, the value of the
lowest energy is obtained by following the energy landscape with a “thermal jump” based on the method of selecting the neighboring state. Generally, SA requires a sufficiently long annealing time, which increases the probability of finding a global minimum without falling into a local minimum. On the other hand, QA can pass throw high energy peak without requiring to follow the energy landscape like SA through the “quantum tunneling” effect [6]. Therefore, QA process can find the global minimum significantly faster than classical SA process [8], [17], [18].

QA uses the quantum adiabatic process [6], [8], [17] starting from the preparation of the ground state by the Hamiltonian which is called the Initial Hamiltonian. Then, we evolve adiabatically to the system until it could transform into another problem of Hamiltonian called Final Hamiltonian [6], [8], [17]. The Final Hamiltonian can be the optimal solution to the problem that the user wants to solve. The QA machines uses its QA process to solve large-scale optimization problems by transforming them into quadratic unconstrained binary optimization (QUBO) or Ising model, which can be operated in one of the real QA machine. When QUBO or Ising model is inserting into the real QA machine, the real QA machine formulate the outputs of its solution. In most cases, this first output solution is not the real minimum or local minimum solution for COP, so refer to the first output value and configure a new QUBO or Ising model through parameter adjustment. Then, the process of solution formulating the real QA machine is updated and repeated several times. Up to the time is the real ground state or the global minimum of the QUBO or Ising model. Meanwhile, an global solution can be obtained if it is formulated according to the solution of the COP. We describe the characteristics of the three real QA machines for solving COP, and the D-Wave quantum annealer is mainly covered in this survey paper.

C. REAL QUANTUM ANNEALING MACHINE

In this section, we describe the three representative real QA machines: CMOS quantum annealer, digital quantum annealer, and D-Wave quantum annealer, which are used for research and commercial purposes. Figure 3 shows the process of solving COP using the real QA machine. Since COP cannot be directly resolved in the real QA machine, it is necessary to formulate the COP in the form of the QUBO or Ising model, which can be operated in one of the real QA machine. When QUBO or Ising model is inserting into the real QA machine, the real QA machine formulate the outputs of its solution. In most cases, this first output solution is not the real minimum or local minimum solution for COP, so refer to the first output value and configure a new QUBO or Ising model through parameter adjustment. Then, the process of solution formulating the real QA machine is updated and repeated several times. Up to the time of the result is the real ground state or the global minimum of the QUBO or Ising model. Meanwhile, an global solution can be obtained if it is formulated according to the solution of the COP. We describe the characteristics of the three real QA machines for solving COP, and the D-Wave quantum annealer is mainly covered in this survey paper.

1) CMOS ANNEALER

Complementary Metal Oxide Semiconductor (CMOS) annealer is a special purpose computer developed by Hitachi to solve COP. This is a quantum annealer that performs optimization using the Ising model using the structure of SRAM with non-von Neumann architecture. CMOS annealer can achieve high directivity based on CMOS integrated circuit
technology [26], [28], [29], [30]. In other words, multiple spins are mounted on one chip, and high scalability can be used by connecting multiple chips. Therefore, when it is not possible to construct an Ising model for COP on a practical scale with a limited number of spins on one chip, it is possible to connect multiple chips by utilizing scalability and ultimately solve COP. Additionally, CMOS annealers have no restrictions on the operating environment.

2) DIGITAL ANNEALER
Digital annealer is a QA machine developed by Fujitsu Laboratory. Digital annealer also follows the same non-von Neumann computational architecture as CMOS annealer for solving large-scale COP [26], [31], [32], [33]. Additionally, there are no restrictions on temperature since the machine can be operated at room temperature. Unlike CMOS, 8,192 fully connected bits are used to solve large-scale COPs, and 64-bit gradation provides a high-accuracy representation of COPs. Additionally, a digital annealer offers a feature for solving a fully connected problem expressed as QUBO or Ising model. This model can be formed from a web application programming interface, and it can formulate the problem to find the optimal solution for COP.

3) D-WAVE ANNEALER
D-Wave quantum annealer is manufactured by D-Wave System Inc of Canada and provides the world’s first quantum annealer for commercial use. Starting from D-Wave One with 128 qubits released in 2011, Advantage with currently available 5640 qubits, and Advantage 2 with more than 7000 qubits being developed. They have a very large number of qubits compared to the limited number of qubits of current general-purpose quantum computers; thus, COP can be effectively solved using QA technology [6], [7], [34]. Therefore, all of the studies to be discussed in this survey use D-Wave quantum annealer to solve practical-scale COPs. Starting from the abstract QA problem formulation, decompose the problem into a form that can be fitted with the QA machine. Then compile it on the cloud service, a process called minor embedding. In the first step, the D-Wave annealer defines the problem of any chosen problem cases into energy minimization. The second step is to decompose the problem into a subproblem by introducing two equivalent forms of the objective function: the Ising model and the QUBO model. Both types are capable of formulating COPs and expressing a single problem using different representations. It is not known which of the two types will give the best results in D-Wave [17], [26]. The quality of the results depends on how well the COP is formulated whether it is a QUBO or Ising model. Thus, it is essential to transform it into a formulation that can operate optimally in quantum annealer according to COP. Additionally, the Ising model can be converted to the QUBO model and vice versa [8], [17], [26]. On the strength of the conversion, D-Wave is first introduced as a general Hamiltonian [8], [10], [35], as shown in Equation 1:

\[ H(t) = A(t)H_0 + B(t)H_1 \]  

Specifically, D-Wave is structuring problems called an Ising Hamiltonian [34], [35], [36], as shown in Equation 2:

\[ \mathcal{H}_{\text{Ising}} = -\frac{A(s)}{2} \sum_i \sigma_i \sigma_j^z + \frac{B(s)}{2} \left( \sum_i h_i \sigma_i^z + \sum_{i>j} J_{i,j} \sigma_i \sigma_j^z \right) \]  

The Ising Hamiltonian generated from Equation 2 is not totally connected, so the embedding process is still required. The D-Wave machine implements the Ising Hamiltonian model into the Chimera graph for full connectivity. Once the problem is fully connected, the developer still has to compile the corresponding Ising Hamiltonian model onto an actual QA hardware or cloud service.

D. DECOMPOSITION
Owing to the fact that solving the COPs requires finding the optimal solution to a large-scale complexity problem, these can be converted to Ising or QUBO models that can be efficiently solved with quantum annealing (QA) in a real quantum annealer. Despite the current quantum hardware limitations, the size of the original COPs still can be a challenge. We cannot directly transform the original COPs via quantum computing with the current quantum hardware limits. To solve this problem, each COP problem requires a different type of algorithm to decompose it into smaller problem sizes, in order to deal with the hardware resources known as decomposition or size reduction [3], [37], [38], [39], [40], [41], [42]. Decomposition is a methodology that reduces the size of the original problem by breaking it down into smaller pieces [43], [44], [45], [46]. Many optimization algorithms for very large sizes of COPs are required to optimize the different types of COP problems to produce a temporally smaller-sized COP, representing the original problem [10], [11]. With this, it can be performed faster, provides better solutions, and uses fewer resources to fit into the real quantum annealer in different problems of its own category.

III. PROBLEM CATEGORIZATION
In this recent survey, we classify numerous of COPs into four different categories, each of which is graph partitioning and clustering, factorization, prediction, and other well-known problems.

A. GRAPH PARTITIONING AND CLUSTERING PROBLEMS
This section describes the graph partitioning and clustering problems, including graph partitioning (GP), community detection, detecting multiple communities, maximum clique (MC), and core-periphery partitioning problems.
1) GRAPH PARTITIONING
The GP problem is a general problem that arises in almost all the fields of mathematics, computer science, chemistry, physics, bio-science, machine learning, and some complex systems [47]. GP is a problem for reducing the size of the graph into a smaller subgraph by partitioning its sets of nodes into a group of clusters. A graph can be partitioned into different communities of different numbers of nodes with high intraconnectivity and low interconnectivity. Ushijima et al. [48] could not define the size of the graph before the division of each community; however, they later balance the graph partition of the community structure by its modularity. The computational resource that they have currently is smaller than the size of the graph problem. To solve the GP problem, there are a few techniques that are commonly used to partition the graph into smaller subgraphs to reduce the complexity or enable parallelization of the problem. Ushijima et al. [48] demonstrated how the GP problem as a COP could be mapped onto the D-Wave hardware with QA to produce a better result.

