INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

QUBO Formulations of Combinatorial Optimization Problems for Quantum Computing Devices

RODOLFO A. QUINTERO[1] AND LUIS F. ZULUAGA[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

ISE Technical Report 23T-016

LEHIGH UNIVERSITY

QUBO Formulations of Combinatorial Optimization Problems for Quantum Computing Devices

Rodolfo A. Quintero *

Luis F. Zuluaga †

MSC2020: 81P68, 68Q12

Article Outline

Keywords

Quantum Computing; Quadratic Unconstrained Binary Optimization; QUBO; Ising model; Combinatorial Optimization; Stable Set Problem; Knapsack Problem

1 Introduction

Quadratic unconstrained binary optimization (QUBO) refers to a class of optimization problems that have been studied independently in different scientific areas such as optimization [see, e.g., 49], physics [see, e.g., 5], and combinatorics [see, e.g., 25]. Formally, a QUBO model or problem is defined as finding the

$$\min_{x \in \{0,1\}^n} x^\top Qx + c^\top x \tag{1}$$

where $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, and $c \in \mathbb{R}^n$. Note that since $x_i^2 = x_i$, for every $i \in \{1, \dots, n\}$, one can rewrite $x^\top Qx + c^\top x = x^\top(Q + \operatorname{diag}(c))x$, where $\operatorname{diag}(c)$ is the diagonal matrix whose diagonal elements are given by the entries of the vector c. Similarly, an equivalent representation of the QUBO problem (1) that commonly appears in the optimization and physics literature is obtained when the binary feasible set of the problem is defined using the values -1 and 1 (instead of 0 and 1); that is the feasible set of the problem is given by $x \in \{-1, 1\}^n$. The equivalence follows after applying the affine transformation $x \to 2x - 1$, which maps $\{0, 1\}^n$ to $\{-1, 1\}^n$. In this case, problem (1) is also referred to as the *Ising model* [see, e.g., 6]. Further, it is clear that when minimization is replaced by maximization in (1), the resulting problem is equivalent to a QUBO by simple taking the negative of the objective funtion.

The QUBO model (1) captures a wide range of *integer* and *combinatorial optimization* (COPT) problems; that is, optimization problems where some or all of the decision variables are restricted to be integers [see,

*Department Industrial and Systems Engineering, Lehigh University, USA roq219@lehigh.edu

†Department of Industrial and Systems Engineering, Lehigh University, USA, luis.zuluaga@lehigh.edu

e.g., 46]. In particular, the *max cut* problem [see, e.g., 22] is a QUBO problem (see Section 2 for details). Further, through reformulation techniques (see Section 2), that in particular involve the penalization of potential constraints in the problem, other COPT problems are equivalent to a QUBO problem. This is the case for well-known COPT problems such as the knapsack problem; the assignment and matching problems; the set covering, set packing and set partitioning problems; the traveling salesman problem; the facility location and network flow problems; among many others. It is worth to recall at this point that, in general, COPT problems are NP-Hard [see, e.g., 35]; that is, there is no known polynomial-time algorithm that can be used to solve them. Thus, QUBO models provide an alternative way to address the solution of these COPT problems that arise in important applications in all areas of knowledge.

An important feature of using QUBO models to solve COPT problems is that although QUBO solution methods are not tailored to exploit the structure of a particular class of COPT problems, the quality of their solutions is in many cases as good or even better than the solutions obtained by the best specialized methods, and are obtained with an efficiency that is comparable or even better than the efficiency of the best specialized methods [36].

The first systematic treatments on methods and theory to solve QUBO problems date back to the 60s with the work done by Hammer and Rudeanu [28], Ginsburgh and Peeterssen [19], and H. Kunzi and W. Oettli [37]. In particular, Hammer's work on the theory of quadratic programming with binary variables [see, e.g, 27] and pseudo boolean optimization [see, e.g., 4, 26, 29] influenced significantly the future work done on QUBO problems during the next decades.

