On Hamiltonians of the Max $k$-cut Problem

Ramin Fakhimi$^1$, Hamidreza Validí$^2$, Illya V. Hicks$^3$, Tamás Terlaky$^1$, and Luis F. Zuluaga$^1$

$^1$Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA
$^2$Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, TX, USA
$^3$Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA

ISE Technical Report 23T-020
On Hamiltonians of the max $k$-cut problem

RAMIN FAKHIMI, Quantum Computing and Optimization Lab, Lehigh University, USA
HAMIDREZA VALIDI, Industrial, Manufacturing & Systems Engineering, Texas Tech University, USA
ILLYA V. HICKS, Computational Applied Mathematics and Operations Research, Rice University, USA
TAMÁS TERLAKY and LUIS F. ZULUAGA, Quantum Computing and Optimization Lab, Lehigh University, USA

The max $k$-cut problem is a challenging combinatorial optimization problem that generalizes the max-cut problem and arises in applications such as statistical physics and scheduling. A Hamiltonian formulation of the max $k$-cut problem allows using quantum computing devices to find feasible solutions for the problem. However, unlike the max cut problem (i.e., the max $k$-cut problem with $k = 2$), the Hamiltonian formulation of the max $k$-cut problem is not unique regarding its penalty coefficients in the objective function. This means penalty coefficients can significantly affect the performance of quantum computing devices. In this paper, we propose and fully characterize two Hamiltonian formulations of the max $k$-cut problem and compare them computationally. Our computational results show the superiority of the Hamiltonian formulation with tighter penalty coefficients when applied to the quantum approximate optimization algorithm (QAOA). The code and data used for these numerical experiments are publicly available on GitHub.

Additional Key Words and Phrases: quantum computing, the max $k$-cut problem, Hamiltonian, QAOA

1 INTRODUCTION

The max $k$-cut problem is among the challenging NP-hard problems [17, 27] with multiple notorious optimization formulations in the literature. Carlson and Nemhauser [6] 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 [10]. 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 we can employ existing optimization formulations to solve the problem on classical solvers, one needs to convert classical formulations to quadratic unconstrained binary optimization (QUBO) to "solve" them via quantum algorithms like the quantum approximate optimization algorithm (QAOA).

In a QUBO formulation of an optimization formulation, we have only one objective function, and all constraints of the optimization formulation are moved to the objective function of the QUBO formulation so that any infeasible solution is penalized. In the quantum context, one important research question is developing efficient QUBO formulations for "solving" combinatorial optimization problems [32]. For more information about QUBO formulations, interested readers are referred to [19, 29]. Once the constrained problem has been formulated as a QUBO problem, it can be converted to a Hamiltonian formulation. Hadfield [20] provided a general framework for converting the classical representation of a pseudo-Boolean objective function $f$ into its Hamiltonian $H_f$. In this paper, we propose new QUBO formulations for the max $k$-cut problem and compare them with a naive QUBO formulation of the problem in a quantum context.
In recent years, quantum computing has gained significant attention due to its potential to provide exponential speedup over classical computers in solving certain optimization problems [1–3, 12, 15, 16, 21, 25]. In particular, the quantum approximate optimization algorithm known as QAOA algorithm has shown promising results in solving the quadratic unconstrained binary optimization (QUBO) formulations [5, 8, 23, 30]. QAOA performance on constrained problems depends on the choice of penalty terms. In addition, it depends on the number of qubits used to represent the problem. While QAOA is mostly considered for the max cut problem, there is growing interest in using it to solve the generalized max $k$-cut problem [18, 28].

This paper explores the boundaries of the QAOA to “solve” the max $k$-cut problem. In Section 2, we provide a brief background on the max $k$-cut problem and quantum computing. In Section 3, we introduce two quadratic unconstrained binary optimization formulations with tight penalty coefficients. A set of computational results is provided in Section 4. Section 5 concludes the paper and provides directions for future research works.

2 PRELIMINARIES

The max $k$-cut problem aims to partition the vertex set of a graph into $k \geq 2$ partitions such that the weight of the cut edges (i.e., edges whose endpoints belong to different partitions) is maximized. When $k = 2$, the problem becomes the well-known max cut problem. Carlson and Nemhauser [6] introduced the following BQO formulation for the max $k$-cut problem. Given a graph $G = (V, E)$, we define $n := |V|$ and $m := |E|$ as the number of its vertices and edges, respectively. Furthermore, let $P := \{1, \ldots, k\}$ be the set of partitions. For every vertex $v \in V$ and every partition $j \in P$, binary variable $x_{v,j}$ is one if vertex $v$ is assigned to partition $j$ and zero otherwise. Then the max $k$-cut problem can be formulated as

$$\begin{align*}
\text{max} & \quad \sum_{(u,v) \in E} w_{uv} \left( 1 - \sum_{j \in P} x_{u,j} x_{v,j} \right) \\
\text{s.t.} & \quad \sum_{j \in P} x_{v,j} = 1, \quad \forall v \in V \\
& \quad x \in \{0,1\}^{n \times k}.
\end{align*}$$

(1a)

Objective function (1a) maximizes the number of cut edges, and constraints (1b) imply that each vertex must be assigned to exactly one partition. Now we introduce a reduced variant of the BQO formulation: R-BQO. In this formulation, the number of variables are reduced by $n$. For every vertex $v \in V$ and partition $k \in P$ (the last partition), we define

$$x_{ok} := 1 - \sum_{j \in P \setminus \{k\}} x_{v,j}. \tag{2}$$

The R-BQO formulation is as follows.

$$\begin{align*}
\text{max} & \quad \sum_{(u,v) \in E} w_{uv} \left( 1 - \sum_{j \in P \setminus \{k\}} x_{u,j} x_{v,j} \right) \left( 1 - \sum_{j \in P \setminus \{k\}} x_{u,j} \right) \left( 1 - \sum_{j \in P \setminus \{k\}} x_{v,j} \right) \\
\text{s.t.} & \quad \sum_{j \in P \setminus \{k\}} x_{v,j} \leq 1, \quad \forall v \in V \\
& \quad x \in \{0,1\}^{n \times (k-1)}.
\end{align*}$$

(3a)
3 QUADRATIC UNCONSTRAINED BINARY OPTIMIZATION FORMULATIONS

One way to solve the max k-cut problem on a quantum computing device is to formulate it as a quadratic unconstrained binary optimization problem. QUBO reformulations of many combinatorial optimization problems can be obtained by penalizing the violation of the constraints of a mixed integer optimization formulation in the objective function. Several studies have explored the QUBO formulation for various optimization problems. Padberg [26] conducted a polyhedral study on a general QUBO formulation for the max cut problem and its naive continuous relaxation. Butenko [4] proposed a QUBO formulation for the maximum independent set problem. Dunning et al. [9] evaluated the performance of different heuristic algorithms for solving the max cut problem using the QUBO formulation. Quintero et al. [31] proposed a QUBO formulation for the maximum k-colorable subgraph problem.

