INDUSTRIAL AND SYSTEMS ENGINEERING



On Hamiltonians of the Max k-cut Problem

RAMIN FAKHIMI¹, HAMIDREZA VALIDI², ILLYA V. HICKS³, TAMÁS TERLAKY¹, AND LUIS F. ZULUAGA¹

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

ISE Technical Report 23T-020



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 $k \ge 2$, 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.

Authors' addresses: Ramin Fakhimi, fakhimi@lehigh.edu, Quantum Computing and Optimization Lab, Lehigh University, Bethlehem, Pennsylvania, USA, 18015; Hamidreza Validi, Industrial, Manufacturing & Systems Engineering, Texas Tech University, , Lubbock, Texas, USA, hvalidi@ttu.edu; Illya V. Hicks, Computational Applied Mathematics and Operations Research, Rice University, Houston, Texas, USA; Tamás Terlaky; Luis F. Zuluaga, Quantum Computing and Optimization Lab, Lehigh University, Bethlehem, Pennsylvania, USA, 18015.

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 $k \ge 2$ 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 $P := \{1, ..., k\}$ be 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 and zero otherwise. Then the max *k*-cut problem can be formulated as

$$\max \quad \sum_{\{u,v\}\in E} w_{uv} \left(1 - \sum_{j\in P} x_{uj} x_{vj} \right)$$
(1a)

s.t.
$$\sum_{j \in P} x_{vj} = 1$$
 $\forall v \in V$ (1b)

$$x \in \{0,1\}^{n \times k}.\tag{1c}$$

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

$$x_{vk} \coloneqq 1 - \sum_{j \in P \setminus \{k\}} x_{vj}.$$
 (2)

The R-BQO formulation is as follows.

$$\max \sum_{\{u,v\}\in E} w_{uv} \left[1 - \sum_{j\in P\setminus\{k\}} x_{uj} x_{vj} - \left(1 - \sum_{j\in P\setminus\{k\}} x_{uj} \right) \left(1 - \sum_{j\in P\setminus\{k\}} x_{vj} \right) \right]$$
(3a)

s.t.
$$\sum_{j \in P \setminus \{k\}} x_{vj} \le 1 \qquad \forall v \in V$$
(3b)

$$x \in \{0,1\}^{n \times (k-1)}$$
. (3c)

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.

$$N_{G}^{+}(v) := \left\{ u \in N_{G}(v) \mid w_{uv} > 0 \right\}, \text{ and } N_{G}^{-}(v) := \left\{ u \in N_{G}(v) \mid w_{uv} < 0 \right\}, \\ d_{v}^{+} := \sum_{u \in N_{G}^{+}(v)} w_{uv}, \text{ and } d_{v}^{-} := \sum_{u \in N_{G}^{-}(v)} w_{uv}.$$

$$(4)$$

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.

$$\max_{x \in \{0,1\}^{n \times k}} q(x) \coloneqq \sum_{\{u,v\} \in E} w_{uv} \left(1 - \sum_{j \in P} x_{uj} x_{vj} \right) - \sum_{v \in V} c_v \left(\sum_{j \in P} x_{vj} - 1 \right)^2.$$
(5)

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 $\hat{x} \in \{0, 1\}^{n \times k}$ 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 5–13 runs in O(knm). 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 \hat{x} be an optimal solution of the QUBO formulation (5). If $c_v \ge \max \left\{ \frac{d_v^*}{k}, -\frac{d_v^*}{2} \right\}$ 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

$$q_1(x) := \sum_{\{u,v\} \in E} w_{uv} \left(1 - \sum_{j \in P} x_{uj} x_{vj} \right), \text{ and } q_2(x) := -\sum_{v \in V} c_v \left(\sum_{j \in P} x_{vj} - 1 \right)^2.$$
(6)

Algorithm 1 Conversion of an infeasible binary solution of the BQO (1) to a feasible solution

Require: (G, \hat{x}, P) 1: let $I_0 := \left\{ v \in V \mid \sum_{j \in P} \hat{x}_{vj} = 0 \right\}$ 2: let $I_1 := \left\{ v \in V \mid \sum_{j \in P} \hat{x}_{vj} > 1 \right\}$ 3: $\bar{x} \leftarrow \hat{x}$ 4: $\ell \leftarrow 1$ 5: while $I_1 \neq \emptyset$ do let $a \in I_1$ 6: $C_{\ell} \coloneqq \left\{ u \in I_1 \mid \hat{x}_{uj} = \hat{x}_{aj}, \forall j \in P \right\}$ 7: $E_{\ell} := \left\{ \{u, v\} \in E^{-} \mid \{u, v\} \cap C_{\ell} \neq \emptyset \right\}$ 8: $P_{\ell} := \left\{ j \in P \mid \hat{x}_{aj} = 1 \right\}$ 9: $\begin{aligned} &\text{let } s \in \underset{j \in P_{\ell}}{\operatorname{argmin}} \left\{ \sum_{\substack{\{u,v\} \in E_{\ell} \\ \{u,v\} \in E_{\ell} \\ \text{fix } \bar{x}_{uj} = 0, \text{ for every vertex } u \in C_{\ell} \text{ and every partition } j \in P_{\ell} \setminus \{s\} \end{aligned} \right. \end{aligned}$ 10: 11: $I_1 \leftarrow I_1 \setminus C_\ell$ 12: $\ell \leftarrow \ell + 1$ 13: 14: **for** every vertex $v \in I_0$ **do**
$$\begin{split} & \text{let } s \in \mathop{\mathrm{argmin}}_{j \in P} \left\{ \sum_{u \in N_G(v)}^{\circ} w_{uv} \bar{x}_{uj} \right\} \\ & \text{fix } \bar{x}_{vs} = 1 \end{split}$$
15: 16: 17: return \bar{x}