Several methods have been proposed in the literature to solve QUBO problems with classical computers (as opposed to quantum computing devices). For instance, different variations of *branch and bound*, or *branch and prune* were part of the first attempts to succesfully solve QUBO problem instances [see, e.g., 24, 49, 66]. Also, continuous or *semidefinite optimization* techniques, together with implicit enumeration and cutting plane techniques, have been used to effectively solve QUBO problem instances [see, e.g., 30, 45, 48]. For most of the COPT problems of interest, the matrix Q in (1) obtained after reformulating the problem as a QUBO is not positive semidefinite; that is, the objective in (1) is not convex. However, for some QUBO problems arising in finance and project management [see, e.g., 44, 58], the matrix Q is positive semidefinite. In that case, one of the best methodologies to solve the problem is to use a combination of branch and bound and cut techniques, together with *interior-point methods* to solve the problem [see, e.g., 2]. When Q is not positive semidefinite, there have been ongoing advances in algorithms to solve more general classes of problems that encompass QUBO problems using a combination of non-linear and convex optimization techniques [see, e.g., 3, 7]. Finally, it is important to mention that heuristics and meta heuristics have also been used to find good approximate solutions when solving medium-large QUBO problem instances. Algorithms like Tabu search, genetic algorithms and simulated annealing are a very common tools in this category [see, e.g., 21, 64].

The recent development of quantum computing devices [see, e.g., 51] that take advantage of the quantum mechanical properties of subatomic particles has created a renewed interest in QUBO reformulations of COPT problems. This is due to the fact that both quantum annealing devices [see, e.g., 9, 17, 33, 40], and algorithms (such as the *quantum approximate optimization algorithm* (QAOA)) for gate-based quantum computers [see, e.g., 15, 65] are able to address the solution of QUBO problems. This opens the door to solve NP-Hard combinatorial optimization problems that can be reformulated as QUBO problems in a fundamentally different way. Although it is not expected that quantum computing devices alone would allow to solve this class of problems in polynomial time [51], there is a strong belief that a combination of quantum computing devices and classical computing devices will provide, complexity and numerically wise, exponential speed-ups in the solution time for these problems [see, e.g., 23, 50].

With this discussion about the relevance of obtaining QUBO reformulations of COPT problems at hand, Section 2 will illustrate the basic techniques used to obtain the desired QUBO reformulations. Further, Section 3 shows how QUBO reformulations of COPT problems can be improved, from a computational point of view, in terms of the number of binary variables and the penalty parameters used to construct the QUBO reformulations. These improvements are particularly relevant when solving these QUBO problems on quantum computing devices, due to their current limitations on the size of the problems they can solve as well as the noisy nature of the devices [see, e.g., 51]. The article finishes in Section 4, with a brief discussion on some of the research avenues that are currently being explored to improve the capacity of quantum devices to solve QUBO problems.

In what follows, we will be focusing on the role of QUBO problems in quantum computing. For the reader interested in further information about QUBO problems from a classical computing and optimization point of view, we direct the reader to the excellent reference [53].

2 QUBO reformulations of COPT problems

As mentioned earlier, some COPT problems evidently belong to the class of QUBO problems. A very representative problem with this property is the *Ising model* [see, e.g., 10]. That is, the Ising model is a problem of the form (1), where $x_i \in \{-1, 1\}$ represents whether the *spin* $i \in \{1, \dots, n\}$ is pointing up or down, the matrix Q represents the coupling between pairs of spins, and the vector c represents the impact of an external magnetic field on the spins. Since its inception, the Ising model has been used to address problems arising in different physical systems (e.g., magnetism, lattice gas, spin glasses), as well as in neuroscience and socio-economics. This also means that so-called Ising computers or machines [see, e.g., 32] can also be used to address the solution of QUBO reformulations of COPT problems [42].

The analog of the Ising model in COPT is the well-known max cut problem; which is naturally formulated as a QUBO problem. Namely, given a graph $G(V, E)$, the max cut problem aims to find a set of vertices $S \subseteq V$, such that the number of edges between vertices in S and the complement of S (i.e., S^c) is maximized. This problem can be formulated as the QUBO problem $\max_{x \in \{-1,1\}^n} \frac{1}{4}(e^\top Ae - x^\top Ax)$, where $x_i = 1$ (resp. $x_i = -1$) indicates that vertex i belongs to S (resp. belongs to S^c) for $i = 1, \dots, n$, A is the node to node incidence matrix of the graph $G(V, E)$, and e denotes the vector of all ones in the appropriate dimension.

