Industrial and
Systems Engineering

Quantum-inspired Formulations for the Max k-cut Problem

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

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

ISE Technical Report 21T-007

LEHIGH
UNIVERSITY

Quantum-inspired formulations for the max k-cut problem

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

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

May 3, 2021

Abstract

Solving combinatorial optimization problems on quantum computers has attracted many researchers since the emergence of quantum computing. The max k-cut problem is a challenging combinatorial optimization problem with multiple well-known optimization formulations. However, its mixed-integer linear optimization (MILO) formulations and mixed integer semidefinite optimization formulation are all time-consuming to be solved. Motivated by recent progress in classic and quantum solvers, we study a binary quadratic optimization (BQO) formulation and two quadratic unconstrained binary optimization formulations. First, we compare the BQO formulation with the MILO formulations. Further, we propose an algorithm that converts any feasible fractional solution of the BQO formulation to a feasible binary solution whose objective value is at least as good as that of the fractional solution. Finally, we find tight penalty coefficients for the proposed quadratic unconstrained binary optimization formulations.

Keywords: quantum computing; max k-cut problem; binary quadratic optimization; QUBO formulation;

1 Introduction

Quantum computing was born in the early 1980s with the work of Benioff (1980) on the quantum mechanical model of computers. Later, Deutsch and Jozsa (1992), Shor (1994), and Grover (1996) proposed quantum algorithms that show quantum speed-up over their classical counterparts. The quantum speed-up comes from two

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

fundamental principles that do not exist in the classical setting: superposition and entanglement. While superposition helps to apply an operation on multiple states at once, entanglement links the states to each other (Preskill, 2018). These fundamental principles open up a new class of problems in computational complexity theory called bounded-error quantum polynomial time (BQP) (e.g., see Nielsen and Chuang, 2011, Sec. 1.4.5). Quantum computers can solve this class of problems with a bounded error in polynomial time. The idea of solving NP-hard problems with a bounded error has attracted many researchers to investigate the potential of quantum computing further.

The max k-cut problem is among the challenging NP-hard problems (Frieze and Jerrum, 1997; Papadimitriou and Yannakakis, 1991). Given a graph $G = (V, E)$ and a positive integer number $k \geq 2$, the aim in the max k-cut problem is to find at most k partitions such that the number of edges with endpoints in different partitions is maximized. This problem and its variants have a wide range of applications in wireless communication (Aardal et al., 2007; Eisenblätter, 2002; Fairbrother et al., 2018), VLSI layout design (Ruen-Wu Chen et al., 1983; Pinter, 1984), micro-aggregation of statistical data (Domingo-Ferrer and Mateo-Sanz, 2002), sports team scheduling (Mitchell, 2003; Elf et al., 2003), ship loading and unloading at ports (Aslidis, 1990; Avriel and Penn, 1993; Avriel et al., 1998), TV commercial scheduling (Bollapragada and Garbiras, 2004), and statistical physics (Liers et al., 2004; Barahona et al., 1988; De Simone et al., 1995). Figure 1 shows optimal solutions of the max k-cut problem with $k \in \{2, 3, 4\}$ on a complete graph with four vertices. Note that the optimal objective value of the problem is nondecreasing in k.

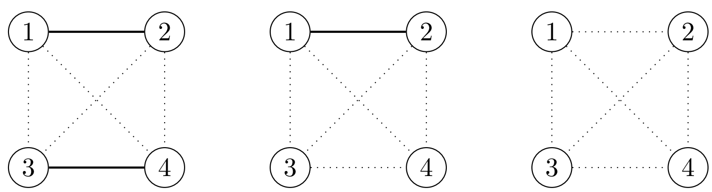

Figure 1: Optimal solutions of the max k-cut problem on a complete graph: $k = 2$ (left); $k = 3$ (middle); $k = 4$ (right). Here dotted edges represent cut edges.

Inspired by the importance of solving combinatorial optimization problems on quantum computers and real-world applications of the max k-cut problem, we study multiple formulations of the problem and compare them analytically and computationally on classic and quantum solvers. One of the basic optimization formulations of the max k-cut problem is an assignment-based mixed integer linear optimization (A-MILO) model with a small number of variables and constraints. Let $n \coloneqq |V|$ and $m \coloneqq |E|$ be the number of vertices and edges of graph $G = (V, E)$, respectively. Furthermore, we define $P \coloneqq \{1, \ldots, k\}$ as the index set of partitions. For every vertex $v \in V$ and partition $j \in P$, binary variable x_{vj} is one if vertex v is assigned to partition j. For every edge $\{u, v\} \in E$, binary variable y_{uv} is one if endpoints of edge $\{u,v\}$ belong to different partitions.

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

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

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

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

Here, constraints (2b) imply that for every set $\{u,v,w\} \subseteq V$, if pairs $\{u,v\}$ and $\{v,w\}$ belong to a partition, then vertices u and w also belong to the same partition. Constraints (2c) imply that vertex set V must be partitioned to at most k partitions. Because of the large number of constraints (2c), one can add them on the fly.

Figure 2 shows the exponential growth of the number of branch-and-bound nodes as the number of vertices increases in Erdös-Rényi random graphs with

$p = 0.9$. Note that each black point in the Figure 2 represents the average number of branch and bound nodes for ten random graphs with density 0.9. We employed Gurobi 9.1 (Gurobi Optimization, 2021) with *default parameters* to solve these instances.

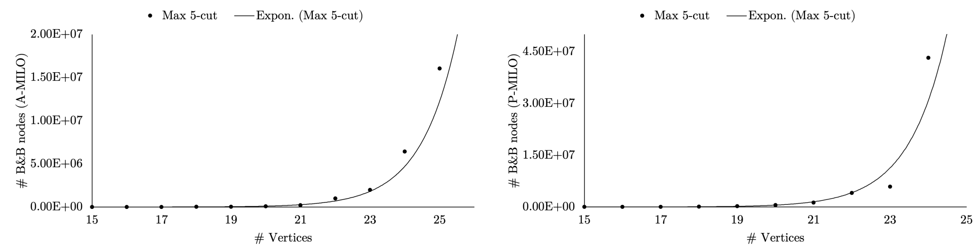

Figure 2: The exponential growth of number of branch and bound nodes with respect to the number of vertices in Erdös-Rényi random graphs with $p = 0.9$ in A-MILO (left) and P-MILO (right) formulations of the max 5-cut problem.

Eisenblätter (2002) proposed a semidefinite reformulation for the P-MILO model (2). In this mixed integer semidefinite optimization (MISDO) reformulation, for every pair of vertices $\{u, v\} \in \binom{V}{2}$, decision variable Z_{uv} is either $\frac{-1}{k-1}$ or 1. Here, for every vertex $v \in V$, variable Z_{vv} is set to one.

$$\text{(MISDO)} \qquad \begin{aligned} \max \quad & \sum_{\{u,v\}\in E} \frac{(k-1)Z_{uv}+1}{k}, && && \text{(3a)} \\ \text{s.t.} \quad & Z \in \mathbb{S}^n_+, && && \text{(3b)} \\ & Z_{vv} = 1, && \forall v \in V. && \text{(3c)} \end{aligned}$$

Furthermore, Eisenblätter (2002) presented a reformulation of the MISDO model (3) that employs binary variables.

Our contributions. Motivated by the poor performance of the A-MILO, P-MILO, and MISDO formulations on classic solvers, we analytically compare the feasible set of the relaxations of the MILO and BQO formulations with each other in Section 3.1. Further, we propose algorithms for converting fractional solutions of the relaxation of the BQO formulation to feasible binary solutions with better objective values. Inspired by the fact that quantum solvers employ unconstrained binary optimization models, we propose (i) two quadratic unconstrained binary optimization (QUBO) models in Section 3.2 and (ii) a higher-order unconstrained binary optimization (HUBO) model in Section 3.2. In Section 4, we provide computational improvements for solving the max k-cut problem.

2 Background

The max k-cut problem generalizes the max cut problem for any integer number $k \geq 2$. There is rich literature on exact methods for solving the max k-cut problem. Chopra and M. Rao (1995) conducted a polyhedral study on the problem and proposed several facet-defining inequalities in the P-MILO context. They also studied A-MILO and P-MILO formulations for the min k-cut problem and proposed multiple facet-defining inequalities in both formulations (Chopra and M. R. Rao, 1993).

