Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 13, Number 1, pages 46–53, January 2024

Various QUBO Formulations of the Graph Isomorphism Problem and Related Problems

Daisuke Takafuji
Shunan University
843-4-2 Gakuendai,
Shunan City, 745-8566, JAPAN

Koji Nakano and Yasuaki Ito
Hiroshima University
1-4-1 Kagamiyama,
Higashi Hiroshima City, 739-8527, JAPAN

Takashi Yazane, Junko Yano, Takumi Kato, Shiro Ozaki, Rie Mori and Ryota Katsuki
Research and Development Headquarters, NTT DATA Group Corporation
Toyosu Center Bldg, Annex, 3-9, Toyosu 3-chome, Koto-ku, Tokyo 135-8671, Japan

***Abstract*—Quadratic Unconstrained Binary Optimization (QUBO) is a combinatorial optimization to find an optimal binary solution vector that minimizes the energy value defined by a quadratic formula of binary variables in the vector. The graph isomorphism problem is one of the applications of combinatorial optimization, and is not known to be solvable in polynomial time. The problem is also not known to be NP-complete. We propose four QUBO formulations for graph isomorphism problem instances. We also propose four QUBO formulations for induced subgraph isomorphism problem instances and two QUBO formulations for subgraph isomorphism problem instances, which are NP-complete problems. We solve QUBO instances defined by our QUBO formulations, using known QUBO solvers: Gurobi optimizer, Fixstars Amplify AE and OpenJij with SA.**

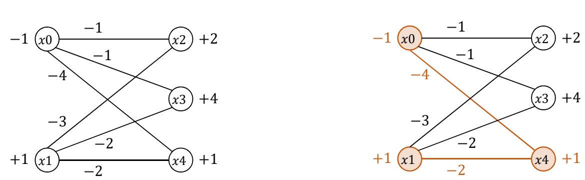
***Index Terms*—Quantum computing, combinatorial optimization, graph isomorphism, QUBO solvers**

I. Introduction

A *Quadratic Unconstrained Binary Optimization* (QUBO) problem is defined by an upper triangular matrix $W = (W_{i,j})$ $(0 \leq i \leq j \leq n-1)$ of size $n \times n$. A QUBO problem $W = (W_{i,j})$ aims to find a binary vector $X = (x_i)$ $(x_i \in \{0, 1\}$ for all i $(0 \leq i \leq n-1))$ that minimizes the energy defined as the sum of quadratic terms of X as follows:

$$E(X) = \sum_{0 \leq i \leq j \leq n-1} W_{i,j} x_i x_j + C,$$

where C is a constant called *offset*. It is known that many NP-hard problems can be reduced to QUBO problems [1]. A QUBO problem can be considered a minimum subgraph problem [2]. For a given n-bit QUBO problem $W = (W_{i,j})$, we consider a weighted undirected graph with vertices $0, 1, \ldots, n-1$ and vertices i and j are connected with an edge (i, j) if $W_{i,j} \neq 0$. We assume that every vertex i and every edge (i, j) take weights $W_{i,i}$ and $W_{i,j}$, respectively. In other words, $W = (W_{i,j})$ is the weight matrix of the graph. For simplicity, we assume weights $W_{i,j}$'s are integers. For an n-bit vector $X = (x_i)$, we consider that each vertex i is selected as a subgraph vertex if $x_i = 1$. Figure 1 shows an example of a weighted undirected graph. The minimum subgraph consists of vertices $x0$, $x1$, and $x4$, and the total weight is $-1+1+1+(-4)+(-2) = -5$, which corresponds to an optimal solution $X = [1, 1, 0, 0, 1]$ of the QUBO problem.



(a) Weighted undirected graph (b) Minimum subgraph

Fig. 1. The weighted undirected graph corresponding to a QUBO problem and the minimum subgraph corresponding to an optimal QUBO solution.

In this paper, we propose QUBO formulations to solve three problem: the graph isomorphism problem, the induced subgraph isomorphism problem, and the subgraph isomorphism problem. Our idea of QUBO formulations is to archive constraints by giving rewards and penalties. Also, we solve QUBO instances defined by our QUBO formulations, using known QUBO solvers: Gurobi optimizer, Fixstars Amplify AE and OpenJij with SA.

First, we introduce the graph isomorphism problem. Two graphs $H_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $H_2 = (\mathcal{V}_2, \mathcal{E}_2)$ are *isomorphic* if and only if there is a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$ such that there is an edge in H_1 between two vertices $u, v \in \mathcal{V}_1$ if and only if there is an edge in H_2 between two vertices $f(u), f(v) \in \mathcal{V}_2$. That is, $(u, v) \in \mathcal{E}_1 \iff (f(u), f(v)) \in \mathcal{E}_2$ holds for any vertices $u, v \in \mathcal{E}_1$. We can also define that one graph is isomorphic to another graph.

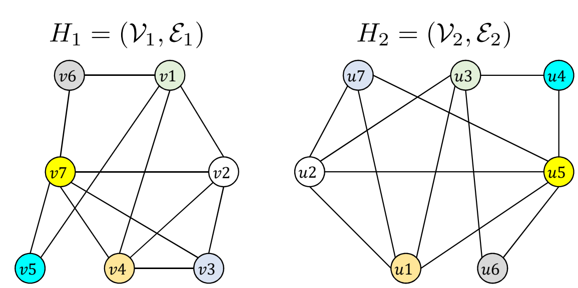


Fig. 2. Two graphs H_1 and H_2. These graphs are isomorphic, because there is a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$ shown in Table I.