2) COMMUNITY DETECTION
In various computation problems, quantum computers are assigned the role of accelerators to address the problem from biology to social network analysis. Through such a complex network of the problem, the community detection technique is used to discover and divide each group into a set of nodes and keep them tightly connected to each other. Community detection or graph clustering is one of the potential methods for partitioning the graph or community into a structure [49], [50]. It is used to solve the problem under three main challenges. The first challenge that Shaydulin et al. [51] tried to solve the community detection using quantum computing. Another challenge is how the limited number of noisy qubits of NISA hardware can solve the large-scale real-world problem. The last challenge is to illustrate one portable and extendable method that can fit into both leading quantum computation paradigms of gate-based universal quantum computing and QA. Shaydulin et al. [51] addressed community detection in two communities and obtained a better modularity solution by splitting the nodes of the graph into different communities.

3) DETECTING MULTIPLE COMMUNITIES
A community detection problem is a COP that can partition the network into communities of densely different connected nodes [52]. The connectivity between nodes belonging to different communities is smaller than those inside a community itself. This problem is important across many fields from chemistry and biology, to the social sciences. There will be twofold problems with these multiple community detection problems that must be solved simultaneously. The first one is to determine the number of communities, and the second is to find those communities all at once. There are many solutions to solve this kind of problem; however, the solution should obtain the “highest quality” possible. This is because the entire community structure can be affected by even the small community with low-quality structure. Negre et al. [53] addressed the multiple community detection problems for two or more communities in a network using QA and compared it with the SA method.

4) MAXIMUM CLIQUE PROBLEM
The MC problem is finding the clique, which is a complete subgraph in a given undirected graph. This is an NP-hard COP, which has important applications in network analysis, bioinformatics, and computational chemistry. Several studies have been made to quickly solve the MC problem by converting the COP format to the QUBO format using a D-Wave machine [54], [55]. However, previous works showed limitations such as fixed input graphs and no performance comparison with the state-of-the-art classical approaches. Chapuis et al. [56] solved the MC problem for an arbitrary input graph and compared it with the state-of-the-art classical approaches. Moreover, it illustrates the algorithm for graph splitting which reduces a large-scale MC problem to a size that can be operated on a D-Wave quantum annealer using various tools and strategies provided by the D-Wave System company. It also demonstrates the performance of the implemented QUBO using the classical algorithm on various graphs in terms of speed. The result of using quantum annealer shows the potential power for arbitrary large-scale MC problems.

5) CORE-PERIPHERY PARTITIONING
In a network, the structure can extract high-level information using the main tools such as clustering or community detection. However, Higham et al. [57] discovered a new core-periphery structure for an undirected network that is observed to rise in various fields specified in the data collection process. The core-periphery partitioning is a structure that expresses itself as only one core node of the network where the other nodes are highly connected, also known as periphery nodes. Higham et al. [57] observed that this type of structure fits in most fields [58]. To perform this structure, it has a new objective function to perform on two essence concepts. The first concept is to determine the quantum annealer’s performance. The second concept is to differentiate between the existing method with this new structure of the core-periphery method. Higham et al. [57] introduced the QA experiment with QUBO formulation of core-periphery partitioning on the Advantage 4.1 system from the D-Wave quantum annealer.

B. FACTORIZATION PROBLEMS
This section explains the COPs under the category of factorization problems in particular prime factorization and matrix factorization problems.

1) PRIME FACTORIZATION
The problem of prime factorization also known as integer factorization is breaking an input number as integer N down
to its factors $p$ and $q$ such that $qp = N$, which multiply together to result in the original number. In order to convert a random integer factorization problem, Jiang et al. [59] have developed a new framework using the Ising model. The results of the factors of an integer $N$ using $O(\log^2(N))$ binary variables (qubits) [60], [61]. At first, it forms the problem and initializes it into an optimization function. After that, it can transform the k-bit coupling $(k \geq 3)$ terms by using ancillary variables into quadratic terms. Jiang et al. [59] showed how the new framework could compute the results by the factorization numbers 15, 143, 59989, and 376289 using 4, 12, 59, and 94 of the logical qubits in quantum computation, respectively. Jiang et al. [59] solved the prime factorization problem with the new framework using QA on D-Wave 2000Q quantum annealer.

2) MATRIX FACTORIZATION
Matrix factorization takes a high-dimensional matrix as input and decomposes it into two low-dimensional matrices. This has been dealt with as a fundamental problem in the field of applied mathematics for many years, and LU and QR decomposition have been widely used to accomplish this. Additionally, matrix factorization can be actively used in machine learning, mainly used to extract features from data. Various feature extraction algorithms have been developed to avoid overfitting and the curse of dimensional, making much progress in image analysis using machine learning. Matrix factorization is one of the most important research tasks in terms of image analysis because it is possible to extract features from a wide variety of datasets regardless of the content of the image [62], [63]. O’Malley et al. [62] applied the input matrix method to facial images to acquire the property and compare the performance of the D-Wave quantum annealer.

C. PREDICTION PROBLEMS
This section presents the prediction problems using D-Wave quantum annealer, such as finite-set model predictive control (MPC) and feature selection problems.

1) FINITE-SET MODEL PREDICTIVE CONTROL
The finite-set MPC problems are classified as an NP-hard COP, which makes it difficult to optimize in a real-time system out of the finite solution sets [7]. The objective of the finite-set MPC problem is to improve the state-control prediction accuracy and speed of a target system, compared to classical control methods. To allow for a finite number of input values for the prediction, finite-set MPC is one of the modern control algorithms that can be considered. This method is more flexible than others to fit with a large time constants system, and allows only a predetermined finite number of input values in the evaluation function to meet user control requirements. Inoue et al. [7] use two scenarios, stabilization of a spring-mass-damper system and dynamic audio quantization, to demonstrate this finite-set MPC problem. To solve this problem, Inoue et al. [7] propose a method to convert the original finite-set MPC problem into a QUBO model, which is then solved by the D-Wave 2000Q quantum annealer.

2) FEATURE SELECTION
Feature Selection is a paradigm that can be used for classification or prediction models, and is categorized as an NP-hard COP [37]. The recommender system optimizes the sample by selecting the appropriate set of features from a vast data catalog. This is usually a demanding and time-consuming task that requires remarkable domain of past user interaction data (also called collaborative information) and the attributes of each item (also called content information). To design an effective hybrid feature selection algorithm for particular recommender systems, there is a need to explore numerous schemes, exploration, and behavior hidden in the past user interaction data. Nembrini et al. [37] introduce two scenarios, cold start scenarios and no purely collaborative recommender scenario, for this feature selection problem. To solve this problem, Nembrini et al. [37] propose an effective hybrid feature selection method for a recommender system represented in its QUBO formulation and conducted by the D-Wave quantum annealer of D-Wave Leap Hybrid V2 and D-Wave Advantage.

D. OTHER WELL-KNOWN PROBLEMS
This section explains other well-known problems in a D-Wave quantum annealer, such as support vector machines (SVMs), nurse scheduling problems (NSP), and traffic flow optimization problems.

1) SUPPORT VECTOR MACHINE
Kernel-based SVM is one of the supervised machine learning algorithms used to solve image classification and linear regression problems. More precisely, it is responsible for training the model parameters from a set of labeled training data to make correct guesses on the test data. These SVMs are known to have higher stability than decision trees or deep neural networks that perform the same role [64], [65]. Therefore, there is an advantage that small fluctuations made by some data in the training data do not have a large effect on the classification result. As opposed to deep learning which requires a large amount of training data, SVM can be used with only a small amount of training data. When deep learning and SVM are used together, a huge advantage can be obtained compared to classifying with only SVM. Willsch et al. [66] presented the studies and formulation to implement the kernel-based training of SVMs could be straightforwardly expressed as a QUBO formulation using D-Wave 2000Q quantum annealer.

2) NURSE SCHEDULING PROBLEM
NSP is categorized as an NP-hard COP [34]. This problem aims to find the optimal solution to assign the schedule for several available nurses over a balanced timetable of two or three shifts while respecting constraints on schedule
and personnel. The two main constraints that NSP has to respect are hard and soft constraints. The first instance of NSP has applied the two-shift system, the day and night duty shifts, to optimize the number of nurses assigned to each duty shift while respecting the period of rest between the duty of each nurse. To make the system more sophisticated, Ikeda et al. [34] introduced additional constraints of three-shift systems: daytime, early-nighttime, and late-night time shifts. Ikeda et al. [34] introduced a healthy working environment to the hospital for all nurses while respecting the constraints. The main prioritization constraints are on the day-off request and the number of duties on late night time shifts for each nurse. Ikeda et al. [34] demonstrated the different performances of how forward annealing and reverse annealing with SA on a real-world problem like NSP.