In the MISDO context, Eisenblätter (2002) developed a semidefinite formulation for the max k-cut problem. Due to the fact that solving the MISDO formulation is time-consuming, many researchers proposed new valid inequalities and algorithms to handle the MISDO formulation more efficiently (G. Wang and Hijazi, 2020; Lu and Deng, 2021; Ghaddar et al., 2011; Sotirov, 2014; Dam and Sotirov, 2016; Sousa et al., 2018).

In the BQO context, Carlson and Nemhauser (1966) proposed a BQO formulation for the max k-cut problem. They proved that their binary assignment variables can be relaxed for finding an optimal solution. Because of the non-convexity of the BQO's objective function, one can employ branch-and-bound methods to solve the quadratic optimization models (Sherali and Tuncbilek, 1995; Liberti and Pantelides, 2006; Belotti et al., 2009; J. Chen and Burer, 2012; C. Chen et al., 2017; Beck and Pan, 2017; Bonami et al., 2018; Bonami et al., 2019).

We can reformulate a BQO formulation as a QUBO formulation by adding a penalty term associated with each of the constraints to the objective function and relaxing those constraints. Padberg (1989) conducted a polyhedral study on a QUBO formulation with n variables. Butenko (2003) proposed a QUBO formulation for the maximum independent set problem. Based on a QUBO reformulation of the maximum independent set problem, Pajouh et al. (2013) developed an efficient local search for the problem. Dunning et al. (2018) conducted an extensive set of experiments to evaluate the performance of different heuristic algorithms for solving the max cut with QUBO formulation. Şeker et al. (2020) provided computational comparisons for the QUBO formulations of the following problems on different solvers: (i) quadratic assignment, (ii) quadratic cycle partition, and (iii) selective graph coloring. Finally, Quintero et al. (2021) proposed a QUBO formulation for the maximum k-colorable subgraph problem.

Quantum algorithms can solve many NP-hard combinatorial problems (including the max k-cut problem) with a bounded error in polynomial time. Farhi et al. (2014) proposed a quantum approximate optimization algorithm (QAOA) that is one of the most promising quantum algorithms for noisy intermediate-scale quantum devices. The QAOA algorithm has the BQP complexity (Farhi and Harrow, 2016) that makes it interesting for the NP-hard problems with an unconstrained binary optimization formulation. Lucas (2014) formulated several combinatorial optimization problems as QUBO formulations. Several authors applied different variants of the QAOA algorithm on the max-cut problem (Farhi et al., 2014; Hadfield et al., 2019; Z. Wang et al., 2020). Further, Fuchs et al. (2021) directly applied the QAOA algorithm on the max k-cut problem without discussing the unconstrained binary optimization formulation of the problem.

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

3 Nonlinear Optimization Formulations

In this section, we study three different nonlinear optimization models: (i) binary quadratic optimization (BQO), (ii) quadratic unconstrained binary optimization (QUBO), and (iii) higher-order unconstrained binary optimization (HUBO). In Section 3.1, we provide an analytical comparison between BQO and MILO formulations. Our polyhedral results also hold for the minimum k-cut problem. Further, we propose an algorithm for converting a feasible fractional solution of the relaxation of the BQO model to a feasible integral solution with a better objective value. Finally, we propose QUBO and HUBO formulations and prove their correctness in Section 3.2. Before providing a discussion on the models, we introduce our notations as follows.

Notations. Recall simple graph $G = (V, E)$ with vertex set V of size n and edge set E of size m. For every vertex $u \in V$, we define its open neighborhood as $N_G(u) \coloneqq \{v \in V \mid \{u, v\} \in E\}$ with $d_G(u) \coloneqq |N_G(u)|$. Furthermore for every vertex $u \in V$, we define the set of incident edges of vertex u as $\delta_G(u) \coloneqq \big\{\{u, v\} \in E \mid v \in V\big\}$. For every vertex subset $S \subseteq V$, we define $G[S]$ as the subgraph induced by S. For every $n \in \mathbb{Z}_{++}$, we define $[n] \coloneqq \{1, \dots, n\}$. Finally, for every set S, we employ $\binom{S}{2}$ to denote all subsets of S of size 2.

3.1 BQO formulation

A BQO formulation for solving the max k-cut problem is provided below. Note that y variables are introduced for analysis purposes.

$$
\text{(BQO)} \qquad
\begin{aligned}
\max \quad & \sum_{\{u,v\} \in E} y_{uv}, && && \text{(4a)}\\
\text{s.t.} \quad & \sum_{j \in P} x_{vj} = 1, && \forall v \in V, && \text{(4b)}\\
& y_{uv} = 1 - \sum_{j \in P} x_{uj} x_{vj}, && \forall \{u,v\} \in E, && \text{(4c)}\\
& x \in \{0,1\}^{n \times k}. && && \text{(4d)}
\end{aligned}
$$

Here, constraints (4b) imply that every vertex $v \in V$ must belong to exactly one partition $j \in P$. Constraints (4c) imply that an edge $\{u, v\} \in E$ is a cut edge if none of its endpoints belong to the same partition.

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

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

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

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

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

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

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

First, we prove Lemma 1 that is employed in our further analysis.

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

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

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

$$1 - a_1 - a_2 + a_1 a_2 = (1 - a_1)(1 - a_2) \geq 0.$$

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

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

$$0 \leq \Big(1 - \sum_{i \in [s]} a_i + \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j\Big)(1 - a_{s+1}), \tag{6a}$$

$$= 1 - \sum_{i \in [s+1]} a_i + \sum_{\{i,j\} \in \binom{[s+1]}{2}} a_i a_j - a_{s+1} \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j, \tag{6b}$$

$$\leq 1 - \sum_{i \in [s+1]} a_i + \sum_{\{i,j\} \in \binom{[s+1]}{2}} a_i a_j + 0. \tag{6c}$$

Inequality (6a) holds by induction assumption and because $1 - a_{s+1} \geq 0$. Equality (6c) holds because $-a_{s+1} \sum_{\{i,j\} \in \binom{[s]}{2}} a_i a_j \leq 0$. This completes the proof. □

Furthermore, we define the feasible space of the relaxation of the BQO formulation (4) as follows.

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

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

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

Now, we show that the BQO formulation is stronger than the A-MILO formulation.

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

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

$$\hat{y}_{uv} = 1 - \sum_{i \in P} \hat{x}_{ui} \hat{x}_{vi}, \tag{7a}$$

$$= \sum_{i \in P} \hat{x}_{ui} - \sum_{i \in P} \hat{x}_{ui} \hat{x}_{vi}, \tag{7b}$$

$$= \hat{x}_{uj} + \sum_{i \in P \setminus \{j\}} \hat{x}_{ui} - \hat{x}_{uj} \hat{x}_{vj} - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui} \hat{x}_{vi}, \tag{7c}$$

$$\geq \hat{x}_{uj} + \sum_{i \in P \setminus \{j\}} \hat{x}_{ui} - \hat{x}_{vj} - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui} \hat{x}_{vi}, \tag{7d}$$

$$= \hat{x}_{uj} - \hat{x}_{vj} + \sum_{i \in P \setminus \{j\}} \hat{x}_{ui}(1 - \hat{x}_{vi}), \tag{7e}$$

$$\geq \hat{x}_{uj} - \hat{x}_{vj}. \tag{7f}$$

Here, equality (7a) holds by constraints (4c). Equality (7b) follows from constraint (4b). Inequality (7d) holds because $\hat{x}_{uj} \leq 1$. Inequality (7f) holds because $\sum_{i \in P \setminus \{j\}} \hat{x}_{ui}(1 - \hat{x}_{vi}) \geq 0$. Similarly, one can show that $(\hat{x}, \hat{y})$ satisfies constraints (1d).