Figure 2 illustrates two graphs H_1, H_2, which are isomorphic, because there is a bijection $f : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ shown in Table I. As shown in Table I, each vertex of H_1 can correspond to the vertex of H_2 with the same color. Then, $(u, v) \in \mathcal{E}_1 \iff (f(u), f(v)) \in \mathcal{E}_2$ holds for any $u, v \in \mathcal{E}_1$. For example, both $(v1, v2) \in \mathcal{E}_1$ and $(f(v1), f(v2)) \in \mathcal{E}_2$ hold because of $f(v1) = u3$ and $f(v2) = u2$. Also, both $(v4, v5) \notin \mathcal{E}_1$ and $(f(v4), f(v5)) \notin \mathcal{E}_2$ hold because of $f(v4) = u1$ and $f(v5) = u4$.

TABLE I
A BIJECTION $f : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ FOR TWO GRAPHS H_1 AND H_2 ILLUSTRATED IN FIG. 2.

$v \in \mathcal{V}_1$	$f(v) \in \mathcal{V}_2$	$v \in \mathcal{V}_1$	$f(v) \in \mathcal{V}_2$	$v \in \mathcal{V}_1$	$f(v) \in \mathcal{V}_2$
$v1$	$u3$	$v4$	$u1$	$v6$	$u6$
$v2$	$u2$	$v5$	$u4$	$v7$	$u5$
$v3$	$u7$				

The graph isomorphism problem is a decision problem, defined as follows:

Graph isomorphism problem

Instance: Two graphs.

Output: Determine whether two graphs are isomorphic.

The graph isomorphism problem is not known to be solvable in polynomial time nor to be NP-complete [3]. It is known that the graph isomorphism problem is in class NP.

Next, we pay attention to an induced subgraph of a graph. Let H be a graph, and $\mathcal{V}'$ be a subset of vertices of H. *The induced subgraph* of H and $\mathcal{V}'$ is defined by the graph, consisting of $\mathcal{V}'$ and the set of all edges whose endvertices are in $\mathcal{V}'$. The induced subgraph isomorphism problem is a decision problem, defined as follows:

Induced subgraph isomorphism problem

Instance: A guest graph and a host graph.

Output: Determine whether the host graph has a vertex set such that an induced subgraph of the host graph and the vertex set is isomorphic to the guest graph.

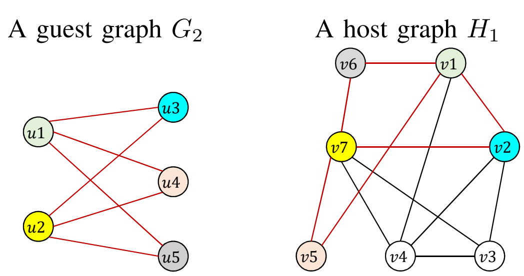


Fig. 3. The induced subgraph of a host graph H_1 and $\{v1, v2, v5, v6, v7\}$ is isomorphic to a guest graph G_2, because each vertex of G_2 can correspond to the vertex of H_1 with the same color.

Figure 3 illustrates an induced subgraph isomorphism problem instance. We focus on an induced subgraph of H_1 and $\{v1, v2, v5, v6, v7\}$. The induced subgraph has an edge set $\{(v1, v2), (v1, v5), (v1, v6), (v2, v7), (v5, v7), (v6, v7)\}$, because of the definition of an induced subgraph. From Figure 3, we can see that the induced subgraph is isomorphic to the graph G_2. Indeed, there is a bijection $f : \{u1, u2, u3, u4, u5\} \rightarrow \{v1, v2, v5, v6, v7\}$ such that $f(u1) = v1$, $f(u2) = v7$, $f(u3) = v2$, $f(u4) = v5$ and $f(u5) = v6$. Then, G_2 has the edge $(u1, u3)$ and $\mathcal{H}_1$ has the edge $(f(u1), f(u3))$, because of $f(u1) = v1$ and $f(u3) = v2$. Also, G_2 does not have the edge $(u4, u5)$ and $\mathcal{H}_1$ does not have the edge $(f(u4), f(u5))$, because of $f(u4) = v5$ and $f(u5) = v6$. If the induced subgraph isomorphism problem instance consists of a guest graph G_2 and a host graph H_1, the optimal solution is "YES". It is known that the induced subgraph isomorphism problem is NP-complete [4].

Finally, we introduce the subgraph isomorphism problem. We consider a subgraph of a graph, consisting of subsets of both vertices and edges of the graph. The subgraph isomorphism problem is also a decision problem, defined as follows:

Subgraph isomorphism problem

Instance: A guest graph and a host graph.

Output: Determine whether the host graph includes a subgraph, isomorphic to the guest graph. Note that a subgraph of the host graph consists of a subset of vertices and a subset of edges, in which endvertices in the subset of vertices of the host graph.

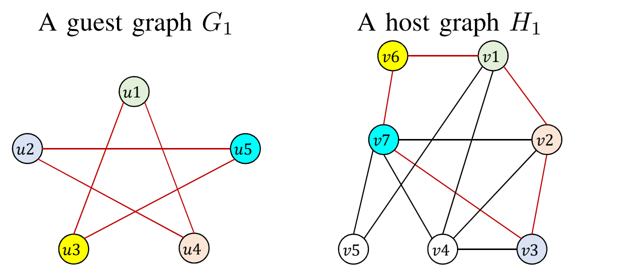


Fig. 4. The subgraph of H_1 is isomorphic to the graph G_1, because each vertex of G_1 can correspond to the vertex with the same color of H_1 and each edge of G_1 can correspond to the red edge of H_1.