Other COPT problems do not have a natural QUBO formulation. Typically, this is due to the problems having constraints (beyond the binary constraints). However, in many cases, these constrained COPT problems, such as the stable set problem or the knapsack problem, can be *reformulated* as a QUBO problem. Here, reformulation means that the resulting QUBO problem should be equivalent to the original problems both in terms of its objective and optimal solutions. This is formally stated below.

Definition 1 (QUBO reformulation). *Let C be an instance of a COPT defined over a finite set $X \subseteq \mathbb{R}^n$ and $Q(C)$ be a QUBO model (that depends on the parameters of C) defined over $\{0, 1\}^{n'}$ (or $\{-1, 1\}^{n'}$). Then, problem $Q(C)$ is said to be a QUBO reformulation of C if there exists a (projection) map $\pi : \mathbb{R}^{n'} \to \mathbb{R}^n$ such that*

1. *If C is feasible, then x^* is an optimal solution for C if and only if there exists an optimal solution y^* for $Q(C)$ such that $\pi(y^*) = x^*$.*

2. *If C is feasible, the optimal values of C and $Q(C)$ are the same.*

There is now a well established and general procedure to obtain QUBO reformulations of a wide range of COPT problems. Namely, when given a pure binary, and linearly equality constrained COPT, *penalization methods* can be used to embed the COPT problem's constraints into its objective to obtain such reformulation. For instance, given $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$, consider the problem

$$\min \left\{ \sum_{i=1}^{n} c_i x_i : Ax = b, x \in \{0, 1\}^n \right\} \tag{2}$$

Then, the linear equality constraints $Ax = b$ can be embedded in the objective by using the quadratic *penalty function* $C\|Ax - b\|^2$ where $C > 0$ is a *penalty* parameter. Specifically, if C is large enough, then the QUBO problem

$$\min_{x \in \{0,1\}^n} \sum_{i=1}^{n} c_i x_i + C\|Ax - b\|^2, \tag{3}$$

is a reformulation of the COPT (2). The value of C above can be more specifically characterized by stating that (3) is a QUBO reformulation of the COPT (2) if $C > c^\top e / \min\{\|Ax - b\|^2 : Ax \neq b, x \in \{0, 1\}^n\}$.

In practice however, it is desirable to take advantage of the particulars of the COPT in question to obtain more constructive results. In particular, depending on how the QUBO reformulation and constraints' penalization is done, different penalty parameters may be needed, additional binary variables may need to be added, and the binary length of the coefficients in the penalization functions may vary considerably. All of these features have an impact on the efficiency with which any current quantum computing device solves (or not) the proposed reformulations [54, 63]. Thus, there is now an interest not only in obtaining QUBO reformulations of COPT problems so their solution can be addressed with quantum computers devices, but also on obtaining *efficient* QUBO reformulations of these problems that are tailored to be more efficiently used in quantum computing devices [see, e.g., 8, 13, 16, 31, 54, 61, 62]. To illustrate this line of research work, this article considers some particular COPT problems. In this section, QUBO reformulations of these problems are obtained. Then in Section 3 more efficient QUBO reformulations of three of these problems are presented.

The first two examples are feasibility COPT problems (i.e., in which the objective function is the zero function). When the COPT of interest is a feasibility problem that can be formulated using linear equality constraints, the desired QUBO reformulation can be obtained using any positive penalty parameter to penalize the constraints' violations. This basically follows from the discussion following (3).

Example 1 (Number Partitioning Problem). *Given a finite set $S = \{s_1, \dots, s_n\}$ of positive integers, the* number partitioning *problem aims to answer the question of whether there is a disjoint partition of $S = S_1 \cup S_2$ such that the sums of the numbers in S_1 and S_2 are equal [see, e.g., 40]. This problem can be formulated as follows.*

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} 0x_i \\ \text{st.} \quad & \sum_{i=1}^{n} s_i x_i = 0 \\ & x \in \{-1,1\}, \end{aligned} \tag{4}$$

where any feasible solution $x \in \{-1,1\}^n$ of (4) corresponds to a desired partition by letting $S_1 = \{s_i : x_i = 1, i \in \{1,\dots,n\}\}$, $S_2 = \{s_i : x_i = 0, i \in \{1,\dots,n\}\}$.