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

$$
\begin{aligned}
\hat{y}_{uv} &= 1 - \sum_{i \in P} \hat{x}_{ui}\hat{x}_{vi}, && \text{(8a)}\\
&= 1 - \hat{x}_{uj}\hat{x}_{vj} - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui}\hat{x}_{vi}, && \text{(8b)}\\
&= 2 - 1 - \hat{x}_{uj}\hat{x}_{vj} - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui}\hat{x}_{vi}, && \text{(8c)}\\
&= 2 - (1 + \hat{x}_{uj}\hat{x}_{vj}) - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui}\hat{x}_{vi}, && \text{(8d)}\\
&\leq 2 - (\hat{x}_{uj} + \hat{x}_{vj}) - \sum_{i \in P \setminus \{j\}} \hat{x}_{ui}\hat{x}_{vi}, && \text{(8e)}\\
&\leq 2 - (\hat{x}_{uj} + \hat{x}_{vj}). && \text{(8f)}
\end{aligned}
$$

Here, equality (8a) holds by constraints (4c). Inequality (8e) holds by Lemma 1 because we have $\hat{x}_{uj} + \hat{x}_{vj} \leq 1 + \hat{x}_{uj}\hat{x}_{vj}$. Furthermore, inequality (8f) follows from $\sum_{i \in P \setminus \{j\}} \hat{x}_{ui}\hat{x}_{vi} \geq 0$.

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

Now, we reduce the number of variables of the BQO formulation (4) by n. For every vertex $v \in V$ and partition $k \in P$, we define

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

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

$$
\begin{aligned}
& \max && \sum_{\{u,v\}\in E} y_{uv}, && && \text{(9a)}\\
& \text{s.t.} && \sum_{j\in P\setminus\{k\}} x_{vj} \le 1, && \forall v \in V, && \text{(9b)}\\
\text{(R-BQO)} \quad & && x_{vk} = 1 - \sum_{j\in P\setminus\{k\}} x_{vj}, && \forall v \in V, && \text{(9c)}\\
& && y_{uv} = 1 - \sum_{j\in P\setminus\{k\}} x_{uj}x_{vj} - x_{uk}x_{vk}, && \forall \{u,v\} \in E, && \text{(9d)}\\
& && x \in \{0,1\}^{n\times(k-1)}. && && \text{(9e)}
\end{aligned}
$$

It should be noted that for every $v \in V$, we have $x_{vk} \in \{0,1\}$. We define the feasible space of the relaxation of the R-BQO formulation (9) as follows.

$$\mathcal{R}_{\text{R-BQO}} \coloneqq \left\{ (x,y) \in [0,1]^{n\times k} \times \mathbb{R}^m \;\middle|\; (x,y) \text{ satisfies constraints (9b)–(9d) } \right\}.$$

Corollary 1 highlights the relationship between projections of the relaxations of BQO, R-BQO and A-MILO formulations on the x space. The following corollary provides a comparison between the projections of our polytopes on the x space.

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

Proof. The first equality is straightforward. The second equality holds by Proposition 1 and the fact that any x that satisfies constraints (1b)–(1e) of the A-MILO formulation also satisfies constraints (4b) in the BQO formulation. □

For analysis purposes, we lift the dimensionality of the BQO formulation by introducing new z variables.

$$z_{uv} \coloneqq \sum_{j\in P} x_{uj}x_{vj}, \qquad \forall \{u,v\} \in \binom{V}{2}. \tag{10}$$

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

$$\mathcal{R}^{+}_{\text{BQO}} \coloneqq \left\{ (x,y,z) \in [0,1]^{n\times k} \times \mathbb{R}^m \times \mathbb{R}^{\binom{n}{2}} \;\middle|\; (x,y) \in \mathcal{R}_{\text{BQO}} \text{ and } z \text{ satisfies (10) } \right\}.$$

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

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

We show that a projection of the lifted BQO formulation on the z space is at least as strong as the P-MILO formulation.

Proposition 2. $\operatorname{proj}_z \mathcal{R}^{+}_{\text{BQO}} \subseteq \mathcal{R}_{\text{P-MILO}}$.

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

$$\hat{z}_{uv} + \hat{z}_{vw} = \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj} + \sum_{j \in P} \hat{x}_{vj}\hat{x}_{wj}, \tag{11a}$$

$$= \sum_{j \in P} \hat{x}_{vj}(\hat{x}_{uj} + \hat{x}_{wj}), \tag{11b}$$

$$\leq \sum_{j \in P} \hat{x}_{vj}(1 + \hat{x}_{uj}\hat{x}_{wj}), \tag{11c}$$

$$= \sum_{j \in P} \hat{x}_{vj} + \sum_{j \in P} \hat{x}_{vj}(\hat{x}_{uj}\hat{x}_{wj}), \tag{11d}$$

$$= 1 + \sum_{j \in P} \hat{x}_{vj}(\hat{x}_{uj}\hat{x}_{wj}), \tag{11e}$$

$$\leq 1 + \sum_{j \in P} \hat{x}_{uj}\hat{x}_{wj}, \tag{11f}$$

$$= 1 + \hat{z}_{uw}. \tag{11g}$$

Here, equality (11a) holds by the definition of $\hat{z}$. Inequality (11c) holds by Lemma 1. Equality (11e) holds by constraints (4b). Inequality (11f) holds by the fact that $\hat{x}_{vj} \leq 1$. Equality (11g) holds by definition (10).

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

$$\sum_{\{u,v\} \in \binom{Q}{2}} \hat{z}_{uv} = \sum_{\{u,v\} \in \binom{Q}{2}} \sum_{j \in P} \hat{x}_{uj}\hat{x}_{vj}, \tag{12a}$$

$$= \sum_{j \in P} \Big(\sum_{\{u,v\} \in \binom{Q}{2}} \hat{x}_{uj}\hat{x}_{vj} \Big), \tag{12b}$$

$$\geq \sum_{j \in P} \Big(\sum_{u \in Q} \hat{x}_{uj} - 1 \Big), \tag{12c}$$

$$= \sum_{u \in Q} \sum_{j \in P} \hat{x}_{uj} - k, \tag{12d}$$

$$= k + 1 - k = 1. \tag{12e}$$

Here, equality (12a) holds by definition (10). Inequality (12c) holds by Lemma 1. Further, equality (12e) holds by constraints (4b) and because $|Q| = k + 1$.

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

Now, for every $\{u, v\} \in \binom{V}{2}$, we show that $\hat{z}_{uv} \leq 1$.

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

Here, the first equality holds by definition (10). The inequality holds because $\hat{x}_{vj} \leq 1$ for every vertex $v \in V$ and every partition $j \in P$. The last equality holds by constraints (4b). This completes the proof. ☐

We show that the relation in Proposition 2 holds strictly for some instances of the max k-cut problem.

Example 1. Consider the max 2-cut problem on a complete graph with three vertices. For every edge $\{u, v\} \in E$, let $\hat{z}_{uv} = {}^1\!/_3$. Note that $\hat{z}$ is a feasible solution of the P-MILO formulation. By constraints (4b) of the BQO formulation, we have $x_{v2} = 1 - x_{v1}$ for every vertex $v \in V$. By definition (10), for every $\{u, v\} \in E$, we have

$$\begin{aligned} \hat{z}_{uv} = \frac{1}{3} &= x_{v1}x_{u1} + (1 - x_{v1})(1 - x_{u1}), \\ &= 1 - x_{v1} - x_{u1} + 2x_{v1}x_{u1}. \end{aligned} \tag{13}$$

Now we define x_{11} and x_{21} with respect to x_{31}.

$$x_{11} = x_{21} = \frac{{}^2\!/_3 - x_{31}}{1 - 2x_{31}} = \alpha. \tag{14}$$

So, we substitute x_{11} and x_{21} with α in equation (13) as follows.

$$3\alpha^2 - 3\alpha + 1 = 0. \tag{15}$$

The equation has no real solution because the discriminant of equation (15) is negative. This means that one cannot find a feasible x based on mapping (10) for the BQO formulation.

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

$$\mathcal{R}^+_{\text{A-MILO}} \coloneqq \left\{(x, y, z) \in [0,1]^{n \times k} \times [0,1]^m \times [0,1]^{\binom{n}{2}} \;\middle|\; (x, y) \in \mathcal{R}_{\text{A-MILO}} \text{ and } z \text{ satisfies (10)}\right\}.$$

Theorem 1 shows that the A-MILO formulation is at least as strong as the P-MILO formulation under mapping (10). Our result is different from that of Fairbrother and Letchford (2017) who studied a projection of the P-MILO formulation on a subspace of the A-MILO formulation. Further, it is different from the result of Alès and Knippel (2020) who provided a comparison between two extended P-MILO formulations with "representative" variables.

Theorem 1. $\operatorname{proj}_z \mathcal{R}^+_{\text{A-MILO}} \subseteq \mathcal{R}_{\text{P-MILO}}$.