Figure 4 illustrates a subgraph isomorphism problem instance. We define a subgraph of H_1 by a vertex set $\{v1, v2, v3, v6, v7\}$ and an edge set $\{(v1, v2), (v2, v3), (v3, v7), (v6, v7), (v1, v6)\}$. From Figure 4, we can see that the subgraph is isomorphic to the graph G_1. Clearly, there is a bijection $f : \{u1, u2, u3, u4, u5\} \rightarrow \{v1, v2, v3, v6, v7\}$ such that $f(u1) = v1$, $f(u2) = v3$, $f(u3) = v6$, $f(u4) = v2$ and $f(u5) = v7$. For example, G_1 has the edge $(u1, u3)$ and the subgraph of $\mathcal{H}_1$ has the edge $(f(u1), f(u3))$, because of $f(u1) = v1$ and $f(u3) = v6$. Also, G_1 does not have the edge $(u4, u5)$ and the subgraph of $\mathcal{H}_1$ does not have the edge $(f(u4), f(u5))$, because of $f(u4) = v2$ and $f(u5) = v7$. We note that the subgraph of $\mathcal{H}_1$ does not have the edge $(f(u4), f(u5))$, though $\mathcal{H}_1$ has the edge $(f(u4), f(u5))$. If the subgraph isomorphism problem instance consists of a guest graph G_1 and a host graph H_1, the optimal solution is "YES". It is known that the subgraph isomorphism problem is NP-complete [4].

For the graph isomorphism problem, the induced subgraph isomorphism problem, and the subgraph isomorphism problem, QUBO formulations are proposed in [5], [6]. For the induced subgraph isomorphism problem, efficient QUBO formulations are proposed in [7]. Moreover, QUBO formulations for many optimization problems are proposed, and references are shown in [8].

II. QUBO Formulations for Graph Isomorphism Problem

We introduce the graph isomorphism problem again.

Graph isomorphism problem
Instance: Two graphs $H_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $H_2 = (\mathcal{V}_2, \mathcal{E}_2)$.
Output: Determine whether H_1 and H_2 are isomorphic, that is, there is a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$ such that $(u, v) \in \mathcal{E}_1 \iff (f(u), f(v)) \in \mathcal{E}_2$ for any $u, v \in \mathcal{V}_1$.

We propose four QUBO formulations for the graph isomorphism problem. We write V_1 and V_2 for the number of vertices of the graph H_1 and H_2, respectively. We define QUBO vector X of size $V_1 \times V_2$ by $X = (x_{0,0},\ x_{0,1},\ \ldots,\ x_{0,V_2-1}, x_{1,0},\ x_{1,1},\ \ldots,\ x_{1,V_2-1}, \ldots,\ x_{V_1-1,0},\ x_{V_1-1,1}, \ldots, x_{V_1-1,V_2-1})$. The condition $x_{i,j} = 1$ signifies that the function f maps a vertex v_i in $\mathcal{V}_1$ to a vertex v_j in $\mathcal{V}_2$, that is, $f(v_i) = v_j$. To define one-hot encoding, we use the following $B(X)$:

$$B(X) = \sum_{j=0}^{V_2-1} (\sum_{i=0}^{V_1-1} x_{i,j} - 1)^2 + \sum_{i=0}^{V_1-1} x_{i,j} (\sum_{j=0}^{V_2-1} x_{i,j} - 1)$$

We describe our idea of QUBO formulations as follows. To define our QUBO formulations, we give rewards to propositions to be satisfied and give penalties to propositions not to be satisfied. From the definition of the graph isomorphism problem, two graphs H_1 and H_2 are isomorphic if and only if there is a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$ such that $(u, v) \in \mathcal{E}_1 \iff (f(u), f(v)) \in \mathcal{E}_2$ for any $u, v \in \mathcal{E}_1$.

From the proposition that $(u, v) \in \mathcal{E}_1 \iff (f(u), f(v)) \in \mathcal{E}_2$, the propositions to be satisfied are as follows: $(u, v) \in \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \in \mathcal{E}_2$, and $(u, v) \notin \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_2$.

The propositions not to be satisfied are as follows: $(u, v) \in \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_2$, and $(u, v) \notin \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \in \mathcal{E}_2$.

First, we select the proposition to be satisfied as follows: $(u, v) \in \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \in \mathcal{E}_2$. Providing rewards for the constraints of the proposition to be satisfied, we define the energy function shown in Eq. (1), called by QUBO formulation A.

$$E(X) = -\sum_{(i,j)\in\mathcal{E}_1} \sum_{(i',j')\in\mathcal{E}_2} x_{i,i'} x_{j,j'} + B(X) \qquad (1)$$

Next, we select the proposition not to be satisfied as follows: $(u, v) \notin \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \in \mathcal{E}_2$. Providing penalties for the constraints of the proposition not to be satisfied, we define the energy function shown in Eq. (2), called by QUBO formulation B.

$$E(X) = +\sum_{(i,j)\notin\mathcal{E}_1} \sum_{(i',j')\in\mathcal{E}_2} x_{i,i'} x_{j,j'} + B(X) \qquad (2)$$

We select the proposition not to be satisfied as follows: $(u, v) \in \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_2$. Providing rewards for the constraints of the proposition not to be satisfied, we define the energy function shown in Eq. (3), called by QUBO formulation C.

$$E(X) = +\sum_{(i,j)\in\mathcal{E}_1} \sum_{(i',j')\notin\mathcal{E}_2} x_{i,i'} x_{j,j'} + B(X) \qquad (3)$$

Finally, we select the proposition to be satisfied as follows: $(u, v) \notin \mathcal{E}_1 \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_2$, Providing rewards for the constraints of the proposition to be satisfied, we define the energy function shown in Eq. (4), called by QUBO formulation D.

$$E(X) = -\sum_{(i,j)\notin\mathcal{E}_1} \sum_{(i',j')\notin\mathcal{E}_2} x_{i,i'} x_{j,j'} + B(X) \qquad (4)$$