3) TRAFFIC FLOW OPTIMIZATION
Neukart et al. [67] mimicked a system of traffic workflow to optimize the congestion problem in real-time. To develop this system workflow, they used the T-Drive trajectory data of cars’ GPS coordinates of 10,357 taxis recorded over one week. There are 15 million data points and 9 million kilometers of trajectories on the dataset. The objective function of the traffic flow optimization problem aims to reduce the travel time of each car between its individual sources to the final destination. The proportional to the objective function of the number of cars is the assumption of the travel time currently occupying the street. Thus, it can reduce the total travel time for each car by reducing the total congestion on the road segments. The quadratic function of the number of travel cars in a specific time commute is determined by the traffic congestion on an individual segment on the road. Neukart et al. [67] splitted the problem into subproblems using the CA and QA method.

IV. METHODS
In section III, we investigate how various types of COPs can be classified into different category and what each COP is. Additionally, we emphasized the importance of the study for each COP. This section introduces a numerous methods for solving the COPs as classified in the previous section with a D-Wave quantum annealer.

A. GRAPH PARTITIONING AND CLUSTERING PROBLEMS
This section describes the existing method for solving the large-scale optimization problem under the category of graph partitioning and clustering problems, such as graph partitioning (GP), community detection, detecting multiple communities, maximum clique (MC), and core-periphery partitioning problems.

1) GRAPH PARTITIONING
There are a few concepts and benchmarks for solving GP problems in different communities. It is commonly called GP, clustering, or community detection. The first two benchmarks which are introduced in this GP are called graph clustering with Thresholding and GP for Walshaw. These two approaches used the same concept of nodes with high intraconnectivity and low interconnectivity to partition the graph into a community structure. The quality of community structure is based on the modularity of graph $G = (V, E)$ with matrices $A$ and $B$ of a vector degree $g$, nodes $i$, edges $ij$, and $2m = \sum_i g_i$ expressed as follows Equation 3:

$$B_{ij} = A_{ij} - \frac{g_i g_j}{2m} = A_{ij} - \frac{g_i g_j}{\sum_i g_i}$$

Another approach for solving the GP problem is k-concurrent. It allows the GP problem to partition a graph into a super-node that consisting of several partition $k$ subnodes encoding either “0” or “1”. This general approach allows the GP problem to partition the graph without recursion and there will be only one subnode as “1” and the rest of the subnodes are “0”. By performing the GP, it can influence the other adjacent vertices within the same part. There will be only subnodes in each super-node that can be set as the graph problem. GP problem using the k-concurrent approach represents the partitioning problem as QUBO in matrix notation as $kN \times kN$ [68], [69], [70]. To quantify the community structure, it used modularity to compare the connectivity edges within the community. Then, it defined the GP problem as a graph, and later formulated it as a problem in order to transform it into an Ising or QUBO model. Ushijima et al. [48] performed the calculation using the QUBO variable that can be mapped onto the Chimera graph of the D-Wave system.

2) COMMUNITY DETECTION
To solve the community detection problem, Shuydulun et al. [51] splitted the nodes of a graph $G = (V, E)$ into different communities while obtaining a better solution for modularity. Due to the limited available number of qubits, quantum computers cannot directly solve a real-world large-scale problem or network. Thus, hybrid quantum-classical local-search algorithms are used to solve these problems in both QA and universal quantum computing. The community detection algorithm tried to get the community detection problem into a quantum computer by splitting the problem into a subproblem [49], [50].
Algorithm 1 shows the input subproblem from the global community detection graph problem to fit the target quantum annealer. First, it randomly selects the community detection problem of Graph $G$. Then, it selects a subproblem by taking the vertex $P$ with the highest potential gain. If the solution is greater than the current result $R$, then result $R$ is the optimal solution. These steps are repeated until each vertex can be efficiently computed if moving them from one community to another. After that, the output result $R$ of each subproblem satisfies the size of the current quantum annealer in order to encode them as a boundary condition and maximize solved using quantum computing.

3) DETECTING MULTIPLE COMMUNITIES

Many solutions have been used to solve community detection; however, there is a gap in quality that needs to be solved. When partitioning the graph into the community, it required the “highest quality” possible since the results can be influenced by each community’s quality [49], [50]. Moreover, Negre et al. [53] partitioned the graph into two or more communities using the k-concurrent approach with the concept of a logical super-node all at once. GP problem using the k-concurrent approach allows the problem to partition a graph into a super-node consisting of several partition $k$ subnodes encoding either “0” or “1”. Moreover, this general approach allows for the partition of the graph without recursion, and there will be only one of the subnodes $k$ as “1” and the rest of the subnodes are “0”. There will be only subnodes in each super-node that can be set as the graph problem and lead to QUBO in matrix notation to partition the graph into more than two communities [68], [69], [70]. Let $G = (V, E)$ be a graph with vertex set $V$ and edge set $E$. The modularity matrix $B$ can be constructed differently from matrix $A$ expressed in Equation 3. Additionally, D-Wave 2X and 2000Q machines with QA are used to partition the problem into the architecture with almost no need for reformulation.

4) MAXIMUM CLIQUE PROBLEM

Chapuis et al. [56] used D-Wave 2000Q machine to solve arbitrary graph size of MC problems. D-Wave 2000Q is a QA quantum annealer with about 1000 qubits, and the topology has a Chimera graph. This graph can provide a natural way for solving Ising or QUBO problems. However, it is necessary to overcome the limitation of the limited number of qubits of the quantum annealer for the arbitrary large-scale MC problems. Therefore, there are several algorithms for reducing the size of the input graph to solve the random MC problem. Through this, the input graph is divided into small subgraphs by removing vertices and edges that do not belong to the MC. The following algorithms are used for making smaller subgraphs.

- Extracting k-core: Solving the MC problem using the k-core approach is finding the clique in the k-core of a graph $G = (V, E)$. The k-core approach is initially picked a vertex $V$ with the degrees less than $k$, take off $V$ and its adjacent edges $E$, modernize the degrees of the existing vertices, and duplicate it if the vertex $V$ still remains.

Algorithm 2 shows the reduction of the size of the graph by first applying to extract the k-core to solve the MC problem. It involves two parameters of graph $P$ and lower bound $LB$. It also combines another approach to choose vertex $Q$ from graph $P$, randomly. For each neighbors of graph $P$ and vertex $Q$, repeating this process by updating the degrees of the remaining vertices until reducing the size of an input graph $P$. By applying this will be changed in graph structure, so at the end apply another extracting the k-core before returning the result of reduced graph.

- Graph partitioning: There are several GP approaches that have been used such as the divide-and-conquer approach. This approach divides the graph into smaller subgraphs in order to solve the clique problem. First, it partitions the graph by dividing the vertices and the cut edge of the graph equally. Then, it combines the subgraph solution into one main solution to solve the original MC problem. Alternatively, Chapuis et al. [56] introduced another CH-partitioning approach for solving the clique problem. Following this approach, it has to divide the problem into a subset by a core partitioning set vertex $C_i$ as a nonempty core sets $C_1, \ldots, C_s$ and halo set vertex of each core set $H_i$ as

$$\text{cost}(\mathcal{P}) = \max_{1 \leq i \leq s} |C_i| + |H_i| \quad (4)$$

- Vertex splitting: The vertex-splitting partitioning is indistinguishable from the above approaches that divide the problem of graph $G$ into subproblems determined by a single vertex. In this way, one subproblem $G_1$ contains all neighbors without vertex, and another subproblem $G_2$ contains only all vertices described as the cost of such given by Equation 4. In case CH-partitioning fails, the vertex splitting partitioning approach is being used
to solve the problems by generating the size of the subproblems smaller than the original one.
- Combine all three methods: As a consequence of the above inefficient approaches, it combined all the approaches into one suitable approach to solve the MC problem on the D-Wave system. With the limitation of the D-Wave, it breaks down previous Algorithms 2 by giving the same input of graph P into smaller subgraphs with a vertices limit. First, it splits the graph P into subgraphs SP and SSP with a vertices limit VL and lower bound LB. Hence, the list of subgraph R is initialized with the original graph P. While the largest subgraph in the list of subgraphs R is greater than the vertex limit VL, the current largest graph SP in the list is returned. Then subgraph SP is removed from the list, and a vertex Q and subgraph SP are chosen for another subgraph SSP. It continuously iterates the steps until it produces subgraphs that fit the D-Wave size limit expressed as Algorithm 3.