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

$$
N_G^+(v) := \{ u \in N_G(v) \mid w_{uv} > 0 \}, \quad \text{and} \quad N_G^-(v) := \{ u \in N_G(v) \mid w_{uv} < 0 \}.
$$

$$
d^+_v := \sum_{u \in N_G^+(v)} w_{uv} \quad \text{and} \quad d^-_v := \sum_{u \in N_G^-(v)} w_{uv}.
$$

We propose two QUBO formulations: (i) QUBO formulation corresponding to the BQO formulation (1), and (ii) R-QUBO formulation corresponding to the R-BQO formulation (3).

3.1 QUBO formulation

We first propose a QUBO formulation inferred from the BQO formulation. In other words, we move constraints (1b) of the BQO formulation to the objective function and penalize them by a vector $c \in \mathbb{R}^n$. Our proposed QUBO formulation is as follows.

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

An optimal solution of the QUBO formulation (5) is not necessarily a feasible solution of the max k-cut problem. We propose Algorithm 1 that converts any binary point $\tilde{x} \in \{0,1\}^{n \times k}$ to a feasible solution of the max k-cut problem. In this algorithm, vertex sets $L_0$ and $L_1$ 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 5–13 runs in $O(kmn)$. The last for loop assigns vertices with no assignment to a partition that locally maximizes the objective function. Lines 14–16 runs in $O(km)$. In total, Algorithm 1 takes time $O(knm)$.

Lemma 3.1 provides a tight lower bound for the penalty vector $c$ in QUBO formulation (5).

**LEMMA 3.1.** Let $c$ be a penalty vector and $\tilde{x}$ be an optimal solution of the QUBO formulation (5). If $c_v \geq \max \left\{ d^+_v, d^-_v \right\}$ for every vertex $v \in V$, then Algorithm 1 returns a feasible solution of the BQO formulation that is optimal for QUBO.

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

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

On Hamiltonians of the max k-cut problem

3
Suppose that $\tilde{x}$ represents an optimal binary solution of the QUBO formulation in which there is a vertex $v \in V$ with multiple partitions; that is, $\sum_{j \in P} \tilde{x}_{vj} > 1$. Let $t_o := \sum_{j \in P} \tilde{x}_{vj}$. The following claim shows every vertex is assigned to at most two partitions in the solution represented by $\tilde{x}$.

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

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

$$q_1(\tilde{x}) - q_1(\hat{x}) = \sum_{u \in N_G(v)} \sum_{j \in P} w_{uv} \tilde{x}_{uj} \hat{x}_{uj} - \sum_{u \in N_G(v)} \sum_{j \in P} w_{uv} \hat{x}_{uj} \hat{x}_{uj}$$

$$= \sum_{u \in N_G(v)} w_{uv} \hat{x}_{ui} \hat{x}_{ui} + \sum_{u \in N_G(v)} w_{uv} \hat{x}_{ui} \hat{x}_{ui}$$

$$\geq 0 + d^-_v,$$

$$\therefore t_o \leq 2.$$
where equality (7a) holds by the definition of \( q_1(\cdot) \), and inequality (7c) holds since \( \sum_{u \in N_{G}^+(v)} w_{uv} \tilde{x}_u \tilde{x}_v \geq 0 \). Finally,

\[
q(\tilde{x}) - q(\tilde{x}) = \left[ q_1(\tilde{x}) - q_1(\tilde{x}) \right] + \left[ q_2(\tilde{x}) - q_2(\tilde{x}) \right] \\
\geq d_0^+ + c_0 \left( t_0 - 1 \right)^2 - (t_0 - 2)^2 \\
= d_0^+ + c_0 (2t_0 - 3) \\
\geq d_0^+ + \max \left\{ \frac{d_0^+-d_0^-}{k} - \frac{d_0^-}{2} \right\} (2t_0 - 3) \\
\geq \begin{cases} 
\frac{d_0^+-d_0^-}{2} (2t_0 - 3) = d_0^+ (2.5 - t_0) > 0 & \text{if } d_0^- < 0, \\
\frac{d_0^-}{2} (2t_0 - 3) > 0 & \text{if } d_0^- = 0. 
\end{cases}
\]

Here, the first inequality holds by inequality (7c). The second inequality holds by the assumption of the lemma. Finally, the last strict inequalities hold because \( d_0^+ - d_0^- > 0 \) and \( t_0 \geq 3 \) by the assumption. This contradicts the fact that \( \tilde{x} \) is an optimal solution of (5). Hence, \( t_0 \leq 2 \).

Recall \( E_\ell = \{ (u, v) \in \mathcal{E}^- | \{ u, v \} \notin C_\ell \} \) from line 8 of Algorithm 1. Furthermore, we have \( I_\ell = \mathcal{U}_{i=1}^{|C_\ell|} C_{\ell i} \) with \( r \) be the number of \( C_\ell \) sets defined in line 7 of the algorithm. For every partition-based class \( \ell \in [r] \), we define \( E'_\ell \) and \( E''_\ell \) as edges with exactly one endpoint in \( E_\ell \) and both endpoints in \( E_\ell \), respectively.

\[
E'_\ell = \{ (u, v) \in E_\ell | \{ u, v \} \notin C_\ell \}, \quad \text{and} \quad E''_\ell = \{ (u, v) \in E_\ell | \{ u, v \} \subseteq C_\ell \},
\]

(8)

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

\[
\tilde{E}'_\ell := \{ (u, v) \in E'_\ell | \tilde{x}_{uj_1} = \tilde{x}_{uj_2} \}, \quad \text{and} \quad \tilde{E}_\ell := \{ (u, v) \in E'_\ell | \tilde{x}_{uj_1} = \tilde{x}_{uj_2} \},
\]