We discuss our QUBO formulations. We write E_1 and E_2 for the number of edges of the graph H_1 and H_2, respectively. Let $A_1 = \frac{1}{2}V_1(V_1 - 1)$ and $A_2 = \frac{1}{2}V_2(V_2 - 1)$. In this paper, we suppose that two graphs H_1, H_2 including in a graph isomorphism problem instance satisfies $V_1 = V_2$ and $E_1 = E_2$. If $V_1 \neq V_2$ or $E_1 \neq E_2$, there is not a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$, that is, two graphs H_1, H_2 are not isomorphic. When an instance consists of two graphs H_1, H_2 with $V_1 \neq V_2$ or $E_1 \neq E_2$, the optimal solution for the graph isomorphism problem is clearly "No". Hence, we suppose that a graph isomorphism problem instance consists of two graphs H_1, H_2 with both $V = V_1 = V_2$ and $E = E_1 = E_2$. We note that $A = A_G = A_H$.

TABLE II
Optimal solutions and the number of non-zero entries in QUBO matrices, of our QUBO formulations for the graph isomorphism problem instances.

	Optimum	♯ of non-zero entries in QUBO matrices
QUBO Form. A	$-E$	$(V-1)V^2 + 2E^2$
QUBO Form. B	0	$(V-1)V^2 + 2E(A-E)$
QUBO Form. C	0	$(V-1)V^2 + 2E(A-E)$
QUBO Form. D	$-(A-E)$	$(V-1)V^2 + 2(A-E)^2$

Table II shows the optimum, that is, the minimum energy, of our QUBO formulations. We can see that the minimum energy of QUBO Formulations A and D depend on the size of an input graph. We can represent our QUBO formulation $E(X)$ in the form of a QUBO matrix. Hence, Table II shows the number of non-zero entries in the QUBO matrices corresponding to instances for the graph isomorphism problem. From Table II, we can obtain the following remark.

Remark 1: If $E < \frac{A}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation A is the smallest among our QUBO formulations. If $E > \frac{A}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation D is the smallest among our QUBO formulations. If $E = \frac{A}{2}$, the number of non-zero entries in the QUBO matrix of our QUBO formulations are equal. □

III. QUBO Formulations for Induced Subgraph Isomorphism Problem

We show the induced subgraph isomorphism problem again.

Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 13, Number 1, pages 46–53, January 2024

Induced subgraph isomorphism problem
Instance: A guest graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ and a host graph $H = (\mathcal{V}_H, \mathcal{E}_H)$.
Output: Determine whether H has a vertex set such that an induced subgraph of H and the vertex set is isomorphic to G. In other words, determine whether there is an injection $f : \mathcal{V}_G \to \mathcal{V}_H$ such that $(u, v) \in \mathcal{E}_G \iff (f(u), f(v)) \in \mathcal{E}_H$ for any $u, v \in \mathcal{V}_G$.

We propose four QUBO formulations for the induced subgraph isomorphism problem. We write V_G and V_H for the number of vertices of the graph G and H, respectively. We define QUBO vector X of size $V_G \times V_H$ by $X = (x_{0,0}, x_{0,1}, \ldots, x_{0,V_G-1}, x_{1,0}, x_{1,1}, \ldots, x_{1,V_G-1}, \ldots, x_{V_H-1,0}, x_{V_H-1,1}, \ldots, x_{V_H-1,V_G-1})$. The condition $x_{i,j} = 1$ signifies that the function f maps a vertex v_i in $\mathcal{V}_G$ to a vertex v_j in $\mathcal{V}_H$, that is, $f(v_i) = v_j$. To achieve one-hot encoding, we use the following $S(X)$:

$$S(X) = \sum_{j=0}^{V_G-1} (\sum_{i=0}^{V_H-1} x_{i,j} - 1)^2 + \sum_{i=0}^{V_H-1} x_{i,j} (\sum_{j=0}^{V_G-1} x_{i,j} - 1),$$

We describe our idea of QUBO formulations as follows. To define our QUBO formulations, we give rewards to propositions to be satisfied and give penalties to propositions not to be satisfied. From the definition of the induced subgraph isomorphism problem, the induced subgraph of H and a vertex set of H is isomorphic to G if and only if there is an injection $f : \mathcal{V}_G \to \mathcal{V}_H$ such that $(u, v) \in \mathcal{E}_G \iff (f(u), f(v)) \in \mathcal{E}_H$ for any $u, v \in \mathcal{V}_G$.

From the proposition that $(u, v) \in \mathcal{E}_G \iff (f(u), f(v)) \in \mathcal{E}_H$, the propositions to be satisfied are as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$, and $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$. The propositions not to be satisfied are as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$, and $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$.

First, we select two propositions to be satisfied as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$ and $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$. Providing rewards the constraints of the propositions to be satisfied, we define the energy function shown in Eq. (5), called by QUBO formulation A.

$$E(X) = - \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} \sum_{(i',j')\in\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} - \sum_{(i,j)\notin\mathcal{E}_\mathcal{G}} \sum_{(i',j')\notin\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + S(X) \qquad (5)$$

Next, we select two propositions not to be satisfied as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$ and $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$. Providing penalties for the constraints of the propositions not to be satisfied, we define the energy function shown in Eq. (6), called by QUBO formulation B.

$$E(X) = + \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} \sum_{(i',j')\notin\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + \sum_{(i,j)\notin\mathcal{E}_\mathcal{G}} \sum_{(i',j')\in\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + S(X) \qquad (6)$$