Now, in order to solve the number partitioning problem by solving a QUBO problem, notice that for any $C > 0$,

$$0 = \min_{x \in \{-1,1\}^n} C \left(\sum_{i=1}^{n} s_i x_i \right)^2, \tag{5}$$

if and only if such partition exists. This follows by noticing that the right hand side of (5) is equal to zero if and only if $\sum_{i=1}^{n} s_i x_i = 0$. Moreover, any optimal solution x^* of (5) corresponds to a desired partition by letting $S_1 = \{s_i : x^*_i = 1, i \in \{1,\dots,n\}\}$, $S_2 = \{s_i : x^*_i = 0, i \in \{1,\dots,n\}\}$. For further details see [40, 43].

Next, we discuss another feasibility COPT problem in detail.

Example 2 (Graph Isomorphism Problem). *Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|V_1| = |V_2|$ and $|E_1| = |E_2|$, the* graph ismorphism *problem [see, e.g., 40] aims to answer if there is a bijective map $\varphi : V_1 \to V_2$ that satisfies the property $(i, j) \in E_1$ if and only if $(\varphi(i), \varphi(j)) \in E_2$. Such maps are known as edge-invariant or edge-preserving mappings. If there exists a bijective edge-invariant map $\varphi : V_1 \to V_2$ between G_1 and G_2, the graphs are said to be isomorphic, and φ is an isomorphism of the graphs. Determining if two graphs are isomorphic is a decision problem that belongs to NP, since we can verify in polynomial time if a given map φ is an isomorphism. But, it is not known if it belongs to P nor if it belongs to NP-complete. Mathematically, deciding if there exists an isomorphism between G_1 and G_2 is* *equivalent to solving the following optimization problem*

$$
\begin{aligned}
\min \quad & \sum_{(i,u)\in(V_1,V_2)} 0x_{iu} \\
\text{st.} \quad & \sum_{u\in V_2} x_{iu} = 1, && \text{for each } i \in V_1 && \text{(6a)} \\
& \sum_{i\in V_1} x_{iu} = 1, && \text{for each } u \in V_2 && \text{(6b)} \\
& x_{iu}x_{jv} = 0, && \text{for each } (i,j) \in E_1 \text{ and } (u,v) \notin E_2 && \text{(6c)} \\
& x_{iu}x_{jv} = 0, && \text{for each } (i,j) \notin E_1 \text{ and } (u,v) \in E_2 && \text{(6d)} \\
& x \in \{0,1\}^{|V_1|\times|V_2|}. && && \text{(6e)}
\end{aligned}
$$

Because every $x \in \{0,1\}^{|V_1|\times|V_2|}$ that is feasible for (6) corresponds to an isomorphism φ_x defined by $\varphi_x(i) := u$, where u is the unique vertex in V_2 such that $x_{iu} = 1$. The set of constraints (6a) and (6b) guarantee that the map φ_x is bijective, whereas the set of constraints (6c) and (6d) force that edges in E_1 are not sent to non-edges in E_2, and similarly, edges in E_2 are not sent to non-edges in E_1 (under φ_x^{-1}), which is the edge-preserving property.

Now, in order to decide if G_1 and G_2 are isomorphic by solving a QUBO problem, notice that for any $C_1 > 0, C_2 > 0$, and $C_3 > 0$,

$$
0 = \min_{x\in\{0,1\}^{V_1\times V_2}} C_1 \sum_{i\in V_1}\left(\sum_{u\in V_2} x_{iu} - 1\right)^2 + C_2 \sum_{u\in V_2}\left(\sum_{i\in V_1} x_{iu} - 1\right)^2 + C_3\left(\sum_{\substack{(i,j)\in E_1\\(u,v)\notin E_2}} x_{iu}x_{jv} + \sum_{\substack{(i,j)\notin E_1\\(u,v)\in E_2}} x_{iu}x_{jv}\right) \tag{7}
$$

if and only if G_1 and G_2 are isomorphic. This follows from the fact that every optimal solution x^* to (7) would correspond to an isomorphism φ_{x^*} defined in the same way as done so above for feasible solutions of (6). For further details see [8].

For other COPT feasibility problems that can be formulated using linear equality constraints, the desired QUBO reformulation can be obtained in a similar way (i.e., as in Example 1 and 2), using any positive penalty parameter (to penalize the constraints' violations). For example, consider the QUBO reformulations of the exact cover problem [40], and some planning problems [56], to name a few.

