ISE
Industrial and
Systems Engineering

The max *k*-cut problem on classical and quantum solvers

RAMIN FAKHIMI[1], HAMIDREZA VALIDI[2], ILLYA V. HICKS[2], TAMÁS TERLAKY[1],
AND LUIS F. ZULUAGA[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA
[2]Department of Computational and Applied Mathematics, Rice University, Houston, TX

ISE Technical Report 21T-007

LEHIGH
UNIVERSITY

Formulations of the max k-cut problem on classical and quantum computers

Ramin Fakhimi*[1], Hamidreza Validi†[2], Illya V. Hicks[2], Tamás Terlaky[1], and Luis F. Zuluaga[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA
[2]Department of Computational and Applied Mathematics, Rice University, Houston, TX

March 28, 2022

Abstract

Recent claims on "solving" combinatorial optimization problems via quantum computers have attracted researchers to work on quantum algorithms. The max k-cut problem is a challenging combinatorial optimization problem with multiple notorious mixed integer linear optimization formulations. In this paper, we revisit the binary quadratic optimization formulation of Carlson and Nemhauser (*Operations Research*, 1966) and provide theoretical and computational comparisons between different mixed integer optimization formulations of the max k-cut problem. While no claim on "quantum advantage" is provided, we introduce quadratic unconstrained binary optimization (QUBO) formulations with tight penalty coefficients. For at least 60% of test instances, the computational experiments show the superiority of the tight models over naïve ones in terms of the quality of feasible solutions in the quantum context. We also design quantum circuits to find feasible solutions using a quantum approximate optimization algorithm (QAOA). A preprocessing procedure is proposed to make instances manageable for quantum computers. In comparison to the existing fixing procedures, our proposed preprocessing algorithm reduces the number of vertices and edges of some existing benchmark instances up to 30% and 25%, respectively. Finally, we conduct an extensive set of experiments on classical and quantum-inspired formulations. Our codes and data are available on GitHub.

Keywords: quantum computing; mixed integer optimization models; the max k-cut problem; QUBO formulation; QAOA

1 Introduction

In 2019, Google officially announced that they achieved "quantum supremacy"; that is, the point at which quantum computers can solve large-scale instances of a problem that cannot be handled on classical computers (Preskill, 2011). They claimed that their 54-qubit Sycamore processor could

*fakhimi@lehigh.edu
†hamidreza.validi@rice.edu

conduct a calculation in 200 seconds while it takes 10,000 years for a powerful supercomputer to calculate it (Arute et al., 2019). In response, IBM shed doubt on their experiments and claimed that the calculation could be done on a classical machine in 2.5 days (Pednault et al., 2019). Recently, Richard Borcherds, a Fields Medalist in 1998, also clarified that "comparing quantum computers with classical ones is useless because it is biased towards quantum computers" in one of his lectures on the theory of numbers (Borcherds, 2021).

Current quantum computers struggle to handle large instances of combinatorial optimization problems due to the limited number of qubits and noise (Preskill, 2018). Guerreschi and Matsuura, 2019 declare that achieving quantum advantage may require several hundreds of qubits. They also show that "solving" small instances of a combinatorial optimization problem (i.e., the max cut problem) on quantum computers takes the same amount of time as solving larger instances of the problem on classical ones. Furthermore, most of quantum algorithms (e.g., quantum approximate optimization algorithm (QAOA) and variational quantum eigensolver (VQE)) provide no optimality guarantee. Although comparing quantum computers with classical ones may not make sense, it is still legitimate to compare classical and quantum-inspired formulations on classical and quantum computers, respectively.

Quantum computing was born in the early 1980s with the work of Benioff (1980) on the quantum mechanical model of computers. Later, Deutsch and Jozsa (1992), Shor (1994), and Grover (1996) proposed quantum algorithms that show quantum speed-up over their classical counterparts. The quantum speed-up comes from two fundamental principles that do not exist in the classical setting: superposition and entanglement. While superposition helps to apply an operation on multiple states at once, entanglement links the states to each other (Preskill, 2018). These fundamental principles open up a new class of problems in computational complexity theory called bounded-error quantum polynomial time (BQP) (e.g., see Nielsen and Chuang, 2011, Section 1.4.5). Quantum computers can solve this class of problems with a bounded error in polynomial time. However, the relation between BQP and NP is not known yet (Nielsen and Chuang, 2011). This casts doubts on the capability of quantum computers in solving NP-hard problems. Given skeptical views on claims about "quantum advantage" in solving NP-hard problems, it is still worth examining the boundary of quantum machines' capabilities in generating feasible solutions for those problems.

The max k-cut problem is among the challenging NP-hard problems (Frieze and Jerrum, 1997; Papadimitriou and Yannakakis, 1991) with multiple notorious optimization models in the literature. To the best of our knowledge, Carlson and Nemhauser (1966) introduced a binary quadratic optimization (BQO) formulation for the max k-cut problem to solve a scheduling problem. They formulated *the min k-partition problem* that is combinatorially equivalent to the max k-cut problem; however, they are different in terms of approximability (Eisenblätter, 2002). Given a graph $G = (V, E)$ with edge weights w and a positive integer number $k \geq 2$, the max k-cut problem seeks to find at most k partitions such that the weights of edges with endpoints in different partitions are maximized. While one can employ existing optimization models to solve the problem on classical solvers, they need to convert classical models to quadratic unconstrained binary optimization (QUBO) for solving them via some quantum algorithms like QAOA.

This paper explores the boundaries of classical and quantum machines for solving the max k-cut problem. We first provide a brief background on the max k-cut problem and quantum computing in Section 2. In Section 3, we provide analytical comparisons between the polytopes of existing classical optimization formulations. In Section 4, we introduce two quadratic unconstrained binary optimization models with tight penalty coefficients. In Section 5, we propose preprocessing algorithms to reduce the size of large-scale instances of the max k-cut problem. A set of computational results is provided in Section 6.

Disclaimer. We provide no computational comparison between quantum and classical solvers. In other words, this paper neither supports nor opposes "quantum advantage". However, we study and assess the performance of different classical and quantum-inspired formulations on classical and quantum machines, respectively.

2 Background

This section provides literature reviews and mathematical backgrounds for (i) solving the max k-cut problem on classical machines and (ii) the QAOA in the quantum context.

2.1 The max k-cut problem on classical machines

The max k-cut problem is a well-known combinatorial optimization problem with a wide range of applications, including but not limited to statistical physics (Barahona et al., 1988; De Simone et al., 1995) and scheduling (Carlson and Nemhauser, 1966). One of the basic optimization formulations of the max k-cut problem is an assignment-based mixed integer linear optimization (A-MILO) model with a small number of variables and constraints. Let $n := |V|$ and $m := |E|$ be the number of vertices and edges of graph $G = (V, E)$, respectively. Furthermore, we define $P := \{1, \dots, k\}$ as the set of partitions. For every vertex $v \in V$ and every partition $j \in P$, binary variable x_{vj} is one if vertex v is assigned to partition j. For every edge $\{u, v\} \in E$, binary variable y_{uv} is one if the endpoints of edge $\{u, v\}$ belong to different partitions.

$$
\begin{aligned}
& \max \quad f(x,y) := \sum_{\{u,v\}\in E} w_{uv} y_{uv} && && \text{(1a)}\\
& \qquad\qquad \text{s.t.} \quad \sum_{j\in P} x_{vj} = 1 && \forall v \in V && \text{(1b)}\\
& \text{(A-MILO)} \qquad x_{uj} - x_{vj} \le y_{uv} && \forall \{u,v\} \in E,\ j \in P && \text{(1c)}\\
& \qquad\qquad\qquad x_{vj} - x_{uj} \le y_{uv} && \forall \{u,v\} \in E,\ j \in P && \text{(1d)}\\
& \qquad\qquad\qquad x_{uj} + x_{vj} + y_{uv} \le 2 && \forall \{u,v\} \in E,\ j \in P && \text{(1e)}\\
& \qquad\qquad\qquad x \in \{0,1\}^{n\times k}, y \in \{0,1\}^{m}. && && \text{(1f)}
\end{aligned}
$$

Here, objective function (1a) maximizes the total weight of cut edges. Constraints (1b) imply that every vertex is assigned to exactly one partition. Constraints (1c) and (1d) imply that if endpoints of an edge belong to different partitions, then it is a cut edge. Constraints (1e) imply that if the endpoints of an edge belong to the same partition, then it cannot be a cut edge. Despite the reasonable size of formulation (1) ($\mathcal{O}(m + kn)$ variables, and $\mathcal{O}(mk)$ constraints and non-zeros), it suffers from a weak continuous relaxation and symmetry issue.

Another classical MILO formulation is a large partition-based MILO (P-MILO) formulation with $\mathcal{O}(n^2)$ variables and $\mathcal{O}\big(\binom{n}{k+1}\big)$ constraints (Chopra and Rao, 1993; Chopra and Rao, 1995). Although the continuous relaxation of this formulation provides a relatively tight upper bound in practice, classical solvers struggle to solve even medium-size instances of the max k-cut problem to optimality. For every pair of vertices $\{u,v\} \in \binom{V}{2}$, we define binary variable z_{uv} as follows: z_{uv} is one if vertices u and v belong to the same partition.

$$
\text{(P-MILO)} \qquad
\begin{aligned}
\max \quad & p(z) \coloneqq \sum_{\{u,v\}\in E} w_{uv}(1 - z_{uv}) && && \text{(2a)}\\
\text{s.t.} \quad & z_{uv} + z_{vw} \le 1 + z_{uw} && && \\
& z_{uw} + z_{uv} \le 1 + z_{vw} && && \\
& z_{vw} + z_{uw} \le 1 + z_{uv} && \forall \{u,v,w\} \subseteq V && \text{(2b)}\\
& \sum_{\{u,v\}\in\binom{Q}{2}} z_{uv} \ge 1 && \forall Q \subseteq V \text{ with } |Q| = k+1 && \text{(2c)}\\
& z \in \{0,1\}^{\binom{n}{2}}. && && \text{(2d)}
\end{aligned}
$$

Here, objective function (2a) maximizes the total weight of cut edges. Constraints (2b) imply that for every set $\{u,v,w\} \subseteq V$, if pairs $\{u,v\}$ and $\{v,w\}$ belong to a partition, then vertices u and w also belong to the same partition. Constraints (2c) imply that vertex set V must be partitioned into at most k partitions. Because of the large number of constraints (2c), one can add them on-the-fly. Chopra and Rao (1995) conducted a polyhedral study on the problem and proposed several facet-defining inequalities in the P-MILO context. They also studied A-MILO and P-MILO formulations for the min k-cut problem and proposed multiple facet-defining inequalities in both formulations (Chopra and Rao, 1993).

Furthermore, G. Wang and Hijazi (2020) propose a reduced P-MILO model that is constructed as follows: (i) graph G is extended to a chordal graph, (ii) all maximal cliques of the chordal graph are found, (iii) binary variables z are created *only* for the edge set of the chordal graph, and (iv) constraints (2b)–(2c) are added *only* for the maximal cliques. The number of variables in their formulation is fewer than or equal to that of the P-MILO's model. They show that their formulation outperforms the P-MILO formulation when the chordalized graph is sparse.

Carlson and Nemhauser (1966) proposed a BQO formulation for the max k-cut problem as follows.

$$
\text{(BQO)} \qquad
\begin{aligned}
\max \quad & g(x) \coloneqq \sum_{\{u,v\}\in E} w_{uv}\Big(1 - \sum_{j\in P} x_{uj}x_{vj}\Big) && && \text{(3a)}\\
\text{s.t.} \quad & \sum_{j\in P} x_{vj} = 1 && \forall v \in V && \text{(3b)}\\
& x \in \{0,1\}^{n\times k}. && && \text{(3c)}
\end{aligned}
$$

Here, objective function (3a) maximizes the number of cut edges, and constraints (3b) imply that each vertex must be assigned to exactly one partition. Carlson and Nemhauser (1966) proved that their binary assignment variables can be relaxed for finding an optimal solution. Since the BQO's objective function is non-convex, one can employ convexification techniques to make it convex. Some commercial solvers (e.g., Gurobi) can handle convex (or even non-convex) binary quadratic optimization models.

Eisenblätter (2002) developed a semidefinite formulation for the max k-cut problem and showed that its continuous relaxation is strong. However, they declare that continuous relaxations of the semidefinite optimization and P-MILO formulations are not comparable. As we focus on mixed integer formulations of the max k-cut problem and provide theoretical and computational comparisons between them, semidefinite optimization and mixed integer semidefinite optimization models are not in the scope of this paper. Interested readers are encouraged to refer to Sotirov (2014), Dam and Sotirov (2016), G. Wang and Hijazi (2020), and Lu and Deng (2021) for more details on semidefinite optimization and mixed integer semidefinite optimization formulations of the max k-cut problem.

2.2 The quantum approximate optimization algorithm (QAOA)

Quantum algorithms can be employed in a wide range of applications from human decision-making in management science (Agrawal and Sharda, 2013) to solving classical optimization problems in operations research (Dunning et al., 2018). Farhi et al. (2014) proposed a quantum approximate optimization algorithm (QAOA) that is one of the most promising quantum algorithms for noisy intermediate-scale quantum devices. The QAOA has the BQP complexity (Farhi and Harrow, 2016) that makes it interesting for NP-hard problems with an unconstrained binary optimization formulation. Lucas (2014) formulated several combinatorial optimization problems as QUBO formulations. Different variants of the QAOA have been applied to the max-cut problem (Farhi et al., 2014; Z. Wang et al., 2020). Basso et al. (2021) demonstrated the superiority of the QAOA (with 11 levels) over classical algorithms for solving the max cut problem on large-girth d-regular graphs when d goes to infinity. Further, Fuchs et al. (2021) directly applied the QAOA algorithm to the max k-cut problem without discussing the unconstrained binary optimization formulation of the problem.

Apolloni et al. (1989) proposed a quantum stochastic optimization method for solving a combinatorial optimization problem in which a solution of the problem is encoded in a ground state of a quantum Hamiltonian. This method is called quantum annealing (Amara et al., 1993; Finnila et al., 1994). The quantum annealing algorithm was, in fact, a quantum-inspired version of the simulated annealing algorithm with a better performance (Kadowaki and Nishimori, 1998). Later, Farhi et al. (2000) and Farhi et al. (2001) proposed the quantum adiabatic algorithm that is an implementation of quantum annealing on a quantum computer. It is also called adiabatic quantum optimization (Reichardt, 2004). The adiabatic quantum computation starts with an initial Hamiltonian that evolves over time. One can construct a Hamiltonian based on the QUBO formulation of the problem. A solution can be extracted from the final Hamiltonian. Aharonov et al. (2007) showed that adiabatic quantum computation is polynomially equivalent to the circuit-based quantum model.

Initially, the QAOA was designed to tackle unconstrained binary optimization problems, particularly the maximum satisfiability problem. The QAOA starts with a quantum state $|s\rangle$, a superposition (combination) of all potential solutions (feasible and infeasible). The algorithm has multiple levels. Each level of the algorithm has two types of unitary operators: (i) phase separation operator (U_P) and (ii) mixing operator (U_M). While the phase separation operator reflects the objective function of the problem on the amplitude of quantum states, the mixing operator is designed to search the feasible space of the problem. Figure 1 illustrates a general scheme of the QAOA with p levels, denoted by QAOA$_p$ (Hadfield, 2018). The last step of the algorithm is measurement: a realization of the quantum state in a classical form.

One can construct a phase separation operator by converting the classical representation of a Boolean objective function into its Hamiltonian. Hadfield (2018, Section 6.3) provided a general framework for the mentioned conversion. Let f and H_f be the objective function of the problem and its corresponding Hamiltonian, respectively. The phase separation operator is defined as $U_P(\gamma) := e^{-i\gamma H_f}$ with $\gamma \in [0, 2\pi]$ and i be the imaginary unit. Let Q be the set of qubits. For every qubit

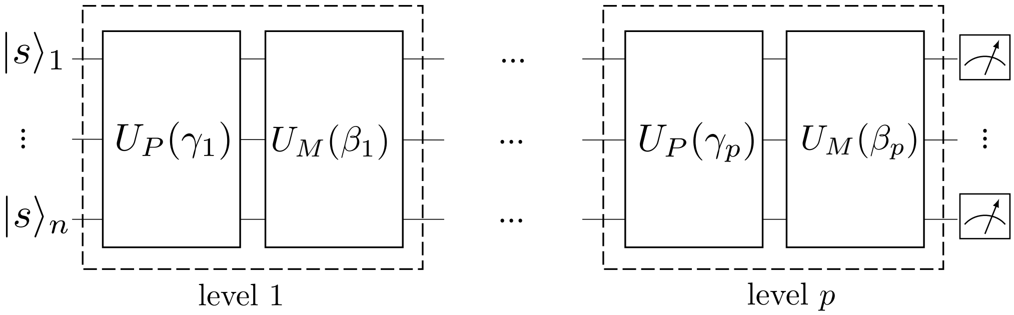


Figure 1: The QAOA$_p$ quantum circuit (see Hadfield, 2018, Figure 5.1).

$j \in Q$, let X_j be the basic Pauli gate X applied to qubit j. Then we define $B \coloneqq \sum_{j\in Q} X_j$. Finally, the mixing operator can be written as $U_M(\beta) \coloneqq e^{-i\beta B}$ with $\beta \in [0, 2\pi]$ and i be the imaginary unit.

Farhi et al. (2014) proved that the QAOA converges to an optimal solution if (i) the number of levels goes to infinity, and (ii) the pair of parameters (γ, β) are globally optimal in the circuit. However, determining these parameters is a challenging task in practice. Researchers usually employ existing derivative-free methods (e.g., COBYLA and Nelder-Mead) for finding parameters γ and β (Z. Wang et al., 2018; Guerreschi and Matsuura, 2019; Zhou et al., 2020). Sack and Serbyn (2021) developed an initialization method to prevent trapping in locally optimal parameters for the max cut problem. Wurtz and Lykov (2021) provided computational evidence on the efficiency of fixed values of parameters for the max cut problem on regular graphs. Medvidović and Carleo (2021) proposed a simulation method for finding optimal parameters γ and β. Harrigan et al. (2021) showed that solving non-hardware-native problems (e.g., the max cut problem on non-planar graphs) via the QAOA is a challenging task as they need "extensive compilation" on real quantum machines.

3 Classical Models: A Theoretical Comparison

In this section, we provide analytical comparisons between BQO and MILO formulations. For analysis purposes, we introduce y variables to the BQO formulation (3): for every edge $\{u, v\} \in E$, y_{uv} equals one if $\{u, v\}$ is a cut edge.

$$y_{uv} = 1 - \sum_{j\in P} x_{uj}x_{vj} \qquad \forall \{u, v\} \in E. \tag{4}$$

Furthermore, we define the polytope of the BQO formulation as follows.

$$\mathcal{R}_{\text{BQO}} \coloneqq \left\{ (x, y) \in [0, 1]^{n\times k} \times \mathbb{R}^m \;\middle|\; (x, y) \text{ satisfies constraints (3b) and (4)} \right\}.$$

The following remark shows that we do not need to impose 0-1 bounds on variables y.

Remark 1. *Constraints $y \in [0, 1]^m$ are implied by the* BQO *formulation* (3) *and constraints* (4).

Proof. Consider a point $(\hat{x}, \hat{y}) \in \mathcal{R}_{\text{BQO}}$. For every edge $\{u, v\} \in E$, we have

$$\hat{y}_{uv} = 1 - \sum_{j\in P} \hat{x}_{uj}\hat{x}_{vj} \geq 1 - \sum_{j\in P} \hat{x}_{uj} = 1 - 1 = 0.$$

Here, the first equality holds by constraints (4). The inequality holds because for every partition $j \in P$, we have $x_{vj} \leq 1$. The second equality holds by constraints (3b). Furthermore, we have

$$\hat{y}_{uv} = 1 - \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} \leq 1 - 0 = 1.$$

Here, the first equality holds by constraints (4). The inequality holds because for every partition $j \in P$, we have $x_{uj} \geq 0$ and $x_{vj} \geq 0$. This finishes the proof. □

For every $n \in \mathbb{Z}_{++}$, we define $[n] \coloneqq \{1, \dots, n\}$. For every set S, we employ $\binom{S}{2}$ to denote all subsets of S with size 2. First, we prove Lemma 1 that will be used in our further analyses.

Lemma 1. *Let $a \in [0,1]^n$ with $n \geq 2$. Then, we have*

$$1 - \sum_{i \in [n]} a_i + \sum_{\{i,j\} \in \binom{[n]}{2}} a_i a_j \geq 0. \tag{5}$$

Proof. We prove the claim by induction. First, we show that the inequality holds for the base case $n = 2$. So, we have

$$1 - a_1 - a_2 + a_1 a_2 = (1 - a_1)(1 - a_2) \geq 0. \tag{6}$$

Here, the inequality holds because for every $i \in \{1, 2\}$, we have $1 - a_i \geq 0$.

Now suppose that inequality (5) holds for $n = s \geq 2$ (induction hypothesis). It suffices to show that it also holds for $n = s + 1$.

$$\begin{aligned} 0 &\leq \Big(1 - \sum_{i \in [s]} a_i + \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j\Big)(1 - a_{s+1}) && \text{(7a)} \\ &= 1 - \sum_{i \in [s+1]} a_i + \sum_{\{i,j\} \in \binom{[s+1]}{2}} a_i a_j - a_{s+1} \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j && \text{(7b)} \\ &\leq 1 - \sum_{i \in [s+1]} a_i + \sum_{\{i,j\} \in \binom{[s+1]}{2}} a_i a_j. && \text{(7c)} \end{aligned}$$

Here, inequality (7a) holds by induction hypothesis and because $1 - a_{s+1} \geq 0$. Inequality (7c) holds because $-a_{s+1} \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j \leq 0$. This completes the proof. □

Now we define the polytope of the A-MILO formulation (1) as follows.

$$\mathcal{R}_{\text{A-MILO}} \coloneqq \Big\{(x, y) \in [0,1]^{n \times k} \times [0,1]^m \;\Big|\; (x, y) \text{ satisfies constraints (1b)–(1e)}\Big\}.$$

Proposition 1 shows that the continuous relaxation of the BQO model is stronger than that of the A-MILO model.

Proposition 1. $\mathcal{R}_{\text{BQO}} \subset \mathcal{R}_{\text{A-MILO}}$.

Proof. Let point $(\hat{x}, \hat{y}) \in \mathcal{R}_{\text{BQO}}$. First, we are to show that $(\hat{x}, \hat{y}) \in \mathcal{R}_{\text{A-MILO}}$. We show that $(\hat{x}, \hat{y})$ satisfies constraints (1c). For every edge $\{u, v\} \in E$ and every partition $j \in P$, we have

$$\begin{aligned}
\hat{y}_{uv} &= 1 - \sum_{i \in P} \hat{x}_{ui}\hat{x}_{vi} && \text{(8a)}\\
&= \sum_{i \in P} \hat{x}_{ui} - \sum_{i \in P} \hat{x}_{ui}\hat{x}_{vi} && \text{(8b)}\\
&= \hat{x}_{uj} + \sum_{i \in P\setminus\{j\}} \hat{x}_{ui} - \hat{x}_{uj}\hat{x}_{vj} - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(8c)}\\
&\geq \hat{x}_{uj} + \sum_{i \in P\setminus\{j\}} \hat{x}_{ui} - \hat{x}_{vj} - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(8d)}\\
&= \hat{x}_{uj} - \hat{x}_{vj} + \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}(1 - \hat{x}_{vi}) && \text{(8e)}\\
&\geq \hat{x}_{uj} - \hat{x}_{vj}. && \text{(8f)}
\end{aligned}$$

Here, equality (8a) holds by constraints (4). Equality (8b) follows from constraint (3b). Inequality (8d) holds because $\hat{x}_{uj} \leq 1$. Inequality (8f) holds because $\sum_{i \in P\setminus\{j\}} \hat{x}_{ui}(1 - \hat{x}_{vi}) \geq 0$. Similarly, one can show that $(\hat{x}, \hat{y})$ satisfies constraints (1d).