Suppose that \hat{x} represents an optimal binary solution of the QUBO formulation in which there is a vertex $v \in V$ with multiple partitions; that is, $\sum_{j \in P} \hat{x}_{vj} > 1$. 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 $t_v \ge 3$. Without loss of generality, we assume that $N_G(v) \ne \emptyset$; 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

$$q_1(\check{x}) - q_1(\hat{x}) = \sum_{u \in N_G(v)} \sum_{j \in P} w_{uv} \hat{x}_{uj} \hat{x}_{vj} - \sum_{u \in N_G(v)} \sum_{j \in P} w_{uv} \check{x}_{uj} \check{x}_{vj}$$
(7a)

$$= \sum_{u \in N_{G}^{+}(v)} w_{uv} \hat{x}_{ui} \hat{x}_{vi} + \sum_{u \in N_{G}^{-}(v)} w_{uv} \hat{x}_{ui} \hat{x}_{vi}$$
(7b)

$$\geq 0 + d_v^-,\tag{7c}$$

q(

where equality (7a) holds by the definition of $q_1(\cdot)$, and inequality (7c) holds since $\sum_{u \in N_G^+(v)} w_{uv} \hat{x}_{ui} \hat{x}_{vi} \ge 0$. Finally,

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

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 $d_v^+ - d_v^- > 0$ and $t_v \ge 3$ by the assumption. This contradicts the fact that \hat{x} is an optimal solution of (5). Hence, $t_v \le 2$.

Recall $E_{\ell} = \{\{u, v\} \in E^- \mid \{u, v\} \cap C_{\ell} \neq \emptyset\}$ from line 8 of Algorithm 1. Furthermore, we have $I_1 = \bigcup_{\ell=1}^r C_{\ell}$ 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.

$$E_{\ell}' = \{\{u, v\} \in E_{\ell} \mid \{u, v\} \not\subseteq C_{\ell}\}, \quad \text{and} \quad E_{\ell}'' = \{\{u, v\} \in E_{\ell} \mid \{u, v\} \subseteq C_{\ell}\},$$
(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 \hat{x} is an optimal solution because we already proved that $|P_{\ell}| \le 2$. 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.

$$\hat{E}'_{\ell} := \left\{ \{u, v\} \in E'_{\ell} \mid \hat{x}_{uj_1} = \hat{x}_{vj_1} \right\}, \text{ and } \tilde{E}'_{\ell} := \left\{ \{u, v\} \in E'_{\ell} \mid \hat{x}_{uj_2} = \hat{x}_{vj_2} \right\},$$
(9)

where $\hat{E}'_{\ell} \cap \tilde{E}'_{\ell} = \emptyset$ 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

$$\sum_{\{u,v\}\in \hat{E}'_{\ell}} w_{uv} \ge \sum_{\{u,v\}\in \tilde{E}'_{\ell}} w_{uv}.$$
(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 $q_1(\tilde{x}) - q_1(\hat{x})$.

$$\begin{array}{c} \hat{x}_{21} = 1 \\ \hat{x}_{22} = 0 \\ \hat{x}_{23} = 1 \end{array} \xrightarrow{\begin{array}{c} x_{11} = 1 \\ \hat{x}_{12} = 1 \\ \hat{x}_{13} = 0 \end{array}} \xrightarrow{\begin{array}{c} x_{41} = 1 \\ \hat{x}_{42} = 1 \\ \hat{x}_{13} = 0 \\ \hat{x}_{13} = 0 \end{array} \xrightarrow{\begin{array}{c} x_{43} = 0 \\ \hat{x}_{43} = 0 \\ -1 \end{array} \xrightarrow{\begin{array}{c} x_{51} = 0 \\ \hat{x}_{53} = 0 \\ -1 \end{array}} \xrightarrow{\begin{array}{c} x_{51} = 0 \\ \hat{x}_{53} = 0 \\ -1 \\ -1 \end{array} \xrightarrow{\begin{array}{c} x_{61} = 0 \\ \hat{x}_{62} = 0 \\ \hat{x}_{63} = 0 \end{array}}$$

Fig. 1. A solution of the QUBO formulation 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) $E_1 = \{\{1,2\}, \{1,3\}, \{1,4\}, \{4,5\}, \{4,6\}\}, E_2 = \{\{1,2\}\}$, and $E_3 = \{\{1,3\}\}$, (iv) $E'_1 = \{\{1,2\}, \{1,3\}, \{4,5\}, \{4,6\}\}, E''_1 = \{\{1,4\}\}, and (v) \hat{E}'_1 = \{\{1,2\}\}, and \tilde{E}'_1 = \{\{1,3\}, \{4,5\}\}.$

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

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 $\{u, v\} \in E$.