Algorithm 3 Graph Splitting

```
INPUT P: Graphs, VL: Vertex Limit, LB: Lower Bound
OUTPUT R: Subgraphs, LB: Lower Bound

1: procedure split(P, VL, LB)
2:     R = [P]
3:     while length(R[−1]) > VL do
4:         SP = R.pop()
5:         Q = choose_vertex(SP)
6:         SSP = extract_subgraph(Q, SP)
7:         remove_vertex(Q, SP)
8:         reduce_graph(SP, LB)
9:         if length(SP) > 0 then
10:             if length(SP) <= VL then
11:                 LB = solve(SP)
12:             else
13:                 sorted_insert(R, SP)
14:             end if
15:         end if
16:         reduce_graph(SSP, LB)
17:         if length(SSP) > 0 then
18:             if length(SSP) <= VL then
19:                 LB = solve(SSP)
20:             end if
21:         end if
22:     end while
23: end procedure
```

5) CORE-PERIPHERY PARTITIONING
As mentioned above, the QUBO problems are expressed as matrices of the network even in the sparse network with a given undirected and unweighted network of N nodes. It denotes the adjacency matrix \( a_{ij} = 1 \) with edges \( i \) and \( j \) and otherwise \( a_{ij} = 0 \) without recursion. It uses \( x_i = 1 \) to assign node \( i \) to the core and \( x_i = 0 \) assigns node \( i \) to the periphery of the core-periphery vector. The starting point of the objective function is to form the QUBO formulation expressed as follows:

\[
\sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \max \{x_i, x_j\} \quad (5)
\]

Then, it expands the problem to QUBO form which has a full matrix \( Q \) written as:

\[
\max_{x \in \{0,1\}} x^T Q x \quad (6)
\]

It can also modify the QUBO form with QA conducted on the D-Wave Advantage 4.1 system.

B. FACTORIZATION PROBLEMS

This section describes how quantum annealer performed to solve two types of factorization problems on size reduction.

1) PRIME FACTORIZATION
Jiang et al. [59] introduced a new procedure using two methods for implementing integer factorization within the QA model. The factors of an input integer \( N \) can be encoded by the problem Hamiltonian \( H_p \) for both the direct method and the modified multiplication table methods. The factors of the \( N = pq \) factorization problem, where \( p \) and \( q \) are prime numbers, can be computed using the first method of the direct method. The explicit multiplication of the binary representation for \( p \) and \( q \) yields a sum of binary products that can be defined by the cost function of \( f (x_1, x_2, x_3, x_4, \ldots, x_{l_1 + l_2 - 2}) = \overline{2^l} \), where \( l_1 = \lfloor \log_2(p) \rfloor \) and \( l_2 = \lfloor \log_2(q) \rfloor \). The local minimization over the individual binary substring bits representing the integers \( p \) and \( q \) is used in a modified multiplication table method. This type of method is used to find the factorization number by dividing the table into multiple blocks and calculating each block separately to produce a final result. It can also determine each block size in order to get the balance of the number of variables and the parameters within the block. The experiment tested both methods using quantum computing hardware with QA.

2) MATRIX FACTORIZATION
The matrix factorization problem is decomposed as matrix \( M \) representing \( x \times y \) into two matrices of matrix \( A \) as \( x \times z \) and matrix \( B \) as \( z \times y \). The study aims to solve the problem from one of each column in matrix \( B \). Here, limits are given for matrices \( A \) and \( B \). All elements of matrix \( A \) must be greater than or equal to zero, and all elements of matrix \( B \) must be binary. Therefore, O’Malley et al. [62], the corresponding matrix factorization is called nonnegative/binary matrix factorization (NBMF) because of each limitation. To solve this matrix factorization propagation, the following algorithm modified from the least-squares is used. Algorithm 4 takes advantage of D-Wave from D-Wave 2X quantum annealer to compute matrix \( B \) where \( \| \cdot \|_F \) is the Frobenius norm and \( AX \) is the column of matrix \( B \). By doing this, it can decompose...
D-Wave 2000Q quantum annealer. The QUBO expression takes form with Equation 8 to solve in the Solver API library from H and converts the it into a QUBO problem. In case of classical SA, it uses Equation 7 as a control problem called an optimal control problem. All the possible combinations of solutions. It is a structure size each element of either 0 or 1. A key output is the matrix factorization problem into two matrices, and even relatively large problems can be analyzed using a D-Wave quantum annealer.

C. PREDICTION PROBLEMS
This section explains prediction problems category in a D-Wave quantum annealer, such as finite-set model predictive control (MPC) and feature selection problems.

1) FINITE-SET MODEL PREDICTIVE CONTROL
Inoue et al. [7] introduce two scenarios such as stabilization of a spring-mass-damper system and dynamic audio quantization. Stabilization of spring-mass-damper system is an unstable physical system that trying to optimize stabilization control. Dynamic audio quantization is an audio signal of human auditory characteristics. To validate their performance in these two problems, they used three different methods including exact solutions, classical simulated annealing (SA), and quantum annealing (QA). The exact solution is the general brute-force search algorithm to find the minimum value of $u(t)$ with parameter length $N$ in Equation 7 by generating all the possible combinations of solutions. It is a structure problem called an optimal control problem.

$$u^*(t) := \arg\min_{u(t) \in U} H$$  \hspace{1cm} (7)

In case of classical SA, it uses Equation 7 as a control problem and converts the it into a QUBO problem $H$ of Equation 8. It solves using the anneal library included in the D-Wave Ocean software package. Lastly, QA method is also uses Equation 7 as control problem and converts it into a QUBO problem of Equation 8 to solve in the Solver API library from D-Wave 2000Q quantum annealer.

$$H = b(t)\T J(t) b(t) + h(t)\T b(t) + c'(t)$$  \hspace{1cm} (8)

The QUBO expression takes form with $b(t)$ as binary design variable, $J(t)$ as a matrix, $h(t)$ as a vector, and $c'(t)$ as a function method at time $T$. Then, the problems are capable to solve in D-Wave 2000Q quantum annealer.

2) FEATURE SELECTION
Nembrini et al. [37] introduce two scenarios of three different datasets for this feature selection problem: the Movie, CiteULike-a, and Xing Challenge 2017 dataset. To solve this problem, the authors propose an effective hybrid feature selection method (CQFS) using QA and classical SA. The author compares the propose CQFS results with the other three baseline algorithms: ItemKNN, TFIDF, and CFeCBF. Each of these algorithms represents a QUBO formulation of its original input problem, small enough to fit with the current quantum annealer.

- ItemKNN: It is the popular baseline algorithm that uses the item-based nearest-neighbor model. These algorithms are applied to compare the item features in terms of similarity with all the original item features.
- Time Frequency-Inverse Document Frequency (TFIDF): It is the second baseline algorithm that uses a variant of ItemKNN in which the features are filtered as a certain quota of those with the highest score.
- Collaborative-Filtering-enriched Content-Based Filtering (CFeCBF): It is the third baseline algorithm that appear to compare with the proposed method in which to approximate the feature weights in a collaborative mode.
- Collaboration-Driven Feature Selection Quantum Annealing (CQFS QA): It is the proposed method to feature the selection and embed it on QPU. This method selects the feature according to how the users interact with them and create a domain knowledge of the user behavior. Unlike the baseline methods, the selection is based on the information-theoretic metric or information retrieval.
- Collaboration-Driven Feature Selection Simulated Annealing (CQFS SA): This proposed method feature exactly the same as the above method but solves on a classical solver using simulated annealing (SA).

Each of these algorithms conducts using both quantum-based and classical-based. In the case of quantum-based, the Movie dataset and CiteULike dataset used D-Wave Leap Hybrid V2 quantum annealer, and Xing Challenge 2017 used D-Wave Advantage quantum annealer. The classical-based uses simulated annealing (SA).

D. OTHER WELL-KNOWN PROBLEMS
This section describes three studies classified as other well-known problems: support vector machine (SVM), nurse scheduling problem (NSP), and traffic flow optimization methods. The method of embedding large COPs is essential because it can be used for various types of COPs.