Finally, we show that $(\hat{x}, \hat{y})$ satisfies constraints (1e). For every edge $\{u, v\} \in E$ and every partition $j \in P$, we have

$$\begin{aligned}
\hat{y}_{uv} &= 1 - \sum_{i \in P} \hat{x}_{ui}\hat{x}_{vi} && \text{(9a)}\\
&= 1 - \hat{x}_{uj}\hat{x}_{vj} - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(9b)}\\
&= 2 - 1 - \hat{x}_{uj}\hat{x}_{vj} - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(9c)}\\
&= 2 - (1 + \hat{x}_{uj}\hat{x}_{vj}) - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(9d)}\\
&\leq 2 - (\hat{x}_{uj} + \hat{x}_{vj}) - \sum_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} && \text{(9e)}\\
&\leq 2 - (\hat{x}_{uj} + \hat{x}_{vj}). && \text{(9f)}
\end{aligned}$$

Here, equality (9a) holds by constraints (4). Inequality (9e) holds by inequality (6) in Lemma 1. Furthermore, inequality (9f) follows from inequality $\sum\limits_{i \in P\setminus\{j\}} \hat{x}_{ui}\hat{x}_{vi} \geq 0$.

Now, we are to show that there exists a point $(\hat{x}, \hat{y}) \in \mathcal{R}_{\text{A-MILO}}$ such that $(\hat{x}, \hat{y}) \notin \mathcal{R}_{\text{BQO}}$. For every $v \in V$, let $\hat{x}_{v1} = \hat{x}_{v2} = 0.5$. For every vertex $v \in V$ and every partition $j \in \{3, 4, \ldots, k\}$, we define $\hat{x}_{vj} = 0$. Furthermore, for every edge $\{u, v\} \in E$, we define $\hat{y}_{uv} = 1$. So, point $(\hat{x}, \hat{y}) \in \mathcal{R}_{\text{A-MILO}} \setminus \mathcal{R}_{\text{BQO}}$ because $(\hat{x}, \hat{y})$ violates constraints (4). Thus, the proof is complete. □

Now we introduce a reduced variant of the BQO formulation: R-BQO. In this model, the number of variables are reduced by n. For every vertex $v \in V$ and partition $k \in P$ (the last partition), we define

$$x_{vk} \coloneqq 1 - \sum_{j \in P \setminus \{k\}} x_{vj}. \tag{10}$$

For analysis purposes, we write the reduced BQO (R-BQO) formulation as follows.

$$\text{(R-BQO)} \qquad \begin{aligned} \max \quad & \sum_{\{u,v\} \in E} w_{uv} \Big[1 - \sum_{j \in P \setminus \{k\}} x_{uj} x_{vj} - \Big(1 - \sum_{j \in P \setminus \{k\}} x_{uj}\Big)\Big(1 - \sum_{j \in P \setminus \{k\}} x_{vj}\Big)\Big] && \text{(11a)} \\ \text{s.t.} \quad & \sum_{j \in P \setminus \{k\}} x_{vj} \leq 1 \qquad \forall v \in V && \text{(11b)} \\ & x \in \{0,1\}^{n \times (k-1)}. && \text{(11c)} \end{aligned}$$

It should be noted that for every vertex $v \in V$, we have $x_{vk} \in \{0,1\}$. We also define the polytope of the R-BQO formulation (11) as follows.

$$\mathcal{R}_{\text{R-BQO}} \coloneqq \Big\{ (x,y) \in [0,1]^{n \times k} \times \mathbb{R}^m \;\Big|\; (x,y) \text{ satisfies constraints (11b), (10), and (4)} \Big\}.$$

Corollary 1 highlights the relationship between projections of the continuous relaxations of BQO, R-BQO and A-MILO formulations on the x space.

Corollary 1. $\operatorname{proj}_x \mathcal{R}_{\text{R-BQO}} = \operatorname{proj}_x \mathcal{R}_{\text{BQO}} = \operatorname{proj}_x \mathcal{R}_{\text{A-MILO}}$.

Proof. The first equality is straightforward. The second equality holds by Proposition 1 and the fact that any x that satisfies constraints (1b)–(1e) of the A-MILO formulation also satisfies constraints (3b) in the BQO formulation (3). □

To conduct a theoretical comparison between polytopes of the BQO and P-MILO formulations, we lift the dimensionality of the BQO formulation by introducing new z variables.

$$z_{uv} \coloneqq \sum_{j \in P} x_{uj} x_{vj}, \qquad \forall \{u,v\} \in \binom{V}{2}. \tag{12}$$

We call the lifted polytope of the $\mathcal{R}_{\text{BQO}}$ as $\mathcal{R}^+_{\text{BQO}}$.

$$\mathcal{R}^+_{\text{BQO}} \coloneqq \Big\{ (x,y,z) \in [0,1]^{n \times k} \times \mathbb{R}^m \times \mathbb{R}^{\binom{n}{2}} \;\Big|\; (x,y) \in \mathcal{R}_{\text{BQO}} \text{ and } z \text{ satisfies (12)} \Big\}.$$

We also define the polytope of the P-MILO formulation as follows.

$$\mathcal{R}_{\text{P-MILO}} \coloneqq \Big\{ z \in [0,1]^{\binom{n}{2}} \;\Big|\; z \text{ satisfies constraints (2b)–(2c)} \Big\}.$$

We show that the continuous relaxation of a projection of the lifted BQO formulation on the z space is stronger than that of the P-MILO formulation.

Proposition 2. $\operatorname{proj}_z \mathcal{R}^+_{\text{BQO}} \subset \mathcal{R}_{\text{P-MILO}}$ *for* $n > k$.

Proof. Consider a point $(\hat{x}, \hat{y}, \hat{z}) \in \mathcal{R}^{+}_{\text{BQO}}$. We are to show that $\hat{z} \in \mathcal{R}_{\text{P-MILO}}$. For every set $\{u, v, w\} \subseteq V$, we show that point $\hat{z}$ satisfies constraints (2b).

$$\begin{aligned}
\hat{z}_{uv} + \hat{z}_{vw} &= \sum_{j\in P} \hat{x}_{uj}\hat{x}_{vj} + \sum_{j\in P} \hat{x}_{vj}\hat{x}_{wj} && \text{(13a)}\\
&= \sum_{j\in P} \hat{x}_{vj}(\hat{x}_{uj} + \hat{x}_{wj}) && \text{(13b)}\\
&\le \sum_{j\in P} \hat{x}_{vj}(1 + \hat{x}_{uj}\hat{x}_{wj}) && \text{(13c)}\\
&= \sum_{j\in P} \hat{x}_{vj} + \sum_{j\in P} \hat{x}_{vj}(\hat{x}_{uj}\hat{x}_{wj}) && \text{(13d)}\\
&= 1 + \sum_{j\in P} \hat{x}_{vj}(\hat{x}_{uj}\hat{x}_{wj}) && \text{(13e)}\\
&\le 1 + \sum_{j\in P} \hat{x}_{uj}\hat{x}_{wj} && \text{(13f)}\\
&= 1 + \hat{z}_{uw}. && \text{(13g)}
\end{aligned}$$

Here, equality (13a) holds by definition (12). Inequality (13c) holds by inequality (6) in Lemma 1. Equality (13e) holds by constraints (3b). Inequality (13f) holds by the fact that $\hat{x}_{vj} \le 1$. Equality (13g) holds by definition (12).

Furthermore, we show that point $\hat{z}$ satisfies constraints (2c). For every vertex set $Q \subseteq V$ with $|Q| = k + 1$, we have

$$\begin{aligned}
\sum_{\{u,v\}\in\binom{Q}{2}} \hat{z}_{uv} &= \sum_{\{u,v\}\in\binom{Q}{2}} \sum_{j\in P} \hat{x}_{uj}\hat{x}_{vj} && \text{(14a)}\\
&= \sum_{j\in P} \Big(\sum_{\{u,v\}\in\binom{Q}{2}} \hat{x}_{uj}\hat{x}_{vj} \Big) && \text{(14b)}\\
&\ge \sum_{j\in P} \Big(\sum_{u\in Q} \hat{x}_{uj} - 1 \Big) && \text{(14c)}\\
&= \sum_{u\in Q} \sum_{j\in P} \hat{x}_{uj} - k && \text{(14d)}\\
&= k + 1 - k = 1. && \text{(14e)}
\end{aligned}$$

Here, equality (14a) holds by definition (12). Inequality (14c) holds by Lemma 1. Equality (14e) holds by constraints (3b) and because $|Q| = k + 1$.

Finally, for every $\{u, v\} \in \binom{V}{2}$, we show that $0 \le \hat{z}_{uv} \le 1$. Because for every vertex $v \in V$ and every partition $j \in P$ we have $\hat{x}_{vj} \ge 0$, it follows that $\hat{z}_{uv} \ge 0$. For every $\{u, v\} \in \binom{V}{2}$, we show that $\hat{z}_{uv} \le 1$.

$$\hat{z}_{uv} = \sum_{j\in P} \hat{x}_{uj}\hat{x}_{vj} \le \sum_{j\in P} \hat{x}_{uj} = 1.$$

Here, the first equality holds by definition (12). The inequality holds because $\hat{x}_{vj} \le 1$ for every vertex $v \in V$ and every partition $j \in P$. The last equality holds by constraints (3b). This implies that $\text{proj}_z\, \mathcal{R}^{+}_{\text{BQO}} \subseteq \mathcal{R}_{\text{P-MILO}}$.

Now we show that the inclusion is strict for any non-trivial instance of the max k-cut problem with $n > k$. For every $\{u, v\} \in \binom{V}{2}$, we define $\hat{z}_{uv}$ as a point that belongs to the polytope of the P-MILO formulation; that is, $\hat{z} \in \mathcal{R}_{\text{P-MILO}}$.

$$\hat{z}_{uv} := \frac{2}{k(k+1)}.$$

For every vertex $v \in V$, let $\mathbf{x}_v \in [0,1]^k$ be the assignment vector of vertex v. By definition (12), we have

$$\hat{z}_{uv} = \hat{\mathbf{x}}_u^T \hat{\mathbf{x}}_v = \|\hat{\mathbf{x}}_u\|_2 \|\hat{\mathbf{x}}_v\|_2 \cos \hat{\theta}_{uv}. \tag{15}$$

By constraints (3b), we have $\|\hat{\mathbf{x}}_v\|_1 = 1$. Then for every vertex $v \in V$, we have

$$\frac{1}{\sqrt{k}} \leq \|\hat{\mathbf{x}}_v\|_2 \leq 1. \tag{16}$$

Here, the first inequality holds because $\|\hat{\mathbf{x}}_v\|_2$ reaches its minimum when $\hat{x}_{vj} = {}^1\!/_k$ for every partition $j \in P$. The second inequality holds because $\|\hat{\mathbf{x}}_v\|_2 \leq \|\hat{\mathbf{x}}_v\|_1 = 1$.

By lines (15) and (16), we have

$$\frac{2}{k(k+1)} \leq \cos \hat{\theta}_{uv} \leq \frac{2}{k+1}.$$

For every $\{u, v\} \in \binom{V}{2}$, this implies that we have the following relations because $k \geq 2$.

$$48^\circ < \arccos\left\{\frac{2}{k+1}\right\} \leq \hat{\theta}_{uv} \leq \arccos\left\{\frac{2}{k(k+1)}\right\} < 90^\circ. \tag{17}$$

Because all vectors $\hat{\mathbf{x}}_v$ are in the positive orthant and $n > k$, there are vectors $\hat{\mathbf{x}}_a$ and $\hat{\mathbf{x}}_b$ with $\hat{\theta}_{ab} \leq 45^\circ$. However, this contradicts the first inequality of (17). So, there is no feasible solution of the BQO formulation that satisfies definition (12). This completes the proof. □

Now we provide a comparison between A-MILO and P-MILO formulations. First, we define the lifted polytope of the A-MILO formulation in (x, y, z) space.

$$\mathcal{R}^{+}_{\text{A-MILO}} := \left\{ (x, y, z) \in [0,1]^{n \times k} \times [0,1]^m \times [0,1]^{\binom{n}{2}} \;\middle|\; (x, y) \in \mathcal{R}_{\text{A-MILO}} \text{ and } z \text{ satisfies (12)} \right\}.$$

Theorem 1 shows that the continuous relaxation of the A-MILO formulation is stronger than the continuous relaxation of the P-MILO formulation under mapping (12). Our result is different from that of Fairbrother and Letchford (2017) who studied a projection of the P-MILO formulation on a subspace of the A-MILO formulation. Further, it is different from the result of Alès and Knippel (2020) who provided a comparison between two extended P-MILO formulations with "representative" variables.

Theorem 1. $\operatorname{proj}_z \mathcal{R}^{+}_{\text{A-MILO}} \subset \mathcal{R}_{\text{P-MILO}}$ *for* $n > k$.

Proof. The proof follows by $\operatorname{proj}_x \mathcal{R}^{+}_{\text{BQO}} = \operatorname{proj}_x \mathcal{R}_{\text{BQO}} = \operatorname{proj}_x \mathcal{R}_{\text{A-MILO}} = \operatorname{proj}_x \mathcal{R}^{+}_{\text{A-MILO}}$ (Corollary 1) and $\operatorname{proj}_z \mathcal{R}^{+}_{\text{BQO}} \subset \mathcal{R}_{\text{P-MILO}}$ (Proposition 2). □

G. Wang and Hijazi (2020) show that their RP-MILO model is as strong as the projection of the P-MILO model on the edges of an extended chordal graph. Further, their computational experiments show the computational superiority of their RP-MILO formulation on chordal graphs.

4 Quadratic Unconstrained Binary Optimization Models

The max k-cut problem can be solved on a quantum machine by formulating the problem as an unconstrained binary optimization formulation. One can move constraints of a mixed integer optimization model of the max k-cut problem to the objective function and penalize them accordingly. Padberg (1989) conducted a polyhedral study on a general form of the QUBO model for the max cut problem and its naïve MILO continuous relaxation. Butenko (2003) proposed a QUBO formulation for the maximum independent set problem. Dunning et al. (2018) conducted an extensive set of experiments to evaluate the performance of different heuristic algorithms for solving the max cut with QUBO formulation. Quintero et al. (2022) proposed a QUBO formulation for the maximum k-colorable subgraph problem.

We choose to reformulate the BQO model as a quadratic unconstrained binary optimization (QUBO) model because of the following reasons:

(i) the BQO model has the least number of variables, which is equal to the number of qubits in quantum circuits;

(ii) the BQO model has the least number of constraints that directly affect the number of gates in quantum circuits;

(iii) the BQO model has no inequality constraint; so, its constraints can be penalized without introducing any auxiliary variable; and

(iv) the quantum approximate optimization algorithm (QAOA) can handle QUBO models.

We first provide some notations. We define edge subsets $E^- := \Big\{\{u,v\} \in E \mid w_{uv} < 0\Big\}$ and $E^+ := \Big\{\{u,v\} \in E \mid w_{uv} > 0\Big\}$. For every vertex v, we also define the following notations.

$$\begin{aligned} N_G^+(v) &:= \Big\{u \in N_G(v) \;\Big|\; w_{uv} > 0\Big\}, \quad \text{and} \quad N_G^-(v) := \Big\{u \in N_G(v) \;\Big|\; w_{uv} < 0\Big\}, \\ d_v^+ &:= \sum_{u \in N_G^+(v)} w_{uv}, \quad \text{and} \quad d_v^- := \sum_{u \in N_G^-(v)} w_{uv}. \end{aligned} \tag{18}$$

We propose two quadratic unconstrained binary optimization formulations. These formulations generalize the existing QUBO model for the max cut problem in which $k = 2$. We first propose a QUBO model inferred from the BQO model. In other words, we move constraints (3b) of the BQO formulation to the objective function and penalize them by a vector $c \in \mathbb{R}_+^n$. Our proposed QUBO model is as follows.

$$\text{(QUBO)} \qquad \max_{x \in \{0,1\}^{n \times k}} q(x) := \sum_{\{u,v\} \in E} w_{uv}\Big(1 - \sum_{j \in P} x_{uj}x_{vj}\Big) - \sum_{v \in V} c_v \Big(\sum_{j \in P} x_{vj} - 1\Big)^2. \tag{19}$$

An optimal solution of the QUBO formulation (19) is not necessarily a feasible solution of the max k-cut problem. We propose Algorithm 1 that converts any binary point $\hat{x} \in \{0,1\}^{n \times k}$ to a feasible solution of the max k-cut problem. In this algorithm, vertex sets I_0 and I_1 represent the set of vertices with no assigned partition and multiple assigned partitions, respectively. The *while* loop assigns vertices with the same multiple assignments to a partition that locally maximizes the objective function over their incident edges with negative weights. Lines 5–15 runs in $\mathcal{O}(knm)$. The

Algorithm 1 Conversion of a binary infeasible solution of the BQO model to a feasible solution

```
Require: $(G, \hat{x}, P)$
 1: let $I_0 \coloneqq \left\{ v \in V \;\middle|\; \sum_{j \in P} \hat{x}_{vj} = 0 \right\}$
 2: let $I_1 \coloneqq \left\{ v \in V \;\middle|\; \sum_{j \in P} \hat{x}_{vj} > 1 \right\}$
 3: $\bar{x} \leftarrow \hat{x}$
 4: $\ell \leftarrow 1$
 5: while $I_1 \neq \varnothing$ do
 6:     let $a \in I_1$
 7:     $C_\ell \coloneqq \{ u \in I_1 \mid \hat{x}_{uj} = \hat{x}_{aj}, \forall j \in P \}$
 8:     $E_\ell \coloneqq \{ \{u, v\} \in E^- \mid \{u, v\} \cap C_\ell \neq \varnothing \}$
 9:     $P_\ell \coloneqq \{ j \in P \mid \hat{x}_{aj} = 1 \}$
10:     let $s \in \operatorname{argmin}_{j \in P_\ell} \left\{ \sum_{\{u,v\} \in E_\ell} w_{uv} \hat{x}_{uj} \hat{x}_{vj} \right\}$
11:     for every vertex $u \in C_\ell$ do
12:         for every partition $j \in P_\ell \setminus \{s\}$ do
13:             fix $\bar{x}_{uj} = 0$
14:     $I_1 \leftarrow I_1 \setminus C_\ell$
15:     $\ell \leftarrow \ell + 1$
16: for every vertex $v \in I_0$ do
17:     let $s \in \operatorname{argmin}_{j \in P} \left\{ \sum_{u \in N_G(v)} w_{uv} \bar{x}_{uj} \right\}$
18:     fix $\bar{x}_{vs} = 1$
19: return $\bar{x}$
```

last *for* loop assigns vertices with no assignment to a partition that locally maximizes the objective function. Lines 16–18 runs in $\mathcal{O}(km)$. In total, Algorithm 1 takes time $\mathcal{O}(knm)$.

One can always set a "big" penalty vector c in QUBO formulation (19) to ensure that there is an optimal solution of the unconstrained formulations such that it represents a feasible solution of the max k-cut problem. The following remark provides naïve penalty coefficients for the QUBO model (19).

Remark 2. *For any vertex $v \in V$, penalty coefficient $c_v = \sum_{\{u,v\} \in E} |w_{uv}|$ ensures there is an optimal solution of the* QUBO *model* (19) *that represents a feasible solution of the max k-cut problem.*

Lemma 2 provides a tight lower bound for the penalty vector c in QUBO formulation (19). The proof is provided in Appendix A.

Lemma 2. *Let c be a penalty vector and $\hat{x}$ be an optimal solution of the* QUBO *model* (19). *If $c_v \geq \max\left\{ d_v^+/k, -d_v^-/2 \right\}$ for every vertex $v \in V$, then Algorithm 1 returns a feasible solution of the* BQO *model that is optimal for* QUBO.

We note that the value of tight penalty coefficients compared to naïve ones is reduced from $O(m)$ to $O(n)$. The following theorem shows that Algorithm 1 returns an optimal solution of the BQO model (3) if an optimal solution of the QUBO model (19) is provided.

Theorem 2. *Let $\hat{x}$ be an optimal solution of the* QUBO *formulation* (19) *with* $c_v \geq \max\left\{\frac{d_v^+}{k}, -\frac{d_v^-}{2}\right\}$ *for every vertex* $v \in V$. *Algorithm 1 returns an optimal solution of the* BQO *formulation* (3).

Proof. Let $\bar{x}$ be a point returned by Algorithm 1 applied on optimal solution $\hat{x}$. Further, assume that x^* represents an optimal solution of the max k-cut problem. Since $\hat{x}$ is an optimal solution of the QUBO formulation (19), we have (i) $q(\hat{x}) \geq q(\bar{x})$, and (ii) $q(\hat{x}) \geq q(x^*)$. By Lemma 2, we have (iii) $q(\bar{x}) \geq q(\hat{x})$. By (i) and (iii), $q(\hat{x}) = q(\bar{x})$. Hence, $q(\bar{x}) \geq q(x^*)$ by (ii). Note that $\bar{x}$ is feasible for the BQO formulation (3) by Lemma 2; so, we have $q(\bar{x}) \leq q(x^*)$. Thus, $q(\hat{x}) = q(\bar{x}) = q(x^*)$ and $\bar{x}$ is also an optimal solution of the BQO formulation (3). □

It should be noted that if $c_v > \max\left\{\frac{d_v^+}{k}, -\frac{d_v^-}{2}\right\}$ for every vertex $v \in V$, then an optimal solution of the QUBO formulation (19) is also optimal for the BQO formulation (3). Example 1 shows that there are some instances of the max k-cut problem for which Theorem 2 does not hold if we have $c_v < \max\left\{\frac{d_v^+}{k}, -\frac{d_v^-}{2}\right\}$ for some vertex $v \in V$.

Example 1. Figure 2 illustrates an instance of the max 3-cut problem with the optimal objective value of 7. Let x^* be an optimal solution with $x^*_{11} = x^*_{31} = x^*_{41} = 1$, $x^*_{22} = 1$, and $x^*_{53} = 1$. See the leftmost side of Figure 2 for an illustration. Furthermore, the QUBO formulation (19) of the max 3-cut problem is written as follows.

$$q(x) = 7 - \sum_{\{u,v\} \in E} w_{uv}\left(x_{u1}x_{v1} + x_{u2}x_{v2} + x_{u3}x_{v3}\right) - \sum_{v \in V} c_v\left(x_{v1} + x_{v2} + x_{v3} - 1\right)^2, \tag{20}$$

Note that $c_1 = -\frac{-2}{2} = 1$, $c_2 = c_3 = \max\left\{\frac{6}{3}, \frac{1}{2}\right\} = 2$, and $c_4 = c_5 = \frac{3}{3} = 1$. Then, we have $q(x^*) = 7$. Now, we change c_5 from 1 to $1 - \epsilon$ for some $\epsilon > 0$. Then, $\hat{x}$ is an optimal solution for the QUBO formulation (20) with $\hat{x}_{11} = \hat{x}_{31} = \hat{x}_{41} = 1$ and $\hat{x}_{22} = 1$. Further, $\hat{x}_{51} = \hat{x}_{52} = \hat{x}_{53} = 0$. See Figure 2 (center) for an illustration. However, this implies $\hat{x}$ is an infeasible solution for the max 3-cut problem with $q(\hat{x}) = 7 + \epsilon$ and $q(\hat{x}) > q(x^*)$.