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

$$=\sum_{\{u,v\}\in E}w_{uv}b_{uv} \tag{11b}$$

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

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

$$\geq \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell}} w_{uv} b_{uv} + \sum_{E^- \setminus (E_1 \cup \dots \cup E_r)} w_{uv} b_{uv}$$
(11e)

$$= \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell}} w_{uv} b_{uv} + 0$$
(11f)

$$= \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E'_{\ell}} w_{uv} b_{uv} + \sum_{\{u,v\} \in E''_{\ell}} w_{uv} b_{uv} \right]$$
(11g)

$$= \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in \tilde{E}'_{\ell} \cup \hat{E}'_{\ell}} w_{uv} b_{uv} + \sum_{\{u,v\} \in E'_{\ell} \setminus (\tilde{E}'_{\ell} \cup \hat{E}'_{\ell})} w_{uv} b_{uv} + \sum_{\{u,v\} \in E''_{\ell}} w_{uv} b_{uv} \right]$$
(11h)

$$= \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in \tilde{E}'_{\ell} \cup \hat{E}'_{\ell}} w_{uv} b_{uv} + 0 + \sum_{\{u,v\} \in E''_{\ell}} w_{uv} b_{uv} \right]$$
(11i)

$$= \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in \hat{E}'_{\ell}} w_{uv} + \sum_{\{u,v\} \in E''_{\ell}} w_{uv} \right].$$
(11j)

Here, inequality (11d) holds because (i) $\hat{x} \geq \tilde{x}$ implies $\sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} - \sum_{j \in P} \tilde{x}_{uj} \tilde{x}_{vj} \geq 0$, and (ii) $w_{uv} \geq 0$ for every edge $\{u, v\} \in E^+$. Inequality (11e) holds because we might have $\ell_1 \in [r]$ and $\ell_2 \in [r]$ for which $E_{\ell_1} \cap E_{\ell_2} \neq \emptyset$; 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 (11f) 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 (11g) holds by definitions (8). Equality (11h) holds by definitions (9). Equality (11i) holds because (i) $\tilde{E}'_{\ell} \cap \hat{E}'_{\ell} = \emptyset$, 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 $\{u, v\} \in E'_{\ell} \setminus (\tilde{E}'_{\ell} \cup \hat{E}'_{\ell})$. 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 (11j) holds because the algorithm implies that for every $\ell \in [r]$, we have the following cases.

(i)
$$b_{uv} = \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} - \sum_{j \in P} \tilde{x}_{uj} \hat{x}_{vj} = 1 - 1 = 0$$
 for every $\{u, v\} \in \tilde{E}'_{\ell}$,
(ii) $b_{uv} = \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} - \sum_{j \in P} \tilde{x}_{uj} \hat{x}_{vj} = 1 - 0 = 1$ for every $\{u, v\} \in \tilde{E}'_{\ell}$, and
(iii) $b_{uv} = \sum_{j \in P} \hat{x}_{uj} \hat{x}_{vj} - \sum_{j \in P} \tilde{x}_{uj} \hat{x}_{vj} = 2 - 1 = 1$ for every $\{u, v\} \in E'_{\ell}$.

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

=

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

PROOF. We have

$$q_2(\tilde{x}) - q_2(\hat{x}) = -\sum_{v \in V} c_v \left[\left(\sum_{j \in P} \tilde{x}_{vj} - 1 \right)^2 - \left(\sum_{j \in P} \hat{x}_{vj} - 1 \right)^2 \right]$$
(12a)

$$= -\sum_{v \in I_1} c_v \left[\left(\sum_{j \in P} \tilde{x}_{vj} - 1 \right)^2 - \left(\sum_{j \in P} \hat{x}_{vj} - 1 \right)^2 \right]$$
(12b)

$$=\sum_{v\in I_1}c_v\tag{12c}$$

$$\geq -\sum_{\ell \in [r]} \sum_{v \in C_{\ell}} 0.5 d_v^{-} \tag{12d}$$

$$= -\sum_{\ell \in [r]} 0.5 \left[\sum_{\{u,v\} \in E'_{\ell}} w_{uv} + 2 \sum_{\{u,v\} \in E''_{\ell}} w_{uv} \right].$$
(12e)

Here, equality (12b) holds because $\hat{x}_{vj} = \tilde{x}_{vj}$ for every vertex $v \in V \setminus I_1$ and every partition $j \in P$. Equality (12c) holds because we have $\sum_{j \in P} \hat{x}_{vj} = 2$ for every vertex $v \in I_1$. Inequality (12d) holds because by assumption we have $c_v \ge -0.5d_v^-$ for every vertex $v \in I_1$. Equality (12e) holds by definitions (8).