We select the propositions to be satisfied as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$. We also select the propositions not to be satisfied as follows: $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \in \mathcal{E}_H$. We provide rewards for the constraints of the proposition to be satisfied. We provide penalties for the constraints of the propositions not to be satisfied. As a result, we define the energy function shown in Eq. (7), called by QUBO formulation C.

$$E(X) = - \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} \sum_{(i',j')\in\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + \sum_{(i,j)\notin\mathcal{E}_\mathcal{G}} \sum_{(i',j')\in\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + S(X) \qquad (7)$$

Finally, We select the propositions to be satisfied as follows: $(u, v) \notin \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$. We also select the propositions not to be satisfied as follows: $(u, v) \in \mathcal{E}_G \implies (f(u), f(v)) \notin \mathcal{E}_H$. We provide rewards for the constraints of the proposition to be satisfied. We provide penalties for the constraints of the propositions not to be satisfied. As a result, we define the energy function shown in Eq. (8), called by QUBO formulation D.

$$E(X) = - \sum_{(i,j)\notin\mathcal{E}_\mathcal{G}} \sum_{(i',j')\notin\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + \sum_{(i,j)\in\mathcal{E}_\mathcal{G}} \sum_{(i',j')\notin\mathcal{E}_\mathcal{H}} x_{i,i'} x_{j,j'} + S(X) \qquad (8)$$

We discuss our QUBO formulations of the induced subgraph isomorphism problem instances. We write E_G and E_H for the number of edges of the graph G and H, respectively. Let $A_G = \frac{1}{2} V_G(V_G - 1)$ and $A_H = \frac{1}{2} V_H(V_H - 1)$. Table III shows the optimum, that is, the minimum energy, of QUBO instances generated by our QUBO formulations. We can see that the minimum energy of instances by QUBO Formulations A, C, and D depend on the size of a guest graph. We can represent our QUBO formulation $E(X)$ in the form of a QUBO matrix. Hence, Table III also shows the number of non-zero entries in QUBO matrices. From Table III, we can obtain the following remark.

Remark 2: If $E_H < \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation C is the smallest among our four QUBO formulations. If $E_H > \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation D is the smallest among our four QUBO formulations. If $E_H = \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of our four QUBO formulations are equal. □

IV. QUBO Formulations for Subgraph Isomorphism Problem

We introduce the subgraph isomorphism problem again.
Subgraph isomorphism problem
Instance: A guest graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ and a host graph $H = (\mathcal{V}_H, \mathcal{E}_H)$.
Output: Determine whether there is a subgraph of H which is isomorphic to G, where a subgraph of H consists of subsets of both vertices and edges of H. In other words, determine where

TABLE III
OPTIMAL SOLUTIONS AND THE NUMBER OF NON-ZERO ENTRIES IN QUBO MATRICES, OF OUR QUBO FORMULATIONS FOR THE INDUCED SUBGRAPH ISOMORPHISM PROBLEM INSTANCES.

	Optimum	$\sharp$ of non-zero entries in QUBO matrices
QUBO Form. A	$-A_G$	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2A_G(A_H - E_H) - 2E_G(A_H - 2E_H)$
QUBO Form. B	0	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2A_G E_H + 2E_G(A_H - 2E_H)$
QUBO Form. C	$-E_G$	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2A_G E_H$
QUBO Form. D	$-(A_G - E_G)$	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2A_G(A_H - E_H)$

there is an injection $f : \mathcal{V}_G \to \mathcal{V}_H$ such that $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \in \mathcal{E}_H$ for any $u, v \in \mathcal{V}_G$.

We propose two QUBO formulations for the subgraph isomorphism problem. We use the same notations as Section III, and we write them again. Let V_G and V_H be the number of vertices of the graph G and H, respectively. We define QUBO vector X of size $V_G \times V_H$ by $X = (x_{0,0}, x_{0,1}, \ldots, x_{0,V_G-1}, x_{1,0}, x_{1,1}, \ldots, x_{1,V_G-1}, \ldots, x_{V_H-1,0}, x_{V_H-1,1}, \ldots, x_{V_H-1,V_G-1})$. The condition $x_{i,j} = 1$ signifies that the function f maps a vertex v_i in $\mathcal{V}_G$ to a vertex v_j in $\mathcal{V}_H$, that is, $f(v_i) = v_j$. As well as the induced subgraph isomorphism problem, we use the following $S(X)$ to archive one-hot encoding:

$$S(X) = \sum_{j=0}^{V_G-1} (\sum_{i=0}^{V_H-1} x_{i,j} - 1)^2 + \sum_{i=0}^{V_H-1} x_{i,j} (\sum_{j=0}^{V_G-1} x_{i,j} - 1)$$

We describe our idea of QUBO formulations as follows. To define our QUBO formulations, we give rewards to propositions to be satisfied and give penalties to propositions not to be satisfied. From the definition of the subgraph isomorphism problem, the subgraph of H is isomorphic to G if and only if there is an injection $f : \mathcal{V}_G \to \mathcal{V}_H$ such that $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \in \mathcal{E}_H$ for any $u, v \in \mathcal{V}_G$. Clearly, the proposition to be satisfied is $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \in \mathcal{E}_H$. The proposition not to be satisfied is $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_H$.