Similarly, we change c_1 from 1 to $1 - \epsilon$ for some $\epsilon > 0$. Then, $\tilde{x}$ is an optimal solution for the QUBO formulation (20) with $\tilde{x}_{11} = \tilde{x}_{12} = 1$, but it is an infeasible solution of the max 3-cut problem with $q(\tilde{x}) = 7 + \epsilon$ and $q(\tilde{x}) > q(x^*)$. See the rightmost side of Figure 2 for an illustration. Hence, an inappropriate choice of c might not provide a solution with the optimal objective value for the max k-cut problem. This implies that our proposed lower bounds for penalty coefficients are tight for some instances.

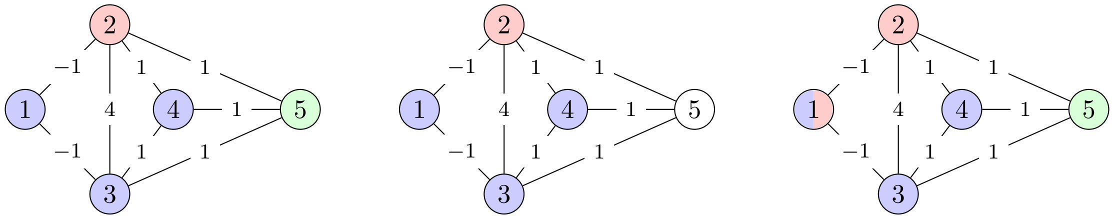


Figure 2: Optimal solutions of the QUBO model for the max 3-cut problem with different penalty coefficients: (left) an optimal solution x^* of the BQO model with $c_1 = c_5 = 1$, (center) an infeasible solution $\hat{x}$ of the BQO model with $c_1 = 1$ and $c_5 = 1 - \epsilon$, and (right) an infeasible solution $\tilde{x}$ of the BQO model with $c_1 = 1 - \epsilon$ and $c_5 = 1$.

Similar to the R-BQO model (11), we propose a reduced QUBO (R-QUBO) model. We define partition set $\bar{P} \coloneqq P \setminus \{k\}$. The R-QUBO formulation with $n(k-1)$ binary variables is provided below.

$$\text{(R-QUBO)} \quad \max_{x \in \{0,1\}^{n \times (k-1)}} \bar{q}(x) \coloneqq \sum_{\{u,v\} \in E} w_{uv} \Big[1 - \sum_{j \in \bar{P}} x_{uj} x_{vj} - \Big(1 - \sum_{j \in \bar{P}} x_{uj}\Big)\Big(1 - \sum_{j \in \bar{P}} x_{vj}\Big)\Big] - \sum_{v \in V} c_v \sum_{\{i,j\} \in \binom{\bar{P}}{2}} x_{vi} x_{vj}. \tag{21}$$

For $k = 2$, it is worth noting that the penalty term disappears because $\binom{\bar{P}}{2} = \varnothing$. We also propose Algorithm 2 that converts any infeasible binary solution $\hat{x} \in \{0,1\}^{n \times (k-1)}$ of the R-BQO model (11) to a feasible solution. Similar to Algorithm 1, Algorithm 2 has time complexity $\mathcal{O}(knm)$.

Algorithm 2 Conversion of a binary infeasible solution of the R-BQO model to a feasible solution

```
Require: $(G, \hat{x}, \bar{P})$
$I \coloneqq \Big\{v \in V \;\Big|\; \sum_{j \in \bar{P}} \hat{x}_{vj} > 1\Big\}$
$\bar{x} \leftarrow \hat{x}$
$\ell \leftarrow 1$
while $I \neq \varnothing$ do
    $C_\ell \coloneqq \{u \in I \mid \hat{x}_{uj} = \hat{x}_{aj}, \forall j \in \bar{P}\}$
    $E_\ell \coloneqq \Big\{\{u,v\} \in E \;\Big|\{u,v\} \cap C_\ell \neq \varnothing\Big\}$
    $\bar{P}_\ell \coloneqq \{j \in \bar{P} \mid \hat{x}_{aj} = 1\}$
    let $s \in \operatorname{argmin}_{j \in \bar{P}_\ell} \Big\{\sum_{\{u,v\} \in E_\ell} w_{uv} \bar{x}_{uj} \bar{x}_{vj}\Big\}$
    for every vertex $u \in C_\ell$ do
        for every partition $j \in \bar{P}_\ell \setminus \{s\}$ do
            fix $\bar{x}_{uj}$ to zero
    $I \leftarrow I \setminus C_\ell$
    $\ell \leftarrow \ell + 1$
return $\bar{x}$
```

The following remark provides naïve penalty coefficients for the R-QUBO model (21).

Remark 3. *For any vertex $v \in V$, penalty coefficient $c_v = \sum_{\{u,v\} \in E^+} k w_{uv} - \sum_{\{u,v\} \in E^-} k^2 w_{uv}$ ensures there is an optimal solution of the* R-QUBO *model* (21) *that represents a feasible solution of the max k-cut problem.*

Lemma 3 provides a tight lower bound for the penalty vector c in R-QUBO formulation (21). The proof is provided in Appendix B.

Lemma 3. *Let c be a penalty vector and $\hat{x}$ be an optimal solution of the* R-QUBO *model* (21)*. If $c_v \geq d_v^+ - d_v^-$ for every vertex $v \in V$, then Algorithm 2 returns a feasible solution of the* R-BQO *model that is optimal for* R-QUBO*.*

We note that the value of tight penalty coefficients compared to naïve ones is reduced from $O(k^2m)$ to $O(n)$. The following theorem shows that Algorithm 2 returns an optimal solution of the R-BQO model (11) if an optimal solution of the R-QUBO model (21) is provided.

Theorem 3. *Suppose $\hat{x}$ is an optimal solution for the* R-QUBO *model* (21) *with $c_v \geq d_v^+ - d_v^-$ for every vertex $v \in V$. Algorithm 2 returns a binary optimal solution of* R-BQO *model* (11).

Proof. The proof is similar to the proof of Theorem 2. □

It should be noted that if $c_v > d_v^+ - d_v^-$ for every vertex $v \in V$, then an optimal solution of the R-QUBO formulation (21) is also a feasible solution for the R-BQO formulation (11). Example 2 shows that there is an instance of the max k-cut problem for which Theorem 3 is violated if $c_v < d_v^+ - d_v^-$ holds for some vertex $v \in V$.

Example 2. Figure 3 illustrates an instance of the max 3-cut problem with the optimal objective value of 6. Let x^* be an optimal solution with $x^*_{21} = x^*_{31} = 1$, $x^*_{11} = x^*_{12} = 0$, $x^*_{41} = x^*_{42} = 0$, and $x^*_{52} = 1$. See the left side of Figure 3 for an illustration. Furthermore, the corresponding R-QUBO formulation is as follows

$$\bar{q}(x) = 5 - \sum_{\{u,v\}\in E} w_{uv}\Big[\sum_{j\in\{1,2\}} x_{uj}x_{vj} + \Big(1 - \sum_{j\in\{1,2\}} x_{uj}\Big)\Big(1 - \sum_{j\in\{1,2\}} x_{vj}\Big)\Big] - \sum_{v\in V} c_v x_{v1}x_{v2}.$$

Let $c_2 = c_3 = 4$ and $c_v = 2$ for every $v \in \{1,4,5\}$. Here, we have $\bar{q}(x^*) = 6$. Now, we change c_2 from 4 to $4-\epsilon$ for some $\epsilon > 0$. Let $\hat{x}$ be an optimal solution of the modified R-QUBO with $\hat{x}_{21} = \hat{x}_{22} = 1$, $\hat{x}_{31} = \hat{x}_{32} = 1$, and $\hat{x}_{vj} = 0$ for $v \in \{1,4,5\}$ and $j \in \{1,2\}$. Then, we have $\bar{q}(\hat{x}) = 6 + \epsilon$ for the modified R-QUBO formulation. Because $\bar{q}(\hat{x}) \geq \bar{q}(x^*)$, an optimal solution of the max 3-cut is not optimal for the modified R-QUBO formulation. Thus, there is an instance for which penalty coefficients are tight.

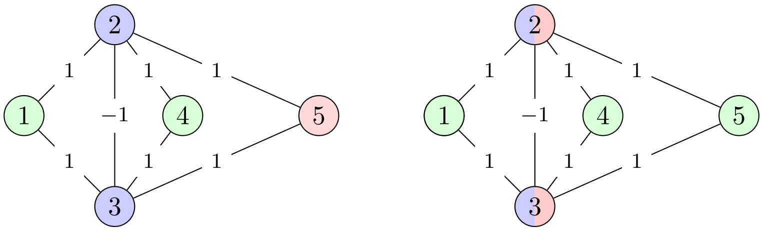


Figure 3: Optimal solutions of the R-QUBO model for the max 3-cut problem with different penalty coefficients: (left) an optimal solution x^* of the BQO model with $c_1 = c_4 = c_5 = 2$ and $c_2 = c_3 = 4$, and (right) an infeasible solution $\hat{x}$ of the BQO model after changing c_2 from 4 to $4-\epsilon$.

5 Preprocessing algorithms

Preprocessing techniques have had a significant impact on the progress of mixed integer optimization solvers (e.g., see Savelsbergh (1994) and Achterberg et al. (2020)). This section provides preprocessing techniques to improve the computational performance of the mixed integer optimization and QUBO formulations on classical and quantum solvers. We (i) adopt two existing preprocessing operations (vertex peeling and biconnected decomposition) and (ii) propose a novel preprocessing operation (folding). These operations are iteratively applied until none is applicable on the graph. The preprocessing techniques are crucial for running the proposed QUBO models on quantum machines since the number of qubits is limited. We also note that the preprocessing algorithms can solve the max k-cut problem for multiple instances of Fuchs et al. (2021).

5.1 Peeling operation

Given a vertex $v \in V$ with (i) all positive incident edges and (ii) a degree less than k, we can safely remove it from the graph and determine its partition in a postprocessing procedure. Let P_v be the set of partitions to which neighbors of vertex v are assigned. Then, we can assign vertex v to a partition $j \in P \setminus P_v$ because the degree of vertex v is less than k. As in our computational experiments, this procedure can be applied iteratively. Méndez-Díaz and Zabala (2006) employed a similar procedure for solving the graph coloring problem. Figure 4 illustrates the peeling operation on a weighted graph.

Remark 4. *Let G be a graph with positive edge weights. Then, solving the max k-cut problem on garph G is equivalent to solving it on the k-core of the graph.*

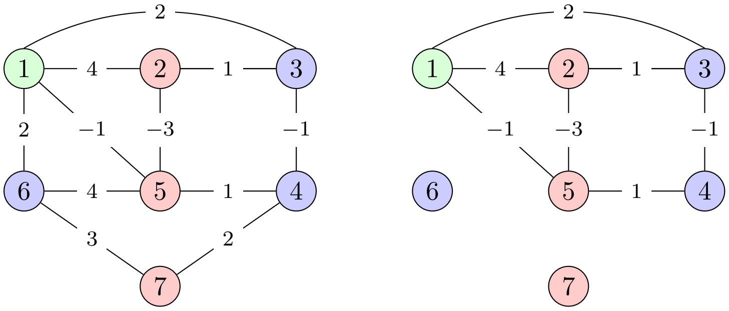


Figure 4: A schematic of the peeling operation for the max 3-cut problem: (left) input graph, and (right) the peeled graph.

5.2 Biconnected decomposition

Fairbrother et al. (2018) proposed a procedure that decomposes the problem based on cut vertices (see e.g., Gross and Yellen, 2003, Section 1.1.3) and their corresponding biconnected components (see e.g., Gross and Yellen, 2003, Section 10.1.4). The procedure starts solving the max k-cut problem on an arbitrary biconnected component of graph G. Then, it solves the problem on a biconnected component that has a common cut vertex with a component whose vertices are already assigned to some partitions. Note that at each step of the procedure, the partition of the common cut vertex is fixed. Figure 5 shows a schematic of biconnected decomposition on an unweighted graph.

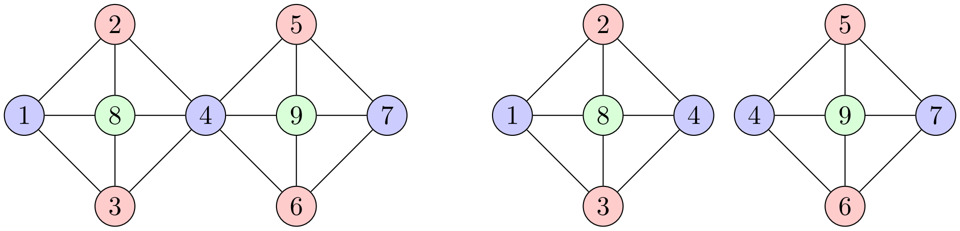


Figure 5: A schematic of biconnected decomposition for the max 3-cut problem: (left) input graph G', and (right) biconnected decomposed graph.

5.3 Folding operation

For ease of notation, we consider the input graph $G = (V, E)$ as a complete graph with $w_{uv} = 0$ for every $\{u, v\} \in \binom{V}{2} \setminus E$. Given vertex pair $\{a, b\} \in \binom{V}{2}$, a folding operation applies the following steps:

(i) remove vertices a and b, and

(ii) add a new vertex c and edges $\{c, v\}$ for every $v \in V \setminus \{a, b\}$ with weights $w_{vc} \coloneqq w_{va} + w_{vb}$.

The graph obtained after applying the folding operation is called the *folded graph* F.

Definition 1 (Safe Folding)**.** A folding operation is *safe* if any optimal solution of the max k-cut problem on the folded graph can be converted to an optimal solution on the original graph.

In Section 6.3, we show that the folding operation can be applied iteratively to reduce the size of the input graph. Figure 6 illustrates a folding operation on vertex pair $\{1, 3\}$ with $a =$ "1", $b =$ "3", and $c =$ "1, 3".

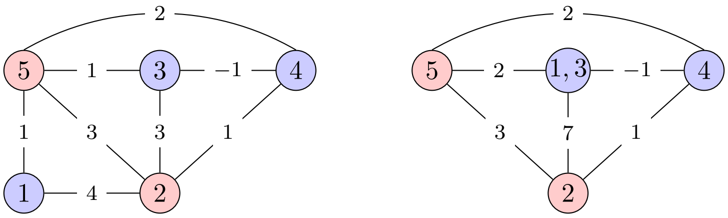


Figure 6: An instance of the max cut problem: (left) an optimal solution on the input graph, and (right) an optimal solution on the folded graph.

For a given vertex $v \in V$, we recall definitions of d_v^+ and d_v^- defined in line (18). Now we introduce some additional notation. For every vertex pair $\{a, b\} \in \binom{V}{2}$, we define vertex set $C(a, b)$ as the set of common neighbors of vertices a and b; that is, $C(a, b) \coloneqq N_G(a) \cap N_G(b)$. For every $v \in C(a, b)$, we define h_v^{ab} and indicator function $\mathbb{1}_v^{ab}$ as follows.

$$h_v^{ab} = \begin{cases} w_{av} & \text{if } |w_{av}| \leq |w_{bv}|, \\ w_{bv} & \text{otherwise.} \end{cases} \qquad \mathbb{1}_v^{ab} = \begin{cases} 1 & \text{if } w_{av} w_{bv} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

For every vertex pair $\{a, b\} \in \binom{V}{2}$, we define β_a and β_b as follows.

$$\beta_a = \sum_{v \in C(a,b)} \mathbb{1}_v^{ab} \big| w_{av} - h_v^{ab} \big|, \qquad \beta_b = \sum_{v \in C(a,b)} \mathbb{1}_v^{ab} \big| w_{bv} - h_v^{ab} \big|.$$

Finally for every vertex pair $\{a, b\} \in \binom{V}{2}$, we define $d_{ab} = d_{ab}^+ + d_{ab}^-$ with

$$d_{ab}^+ \coloneqq \sum_{v \in C(a,b)} \max\{h_v^{ab}, 0\}, \qquad d_{ab}^- \coloneqq \sum_{v \in C(a,b)} \min\{h_v^{ab}, 0\}.$$

Lemma 4 proposes a condition under which we can safely apply the folding operation on a given pair of vertices. The proof is provided in Appendix C.

Lemma 4. *Let graph $G = (V, E)$. There is an optimal solution of the max k-cut problem with vertex pair $\{a, b\} \in \binom{V}{2}$ assigned to the same partition if*

$$d_{ab}^+ - d_{ab}^- - \frac{3}{2} \min\{w_{ab}, 0\} \geq \max\{d_a^+ - d_a^- + \beta_a, d_b^+ - d_b^- + \beta_b\} - \alpha^*, \tag{22}$$

where

$$\alpha^* \coloneqq \min_{q,t} \left\{ q_2 - q_1 \,\middle|\, \begin{array}{ll} q_j = \sum_{v \in C(a,b)} h_v^{ab} t_{vj}, \ \forall j \in P; & q_1 \leq q_2 \leq \cdots \leq q_k; \\ \sum_{j \in P} t_{vj} = 1, \ \forall v \in C(a,b); & t \in \{0,1\}^{|C(a,b)| \times k} \end{array} \right\}.$$

Lemma 5 shows a mapping of feasible solutions with pairs assigned to the same partition from the input graph to the folded one.

Lemma 5. *Let $\hat{x} \in \{0,1\}^{n \times k}$ be a feasible solution of the max k-cut problem in graph G with a vertex pair $\{a, b\} \in \binom{V}{2}$ assigned to the same partition. Then there is a feasible solution $\bar{x} \in \{0,1\}^{|V(F)| \times k}$ in the folded graph F with $g(\bar{x}) = g(\hat{x})$.*

Proof. Let $c \in V(F)$ be the vertex that represents the combined vertices a and b in graph G. For any partition $j \in P$, we define $\bar{x}_{cj} \coloneqq \hat{x}_{aj} = \hat{x}_{bj}$ and $\bar{x}_{vj} \coloneqq \hat{x}_{vj}$ for every vertex $v \in V \setminus \{a, b\}$.

Let $D_{ab} = \delta(a) \cup \delta(b)$ with $\delta(v)$ be the set of incident edges of vertex $v \in V$. For any feasible point $\hat{x} \in \{0,1\}^{n \times k}$, we rewrite $g(\hat{x}) = g'_{ab}(\hat{x}) + g''_{ab}(\hat{x})$ with

$$g'_{ab}(\hat{x}) = \sum_{\{u,v\} \in D_{ab}} w_{uv} \Big[1 - \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} \Big], \qquad g''_{ab}(\hat{x}) = \sum_{\{u,v\} \in E \setminus D_{ab}} w_{uv} \Big[1 - \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} \Big].$$

For any feasible point $\bar{x} \in \{0,1\}^{|V(F)| \times k}$, we have $g(\bar{x}) = g'_c(\bar{x}) + g''_c(\bar{x})$ on the folded graph F with

$$g'_c(\bar{x}) = \sum_{v \in N_F(c)} w_{cv} \Big[1 - \sum_{j \in P} \bar{x}_{cj} \bar{x}_{vj} \Big], \qquad g''_c(\bar{x}) = g''_{ab}(\bar{x}).$$

In the folded graph F, we have $N_F(c) = \Big(N_G(a) \cup N_G(b)\Big) \setminus \{a, b\}$. Then, we have

$$\begin{aligned}
g'_{ab}(\hat{x}) &= \sum_{\{u,v\} \in D_{ab}} w_{uv} \Big[1 - \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj}\Big] && (23a)\\
&= \sum_{v \in N_G(a)} w_{av} \Big[1 - \sum_{j \in P} \hat{x}_{aj}\hat{x}_{vj}\Big] + \sum_{v \in N_G(b)} w_{bv} \Big[1 - \sum_{j \in P} \hat{x}_{bj}\hat{x}_{vj}\Big] - w_{ab}\Big[1 - \sum_{j \in P} \hat{x}_{aj}\hat{x}_{bj}\Big] && (23b)\\
&= \sum_{v \in N_G(a)} w_{av} \Big[1 - \sum_{j \in P} \hat{x}_{aj}\hat{x}_{vj}\Big] + \sum_{v \in N_G(b)} w_{bv} \Big[1 - \sum_{j \in P} \hat{x}_{bj}\hat{x}_{vj}\Big] && (23c)\\
&= \sum_{v \in N_F(c)} w_{av} \Big[1 - \sum_{j \in P} \bar{x}_{cj}\bar{x}_{vj}\Big] + \sum_{v \in N_F(c)} w_{bv} \Big[1 - \sum_{j \in P} \bar{x}_{cj}\bar{x}_{vj}\Big] && (23d)\\
&= \sum_{v \in N_F(c)} (w_{av} + w_{bv}) \Big[1 - \sum_{j \in P} \bar{x}_{cj}\bar{x}_{vj}\Big] = \sum_{v \in N_F(c)} w_{cv} \Big[1 - \sum_{j \in P} \bar{x}_{cj}\bar{x}_{vj}\Big] = g'_c(\bar{x}). && (23e)
\end{aligned}$$

Here, equality (23c) holds because $\hat{x}_{aj} = \hat{x}_{bj}$ for every partition $j \in P$ and

$$\sum_{j \in P} \hat{x}_{aj}\hat{x}_{bj} = \sum_{j \in P} \hat{x}^2_{aj} = \sum_{j \in P} \hat{x}_{aj} = 1.$$

Equality (23d) holds because $w_{av} = 0$ for every $v \in V \setminus N(a)$ and $w_{bv} = 0$ for every $v \in V \setminus N(b)$. By construction of $\bar{x}$, we also have $g''_c(\bar{x}) = g''_{ab}(\hat{x})$. Hence, $g(\bar{x}) = g(\hat{x})$ in the BQO formulation (3). □

Theorem 4 (Folding Operation)**.** *Given graph $G = (V, E)$, the folding operation on vertex pair $\{a, b\} \in \binom{V}{2}$ is safe for the max k-cut problem if inequality (41) holds.*

Proof. The proof follows by Lemmata 4 and 5. □

Remark 5 (Folding for the max cut problem)**.** *Given graph $G = (V, E)$, the folding operation on vertex pair $\{a, b\} \in \binom{V}{2}$ is safe for the max cut problem if the following condition holds.*

$$d^+_{ab} - d^-_{ab} - 2\min\{w_{ab}, 0\} \geq \max\{d^+_a - d^-_a + \beta_a, d^+_b - d^-_b + \beta_b\} - \alpha^*. \qquad (24)$$

Proof. We note that Case 2 of Lemma 5 (see Appendix C) does not happen for the max cut problem. So, condition (24) is valid. □

For every vertex pair $\{a, b\} \in \binom{V}{2}$, Example 3 shows that if condition (22) does not hold, then the folding operation is not necessarily safe.

Example 3. Consider an instance of the max cut problem with $\epsilon \in [0, 2]$ that is illustrated in Figure 7 (left). Note that the optimal objective value is 10. Here, we have $d^+_{1,3} = 2 - \epsilon$, $d^-_{1,3} = 0$, $d^+_1 = 4$, $d^-_1 = 0$, $d^+_3 = 3 - \epsilon$, $d^-_3 = -1$, and $\alpha^* = 2 - \epsilon$. Suppose that we apply the folding operation on vertices 1 and 3. Then, the folding condition of Theorem 4 can be written as follows.

$$2 - \epsilon = d^+_{1,3} - d^-_{1,3} \geq \max\left\{d^+_1 - d^-_1, d^+_3 - d^-_3\right\} - \alpha^* = \max\left\{4, 4 - \epsilon\right\} - (2 - \epsilon) = 2 + \epsilon.$$

Here, if $\epsilon = 0$, then we can safely apply the folding operation on vertices 1 and 3 by Theorem 4. However, if $\epsilon \in (0, 2]$, then Figure 7 (right) shows that the folding condition does not hold and the optimal objective value changes from 10 to $10 - \epsilon$ by applying the folding operation on vertices 1 and 3.