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

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

PROOF. By Claims 2 and 3, we have

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

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

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 15 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 16 of the algorithm, we have $\bar{x}_{vs} = 1$. By definitions (6), we have

$$q_1(\bar{x}) - q_1(\bar{x}) = -\sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us}, \text{ and } q_2(\bar{x}) - q_2(\bar{x}) = c_v.$$
(13)

Then, we have the following cases.

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

$$c_{v} \geq \frac{d_{v}^{+}}{k} \geq \min_{j \in P} \left\{ \sum_{u \in N_{G}^{+}(v)} w_{uv} \tilde{x}_{uj} \right\} \geq \min_{j \in P} \left\{ \sum_{u \in N_{G}(v)} w_{uv} \tilde{x}_{uj} \right\} = \sum_{u \in N_{G}(v)} w_{uv} \tilde{x}_{us}.$$
(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 $c_v - \sum_{u \in N_G(v)} w_{uv} \tilde{x}_{us} \ge 0$. Then, $q(\tilde{x}) - q(\tilde{x}) \ge 0$ by lines (13) and (14).

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

By Claims 4 and 5, $q(\bar{x}) - q(\hat{x}) \ge 0$. 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 $c_v \ge \max \left\{ \frac{d_v^+}{k}, -\frac{d_v^-}{2} \right\}$ for every vertex $v \in V$. Algorithm 1 returns an optimal solution of the BQO formulation (1).

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

It should be noted that if $c_v > \max \{\frac{d_v^*}{k}, -\frac{d_v}{2}\}$ 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 $c_v < \max \{\frac{d_v^*}{k}, -\frac{d_v}{2}\}$ 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 $x_{53}^* = 1$. 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.

$$q(x) = 7 - \sum_{\{u,v\} \in E} w_{uv} (x_{u1} x_{v1} + x_{u2} x_{v2} + x_{u3} x_{v3}) - \sum_{v \in V} c_v (x_{v1} + x_{v2} + x_{v3} - 1)^2.$$
(15)

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 (15) 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 (15) 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.



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 \hat{x} of the BQO formulation with $c_1 = 1$ and $c_5 = 1 - \epsilon$, and (right) an infeasible solution \hat{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 naïve 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 $\overline{P} := P \setminus \{k\}$. The R-QUBO formulation with n(k - 1) binary variables is provided below.

$$\max_{x \in \{0,1\}^{n \times (k-1)}} \bar{q}(x) \coloneqq \sum_{\{u,v\} \in E} w_{uv} \left[1 - \sum_{j \in \bar{P}} x_{uj} x_{vj} - \left(1 - \sum_{j \in \bar{P}} x_{uj} \right) \left(1 - \sum_{j \in \bar{P}} x_{vj} \right) \right] - \sum_{v \in V} c_v \sum_{\{i,j\} \in \binom{\bar{P}}{2}} x_{vi} x_{vj}.$$
(16)

For k = 2, it is worth noting that the penalty term disappears because $(\frac{\bar{P}}{2}) = \emptyset$. We also propose Algorithm 2 that converts any infeasible binary solution $\hat{x} \in \{0, 1\}^{n \times (k-1)}$ 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}, \bar{P})
   1: I := \left\{ v \in V \mid \sum_{j \in \bar{P}} \hat{x}_{vj} > 1 \right\}
   2: \bar{x} \leftarrow \hat{x}
  3: \ell \leftarrow 1
   4: while I \neq \emptyset do
                 C_{\ell} \coloneqq \left\{ u \in I \mid \hat{x}_{uj} = \hat{x}_{aj}, \forall j \in \bar{P} \right\}
   5:
                  E_{\ell} := \left\{ \{u, v\} \in E \mid \{u, v\} \cap C_{\ell} \neq \emptyset \right\}
   6:
                 \bar{P}_{\ell} := \left\{ j \in \bar{P} \mid \hat{x}_{aj} = 1 \right\}
   7:
                 let s \in \underset{j \in \bar{P}_{\ell}}{\operatorname{argmin}} \left\{ \sum_{\{u,v\} \in E_{\ell}} w_{uv} \bar{x}_{uj} \bar{x}_{vj} \right\}
   8:
                  fix \bar{x}_{uj} = 0 for every vertex u \in C_{\ell} and every partition j \in \bar{P}_{\ell} \setminus \{s\}
   9:
                  I \leftarrow I \setminus C_{\ell}
 10:
                  \ell \leftarrow \ell + 1
 11:
 12: return \bar{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 $c_v \ge d_v^+ - d_v^-$ 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 \bar{x} is a feasible solution of the R-BQO formulation returned by Algorithm 2. We have $I = \bigcup_{\ell=1}^{r} C_{\ell}$ 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.