Proof. The proof follows by Corollary 1 and Proposition 2. ☐

Further, the point $\hat{z}$ explained in Example 1 is also a point that belongs to set $\mathcal{R}_{\text{P-MILO}} \setminus \operatorname{proj}_z \mathcal{R}^+_{\text{A-MILO}}$ for the max 2-cut problem.

Although a projection of A-MILO formulation on z space is at least as strong as the P-MILO formulation, an optimal solution (x^*, y^*) of the relaxation of the A-MILO formulation with

$$
\begin{aligned}
& x^*_{uj} = {}^1\!/\!{}_k, && \forall u \in V,\ \forall j \in P,\\
& y^*_e = \min\{2(1 - {}^1\!/\!{}_k), 1\} = 1, && \forall e \in E.
\end{aligned}
$$

Remark 2 shows that the optimal objective value for the relaxation of the P-MILO formulation lies between $m(1 - {}^1\!/\!{}_k)$ and m.

Remark 2. *Let (x^*, y^*) and z^* be optimal solutions of the relaxations of* A-MILO *and* P-MILO *formulations, respectively. Then, for any $k \geq 2$, we have*

$$
m(1 - {}^1\!/\!{}_k) \leq \sum_{e \in E} (1 - z^*_e) \leq m.
$$

Now, we recall the BQO formulation (4). In combinatorial optimization problems, converting a fractional solution of the relaxation of a formulation to a binary solution is a challenging task (Stozhkov et al., 2020). Carlson and Nemhauser (1966) studied optimality conditions for the BQO formulation. Specifically, they proved that one can relax integrality constraints (4d). We propose an algorithm that generalizes their result. In other words, Algorithm 1 converts any feasible solution of the relaxation of the BQO formulation (4) to a feasible solution of the max k-cut problem without decreasing the objective value. This algorithm runs in time $\mathcal{O}(km)$.

Algorithm 1 Conversion of a fractional solution to a binary solution of the BQO model

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

1: $\bar{x} \leftarrow \hat{x}$
2: **for** every vertex $v \in V$ **do**
3:     **for** every $j \in P$ **do**
4:         $b_{vj} \leftarrow \sum_{u \in N_G(v)} \bar{x}_{uj}$
5:     let s be a partition in $\operatorname{argmin}_{j \in P}\{b_{vj}\}$
6:     fix $\bar{x}_{vs}$ to one, and set $\bar{x}_{vj}$ to zero for every $j \in P \setminus \{s\}$
7: **return** $\bar{x}$

Theorem 2. *Let $\hat{x}$ be a solution of the relaxation of the* BQO *formulation* (4)*. Algorithm 1 converts $\hat{x}$ to a binary solution $\bar{x}$ with $g(\bar{x}) \geq g(\hat{x})$.*

Proof. Solution $\bar{x}$ is feasible for the BQO formulation (4) because we assign exactly one partition to every vertex $v \in V$ on line 5 of Algorithm 1.

Let $\delta(v)$ be the set of incident edges of vertex $v \in V$. For every vertex $\bar{v} \in V$, we define $g(x)$ as $g(x) = g_1^{\bar{v}}(x) + g_2^{\bar{v}}(x)$ with

$$
g_1^{\bar{v}}(x) = |E| - \sum_{\{u,v\} \in E \setminus \delta(\bar{v})} \sum_{j \in P} x_{uj} x_{vj}, \qquad g_2^{\bar{v}}(x) = - \sum_{u \in N_G(\bar{v})} \sum_{j \in P} x_{uj} x_{\bar{v}j}. \tag{16}
$$

Let $\bar{v} \in V$ be a vertex with $\hat{x}_{\bar{v}j} < 1$ for every $j \in P$. Then, we have

$$\begin{aligned} g_2^{\bar{v}}(\hat{x}) &= -\sum_{u \in N_G(\bar{v})} \sum_{j \in P} \hat{x}_{uj}\hat{x}_{\bar{v}j}, \\ &= -\sum_{j \in P} \hat{x}_{\bar{v}j} \sum_{u \in N_G(\bar{v})} \hat{x}_{uj}, \\ &= -\sum_{j \in P} \hat{x}_{\bar{v}j} b_{\bar{v}j}, \end{aligned} \tag{17}$$

where $b_{\bar{v}j} = \sum_{u \in N_G(\bar{v})} \hat{x}_{uj}$. Define $b_{\bar{v}}^{\min} \coloneqq \min_{j \in P}\{b_{\bar{v}j}\}$. Because $\hat{x}$ is a feasible solution for the relaxation of the BQO formulation (4), $\sum_{j \in P} \hat{x}_{\bar{v}j} = 1$. By (17), we have

$$\begin{aligned} g_2^{\bar{v}}(\hat{x}) &= -\sum_{j \in P} \hat{x}_{\bar{v}j} b_{\bar{v}j}, \\ &\leq -\sum_{j \in P} \hat{x}_{\bar{v}j} b_{\bar{v}}^{\min}, \\ &= -b_{\bar{v}}^{\min}. \end{aligned} \tag{18}$$

By construction, we have $g_1^{\bar{v}}(\bar{x}) = g_1^{\bar{v}}(\hat{x})$ and $g_2^{\bar{v}}(\bar{x}) = -b_{\bar{v}}^{\min}$. By (18), we have

$$g(\hat{x}) = g_1^{\bar{v}}(\hat{x}) + g_2^{\bar{v}}(\hat{x}) \leq g_1^{\bar{v}}(\hat{x}) - b_{\bar{v}}^{\min} = g_1^{\bar{v}}(\bar{x}) + g_2^{\bar{v}}(\bar{x}) = g(\bar{x}).$$

This completes the proof. □

Corollary 2 (cf. Carlson and Nemhauser (1966))**.** *Suppose $\hat{x}$ is optimal for the relaxation of the* BQO *formulation. Algorithm 1 returns a binary optimal solution.*

Proof. Let x^* represent an optimal solution of the max k-cut problem. By Theorem 2, $\bar{x}$ represents a feasible solution of the max k-cut problem and we have $g(\bar{x}) \geq g(\hat{x})$. Because $\hat{x}$ is an optimal solution of the relaxation of the BQO formulation (4), we have $g(\hat{x}) \geq g(x^*)$. Thus, $g(\bar{x}) = g(x^*)$ and $\bar{x}$ is also an optimal solution of the BQO formulation (4). □

3.2 The unconstrained binary optimization

To solve the max k-cut problem with quantum computers, one needs to formulate it as an unconstrained binary optimization formulation. In this section, we propose the following unconstrained binary optimization formulations: (i) two quadratic unconstrained binary optimization (QUBO) formulations, and (ii) one higher-order unconstrained binary optimization (HUBO) formulation.

First, we propose a QUBO formulation inferred from the BQO formulation. In other words, we move constraints (4b) of the BQO formulation to the objective function and penalize them by a vector $w \in \mathbb{R}_+^n$.

$$\max_{x \in \{0,1\}^{n \times k}} q(x) \coloneqq |E| - \sum_{\{u,v\} \in E} \sum_{j \in P} x_{uj}x_{vj} - \sum_{v \in V} w_v \Big(\sum_{j \in P} x_{vj} - 1 \Big)^2. \tag{19}$$

Similarly, we propose a HUBO formulation as follows.

$$\max_{x \in \{0,1\}^{n\times k}} h(x) \coloneqq |E| - \sum_{\{u,v\}\in E} \sum_{j\in P} x_{uj}x_{vj} - \sum_{v\in V} w_v \prod_{j\in P}(1 - x_{vj}). \tag{20}$$

Note that formulations (19) and (20) do not necessarily provide feasible solutions for the max k-cut problem. So, we propose Algorithm 2 that converts any binary solution of the QUBO formulation (19) and HUBO formulation (20) to a feasible solution of the max k-cut problem. It suffices to show that the resulting solution satisfies constraints (4b). By lines 4–8 of Algorithm 2, we make sure that every vertex is assigned to at least one partition. Further, by lines 9–13 of Algorithm 2, every vertex is assigned to at most one partition. Algorithm 2 takes time $\mathcal{O}(km)$.