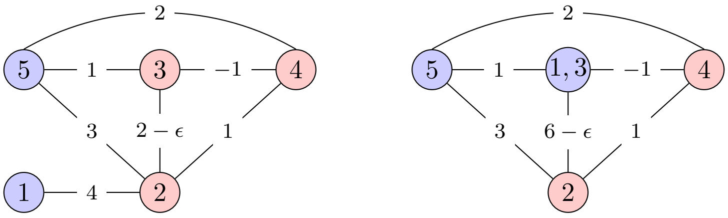


Figure 7: An instance of the max cut problem: (left) an optimal solution on the input graph, and (right) an optimal solution on the folded graph.

Corollary 2. *Let $\{a, b\} \in \binom{V}{2}$ be a pair of vertices in an instance of the max cut problem with $d^+_{ab} - d^-_{ab} - 2\min\{w_{ab}, 0\} < 0.5 \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\}$. Then, the folding operation on vertices a and b is not safe.*

Proof. It suffices to show that the condition of Theorem 4 does not hold. By the contrapositive. Suppose that the condition of Theorem 4 holds for vertex pair $\{a, b\}$; that is, we have

$$\begin{aligned}
d^+_{ab} - d^-_{ab} - 2\min\{w_{ab}, 0\} &\geq \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\} - \alpha^*, && \text{(25a)}\\
&\geq \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\} - (q_2 - q_1), && \text{(25b)}\\
&\geq \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\} - \max\{q_2 - q_1\}, && \text{(25c)}\\
&= \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\} - (d^+_{ab} - d^-_{ab}). && \text{(25d)}
\end{aligned}$$

Here, inequality (25b) holds because $\alpha^* \leq q_2 - q_1$. Equality (25d) holds because

$$0 \leq q_2 - q_1 \leq d^+_{ab} - d^-_{ab}.$$

Thus, we have

$$d^+_{ab} - d^-_{ab} - 2\min\{w_{ab}, 0\} \geq 0.5 \max\left\{d^+_a - d^-_a, d^+_b - d^-_b\right\}.$$

□

We finally note that all the preprocessing algorithms are applied in the following order: (i) peeling operation, (ii) biconnected decomposition, and (iii) folding operation. We iteratively apply these operations until none of them is operational. In this case, we solve the max k-cut problem on the reduced graph(s).

6 Computational Experiments

In this section, we provide our computational experiments on (i) mixed integer optimization models, (ii) QUBO and R-QUBO models, and (iii) preprocessing algorithms. We run the computational experiments on a machine with Dual Intel Xeon® CPU E5-2630 @ 2.20 GHz (20 cores) and 64 GB of RAM. We have developed the Python package `MaxKcut` (Fakhimi and Validi, 2022) to conduct the computational experiments. We employ `QISKIT` (IBM Research, 2021) Python Package to implement the QAOA for solving our proposed QUBO formulations. Furthermore, we use Gurobi 9.5.0 (Gurobi Optimization, LLC, 2021) to solve our mixed integer optimization models.

In the Gurobi solver, we set the number of threads and time limit (TL) to 10 and 3,600 seconds, respectively.

In our experiments, we employ three sets of instances: (i) band (G. Wang and Hijazi, 2020), (ii) spinglass (G. Wang and Hijazi, 2020), and (iii) weighted transportation network (Stabler B., 2019; STOM-Group, 2019; Salemi and Buchanan, 2021). For a given set of vertices and number of partitions k, the edge set of a band graph is defined as $E = \left\{\{i,j\} \in \binom{V}{2} : j - i \leq k+1, i < j\right\}$. In these graphs, half of the edge weights are -1, and the others are $+1$.

6.1 Experiments on mixed integer optimization models

In this section, we provide computational comparisons between the classical models explained in Section 3. In all sets of instances except band graphs, we observe that our computational results match our theoretical ones on the strength of mixed integer programming formulations of the max k-cut problem.

In our A-MILO implementation, we employ clique inequalities of Chopra and Rao (1993) for strengthening purposes. We also apply an integer separation procedure for the P-MILO implementation that takes $O(n^2 + (n-k)k^2)$. Steps of the integer separation at each integral node of the branch-and-bound algorithm are provided below.

(i) Let $\hat{z}$ be an integer solution in the branch-and-bound-node;

(ii) let $\hat{E} = \left\{\{u,v\} \in \binom{V}{2} \mid \hat{z}_{uv} = 1\right\}$;

(iii) find connected components of the induced subgraph $\hat{G} = G[\hat{E}]$;

(iv) let $\hat{S} \subseteq V(\hat{G})$ be a set of representatives of the components in the induced subgraph $\hat{G}$;

(v) let $\hat{M}$ be an ordering of the vertex set $\hat{S}$;

(vi) for any $k+1$ consecutive set of vertices in the ordering, constraint (2c) is added on-the-fly.

For the RP-MILO implementation, we recall the following procedure from G. Wang and Hijazi (2020): (i) input graph is extended to a chordal graph; (ii) binary variables z are created *only* for the chordal edge set; (iii) all maximal cliques of the chordal graph are found; and (iv) constraints (2b)–(2c) are added on-the-fly only for the maximal cliques. The integer separation procedure is similar to the P-MILO model.

Tables 1–2 show our computational results on the spinglass instances. We observe the superiority of the A-MILO and BQO formulations over P-MILO and RP-MILO models. Specifically, we observe that P-MILO solves none of the spinglass instances under a one-hour time limit. Furthermore, we see the superiority of the BQO model over the A-MILO formulation for $k \in \{3, 4\}$. This shows that our computational results match the theoretical ones for the sparse spinglass instances. On the other hand, Table 3 shows the superiority of the RP-MILO formulation on band graphs with $k \in \{3, 4\}$. We note that all band graphs are chordal and sparse. This explains why RP-MILO outperforms the other mixed integer optimization models on this set of instances. Our computational experiments show that the performance of the RP-MILO formulation deteriorates as the graph gets denser after the chordalization procedure.

Table 1: Results of MILO and BQO models for the max k-cut problem on spinglass2g instances.

instance				P-MILO		RP-MILO		A-MILO		BQO	
name	n	m	k	obj	time	obj	time	obj	time	obj	time
12×12	144	288	2	0	TL	10,215,826	9.74	10,215,826	1.30	10,215,826	**1.22**
13×13	169	338	2	0	TL	12,122,930	230.95	12,122,930	1.19	12,122,930	**1.16**
14×14	196	392	2	0	TL	14,068,479	43.21	14,068,479	**1.13**	14,068,479	1.15
15×15	225	450	2	0	TL	15,270,477	313.08	15,270,477	1.53	15,270,477	**1.36**
16×16	256	512	2	0	TL	17,641,750	462.38	17,641,750	**1.66**	17,641,750	2.53
12×12	144	288	3	2,554,241	TL	11,456,798	8.41	11,456,798	3.35	11,456,798	**1.88**
13×13	169	338	3	0	TL	13,528,955	53.23	13,528,955	10.31	13,528,955	**2.06**
14×14	196	392	3	214,110	TL	15,436,695	51.56	15,436,695	9.52	15,436,695	**2.72**
15×15	225	450	3	0	TL	1,7345,778	515.64	17,345,778	17.85	17,345,778	**3.64**
16×16	256	512	3	0	TL	19,683,323	678.26	19,683,323	16.57	19,683,323	**5.10**
12×12	144	288	4	419,382	TL	11,508,802	11.98	11,508,802	4.79	11,508,802	**2.40**
13×13	169	338	4	280,291	TL	13,581,434	48.00	13,581,434	7.27	13,581,434	**2.14**
14×14	196	392	4	0	TL	15,514,489	109.69	15,514,489	7.36	15,514,489	**2.48**
15×15	225	450	4	678,328	TL	17,464,835	161.24	17,464,835	12.47	17,464,835	**2.89**
16×16	256	512	4	1,547,889	TL	19,777,787	108.37	19,777,787	20.57	19,777,787	**3.45**

Table 2: Results of MILO and BQO models for the max k-cut problem on spinglass2pm instances.

instance				P-MILO		RP-MILO		A-MILO		BQO	
name	n	m	k	obj	time	obj	time	obj	time	obj	time
12×12	144	288	2	2	TL	104	30.42	104	**1.24**	104	1.30
13×13	169	338	2	2	TL	114	232.00	114	**1.34**	114	1.41
14×14	196	392	2	0	TL	132	TL	132	**1.47**	132	1.56
15×15	225	450	2	0	TL	144	TL	146	1.68	146	**1.62**
16×16	256	512	2	6	TL	178	2049.92	178	**1.84**	178	2.29
12×12	144	288	3	2	TL	120	15.93	120	5.72	120	**1.90**
13×13	169	338	3	2	TL	138	161.33	138	13.06	138	**6.83**
14×14	196	392	3	0	TL	161	2225.13	161	16.81	161	**4.09**
15×15	225	450	3	0	TL	179	349.98	179	36.04	179	**7.00**
16×16	256	512	3	6	TL	205	TL	211	47.79	211	**12.86**
12×12	144	288	4	11	TL	120	17.14	120	6.63	120	**1.90**
13×13	169	338	4	6	TL	139	52.90	139	10.87	139	**3.42**
14×14	196	392	4	4	TL	162	96.09	162	10.98	162	**2.87**
15×15	225	450	4	0	TL	180	254.93	180	16.02	180	**4.69**
16×16	256	512	4	6	TL	213	220.94	213	19.22	213	**4.76**

Table 3: Results of MILO and BQO models for the max k-cut problem on band instances.

instance				P-MILO		RP-MILO		A-MILO		BQO	
name	n	m	k	obj	time	obj	time	obj	time	obj	time
band050_3	50	190	3	49	76.18	49	**1.05**	49	29.01	49	2.75
band100_3	100	390	3	19	TL	99	**1.71**	99	138.32	99	8.98
band150_3	150	590	3	20	TL	150	**1.94**	150	83.60	150	15.40
band200_3	200	790	3	20	TL	199	**2.53**	199	361.09	199	21.89
band250_3	250	990	3	8	TL	249	**2.96**	249	426.11	249	39.36
band050_4	50	235	4	59	28.37	59	**1.23**	59	397.93	59	10.22
band100_4	100	485	4	25	TL	117	**2.25**	117	TL	117	456.23
band150_4	150	735	4	23	TL	175	**3.35**	174	TL	175	1,480.63
band200_4	200	985	4	24	TL	234	**5.83**	230	TL	234	2,498.83
band250_4	250	1,235	4	27	TL	292	**6.83**	288	TL	292	TL

6.2 Experiments on QUBO models

In this section, we test the performance of our QUBO models on the QISKIT quantum simulator. To alleviate the computational burden, we employ QAOA$_1$ (i.e., the QAOA with one level) for solving the proposed QUBO formulations.[1] We also use the brute-force search method to determine near-optimal values of γ and β. To do so, we divide the search space of γ and β into a 50 by 50 grid. In our quantum experiments, we set the quantum simulator to `qasm_simulator`. To construct the phase separation operator $U_P(\gamma)$, we briefly discuss Hamiltonian matrices corresponding to our QUBO formulations. For more technical details on quantum circuits, interested readers are referred to Nielsen and Chuang (2011).

Let I and Z be the Pauli matrices applied to a single qubit. Then for every vertex $v \in V$ and every partition $j \in P$, matrix H_{vj} is defined as the Hamiltonian of binary clause x_{vj}. For vertices $u, v \in V$ and partitions $i, j \in P$, matrix $H_{ui,vj}$ is defined as the Hamiltonian of binary clause $x_{ui}x_{vj}$.

$$H_{vj} = \frac{1}{2}\big(I_{vj} - Z_{vj}\big), \qquad H_{ui,vj} = \frac{1}{2^2}\big(I_{ui} - Z_{ui}\big) \otimes \big(I_{vj} - Z_{vj}\big).$$

Furthermore, let $\bar{H}_{uj,vj}$ be the Hamiltonian matrix of $(x_{vj} - x_{uj})^2$.

$$\bar{H}_{uj,vj} = \frac{1}{2}\big(I_{uj} \otimes I_{vj} - Z_{uj} \otimes Z_{vj}\big).$$

Finally, we define the following Hamiltonian matrices H_q and $H_{\bar{q}}$ corresponding to simplified versions of QUBO and R-QUBO formulations, respectively.

$$\begin{aligned} H_q &= -\sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} H_{uj,vj} - \sum_{v\in V} c_v \Big[2 \sum_{\{i,j\}\in\binom{P}{2}} H_{vi,vj} - \sum_{j\in P} H_{vj}\Big], \\ H_{\bar{q}} &= \sum_{\{u,v\}\in E} w_{uv} \Big[\sum_{j\in\bar{P}} \bar{H}_{uj,vj} - \sum_{\{i,j\}\in\binom{\bar{P}}{2}} \Big(H_{ui,vj} + H_{uj,vi}\Big)\Big] - \sum_{v\in V} c_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} H_{vi,vj}. \end{aligned}$$

The simplified versions of QUBO and R-QUBO formulations are provided in Appendix D.

[1] Finding optimal parameters of the QAOA becomes more difficult as the number of levels of the QAOA increases.

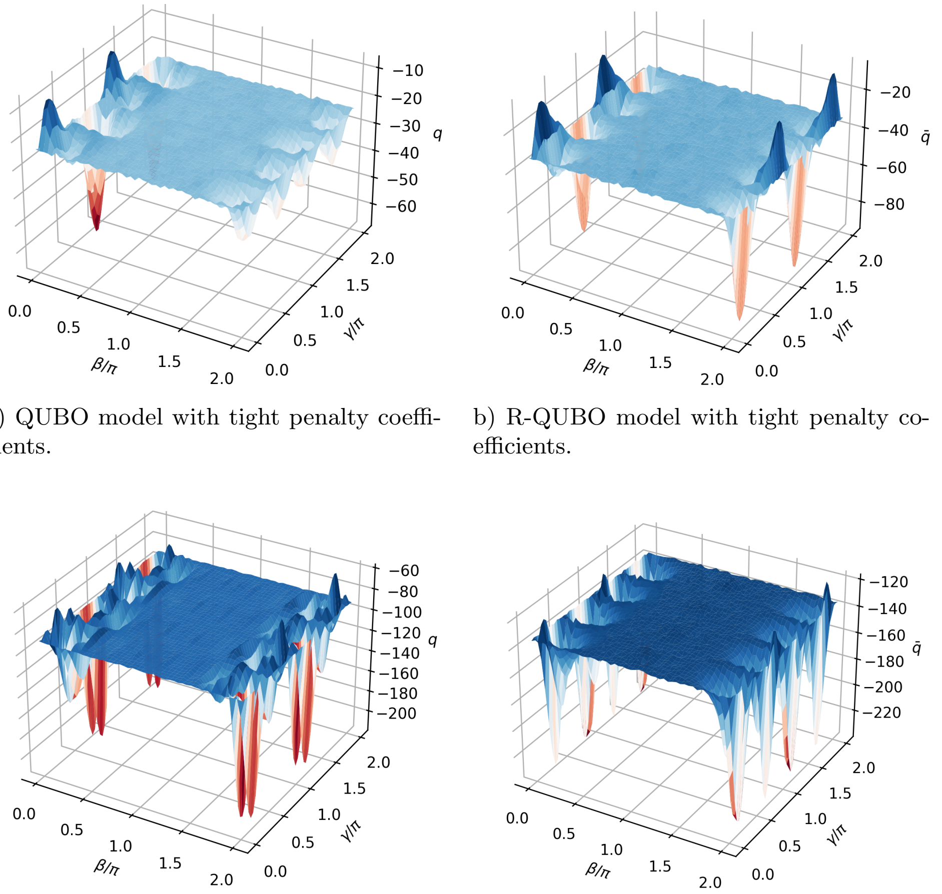
Now we provide our experimental results and analyses on (i) parameters γ and β; (ii) effect of penalty coefficients; (iii) effect of gate noise; and (iv) comparison of QUBO and R-QUBO models. We run a quantum circuit 10,000 times to calculate the expected value of the objective value in QAOA_1 (i.e., QAOA with one level). We conduct all experiments on Erdős-Rényi random graphs with density percentages 20, 40, 60 and 80; negative edge percentages 0, 40, and 80; and $k \in \{3, 4\}$. We also test the effect of tight and naïve penalty coefficients on QUBO and R-QUBO models. We refer $\max\left\{\frac{d_v^+}{k}, -\frac{d_v^-}{2}\right\}$ and $d_v^+ - d_v^-$ as tight penalty coefficients for QUBO and R-QUBO, respectively (see Theorems 2 and 3). For the sake of fairness in our computational comparisons, we propose new "naïve" penalty coefficients that are smaller than those provided in Remarks 2 and 3. By new "naïve", we mean sufficiently large penalty coefficients $d_v^+ - d_v^-$ and $k(d_v^+ - d_v^-)$ for QUBO and R-QUBO, respectively.

Analysis on γ and β. The theoretical convergence results of the QAOA hold with optimal values of γ and β. However, finding optimal γ and β is a challenging task due to the non-convexity of the expected objective value function. We employ a brute-force algorithm to find near-optimal values of γ and β for which the expected objective value function is maximized. Figure 8 illustrates the non-convex, non-linear, and periodic behavior of the expected objective values of QUBO and R-QUBO with respect to different values of γ and β. The "best" local expected objective values can be identified as the peaks in these figures. In our experiments, we use γ and β that return the highest peak. For $k = 2$, Z. Wang et al. (2018) proposed a closed-form formula for calculating the expected objective value as a function of γ and β. To the best of our knowledge, there is no closed-form formula for calculating the expected objective value when $k > 2$.

Effect of penalty coefficients. Here, we assess the effect of tight and naïve penalty coefficients. In terms of feasibility percentages, Table 4 shows that naïve penalty coefficients outperform tight ones in roughly 75% and 80% of cases for QUBO and R-QUBO models, respectively. This observation is intuitive because larger penalty coefficients increase the chance of obtaining feasible solutions. On the other hand, we observe that the expected objective value of feasible solutions improves as we move towards tight penalty coefficients. Statistically, our experiments show the superiority of tight penalty coefficients over the naïve ones in roughly 70% and 60% of cases for the QUBO and R-QUBO models, respectively. This observation can be explained by the fact that smaller penalty coefficients increase the impact of the max k-cut objective function (i.e., terms q_1 and $\bar{q}_1$ in QUBO and R-QUBO models, respectively) on the objective value of QUBO and R-QUBO formulations.

QUBO vs. R-QUBO. Here, we investigate the computational performance of QUBO and R-QUBO models using the QAOA. Table 4 shows that the R-QUBO model consistently outperforms the QUBO one in terms of the percentage of feasible solutions. Two explanations can be provided for this observation: (i) a vertex can be assigned to either no partition or multiple partitions in the QUBO while it can be assigned to at least one partition in the R-QUBO; and (ii) for every vertex, the QUBO has k assignment variables while the R-QUBO has $k - 1$ assignment variables; so, the QAOA applied on the QUBO is more likely to return an infeasible solution.

With tight penalty coefficients, we observe that the quality of feasible solutions in the QUBO is better than that of the R-QUBO in roughly 80% of cases. One can justify this observation by the fact that the values of penalty coefficients of the QUBO are less than those of the R-QUBO. This



a) QUBO model with tight penalty coefficients.

b) R-QUBO model with tight penalty coefficients.

c) QUBO model with naïve penalty coefficients.

d) R-QUBO model with naïve penalty coefficients.

Figure 8: Results of the QAOA$_1$ for the max 4-cut model to solve band8_4 instance with 8 vertices.

observation also explains why the quality of the feasible solutions of the QUBO with naïve penalty coefficients is competitive with that of the R-QUBO with tight penalty coefficients in which the penalty coefficients are the same.

Effect of penalty coefficients on noisy quantum circuits. Here, we conduct our experiments on a noisy simulator with 99% gate fidelities for single-qubit gates. In better words, the probability of obtaining a wrong output from any single-qubit gate is 0.01. In order to obtain a fair assessment of penalty coefficients under the noise effect, we run our experiments on instances with the largest circuit depth. Table 5 shows that the superiority of tight models over naïve ones is amplified in the presence of noise. We summarize our observations from Table 5 as follows:

(i) with tight coefficients under the noise effect, the percentage of feasible solutions and the expected value of the objective of feasible solutions are deteriorated by 5% and 0%, respectively;

(ii) with naïve coefficients under the noise effect, the percentage of feasible solutions and the expected value of the objective of feasible solutions are worsened by 12% and 3%, respectively.

Our observation is consistent with the explanation of Li et al. (2022) on direct dependence of the Hamiltonian simulation's noise to time. In our experiments, penalty coefficients play the role of time in a Hamiltonian simulation. Thus, increasing penalty coefficients leads to larger errors in final results.

In Table 5, we note that the simulator struggles in finding feasible solutions for the QUBO models in the presence of noise when $k = 4$. The `Qiskit` simulator behaves as follows: it calculates a state vector, and then it samples from the final state vector in the absence of noise. However, it computes a state vector for each shot in the presence of noise.

Table 4: Results of QAOA_1 for QUBO and R-QUBO models with tight and naïve penalty coefficients on a graph with 8 vertices.

instance			QUBO				R-QUBO				optimal obj
			tight		naïve		tight		naïve		
k	m	neg%	feas%	E(obj)	feas%	E(obj)	feas%	E(obj)	feas%	E(obj)	
3	7	0	1.01	4.96	3.52	4.77	50.13	4.83	55.67	4.79	7
3	7	40	2.50	2.58	4.46	2.16	42.59	2.32	52.75	2.14	5
3	7	80	1.79	−3.36	2.90	−3.41	32.27	−3.42	50.84	−3.35	1
3	12	0	1.68	8.66	6.34	8.09	68.84	8.43	80.71	8.21	12
3	12	40	3.80	3.21	5.96	2.89	59.24	3.07	83.91	2.85	6
3	12	80	2.22	−5.27	3.46	−5.34	47.08	−4.93	78.67	−5.23	0
3	18	0	0.79	12.84	9.91	12.25	73.62	12.43	81.70	12.20	16
3	18	40	3.76	4.34	11.97	4.11	69.48	4.53	76.40	4.16	10
3	18	80	2.52	−7.77	7.77	−7.68	51.21	−6.86	73.06	−7.52	0
3	23	0	2.02	15.81	1.50	13.59	76.83	15.93	94.79	15.62	20
3	23	40	5.34	5.53	17.03	4.92	75.83	5.49	91.59	4.94	11
3	23	80	3.27	−8.34	9.67	−8.57	57.75	−7.95	82.79	−8.54	0
4	7	0	0.68	5.47	0.53	5.08	24.79	4.95	21.92	5.11	7
4	7	40	0.91	2.70	0.54	2.33	18.05	2.14	18.54	2.24	5
4	7	80	0.32	−3.72	0.32	−3.72	1.12	−3.82	17.18	−3.62	1
4	12	0	1.26	9.51	4.05	9.22	57.67	8.40	61.75	8.77	12
4	12	40	1.64	3.35	3.74	3.22	32.88	2.97	59.15	2.95	6
4	12	80	0.46	−5.83	1.92	−5.85	10.51	−5.78	41.15	−5.96	0
4	18	0	0.37	13.92	3.46	13.57	64.85	12.61	32.54	12.20	18
4	18	40	1.16	5.01	4.03	4.54	41.16	4.66	35.92	4.41	11
4	18	80	1.33	−8.38	3.65	−8.72	11.92	−8.35	48.71	−8.88	0
4	23	0	1.55	17.79	0.04	18.00	44.85	17.21	32.38	17.31	22
4	23	40	3.09	6.02	2.66	5.28	56.70	5.62	12.80	4.11	11
4	23	80	0.50	−9.52	0.05	−8.20	14.99	−9.44	33.43	−9.63	0

6.3 Experiments on preprocessing algorithms

In this section, we test the preprocessing algorithms discussed in Section 5. In practice, we observe the efficiency of our proposed folding operation on two sets of existing sparse instances when $k = 2$:

Table 5: Results of QAOA_1 with one percent gate noise for QUBO and R-QUBO models with tight and naïve penalty coefficients on a graph with 8 vertices.

instance			QUBO				R-QUBO				optimal obj
			tight		naïve		tight		naïve		
k	m	neg%	feas%	E(obj)	feas%	E(obj)	feas%	E(obj)	feas%	E(obj)	
3	23	0	1.96	15.85	1.57	13.25	76.88	15.66	86.75	15.64	20
3	23	40	5.41	5.62	16.99	4.87	71.52	5.37	82.97	4.96	11
3	23	80	3.29	-8.52	10.18	-8.46	54.11	-7.96	75.50	-8.56	0
4	23	0	NA	NA	NA	NA	43.92	16.97	22.28	16.61	22
4	23	40	NA	NA	NA	NA	47.91	5.53	8.00	3.40	11
4	23	80	NA	NA	NA	NA	12.66	-9.45	25.31	-9.40	0

(i) weighted transportation networks (Stabler B., 2019; STOM-Group, 2019; Salemi and Buchanan, 2021), and (ii) weighted band graphs in which 50% of edges have weights of -1 and the others have weights of $+1$ (G. Wang and Hijazi, 2020).