1) SUPPORT VECTOR MACHINE
Unlike conventional SVMs, quantum SVM automatically generates several optimal solutions to a given optimization problem. The combination of these solutions showed the potential to improve the classification performance on test data compared to SVM. Willsch et al. [66] employed the D-Wave 2000Q to generate such solutions. Additionally, in previous SVM studies using quantum computers, gate-based quantum computers were used as general-purpose quantum computers. However, very few studies were conducted, and most of the classification tasks were already performed in the preprocessing process. Willsch et al. [66]

Algorithm 4 Alternating Least Squares
INPUT $M$: Matrix, $z$: Columns
OUTPUT $A$: Matrix, $B_{2\times y} = \{0, 1\}$: Matrix randomly initialize each element of either 0 or 1

1: while not matched do
2:   \[ A := \arg\min_{x \in \mathbb{R}^{n \times k}} \|M - XB\|_F + \alpha \|X\|_F \]
3:   \[ B := \arg\min_{x \in \{0, 1\}^{2 \times y}} \|M - AX\|_F \]
4: end while








classified the problem using a commercial quantum annealer with a larger number of available qubits than a gate-based quantum computer. In the process of training the training data, a special method for overcoming the hardware limitations of the current quantum annealer has not been discussed by using input data that can be directly trained in the quantum annealer. However, Willisch et al. [66] conclude that SVMs trained on quantum annealers would be very valuable for classification tasks on possible difficult problems with several small training data. Following this implementation, it can discover the optimal solution to the SVM problem and embed it directly onto the D-Wave expressed as a QUBO formulation.

2) NURSE SCHEDULING PROBLEM
Ikeda et al. [34] used a randomized embedding algorithm with D-Wave 2000Q machine for the NSP. First, it compares the annealing method of forward annealing with SA. Then, it gets the result of forward annealing to improve into a better result using the reverse annealing method [34]. Theoretically, forward annealing is a method for initializing the Hamiltonian \( H_0 \) to the final Hamiltonian \( H_1 \) to find the ground state and measure it simultaneously [71, 72, 73]. Similarly, reverse annealing is a method that is used to build on the result of forward annealing by annealing backward of the process in order to produce better results [72, 73, 74]. Solution frequency can be estimated in these studies by setting the annealing time of 200\( \mu \)s and 1000 samples for each problem. The function of applying to the two-shift system and constraints \( G(n, d) = h_1(n)h_2(d) \) in such a way is given by:

\[
\begin{align*}
    h_1(n) &= \begin{cases} 
    3 & \text{busy} \\
    2 & \text{moderate} \\
    1 & \text{idle} 
\end{cases} \\
    h_2(d) &= \begin{cases} 
    2 & \text{weekend or night} \\
    1 & \text{weekday} 
\end{cases} 
\end{align*}
\]

It can also turn the options of the work by \( h_2(d) \):

\[
\begin{align*}
    h_2(d) &= 2 \quad \text{weekend or night} \\
           & 1 \quad \text{weekday} 
\end{align*}
\]

It can add another additional system of three-shift systems, the Hamiltonian equation includes a new parameter:

\[
    h_2(d) = \alpha(d)h_2'(d) 
\]

as follows:

\[
\begin{align*}
    \alpha(d) &= \begin{cases} 
    2 & \text{weekend} \\
    1 & \text{weekday} 
\end{cases} \\
    h_2'(d) &= \begin{cases} 
    2 & \text{late nighttime} \\
    1.5 & \text{early nighttime} \\
    1 & \text{daytime} 
\end{cases} 
\end{align*}
\]

To introduce a healthy work environment, the new Hamiltonian term where the priority will be \( g(n, d) \):

\[
    g(n, d) = \begin{cases} 
    2g & \text{high priority} \\
    1.5g & \text{middle priority} \\
    g & \text{low priority} 
\end{cases} 
\]

By applying this additional constraint to the system, the schedule is more sophisticated by highly considering the day-off request from all nurses to make a better work environment in the hospital.

3) TRAFFIC FLOW OPTIMIZATION
Traffic flow optimization problems can be solved using the objective function equation required to define the variables for the QUBO as the classical preprocessing. It can evaluate the performance of the QUBO form using qbsolv algorithms. In the first step, it has to preprocess the map and GPS dataset \( T \). Then, the traffic congestion area is identified \( CR \) for each car \( c \) with their commute route \( MSP \). Each car in the dataset will find the possible commute route \( APR \). It assigns two alternative routes \( TPR \). After that, it can formulate the problems as QUBO to minimize the congestion in road segments overlapping. Next, among the route assignments, it gets the best solution that reduces congestion in the entire traffic graph. Additionally, if it cannot find the best route for each car, it can repeat step two to the end over again to identify there is no traffic congestion. In this way, it can optimize the total congestion by assigning each car route to the segment. Neukart et al. [67] expressed using pseudo-code of the Algorithms 5 as follows:

```
Algorithm 5 Traffic Flow Optimization

INPUT T: Traffic Congestion Dataset
OUTPUT R: Optimal Route for each Car c
1: for each car c ∈ T do
2:    CR = current_routes(c, T)
3:    end for
4: for each car c’s ∈ CR do
5:    MSD = map_source_destination(c, CR)
6:    end for
7: for each source and destination do
8:    APR = all_possible_routes(c, MSD)
9:    TPR = two_possible_routes(c, APR)
10: end for
11: for each car c ∈ source and destination do
12:    R = route(c, TPR)
13: end for
```

V. PERFORMANCE EVALUATION
This section describes the outstanding optimization formulation and the performance comparison of QA approaches for each COP as mentioned above and the SA approaches. We also discuss the improvement methods to help the large-scale problem in terms of performance results and embed them directly into the D-Wave quantum annealer.

A. OPTIMIZATION FORMULATION
This section describes the outperform optimization formulation in solving COPs namely the k-concurrent approach, qbsolv algorithms, and reverse annealing methods.
1) K-CONCURRENT APPROACH
Among the category of graph partitioning and clustering problems, we observed that the k-concurrent approach is the ideal approach for improvement solutions. As stated in the GP problem [48], k-concurrent improves the quality of results. Similarly, detecting multiple community problems [53] using QA with the k-concurrent approach can obtain the community structure instantaneously. This approach uses the concept of super-nodes to improve the quality of partitioning over the existing tools by partitioning random graphs and community structures. By doing this, it does not have to do the recursion of the problem itself and can render a highly optimized community structure. For the small graph structure, the k-concurrent approach can embed directly on QPU. Meanwhile, it required hybrid classical-quantum qbsolv algorithms for the large graph structure. Additionally, the quality of the results can be determined by the limited number of available qubits and the sparse connectivity of the current system.

2) QBSOLV ALGORITHMS
Neukart et al. [67] optimized the traffic flow using QA on D-Wave QPU. Due to the nature of the real-world size and its own complexity, they used qbsolv algorithms to solve the congestion on the road. They expected to generate a random route for each car on the segment to reduce traffic congestion. The results show that using qbsolv algorithms can reduce congestion on the road compared to the original routes. The algorithm by Neukart et al. [67] aimed to only reduce the original route congestion with a limited number of routes and cars, with no communication to the infrastructure and other traffic participants. By dedicated D-Wave QPU to solve this kind of real-world problem, it results in more suitable optimization of the problems. In the future, Neukart et al. [67] hoped that their algorithms can leverage the performance to embed the larger problem directly on the D-Wave QPU.

3) REVERSE ANNEALING
The application of the NSP [34] has satisfied all the constraints in the QUBO form to performance on D-Wave 2000Q quantum annealer. We observed successful results with reverse annealing of fixed sample size and annealing time to match all constraints in NSP, but not uniformly. This shows that there will be a better method for solving real-world problems application with new various fields. Currently, Ike et al. [34] are struggling to find a realistic number of nurses and working days for the problems. However, there is a way in which the larger problem can be solved by embedding the problem into a subproblem. Even though it confirmed that reverse annealing improved the solution for the ground state, it depends on the variable and parameter of the problem. However, it cannot deny that SA seems to find the solution of NSP quite nicely compared to QA. Due to the noise of the processor affecting the probability of finding the ground state, Ike et al. [34] promised to further apply this method to different variety of scheduling problems.