Now, when a COPT problem formulation uses an objective, together with linear and nonlinear equality constraints, the desired QUBO reformulation can be obtained using basic penalization techniques too [see, e.g., 20].

Example 3 (Stable Set Problem). *Let $G(V,E)$ be a simple graph. A subset $S \subseteq V$ is called an stable or independent set if there is no edge in E between any pair of vertices in S. The stable set or* independent *set problem asks to find a stable set S^* of maximum size, and $|S^*|$ is known as the stability number of G, which is denoted by $\alpha(G)$. Let $V = \{1,2,\dots,n\}$, then the stable set problem can be formulated as*

$$
\begin{aligned}
\alpha(G) = \max \quad & \sum_{i=1}^{n} x_i \\
\text{st.} \quad & x_ix_j = 0, \quad \text{for each } (i,j) \in E \\
& x \in \{0,1\}^n,
\end{aligned} \tag{8}
$$

where given an optimal solution x^* of (8), the associated maximum stable set S^* is defined by $S^* := \{i \in V : x_i^* = 1\}$. Now, to obtain a QUBO reformulation of the stable set problem (8), it is simple to set $C \geq n$ and consider the QUBO

$$
\max_{x\in\{0,1\}^n} \sum_{i=1}^{n} x_i - C \sum_{(i,j)\in E} x_ix_j. \tag{9}
$$

The fact that (9) *is a QUBO reformulation of* (8) *follows from the fact that* $1 \le \alpha(G)$, *and* $\max_{x \in \{0,1\}^n} e^\top x = n$. *Thus, if an optimal solution* x^* *of* (9) *violates any of the constraints* $x_i x_j = 0$, *for each* $(i,j) \in E$ *in* (8), *then the objective of the QUBO* (9) *will be less than or equal to zero, which contradicts the optimality of* x^*, *as* $x = (1,0,\dots,0)$ *is feasible for* (9) *and has an objective value equal to 1. This means that* x^* *must be feasible for* (8), *which together with the fact that* (9) *is a relaxation of* (8) *shows the reformulation equivalence between the two problems.*

As shown in Example 6 later, it is well-known that it is enough for the penalty parameter C *in* (9) *to be greater than 1 with a bit more of analysis [see, e.g., 1, 43, 54], and that in fact,* C *can be set to equal* 1 *as long as a greedy post-processing algorithm is performed on the optimal solution* x^* *of* (9) *(without the post-processing algorithm the QUBO provides the stable set number, but not necessarily a stable set) [54] .*

As illustrated in Example 3, a basic approach to finding QUBO reformulations of COPT problems might lead to QUBO formulations that are equivalent to the COPT problem only when the penalty parameters are greater than a large lower bound. For example, consider the QUBO reformulations for the maximum clique problem [40], the traveling salesman problem [40, 43], and the minimax matching problem [40]. Moreover, in some cases, the desired QUBO reformulation is only guaranteed to be obtained for an unknown large enough value of the penalty parameter(s). For example, consider the QUBO reformulations of the job shop scheduling problem [60], the de-conflicting optimal trajectories problem [59], the traveling salesman problem with time windows [47], and some of the problems discussed in [20].

When a COPT is formulated with inequality constraints, the basic technique to obtain corresponding QUBO reformulations is to add slack variables to turn inequality constraints into equality constraints and if necessary, represent the slack variables using binary variables. Then, a QUBO reformulation of the inequality constrained problem can be obtained using the technique illustrated in Example 3 for equality constrained COPT problems.

Example 4 (The Knapsack Problem). *Given* n *items, each of which has an associated profit* $c_i \in \mathbb{Z}_{++}$ *and weight* $w_i \in \mathbb{Z}_{++}$ *for* $i \in \{1,\dots,n\}$, *and a knapsack with weight capacity* $W \in \mathbb{Z}_{++}$, *where* $\mathbb{Z}_{++}$ *denotes the set of positive integers; the aim of in the* knapsack problem *is to pack as many items as possible to maximize the value of the packed knapsack. The knapsack problem can be formulated as*

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{st.} \quad & \sum_{i=1}^{n} w_i x_i \le W \\ & x \in \{0,1\}^n. \end{aligned} \tag{10}$$