First, we select the proposition to be satisfied as follows: $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \in \mathcal{E}_H$. Providing rewards for the constraints of the proposition to be satisfied, we define the energy function shown in Eq. (9), called by QUBO formulation A.

$$E(X) = -\sum_{(i,j)\in\mathcal{E_G}} \sum_{(i',j')\in\mathcal{E_H}} x_{i,i'} x_{j,j'} + S(X) \qquad (9)$$

Next, we select the proposition not to be satisfied as follows: $(u,v) \in \mathcal{E}_G \Longrightarrow (f(u), f(v)) \notin \mathcal{E}_H$. Providing penalties for the constraints of the proposition not to be satisfied, we define the energy function shown in Eq. (10), called by QUBO formulation B.

$$E(X) = +\sum_{(i,j)\in\mathcal{E_G}} \sum_{(i',j')\notin\mathcal{E_H}} x_{i,i'} x_{j,j'} + S(X) \qquad (10)$$

TABLE IV
OPTIMAL SOLUTIONS AND THE NUMBER OF NON-ZERO ENTRIES IN QUBO MATRICES, OF OUR QUBO FORMULATIONS FOR THE SUBGRAPH ISOMORPHISM PROBLEM INSTANCES.

	Optimum	$\sharp$ of non-zero entries in QUBO matrices
QUBO Form. A	$-E_G$	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2E_G E_H$
QUBO Form. B	0	$\frac{1}{2}(V_G + H_G - 2)V_G V_H$ $+2E_G(A_H - E_H)$

We discuss our QUBO formulations for the subgraph isomorphism problem instances. Let E_G and E_H be the number of edges of the graph G and H, respectively. Let $A_G = \frac{1}{2}V_G(V_G - 1)$ and $A_H = \frac{1}{2}V_H(V_H - 1)$. Table IV shows the optimum, that is, the minimum energy, of QUBO instances generated by our QUBO formulations. We can see that the minimum energy of instances by QUBO Formulation A depends on the number of edges of a guest graph. Also, Table IV shows the number of non-zero entries in QUBO matrices for the subgraph isomorphism problem. From Table IV, we can obtain the following remark.

Remark 3: If $E_H < \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation A is smaller than that of QUBO Formulation B. If $E_H > \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation B is smaller than that of QUBO Formulation A. If $E_H = \frac{A_H}{2}$, the number of non-zero entries in the QUBO matrix of QUBO Formulation B is equal to that of QUBO Formulation A. □

V. EXPERIMENT RESULTS

We solve QUBO instances defined by our formulations using QUBO solvers: Gurobi optimizer, Fixstars Amplify AE, and OpenJij with SA. Gurobi optimizer [9] is a commercial Mixed Integer Programming (MIP) solver supporting quadratic objectives, which can solve QUBO problems. Fixstars Amplify AE [10] is a Cloud Platform for Quantum Annealing using GPUs. OpenJij [11] is a heuristic optimization library for the Ising model and QUBO. Both Gurobi and OpenJij run on an Intel Xeon Platinum 8358 CPU with 32 physical cores.

For the three problems in this paper, we generate problem instances so as to all optimal solutions are "YES". Every problem instance consists of regular graphs. A graph is regular if any vertices of the graph have the same number of edges. A δ-regular graph means that each vertex of the graph has δ edges. To generate regular graphs, we use Network Analysis in Python [12], called NetworkX [13].

We terminate QUBO solvers when the QUBO solver finds the optimal solution or the computing time is over 100 seconds. We run Gurobi once per instance, and we can obtain one solution. On the other hand, we run Amplify AE and OpenJij 10 times per instance. Hence, we can obtain 10 solutions each, and we use average solutions.

A. Graph Isomorphism Problem

We describe how to generate graph isomorphism problem instances. We generate a random regular graph, and generate

Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 13, Number 1, pages 46–53, January 2024

TABLE V
THE NUMBER OF NON-ZERO ENTRIES IN $W_{\langle i,j \rangle, \langle k,l \rangle}$ AND SOLUTIONS OF QUBO INSTANCES FROM GRAPH ISOMORPHISM PROBLEM INSTANCES, OBTAINED BY QUBO SOLVERS.

Instance	Two 90-node 22-regular graphs								Two 90-node 68-regular graphs							
# of non-zero	QUBO Form. A		QUBO Form. B		QUBO Form. C		QUBO Form. D		QUBO Form. A		QUBO Form. B		QUBO Form. C		QUBO Form. D	
entries	**2,689,200**		6,698,700		6,698,700		18,909,450		19,456,200		6,512,400		6,512,400		**2,515,050**	
	energy		energy		energy		energy		energy		energy		energy		energy	
Optimum	-990		0		0		-3015		-3060		0		0		-945	
Gurobi	$\mathbf{-990}$		59		58		-2662		-2703		58		58		$\mathbf{-945}$	
	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.	Ave. energy	# of Opt.
Amplify AE	$\mathbf{-990}$	(10)	**0**	(10)	**0**	(10)	-2291.2	(0)	-2543.4	(0)	**0**	(10)	**0**	(10)	$\mathbf{-945}$	(10)
OpenJij	-302.2	(0)	59.2	(0)	59.2	(0)	-1433.7	(0)	-1510.9	(0)	59.4	(0)	58.8	(0)	-283.2	(0)

another regular graph by shuffling vertices of the regular graph. As a result, the two regular graphs are isomorphic. The optimal solution for the instance consisting of the two regular graphs is "Yes". Hence, the optimal solution of QUBO for the instance is equal to the optimal energy. We generate two instances: one consists of two 90-node 22-regular graphs with 990 edges each, and the other consists of two 90-node 68-regular graphs with 3060 edges each. Clearly, each QUBO vector is 8000 bits.

Table V shows the number of non-zero entries in QUBO matrices defined by our four QUBO formulations of the graph isomorphism problem. Table V also shows average solutions obtained by Amplify AE and OpenJij. Because Gurobi runs once per instance, Table V shows one solution per instance for Gurobi. In Table V, the number in the parenthesis represents the number of optimal solutions obtained by Amplify AE or OpenJij. When the number of non-zero entries in the QUBO matrix is large, Gurobi, Amplify AE, and OpenJij cannot find the optimum within 100 seconds. We note that OpenJij cannot find it even if the number of non-zero entries in the QUBO matrix is small.