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

Require: $(G, \hat{x}, k)$
1: $\bar{x} \leftarrow \hat{x}$
2: $I_0 \coloneqq \{v \in V \mid \sum_{j\in P} \bar{x}_{vj} = 0\}$
3: $I_1 \coloneqq \{v \in V \mid \sum_{j\in P} \bar{x}_{vj} > 1\}$
4: **for** every vertex $v \in I_0$ **do**
5:     **for** every partition $j \in P$ **do**
6:         define $N^j_G(v) \coloneqq \{u \in N_G(v) \mid \bar{x}_{uj} = 1\}$
7:     let s be a partition in $\text{argmin}_{j\in P}\{|N^j_G(v)|\}$
8:     fix $\bar{x}_{vs}$ to one
9: **for** every vertex $v \in I_1$ **do**
10:     define $L \coloneqq \{j \in P \mid \bar{x}_{vj} = 1\}$
11:     let i be a partition in L
12:     **for** every partition $j \in L \setminus \{i\}$ **do**
13:         fix $\bar{x}_{vj}$ to zero
14: **return** $\bar{x}$

One can always set a "big" w in formulations (19) and (20) to ensure that an optimal solution of the unconstrained formulations always represents a feasible solution for the max k-cut problem. However, large values of the elements of w have a detrimental effect on the performance of quantum computing algorithms (Quintero et al., 2021). Theorems 3 and 4 propose smallest values for elements of the penalty vector w.

Theorem 3. *Let $w \in \mathbb{R}^n_+$ be a penalty vector. Suppose $\hat{x}$ is a solution of the* QUBO *formulation* (19)*. Algorithm 2 returns a feasible solution $\bar{x}$ of the* BQO *formulation* (4) *with $q(\bar{x}) \geq q(\hat{x})$ if for every vertex $v \in V$,*

$$w_v \geq \left\lfloor \frac{d(v)}{k} \right\rfloor.$$

Proof. First, we define $q(x) = q_1(x) + q_2(x)$ with

$$q_1(x) = |E| - \sum_{\{u,v\}\in E} \sum_{j\in P} x_{uj}x_{vj}, \quad \text{and} \quad q_2(x) = -\sum_{v\in V} w_v \Big(\sum_{j\in P} x_{vj} - 1\Big)^2.$$

First, suppose that $\hat{x}$ represents a binary solution in which there exists a vertex v with multiple partitions; i.e., $\sum_{j\in P} \hat{x}_{vj} > 1$. By line 11 of Algorithm 2, let i be a partition in set L, where $L := \{j \in P \mid \hat{x}_{vj} = 1\}$. Now by line 12–13 of the algorithm, we fix $\hat{x}_{vj}$ for every $j \in L \setminus \{i\}$. Because this procedure can eliminate some of the bilinear terms of $q_1(\cdot)$, we have $q_1(\bar{x}) \geq q_1(\hat{x})$. Further, $q_2(\bar{x}) \geq q_2(\hat{x})$ because the procedure increases $q_2(\cdot)$ by removing the penalty term corresponding to vertex v. Hence, we have $q(\bar{x}) \geq q(\hat{x})$.

Now, assume that $\hat{x}$ is an optimal solution of formulation (19) in which there exists a vertex v with no partition; i.e., $\sum_{j\in P} \hat{x}_{vj} = 0$. For ease of notation, we define set $N^j_G(v)$ as

$$N^j_G(v) := \{u \in N_G(v) \mid \bar{x}_{uj} = 1\},$$

and let $d^j_G(v) = |N^j_G(v)|$ for every $j \in P$. Note that we have $d(v) \geq \sum_{j\in P} d^j_G(v)$ because there might be a neighbor $u \in N_G(v)$ with no assigned partition. By line 7 of the algorithm, let s be a partition with minimum value of $d^j_G(v)$ among all $j \in P$; i.e., $d^s_G(v) \leq d^j_G(v)$ for all $j \in P$. Now, we fix $x_{vs} = 1$ in solution $\hat{x}$ and call the resulting solution $\bar{x}$. Then, the following equalities hold

$$q_1(\bar{x}) = q_1(\hat{x}) - d^s_G(v), \quad \text{and} \quad q_2(\bar{x}) = q_2(\hat{x}) + w_v.$$

Because $d_G(v) \geq k d^s_G(v)$, we have

$$w_v \geq \left\lfloor \frac{d(v)}{k} \right\rfloor \geq \left\lfloor \frac{k d^s_G(v)}{k} \right\rfloor = d^s_G(v). \tag{21}$$

Note that $q(\bar{x}) = q(\hat{x}) + w_v - d^s_G(v)$. Because $w_v - d^s_G(v) \geq 0$ by (21), we have $q(\bar{x}) \geq q(\hat{x})$. Thus, the proof is complete. $\square$

Corollary 3. *Suppose $\hat{x}$ is optimal for the* QUBO *model* (19) *with $w_v \geq \left\lfloor \frac{d(v)}{k} \right\rfloor$ for all $v \in V$. Algorithm 2 returns a binary optimal solution of the* BQO *model* (4).

Proof. Let x^* represent an optimal solution of the max k-cut problem. Let $\bar{x}$ be a point returned by Algorithm 2 applied on solution $\hat{x}$. Since $\hat{x}$ is an optimal solution of the QUBO formulation (19), we have (i) $q(\hat{x}) \geq q(\bar{x})$, and (ii) $q(\hat{x}) \geq q(x^*)$. By Theorem 3, we have (iii) $q(\bar{x}) \geq q(\hat{x})$. By (i) and (iii), we have $q(\hat{x}) = q(\bar{x})$. Hence, by (ii) we have $q(\bar{x}) \geq q(x^*)$. Further, because of feasibility of $\bar{x}$ for the BQO formulation (4), we have $q(\bar{x}) \leq q(x^*)$. Hence, $q(\hat{x}) = q(\bar{x}) = q(x^*)$ and $\bar{x}$ is also an optimal solution of the BQO formulation (4). $\square$

Similarly, we can prove Theorem 4 and Corollary 4 for the HUBO formulation (20).

Theorem 4. *Let $w \in \mathbb{R}^n_+$ be a penalty vector. Suppose $\hat{x}$ is a solution of the* HUBO *formulation* (20)*. Algorithm 2 returns a feasible solution $\bar{x}$ of the* BQO *formulation* (4) *with $h(\bar{x}) \geq h(\hat{x})$ if for every vertex $v \in V$,*

$$w_v \geq \left\lfloor \frac{d(v)}{k} \right\rfloor.$$

Corollary 4. *Suppose $\hat{x}$ is optimal for the* HUBO *model* (20) *with $w_v \geq \left\lfloor \frac{d(v)}{k} \right\rfloor$ for all $v \in V$. Algorithm 2 returns a binary optimal solution of the* BQO *model* (4)*.*

It should be noted that if $w_v > \left\lfloor \frac{d(v)}{k} \right\rfloor$ for every vertex $v \in V$, then an optimal solution of both the HUBO formulation (20) and the QUBO formulation (19) is also a feasible solution for the BQO formulation (4). Further, the following example shows that there exist instances of the max k-cut problem for which if condition $w_v < \left\lfloor \frac{d(v)}{k} \right\rfloor$ holds for some vertex $v \in V$, then Algorithm 2 does not provide a feasible solution that satisfies the statement of Theorems 3 and 4.

Example 2. Consider the max 2-cut problem for a graph illustrated in Figure 3. The optimal objective value of this problem is 4. Let x^* be an optimal solution of the max 2-cut problem with $x^*_{11} = x^*_{22} = x^*_{32} = x^*_{41} = 1$ (See Figure 3).

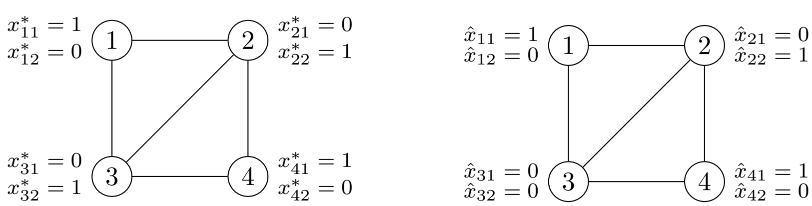

Figure 3: Optimal solution x^* (left) and infeasible solution $\hat{x}$ (right).