B. PERFORMANCE COMPARISON
Table 1 presents the problem and solution comparison using the real quantum machine of the D-Wave quantum annealer with different category of graph partitioning and clustering, factorization, prediction, and other well-known problems in terms of the problem, optimization formulation, and performance comparison with D-Wave hardware. For each problem, there are several optimization formulations that can be solved. We can see that some formulation produces considerably outstanding results, while others are unaffected in producing a result. For the category of graph partitioning and clustering problems, we observed that the k-concurrent approach improves the quality of the community structure within the annealing time all at once. Furthermore, we observed that finite-set MPC using both QA can be more achievable compared to the SA method. Also, the proposed method of CQFS performs a better result in both QA and SA methods with comparable time. For other well-known problems, we observed that NSP using reverse annealing does improve the probability of success to all constraints. Additionally, the traffic flow optimization problem with qbsolv algorithms effectively minimizes the traffic congestion on the original route. However, some other formulations, as shown in the table, can eventually solve the problem but are not the significant leading optimization formulation. It notices that each problem used various optimization formulations on different D-Wave hardware to produce performance comparison results.

Algorithm 6 Embedding Algorithms

INPUT H: Hardware Graph
OUTPUT LV: Logical Variable

1: while unused qubit ∃ H do
2:     LV = determine(LV)
3:     SP = shortest_path(LV)
4:     if multiple assignment! → SP then
5:         qubit = LV
6:         LV = SP
7:     else
8:         LV = drop(LV)
9:     end if
10: end while

Even though we observed an improvement on a few methods for a specific problem, there are so many large problems of COPs that cannot be embedded directly into the D-Wave quantum annealer. To improve the solution of embedding large-scale problems with the limited available number of qubits into the D-Wave system, it has to partition the large problem into the subproblem using quantum annealer with qbsolv [62], [75], [76]. There are problems in obtaining the solutions while embedding the large problem since the subproblem is continuously updated. However, it uses greedy algorithms to find the local minimum and then embeds them into the Chimera graph of the D-Wave 2000Q to evaluate their
Table 1. Problem and solution comparison using the real quantum machine.

| Category                        | Problem                          | Optimization Formulation                  | Performance Comparison                                                                 | D-Wave Hardware                        |
|---------------------------------|----------------------------------|------------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------|
| **Graph Partitioning and Clustering Problems** |                                 |                                          |                                                                                        |                                        |
|                                  | Graph Partitioning [48]          | Graph Partitioning with Thresholding     | Thresholding did not change the community results.                                     | D-Wave 2X Runtime: 600s                 |
|                                  |                                  | Graph Partitioning for Walshaw           | Walshaw archived equally or outperform results.                                         |                                        |
|                                  |                                  | K-concurrent Graph Partitioning         | K-concurrent improved the quality of community structure.                               |                                        |
|                                  | Community Detection [51]         | Adiabatic Quantum Computation (AQC)      | Both quantum methods were capable to achieved the results.                               | D-Wave QA Runtime: N/A                  |
|                                  |                                  | Quantum Approximate Optimization Algorithms (QAOA) |                                                                                        |                                        |
|                                  | Detecting Multiple Communities [53] | K-concurrent Community Detection     | K-concurrent quantified community structure within the annealing time, instantaneously. | D-Wave 2X D-Wave 2000Q Runtime: 20μs    |
|                                  |                                  | Extracting k-core                      |                                                                                        |                                        |
|                                  |                                  | Graph Partitioning                     |                                                                                        |                                        |
|                                  |                                  | Vertex Splitting                       |                                                                                        |                                        |
|                                  |                                  | Combined Methods                       |                                                                                        |                                        |
|                                  | Core-Periphery Partitioning [57] | Core-Periphery Partitioning             | Core-periphery partitioning used D-Wave annealer could handle QUBO.                     | D-Wave Advantage 4.1 Run-time: N/A     |
| **Factorization Problems**       | Prime Factorization [59]         | Direct Method                           | MMTM guaranteed the value of prime factorization.                                      | D-Wave 2000Q Runtime: N/A              |
|                                  | Modified Multiplication Table Method |                                  |                                                                                        |                                        |
|                                  | Matrix Factorization [62]        | Least Square Algorithm                 | Both algorithms obtained the solution in polynomial time.                               | D-Wave 2X Runtime: N/A                 |
|                                  |                                  | Nonnegative/ Binary Matrix Factorization |                                                                                        |                                        |
| **Prediction Problems**          | Finite-Set Model Predictive Control [7] | Exact Solution             | Finite-Set MPC problems used quantum annealing to produces a solution closer to the exact solution. | D-Wave 2000Q Runtime: 0.79s             |
|                                  |                                  | Simulated Annealing (SA)                |                                                                                        |                                        |
|                                  |                                  | Quantum Annealing (QA)                  |                                                                                        |                                        |
|                                  | Feature Selection [37]           | ItemKNN                                 | As compared with ItemKNN, TFIDF, and CFeCBF, CQFS shows better diversity and accuracy which can be solved using either quantum or classical solvers. | D-Wave Leap Hybrid V2 D-Wave Advantage Runtime: Various |
|                                  |                                  | TFIDF                                   |                                                                                        |                                        |
|                                  |                                  | CFeCBF                                  |                                                                                        |                                        |
|                                  |                                  | CQFS QA                                 |                                                                                        |                                        |
|                                  |                                  | CQFS SA                                 |                                                                                        |                                        |
| **Other Well-Known Problems**   | Support Vector Machine [66]      | SVM measured by AUROC                 | SVMs can be solved directly by QUBOs formulation on a quantum annealer.                 | D-Wave 2000Q Runtime: N/A              |
|                                  |                                  | SVM measured by AUPRC                  |                                                                                        |                                        |
|                                  |                                  | SVM measured by Accuracy                |                                                                                        |                                        |
|                                  | Nurse Scheduling Problem [34]    | Forward Annealing                      | Reverse annealing did improved the probability of success.                              | D-Wave 2000Q Runtime: 20μs             |
|                                  | Traffic Flow Optimization [67]   | Reverse Annealing                      |                                                                                        |                                        |
|                                  |                                  | Qbsolv Algorithms                      | Qbsolv algorithms minimized the traffic congestion on the original route.               | D-Wave 2X Runtime: 22μs                |

Performance. It uses conventional local search algorithms to obtain better solutions. Additionally, it uses a complete-graph embedding algorithm for embedding the subproblem into the D-Wave quantum annealer to improve accuracy. Thus, it is not efficient to use so many resources and algorithms to perform such a large problem. Okada et al. [77] proposed new embedding algorithms for efficiently embedding larger subproblems with better solutions into D-Wave 2000Q, as shown in Algorithm 6. These embedding algorithms required input as any hardware graph of the subproblem to produce an output of its logical variable \( LV \). Although there is an existence of the unused qubit in the subproblem, it determines the logical variable \( LV \) and calculates the shortest path of the logical variable \( LV \). If the multiple assignments of the logical variable \( LV \) are not necessary then qubits associated with the root are reserved by the logical variable \( LV \). Thus, the logical variable \( LV \) is assigned to the qubits in the shortest paths \( SP \). Otherwise, it drops the logical variable \( LV \) from the subproblem shown below.

Okada et al. [77] showed improvement results in terms of the solutions by embedding larger subproblems. The embedding algorithms have a drastically reduced computation time. Additionally, the implementation of D-Wave 2000Q has been proposed as a reverse annealing method to search for the initial state and promises to produce a better local search. Okada et al. [77] proposed to combine this new method of reverse annealing with the newly embedded algorithms to improve the solution by embedding a larger subproblem into the D-Wave quantum annealer.

VI. CONCLUSION AND DISCUSSION
In this survey, we explained several studies using quantum annealers, which are completely different from existing computers, to solve COP on a practical scale. Each study constructed QUBO or Ising model for COP to use a D-Wave quantum annealer. Since it is not easy to directly run with the actual scale of COP due to the hardware limitation of the current D-Wave machine, various decomposition or
size reduction methods for making the COP problem into a form that can be executed on the D-Wave system were also described. Additionally, several studies have suggested algorithms to improve the performance of the output from D-Wave. Thus, it showed the potential power of solving various COPs and performance improvement in the future generation of D-Wave.