(9)

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

\[
\sum_{(u, v) \in \tilde{E}'_\ell} w_{uv} \geq \sum_{(u, v) \in \tilde{E}_\ell} w_{uv}
\]

(10)

Figure 1 illustrates a solution of the QUBO formulation for the max 3-cut problem with its associated sets defined above. Let \( \tilde{x} \) be the output of Algorithm 1 up to line 14. Now we provide a lower bound for \( q_1(\tilde{x}) - q_1(\tilde{x}) \).

\[
\text{Claim 2.} \quad q_1(\tilde{x}) - q_1(\tilde{x}) \geq \sum_{\ell \in [r]} \left[ \sum_{(u, v) \in \tilde{E}'_\ell} w_{uv} + \sum_{(u, v) \in \tilde{E}_\ell} w_{uv} \right].
\]
Proof. For ease of notation, we define $b_{uv} := \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj} - \sum_{j \in P} \check{x}_{uj}\check{x}_{vj}$ for every edge $\{u, v\} \in E$.

$$q_1(\hat{x}) - q_1(\check{x}) = \sum_{\{u, v\} \in E} \sum_{j \in P} w_{uv} \hat{x}_{uj}\hat{x}_{vj} - \sum_{\{u, v\} \in E} \sum_{j \in P} w_{uv} \check{x}_{uj}\check{x}_{vj}$$  \hspace{1cm} (11a)

$$= \sum_{\{u, v\} \in E} w_{uv} b_{uv}$$  \hspace{1cm} (11b)

$$= \sum_{\{u, v\} \in E^-} w_{uv} b_{uv} + \sum_{\{u, v\} \in E^+} w_{uv} b_{uv}$$  \hspace{1cm} (11c)

$$\geq \sum_{\{u, v\} \in E^-} w_{uv} b_{uv} + 0$$  \hspace{1cm} (11d)

$$\geq \sum_{\ell \in [r]} \sum_{\{u, v\} \in E_\ell} w_{uv} b_{uv} + \sum_{\{u, v\} \in E_\ell} w_{uv} b_{uv}$$  \hspace{1cm} (11e)

$$= \sum_{\ell \in [r]} \sum_{\{u, v\} \in E_\ell} w_{uv} b_{uv} + \sum_{\{u, v\} \in E_\ell} w_{uv} b_{uv}$$  \hspace{1cm} (11f)

$$= \sum_{\ell \in [r]} \sum_{\{u, v\} \in E_\ell' \cup E_\ell''} w_{uv} b_{uv} + \sum_{\{u, v\} \in E_\ell' \cup E_\ell''} w_{uv} b_{uv}$$  \hspace{1cm} (11g)

$$= \sum_{\ell \in [r]} \sum_{\{u, v\} \in E_\ell'} w_{uv} b_{uv} + \sum_{\{u, v\} \in E_\ell''} w_{uv} b_{uv}$$  \hspace{1cm} (11h)

Here, inequality (11d) holds because (i) $\hat{x} \geq \check{x}$ implies $\sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj} - \sum_{j \in P} \check{x}_{uj}\check{x}_{vj} \geq 0$, and (ii) $w_{uv} \geq 0$ for every edge $\{u, v\} \in E^+$. Inequality (11c) 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 1 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 (11e) holds because $b_{uv} = 0$ for every edge $\{u, v\} \in E^- \setminus (E_1 \cup \cdots \cup E_r)$. Note that $\hat{x}_{uj} = \check{x}_{uj}$ and $\hat{x}_{vj} = \check{x}_{vj}$ for every edge $\{u, v\} \in E^- \setminus (E_1 \cup \cdots \cup E_r)$ and every partition $j \in P$. Equality (11f) holds by definitions (8). Equality (11g) holds by definitions (9). Equality (11h) holds because (i) $E_\ell' \cap E_\ell'' = \varnothing$, and (ii) $b_{uv} = 0$ for every edge $\{u, v\} \in E_\ell' \setminus (E_\ell' \cup E_\ell'')$. Note that $\sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj} = \sum_{j \in P} \check{x}_{uj}\check{x}_{vj} = 0$ for every edge $\{u, v\} \in E_\ell' \setminus (E_\ell' \cup E_\ell'')$. Finally, equality (11i) holds because the algorithm implies that for every $\ell \in [r]$, we have the following cases.

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

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

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

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

Claim 3. $q_2(\hat{x}) - q_2(\check{x}) \geq -\sum_{\ell \in [r]} 0.5 \left[ \sum_{\{u, v\} \in E_\ell} w_{uv} + 2 \sum_{\{u, v\} \in E_\ell'} w_{uv} \right]$. 


The following claim shows that \( \hat{x} \) is an optimal solution of the QUBO formulation (5).

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

**Proof.** By Claims 2 and 3, we have

\[
q(\hat{x}) - q(\hat{x}) \geq \sum_{\ell \in [r]} \left| \sum_{\{u,v\} \in E'_\ell} w_{uv} + \sum_{\{u,v\} \in E'_\ell} w_{uv} \right| - \sum_{\ell \in [r]} \frac{1}{2} \left| \sum_{\{u,v\} \in E'_\ell} w_{uv} + 2 \sum_{\{u,v\} \in \bar{E}'_\ell} w_{uv} \right| = 0.5 \sum_{\ell \in [r]} \left| \sum_{\{u,v\} \in E'_\ell} w_{uv} - \sum_{\{u,v\} \in \bar{E}'_\ell} w_{uv} - \sum_{\{u,v\} \in (E'_\ell \cup \bar{E}'_\ell)} w_{uv} \right| \geq 0.
\]

Here, the last inequality holds by (i) inequality (10), and (ii) the fact that \( w_{uv} < 0 \) for every edge \( \{u,v\} \in E'_\ell \setminus (\bar{E}'_\ell \cup \bar{E}'_\ell) \).

Hence, Algorithm 1 returns an optimal solution of the QUBO formulation (1) such that every vertex is assigned to at most one partition.

Let \( \hat{x} \) be the output of Algorithm 1.