Tables 6–7 provide details of our preprocessing algorithms for $k = 2$ on *non-planar* weighted transportation networks and band graphs, respectively. We note that the max cut problem is easy on planar graphs (Hadlock, 1975). In weighted transportation networks, the preprocessing algorithms reduce the number of vertices and edges, on average, by 11.83% and 8.61%, respectively. In weighted band graphs, the preprocessing algorithms reduce the number of vertices and edges, on average, by 9.03% and 13.61%, respectively. Furthermore, the folding operation is the *only* operation that works on the weighted band graphs. Figure 9 shows the decomposition tree of Austin's transportation network. By applying the folding operation at node 9 of the tree, other preprocessing algorithms, including folding, can further decompose the graph and reduce the number of vertices of the largest component from 6,911 to 6,610. We note that a tiny portion of the preprocessing time is spent solving the MILO formulation (42). For example, only 3 seconds out of 135 seconds of Chicago's preprocessing time is spent solving model (42).

Table 8 illustrates the effect of folding operation on solving the max cut problem by QAOA for the R-QUBO model with $p = 1$. In all cases, the expected value of the objective values improves after applying the folding operation. For 6 out of 17 instances, the folding operation helps the quantum simulator solve the max cut problem to optimality while they are not solved to optimality without folding.

Table 6: Preprocessing results for weighted transportation networks with $k = 2$.

instances			peel + decom.			peel + decom. + fold			% reduction	
name	n	m	n'	m'	t'	n''	m''	t''	vertex	edge
Albany	90	149	90	149	0.00	**87**	**145**	0.06	3.33	2.68
Buffalo	90	149	90	149	0.00	90	149	0.05	0.00	0.00
DC-NY-BOS	317	509	293	485	0.01	**289**	**479**	0.16	8.83	5.89
Korean	324	440	255	361	0.02	**230**	**332**	0.49	29.01	24.55
Anaheim	416	634	395	613	0.01	**368**	**583**	0.41	11.54	8.04
Barcelona	930	1,798	906	1,774	0.03	**890**	**1,743**	0.60	4.30	3.06
Rome	3,353	4,831	2,689	4,167	0.11	**2,579**	**4,044**	7.78	23.08	16.29
Austin	7,388	10,591	6,911	10,109	0.38	**6,610**	**9,783**	63.53	10.53	7.63
Chicago	12,979	20,627	11,138	18,786	0.84	**10,923**	**18,556**	134.84	15.84	10.04

Table 7: Preprocessing results for weighted band graphs with $k = 2$.

instances			peel + decom.			peel + decom. + fold			% reduction	
name	n	m	n'	m'	t'	n''	m''	t''	vertex	edge
band050_2	50	144	50	144	0.00	**44**	**121**	0.21	12.00	15.97
band100_2	100	294	100	294	0.00	**91**	**259**	0.49	9.00	11.90
band150_2	150	444	150	444	0.00	**137**	**390**	0.74	8.67	12.16
band200_2	200	594	200	594	0.01	**181**	**513**	0.99	9.50	13.64
band250_2	250	744	250	744	0.01	**235**	**637**	1.21	6.00	14.38

Table 8: Effect of the folding operation on results of the QAOA for the R-QUBO model with $p = 1$ and $k = 2$. $E(\text{obj})$ and STD(obj) represent expected value and standard deviation of objective values, respectively.

	input graph					folded graph					optimal
name	n	m	E(obj)	STD(obj)	best	n	m	E(obj)	STD(obj)	best	obj.
band17_2	17	45	4.21	1.03	16	14	33	**4.51**	1.28	16	16
band18_2	18	48	2.58	1.12	17	15	36	**4.51**	1.32	17	17
band19_2	19	51	2.39	1.09	17	15	35	**4.39**	1.36	17	17
band20_2	20	54	2.07	1.15	18	15	35	**5.96**	1.50	18	18
band21_2	21	57	4.49	1.12	20	17	41	**7.12**	1.51	20	20
band22_2	22	60	3.28	1.20	21	17	40	**7.53**	1.43	21	21
band23_2	23	63	4.46	1.24	22	18	43	**8.67**	1.49	22	22
band24_2	24	66	5.19	1.29	22	20	50	**6.38**	1.50	**24**	24
band25_2	25	69	5.97	1.29	24	20	50	**8.50**	1.57	**26**	26
band26_2	26	72	5.78	1.34	26	21	52	**8.85**	1.64	**28**	28
band27_2	27	75	4.14	1.36	24	22	56	**7.59**	1.57	**26**	26
band28_2	28	78	4.67	1.40	24	23	58	**8.02**	1.60	**28**	28
band29_2	29	81	4.85	1.48	26	25	65	**7.52**	1.65	28	29
band30_2	30	84	4.87	1.49	27	26	68	**7.14**	1.68	27	29
band31_2	31	87	5.47	1.46	26	26	67	**9.42**	1.69	29	31
band32_2	32	90	5.99	1.51	29	26	68	**10.77**	1.77	**33**	33
band33_2	33	93	–	–	–	28	73	**10.21**	1.78	31	33

7 Conclusion

This paper explores the boundaries of classical and quantum solvers for the max k-cut problem. We compare four mixed integer optimization models of the max k-cut problem analytically and computationally. We propose two quadratic unconstrained binary optimization models with tight penalty coefficients based on the BQO model discussed in the paper. We conduct a series of experiments on the QUBO models and compare them computationally with one another. For reducing the size of input graphs, we present an iterative preprocessing framework. We also propose a folding operation that uses a mixed integer optimization formulation in the preprocessing procedure. Our computational experiments illustrate the effectiveness of folding for the max cut problem on some existing sets of graphs. There is a wide range of research directions for the operations researchers who are interested in quantum computing: finding optimal values for parameters γ and β in QAOA

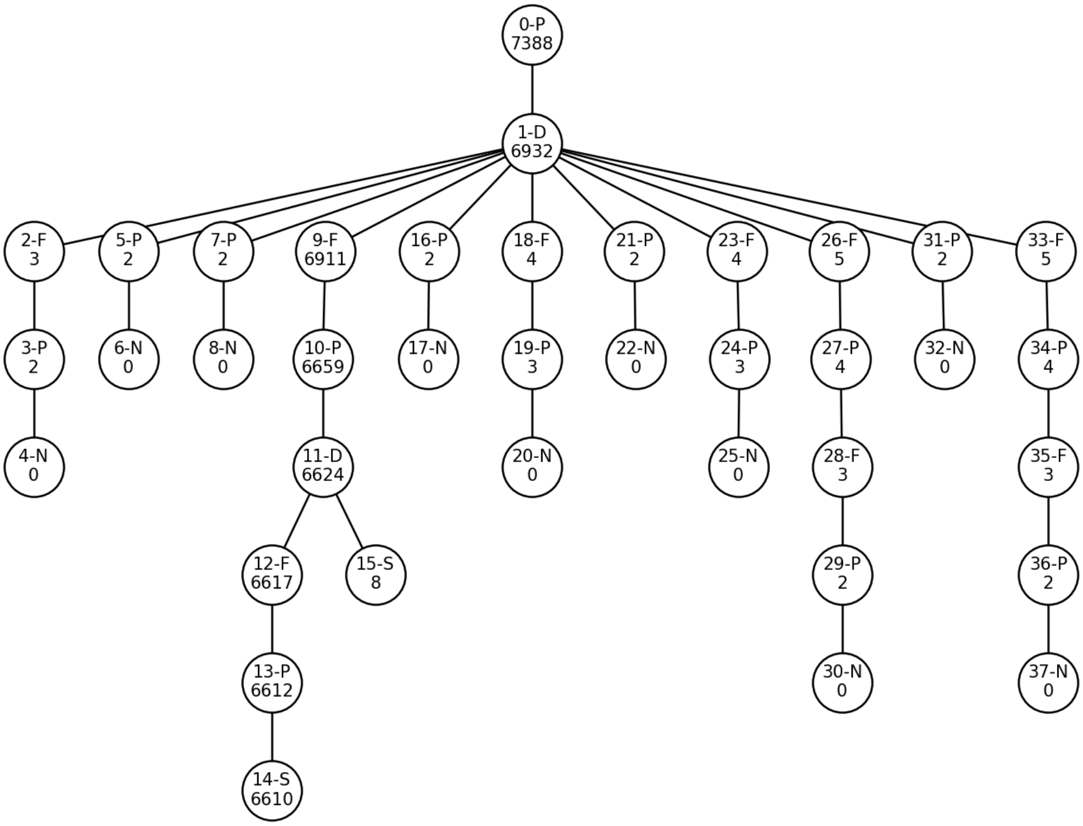


Figure 9: The decomposition tree of Austin's transportation network: here, P, D, and F denote peeling, biconnected decomposition, and folding operations, respectively. The first number denotes the order of decomposition at each node, and the last number shows the number of vertices in the corresponding subgraph. At each leaf node of the tree, a solver is called only if its status is S. The solver is not called if the status is N.

circuits, proposing efficient QUBO models for other well-known mixed integer optimization models, and identifying specific structures of the graph for which classical/quantum solver is the superior one.

Acknowledgement

This work is supported by the Defense Advanced Research Projects Agency (DARPA), ONISQ grant W911NF2010022 titled *The Quantum Computing Revolution and Optimization: Challenges and Opportunities.*

References

Achterberg, Tobias, Robert E Bixby, Zonghao Gu, Edward Rothberg, and Dieter Weninger (2020). “Presolve reductions in mixed integer programming”. *INFORMS Journal on Computing* 32.2, pp. 473–506. DOI: 10.1287/ijoc.2018.0857.

Agrawal, Paras M and Ramesh Sharda (2013). “OR forum—Quantum mechanics and human decision making”. *Operations Research* 61.1, pp. 1–16. DOI: 10.1287/opre.1120.1068.

Aharonov, Dorit, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev (2007). “Adiabatic quantum computation is equivalent to standard quantum computation”. *SIAM Journal on Computing* 37.1, pp. 166–194. DOI: 10.1137/S0097539705447323.

Alès, Zacharie and Arnaud Knippel (2020). “The k-partitioning problem: Formulations and branch-and-cut”. *Networks* 76.3, pp. 323–349. DOI: 10.1002/net.21944.

Amara, Patricia, D. Hsu, and John E. Straub (1993). “Global energy minimum searches using an approximate solution of the imaginary time Schroedinger equation”. *The Journal of Physical Chemistry* 97.25, pp. 6715–6721. DOI: 10.1021/j100127a023.

Apolloni, B., C. Carvalho, and D. de Falco (1989). “Quantum stochastic optimization”. *Stochastic Processes and their Applications* 33.2, pp. 233–244. DOI: 10.1016/0304-4149(89)90040-9.

Arute, Frank, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. (2019). “Quantum supremacy using a programmable superconducting processor”. *Nature* 574.7779, pp. 505–510. DOI: 10.1038/s41586-019-1666-5.

Barahona, Francisco, Martin Grötschel, Michael Jünger, and Gerhard Reinelt (1988). “An application of combinatorial optimization to statistical physics and circuit layout design”. *Operations Research* 36.3, pp. 493–513. DOI: 10.1287/opre.36.3.493.

Basso, Joao, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou (2021). “The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model”. *arXiv preprint arXiv:2110.14206*. URL: https://arxiv.org/abs/2110.14206.

Benioff, Paul (1980). “The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines”. *Journal of Statistical Physics* 22.5, pp. 563–591. DOI: 10.1007/BF01011339.

Borcherds, Richard (2021). *The teapot test for quantum computers.* URL: https://bit.ly/3eg4EXi.

Butenko, Sergiy (2003). “Maximum independent set and related problems, with applications”. PhD thesis. University of Florida. URL: https://bit.ly/3moSZKc.

Carlson, R. C. and George L. Nemhauser (1966). “Scheduling to minimize interaction cost”. *Operations Research* 14.1, pp. 52–58. DOI: 10.1287/opre.14.1.52.

Chopra, Sunil and M. R. Rao (1993). “The partition problem”. *Mathematical Programming* 59.1, pp. 87–115. DOI: 10.1007/BF01581239.

— (1995). “Facets of the k-partition polytope”. *Discrete Applied Mathematics* 61.1, pp. 27–48. DOI: 10.1016/0166-218X(93)E0175-X.

Dam, Edwin R van and Renata Sotirov (2016). “New bounds for the max-k-cut and chromatic number of a graph”. *Linear Algebra and its Applications* 488, pp. 216–234. DOI: 10.1016/j.laa.2015.09.043.

De Simone, Caterina, Martin Diehl, Michael Jünger, Petra Mutzel, Gerhard Reinelt, and Giovanni Rinaldi (1995). “Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm". *Journal of Statistical Physics* 80.1, pp. 487–496. DOI: 10.1007/BF02178370.

Deutsch, David and Richard Jozsa (1992). "Rapid solution of problems by quantum computation". *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907, pp. 553–558. DOI: 10.1098/rspa.1992.0167.

Dunning, Iain, Swati Gupta, and John Silberholz (2018). "What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO". *INFORMS Journal on Computing* 30.3, pp. 608–624. DOI: 10.1287/ijoc.2017.0798.

Eisenblätter, Andreas (2002). "The semidefinite relaxation of the k-partition polytope is strong". *Integer Programming and Combinatorial Optimization*. Ed. by William J. Cook and Andreas S. Schulz. Berlin, Heidelberg: Springer, pp. 273–290. DOI: 10.1007/3-540-47867-1_20.

Fairbrother, Jamie and Adam N. Letchford (2017). "Projection results for the k-partition problem". *Discrete Optimization* 26, pp. 97–111. DOI: 10.1016/j.disopt.2017.08.001.

Fairbrother, Jamie, Adam N. Letchford, and Keith Briggs (2018). "A two-level graph partitioning problem arising in mobile wireless communications". *Computational Optimization and Applications* 69.3, pp. 653–676. DOI: 10.1007/s10589-017-9967-9.

Fakhimi, Ramin and Hamidreza Validi (2022). *Max k-cut*. URL: https://github.com/qcol-lu/maxkcut.

Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann (2014). "A quantum approximate optimization algorithm". *arXiv preprint*. URL: https://arxiv.org/abs/1411.4028.

Farhi, Edward, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda (2001). "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem". *Science* 292.5516, pp. 472–475. DOI: 10.1126/science.1057726.

Farhi, Edward, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser (2000). "Quantum computation by adiabatic evolution". *arXiv preprint*. URL: https://arxiv.org/abs/quant-ph/0001106.

Farhi, Edward and Aram W. Harrow (2016). "Quantum supremacy through the quantum approximate optimization algorithm". *arXiv preprint*. URL: https://arxiv.org/abs/1602.07674.

Finnila, A.B., M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll (1994). "Quantum annealing: A new method for minimizing multidimensional functions". *Chemical Physics Letters* 219.5, pp. 343–348. DOI: 10.1016/0009-2614(94)00117-0.

Frieze, Alan and Mark Jerrum (1997). "Improved approximation algorithms for MAX k-CUT and MAX BISECTION". *Algorithmica* 18.1, pp. 67–81. DOI: 10.1007/BF02523688.

Fuchs, Franz G, Herman Øie Kolden, Niels Henrik Aase, and Giorgio Sartor (2021). "Efficient encoding of the weighted MAX k-CUT on a quantum computer using QAOA". *SN Computer Science* 2.2, pp. 1–14. DOI: 10.1007/s42979-020-00437-z.

Gross, J.L. and J. Yellen (2003). *Handbook of Graph Theory*. Discrete Mathematics and Its Applications. CRC Press. URL: https://bit.ly/3Fil4dG.

Grover, Lov K. (1996). "A fast quantum mechanical algorithm for database search". *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, pp. 212–219. DOI: 10.1145/237814.237866.

Guerreschi, Gian Giacomo and Anne Y Matsuura (2019). "QAOA for Max-Cut requires hundreds of qubits for quantum speed-up". *Scientific reports* 9.1, p. 6903. DOI: 10.1038/s41598-019-43176-9.

Gurobi Optimization, LLC (2021). *Gurobi Optimizer Reference Manual*. URL: https://gurobi.com/.

Hadfield, Stuart (2018). *Quantum Algorithms for Scientific Computing and Approximate Optimization*. URL: https://arxiv.org/pdf/1805.03265.pdf.

Hadlock, F. (1975). "Finding a Maximum Cut of a Planar Graph in Polynomial Time". *SIAM Journal on Computing* 4.3, pp. 221–225. DOI: 10.1137/0204019.

Harrigan, Matthew P., Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Daniel Eppens, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff, Steve Habegger, Alan Ho, Sabrina Hong, Trent Huang, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander N. Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Martin Leib, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Charles Neill, Florian Neukart, Murphy Yuezhen Niu, Thomas E. O'Brien, Bryan O'Gorman, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Andrea Skolik, Vadim Smelyanskiy, Doug Strain, Michael Streif, Marco Szalay, Amit Vainsencher, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Leo Zhou, Hartmut Neven, Dave Bacon, Erik Lucero, Edward Farhi, and Ryan Babbush (2021). "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor". *Nature Physics* 17.3, pp. 332–336. DOI: 10.1038/s41567-020-01105-y.

IBM Research, Qiskit community (2021). *Qiskit: An Open-source Framework for Quantum Computing*. DOI: 10.5281/zenodo.2573505.

Kadowaki, Tadashi and Hidetoshi Nishimori (1998). "Quantum annealing in the transverse Ising model". *Physical Review E* 58.5, pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355.

Li, Boxi, Shahnawaz Ahmed, Sidhant Saraogi, Neill Lambert, Franco Nori, Alexander Pitchford, and Nathan Shammah (2022). "Pulse-level noisy quantum circuits with QuTiP". *Quantum* 6, p. 630. ISSN: 2521-327X. DOI: 10.22331/q-2022-01-24-630.

Lu, Cheng and Zhibin Deng (2021). "A branch-and-bound algorithm for solving max-k-cut problem". *Journal of Global Optimization*, pp. 1–23. DOI: 10.1007/s10898-021-00999-z.

Lucas, Andrew (2014). "Ising formulations of many NP problems". *Frontiers in Physics* 2, p. 5. DOI: 10.3389/fphy.2014.00005.

Medvidović, Matija and Giuseppe Carleo (2021). "Classical variational simulation of the Quantum Approximate Optimization Algorithm". *npj Quantum Information* 7.1, p. 101. DOI: 10.1038/s41534-021-00440-z.

Méndez-Díaz, Isabel and Paula Zabala (2006). "A Branch-and-Cut algorithm for graph coloring". *Discrete Applied Mathematics* 154.5, pp. 826–847. DOI: 10.1016/j.dam.2005.05.022.

Nielsen, Michael A. and Isaac L. Chuang (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. USA: Cambridge University Press. URL: https://bit.ly/33TrYs5.

Padberg, Manfred (1989). "The boolean quadric polytope: some characteristics, facets and relatives". *Mathematical Programming* 45.1, pp. 139–172. DOI: 10.1007/BF01589101.

Papadimitriou, Christos H. and Mihalis Yannakakis (1991). "Optimization, approximation, and complexity classes". *Journal of Computer and System Sciences* 43.3, pp. 425–440. DOI: 10.1016/0022-0000(91)90023-X.

Pednault, Edwin, John Gunnels, Dmitri Maslov, and Jay Gambetta (2019). *On quantum supremacy.* URL: https://ibm.co/30RcVhi.

Preskill, John (2011). "Quantum computing and the entanglement frontier". *arXiv preprint.* URL: https://arxiv.org/abs/1203.5813.

— (2018). "Quantum Computing in the NISQ era and beyond". *Quantum* 2, p. 79. DOI: 10.22331/q-2018-08-06-79.

Quintero, Rodolfo, David Bernal, Tamás Terlaky, and Luis F. Zuluaga (2022). "Characterization of QUBO reformulations for the maximum k-colorable subgraph problem". *Quantum Information Processing* 21.3, p. 89. DOI: 10.1007/s11128-022-03421-z.

Reichardt, Ben W. (2004). "The quantum adiabatic optimization algorithm and local minima". *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing.* New York, NY, USA: Association for Computing Machinery, pp. 502–510. DOI: 10.1145/1007352.1007428.

Sack, Stefan H. and Maksym Serbyn (2021). "Quantum annealing initialization of the quantum approximate optimization algorithm". *Quantum* 5, p. 491. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-01-491.

Salemi, Hosseinali and Austin Buchanan (2021). *Implementation of solving the distance-based critical node problem.* URL: https://github.com/halisalemi/DCNP.

Savelsbergh, Martin WP (1994). "Preprocessing and probing techniques for mixed integer programming problems". *ORSA Journal on Computing* 6.4, pp. 445–454. DOI: 10.1287/ijoc.6.4.445.

Shor, P. W. (1994). "Algorithms for quantum computation: discrete logarithms and factoring". *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

Sotirov, Renata (2014). "An efficient semidefinite programming relaxation for the graph partition problem". *INFORMS Journal on Computing* 26.1, pp. 16–30. DOI: 10.1287/ijoc.1120.0542.

Stabler B. Bar-Gera H., Sall E. (2019). *Hazmat network data.* URL: https://github.com/bstabler/.

STOM-Group (2019). *Hazmat network data.* URL: https://github.com/STOM-Group/Hazmat-Network-Data.

Wang, Guanglei and Hassan Hijazi (2020). "Exploiting sparsity for the min k-partition problem". *Mathematical Programming Computation* 12.1, pp. 109–130. DOI: 10.1007/s12532-019-00165-3.

Wang, Zhihui, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel (2018). "Quantum approximate optimization algorithm for MaxCut: A fermionic view". *Physical Review A* 97 (2), p. 022304. DOI: 10.1103/PhysRevA.97.022304.