Then, the QUBO formulation (19) of the max 2-cut problem is written as follows.

$$q(x) = 5 - \sum_{\{u,v\} \in E} \left(x_{u1}x_{v1} + x_{u2}x_{v2}\right) - \sum_{v \in V} w_v \left(x_{v1} + x_{v2} - 1\right)^2 \tag{22}$$

Further, the HUBO formulation (20) of the max 2-cut problem is written as follows.

$$h(x) = 5 - \sum_{\{u,v\} \in E} \left(x_{u1}x_{v1} + x_{u2}x_{v2}\right) - \sum_{v \in V} w_v \left(1 - x_{v1}\right)\left(1 - x_{v2}\right) \tag{23}$$

First, let $w_1 = w_4 = \left\lfloor \frac{2}{2} \right\rfloor = 1$ and $w_2 = w_3 = \left\lfloor \frac{3}{2} \right\rfloor = 1$. Then, we have $q(x^*) = h(x^*) = 4$.

Now, we change w_3 from 1 to $1 - \epsilon$ for some $\epsilon > 0$. Then, $\hat{x}$ is an optimal solution for both QUBO and HUBO formulations (22) and (23) such that $\hat{x}_{11} = \hat{x}_{22} = \hat{x}_{41} = 1$. Further, $\hat{x}_{31} = \hat{x}_{32} = 0$ (See Figure 3). However, this implies that $\hat{x}$ is an infeasible solution for the max 2-cut problem with $q(\hat{x}) = h(\hat{x}) = 4 + \epsilon$ and $q(\hat{x}) = h(\hat{x}) > q(x^*) = h(x^*)$. Hence, an inappropriate choice of w might not provide a solution with the optimal objective value for the max k-cut problem.

Similar to the R-BQO, now we propose a reduced QUBO (R-QUBO) model. First, we define $\bar{P} \coloneqq P \setminus \{k\}$. Then, we reformulate the max k-cut problem as the following R-QUBO formulation with $n(k-1)$ binary variables.

$$
\begin{aligned}
\max_{x \in \{0,1\}^{n\times(k-1)}} \bar{q}(x) \coloneqq & |E| - \sum_{\{u,v\}\in E} \Big(\sum_{j\in\bar{P}} x_{uj}x_{vj} + \big(1 - \sum_{j\in\bar{P}} x_{uj}\big)\big(1 - \sum_{j\in\bar{P}} x_{vj}\big) \Big) \\
& - \sum_{v\in V} w_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj}.
\end{aligned} \tag{24}
$$

It is worth noting that for $k = 2$, the penalty term disappears because $\binom{\bar{P}}{2} = \varnothing$. We can also rewrite the R-QUBO formulation (24) as follows

$$
\begin{aligned}
\max_{x \in \{0,1\}^{n\times(k-1)}} \bar{q}(x) = & \sum_{\{u,v\}\in E} \Big(\sum_{j\in\bar{P}} x_{uj}(1 - x_{vj}) + \big(1 - \sum_{j\in\bar{P}} x_{uj}\big) \sum_{j\in\bar{P}} x_{vj} \Big) \\
& - \sum_{v\in V} w_v \sum_{\{i,j\}\in\binom{\bar{P}}{2}} x_{vi}x_{vj}.
\end{aligned} \tag{25}
$$

Algorithm 3 converts any infeasible binary solution of the R-BQO formulation to a feasible solution. Algorithm 3 has complexity $\mathcal{O}(kn)$.

Algorithm 3 Conversion of a binary infeasible solution of R-BQO to a feasible solution

Require: $(G, \hat{x}, k)$
  $\bar{x} \leftarrow \hat{x}$
  $I \coloneqq \{v \in V \mid \sum_{j\in\bar{P}} \bar{x}_{vj} > 1\}$
  for every vertex $v \in I$ **do**
    select partition $i \in \bar{P}$ where $\bar{x}_{vi} = 1$
    for every partition $j \in \bar{P} \setminus \{i\}$ **do**
      fix $\bar{x}_{vj}$ to zero
  return $\bar{x}$

Theorem 5. *Let $w \in \mathbb{R}^n_+$ be a penalty vector. Suppose $\hat{x}$ is a solution of the* R-QUBO *formulation* (25). *Algorithm 3 returns a feasible solution $\bar{x}$ of the* R-BQO *formulation* (9) *with $\bar{q}(\bar{x}) \geq \bar{q}(\hat{x})$ if for every vertex $v \in V$,*

$$w_v \geq d(v).$$

Proof. Let $\hat{x}$ represent a binary solution of the R-QUBO formulation with a vertex v assigned to more than one partition. First, we define $\bar{P}_v$ as follows

$$\bar{P}_v \coloneqq \{j \in \bar{P} \mid \hat{x}_{vj} = 1\}.$$

Further, we define $t \coloneqq |\bar{P}_v|$. Here, we have $t > 1$. Let partition $j \in \bar{P}_v$. We fix $\hat{x}_{vj}$

to zero and call the resulting solution $\bar{x}$. Then, we have

$$\begin{aligned}\bar{q}(\bar{x}) &= \bar{q}(\hat{x}) + \sum_{u \in N_G(v)} \Big(\hat{x}_{uj} + \big(\sum_{j \in \bar{P}} \hat{x}_{uj} - 1\big)\Big) + (t-1)w_v, \\ &= \bar{q}(\hat{x}) + \sum_{u \in N_G(v)} \Big(\hat{x}_{uj} + \sum_{j \in \bar{P}} \hat{x}_{uj}\Big) - d(v) + (t-1)w_v \\ &\geq \bar{q}(\hat{x}) + 0 - d(v) + (t-1)w_v \\ &\geq \bar{q}(\hat{x}).\end{aligned}$$

Here, the first equality holds by the R-QUBO formulation (25). Further, the last inequality holds because $t > 1$ and $w_v \geq d(v)$. Hence, $\bar{q}(\bar{x}) \geq \bar{q}(\hat{x})$. It implies that if we iteratively apply lines 4-6 of Algorithm 3, then the objective value $\bar{q}(\cdot)$ does not decrease. This completes the proof. $\square$

Corollary 5. *Suppose $\hat{x}$ is optimal for the* R-QUBO *model* (25)*, where $w_v \geq d(v)$ for all $v \in V$. Algorithm 3 returns a binary optimal solution of* R-BQO *model* (9).

Proof. The proof is similar to the proof of Corollary 3. $\square$

It should be noted that if $w_v > d(v)$ for every vertex $v \in V$, then an optimal solution of the R-QUBO formulation (25) is also a feasible solution for the R-BQO formulation (9). Example 3 shows that there are some instances of the max k-cut problem for which if we have $w_v < d(v)$ for some vertex $v \in V$, then the result of Theorem 5 does not hold.

Example 3. Consider the max 3-cut problem for the graph illustrated in Figure 4. Let x^* be an optimal solution to the problem. Note that the optimal objective value is 3. Here, vertex 3 belongs to partition 2, and other vertices belong to partition 1.

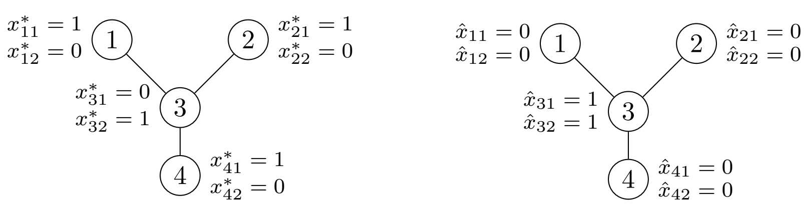

Figure 4: Optimal solution x^* (left) and infeasible solution $\hat{x}$ (right).

The corresponding R-QUBO formulation is as follows

$$\bar{q}(x) = \sum_{\{u,v\} \in E} \Big(\sum_{j \in \{1,2\}} x_{uj}(1 - x_{vj}) + \big(1 - \sum_{j \in \{1,2\}} x_{uj}\big) \sum_{j \in \{1,2\}} x_{vj}\Big) - \sum_{v \in V} w_v x_{v1} x_{v2}.$$

Let $w_3 = d_3 = 3$ and for every $v \in \{1, 2, 4\}$, let $w_v = d(v) = 1$. Here, we have $\bar{q}(x^*) = 3$. Now, suppose we change w_3 to $3 - \epsilon$ for some $\epsilon > 0$. Let $\hat{x}$ be an optimal solution of the modified R-QUBO with $\hat{x}_{31} = \hat{x}_{32} = 1$, and $\hat{x}_{vj} = 0$ for $v \in \{1, 2, 4\}$ and $j \in \{1, 2\}$. Then, we have $\bar{q}(\hat{x}) = 3 + \epsilon$ for the modified R-QUBO formulation. Because $\bar{q}(\hat{x}) \geq \bar{q}(x^*)$, an optimal solution of the max 3-cut is not optimal for the modified R-QUBO formulation.