**Claim 5.** \( q(\hat{x}) - q(\hat{x}) \geq 0 \).

**Proof.** Suppose that \( \hat{x} \) represents an optimal solution in which a vertex \( v \in V \) is assigned to no partition; that is, \( \sum_{j \in P} \hat{x}_{vj} = 0 \). By line 15 of Algorithm 1, let \( s \) be a partition with minimum value of \( \sum_{u \in N_G(v)} w_{uv} \hat{x}_{uj} \) among all \( j \in P \). By line 16 of the algorithm, we have \( \hat{x}_{us} = 1 \). By definitions (6), we have

\[
q_1(\hat{x}) - q_1(\hat{x}) = - \sum_{u \in N_G(v)} w_{uv} \hat{x}_{us}, \quad \text{and} \quad q_2(\hat{x}) - q_2(\hat{x}) = c_v.
\]

Then, we have the following cases.

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

\[
c_v \geq \frac{d_v^+}{k} \geq \min_{j \in P} \left( \sum_{u \in N_G(v)} w_{uv} \hat{x}_{uj} \right) \geq \min_{j \in P} \left( \sum_{u \in N_G(v)} w_{uv} \hat{x}_{uj} \right) = \sum_{u \in N_G(v)} w_{uv} \hat{x}_{us}. \tag{14}
\]
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 $c_o = \sum_{u \in N_G^+(v)} w_{uv} \bar{x}_{us} \geq 0$. Then, $q(\bar{x}) - q(\bar{x}) \geq 0$ by lines (13) and (14).

(ii) $\sum_{u \in N_G^+(v)} w_{uv} \bar{x}_{us} < 0$. By line (13), it follows that $q(\bar{x}) - q(\bar{x}) \geq 0$.

\[ \square \]

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

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 formulation (1) if an optimal solution of the QUBO formulation (5) is provided.

**Theorem 3.2.** Let $\hat{x}$ be an optimal solution of the QUBO formulation (5) with $c_o \geq \max \{c_i/k_i, -c_i/k_i\}$ for every vertex $v \in V$. Algorithm 1 returns an optimal solution of the BQO formulation (1).

**Proof.** Let $\hat{x}$ 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 (5), we have (i) $q(\hat{x}) \geq q(\bar{x})$, and (ii) $q(\bar{x}) \geq q(x^*)$. By Lemma 3.1, we have (iii) $q(\bar{x}) \geq q(\bar{x})$. By (i) and (iii), $q(\bar{x}) = q(\bar{x})$. Hence, $q(\bar{x}) \geq q(x^*)$ by (ii). Note that $\bar{x}$ is feasible for the BQO formulation (1) by Lemma 3.1; so, we have $q(\bar{x}) \leq q(x^*)$. Thus, $q(\bar{x}) = q(\bar{x}) = q(x^*)$ and $\bar{x}$ is also an optimal solution of the BQO formulation (1).

It should be noted that if $c_o > \max \{c_i/k_i, -c_i/k_i\}$ for every vertex $v \in V$, then an optimal solution of the QUBO formulation (5) is also optimal for the BQO formulation (1). Example 3.3 shows that there are some instances of the max $k$-cut problem for which Theorem 3.1 does not hold if we have $c_o < \max \{c_i/k_i, -c_i/k_i\}$ for some vertex $v \in V$.

**Example 3.3.** 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 $x_{33}^* = 1$. See the leftmost side of Figure 2 for an illustration. Furthermore, the QUBO formulation (5) of the max 3-cut problem is written as follows.

$$q(x) = 7 - \sum_{(u,v) \in E} w_{uv}(x_u x_{v1} + x_u x_{v2} + x_{v3} x_v) - \sum_{v \in V} c_o(x_{v1} + x_{v2} + x_{v3} - 1)^2. \quad (15)$$

Note that $c_1 = -2/2 = 1$, $c_2 = c_3 = \max \left\{ \frac{6}{1}, \frac{1}{2} \right\} = 2$, and $c_4 = c_5 = \frac{3}{2} = 1$. Then, we have $q(x^*) = 7$. Now, we change $c_5$ from 1 to $1 - \varepsilon$ for some $\varepsilon > 0$. Then, $\bar{x}$ is an optimal solution for the QUBO formulation (15) with $\bar{x}_{11} = \bar{x}_{31} = \bar{x}_{41} = 1$ and $\bar{x}_{22} = 1$. Further, $\bar{x}_{31} = \bar{x}_{52} = \bar{x}_{33} = 0$. See Figure 2 (center) for an illustration. However, this implies $\bar{x}$ is an infeasible solution for the max 3-cut problem with $q(\bar{x}) = 7 + \varepsilon$ and $q(\bar{x}) > q(x^*)$.

Similarly, we change $c_1$ from 1 to $1 - \varepsilon$ for some $\varepsilon > 0$. Then, $\bar{x}$ is an optimal solution for the QUBO formulation (15) with $\bar{x}_{11} = \bar{x}_{12} = 1$, but it is an infeasible solution of the max 3-cut problem with $q(\bar{x}) = 7 + \varepsilon$ and $q(\bar{x}) > q(x^*)$. See the rightmost side of Figure 2 for an illustration. Hence, an inappropriate choice of $\varepsilon$ 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.
Algorithm 2

On Hamiltonians of the max $k$-cut problem

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

One can always set a "big" penalty vector $c$ in QUBO formulation (5) 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 naive penalty coefficients for the QUBO formulation (5).

**Remark 1.** For any vertex $v \in V$, penalty coefficient $c_v = d_v^+ - d_v^-$ ensures an optimal solution of the QUBO formulation (5) that represents a feasible solution of the max $k$-cut problem.

### 3.2 R-QUBO formulation

Similar to the R-BQO formulation (3), we propose a reduced QUBO (R-QUBO) formulation. We define partition set $\hat{P} := P \setminus \{k\}$. The R-QUBO formulation with $n(k-1)$ binary variables is provided below.

$$
\max_{x \in \{0,1\}^{n \times (k-1)}} q(x) := \sum_{(u,v) \in E} w_{uv} \left( 1 - \sum_{j \in \hat{P}} x_{uj}x_{vj} - \left( 1 - \sum_{j \in \hat{P}} x_{uj} \right) \left( 1 - \sum_{j \in \hat{P}} x_{vj} \right) \right) - \sum_{v \in V} c_v \sum_{(i,j) \in \hat{P}} x_{ij}. \tag{16}
$$

For $k = 2$, it is worth noting that the penalty term disappears because $\hat{P} = \emptyset$. We also propose Algorithm 2 that converts any infeasible binary solution $\tilde{x} \in \{0,1\}^{n \times (k-1)}$ of the R-BQO formulation (3) to a feasible solution. Similar to Algorithm 1, Algorithm 2 has time complexity $O(knm)$.