Wang, Zhihui, Nicholas C. Rubin, Jason M. Dominy, and Eleanor G. Rieffel (2020). "XY mixers: Analytical and numerical results for the quantum alternating operator ansatz". *Physical Review A* 101 (1), p. 012320. DOI: 10.1103/PhysRevA.101.012320.

Wurtz, Jonathan and Danylo Lykov (2021). "Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs". *Physical Review A* 104.5, p. 052419. DOI: 10.1103/PhysRevA.104.052419.

Zhou, Leo, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin (2020). "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices". *Physical Review X* 10 (2), p. 021067. DOI: 10.1103/PhysRevX.10.021067.

Appendix A – Tightness of Penalty Coefficients in the QUBO Model

Lemma. *Let c be a penalty vector and $\hat{x}$ be an optimal solution of the* QUBO *model* (19)*. If $c_v \geq \max\left\{d_v^+/k, -d_v^-/2\right\}$ for every vertex $v \in V$, then Algorithm 1 returns a feasible solution of* BQO *that is optimal for* QUBO*.*

Proof. Let $q(x) = q_1(x) + q_2(x)$ with

$$q_1(x) := \sum_{\{u,v\}\in E} w_{uv} - \sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} x_{uj}x_{vj}, \quad \text{and} \quad q_2(x) := -\sum_{v\in V} c_v \Big(\sum_{j\in P} x_{vj} - 1\Big)^2. \tag{26}$$

Suppose that $\hat{x}$ represents an optimal binary solution of the QUBO model in which there is a vertex $v \in V$ with multiple partitions; that is, $\sum_{j\in P} \hat{x}_{vj} > 1$. Let $t_v := \sum_{j\in P} \hat{x}_{vj}$. The following claim shows every vertex is assigned to at most two partitions in the solution represented by $\hat{x}$.

Claim 1. For every vertex $v \in V$, we have $t_v \leq 2$.

Proof. Suppose not. Then, there is a vertex $v \in V$ such that $t_v \geq 3$. Without loss of generality, we assume that $N_G(v) \neq \varnothing$; that is, $d_v^+ - d_v^- > 0$. We also define $\check{x} \in \{0,1\}^{n\times k}$ as follows: for every vertex $u \in V \setminus \{v\}$ and every partition $j \in P$, we set $\check{x}_{uj} := \hat{x}_{uj}$. Let $i \in P$ with $\hat{x}_{vi} = 1$. Then we set $\check{x}_{vi} := 0$ and $\check{x}_{vj} := \hat{x}_{vj}$ for every partition $j \in P \setminus \{i\}$. Thus, we have

$$\begin{aligned} q_1(\check{x}) - q_1(\hat{x}) &= \sum_{u\in N_G(v)} \sum_{j\in P} w_{uv}\hat{x}_{uj}\hat{x}_{vj} - \sum_{u\in N_G(v)} \sum_{j\in P} w_{uv}\check{x}_{uj}\check{x}_{vj} & \text{(27a)} \\ &= \sum_{u\in N_G^+(v)} w_{uv}\hat{x}_{ui}\hat{x}_{vi} + \sum_{u\in N_G^-(v)} w_{uv}\hat{x}_{ui}\hat{x}_{vi} & \text{(27b)} \\ &\geq 0 + d_v^-, & \text{(27c)} \end{aligned}$$

where equality (27a) holds by the definition of $q_1(\cdot)$, and inequality (27c) holds since $\sum_{u\in N_G^+(v)} w_{uv}\hat{x}_{ui}\hat{x}_{vi} \geq 0$. Finally,

$$\begin{aligned} q(\check{x}) - q(\hat{x}) &= \big[q_1(\check{x}) - q_1(\hat{x})\big] + \big[q_2(\check{x}) - q_2(\hat{x})\big] \\ &\geq d_v^- + c_v\Big[(t_v - 1)^2 - (t_v - 2)^2\Big] \\ &= d_v^- + c_v(2t_v - 3) \\ &\geq d_v^- + \max\left\{\frac{d_v^+}{k}, -\frac{d_v^-}{2}\right\}(2t_v - 3) \\ &\geq \begin{cases} d_v^- - \frac{d_v^-}{2}(2t_v - 3) = d_v^-(2.5 - t_v) > 0 & \text{if } d_v^- < 0, \\ \frac{d_v^+}{k}(2t_v - 3) > 0 & \text{if } d_v^- = 0. \end{cases} \end{aligned}$$

Here, the first inequality holds by inequality (27c). The second inequality holds by the assumption of the lemma. Finally, the last strict inequalities hold because $d_v^+ - d_v^- > 0$ and $t_v \geq 3$ by the assumption. This contradicts the fact that $\hat{x}$ is an optimal solution of (19). Hence, $t_v \leq 2$. ■

Recall $E_\ell = \big\{\{u,v\} \in E^- \mid \{u,v\} \cap C_\ell \neq \varnothing\big\}$ from line 8 of Algorithm 1. Furthermore, we have $I_1 = \cup_{\ell=1}^r C_\ell$ with r be the number of C_ℓ sets defined in line 7 of the algorithm. For every partition-based class $\ell \in [r]$, we define E'_ℓ and E''_ℓ as edges with exactly one endpoint in E_ℓ and both endpoints in E_ℓ, respectively.

$$E'_\ell = \big\{\{u,v\} \in E_\ell \mid \{u,v\} \not\subseteq C_\ell\big\}, \quad \text{and} \quad E''_\ell = \big\{\{u,v\} \in E_\ell \mid \{u,v\} \subseteq C_\ell\big\}, \tag{28}$$

where $E_\ell = E'_\ell \cup E''_\ell$ and $E'_\ell \cap E''_\ell = \varnothing$. Note that $P_\ell = \{j_1, j_2\}$ when $\hat{x}$ is an optimal solution because we already proved that $|P_\ell| \leq 2$. We also define edge sets $\hat{E}'_\ell \subseteq E'_\ell$ and $\tilde{E}'_\ell \subseteq E'_\ell$ with endpoints assigned to partitions j_1 and j_2, respectively.

$$\hat{E}'_\ell \coloneqq \big\{\{u,v\} \in E'_\ell \mid \hat{x}_{uj_1} = \hat{x}_{vj_1}\big\}, \text{ and } \tilde{E}'_\ell \coloneqq \big\{\{u,v\} \in E'_\ell \mid \hat{x}_{uj_2} = \hat{x}_{vj_2}\big\}, \tag{29}$$

where $\hat{E}'_\ell \cap \tilde{E}'_\ell = \varnothing$ and $\hat{E}'_\ell \cup \tilde{E}'_\ell \subseteq E'_\ell$. We note that $\hat{E}'_\ell \cup \tilde{E}'_\ell$ contains edges whose endpoints belong to exactly one common partition. Without loss of generality, suppose that

$$\sum_{\{u,v\} \in \hat{E}'_\ell} w_{uv} \geq \sum_{\{u,v\} \in \tilde{E}'_\ell} w_{uv}. \tag{30}$$

Figure 10 illustrates a solution of the QUBO model for the max 3-cut problem with its associated sets defined above.

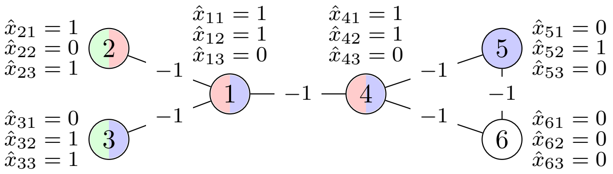


Figure 10: A solution of the QUBO model for the max 3-cut problem with (i) $C_1 = \{1,4\}$, $C_2 = \{2\}$, and $C_3 = \{3\}$; (ii) $P_1 = \{1,2\}$, $P_2 = \{1,3\}$, and $P_3 = \{2,3\}$, (iii) $E_1 = \big\{\{1,2\},\{1,3\},\{1,4\},\{4,5\},\{4,6\}\big\}$, $E_2 = \big\{\{1,2\}\big\}$, and $E_3 = \big\{\{1,3\}\big\}$, (iv) $E'_1 = \big\{\{1,2\},\{1,3\},\{4,5\},\{4,6\}\big\}$, $E''_1 = \big\{\{1,4\}\big\}$, and (v) $\hat{E}'_1 = \big\{\{1,2\}\big\}$, and $\tilde{E}'_1 = \big\{\{1,3\},\{4,5\}\big\}$.

Let $\tilde{x}$ be the output of Algorithm 1 up to line 16. Now we provide a lower bound for $q_1(\tilde{x}) - q_1(\hat{x})$.

Claim 2. $q_1(\tilde{x}) - q_1(\hat{x}) \geq \sum_{\ell \in [r]} \Big[\sum_{\{u,v\} \in \hat{E}'_\ell} w_{uv} + \sum_{\{u,v\} \in E''_\ell} w_{uv}\Big]$.

Proof. For ease of notation, we define $b_{uv} \coloneqq \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj} - \sum_{j \in P} \tilde{x}_{uj}\tilde{x}_{vj}$ for every edge $\{u,v\} \in E$.

$$
\begin{align}
q_1(\tilde{x}) - q_1(\hat{x}) &= \sum_{\{u,v\}\in E}\sum_{j\in P} w_{uv}\hat{x}_{uj}\hat{x}_{vj} - \sum_{\{u,v\}\in E}\sum_{j\in P} w_{uv}\tilde{x}_{uj}\tilde{x}_{vj} \tag{31a}\\
&= \sum_{\{u,v\}\in E} w_{uv}b_{uv} \tag{31b}\\
&= \sum_{\{u,v\}\in E^-} w_{uv}b_{uv} + \sum_{\{u,v\}\in E^+} w_{uv}b_{uv} \tag{31c}\\
&\geq \sum_{\{u,v\}\in E^-} w_{uv}b_{uv} + 0 \tag{31d}\\
&\geq \sum_{\ell\in[r]}\sum_{\{u,v\}\in E_\ell} w_{uv}b_{uv} + \sum_{E^-\setminus(E_1\cup\cdots\cup E_r)} w_{uv}b_{uv} \tag{31e}\\
&= \sum_{\ell\in[r]}\sum_{\{u,v\}\in E_\ell} w_{uv}b_{uv} + 0 \tag{31f}\\
&= \sum_{\ell\in[r]}\Big[\sum_{\{u,v\}\in E'_\ell} w_{uv}b_{uv} + \sum_{\{u,v\}\in E''_\ell} w_{uv}b_{uv}\Big] \tag{31g}\\
&= \sum_{\ell\in[r]}\Big[\sum_{\{u,v\}\in \tilde{E}'_\ell\cup\hat{E}'_\ell} w_{uv}b_{uv} + \sum_{\{u,v\}\in E'_\ell\setminus(\tilde{E}'_\ell\cup\hat{E}'_\ell)} w_{uv}b_{uv} + \sum_{\{u,v\}\in E''_\ell} w_{uv}b_{uv}\Big] \tag{31h}\\
&= \sum_{\ell\in[r]}\Big[\sum_{\{u,v\}\in \tilde{E}'_\ell} w_{uv}b_{uv} + \sum_{\{u,v\}\in \hat{E}'_\ell} w_{uv}b_{uv} + 0 + \sum_{\{u,v\}\in E''_\ell} w_{uv}b_{uv}\Big] \tag{31i}\\
&= \sum_{\ell\in[r]}\Big[0 + \sum_{\{u,v\}\in \hat{E}'_\ell} w_{uv} + \sum_{\{u,v\}\in E''_\ell} w_{uv}\Big]. \tag{31j}
\end{align}
$$

Here, inequality (31d) holds because (i) $\hat{x} \geq \tilde{x}$ implies that $\sum_{j\in P}\hat{x}_{uj}\hat{x}_{vj} - \sum_{j\in P}\tilde{x}_{uj}\tilde{x}_{vj} \geq 0$, and (ii) $w_{uv} \geq 0$ for every edge $\{u,v\} \in E^+$. Inequality (31e) holds because we might have $\ell_1 \in [r]$ and $\ell_2 \in [r]$ for which $E_{\ell_1} \cap E_{\ell_2} \neq \varnothing$; that is, some negative edges can be double counted in the first term of the inequality. For example, see Figure 10 in which (i) $E_1 \cap E_2 = \{1,2\}$ and (ii) $E^- \setminus (E_1 \cup E_2 \cup E_3) = \{5,6\}$. Equality (31f) holds because $b_{uv} = 0$ for every edge $\{u,v\} \in E^- \setminus (E_1 \cup \cdots \cup E_r)$. Note that $\hat{x}_{uj} = \tilde{x}_{uj}$ and $\hat{x}_{vj} = \tilde{x}_{vj}$ for every edge $\{u,v\} \in E^- \setminus (E_1 \cup \cdots \cup E_r)$ and every partition $j \in P$. Equality (31g) holds by definitions (28). Equality (31h) holds by definitions (29). Equality (31i) holds because (i) $\tilde{E}'_\ell \cap \hat{E}'_\ell = \varnothing$, and (ii) $b_{uv} = 0$ for every edge $\{u,v\} \in E'_\ell \setminus (\tilde{E}'_\ell \cup \hat{E}'_\ell)$. Note that $\sum_{j\in P}\hat{x}_{uj}\hat{x}_{vj} = \sum_{j\in P}\tilde{x}_{uj}\tilde{x}_{vj} = 0$ for every edge $\{u,v\} \in E'_\ell \setminus (\tilde{E}'_\ell \cup \hat{E}'_\ell)$. Finally, equality (31j) holds because the algorithm implies that for every $\ell \in [r]$, we have the following cases.

(i) $b_{uv} = \sum_{j\in P}\hat{x}_{uj}\hat{x}_{vj} - \sum_{j\in P}\tilde{x}_{uj}\tilde{x}_{vj} = 1 - 1 = 0$ for every $\{u,v\} \in \tilde{E}'_\ell$,

(ii) $b_{uv} = \sum_{j\in P}\hat{x}_{uj}\hat{x}_{vj} - \sum_{j\in P}\tilde{x}_{uj}\tilde{x}_{vj} = 1 - 0 = 1$ for every $\{u,v\} \in \hat{E}'_\ell$, and

(iii) $b_{uv} = \sum_{j\in P}\hat{x}_{uj}\hat{x}_{vj} - \sum_{j\in P}\tilde{x}_{uj}\tilde{x}_{vj} = 2 - 1 = 1$ for every $\{u,v\} \in E''_\ell$.

■

Furthermore, we provide a lower bound for $q_2(\tilde{x}) - q_2(\hat{x})$.

Claim 3. $q_2(\tilde{x}) - q_2(\hat{x}) \geq -\sum_{\ell\in[r]} 0.5\Big[\sum_{\{u,v\}\in E'_\ell} w_{uv} + 2\sum_{\{u,v\}\in E''_\ell} w_{uv}\Big]$.

Proof. We have

$$\begin{aligned}
q_2(\tilde{x}) - q_2(\hat{x}) &= -\sum_{v\in V} c_v\Big[\Big(\sum_{j\in P}\tilde{x}_{vj} - 1\Big)^2 - \Big(\sum_{j\in P}\hat{x}_{vj} - 1\Big)^2\Big] && \text{(32a)}\\
&= -\sum_{v\in I_1} c_v\Big[\Big(\sum_{j\in P}\tilde{x}_{vj} - 1\Big)^2 - \Big(\sum_{j\in P}\hat{x}_{vj} - 1\Big)^2\Big] && \text{(32b)}\\
&= \sum_{v\in I_1} c_v && \text{(32c)}\\
&\geq -\sum_{\ell\in[r]}\sum_{v\in C_\ell} 0.5 d_v^- && \text{(32d)}\\
&= -\sum_{\ell\in[r]} 0.5\Big[\sum_{\{u,v\}\in E'_\ell} w_{uv} + 2\sum_{\{u,v\}\in E''_\ell} w_{uv}\Big]. && \text{(32e)}
\end{aligned}$$

Here, equality (32b) holds because $\hat{x}_{vj} = \tilde{x}_{vj}$ for every vertex $v \in V \setminus I_1$ and every partition $j \in P$. Equality (32c) holds because we have $\sum_{j\in P}\hat{x}_{vj} = 2$ for every vertex $v \in I_1$. Inequality (32d) holds because by assumption we have $c_v \geq -0.5d_v^-$ for every vertex $v \in I_1$. Equality (32e) holds by definitions (28). ■

The following claim shows that $\tilde{x}$ is an optimal solution of the QUBO formulation (19).

Claim 4. $q(\tilde{x}) - q(\hat{x}) \geq 0$.

Proof. By Claims 2 and 3, we have

$$\begin{aligned}
q(\tilde{x}) - q(\hat{x}) &\geq \sum_{\ell\in[r]}\Big[\sum_{\{u,v\}\in \hat{E}'_\ell} w_{uv} + \sum_{\{u,v\}\in E''_\ell} w_{uv}\Big] - \sum_{\ell\in[r]} 0.5\Big[\sum_{\{u,v\}\in E'_\ell} w_{uv} + 2\sum_{\{u,v\}\in E''_\ell} w_{uv}\Big]\\
&= 0.5\sum_{\ell\in[r]}\Big[\sum_{\{u,v\}\in \hat{E}'_\ell} w_{uv} - \sum_{\{u,v\}\in \tilde{E}'_\ell} w_{uv} - \sum_{\{u,v\}\in E'_\ell\setminus(\hat{E}'_\ell\cup\tilde{E}'_\ell)} w_{uv}\Big] \geq 0.
\end{aligned}$$

Here, the last inequality holds by (i) inequality (30), and (ii) the fact that $w_{uv} < 0$ for every edge $\{u, v\} \in E'_\ell\setminus(\hat{E}'_\ell\cup\tilde{E}'_\ell)$. Hence, Algorithm 1 returns an optimal solution of the QUBO formulation (1) such that every vertex is assigned to at most one partition. ■

Let $\bar{x}$ be the output of Algorithm 1.

Claim 5. $q(\bar{x}) - q(\tilde{x}) \geq 0$.

Proof. Suppose that $\tilde{x}$ represents an optimal solution in which a vertex $v \in V$ is assigned to no partition; that is, $\sum_{j\in P}\tilde{x}_{vj} = 0$. By line 17 of Algorithm 1, let s be a partition with minimum value of $\sum_{u\in N_G(v)} w_{uv}\tilde{x}_{uj}$ among all $j \in P$. By line 18 of the algorithm, we have $\bar{x}_{vs} = 1$. By definitions (26), we have

$$q_1(\bar{x}) - q_1(\tilde{x}) = -\sum_{u\in N_G(v)} w_{uv}\tilde{x}_{us}, \quad \text{and} \quad q_2(\bar{x}) - q_2(\tilde{x}) = c_v. \tag{33}$$

Then, we have the following cases.

(i) $\sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us} \geq 0$. Hence,

$$c_v \geq \frac{d_v^+}{k} \geq \min_{j \in P} \left\{ \sum_{u \in N_G^+(v)} w_{uv} \tilde{x}_{uj} \right\} \geq \min_{j \in P} \left\{ \sum_{u \in N_G(v)} w_{uv} \tilde{x}_{uj} \right\} = \sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us}. \tag{34}$$

Here, the first inequality holds by assumption. The second inequality holds because (i) the minimum value of a set of numbers is less than or equal to their average, and (ii) every vertex is assigned to at most one partition. The last inequality holds because for every vertex $u \in N_G^-(v)$, we have $w_{uv} < 0$. The last equality holds by the definition of s. So, we have $c_v - \sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us} \geq 0$. Then, $q(\bar{x}) - q(\tilde{x}) \geq 0$ by lines (33) and (34).

(ii) $\sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us} < 0$. By line (33), it follows that $q(\bar{x}) - q(\tilde{x}) \geq 0$.

■

By Claims 4 and 5, $q(\bar{x}) - q(\hat{x}) \geq 0$. So, $\bar{x}$ is also an optimal solution of the QUBO model (19). □

Appendix B – Tightness of Penalty Coefficients in the R-QUBO Model

Lemma. *Let c be a penalty vector and $\hat{x}$ be an optimal solution of the* R-QUBO *model* (21)*. If $c_v \geq d_v^+ - d_v^-$ for every vertex $v \in V$, then Algorithm 2 returns a feasible solution of* R-BQO *that is optimal for* R-QUBO*.*

Proof. Let $\hat{x} \in \{0,1\}^{n\times(k-1)}$ be a binary solution of the R-QUBO formulation with some vertices assigned to more than one partition. Assume that $\bar{x}$ is a feasible solution of the R-BQO model returned by Algorithm 2. We have $I = \cup_{\ell=1}^{r} C_\ell$ with r be the number of C_ℓ sets defined in line 5 of Algorithm 2. We note that for every $\ell \in [r]$, (i) vertex set C_ℓ is a set of vertices that are assigned to the same multiple partitions, and (ii) without loss of generality assume every vertex $u \in C_\ell$ is assigned to a less or equal number of partitions than that of any vertex $v \in C_{\ell+1}$. For any x and any edge $\{u,v\} \in E$, we define function $h_{uv}(x)$ as follows.

$$h_{uv}(x) = w_{uv}\Big[\sum_{j\in\bar{P}} x_{uj}x_{vj} + \Big(1 - \sum_{j\in\bar{P}} x_{uj}\Big)\Big(1 - \sum_{j\in\bar{P}} x_{vj}\Big)\Big]$$

For every vertex $v \in V$, we define $t_v := \sum_{j\in\bar{P}} \hat{x}_{vj}$. For every edge $\{u,v\} \in E$, (i) let $t_{uv} := \sum_{j\in\bar{P}} \hat{x}_{uj}\hat{x}_{vj}$; and (ii) terms $h_{uv}(\hat{x})$ and $h_{uv}(\bar{x})$ are simplified as follows.

$$h_{uv}(\hat{x}) = w_{uv}\big[t_{uv} + (1-t_u)(1-t_v)\big], \tag{35a}$$

$$h_{uv}(\bar{x}) = \begin{cases} w_{uv}\sum_{j\in\bar{P}} \bar{x}_{uj}\bar{x}_{vj} & t_u + t_v \neq 0 \\ w_{uv} & t_u + t_v = 0 \end{cases} \leq \begin{cases} w_{uv}\min\{t_{uv},1\} & t_u + t_v \neq 0, w_{uv} > 0 \\ 0 & t_u + t_v \neq 0, w_{uv} \leq 0 \\ w_{uv} & t_u + t_v = 0. \end{cases} \tag{35b}$$

Here, inequality (35b) holds because (i) $\bar{x}$ is feasible for the max k-cut problem and (ii) $\bar{x} \leq \hat{x}$. Now, we recall the positive edge set as $E^+ := \big\{\{u,v\} \in E \mid w_{uv} > 0\big\}$.

Claim 6. $\sum_{\{u,v\}\in E^+}\big[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\big] \geq -\sum_{v\in I}\sum_{u\in N_G^+(v)} w_{uv}(t_v - 1)\frac{t_v}{2}$.