However, it is expected that much efforts is required to solve various COPs. This survey provides basic guidelines for effectively solving various COPs in quantum annealers by analyzing several studies, but it may not be applicable to specific COPs. Many studies that use the classical computer to solve COP are also organized in the form of “problem-algorithm-results”. This is to improve the performance through customization to obtain better performance than previous studies that solved the same COP. However, these studies produced very little scientific knowledge. Therefore, according to the COP characteristics, it is essential to make a standardized guideline for making QUBO or Ising model operate on quantum annealer. Additionally, to effectively solve the COP, it is necessary to consider the hardware characteristics. Until now, QUBO or Ising model was created considering the topology of the current available D-Wave version. Although the current hardware limitations restrict the size of problems that can be solved, we believe that this will be overcome in the near future as a shred of solid evidence. Ohzeki et al. [21] propose a new approach to solve a large-scale optimization problem in statistical mechanics called the Hubbard-Stratonovich transformation. This approach investigated traffic flow optimization problems in Sendai and Kyoto cities in Japan, and it yields better results. Li et al. [78] also suggest that nested quantum annealing correction (NQAC) can be used to solve the machine learning models, which eventually produces a better performance overall with longer anneal time and interpretation of the results. Peng et al. [46] introduce a novel method of surrogate lagrangian relaxation (SLR) to solve the unit commitment (UC) problem, which overcomes the need to determine the more significant number of the binary variable with the quantum resources' limitations to improve the scalability of the results. Raisuddin [79] declares the new hybrid technique of unified formulation (FEqa) to formulate the finite element problem on the classical computer. This technique is analogous to the aforementioned methods. Then forward the rest of the work to the quantum annealer to solve the problem more effectively and efficiently. In the future, there will be many types of quantum annealers and each of them will have its own hardware characteristics. Through various studies, transforming COP into the most appropriate form depending on the quantum annealer should be done to quickly derive the optimal value. Current QA technology continues to evolve to solve complex problems in the NISQ era. Various vendors, such as Hitachi, Fujitsu, and D-Wave systems, are making their own quantum annealers and developing technologies to solve problems much faster and more effectively. For instance, for D-Wave Advantage 2 of the D-Wave system, it announced that it would boast a QPU containing more than 7,000 qubits and 20-way qubit connectivity by developing a new topology, Zephyr. This is the number of qubits, which is 1800 more than Advantage, the latest D-Wave quantum annealer, and 5-way qubit connectivity is more achieved. This is expected to find the optimal solution faster than the previous machines [67], [68], [69].

REFERENCES

[1] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity. North Chelmsford, MA, USA: Courier Corporation, 1998.
[2] W. Cook, L. Lovász, and P. D. Seymour, Combinatorial Optimization: Papers From the DIMACS Special Year, vol. 20. American Mathematical Soc., 1995.
[3] Y. Bengio, A. Lodi, and A. Provost, “Machine learning for combinatorial optimization: A methodological tour d’horizon,” Eur. J. Oper. Res., vol. 290, no. 2, pp. 405–421, 2021.
[4] N. Vesselino, R. Steinert, D. F. Perez-Ramirez, and M. Boman, “Learning combinatorial optimization on graphs: A survey with applications to networking,” IEEE Access, vol. 8, pp. 120388–120416, 2020.
[5] S. V. Yakovlev and O. S. Pichugina, “Properties of combinatorial optimization problems over polyhedral-spherical sets,” Cybern. Syst. Anal., vol. 54, no. 1, pp. 99–109, Feb. 2018.
[6] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, “Perspectives of quantum annealing: Methods and implementations,” Rep. Prog. Phys., vol. 83, no. 5, May 2020, Art. no. 054401.
[7] D. Inoue and H. Yoshida, “Model predictive control for finite input systems using the D-wave quantum annealer,” Sci. Rep., vol. 10, no. 1, pp. 1–10, Dec. 2020.
[8] E. K. Grant and T. S. Humble, Adiabatic Quantum Computing and Quantum Annealing. Oxford Research Encyclopedia of Physics, 2020.
[9] V. Guanavan, M. Sobhana, G. Amardha, P. Yellamma, O. R. Devi, K. B. Prakash, and J. Naren, “Quantum inspired meta-heuristic approach for optimization of genetic algorithm,” Comput. Electr. Eng., vol. 94, Sep. 2021, Art. no. 107356.
[10] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Maurer, and C. Linhoff-Poopen, “A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer,” Frontiers ICT, vol. 6, p. 13, Jun. 2019.
[11] S. Muthuraman and V. P. Venkatesan, “A comprehensive study on hybrid meta-heuristic approaches used for solving combinatorial optimization problems,” in Proc. World Congr. Comput. Commun. Technol. (WCCCT), Feb. 2017, pp. 185–190.
[12] M. Borowski, P. Gora, K. Karnas, M. Blajda, K. Król, A. Matyjasek, D. Bunczyk, M. Szewczyk, and M. Kutwin, “New hybrid quantum annealing algorithms for solving vehicle routing problem,” in Proc. Int. Conf. Comput. Sci. Springer, 2020, pp. 546–561.
[13] H. M. Waidyasonoyiya and M. Hariyama, “Temporal and spatial parallel processing of simulated quantum annealing on a multicore CPU,” J. Supercomput., vol. 78, no. 6, pp. 8733–8750, Apr. 2022.
[14] R. Hamefyl, T. Inagaki, P. L. McMahon, D. Ventrelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, and K. Enbutus, “Experimental investigation of performance differences between coherent ising machines and a quantum annealer,” Sci. Adv., vol. 5, no. 5, May 2019, Art. no. eaau0823.
[15] T. Sato, M. Ohzeki, and K. Tanaka, “Assessment of image generation by quantum annealer,” Sci. Rep., vol. 11, no. 1, pp. 1–10, Dec. 2021.
[16] H. N. Dijidev, G. Chapuis, G. Hahn, and G. Rizk, “Efficient combinatorial optimization using quantum annealing,” 2018, arXiv:1801.08653.
[17] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, “Quantum annealing for industry applications: Introduction and review,” 2021, arXiv:2112.07401.
[18] F. Hu, L. Lamata, M. Sanz, X. Chen, X. Chen, C. Wang, and E. Solano, “Quantum computing cryptography: Finding cryptographic Boolean functions with quantum annealing by a 2000 qubit D-wave quantum computer,” Phys. Lett. A, vol. 384, no. 10, Apr. 2020, Art. no. 126214.
[19] M. Jayaseelan, S. K. Manipandian, A. N. Jordan, and N. P. Bigelow, “Quantum measurement arrow of time and fluctuation relations for measuring spin of ultracold atoms,” Nature Commun., vol. 12, no. 1, pp. 1–7, Dec. 2021.
Khan and Versaggi, "Quantum Computing Experimentation With Amazon Braket: Explore Amazon Braket Quantum Computing to Solve Combinatorial Optimization Problems."

M. Ohzeki, "Breaking limitation of quantum annealer in solving optimization problems under constraints," Sci. Rep., vol. 10, no. 1, pp. 1–12, Dec. 2020.

X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio, "Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers," IEEE Trans. Quantum Eng., vol. 3, pp. 1–20, 2022.

N. Mohseni, P. Mcmahon, and T. Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," Nature Phys., vol. 4, no. 6, pp. 363–379, Jun. 2022.

H. Irie, H. Liang, T. Doi, S. Gongyo, and T. Hatsuuda, "Hybrid quantum annealing via molecular dynamics," Sci. Rep., vol. 11, no. 1, pp. 1–9, Dec. 2021.

J. Hatzy, "Solving combinatorial optimization problems on a d-wave quantum annealer and development of higher level hardware abstraction layers," Ph.D. dissertation, Technische Hochschule Köln, 2021.

M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and Y. Katograber, "Physics-inspired optimization for quadratic unconstrained problems using a digital annealer," Frontiers Phys., vol. 7, p. 48, Apr. 2019.

T. Stollenwerk, B. O’Gorman, D. Venturelli, S. Mandra, O. Rodionova, H. Ng, B. Sridhar, E. G. Rieffel, and R. Biswas, "Quantum annealing applied to de-conflicting optimal trajectories for air traffic management," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 1, pp. 285–297, Jan. 2020.

O. Pakin, "Targeting classical code to a quantum annealer," in Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst., Apr. 2019, pp. 529–543.

T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "2*0.6 A 2x30k-spin multipich scalable annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2019, pp. 52–54.

K. H. Imanmotokoro, S. Imanfard, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, and M. Motomura, "Statica: A 512-spin 0.25 m-weight annealing processor with an all-spin-updates-at-one-architecture for combinatorial optimization with complete spin–spin interactions," IEEE J. Solid-State Circuits, vol. 56, no. 1, pp. 165–178, 2020.