**Algorithm 2 Conversion of an infeasible binary solution of the R-BQO (3) to a feasible solution**

**Require:** $(G, \hat{x}, \hat{P})$

1: $I := \{v \in V \mid \sum_{j \in \hat{P}} \hat{x}_{vj} > 1\}$
2: $\hat{x} \leftarrow \hat{x}$
3: $\ell \leftarrow 1$
4: **while** $I \neq \emptyset** do
5: $C_\ell := \{u \in I \mid \hat{x}_{uj} = \hat{x}_{uj}, \forall j \in \hat{P}\}$
6: $E_\ell := \{(u,v) \in E \mid \{u,v\} \cap C_\ell \neq \emptyset\}$
7: $\hat{P}_\ell := \{j \in \hat{P} \mid \hat{x}_{aj} = 1\}$
8: **let** $s \in \text{argmin}_{j \in \hat{P}_\ell} \left( \sum_{(u,v) \in E_\ell} w_{uv}x_{uj}x_{vj} \right)$
9: **fix** $\hat{x}_{aj} = 0$ for every vertex $u \in C_\ell$ and every partition $j \in \hat{P}_\ell \setminus \{s\}$
10: $I \leftarrow I \setminus C_\ell$
11: $\ell \leftarrow \ell + 1$
12: **return** $\hat{x}$

Lemma 3.4 provides a tight lower bound for the penalty vector $c$ in R-QUBO formulation (16).
Lemma 3.4. Let $c$ be a penalty vector and $\hat{x}$ be an optimal solution of the R-QUBO formulation (16). 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 formulation 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 $\hat{x}$ is a feasible solution of the R-BQO formulation returned by Algorithm 2. We have $I = \bigcup_{l=1}^{r} C_l$ with $r$ be the number of $C_l$ sets defined in line 5 of Algorithm 2. We note that for every $t \in [r]$, (i) vertex set $C_t$ is a set of vertices that are assigned to the same multiple partitions, and (ii) without loss of generality assume every vertex $u \in C_t$ is assigned to a less or equal number of partitions than that of any vertex $v \in C_{t+1}$. For any $x$ and any edge $\{u, v\} \in E$, we define function $h_{uv}(x)$ as follows.

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

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

$$h_{uv}(\hat{x}) = w_{uv} \left[ t_{uv} + (1 - t_u)(1 - t_v) \right] , \tag{17a}$$

$$h_{uv}(\check{x}) = \begin{cases} w_{uv} \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} & t_u + t_v \neq 0 \leq w_{uv} \min\{t_{uv}, 1\} \quad t_u + t_v \neq 0, w_{uv} > 0 \\ w_{uv} & t_u + t_v = 0 \quad t_u + t_v = 0, w_{uv} \leq 0 \\ w_{uv} & t_u + t_v = 0 \quad w_{uv} \geq 0 \end{cases} \tag{17b}$$

Here, inequality (17b) holds because (i) $\check{x}$ is feasible for the max $k$-cut problem and (ii) $\check{x} \leq \hat{x}$. Now, we recall the positive edge set as $E^+ := \{ \{u, v\} \in E | w_{uv} > 0 \}$.

Claim. $\sum_{\{u, v\} \in E^+} \left[ h_{uv}(\hat{x}) - h_{uv}(\check{x}) \right] \geq -\sum_{e \in I} \sum_{u \in N^+_v(e)} w_{uv}(t_{uv} - 1) \frac{L}{2}.$

Proof. For every edge $\{u, v\} \in E^+$, we bound $h_{uv}(\hat{x}) - h_{uv}(\check{x})$ by lines (17) as follows:

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

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

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

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

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

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

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

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

(iv) if $\{u, v\} \cap I = \emptyset$, then $h_{uv}(\hat{x}) - h_{uv}(\check{x}) = 0$ because $h_{uv}(\check{x}) = h_{uv}(\hat{x})$ by Algorithm 2.
Now we define $\tilde{E}^+$ as the set of positive edges with exactly one endpoint assigned to multiple partitions (i.e., $\tilde{E}^+ := \{(u, v) \in E^+ \mid |\{u, v\} \cap I| = 1\}$). Hence, we have

$$\sum_{(u, v) \in \tilde{E}^+} \left[ h_{uv}(\tilde{x}) - h_{uv}(\check{x}) \right] \geq \sum_{(u, v) \in \tilde{E}^+} w_{uv}(1 - \max\{t_u, t_v\})$$

(18a)

$$= \sum_{v \in I} \sum_{u \in N_G^I(v) \setminus I} w_{uv}(1 - \max\{t_u, t_v\})$$

(18b)

$$= \sum_{v \in I} \sum_{u \in N_G^I(v) \setminus I} w_{uv}(1 - t_v)$$

(18c)

$$\geq \sum_{v \in I} \sum_{u \in N_G^I(v)} w_{uv}(1 - t_v)$$

(18d)

$$\geq - \sum_{v \in I} \sum_{u \in N_G^I(v)} w_{uv}(t_v - 1) \frac{t_v}{Z}.$$ 

(18e)

Here, inequality (18a) holds by items (i)–(iv). Inequality (18d) holds because (i) for every vertex $v \in I$ we have $t_v \geq 2$, and (ii) $w_{uv} \geq 0$ for every edge $(u, v) \in \tilde{E}^+$. Inequality (18e) holds because for every vertex $v \in I$ we have $t_v \geq 2$. 