B. Induced Subgraph Isomorphism Problem and Subgraph Isomorphism Problem

We describe how to generate instances consisting of a guest graph and a host graph such that a guest graph, say G, is a V_G-node δ_g-regular graph and a host graph, say H, is V_H-node δ_h-regular graph. Using the Python package NetworkX, we generate a random V_G-node δ_g-regular graph G and another random V_H-node δ_h-regular graph K. We select V_G vertices from the graph K randomly, called the *core* vertices set.

When we generate an induced subgraph isomorphism problem instance, we reshape the graph K to a δ_h-regular graph by exchanging edges such that an induced subgraph of the new graph and the core vertices set is isomorphic to G. We can regard the new δ_h-regular graph as a host graph H in the instance of the problem. As a result, there is a vertex set such that an induced subgraph of H and the vertex set is isomorphic to the guest graph. The optimal solution for the problem instance is "Yes".

On the other hand, when we generate a subgraph isomorphism problem instance, we reshape the graph K to a δ_h-regular graph by exchanging edges such that a subgraph of the core vertices set is isomorphic to G. The subgraph is defined by the core vertices set and an edge set, in which endvertices are included in the core vertices set. We can regard the new δ_h-regular graph as a host graph H in the instance of the problem. As a result, a guest graph is isomorphic to the subgraph of a host graph. The optimal solution for the problem instance is "Yes". Hence, the optimal solution of QUBO for the instance is equal to the optimal energy.

For our experiment, we generate 8-instances with $V_G = 8, 64$ and $V_G \times V_H = 8192$. Hence, $V_H = 1024, 128$ holds. We also set $\delta_g = \frac{3}{8}V_G$ and $\delta_h = \frac{3}{8}V_H, \frac{5}{8}V_H$.

Table VI shows the number of non-zero entries in QUBO matrices defined by our four QUBO formulations of the induced subgraph isomorphism problem. Table VI also shows average solutions obtained by Amplify AE and OpenJij. Because Gurobi runs once per instance, Table VI shows one solution per instance for Gurobi. In addition, Table VI shows the number of optimal solutions, obtained by Amplify AE or OpenJij, which are represented as the number in the parenthesis. We note that a host graph H contains E_H edges, and let $A_H = \frac{1}{2}V_H(V_H - 1)$. As we can see from Remark 2, if a host graph H is a 1024-node 384-regular graph or a 128-node 48-regular graph, then $E_H < \frac{A_H}{2}$ holds and the number of non-zero entries in the QUBO matrix of QUBO Formulation C is the smallest. If H is a 1024-node 640-regular graph or a 128-node 80-regular graph, then $E_H > \frac{A_H}{2}$ holds and the number of non-zero entries in the QUBO matrix of QUBO Formulation D is the smallest. Table VI shows that Remark 2 is satisfied.

From Table VI, Amplify AE can find the optimum energy within 100 seconds when the number of non-zero entries in the QUBO matrix is the smallest among four QUBO formulations. Regardless of the number of non-zero entries in the QUBO matrix Gurobi can find the optimum energy within 100 seconds when a guest graph G of instances is an 8-node 3-regular graph. However, Gurobi and OpenJij cannot find them when a guest graph G of instances is an 64-node 24-regular graph.

Table VII shows the number of non-zero entries in QUBO matrices defined by our two QUBO formulations of the subgraph isomorphism problem. Table VII also shows average solutions obtained by Amplify AE and OpenJij. Because Gurobi runs once per instance, Table VII shows one solution per

Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 13, Number 1, pages 46–53, January 2024

TABLE VI
THE NUMBER OF NON-ZERO ENTRIES IN $W_{\langle i,j\rangle,\langle k,l\rangle}$ AND SOLUTIONS OF QUBO INSTANCES FROM INDUCED SUBGRAPH ISOMORPHISM PROBLEM INSTANCES, OBTAINED BY QUBO SOLVERS.

Instance	A guest graph G is an 8-node 3-regular graph							
	A host graph H is a 1024-node 384-regular graph				A Host graph H is a 1024-node 640-regular graph			
# of non-zero entries	QUBO Form. A	QUBO Form. B	QUBO Form. C	QUBO Form. D	QUBO Form. A	QUBO Form. B	QUBO Form. C	QUBO Form. D
	19,415,040	18,370,560	**15,237,120**	22,548,480	18,366,464	19,419,136	22,577,152	**15,208,448**
	energy	energy	energy	energy	energy	energy	energy	energy
Optimum	−28	0	−12	−16	−28	0	−12	−16
Gurobi	**−28**	**0**	**−12**	**−16**	**−28**	**0**	**−12**	**−16**
	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)
Amplify AE	**−28** (10)	**0** (10)	**−12** (10)	**−16** (10)	**−28** (10)	**0** (10)	**−12** (10)	**−16** (10)
OpenJij	−18.2 (0)	**0** (10)	−7.9 (0)	−10.7 (0)	−17.4 (0)	**0** (10)	−8 (0)	−9.8 (0)

Instance	A guest graph G is a 64-node 24-regular graph							
	A host graph H is a 128-node 48-regular graph				A host graph H is a 128-node 80-regular graph			
# of non-zero entries	QUBO Form. A	QUBO Form. B	QUBO Form. C	QUBO Form. D	QUBO Form. A	QUBO Form. B	QUBO Form. C	QUBO Form. D
	18,124,800	16,220,160	**13,172,736**	21,172,224	16,158,720	18,186,240	21,430,272	**12,914,688**
	energy	energy	energy	energy	energy	energy	energy	energy
Optimum	−2016	0	−768	−1248	−2016	0	−768	−1248
Gurobi	−1537	48	−296	−767	−1474	48	−247	−716
	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)
Amplify AE	−1419.3 (0)	**0** (10)	**−768** (10)	−966.3 (4)	−1646.3 (4)	**0** (10)	**−768** (10)	**−1248** (10)
OpenJij	−1088.2 (0)	47.3 (0)	−193.8 (0)	−423.4 (0)	−977.6 (0)	48.8 (0)	−177.3 (0)	−352.4 (0)