After introducing a slack variable for the inequality constraint in (10) *and representing the slack using binary variables, problem* (10) *can be rewriting as follows*

$$\begin{aligned} z = \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{st.} \quad & \sum_{i=1}^{n} w_i x_i = \sum_{k=1}^{W} k y_k \\ & \sum_{k=1}^{W} y_k = 1 \\ & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{11}$$

Then, for any penalty parameters $C_1 > \sum_{i=1}^{n} c_i$ *and* $C_2 > \sum_{i=1}^{n} c_i$, *the knapsack problem* (11) *can be reformulated as the following QUBO problem*

$$\begin{aligned} \max \quad & f(x,y) := \sum_{i=1}^{n} c_i x_i - C_1 \left(\sum_{i=1}^{n} w_i x_i - \sum_{k=1}^{W} k y_k \right)^2 - C_2 \left(1 - \sum_{k=1}^{W} y_k \right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{12}$$

To see this, note that $0 \le z^*$*, and* $\max_{x \in \{0,1\}^n} e^\top c < \min\{C_1, C_2\}$*. Thus, if an optimal solution* $(x^*, y^*) \in \{0,1\}^{n+W}$ *of* (12) *violates any of the equality constraints in* (11)*, then the objective of the QUBO* (12) *will be less than zero, which contradicts the optimality of* (x^*, y^*)*, as* $(x, y) = (0, \dots, 0)$ *is feasible for* (12) *and has an objective equal to zero. This means that* (x^*, y^*) *must be feasible for* (11)*, which together with the fact that* (12) *is a relaxation of* (11) *shows that the reformulation equivalence between the two problems [see 40, for further details].*

The technique of introducing slack binary variables used in Example 4 has been used to obtain QUBO reformulations of COPT problems with inequality constraints. For example, consider the maximum clique QUBO reformulation provided in [40], and the inequality constrained COPT problems considered in [63].

3 Efficient QUBO formulations

As shown in the Section 2, QUBO reformulations of COPT problems can be obtained by making an appropriate choice of the COPT formulation, using penalization techniques, and performing analysis on the associated QUBO. However, the particular way in which this process is carried out, affects the value of the penalty parameters, the objective coefficients, as well as the number of binary variables used to obtain the desired QUBO reformulation. As mentioned earlier, in turn, these factors can substantially affect the ability of quantum computing devices to efficiently solve the QUBO reformulation of a COPT problem. This efficiency is key towards the goal of showing that quantum computing devices have the capacity to outperform classical computers, a goal referred as *quantum supremacy* [see, e.g., 51]. For this reason, there is an interest specially in the area of quantum computing to, beyond obtaining QUBO reformulations of COPT problems, obtain *efficient* QUBO reformulations of these problems; that is, QUBO reformulations that are tailored to be more efficiently used in quantum computing devices [see, e.g., 8, 13, 16, 31, 54, 61, 62]. Typically, this is done by further exploiting the structure and nature of the COPT problem of interest, in order to obtain QUBO reformulations using tighter penalty constants, tighter objective coefficients, and the least number of binary variables. This type of work can be illustrated by revisiting some of the examples considered in Section 2.

First, reconsider the graph isomorphism problem considered in Example 2.

Example 5 (Graph Isomorphism Problem (revisited)). *Notice that if* $\varphi : V_1 \to V_2$ *is a bijective map with the edge-preserving property, it must hold true that* $\deg(i) = \deg(\varphi(i))$*, where* $\deg(i) = |\{j \in V_1 : (i,j) \in E_1\}|$ *is the number of neighbors of the vertex* $i \in V_1$*. Therefore, any feasible solution* x *for* (6) *satisfies* $x_{iu} = 0$ *for every* $(i,u) \in V_1 \times V_2$ *such that* $\deg(i) \neq \deg(u)$*. Thus, it suffices to only consider pairs* $(i,u) \in V_1 \times V_2$ *such that* $\deg(i) = \deg(u)$ *[31]. Let* $S = \{(i,u) \in V_1 \times V_2 : \deg(i) = \deg(u)\}$*, then, problem* (6) *can be written as:*