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

$$E^-_{\ell} := \{(u, v) \in E^- \mid |\{u, v\} \cap C_{\ell'}| \neq \emptyset, \{u, v\} \cap C_{\ell} = \emptyset, \ell' < \ell, \forall \ell' \in [r]\}.$$ 

In other words, for every $\ell \in [r]$ we define $E^-_{\ell}$ as the set of negative edges with (i) at least one endpoint, say $u$, in $C_{\ell}$, (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 $C_{\ell}$ is defined by line 5 of Algorithm 2, and (ii) for every vertex $v \in C_{\ell}$ partition set $P_{\ell}$ is the set of partitions to which vertex $v$ is assigned (see line 7 of Algorithm 2). We also partition edge set $E^-_{\ell}$ to negative edge sets $E^-_{1\ell}, E^-_{2\ell}, E^-_{3\ell},$ and $E^-_{4\ell}$ as follows.

$$E^-_{1\ell} := \{(u, v) \in E^-_{\ell} \mid |\{u, v\} \cap C_{\ell'}| \neq \emptyset, \{u, v\} \cap C_{\ell} = \emptyset, \text{ and } P_{\ell'} \subseteq \bar{P}_{\ell}\}.$$ 

$$E^-_{2\ell} := \{(u, v) \in E^-_{\ell} \mid |\{u, v\} \cap C_{\ell'}| \neq \emptyset, \{u, v\} \cap C_{\ell} = \emptyset, \text{ and } P_{\ell'} \not\subseteq \bar{P}_{\ell}\}.$$ 

$$E^-_{3\ell} := \{(u, v) \in E^-_{\ell} \mid |\{u, v\} \cap C_{\ell'}| = \emptyset, \{u, v\} \subseteq C_{\ell}\}.$$ 

$$E^-_{4\ell} := \{(u, v) \in E^-_{\ell} \mid |\{u, v\} \cap (V \setminus I) = \emptyset\}.$$ 

Here, set $E^-_{1\ell}$ 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 3, $E^-_{11} = \{(1, 2), (1, 3)\}$ and $E^-_{21} = E^-_{31} = \emptyset$. Set $E^-_{12}$ 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 $j \in P_{\ell'} \setminus \bar{P}_{\ell}$ such that $|P_{\ell'}| \leq |\bar{P}_{\ell}|$. In Figure 3, $E^-_{12} = \emptyset$, $E^-_{22} = \{(2, 3)\}$ and $E^-_{32} = \emptyset$. Set $E^-_{13}$ 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 3, $E^-_{13} = \{(1, 4)\}$ and $E^-_{23} = E^-_{33} = \emptyset$. Set $E^-_{14}$ is the set of negative edges with exactly one endpoint assigned to multiple partitions. In Figure 3, $E^-_{14} = \{(4, 5), (4, 6)\}$ and $E^-_{24} = E^-_{34} = \emptyset$. 

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

\[
\sum_{(u,v) \in E^-} \left[ \sum_{t \in \{r\}} w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1) + \sum_{(u,v) \in E^+_r} w_{uv}(t_u t_v - \max\{t_u, t_v\}) \right] + \sum_{(u,v) \in E^+_r} w_{uv}(t_u t_v - 1) + \sum_{(u,v) \in E^+_r} w_{uv} \geq 0.
\]

Proof. For every edge \( (u,v) \in E^- \), we bound \( h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \) by lines (17) as follows:

(i) if \( (u,v) \in E^-_1 \), then we have \( h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1) \) because \( t_{uv} = \min\{t_u, t_v\} \) and \( h_{uv}(\bar{x}) \leq 0 \) and

\[
h_{uv}(\bar{x}) = w_{uv}(t_u t_v - t_u - t_v + 1) = w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1);
\]

(ii) if \( (u,v) \in E^-_2 \), then we have \( h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \geq w_{uv}(t_u t_v - \max\{t_u, t_v\}) \) because \( t_{uv} \leq \min\{t_u, t_v\} - 1 \) and \( h_{uv}(\bar{x}) \leq 0 \) and

\[
h_{uv}(\bar{x}) = w_{uv}(t_u t_v - t_u - t_v + 1) \geq w_{uv}(t_u t_v - \max\{t_u, t_v\}) ;
\]

(iii) if \( (u,v) \in E^-_3 \), then we have \( h_{uv}(\hat{x}) - h_{uv}(\bar{x}) = w_{uv}(t_u t_v - 1) \) 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

\[
h_{uv}(\bar{x}) = w_{uv}(t_u t_v - t_u - t_v + 1) = w_{uv}(t_u t_v - 1 + 1), \quad \text{and} \quad h_{uv}(\bar{x}) = w_{uv};
\]

(iv) if \( (u,v) \in E^-_4 \), 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

\[
h_{uv}(\bar{x}) = w_{uv}(t_u t_v - t_u - t_v + 1) \geq w_{uv}.
\]

(v) if \( (u,v) \in E^- \setminus \cup_{r \in \{1\}} E^-_r \), 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), \( \sum_{(u,v) \in E^-} [h_{uv}(\hat{x}) - h_{uv}(\bar{x})] \) is bounded below by (19).
Let $\bar{q}(x) = \bar{q}_1(x) + \bar{q}_2(x)$ with

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

(20a)

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

(20b)

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

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

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

\[
\bar{q}_1(\hat{x}) - \bar{q}_1(\bar{x}) \geq \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} (t_u t_v - \max\{t_u, t_v\} + 1) + \sum_{(u,v) \in E_{\ell}' \cup E_{\ell}''} w_{uv} (t_u t_v - \max\{t_u, t_v\})
\]

\[
+ \sum_{(u,v) \in E_{\ell}'} w_{uv} (t_u - 1) + \sum_{(u,v) \in E_{\ell}''} w_{uv} (t_v - 1) \frac{t_u}{2}.
\]

(21)

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

\[
\bar{q}_2(\hat{x}) - \bar{q}_2(\bar{x}) = \sum_{\ell \in [r]} \left( \sum_{u \in \mathcal{N}_G(v)} \frac{t_u}{2} \right) \left[ \sum_{u \in \mathcal{N}_G(v)} w_{uv} - \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} w_{uv} \right]
\]

\[
- \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv} + \sum_{u \in \mathcal{N}_G(v) \setminus \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv}
\]

\[
+ \sum_{u \in \mathcal{N}_G(v) \setminus \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv} - \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \setminus \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv} + \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv}
\]

\[
+ \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv}
\]

\[
- \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv} + \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \setminus \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv}
\]

\[
+ \sum_{\ell \in [r]} \sum_{u \in \mathcal{N}_G(v) \cap \mathcal{C}_\ell} \left( \frac{t_u}{2} \right) w_{uv}.
\]

(22a)

Here, equality (22c) holds by definitions of the partitions of $E_\ell$. Recall that edge sets $E_\ell$’s are mutually exclusive. Furthermore, as we shift from the vertex-based summation in equality (22b) to the edge-based summation in equality (22c), we need to consider the corresponding vertex coefficients $\left( \frac{t_u}{2} \right) + \left( \frac{t_v}{2} \right)$ for a given edge $(u,v) \in E^-$. Inequality (22d) holds because $\left( \frac{t_u}{2} \right) \geq 1$ for every vertex $v \in \ell$; so, $\max \left\{ \left( \frac{t_u}{2} \right), \left( \frac{t_v}{2} \right) \right\} \geq 1$ for every $\ell \in [r]$ and every edge $(u,v) \in E_\ell'$. 