4 Computational improvements

To enhance the performance of our formulations, we discuss (i) preprocessing algorithms and (ii) approaches for strengthening the formulations.

4.1 Preprocessing algorithms

This section provides a short discussion on known reduction and decomposition techniques for the max k-cut problem. First, we provide a basic definition from the graph theory context.

Definition 1 (k-core (see e.g. Gross and Yellen, 2003))**.** The k-core of a graph is the largest subgraph with all its vertex degrees at least k.

Batagelj and Zaversnik (2003) proposed an algorithm that finds the k-core of a graph by removing vertices with degrees less than k, iteratively. This algorithm runs in $\mathcal{O}(m)$. One can solve the max k-cut problem only on the k-core of a graph instead of solving it on the original graph. The vertices that are not in the k-core can be assigned in a post-process procedure. This procedure acts in the reverse direction of the k-core algorithm proposed by Batagelj and Zaversnik, 2003. Then, all edges that are not in the k-core will be cut-edges. Méndez-Díaz and Zabala (2006) employed a similar procedure for solving the graph coloring problem. Further, one can employ this idea to solve the max k-cut problem in time $\mathcal{O}(m)$ for all grid instances of G. Wang and Hijazi (2020) with $k > 2$.

Further, Fairbrother et al. (2018) proposed a procedure that decomposes the problem based on cut vertices and their corresponding biconnected components. Definitions of cut vertices and biconnected components are explained as follows.

Definition 2 (cut vertex (see e.g. Gross and Yellen, 2003))**.** A vertex $v \in V$ is a cut vertex if $G[V \setminus \{v\}]$ has at least two components.

Definition 3 (biconnected component (see e.g. Gross and Yellen, 2003))**.** A biconnected component Q is a maximal biconnected subgraph; i.e., a maximal subgraph that has no cut vertex.

Now define $\Theta = (Q_1, Q_2, \ldots, Q_\ell)$ as an ordering of biconnected components of graph G such that each biconnected component has a common cut vertex with the previous ones. One can find such an ordering by traversing the block tree of graph G. Algorithm 4 returns an optimal solution of the max k-cut problem on graph G. In this algorithm, subroutine *solve*(G, k, v, a_v) solves the max k-cut problem on graph G with a predefined assignment a_v for vertex v. It returns an optimal assignment a^* and the optimal number of cut-edges f^*. Further, subroutine *modify*$(a^*, \hat{a})$ modifies the assignment a^* based on the assignment of vertices in $\hat{a}$ and returns the updated assignment a^*.

Algorithm 4 Biconnected component decomposition

Require: (G, k)
1: Find an ordering of Θ
2: $a^* \leftarrow \{0\}^n$
3: select $v \in Q_1$ and $a_v \in P$
4: $(\hat{a}, \hat{f}) \leftarrow \text{solve}(G[Q_1], k, v, a_v)$
5: $a^* \leftarrow \text{modify}(a^*, \hat{a})$
6: $f^* \leftarrow \hat{f}$
7: **for** $i \in \{2, \ldots, \ell\}$ **do**
8: $\quad v \leftarrow Q_i \cap Q_{i-1}$
9: $\quad (\hat{a}, \hat{f}) \leftarrow \text{solve}(G[Q_i], k, v, a^*_v)$
10: $\quad a^* \leftarrow \text{modify}(a^*, \hat{a})$
11: $\quad f^* \leftarrow f^* + \hat{f}$
12: **return** f^* and a^*

4.2 Strengthening the formulations

The objective function of the BQO formulation (4) is indefinite; i.e., formulation (4) is a non-convex optimization model. This section discusses the convexification techniques of the objective function (4a). For the ease of notation, we rewrite the BQO formulation (4) of the max k-cut problem as the BQO formulation (26) of the min k-partition problem which is equivalent to the max k-cut problem.

$$\begin{aligned} \min \quad & \sum_{j \in P} \mathbf{x}_j^T A \mathbf{x}_j, \\ \text{s.t.} \quad & \sum_{j \in P} x_{vj} = 1, \qquad \forall v \in V, \\ & x_{vj} \in \{0, 1\}, \quad \forall v \in V,\ j \in P, \end{aligned} \tag{26}$$

where A is the adjacency matrix of graph G and $\mathbf{x}_j \in \{0,1\}^n$ represents the assignment vector of partition $j \in P$. We can rewrite the objective function of the BQO formulation (26) as follows

$$\sum_{j \in P} \mathbf{x}_j^T A \mathbf{x}_j + \sum_{j \in P} \Big(\mathbf{x}_j^T D \mathbf{x}_j - d^T \mathbf{x}_j \Big), \tag{27}$$

where the diagonal matrix $D \coloneqq \text{diag}(d)$ for a given $d \in \mathbb{R}^n$ (Hammer and Rubin, 1970). To convexify function (27), we seek to find the diagonal matrix D such that (i) the Hessian matrix $H = A + D$ becomes positive semidefinite and (ii) the relaxation of the reformulated BQO formulation (26) provides a tight bound. In other words, we are to solve the following optimization problem.

$$\min \left\{ \text{Tr}(D) \;\middle|\; A + D \succcurlyeq 0 \right\}, \tag{28}$$

where $\mathrm{Tr}(D)$ is the trace of matrix D. We can analytically specify the eigenvalues of the adjacency matrix for some graph structures such as complete graphs and bipartite graphs, to name a few. This means that we can easily solve (28) on such graphs. Here, we define $D = -\lambda^A_{\min} I$ with $\lambda^A_{\min}$ as the minimum eigenvalue of adjacency matrix A and I as the identity matrix. To obtain $\lambda^A_{\min}$, we employ power iteration method (see e.g. Anton and Rorres, 2013, section 10.3). The power iteration method gives an eigenvalue with the maximum absolute value. Now, we define $F \coloneqq uI - A$, where

$$u = \sqrt{2m\Big(\frac{n-1}{n}\Big)}$$

is an upper bound for $\lambda^A_{\max}$ (Wilf, 1967). Then, $\lambda^A_{\min} = u - \lambda^F_{\max}$ where $\lambda^F_{\max}$ is the maximum eigenvalue of matrix F that can be obtained by the power iteration method.

Similarly, we can determine diagonal matrix D such that the quadratic function (27) is negative semidefinite, where $D = -\lambda^A_{\max} I$. To do so, we use

$$\ell = -\sqrt{\frac{n}{2}\left\lfloor\frac{(n+1)}{2}\right\rfloor}$$

as a lower bound for $\lambda^A_{\min}$ (Hong, 1993). Let us define $W \coloneqq A - \ell I$. Then, we can find $\lambda^A_{\max} = \lambda^W_{\max} + \ell$ by applying the power iteration method on matrix W.

Further, we can reformulate the BQO formulation (26) by substituting its objective function with function (27). Because objective function (27) is concave and constraints of the BQO formulation (26) are linear, any optimal solution lies on the boundary of the feasible set. Hence, optimal solutions of the relaxation of the reformulated BQO formulation (4) are binary points. It should be noted that the optimal objective value of the relaxation of the reformulated BQO formulation with objective function (27) is not necessarily equal to the optimal objective value of the relaxation of the BQO formulation (26).

5 Conclusion

This paper studied different formulations of the max k-cut problem that can be solved by classic or quantum solvers. We analytically compared all the classic linear and quadratic formulations with each other. We also proposed procedures for converting a feasible fractional solution of the constrained quadratic formulation to a feasible binary solution. Further, we proposed three unconstrained quadratic formulations that quantum solvers can handle. Finally, we provided algorithms and procedures to improve the computational performance of all formulations. In the future, we will assess the performance of the studied formulations on an extensive set of instances.

Acknowledgement

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

References

Aardal, Karen I., Stan P. M. van Hoesel, Arie M. C. A. Koster, Carlo Mannino, and Antonio Sassano (2007). "Models and solution techniques for frequency assignment problems". *Annals of Operations Research* 153.1, pp. 79–129. DOI: 10.1007/s10479-007-0178-0.

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

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

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

Anton, Howard and Chris Rorres (2013). *Elementary linear algebra: applications version*. John Wiley & Sons.