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

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

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

$$h_{uv}(\bar{x}) = \begin{cases} w_{uv} \sum_{j \in \bar{P}} \bar{x}_{uj} \bar{x}_{vj} & t_u + t_v \neq 0 \\ y_{uv} & t_u + t_v = 0 \end{cases} \leq \begin{cases} w_{uv} \min\{t_{uv}, 1\} & t_u + t_v \neq 0, w_{uv} > 0 \\ 0 & t_u + t_v \neq 0, w_{uv} \le 0 \\ w_{uv} & t_u + t_v = 0. \end{cases}$$
(17b)

Here, inequality (17b) 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 $E^+ := \{\{u, v\} \in E \mid w_{uv} > 0\}$.

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

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

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

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

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

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

(iii) if $\min\{t_u, t_v\} = 1$ and $\max\{t_u, t_v\} > 1$, then $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \ge w_{uv}(1 - \max\{t_u, t_v\})$ because $h_{uv}(\bar{x}) \le w_{uv} \min\{t_{uv}, 1\} \le w_{uv} t_{uv}$, 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 $w_{uv} \ge 0$ and $\max\{t_u, t_v\} > 1$, thus

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

(iv) if $\{u, v\} \cap I = \emptyset$, 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., $\bar{E}^+ := \{\{u, v\} \in E^+ \mid |\{u, v\} \cap I| = 1\}$). Hence, we have

$$\sum_{\{u,v\}\in E^+} \left[h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \right] \ge \sum_{\{u,v\}\in \bar{E}^+} w_{uv}(1 - \max\{t_u, t_v\})$$
(18a)

$$\sum_{v \in I} \sum_{u \in N_G^+(v) \setminus I} w_{uv} (1 - \max\{t_u, t_v\})$$
(18b)

$$= \sum_{v \in I} \sum_{u \in N_G^+(v) \setminus I} w_{uv} (1 - t_v)$$
(18c)

$$\geq \sum_{v \in I} \sum_{u \in N_{c}^{+}(v)} w_{uv}(1-t_{v})$$
(18d)

$$\geq -\sum_{v \in I} \sum_{u \in N_{G}^{+}(v)} w_{uv}(t_{v}-1) \frac{t_{v}}{2}.$$
(18e)

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

We recall the negative edge set as $E^- := \{\{u, v\} \in E \mid w_{uv} < 0\}$. For every $\ell \in [r]$, we define E^-_{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.

$$E_{\ell}^{-} := \left\{ \{u, v\} \in E^{-} \mid \{u, v\} \cap C_{\ell} \neq \emptyset, \{u, v\} \cap C_{\ell'} = \emptyset, \ \ell' < \ell, \forall \ell' \in [r] \right\}.$$

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}_{ℓ} 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.

$$\begin{split} E_{\ell 1}^{-} &:= \left\{ \{u, v\} \in E_{\ell}^{-} \mid \{u, v\} \cap C_{\ell'} \neq \emptyset, \{u, v\} \cap C_{\ell} \neq \emptyset, \text{ and } \bar{P}_{\ell'} \subset \bar{P}_{\ell} \right\} \\ E_{\ell 2}^{-} &:= \left\{ \{u, v\} \in E_{\ell}^{-} \mid \{u, v\} \cap C_{\ell'} \neq \emptyset, \{u, v\} \cap C_{\ell} \neq \emptyset, \text{ and } \bar{P}_{\ell'} \notin \bar{P}_{\ell} \right\} \\ E_{\ell 3}^{-} &:= \left\{ \{u, v\} \in E_{\ell}^{-} \mid \{u, v\} \subseteq C_{\ell} \right\}, \\ E_{\ell 4}^{-} &:= \left\{ \{u, v\} \in E_{\ell}^{-} \mid \{u, v\} \cap (V \setminus I) \neq \emptyset \right\}. \end{split}$$

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 3, $E_{11}^- = \{\{1, 2\}, \{1, 3\}\}$ and $E_{21}^- = E_{31}^- = \emptyset$. 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}_{ℓ} 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 3, $E_{12}^- = \emptyset$, $E_{22}^- = \{\{2, 3\}\}$ and $E_{32}^- = \emptyset$. 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 3, $E_{13}^- = \{\{1, 4\}\}$ and $E_{23}^- = E_{33}^- = \emptyset$. Set $E_{\ell 4}^-$ is the set of negative edges with exactly one endpoint assigned to multiple partitions. In Figure 3, $E_{14}^- = \{\{4, 5\}, \{4, 6\}\}$ and $E_{24}^- = E_{34}^- = \emptyset$.

Fakhimi et al.