On Hamiltonians of the max $k$-cut problem

13
By inequalities (21) and (22d), we have
\[
\bar{q}(\hat{x}) - \bar{q}(\check{x}) \geq \sum_{\ell \in [r]} \left( \sum_{(u,v) \in E_{\ell}} w_{uv} \left( t_u t_v - \max\{t_u, t_v\} + 1 \right) + \sum_{(u,v) \in E_{\ell}} w_{uv} \left( t_u t_v - \max\{t_u, t_v\} \right) \right)
- \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} \left( \frac{t_u}{2} + \frac{t_v}{2} \right)
= \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} \left( 2\left( t_u t_v - \max\{t_u, t_v\} + 1 \right) - t_u (t_v - 1) - t_v (t_u - 1) \right)
+ \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} \left( 2\left( t_u t_v - \max\{t_u, t_v\} \right) - t_u (t_v - 1) - t_v (t_u - 1) \right)
- \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} \left( (t_u - t_v)^2 + 2 \max\{t_u, t_v\} - t_u - t_v - 2 \right)
- \sum_{\ell \in [r]} \sum_{(u,v) \in E_{\ell}} w_{uv} \left( (t_u - t_v)^2 + 2 \max\{t_u, t_v\} - t_u - t_v \right) \geq 0.
\]
Here, the last inequality holds because for every $\ell \in [r]$ and every $(u,v) \in E_{\ell}$, 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. \qed

We note that the value of tight penalty coefficients compared to naïve ones is reduced from $O(k^2 m)$ to $O(n)$. The following theorem shows that Algorithm 2 returns an optimal solution of the R-BQO formulation (3) if an optimal solution of the R-QUBO formulation (16) is provided.

**Theorem 3.5.** Suppose $\hat{x}$ is an optimal solution for the R-QUBO formulation (16) with $c_v \geq d^+_v - d^-_v$ for every vertex $v \in V$. Algorithm 2 returns a binary optimal solution of R-BQO formulation (3).

**Proof.** The proof is similar to the proof of Theorem 3.2. \qed

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 (16) is also a feasible solution for the R-BQO formulation (3). Example 3.6 shows that there is an instance of the max $k$-cut problem for which Theorem 3.4 is violated if $c_v < d^+_v - d^-_v$ holds for some vertex $v \in V$.

**Example 3.6.** Figure 4 illustrates an instance of the max $3$-cut problem with the optimal objective value of 6. Let $x^*$ be an optimal solution with $x^*_{21} = x^*_{31} = 1, x^*_{13} = x^*_{12} = 0, x^*_{41} = x^*_{42} = 0, x^*_{52} = 1.$ See the left side of Figure 4 for an illustration.

Furthermore, the corresponding R-QUBO formulation is as follows

\[
\tilde{q}(x) = 5 - \sum_{(u,v) \in E} w_{uv} \left( \sum_{j \in \{1,2\}} x_{uj} x_{vj} + \left( 1 - \sum_{j \in \{1,2\}} x_{uj} \right) \left( 1 - \sum_{j \in \{1,2\}} x_{vj} \right) \right) - \sum_{v \in V} c_v x_v x_{v2}.
\]

Let $c_2 = c_3 = 4$ and $c_v = 2$ for every $v \in \{1,4,5\}$. Here, we have $\tilde{q}(x^*) = 6$. 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 $\hat{x}_{21} = \hat{x}_{22} = 1, \hat{x}_{31} = \hat{x}_{32} = 1, \hat{x}_{42} = 0, and \hat{x}_{52} = 1$. See the right side of Figure 4 for an illustration.
\( v \in \{1, 4, 5\} \) and \( j \in \{1, 2\} \). Then, we have \( \tilde{q}(\hat{x}) = 6 + \epsilon \) for the modified R-QUBO formulation. Because \( \tilde{q}(\hat{x}) \geq \tilde{q}(x^*) \), 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.

![Optimal solutions of the R-QUBO formulation for the max 3-cut problem with different penalty coefficients](image)

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

The following remark provides naïve penalty coefficients for the R-QUBO formulation (16).

**Remark 2.** For any vertex \( v \in V \), penalty coefficient \( c_v = k(d^+_{d_v} - d^−_{d_v}) \) ensures there is an optimal solution of the R-QUBO formulation (16) that represents a feasible solution of the max \( k \)-cut problem.

## 4 Computational Experiments

In this section, we evaluate the performance of the QAOA on the IBM quantum machines to solve the proposed QUBO formulations of the max \( k \)-cut problem. We run the computational experiments on the IBM quantum machines ibmq_montreal 1.11.31 and ibm_washington 1.6.13. IBM quantum machine ibmq_montreal has 27 qubits with median CNOT and readout errors \( 1.071 \times 10^{-2} \) and \( 1.800 \times 10^{-2} \), respectively. Furthermore, its T1 (thermal relaxation time) and T2 (dephasing time) are 85.2 us and 76.31 us, respectively. IBM quantum machine ibm_washington has 127 qubits with median CNOT and readout errors \( 1.377 \times 10^{-2} \) and \( 1.480 \times 10^{-2} \), respectively. This machine has T1 and T2 of 92.3 us and 80.43 us, respectively.

We have developed the Python package MaxKcut [11] to conduct the computational experiments. We employ QISKIT Python package [22] to implement the QAOA for solving our proposed QUBO formulations. Due to the existing restriction on the number of qubits, we conduct all experiments on Erdős-Rényi random graphs with \( n \in \{8, 30\} \) and \( k = 3 \). The generated Erdős-Rényi random graphs have density percentages of 20 and 80 with negative edge percentages of 0 and 40. So, we generate eight instances in total. The absolute value of all edge weights is one. For every instance na_pb_negc, parameters \( a, b, \) and \( c \) denote the number of vertices, graph density, and the fraction of negative edges. To alleviate the computational burden, we employ QAOA \( \beta \) (i.e., the QAOA with one level) for solving the proposed QUBO formulations. We use the COBYLA derivative-free optimization method to determine near-optimal values of \( \gamma \in [0, 2\pi] \) and \( \beta \in [0, \pi] \).