Proof. For every edge $\{u,v\} \in E^+$, we bound $h_{uv}(\hat{x}) - h_{uv}(\bar{x})$ by lines (35) as follows:

(i) if $\{u,v\} \subseteq I$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq 0$ because $h_{uv}(\bar{x}) \leq w_{uv}\min\{t_{uv},1\} \leq w_{uv}$, and

$$h_{uv}(\hat{x}) = w_{uv}\big[t_{uv} + (t_u - 1)(t_v - 1)\big] \geq w_{uv}\big[0 + (2-1)(2-1)\big] = w_{uv}.$$

(ii) if $\min\{t_u,t_v\} = 0$ and $\max\{t_u,t_v\} > 1$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) = w_{uv}\big(1 - \max\{t_u,t_v\}\big)$ because $t_{uv} = \min\{t_u,t_v\} = 0$, $h_{uv}(\bar{x}) \leq w_{uv}\min\{t_{uv},1\} = 0$, and

$$h_{uv}(\hat{x}) = w_{uv}\big[0 + (1 - \max\{t_u,t_v\})(1 - \min\{t_u,t_v\})\big] = w_{uv}\big(1 - \max\{t_u,t_v\}\big).$$

(iii) if $\min\{t_u,t_v\} = 1$ and $\max\{t_u,t_v\} > 1$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}\big(1 - \max\{t_u,t_v\}\big)$ because $h_{uv}(\bar{x}) \leq w_{uv}\min\{t_{uv},1\} \leq w_{uv}t_{uv}$, and

$$h_{uv}(\hat{x}) = w_{uv}\big[t_{uv} + (1 - \max\{t_u,t_v\})(1 - \min\{t_u,t_v\})\big] = w_{uv}t_{uv}.$$

We also have $w_{uv} \geq 0$ and $\max\{t_u,t_v\} > 1$, thus

$$h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq 0 \geq w_{uv}\big(1 - \max\{t_u,t_v\}\big).$$

(iv) if $\{u,v\} \cap I = \varnothing$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) = 0$ because $h_{uv}(\hat{x}) = h_{uv}(\bar{x})$ by Algorithm 2.

Now we define $\bar{E}^+$ as the set of positive edges with exactly one endpoint assigned to multiple partitions (i.e., $\bar{E}^+ \coloneqq \big\{\{u,v\} \in E^+ \mid |\{u,v\} \cap I| = 1\big\}$). Hence, we have

$$\begin{aligned}
\sum_{\{u,v\} \in E^+} \big[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\big] &\geq \sum_{\{u,v\} \in \bar{E}^+} w_{uv}(1 - \max\{t_u, t_v\}) && \text{(36a)}\\
&= \sum_{v \in I} \sum_{u \in N_G^+(v) \setminus I} w_{uv}(1 - \max\{t_u, t_v\}) && \text{(36b)}\\
&\geq \sum_{v \in I} \Bigg[\sum_{u \in N_G^+(v) \setminus I} w_{uv}(1 - t_v) + \sum_{u \in N_G^+(v) \cap I} w_{uv}(1 - t_v) \Bigg] && \text{(36c)}\\
&= -\sum_{v \in I} \sum_{u \in N_G^+(v)} w_{uv}(t_v - 1) && \text{(36d)}\\
&\geq -\sum_{v \in I} \sum_{u \in N_G^+(v)} w_{uv}(t_v - 1)\frac{t_v}{2}. && \text{(36e)}
\end{aligned}$$

Here, inequality (36a) holds by items (i)–(iv). Inequality (36c) holds because (i) for every vertex $v \in I$ we have $t_v \geq 2$, and (ii) $w_{uv} \geq 0$ for every edge $\{u,v\} \in E^+$. Inequality (36e) holds because for every vertex $v \in I$ we have $t_v \geq 2$. ■

We recall the negative edge set as $E^- \coloneqq \big\{\{u,v\} \in E \mid w_{uv} < 0\big\}$. For every $\ell \in [r]$, we define E_ℓ^- as follows. We note that each edge set E_ℓ^- is defined as the incident edges with negative weights corresponding to each vertex set C_ℓ such that E_ℓ^-s are mutually exclusive.

$$E_\ell^- \coloneqq \Big\{\{u,v\} \in E^- \;\Big|\{u,v\} \cap C_\ell \neq \varnothing, \{u,v\} \cap C_{\ell'} = \varnothing,\; \ell' < \ell, \forall \ell' \in [r]\Big\}.$$

In other words, for every $\ell \in [r]$ we define E_ℓ^- as the set of negative edges with (i) at least one endpoint, say u, in C_ℓ, (ii) no endpoint belongs to set $C_1 \cup \cdots \cup C_{\ell-1}$ and (iii) the other endpoint is assigned to a less or equal number of partitions than that of vertex u. For every $\ell \in [r]$, recall that (i) vertex set C_ℓ is defined by line 5 of Algorithm 2, and (ii) for every vertex $v \in C_\ell$ partition set $\bar{P}_\ell$ is the set of partitions to which vertex v is assigned (see line 7 of Algorithm 2). We also partition edge set E_ℓ^- to negative edge sets $E_{\ell 1}^-$, $E_{\ell 2}^-$, $E_{\ell 3}^-$, and $E_{\ell 4}^-$ as follows.

$$\begin{aligned}
E_{\ell 1}^- &\coloneqq \big\{\{u,v\} \in E_\ell^- \mid \{u,v\} \cap C_{\ell'} \neq \varnothing, \{u,v\} \cap C_\ell \neq \varnothing, \text{ and } \bar{P}_{\ell'} \subset \bar{P}_\ell\big\},\\
E_{\ell 2}^- &\coloneqq \big\{\{u,v\} \in E_\ell^- \mid \{u,v\} \cap C_{\ell'} \neq \varnothing, \{u,v\} \cap C_\ell \neq \varnothing, \text{ and } \bar{P}_{\ell'} \not\subset \bar{P}_\ell\big\},\\
E_{\ell 3}^- &\coloneqq \big\{\{u,v\} \in E_\ell^- \mid \{u,v\} \subseteq C_\ell\big\},\\
E_{\ell 4}^- &\coloneqq \big\{\{u,v\} \in E_\ell^- \mid \{u,v\} \cap (V \setminus I) \neq \varnothing\big\}.
\end{aligned}$$

Here, set $E_{\ell 1}^-$ represents the set of negative edges with (i) both endpoints assigned to multiple partitions and (ii) the set of assigned partitions of one endpoint is a proper subset of the assigned partitions of the other endpoint. For example in Figure 11, $E_{11}^- = \big\{\{1,2\},\{1,3\}\big\}$ and $E_{21}^- = E_{31}^- = \varnothing$. Set $E_{\ell 2}^-$ is the set of negative edges with (i) both endpoints assigned to multiple partitions, (ii) one endpoint is assigned to partition set $\bar{P}_\ell$ and the other endpoint is assigned to a partition

$j \in \bar{P}_{\ell'} \setminus \bar{P}_{\ell}$ such that $|\bar{P}_{\ell'}| \leq |\bar{P}_{\ell}|$. In Figure 11, $E^-_{12} = \varnothing$, $E^-_{22} = \big\{\{2,3\}\big\}$ and $E^-_{32} = \varnothing$. Set $E^-_{\ell 3}$ is the set of negative edges such that both endpoints are assigned to a partition set with a size of at least 2. In Figure 11, $E^-_{13} = \big\{\{1,4\}\big\}$ and $E^-_{23} = E^-_{33} = \varnothing$. Set $E^-_{\ell 4}$ is the set of negative edges with exactly one endpoint assigned to multiple partitions. In Figure 11, $E^-_{14} = \big\{\{4,5\},\{4,6\}\big\}$ and $E^-_{24} = E^-_{34} = \varnothing$.

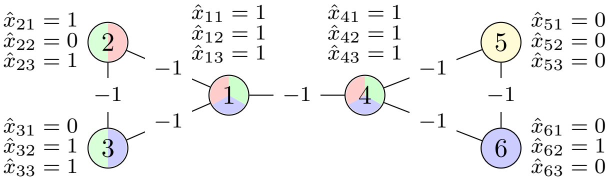


Figure 11: A solution of the R-QUBO model for the max 4-cut problem with (i) $C_1 = \{1,4\}$, $C_2 = \{2\}$, and $C_3 = \{3\}$; (ii) $\bar{P}_1 = \{1,2,3\}$, $\bar{P}_2 = \{1,3\}$, and $\bar{P}_3 = \{2,3\}$, (iii) $E^-_1 = \big\{\{1,2\},\{1,3\},\{1,4\},\{4,5\},\{4,6\}\big\}$, $E^-_2 = \big\{\{2,3\}\big\}$, and $E^-_3 = \varnothing$.

Claim 7. $\sum_{\{u,v\}\in E^-} \big[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\big]$ is bounded below by

$$\sum_{\ell\in[r]} \Bigg[\sum_{\{u,v\}\in E^-_{\ell 1}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\} + 1\big) + \sum_{\{u,v\}\in E^-_{\ell 2}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\}\big) + \sum_{\{u,v\}\in E^-_{\ell 3}} w_{uv} t_{uv}(t_{uv} - 1) + \sum_{\{u,v\}\in E^-_{\ell 4}} w_{uv} \Bigg]. \tag{37}$$

Proof. For every edge $\{u,v\} \in E^-$, we bound $h_{uv}(\hat{x}) - h_{uv}(\bar{x})$ by lines (35) as follows:

(i) if edge $\{u,v\} \in E^-_{\ell 1}$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}\big(t_u t_v - \max\{t_u, t_v\} + 1\big)$ because $t_{uv} = \min\{t_u, t_v\}$ and

$$h_{uv}(\hat{x}) = w_{uv}\big(t_{uv} + t_u t_v - t_u - t_v + 1\big) = w_{uv}\big(t_u t_v - \max\{t_u, t_v\} + 1\big), \quad \text{and} \quad h_{uv}(\bar{x}) \leq 0;$$

(ii) if edge $\{u,v\} \in E^-_{\ell 2}$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}\big(t_u t_v - \max\{t_u, t_v\}\big)$ because $t_{uv} \leq \min\{t_u, t_v\} - 1$ and

$$h_{uv}(\hat{x}) = w_{uv}\big(t_{uv} + t_u t_v - t_u - t_v + 1\big) \geq w_{uv}\big(t_u t_v - \max\{t_u, t_v\}\big), \quad \text{and} \quad h_{uv}(\bar{x}) \leq 0;$$

(iii) if edge $\{u,v\} \in E^-_{\ell 3}$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) = w_{uv} t_{uv}(t_{uv} - 1)$ because we assign both endpoints of $\{u,v\}$ to the same partition by Algorithm 2 and we have $t_{uv} = t_u = t_v$ and

$$h_{uv}(\hat{x}) = w_{uv}\big(t_{uv} + t_u t_v - t_u - t_v + 1\big) = w_{uv}\big(t_{uv}^2 - t_{uv} + 1\big), \quad \text{and} \quad h_{uv}(\bar{x}) = w_{uv};$$

(iv) if edge $\{u,v\} \in E^-_{\ell 4}$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}$ since $t_{uv} \leq 1$ and $h_{uv}(\bar{x}) \leq 0$ and

$$\begin{aligned} h_{uv}(\hat{x}) &= w_{uv}\big(t_{uv} + t_u t_v - t_u - t_v + 1\big) \\ &= w_{uv}\big(t_{uv} + t_u t_v - \min\{t_u, t_v\} - \max\{t_u, t_v\} + 1\big) \\ &= \begin{cases} w_{uv}(1 - \max\{t_u, t_v\}) & \text{if } \min\{t_u, t_v\} = 0, \\ w_{uv} t_{uv} & \text{if } \min\{t_u, t_v\} = 1, \end{cases} \\ &\geq w_{uv}. \end{aligned}$$

(v) if $\{u,v\} \in E^- \setminus \cup_{\ell \in [r]} E^-_\ell$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) = 0$ because $h_{uv}(\hat{x}) = h_{uv}(\bar{x})$ by Algorithm 2.

By items (i)–(v), $\sum_{\{u,v\} \in E^-} \big[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\big]$ is bounded below by (37). ■

Let $\bar{q}(x) = \bar{q}_1(x) + \bar{q}_2(x)$ with

$$\bar{q}_1(x) := \sum_{\{u,v\} \in E} w_{uv}\Big[1 - \sum_{j \in \bar{P}} x_{uj} x_{vj} - \Big(1 - \sum_{j \in \bar{P}} x_{uj}\Big)\Big(1 - \sum_{j \in \bar{P}} x_{vj}\Big)\Big] \tag{38a}$$

$$\bar{q}_2(x) := -\sum_{v \in V} c_v \sum_{\{i,j\} \in \binom{\bar{P}}{2}} x_{vi} x_{vj}. \tag{38b}$$

Now we show that the R-QUBO objective value of $\bar{x}$ is greater than or equal to that of $\hat{x}$.

Claim 8. $\bar{q}(\bar{x}) - \bar{q}(\hat{x}) \geq 0$.

Proof. We can bound $\bar{q}_1(\bar{x}) - \bar{q}_1(\hat{x})$ as follows (i) because $\bar{q}_1(\bar{x}) - \bar{q}_1(\hat{x}) = \sum_{\{u,v\} \in E} \big[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\big]$, and (ii) by Claims 6 and 7.

$$\begin{aligned} \bar{q}_1(\bar{x}) - \bar{q}_1(\hat{x}) \geq \sum_{\ell \in [r]} \Bigg[&\sum_{\{u,v\} \in E^-_{\ell 1}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\} + 1\big) + \sum_{\{u,v\} \in E^-_{\ell 2}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\}\big) \\ &+ \sum_{\{u,v\} \in E^-_{\ell 3}} w_{uv} t_{uv}(t_{uv} - 1) + \sum_{\{u,v\} \in E^-_{\ell 4}} w_{uv} \Bigg] - \sum_{v \in I} \sum_{u \in N^+_G(v)} w_{uv}(t_v - 1)\frac{t_v}{2}. \end{aligned} \tag{39}$$

Furthermore, we have the following arguments for $\bar{q}_2(\bar{x}) - \bar{q}_2(\hat{x})$.

$$
\begin{align}
\bar{q}_2(\bar{x}) - \bar{q}_2(\hat{x}) = & \sum_{v\in I} \binom{t_v}{2} \Bigg[\sum_{u\in N_G^+(v)} w_{uv} - \sum_{u\in N_G^-(v)} w_{uv} \Bigg] \tag{40a}\\
= & \sum_{v\in I} \binom{t_v}{2} \sum_{u\in N_G^+(v)} w_{uv} - \sum_{\ell\in[r]} \sum_{v\in C_\ell} \binom{t_v}{2} \Bigg[\sum_{u\in N_G^-(v)\cap\big(I\setminus C_\ell\big)} w_{uv} + \sum_{u\in N_G^-(v)\cap C_\ell} w_{uv} \nonumber\\
& + \sum_{u\in N_G^-(v)\setminus I} w_{uv} \Bigg] \tag{40b}\\
= & \sum_{v\in I} \binom{t_v}{2} \sum_{u\in N_G^+(v)} w_{uv} - \sum_{\ell\in[r]} \Bigg[\sum_{\{u,v\}\in E^-_{\ell 1}\cup E^-_{\ell 2}} w_{uv} \Bigg[\binom{t_u}{2} + \binom{t_v}{2} \Bigg] \nonumber\\
& + \sum_{\{u,v\}\in E^-_{\ell 3}} w_{uv} t_{uv}(t_{uv}-1) + \sum_{\{u,v\}\in E^-_{\ell 4}} w_{uv} \max\Bigg\{ \binom{t_u}{2}, \binom{t_v}{2} \Bigg\} \Bigg] \tag{40c}\\
\geq & \sum_{v\in I} \binom{t_v}{2} \sum_{u\in N_G^+(v)} w_{uv} - \sum_{\ell\in[r]} \Bigg[\sum_{\{u,v\}\in E^-_{\ell 1}\cup E^-_{\ell 2}} w_{uv} \Bigg[\binom{t_u}{2} + \binom{t_v}{2} \Bigg] \nonumber\\
& + \sum_{\{u,v\}\in E^-_{\ell 3}} w_{uv} t_{uv}(t_{uv}-1) + \sum_{\{u,v\}\in E^-_{\ell 4}} w_{uv} \Bigg]. \tag{40d}
\end{align}
$$

Here, equality (40c) holds by definitions of the partitions of E^-_ℓ. Recall that edge sets E^-_ℓs are mutually exclusive. Furthermore, as we shift from the vertex-based summation in equality (40b) to the edge-based summation in equality (40c), we need to consider the corresponding vertex coefficients $\binom{t_u}{2} + \binom{t_v}{2}$ for a given edge $\{u, v\} \in E^-$. Inequality (40d) holds because $\binom{t_v}{2} \geq 1$ for every vertex $v \in I$; so, $\max\left\{\binom{t_u}{2}, \binom{t_v}{2}\right\} \geq 1$ for every $\ell \in [r]$ and every edge $\{u, v\} \in E^-_{\ell 4}$.

By inequalities (39) and (40d), we have

$$\begin{aligned}
\bar{q}(\bar{x}) - \bar{q}(\hat{x}) \geq & \sum_{\ell \in [r]} \Bigg[\sum_{\{u,v\} \in E^-_{\ell 1}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\} + 1\big) + \sum_{\{u,v\} \in E^-_{\ell 2}} w_{uv}\big(t_u t_v - \max\{t_u, t_v\}\big) \Bigg] \\
& - \sum_{\ell \in [r]} \sum_{\{u,v\} \in E^-_{\ell 1} \cup E^-_{\ell 2}} w_{uv} \left[\binom{t_u}{2} + \binom{t_v}{2} \right] \\
= & \sum_{\ell \in [r]} \sum_{\{u,v\} \in E^-_{\ell 1}} \frac{w_{uv}}{2} \Big[2\big(t_u t_v - \max\{t_u, t_v\} + 1\big) - t_u(t_u - 1) - t_v(t_v - 1) \Big] \\
& + \sum_{\ell \in [r]} \sum_{\{u,v\} \in E^-_{\ell 2}} \frac{w_{uv}}{2} \Big[2\big(t_u t_v - \max\{t_u, t_v\}\big) - t_u(t_u - 1) - t_v(t_v - 1) \Big] \\
= & - \sum_{\ell \in [r]} \sum_{\{u,v\} \in E^-_{\ell 1}} \frac{w_{uv}}{2} \Big[(t_u - t_v)^2 + 2\max\{t_u, t_v\} - t_u - t_v - 2 \Big] \\
& - \sum_{\ell \in [r]} \sum_{\{u,v\} \in E^-_{\ell 2}} \frac{w_{uv}}{2} \Big[(t_u - t_v)^2 + 2\max\{t_u, t_v\} - t_u - t_v \Big] \geq 0.
\end{aligned}$$

Here, the last inequality holds because for every $\ell \in [r]$ and every $\{u, v\} \in E^-_{\ell 1}$, we have (i) $(t_u - t_v)^2 \geq 1$ and (ii) $\max\{t_u, t_v\} \geq \min\{t_u, t_v\} + 1$. In other words, we have

$$(t_u - t_v)^2 + 2\max\{t_u, t_v\} - t_u - t_v - 2 \geq 1 + (\max\{t_u, t_v\} + \min\{t_u, t_v\} + 1) - t_u - t_v - 2 = 0.$$

■

This finishes the proof. □

Appendix C – Folding Operation

Lemma. *Let graph $G = (V, E)$. There is an optimal solution of the max k-cut problem with vertex pair $\{a, b\} \in \binom{V}{2}$ assigned to the same partition if*

$$d^+_{ab} - d^-_{ab} - \frac{3}{2}\min\{w_{ab}, 0\} \geq \max\{d^+_a - d^-_a + \beta_a, d^+_b - d^-_b + \beta_b\} - \alpha^*, \tag{41}$$

where

$$\alpha^* := \min_{q,t}\left\{ q_2 - q_1 \,\middle|\, \begin{array}{ll} q_j = \sum_{v \in C(a,b)} h^{ab}_v t_{vj},\ \forall j \in P; & q_1 \leq q_2 \leq \cdots \leq q_k; \\ \sum_{j \in P} t_{vj} = 1,\ \forall v \in C(a,b); & t \in \{0,1\}^{|C(a,b)| \times k} \end{array} \right\}. \tag{42}$$

Proof. Let point $\hat{x} \in \{0,1\}^{n \times k}$ be a feasible solution of the max k-cut problem. We are to show that there exists a feasible solution $\bar{x} \in \{0,1\}^{n \times k}$ such that (i) $\bar{x}_{aj} = \bar{x}_{bj}$ for every partition $j \in P$, and (ii) its objective value is greater than or equal to that of $\hat{x}$. Let $D_{ab} = \delta(a) \cup \delta(b)$, where $\delta(v)$ represents the set of incident edges of vertex $v \in V$. For any point $x \in \{0,1\}^{n \times k}$, we rewrite $g(x) = g'_{ab}(x) + g''_{ab}(x)$ with

$$g'_{ab}(x) = \sum_{\{u,v\} \in D_{ab}} w_{uv}\Big(1 - \sum_{j \in P} x_{uj}x_{vj}\Big), \qquad g''_{ab}(x) = \sum_{\{u,v\} \in E \setminus D_{ab}} w_{uv}\Big(1 - \sum_{j \in P} x_{uj}x_{vj}\Big).$$

For every partition $j \in P$, we define $q_j := \sum_{v \in C(a,b)} h^{ab}_v \hat{x}_{vj}$. Without loss of generality, assume that $q_1 \leq q_2 \leq \cdots \leq q_k$. We define $\bar{x}_{a1} = \bar{x}_{b1} = 1$. For every vertex $v \in V \setminus \{a, b\}$ and every partition $j \in P$, we define $\bar{x}_{vj} := \hat{x}_{vj}$. We note that if $\hat{x}_{aj} = \hat{x}_{bj}$ for every $j \in P$, then we are done. Also $g''_{ab}(\hat{x}) = g''_{ab}(\bar{x})$ holds for any vertex pair $\{a, b\} \in \binom{V}{2}$.

Now we consider the following cases in which vertices a and b are assigned to different partitions in solution $\hat{x}$. For every vertex $v \in V$, we also define $d_v := d^+_v + d^-_v$.