E. Cohen, A. Senderovich, and J. C. Beck, "An ising framework for constraint-based clustering on special purpose hardware," in Proc. Int. Conf. Integ. Constraint Program., Artif. Intell., Oper. Res. Springer, 2020, pp. 130–147.

S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto, and H. Tamura, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC), 2020, pp. 667–672.

A. Maruo, H. Igarashi, H. Oshima, and S. Shimokawa, "Optimization of planar magnet array using digital annealer," IEEE Trans. Magn., vol. 56, no. 3, pp. 1–4, Mar. 2020.

K. Ikeda, Y. Nakamura, and T. S. Humble, "Application of quantum annealer to nurse scheduling problem," Sci. Rep., vol. 9, no. 1, pp. 1–10, Dec. 2019.

Y.-L. Fang and P. A. Warburton, "Minimizing minor embedding energy: no application in quantum annealing," Quantum Inf. Process., vol. 19, no. 7, pp. 1–29, Jul. 2020.

R. Ayazadeh, M. Halem, and T. Finin, "Reinforcement quantum annealing with D-Wave 2000Q," 2018, arXiv:1810.07765.

H. Irie, H. Liang, T. Doi, S. Gongyo, and T. Hatsuuda, "Hybrid quantum annealing via molecular dynamics," Sci. Rep., vol. 11, no. 1, pp. 1–9, Dec. 2021.

S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, and I. Safro, "Multilevel combinatorial optimization across quantum architectures," ACM Trans. Quantum Comput., vol. 2, no. 1, pp. 1–29, Apr. 2021.

Y. Alexeev, M. A. Bragin, and Y. Zhou, "Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation," IEEE Trans. Power Syst., early access, Jun. 10, 2022, doi: 10.1109/TPWRS.2022.3181221.

F. Peng, P. Zhang, M. A. Bragin, and Y. Zhou, "Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation," IEEE Trans. Power Syst., early access, Jun. 10, 2022, doi: 10.1109/TPWRS.2022.3181221.

M. A. Bragin, Y. Zhou, and F. Peng, "Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation," IEEE Trans. Power Syst., early access, Jun. 10, 2022, doi: 10.1109/TPWRS.2022.3181221.

O. Moreira, M. Popp, and C. Schulz, "Graph partitioning with acyclicity constraints," 2017, arXiv:1704.00705.

H. Ushijima-Mwesigwa, C. F. A. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the D-Wave system," in Proc. 2nd Int. Workshop Post Moores Era Supercomputing, Nov. 2017, pp. 22–29.

S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, "Quantum annealing on the D-Wave 2000Q," 2018, arXiv:1810.07765.

L. Peel, D. B. Larremore, and A. Clauset, "A ground truth about metadata and community detection in networks," Sci. Adv., vol. 3, no. 5, May 2017, Art. no. e1602548.

C. F. A. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski, "Detecting multiple communities using quantum annealing on the D-Wave system," Phys. Rev. C, vol. 95, no. 3, Sep. 2017, doi: 10.1109/TPWRS.2022.3181221.

E. Pelofske, G. Hahn, and H. Dijdev, "Optimizing the spin reversal transform on the D-wave 2000Q," in Proc. IEEE Int. Conf. Rebooting Comput. (ICRC), Nov. 2019, pp. 1–8.

E. Pelofske, G. Hahn, and H. Dijdev, "Solving large maximum clique problems on a quantum annealer," in Proc. Int. Workshop Quantum Technol. Optim. Problems. Springer, 2019, pp. 123–135.

G. Chuprains, H. Dijdev, and J. I. Zitnik, "Finding maximum cliques on the D-Wave quantum annealer," J. Signal Process. Syst., vol. 91, nos. 3–4, pp. 363–377, Mar. 2019.

C. F. Higham, D. J. Higham, and F. Tudisco, "Testing a QUBO formulation of core-periphery partitioning on a quantum annealer," 2022, arXiv:2201.01543.

R. J. Gallagher, J.-G. Young, and B. F. Welles, "A clarified typology of core-periphery structure in networks," Sci. Adv., vol. 7, no. 12, Mar. 2021, Art. no. eabc9800.

S. Jiang, K. A. Brit, A. J. McCaskey, T. S. Humble, and S. Kais, "Quantum annealing for prime factorization," Sci. Rep., vol. 8, no. 1, pp. 1–9, 2018.

R. Dridi and H. Alghassi, "Prime factorization using quantum annealing on D-Wave," in Proc. IEEE Int. Conf. Rebooting Comput. (ICRC), Nov. 2019, pp. 1–8.

R. J. Gallagher, J. G. Young, and B. F. Welles, "A clarified typology of core-periphery structure in networks," Sci. Adv., vol. 7, no. 12, Mar. 2021, Art. no. eabc9800.
[64] A. Delibasic, G. Cavallaro, M. Willsch, F. Melgani, M. Riedel, and K. Michielsen, “Quantum support vector machine algorithms for remote sensing data classification,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2021, pp. 2608–2611.

[65] G. Cavallaro, D. Willsch, M. Willsch, K. Michielsen, and M. Riedel, “Approaching remote sensing image classification with ensembles of support vector machines on the D-Wave quantum annealer,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Sep. 2020, pp. 1973–1976.

[66] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, “Support vector machines on the D-wave quantum annealer,” Comput. Phys. Commun., vol. 248, Mar. 2020, Art. no. 107006.

[67] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney, “Traffic flow optimization using a quantum annealer,” Frontiers ICT, vol. 4, p. 29, Dec. 2017.

[68] R. Shaydulin, H. Ushijima-Mwesigwa, C. F. A. Negre, I. Safro, S. M. Minisiewski, and Y. Alexeev, “A hybrid approach for solving optimization problems on small quantum computers,” Computer, vol. 52, no. 6, pp. 18–26, Jun. 2019.

[69] S. Akbar and S. K. Saritha, “Towards quantum computing based community detection,” Comput. Sci. Rev., vol. 38, Nov. 2020, Art. no. 100313.

[70] T. Yoshimura, T. Shirai, M. Tawada, and N. Togawa, “QUBO matrix distorting method for consumer applications,” in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Jan. 2022, pp. 01–06.

[71] E. J. Crosson and D. A. Lidar, “Prospects for quantum enhancement with diabatic quantum annealing,” Nature Rev. Phys., vol. 3, no. 7, pp. 466–489, Jul. 2021.

[72] J. Golden and D. O’Malley, “Reverse annealing for nonnegative/binary matrix factorization,” PLoS ONE, vol. 16, no. 1, Jan. 2021, Art. no. e0244026.

[73] D. Venturelli and A. Kondratyev, “Reverse quantum annealing approach to portfolio optimization problems,” Quantum Mach. Intell., vol. 1, nos. 1–2, pp. 17–30, May 2019.

[74] G. Passarelli, K.-W. Yip, D. A. Lidar, H. Nishimori, and P. Lucignano, “Reverse quantum annealing of the p-spin model with relaxation,” Phys. Rev. A, Gen. Phys., vol. 101, no. 2, Feb. 2020, Art. no. 022331.

[75] A. Teplukhin, B. K. Kendrick, S. Tretiak, and P. A. Dub, “Electronic structure with direct diagonalization on a D-wave quantum annealer,” Sci. Rep., vol. 10, no. 1, pp. 1–11, Dec. 2020.

[76] Y. Ding, X. Chen, L. Lamata, E. Solano, and M. Sanz, “Implementation of a hybrid classical-quantum annealing algorithm for logistic network design,” 2019, arXiv:1906.10074.

[77] S. Okada, M. Ohzeki, M. Terabe, and S. Taguchi, “Improving solutions by embedding larger subproblems in a D-Wave quantum annealer,” Sci. Rep., vol. 9, no. 1, pp. 1–10, Dec. 2019.

[78] R. Y. Li, T. Albash, and D. A. Lidar, “Limitations of error corrected quantum annealing in improving the performance of Boltzmann machines,” Quantum Sci. Technol., vol. 5, no. 4, Aug. 2020, Art. no. 045010.

[79] O. M. Rausudden and S. De, “FEqa: Finite element computations on quantum annealers,” Comput. Methods Appl. Mech. Eng., vol. 395, May 2022, Art. no. 115014.