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

Aslidis, A (1990). "Minimizing of overstowage in container ship operations". *Operational Research* 90, pp. 457–471.

Avriel, Mordecai and Michal Penn (1993). "Exact and approximate solutions of the container ship stowage problem". *Computers & industrial engineering* 25.1-4, pp. 271–274. DOI: 10.1016/0360-8352(93)90273-Z.

Avriel, Mordecai, Michal Penn, Naomi Shpirer, and Smadar Witteboon (1998). "Stowage planning for container ships to reduce the number of shifts". *Annals of Operations Research* 76, pp. 55–71. DOI: 10.1023/A:1018956823693.

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

Batagelj, Vladimir and Matjaz Zaversnik (2003). "An $O(m)$ algorithm for cores decomposition of networks". *arXiv preprint*. URL: https://arxiv.org/abs/cs/0310049.

Beck, Amir and Dror Pan (2017). "A branch and bound algorithm for nonconvex quadratic optimization with ball and linear constraints". *Journal of Global Optimization* 69.2, pp. 309–342. DOI: 10.1007/s10898-017-0521-1.

Belotti, Pietro, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter (2009). "Branching and bounds tightening techniques for non-convex MINLP". *Optimization Methods & Software* 24.4-5, pp. 597–634. DOI: 10.1080/10556780903087124.

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

Bollapragada, Srinivas and Marc Garbiras (2004). "Scheduling commercials on broadcast television". *Operations Research* 52.3, pp. 337–345. DOI: 10.1287/opre.1030.0083.

Bonami, Pierre, Oktay Günlük, and Jeff Linderoth (2018). "Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods". *Mathematical Programming Computation* 10.3, pp. 333–382. DOI: 10.1007/s12532-018-0133-x.

Bonami, Pierre, Andrea Lodi, Jonas Schweiger, and Andrea Tramontani (2019). "Solving quadratic programming by cutting planes". *SIAM Journal on Optimization* 29.2, pp. 1076–1105. DOI: 10.1137/16M107428X.

Butenko, Sergiy (2003). "Maximum independent set and related problems, with applications". PhD thesis. University of Florida. URL: http://etd.fcla.edu/UF/UFE0001011/butenko_s.pdf.

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

Chen, Chen, Alper Atamtürk, and Shmuel S. Oren (2017). "A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables". *Mathematical Programming* 165.2, pp. 549–577. DOI: 10.1007/s10107-016-1095-2.

Chen, Jieqiu and Samuel Burer (2012). "Globally solving nonconvex quadratic programming problems via completely positive programming". *Mathematical Programming Computation* 4.1, pp. 33–52. DOI: 10.1007/s12532-011-0033-9.

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

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

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

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

Deutsch, David and Richard Jozsa (1992). "Rapid solution of problems by quantum computation". *Proceedings of the Royal Society of London. Series A: Mathemat-*

ical and Physical Sciences 439.1907, pp. 553–558. DOI: 10.1098/rspa.1992.0167.

Domingo-Ferrer, J. and J. M. Mateo-Sanz (2002). "Practical data-oriented microaggregation for statistical disclosure control". *IEEE Transactions on Knowledge and Data Engineering* 14.1, pp. 189–201. DOI: 10.1109/69.979982.

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

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

Elf, Matthias, Michael Jünger, and Giovanni Rinaldi (2003). "Minimizing breaks by maximizing cuts". *Operations Research Letters* 31.5, pp. 343–349. DOI: 10.1016/S0167-6377(03)00025-7.

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

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

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

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

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

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

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

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

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

Ghaddar, Bissan, Miguel F. Anjos, and Frauke Liers (2011). "A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem". *Annals of Operations Research* 188.1, pp. 155–174. DOI: 10.1007/s10479-008-0481-4.

Gross, J.L. and J. Yellen (2003). *Handbook of Graph Theory*. Discrete Mathematics and Its Applications. CRC Press. URL: https://books.google.com/books?id=mKkIGIea%5C_BkC.

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

Gurobi Optimization, LLC (2021). *Gurobi Optimizer Reference Manual*. Version 9.1. URL: http://www.gurobi.com.

Hadfield, Stuart, Zhihui Wang, Bryan O'Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas (2019). "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz". *Algorithms* 12.2. DOI: 10.3390/a12020034.

Hammer, Peter L and Abraham A Rubin (1970). "Some remarks on quadratic programming with 0-1 variables". *RAIRO-Operations Research-Recherche Opérationnelle* 4.V3, pp. 67–79. URL: http://www.numdam.org/article/RO_1970__4_3_67_0.pdf.

Hong, Yuan (1993). "Bounds of eigenvalues of graphs". *Discrete Mathematics* 123.1, pp. 65–74. ISSN: 0012-365X. DOI: 10.1016/0012-365X(93)90007-G.

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

Liberti, Leo and Constantinos C Pantelides (2006). "An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms". *Journal of Global Optimization* 36.2, pp. 161–189. DOI: 10.1007/s10898-006-9005-4.

Liers, Frauke, Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi (2004). "Computing exact ground states of hard Ising spin glass problems by branch-and-cut". *New optimization algorithms in physics* 50.47-68, p. 6. DOI: 10.1002/3527603794.ch4.

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

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

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

Mitchell, John E. (2003). "Realignment in the national football league: Did they do it right?" *Naval Research Logistics (NRL)* 50.7, pp. 683–701. DOI: 10.1002/nav.10084.

Nielsen, Michael A. and Isaac L. Chuang (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. USA: Cambridge University Press. ISBN: 1107002176. URL: http://csis.pace.edu/~ctappert/cs837-19spring/QC-textbook.pdf.

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

Pajouh, Foad Mahdavi, Balabhaskar Balasundaram, and Oleg A Prokopyev (2013). “On characterization of maximal independent sets via quadratic optimization”. *Journal of Heuristics* 19.4, pp. 629–644. DOI: 10.1007/s10732-011-9171-5.

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

Pinter, Ron Y. (Oct. 1984). “Optimal layer assignment for interconnect”. *Advances in VLSI and Computer Systems* 1.2, pp. 123–137. URL: https://dl.acm.org/doi/abs/10.5555/2334.2335.

Preskill, John (Aug. 2018). “Quantum Computing in the NISQ era and beyond”. *Quantum* 2, p. 79. DOI: 10.22331/q-2018-08-06-79.

Quintero, Rodolfo, David Bernal, Tamás Terlaky, and Luis F Zuluaga (2021). “Characterization of QUBO reformulations for the maximum k-colorable subgraph problem”. *arXiv preprint*. URL: https://arxiv.org/abs/2101.09462.

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

Ruen-Wu Chen, Y. Kajitani, and Shu-Park Chan (1983). “A graph-theoretic via minimization algorithm for two-layer printed circuit boards”. *IEEE Transactions on Circuits and Systems* 30.5, pp. 284–299. DOI: 10.1109/TCS.1983.1085357.

Şeker, Oylum, Neda Tanoumand, and Merve Bodur (2020). “Digital Annealer for quadratic unconstrained binary optimization: a comparative performance analysis”. *arXiv preprint*. URL: https://arxiv.org/abs/2012.12264.

Sherali, Hanif D. and Cihan H. Tuncbilek (1995). “A reformulation-convexification approach for solving nonconvex quadratic programming problems”. *Journal of Global Optimization* 7.1, pp. 1–31. DOI: 10.1007/BF01100203.

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

Smelyanskiy, V.N., U.V. Toussaint, and D.A. Timucin (2001). “Simulations of the adiabatic quantum optimization for the set partition problem”. *arXiv preprint*. URL: https://arxiv.org/abs/quant-ph/0112143.

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

Sousa, Vilmar Jefté Rodrigues de, Miguel F Anjos, and Sébastien Le Digabel (2018). “Computational study of valid inequalities for the maximum k-cut problem”.

Annals of Operations Research 265.1, pp. 5–27. DOI: 10.1007/s10479-017-2448-9.

Stozhkov, Vladimir, Austin Buchanan, Sergiy Butenko, and Vladimir Boginski (2020). "Continuous cubic formulations for cluster detection problems in networks". *Mathematical Programming*. DOI: 10.1007/s10107-020-01572-4.

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

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

Wilf, H. S. (1967). "The eigenvalues of a graph and its chromatic number". *Journal of the London Mathematical Society* 1.1, pp. 330–332. DOI: 10.1112/jlms/s1-42.1.330.