Fig. 3. A solution of the R-QUBO formulation for the max 4-cut problem with (i) $C_1 = \{1, 4\}, C_2 = \{2\}, \text{ and } C_3 = \{3\}; (ii) \bar{P}_1 = \{1, 2, 3\}, \bar{P}_2 = \{1, 3\}, \text{ and } \bar{P}_3 = \{2, 3\}, (iii) E_1^- = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{4, 5\}, \{4, 6\}\}, E_2^- = \{\{2, 3\}\}, \text{ and } E_3^- = \emptyset.$

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

$$\sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E_{\ell_1}^-} w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1) + \sum_{\{u,v\} \in E_{\ell_2}^-} w_{uv}(t_u t_v - \max\{t_u, t_v\}) + \sum_{\{u,v\} \in E_{\ell_3}^-} w_{uv}t_{uv}(t_{uv} - 1) + \sum_{\{u,v\} \in E_{\ell_4}^-} w_{uv} \right].$$

$$(19)$$

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 edge $\{u, v\} \in E_{\ell_1}^-$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \ge w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1)$ because $t_{uv} = \min\{t_u, t_v\}$ and $h_{uv}(\bar{x}) \le 0$ and

$$h_{uv}(\hat{x}) = w_{uv}(t_{uv} + t_u t_v - t_u - t_v + 1) = w_{uv}(t_u t_v - \max\{t_u, t_v\} + 1);$$

(ii) if edge $\{u, v\} \in E_{\ell^2}^-$, then we have $h_{uv}(\hat{x}) - h_{uv}(\bar{x}) \ge w_{uv}(t_u t_v - \max\{t_u, t_v\})$ because $t_{uv} \le \min\{t_u, t_v\} - 1$ and $h_{uv}(\bar{x}) \le 0$ and

$$h_{uv}(\hat{x}) = w_{uv}(t_{uv} + t_u t_v - t_u - t_v + 1) \ge w_{uv}(t_u t_v - \max\{t_u, t_v\})$$

(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

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

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

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

(v) if $\{u, v\} \in E^- \setminus \bigcup_{\ell \in [r]} E^-_{\ell}$, 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), $\sum_{\{u,v\}\in E^-} \left[h_{uv}(\hat{x}) - h_{uv}(\bar{x})\right]$ is bounded below by (19).

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

$$\bar{q}_{1}(x) := \sum_{\{u,v\} \in E} w_{uv} \left[1 - \sum_{j \in \bar{P}} x_{uj} x_{vj} - \left(1 - \sum_{j \in \bar{P}} x_{uj} \right) \left(1 - \sum_{j \in \bar{P}} x_{vj} \right) \right]$$
(20a)

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

$$\tag{20b}$$

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

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

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

$$\bar{q}_{1}(\bar{x}) - \bar{q}_{1}(\hat{x}) \geq \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E_{\ell_{1}}^{-}} w_{uv}(t_{u}t_{v} - \max\{t_{u}, t_{v}\} + 1) + \sum_{\{u,v\} \in E_{\ell_{2}}^{-}} w_{uv}(t_{u}t_{v} - \max\{t_{u}, t_{v}\}) + \sum_{\{u,v\} \in E_{\ell_{3}}^{-}} w_{uv}t_{uv}(t_{uv} - 1) + \sum_{\{u,v\} \in E_{\ell_{4}}^{-}} w_{uv} \right] - \sum_{v \in I} \sum_{u \in N_{G}^{+}(v)} w_{uv}(t_{v} - 1) \frac{t_{v}}{2}.$$
(21)

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

$$\bar{q}_{2}(\bar{x}) - \bar{q}_{2}(\bar{x}) = \sum_{v \in I} {\binom{t_{v}}{2}} \left[\sum_{u \in N_{G}^{+}(v)} w_{uv} - \sum_{u \in N_{G}^{-}(v)} w_{uv} \right]$$

$$= \sum_{v \in I} {\binom{t_{v}}{2}} \sum_{u \in N_{G}^{+}(v)} w_{uv} - \sum_{\ell \in [r]} \sum_{v \in C_{\ell}} {\binom{t_{v}}{2}} \left[\sum_{u \in N_{G}^{-}(v) \cap (I \setminus C_{\ell})} w_{uv} + \sum_{u \in N_{G}^{-}(v) \cap C_{\ell}} w_{uv} + \sum_{u \in N_{G}^{-}(v) \setminus I} w_{uv} \right]$$

$$(22a)$$

$$+ \sum_{u \in N_{G}^{-}(v) \setminus I} w_{uv}$$

$$= \sum_{v \in I} {\binom{t_v}{2}} \sum_{u \in N_G^+(v)} w_{uv} - \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E_{\ell_1}^- \cup E_{\ell_2}^-} w_{uv} \left[{\binom{t_u}{2}} + {\binom{t_v}{2}} \right] + \sum_{\{u,v\} \in E_{\ell_3}^-} w_{uv} t_{uv} (t_{uv} - 1) + \sum_{\{u,v\} \in E_{\ell_4}^-} w_{uv} \max\left\{ {\binom{t_u}{2}}, {\binom{t_v}{2}} \right\} \right]$$
(22c)

$$\geq \sum_{v \in I} {\binom{t_v}{2}} \sum_{u \in N_G^+(v)} w_{uv} - \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E_{\ell_1}^- \cup E_{\ell_2}^-} w_{uv} \left[\binom{t_u}{2} + \binom{t_v}{2} \right] + \sum_{\{u,v\} \in E_{\ell_3}^-} w_{uv} t_{uv} (t_{uv} - 1) \right] \\ + \sum_{\{u,v\} \in E_{\ell_4}^-} w_{uv} \left].$$

$$(22d)$$

Here, equality (22c) holds by definitions of the partitions of E_{ℓ}^- . Recall that edge sets E_{ℓ}^- 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 $\binom{t_u}{2} + \binom{t_v}{2}$ for a given edge $\{u, v\} \in E^-$. Inequality (22d) holds because $\binom{t_v}{2} \ge 1$ for every vertex $v \in I$; so, max $\{\binom{t_u}{2}, \binom{t_v}{2}\} \ge 1$ for every $\ell \in [r]$ and every edge $\{u, v\} \in E_{\ell 4}^-$.