Case 1. Assume that (i) vertex b is assigned to partition 1 in solution $\hat{x}$, and (ii) for every vertex $v \in C(a,b)$, we have $w_{av} = w_{bv} = h^{ab}_v$. So, $\beta_a = \beta_b = 0$. We bound $\sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{j \in P} \bar{x}_{aj}\bar{x}_{vj}\Big)$ as follows.

$$\begin{aligned}
\sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{j \in P} \bar{x}_{aj}\bar{x}_{vj}\Big) &= \sum_{v \in N_G(a)} w_{av}(1 - \bar{x}_{v1}) && \text{(43a)}\\
&= \sum_{v \in N_G(a)} w_{av} - \sum_{v \in N_G(a)} w_{av}\bar{x}_{v1} && \text{(43b)}\\
&= d_a - \sum_{v \in C(a,b)} w_{av}\bar{x}_{v1} - \sum_{v \in N_G(a) \setminus (C(a,b) \cup \{b\})} w_{av}\bar{x}_{v1} - w_{ab}\bar{x}_{b1} && \text{(43c)}\\
&= d_a - \sum_{v \in C(a,b)} w_{av}\hat{x}_{v1} - \sum_{v \in N_G(a) \setminus (C(a,b) \cup \{b\})} w_{av}\bar{x}_{v1} - w_{ab}\bar{x}_{b1} && \text{(43d)}\\
&\geq d_a - q_1 - \big(d^+_a - d^+_{ab} - \max\{w_{ab}, 0\}\big) - w_{ab} && \text{(43e)}\\
&= d^+_{ab} - q_1 + d^-_a - \min\{w_{ab}, 0\}. && \text{(43f)}
\end{aligned}$$

Here, equality (43a) holds because $\bar{x}_{aj} = 0$ for every $j \in P \setminus \{1\}$. Equality (43c) holds by the definition of d_a. Equality (43d) holds by the definition of q_1. Inequality (43e) holds because

$$\begin{aligned}\sum_{v \in N_G(a) \setminus (C(a,b) \cup \{b\})} w_{av} \bar{x}_{v1} &\leq \sum_{v \in N_G^+(a) \setminus (C(a,b) \cup \{b\})} w_{av} \bar{x}_{v1} \\ &\leq \sum_{v \in N_G^+(a) \setminus (C(a,b) \cup \{b\})} w_{av} \\ &= d_a^+ - d_{ab}^+ - \max\{w_{ab}, 0\}.\end{aligned}$$

We know that vertex a is assigned to a partition other than 1, say $j \in P \setminus \{1\}$ in solution $\hat{x}$. Now we provide an upper bound for $\sum_{v \in N_G(a)} w_{av} \Big(1 - \sum_{i \in P} \hat{x}_{ai} \hat{x}_{vi}\Big)$ as follows.

$$\begin{aligned}\sum_{v \in N_G(a)} w_{av} \Big(1 - \sum_{i \in P} \hat{x}_{ai} \hat{x}_{vi}\Big) &= \sum_{v \in N_G(a)} w_{av}(1 - \hat{x}_{vj}) &\text{(44a)} \\ &= \sum_{v \in N_G(a)} w_{av} - \sum_{v \in N_G(a)} w_{av} \hat{x}_{vj} &\text{(44b)} \\ &= d_a - \sum_{v \in C(a,b)} w_{av} \hat{x}_{vj} - \sum_{v \in N_G(a) \setminus (C(a,b) \cup \{b\})} w_{av} \hat{x}_{vj} - w_{ab} \hat{x}_{bj} &\text{(44c)} \\ &\leq d_a - q_2 - \big(d_a^- - d_{ab}^- - \min\{w_{ab}, 0\}\big) &\text{(44d)} \\ &= d_a^+ - q_2 + d_{ab}^- + \min\{w_{ab}, 0\}. &\text{(44e)}\end{aligned}$$

Here, equality (44a) holds because $\hat{x}_{ai} = 0$ for every $i \in P \setminus \{j\}$. Equality (44c) holds by the definition of d_a. Inequality (44d) holds because $\sum_{v \in C(a,b)} w_{av} \hat{x}_{vj} \geq q_2$ and

$$\begin{aligned}\sum_{v \in N_G(a) \setminus (C(a,b) \cup \{b\})} w_{av} \hat{x}_{vj} &\geq \sum_{v \in N_G^-(a) \setminus (C(a,b) \cup \{b\})} w_{av} \hat{x}_{vj} \\ &\geq \sum_{v \in N_G^-(a) \setminus (C(a,b) \cup \{b\})} w_{av} \\ &= d_a^- - d_{ab}^- - \min\{w_{ab}, 0\}.\end{aligned}$$

By the theorem's assumption, we also have

$$d_{ab}^+ - d_{ab}^- - 2\min\{w_{ab}, 0\} \geq d_{ab}^+ - d_{ab}^- - \frac{3}{2}\min\{w_{ab}, 0\} \geq d_a^+ - d_a^- - \alpha^* \geq d_a^+ - d_a^- - (q_2 - q_1). \quad (45)$$

Here, the second inequality follows from condition (42) and the last inequality holds by the fact that $-\alpha^* \geq -(q_2 - q_1)$. Thus, we have

$$\begin{aligned}\sum_{v \in N_G(a)} w_{av} \Big(1 - \sum_{j \in P} \bar{x}_{aj} \bar{x}_{vj}\Big) &\geq d_{ab}^+ - q_1 + d_a^- - \min\{w_{ab}, 0\} \\ &\geq d_a^+ - q_2 + d_{ab}^- + \min\{w_{ab}, 0\} \\ &\geq \sum_{v \in N_G(a)} w_{av} \Big(1 - \sum_{j \in P} \hat{x}_{aj} \hat{x}_{vj}\Big). \quad (46)\end{aligned}$$

Here, the first inequality follows from line (43f). The second inequality follows from inequality (45). The last inequality follows from line (44e). Since (i) $g''_{ab}(\hat{x}) = g''_{ab}(\bar{x})$ and (ii) by inequality (46), we have $g(\bar{x}) \geq g(\hat{x})$. We finally note that a similar argument holds when vertex a is assigned to partition 1 in solution $\hat{x}$.

Case 2. Assume that (i) neither vertex a nor b is assigned to partition 1 in solution $\hat{x}$, and (ii) for any vertex $v \in C(a,b)$, we have $w_{av} = w_{bv} = h_v^{ab}$. We provide a lower bound for $g'_{ab}(\bar{x})$ as follows.

$$\begin{aligned}
g'_{ab}(\bar{x}) &= \sum_{\{u,v\}\in D_{ab}} w_{uv}\Big(1 - \sum_{j\in P} \bar{x}_{uj}\bar{x}_{vj}\Big) && \text{(47a)}\\
&= \sum_{v\in N_G(a)} w_{av}(1-\bar{x}_{v1}) + \sum_{v\in N_G(b)} w_{bv}(1-\bar{x}_{v1}) - w_{ab}\Big(1 - \sum_{j\in P}\bar{x}_{aj}\bar{x}_{bj}\Big) && \text{(47b)}\\
&= \sum_{v\in N_G(a)} w_{av} - \sum_{v\in N_G(a)} w_{av}\bar{x}_{v1} + \sum_{v\in N_G(b)} w_{bv} - \sum_{v\in N_G(b)} w_{bv}\bar{x}_{v1} - 0 && \text{(47c)}\\
&= d_a - \sum_{v\in C(a,b)} w_{av}\bar{x}_{v1} - \sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\bar{x}_{v1} - w_{ab}\bar{x}_{a1}\\
&\quad + d_b - \sum_{v\in C(a,b)} w_{bv}\bar{x}_{v1} - \sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\bar{x}_{v1} - w_{ab}\bar{x}_{b1} && \text{(47d)}\\
&= d_a - 2q_1 - \sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\bar{x}_{v1} + d_b - \sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\bar{x}_{v1} - 2w_{ab} && \text{(47e)}\\
&\geq d_a - 2q_1 - \big(d_a^+ - d_{ab}^+ - \max\{w_{ab},0\}\big) + d_b - \big(d_b^+ - d_{ab}^+ - \max\{w_{ab},0\}\big) - 2w_{ab} && \text{(47f)}\\
&= 2d_{ab}^+ - 2q_1 + d_a^- + d_b^- - 2\min\{w_{ab},0\}. && \text{(47g)}
\end{aligned}$$

Here, equality (47b) holds because $\bar{x}_{aj} = \bar{x}_{bj} = 0$ for every partition $j \in P \setminus \{1\}$. Equality (47d) holds by the definition of d_a. Equality (47e) holds (i) by the definition of q_1, and (ii) because $\bar{x}_{a1} = \bar{x}_{b1} = 1$. Inequality (47f) holds because

$$\begin{aligned}
\sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\bar{x}_{v1} &\leq \sum_{v\in N_G^+(a)\setminus(C(a,b)\cup\{b\})} w_{av}\bar{x}_{v1}\\
&\leq \sum_{v\in N_G^+(a)\setminus(C(a,b)\cup\{b\})} w_{av}\\
&= d_a^+ - d_{ab}^+ - \max\{w_{ab},0\}.
\end{aligned}$$

Similar bound holds for $\sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\bar{x}_{v1}$.

Now we provide an upper bound for $g'(\hat{x})$. We note that vertices a and b are assigned to partitions $j \in P \setminus \{1\}$ and $j' \in P \setminus \{1\}$, respectively.

$$
\begin{aligned}
g'(\hat{x}) &= \sum_{\{u,v\}\in D_{ab}} w_{uv}\Big(1 - \sum_{i\in P} \hat{x}_{ui}\hat{x}_{vi}\Big) && \text{(48a)}\\
&= \sum_{v\in N_G(a)} w_{av}(1-\hat{x}_{vj}) + \sum_{v\in N_G(b)} w_{bv}(1-\hat{x}_{vj'}) - w_{ab}\Big(1 - \sum_{i\in P}\hat{x}_{ai}\hat{x}_{bi}\Big) && \text{(48b)}\\
&= \sum_{v\in N_G(a)} w_{av} - \sum_{v\in N_G(a)} w_{av}\hat{x}_{vj} + \sum_{v\in N_G(b)} w_{bv} - \sum_{v\in N_G(b)} w_{bv}\hat{x}_{vj'} - w_{ab} && \text{(48c)}\\
&= d_a - \sum_{v\in C(a,b)} w_{av}\hat{x}_{vj} - \sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\hat{x}_{vj}\\
&\quad + d_b - \sum_{v\in C(a,b)} w_{bv}\hat{x}_{vj'} - \sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\hat{x}_{vj'} - w_{ab} && \text{(48d)}\\
&\le d_a - q_2 - \sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\hat{x}_{vj} + d_b - q_2 - \sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\hat{x}_{vj'} - w_{ab} && \text{(48e)}\\
&\le d_a - q_2 - \big(d_a^- - d_{ab}^- - \min\{w_{ab},0\}\big) + d_b - q_2 - \big(d_b^- - d_{ab}^- - \min\{w_{ab},0\}\big) - w_{ab} && \text{(48f)}\\
&= d_a^+ + d_b^+ - 2q_2 + 2d_{ab}^- + 2\min\{w_{ab},0\} - w_{ab}. && \text{(48g)}
\end{aligned}
$$

Here, equality (48d) holds by the definitions of d_a and d_b. Inequality (48e) holds by inequalities $\sum_{v\in C(a,b)} w_{av}\hat{x}_{vj} \ge q_2$ and $\sum_{v\in C(a,b)} w_{bv}\hat{x}_{vj} \ge q_2$. Inequality (48f) holds because

$$
\begin{aligned}
\sum_{v\in N_G(a)\setminus(C(a,b)\cup\{b\})} w_{av}\hat{x}_{vj} &\ge \sum_{v\in N_G^-(a)\setminus(C(a,b)\cup\{b\})} w_{av}\hat{x}_{vj}\\
&\ge \sum_{v\in N_G^-(a)\setminus(C(a,b)\cup\{b\})} w_{av}\\
&= d_a^- - d_{ab}^- - \min\{w_{ab},0\}.
\end{aligned}
$$

Similar bound holds for $\sum_{v\in N_G(b)\setminus(C(a,b)\cup\{a\})} w_{bv}\hat{x}_{vj'}$. We also have

$$
\begin{aligned}
d_{ab}^+ - d_{ab}^- - \frac{3}{2}\min\{w_{ab},0\} &\ge d_a^+ - d_a^- - \alpha^* \ge d_a^+ - d_a^- - (q_2 - q_1) && \text{(49a)}\\
d_{ab}^+ - d_{ab}^- - \frac{3}{2}\min\{w_{ab},0\} &\ge d_b^+ - d_b^- - \alpha^* \ge d_b^+ - d_b^- - (q_2 - q_1). && \text{(49b)}
\end{aligned}
$$

Here, the first inequalities follow from condition (42) and the last inequalities hold by the fact that $-\alpha^* \ge -(q_2 - q_1)$[2]. By adding inequalities (49a) and (49b), we have

$$
2(d_{ab}^+ - d_{ab}^-) - 3\min\{w_{ab},0\} \ge d_a^+ - d_a^- + d_b^+ - d_b^- - 2(q_2 - q_1). \tag{50}
$$

We first rewrite inequality (50) as follows.

$$
2d_{ab}^+ - 2q_1 + d_a^- + d_b^- - 2\min\{w_{ab},0\} \ge d_a^+ + d_b^+ - 2q_2 + 2d_{ab}^- + \min\{w_{ab},0\}. \tag{51}
$$

[2] Although the optimal objective value of the continuous relaxation of the MILO formulation (42) is a valid lower bound for $q_2 - q_1$, it is always zero.

If $w_{ab} < 0$, we have

$$\begin{aligned}\sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{j \in P} \bar{x}_{aj}\bar{x}_{vj}\Big) &\geq 2d^+_{ab} - 2q_1 + d^-_a + d^-_b - 2\min\{w_{ab}, 0\} \\ &\geq d^+_a + d^+_b - 2q_2 + 2d^-_{ab} + \min\{w_{ab}, 0\} \\ &= d^+_a + d^+_b - 2q_2 + 2d^-_{ab} + 2\min\{w_{ab}, 0\} - w_{ab} \\ &\geq \sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{i \in P} \hat{x}_{ai}\hat{x}_{vi}\Big).\end{aligned}$$

Here, the first inequality follows from line (47g). The second inequality follows from inequality (51). The last inequality follows from line (48g).

If $w_{ab} \geq 0$, we have

$$\begin{aligned}\sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{j \in P} \bar{x}_{aj}\bar{x}_{vj}\Big) &\geq 2d^+_{ab} - 2q_1 + d^-_a + d^-_b \\ &\geq d^+_a + d^+_b - 2q_2 + 2d^-_{ab} \\ &\geq d^+_a + d^+_b - 2q_2 + 2d^-_{ab} - w_{ab} \\ &\geq \sum_{v \in N_G(a)} w_{av}\Big(1 - \sum_{i \in P} \hat{x}_{ai}\hat{x}_{vi}\Big).\end{aligned}$$

Here, the inequalities follow from the similar argument provided for the case $w_{ab} < 0$. We note that $\min\{w_{ab}, 0\} = 0$ when $w_{ab} \geq 0$.

Case 3. Assume that there is a vertex $\bar{v} \in C(a, b)$ with $|w_{a\bar{v}}| \neq |w_{b\bar{v}}|$. Without loss of generality, suppose that $|w_{a\bar{v}}| > |w_{b\bar{v}}|$. Next, we construct graph $\tilde{G} = (\tilde{V}, \tilde{E})$ with $\tilde{V} := V \cup \{\bar{u}\}$ and $\tilde{E} := E \cup \{a, \bar{u}\}$. Now, we define edge weights of graph $\tilde{G}$ as follows: $\tilde{w}_{a\bar{v}} := w_{b\bar{v}}$, $\tilde{w}_{a\bar{u}} := w_{a\bar{v}} - w_{b\bar{v}}$, and $\tilde{w}_{uv} := w_{uv}$ for every edge $\{u, v\} \in E \setminus \{a, \bar{v}\}$. By (i) $\tilde{C}_{ab} = C_{ab}$ and (ii) the definition of edge weights $\tilde{w}$, $\tilde{h}^{ab}_v = h^{ab}_v$ for every vertex $v \in \tilde{C}_{ab}$. Thus, we have (i) $\tilde{d}^+_{ab} = d^+_{ab}$ and $\tilde{d}^-_{ab} = d^-_{ab}$, (ii) $\tilde{\beta}_a = \tilde{\beta}_b = 0$, and (iii) $\tilde{\alpha} = \alpha^*$ with $\tilde{\alpha}$ be the optimal objective value of (42) on graph $\tilde{G}$.

We first show that there exists a solution $x' \in \{0, 1\}^{(n+1)\times k}$ in graph $\tilde{G}$ such that its associated objective value is equal to that of the $\hat{x} \in \{0, 1\}^{n\times k}$ in graph G. For any vertex $v \in \tilde{V} \setminus \{\bar{u}\}$ and every partition $j \in P$, we set $x'_{vj} := \hat{x}_{vj}$. For every partition $j \in P$, we set $x'_{\bar{u}j} := \hat{x}_{\bar{v}j}$. So, we have

$$\begin{aligned}g(x') &= \sum_{\{u,v\} \in \tilde{E}} \tilde{w}_{uv}\Big(1 - \sum_{j \in P} x'_{uj}x'_{vj}\Big) \\ &= \sum_{\{u,v\} \in \tilde{E} \setminus (\{a,\bar{u}\} \cup \{a,\bar{v}\})} \tilde{w}_{uv}\Big(1 - \sum_{j \in P} x'_{uj}x'_{vj}\Big) + \tilde{w}_{a\bar{u}}\Big(1 - \sum_{j \in P} x'_{aj}x'_{\bar{u}j}\Big) + \tilde{w}_{a\bar{v}}\Big(1 - \sum_{j \in P} x'_{aj}x'_{\bar{v}j}\Big) \\ &= \sum_{\{u,v\} \in \tilde{E} \setminus (\{a,\bar{u}\} \cup \{a,\bar{v}\})} w_{uv}\Big(1 - \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj}\Big) + (w_{a\bar{v}} - w_{b\bar{v}} + w_{b\bar{v}})\Big(1 - \sum_{j \in P} \hat{x}_{aj}\hat{x}_{\bar{v}j}\Big) = g(\hat{x}).\end{aligned}$$

Here, the third equality holds because $\tilde{w}_{a\bar{u}} = w_{a\bar{v}} - w_{b\bar{v}}$.

Now, we consider the following cases.

(i) $w_{a\bar{v}}w_{b\bar{v}} \geq 0$. Then, we have $\beta_a = \beta_b = \tilde{\beta}_a = \tilde{\beta}_b = 0$. Thus, we have

$$\begin{aligned}\tilde{d}^+_{ab} - \tilde{d}^-_{ab} - \frac{3}{2}\min\{\tilde{w}_{ab}, 0\} &= d^+_{ab} - d^-_{ab} - \frac{3}{2}\min\{w_{ab}, 0\} \\ &\geq \max\{d^+_a - d^-_a, d^+_b - d^-_b\} - \alpha^* \\ &= \max\{\tilde{d}^+_a - \tilde{d}^-_a, \tilde{d}^+_b - \tilde{d}^-_b\} - \tilde{\alpha} \\ &= \max\{\tilde{d}^+_a - \tilde{d}^-_a + \tilde{\beta}_a, \tilde{d}^+_b - \tilde{d}^-_b + \tilde{\beta}_b\} - \tilde{\alpha}.\end{aligned}$$

Here, the inequality follows from inequality (42). The last equality holds by the fact that $\tilde{d}^+_a = d^+_a$, $\tilde{d}^+_b = d^+_b$, $\tilde{d}^-_a = d^-_a$, and $\tilde{d}^-_b = d^-_b$.

(ii) $w_{a\bar{v}}w_{b\bar{v}} < 0$. Since $|w_{a\bar{v}}| > |w_{b\bar{v}}|$, we have $h^{ab}_{\bar{v}} = w_{b\bar{v}}$ that implies $\beta_a = |w_{a\bar{v}} - h^{ab}_{\bar{v}}| = |w_{a\bar{u}}|$ and $\beta_b = 0$. We also have $\tilde{\beta}_a = \tilde{\beta}_b = 0$. Thus,

$$\begin{aligned}\tilde{d}^+_{ab} - \tilde{d}^-_{ab} - \frac{3}{2}\min\{\tilde{w}_{ab}, 0\} &= d^+_{ab} - d^-_{ab} - \frac{3}{2}\min\{w_{ab}, 0\} \\ &\geq \max\{d^+_a - d^-_a + |w_{au}|, d^+_b - d^-_b\} - \alpha^* \\ &= \max\{\tilde{d}^+_a - \tilde{d}^-_a, \tilde{d}^+_b - \tilde{d}^-_b\} - \tilde{\alpha} \\ &= \max\{\tilde{d}^+_a - \tilde{d}^-_a + \tilde{\beta}_a, \tilde{d}^+_b - \tilde{d}^-_b + \tilde{\beta}_b\} - \tilde{\alpha}.\end{aligned}$$

Here, the inequality follows from (42). The last equality holds by the fact that $\tilde{d}^+_a - \tilde{d}^-_a = d^+_a - d^-_a + |w_{a\bar{u}}|$, $\tilde{d}^+_b = d^+_b$, and $\tilde{d}^-_b = d^-_b$.

By cases 1 and 2, there exists solution $x'' \in \{0,1\}^{(n+1)\times k}$ such that $g(x'') \geq g(x')$. By the construction of x'', we have $x''_{\bar{u}j} = x''_{\bar{v}j}$ for every partition $j \in P$. Thus there exists solution $\bar{x} \in \{0,1\}^{n\times k}$ such that for every vertex $v \in V$ and every partition $j \in P$, we have $\bar{x}_{vj} = x''_{vj}$. Similar to our argument on equality $g(x') = g(\hat{x})$ in Case 3, we have $g(\bar{x}) = g(x'')$. Thus, $g(\bar{x}) \geq g(\hat{x})$. We note that our arguments hold when we have more than one vertex $v \in C(a,b)$ with $|w_{a\bar{v}}| \neq |w_{b\bar{v}}|$. In such a case, one can add an auxiliary vertex for every vertex $v \in C(a,b)$. □

Appendix D – Simplified QUBO and R-QUBO Models

The simplified version of the QUBO model (19) is provided as follows.

$$
\begin{aligned}
q(x) &= \sum_{\{u,v\}\in E} w_{uv} - \sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} x_{uj}x_{vj} - \sum_{v\in V} c_v \Big(\sum_{j\in P} x_{vj} - 1\Big)^2, \\
&= \sum_{\{u,v\}\in E} w_{uv} - \sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} x_{uj}x_{vj} - \sum_{v\in V} c_v \Big[\Big(\sum_{j\in P} x_{vj}\Big)^2 - 2\sum_{j\in P} x_{vj} + 1\Big], \\
&= \sum_{\{u,v\}\in E} w_{uv} - \sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} x_{uj}x_{vj} - \sum_{v\in V} c_v \Big(\sum_{j\in P} x_{vj}^2 + 2\sum_{\{i,j\}\in\binom{P}{2}} x_{vi}x_{vj} - 2\sum_{j\in P} x_{vj} + 1\Big), \\
&= \sum_{\{u,v\}\in E} w_{uv} - \sum_{\{u,v\}\in E} w_{uv} \sum_{j\in P} x_{uj}x_{vj} - \sum_{v\in V} c_v \Big(2\sum_{\{i,j\}\in\binom{P}{2}} x_{vi}x_{vj} - \sum_{j\in P} x_{vj} + 1\Big).
\end{aligned}
$$

Here, the last equality holds because $x_{vj} \in \{0,1\}$ for every vertex $v \in V$ and every partition $j \in P$; so, we have $x_{vj}^2 = x_{vj}$. Furthermore, the simplified version of the R-QUBO model (21) is provided below.

$$
\begin{aligned}
\bar{q}(x) &= \sum_{\{u,v\}\in E} w_{uv} \Big[1 - \sum_{j\in\bar{P}} x_{uj}x_{vj} - \Big(1 - \sum_{j\in\bar{P}} x_{uj}\Big)\Big(1 - \sum_{j\in\bar{P}} x_{vj}\Big)\Big] - \sum_{v\in V} c_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj} \\
&= \sum_{\{u,v\}\in E} w_{uv} \Big[\sum_{j\in\bar{P}} x_{uj} + \sum_{j\in\bar{P}} x_{vj} - \sum_{j\in\bar{P}} x_{uj}x_{vj} - \Big(\sum_{j\in\bar{P}} x_{uj}\Big)\Big(\sum_{j\in\bar{P}} x_{vj}\Big)\Big] - \sum_{v\in V} c_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj}, \\
&= \sum_{\{u,v\}\in E} w_{uv} \Big(\sum_{j\in\bar{P}} (x_{uj} - 2x_{uj}x_{vj} + x_{vj}) - \sum_{\{i,j\}\in\binom{\bar{P}}{2}} (x_{vi}x_{uj} + x_{ui}x_{vj})\Big) - \sum_{v\in V} c_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj}, \\
&= \sum_{\{u,v\}\in E} w_{uv} \Big(\sum_{j\in\bar{P}} (x_{uj} - x_{vj})^2 - \sum_{\{i,j\}\in\binom{\bar{P}}{2}} (x_{vi}x_{uj} + x_{ui}x_{vj})\Big) - \sum_{v\in V} c_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj}.
\end{aligned}
$$

Here, the last equality holds because $x_{vj} \in \{0,1\}$ for every vertex $v \in V$ and every partition $j \in P$; so, we have $x_{vj} = x_{vj}^2$.