To construct the phase separation operator \( U_\beta(\gamma) \), we briefly discuss Hamiltonian matrices corresponding to our QUBO formulations. For more technical details on quantum circuits, interested readers are referred to Hadfield [20].

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 \( H_{0i} \) is defined as the Hamiltonian of binary clause \( x_{ij} \). For vertices \( u, v \in V \) and partitions \( i, j \in P \), matrix \( H_{ul,vj} \)
is defined as the Hamiltonian of binary clause $x_{ai}x_{aj}$:

$$H_{ai} = \frac{1}{2}(I_{ai} - Z_{ai}), \quad H_{ai,oj} = \frac{1}{2}(I_{ai} - Z_{ai}) \otimes (I_{aj} - Z_{aj}).$$

Furthermore, let $H_{ai,oj}$ be the Hamiltonian matrix of $(x_{ai} - x_{aj})^2$, i.e.,

$$H_{ai,oj} = \frac{1}{2}(I_{ai} \otimes I_{aj} - Z_{ai} \otimes Z_{aj}).$$

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

$$H_q = - \sum_{\{a,b\} \in E} w_{ab} \sum_{j \in P} H_{aj,oj} - \sum_{a \in V} c_a \left[ 2 \sum_{(i,j) \in \binom{V}{2}} H_{ai,oj} - \sum_{j \in P} H_{aj} \right]$$

$$\bar{H}_q = \sum_{\{a,b\} \in E} w_{ab} \sum_{j \in P} \bar{H}_{aj,oj} - \sum_{(i,j) \in \binom{V}{2}} \left( H_{ai,oj} + H_{aj,oi} \right) - \sum_{a \in V} c_a \sum_{(i,j) \in \binom{V}{2}} H_{ai,oj}.$$  

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

We run a quantum circuit 4,000 times (the maximum allowable number of shots in IBM quantum machines) to calculate the expected value of the objective value in QAOA$_1$ (i.e., QAOA with one level). We also investigate the effect of tight and naive penalty coefficients on QUBO and R-QUBO formulations. We refer max $\{d_i^c - d_i^o, d_i^c - d_i^o\}$ as tight penalty coefficients for QUBO and R-QUBO, respectively (see Theorems 3.2 and 3.5). For naive penalty coefficients, we employ penalty coefficients provided in Remarks 1 and 2.

**Effect of penalty coefficients.** IBM quantum machines failed several times in the optimization process of tuning circuits’ parameters. In these cases, we restarted the algorithm from the point it had stopped. Generally, the QAOA performance in solving QUBO formulations with tight penalty coefficients was more stable. In particular, the R-QUBO formulation with naive penalty coefficients failed twice compared to its tight variants. The value of the penalty coefficients can affect the objective value of the QUBO formulation. We may have a worse objective value for the same infeasible solution. To have a fair comparison, we fed all solutions obtained by QAOA to the respective tight formulations to eliminate the effect of penalty coefficients. Figure 5 demonstrates that QUBO and R-QUBO formulations with tight penalty coefficients have a slightly better expected solution quality than their naive counterparts.
Effect of the problem instances. Figures 5 and 6 show that the percentage of negative-weight edges adversely affects the performance of the QAOA. However, QAOA is almost indifferent to graph density. Figure 6 shows that the QAOA finds optimal solutions for small instances with $n = 8$. The QAOA$_p$ is a $p$-local algorithm whose output depends only on the vertex’s radius $p$ neighborhood [24]. This characteristic is known as locality. It follows, therefore, QAOA works well when the number of levels $p$ is sufficiently large [3, 7, 13, 14]. Recent studies show that the QAOA’s level $p$ needs to grow at least logarithmically with problem size $n$ for specific combinatorial optimization problems [13]. This characteristic explains why QAOA performance drops as the instance size grows to $n = 30$.

QUBO vs. R-QUBO. R-QUBO requires fewer qubits and has a smaller quantum circuit. This property makes it more resilient in the face of quantum noise. In our experiments, when the QAOA is applied to the QUBO formulation (5), the IBM quantum machines fail twice compared to the case when applied to R-QUBO formulation (16). Figure 5 shows that the R-QUBO model consistently outperforms the QUBO model regarding the expected objective value. Two explanations can be provided for this observation: (i) in the QUBO model, a vertex can be assigned to either no partition or multiple partitions, while it can be assigned to at least one partition in the R-QUBO model; and (ii) the number of assignment opportunities for every vertex is $k$ in the QUBO model while it is $k - 1$ in the R-QUBO model. On the other hand, Figure 6 shows that the tight QUBO formulation outperforms the naïve QUBO formulation and the R-QUBO formulations with respect to the best solution found.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose two quadratic unconstrained binary optimization models with tight penalty coefficients. We conduct a set of experiments on the QUBO models and compare them with each other computationally. This paper explores the boundaries of classical and quantum solvers for the max $k$-cut problem. We see a wide range of research directions: finding optimal values for parameters $\gamma$ and $\beta$ in QAOA 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.

6 ACKNOWLEDGMENTS

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


A.1 Simplified QUBO and R-QUBO formulations

The simplified version of the QUBO formulation (5) is provided as follows.

\[
q(x) = \sum_{(u,v) \in E} w_{uv} - \sum_{(u,v) \in E} w_{uv} \sum_{j \in P} x_{uj}x_{vj} - \sum_{v \in V} c_v \sum_{j \in P} x_{vj} - 1 \right)^2,
\]

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

Furthermore, the simplified version of the R-QUBO formulation (16) is provided below.

\[
\hat{q}(x) = \sum_{(u,v) \in E} w_{uv} \left[ 1 - \sum_{j \in P} x_{uj}x_{vj} - \left( 1 - \sum_{j \in P} x_{uj} \right) \right] - \sum_{v \in V} c_v \sum_{(i,j) \in E} x_{iujx_{ij}}
\]

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