By inequalities (21) and (22d), we have

$$\begin{split} \bar{q}(\bar{x}) &- \bar{q}(\hat{x}) \geq \sum_{\ell \in [r]} \left[\sum_{\{u,v\} \in E_{\ell_1}^-} w_{uv} \left(t_u t_v - \max\{t_u, t_v\} + 1 \right) + \sum_{\{u,v\} \in E_{\ell_2}^-} w_{uv} \left(t_u t_v - \max\{t_u, t_v\} \right) \right] \\ &- \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell_1}^- \cup E_{\ell_2}^-} w_{uv} \left[\binom{t_u}{2} + \binom{t_v}{2} \right] \\ &= \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell_1}^-} \frac{w_{uv}}{2} \left[2(t_u t_v - \max\{t_u, t_v\} + 1) - t_u(t_u - 1) - t_v(t_v - 1) \right] \\ &+ \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell_2}^-} \frac{w_{uv}}{2} \left[2(t_u t_v - \max\{t_u, t_v\}) - t_u(t_u - 1) - t_v(t_v - 1) \right] \\ &= -\sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell_1}^-} \frac{w_{uv}}{2} \left[(t_u - t_v)^2 + 2\max\{t_u, t_v\} - t_u - t_v - 2 \right] \\ &- \sum_{\ell \in [r]} \sum_{\{u,v\} \in E_{\ell_2}^-} \frac{w_{uv}}{2} \left[(t_u - t_v)^2 + 2\max\{t_u, t_v\} - t_u - t_v \right] \geq 0. \end{split}$$

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

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

This finishes the proof.

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 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 $c_v \ge d_v^+ - d_v^-$ 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.

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 $x_{21}^* = x_{31}^* = 1$, $x_{11}^* = x_{12}^* = 0$, $x_{41}^* = x_{42}^* = 0$, and $x_{52}^* = 1$. See the left side of Figure 4 for an illustration.

Furthermore, the corresponding R-QUBO formulation is as follows

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

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 $\bar{q}(\hat{x}) \ge \bar{q}(x^*)$, 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.



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 – ϵ .

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

REMARK 2. For any vertex $v \in V$, penalty coefficient $c_v = k(d_v^+ - d_v^-)$ 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 1.071×10^{-2} and 1.800×10^{-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 1.377×10^{-2} and 1.480×10^{-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 QAOA₁ (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 $U_P(\gamma)$, 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_{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 xuixvj.

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

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

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

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

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

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 QAOA₁ (i.e., QAOA with one level). We also investigate the effect of tight and naïve penalty coefficients on QUBO and R-QUBO formulations. We refer max $\{d_v^*/k, -d_v^-/2\}$ and $d_v^+ - d_v^-$ as tight penalty coefficients for QUBO and R-QUBO, respectively (see Theorems 3.2 and 3.5). For naïve 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 naïve 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 naïve counterparts.



Fig. 5. Expected objective value of QAOA1 for the weighted max 3-cut problems' solutions fed to the tight variant of the corresponding formulation.

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.



Fig. 6. The best objective value of $QAOA_1$ for the weighted max 3-cut problems' solutions fed to the tight variant of the corresponding formulation.

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 γ and β 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