TABLE VII
THE NUMBER OF NON-ZERO ENTRIES IN $W_{\langle i,j\rangle,\langle k,l\rangle}$ AND SOLUTIONS OF QUBO INSTANCES FROM SUBGRAPH ISOMORPHISM PROBLEM INSTANCES, OBTAINED BY QUBO SOLVERS.

Instance	G: an 8-node 3-regular graph				G: a 64-node 24-regular graph			
	H: a 1024-node 384-regular graph		H: a 1024-node 640-regular graph		H: a 128-node 48-regular graph		H: a 128-node 80-regular graph	
# of non-zero entries	QUBO Form. A	QUBO Form. B	QUBO Form. A	QUBO Form. B	QUBO Form. A	QUBO Form. B	QUBO Form. A	QUBO Form. B
	8,945,664	12,079,104	12,091,392	**8,933,376**	**5,505,024**	8,552,448	8,650,752	**5,406,720**
	energy	energy	energy	energy	energy	energy	energy	energy
Optimum	−12	0	−12	0	−768	0	−768	0
Gurobi	**−12**	**0**	**−12**	**0**	−587	36	−708	21
	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)	Ave. energy (# of Opt.)
Amplify AE	**−12** (10)	**0** (10)	**−12** (10)	**0** (10)	−596.2 (0)	9.5 (7)	−684.8 (0)	16.9 (0)
OpenJij	−7.4 (0)	**0** (10)	−10.6 (2)	**0** (10)	−331.2 (0)	37.2 (0)	−499.6 (0)	22.4 (0)

instance for Gurobi. In addition, Table VII shows the number of optimal solutions, obtained by Amplify AE or OpenJij, which are represented as the number in the parenthesis. As we can see from Remark 3, if a host graph H is a 1024-node 384-regular graph or a 128-node 48-regular graph, then $E_H < \frac{A_H}{2}$ holds and the number of non-zero entries in the QUBO matrix of QUBO Formulation A is smaller than that of QUBO Formulation B. If H is a 1024-node 640-regular graph or a 128-node 80-regular graph, then $E_H > \frac{A_H}{2}$ holds and the number of non-zero entries in the QUBO matrix of QUBO Formulation B is smaller than that of QUBO Formulation A. Table VII shows this fact.

From Tables VII, when a guest graph G is an 8-node 3-regular graph, Gurobi and Amplify AE can find the optimal solutions within 100 seconds even if the number of non-zero entries in the QUBO matrix is large. Though OpenJij cannot find the optimal solutions of two QUBO instances generated by QUBO Formulation A, OpenJij can find them of two QUBO instances of QUBO Formulation B. When a guest graph G is a 64-node 24-regular graph, no QUBO solver can find the optimal solutions within 100 seconds even if the number of non-zero entries in the QUBO matrix is small.

VI. CONCLUSION

We have proposed various QUBO formulations for solving the graph isomorphism problem, the induced subgraph isomorphism problem, and the subgraph isomorphism problem. We have shown the experiment results using three QUBO solvers: Gurobi optimizer, Fixstars Amplify AE, and OpenJij with SA. Solving QUBO problems defined by our QUBO formulations, we can solve the graph isomorphism problem, and the related problems. Experiment results show that we can define various QUBO formulations by giving rewards and penalties to archive constraints.

REFERENCES

[1] F. Glover and G. Kochenberger, “A tutorial on formulating QUBO models,” *CoRR*, 2018.

[2] R. Yasudo, K. Nakano, Y. Ito, Y. Kawamata, R. Katsuki, S. Ozaki, T. Yazane, and K. Hamano, “Graph-theoretic formulation of QUBO for scalable local search on GPUs,” in *Proc. of International Parallel and Distributed Processing Systems Workshops*, 2022, pp. 425–434.

[3] J. Köbler, U. Schöning, and J. Torán, *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhäuser, 1993.
[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979.
[5] C. S. Calude, M. J. Dinneen, and R. Hua, “QUBO formulations for the graph isomorphism problem and related problems,” *Theoretical Computer Science*, vol. 701, pp. 54–69, November 2017.
[6] R. Hua and M. J. Dinneen, “Improved QUBO formulation of the graph isomorphism problem,” *SN Computer Science*, vol. 1, January 2020.
[7] N. Yoshimura, M. Tawada, S. Tanaka, J. Arai, S. Yagi, H. Uchiyama, and N. Togawa, “Mapping induced subgraph isomorphism problems to ising models and its evaluations by an ising machine,” *IEICE Trans. Inf. & Syst.*, pp. 481–489, 2021.
[8] D. Ratke, “List of qubo formulations.” [Online]. Available: https://blog.xa0.de/post/List-of-QUBO-formulations/
[9] Gurobi optimization. [Online]. Available: https://www.gurobi.com/
[10] F. A. Corporation. Effective cloud platform for quantum annealing. [Online]. Available: https://amplify.fixstars.com/en/
[11] Openjij: Framework for the ising model and qubo. [Online]. Available: https://www.openjij.org/
[12] A. Martelli, *Python in a Nutshell*. O’Reilly Media Inc, 2003.
[13] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using networkx,” in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11–15.