- Boaz Barak and Kunal Marwaha. 2022. Classical algorithms and quantum limitations for maximum cut on high-girth graphs. https://doi.org/10.423 0/LIPICS.ITCS.2022.14
- [2] Joao Basso, 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. https://arxiv.org/abs/2110.14206
- [3] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. 2020. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters* 125 (2020), 260505. Issue 26. https://doi.org/10.1103/PhysRevLett.125.260505
- [4] Sergiy Butenko. 2003. Maximum independent set and related problems, with applications. Ph.D. Dissertation. University of Florida. https://bit.ly/3moSZKc
- [5] Colin Campbell and Edward Dahl. 2022. QAOA of the Highest Order. In 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C). IEEE, remote, 141–146. https://doi.org/10.1109/ICSA-C54293.2022.00035
- [6] R. C. Carlson and George L. Nemhauser. 1966. Scheduling to minimize interaction cost. Operations Research 14, 1 (1966), 52–58. https://doi.org/10.1287/opre.14.1.52
- [7] Chi-Ning Chou, Peter J. Love, Juspreet Singh Sandhu, and Jonathan Shi. 2022. Limitations of Local Quantum Algorithms on Random MAX-k-XOR and Beyond. In 49th International Colloquium on Automata, Languages, and Programming (ICALP 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 229), Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 41:1–41:20. https://doi.org/10.4230/LIPIcs.ICALP.2022.41
- [8] Nike Dattani. 2019. Quadratization in discrete optimization and quantum mechanics. https://doi.org/10.48550/ARXIV.1901.04405
- [9] Iain Dunning, 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 (2018), 608–624. https://doi.org/10.1287/ijoc.2017.0798
- [10] Andreas Eisenblätter. 2002. The semidefinite relaxation of the k-partition polytope is strong. In Integer Programming and Combinatorial Optimization, William J. Cook and Andreas S. Schulz (Eds.). Springer, Berlin, Heidelberg, 273–290. https://doi.org/10.1007/3-540-47867-1_20
- [11] Ramin Fakhimi and Hamidreza Validi. 2022. Max k-cut. https://github.com/qcol-lu/maxkcut
- [12] Ramin Fakhimi and Hamidreza Validi. 2023. Quantum Approximate Optimization Algorithm (QAOA). https://engineering.lehigh.edu/sites/engineering.lehigh.edu/files/_DEPARTMENTS/ise/pdf/tech-papers/23/23T_014.pdf Accepted for Encyclopedia of Optimization.
- [13] Edward Farhi, David Gamarnik, and Sam Gutmann. 2020. The quantum approximate optimization algorithm needs to see the whole graph: A typical case. https://doi.org/10.48550/ARXIV.2004.09002
- [14] Edward Farhi, David Gamarnik, and Sam Gutmann. 2020. The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples. https://doi.org/10.48550/ARXIV.2005.08747
- [15] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. arXiv:1412.6062 arXiv:org/abs/1412.6062
- [16] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2015. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. https://doi.org/10.48550/ARXIV.1412.6062
- [17] Alan Frieze and Mark Jerrum. 1997. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. Algorithmica 18, 1 (1997), 67–81. https://doi.org/10.1007/BF02523688
- [18] Franz G Fuchs, 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 (2021), 1–14. https://doi.org/10.1007/s42979-020-00437-z
- [19] Fred Glover, Gary Kochenberger, and Yu Du. 2019. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. 4OR 17, 4 (2019), 335–371. https://doi.org/10.1007/s10288-019-00424-y
- [20] Stuart Hadfield. 2021. On the representation of Boolean and real functions as Hamiltonians for quantum computing. ACM Transactions on Quantum Computing 2, 4 (2021), 21 pages. https://doi.org/10.1145/3478519
- [21] M. B. Hastings. 2019. Classical and quantum bounded depth approximation algorithms. https://doi.org/10.48550/ARXIV.1905.07047
- [22] Qiskit community IBM Research. 2021. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505
- [23] Andrew Lucas. 2014. Ising formulations of many NP problems. Frontiers in Physics 2 (2014), 5. https://doi.org/10.3389/fphy.2014.00005
- [24] Kunal Marwaha. 2021. Local classical MAX-CUT algorithm outperforms p = 2 QAOA on high-girth regular graphs. Quantum 5 (2021), 437. https://doi.org/10.22331/q-2021-04-20-437
- [25] Kunal Marwaha and Stuart Hadfield. 2022. Bounds on approximating Max kXOR with quantum and classical local algorithms. Quantum 6 (2022), 757. https://doi.org/10.22331/q-2022-07-07-757
- [26] Manfred Padberg. 1989. The boolean quadric polytope: some characteristics, facets and relatives. Mathematical Programming 45, 1 (1989), 139–172. https://doi.org/10.1007/BF01589101
- [27] Christos H. Papadimitriou and Mihalis Yannakakis. 1991. Optimization, approximation, and complexity classes. J. Comput. System Sci. 43, 3 (1991), 425–440. https://doi.org/10.1016/0022-0000(91)90023-X
- [28] Massimiliano Proietti, Filippo Cerocchi, and Massimiliano Dispenza. 2022. Native measurement-based quantum approximate optimization algorithm applied to the Max K-Cut problem. Phys. Rev. A 106 (Aug 2022), 022437. Issue 2. https://doi.org/10.1103/PhysRevA.106.022437

- [29] Abraham P Punnen. 2022. The Quadratic Unconstrained Binary Optimization Problem. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-031-04520-2
- [30] Rodolfo Quintero, 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 (2022), 89. https://doi.org/10.1007/s11128-022-03421-z
- [31] Rodolfo Quintero, 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 (2022), 89. https://doi.org/10.1007/s11128-022-03421-z
- [32] Rodolfo A Quintero and Luis F Zuluaga. 2022. QUBO Formulations of Combinatorial Optimization Problems for Quantum Computing Devices. In Encyclopedia of Optimization. Springer, 1–13.

A APPENDICES

A.1 Simplified QUBO and R-QUBO formulations

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

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

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 formulation (16) is provided below.

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

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 $x_{vj} = x_{vj}^2$.