$$
\begin{aligned}
\min \quad & \sum_{(i,u) \in S} 0 x_{iu} \\
\text{st.} \quad & \sum_{\substack{u \in V_2 \\ (i,u) \in S}} x_{iu} = 1, && \text{for each } i \in V_1 \quad \text{(13a)} \\
& \sum_{\substack{i \in V_1 \\ (i,u) \in S}} x_{iu} = 1, && \text{for each } u \in V_2 \quad \text{(13b)} \\
& \sum_{ij \in E_1} \left(\sum_{(i,u) \in S} x_{iu} \left(\sum_{(j,v) \in S} (1 - e_{jv}) x_{jv} \right) \right) = 0 && \text{(13c)} \\
& x \in \{0,1\}^S, && \text{(13d)}
\end{aligned}
$$

where, for each $(j,v) \in V_2 \times V_2$*,*

$$
e_{jv} = \begin{cases} 1, & \text{if } (j,v) \in E_2 \\ 0, & \text{otherwise} \end{cases}.
$$

Notice that this is an improved formulation because the two sets of constraints (6c) *and* (6d) *from* (6) *were collapsed into the single constraint* (13c)*. Similarly, the number of binary variables of problem* (13) *is* $|S|$*, whereas in* (6) *the number is* $|V_1 \times V_2|$*. For sparse graphs this represents a considerable decrease on the number of binary variables used in the problem. However, there is a pre-processing cost to construct* S *and the search method in* S *to build constraints* (13a) *and* (13b) *which needs to be taken into account (this cost is a lower degree polynomial on* $|V_1|$ *and* $|E_1|$*).*

Now, taking advantage of (13)*, notice that given positive constants* $C_1 > 0, C_2 > 0$ *and* $C_3 > 0$*, the graph* G_1 *is isomorphic to* G_2 *if and only if*

$$0 = \min_{x \in \{0,1\}^S} C_1 \sum_{i \in V_1} \left(\sum_{\substack{u \in V_2 \\ (i,u) \in S}} x_{iu} - 1 \right)^2 + C_2 \sum_{u \in V_2} \left(\sum_{\substack{i \in V_1 \\ (i,u) \in S}} x_{iu} - 1 \right)^2 + C_3 E(x) \tag{14}$$

where $E(x) := \sum_{ij \in E_1} \left(\sum_{(i,u) \in S} x_{iu} \left(\sum_{(j,v) \in S} (1 - e_{jv}) x_{jv} \right) \right)$*. Analogously to* (7)*, every optimal solution* x^* *for* (14) *corresponds to an isomorphism* φ_{x^*} *defined by* $\varphi_{x^*}(i) = u$*, where* u *is the unique vertex in* V_1 *such that* $x^*_{iu} = 1$*. Thus,* (14) *is a QUBO reformulation for* (13)*. The QUBO* (14) *is more efficient than the QUBO* (7)*, as the former uses a lower number of binary variables. This is relevant given the small number of* qubits *[see, e.g., 51] available in current and near-future quantum computing devices that are used to represent binary variables. For further details see [31].*

Additional reduction in the number of binary variables in (14) *might be attained by looking in more detail at the the pre-processing used to obtain the set* S*. If graph* G_2 *has many edges, for instance if* $|E_2| > \frac{n(n-1)}{4}$*, then, there will be many non-zero terms in the left hand side of equation* (13c)*. Therefore, to reduce the size of the set* S*, one could instead solve problem* (13) *for the complement graphs* $\bar{G}_1$ *and* $\bar{G}_2$*, and decide if they are isomorphic. This can be done because* G_1 *is isomorphic to* G_2 *if and only if* $\bar{G}_1$ *is isomorphic to* $\bar{G}_2$*, and sometimes it is easier to solve the problem for the complement graphs.*

Next, reconsider the stable set problem considered in Example 3.

Example 6 (Stable Set Problem (revisited))**.** *In Example 3 it was shown that* (9) *is a reformulation of the stable set problem for any penalty parameter* $C > n$ *(where* n *is the number of vertices in the graph of interest). A bit more of analysis shows that in fact, it is enough for the penalty parameter to satisfy* $C > 1$ *[see, e.g., 1, 43, 54]. Loosely speaking, this follows from the fact that if an optimal solution* x^* *of* (9) *is not feasible for* (8)*, then there are edges in* E *between some of the vertices in* $S^* = \{i \in \{1, \dots, n\} : x^*_i = 1\}$*. The penalty for having one of these edges is* $C > 1$*, while the benefit is adding exactly* 1 *to the term* $e^\top x^*$ *in the objective of the QUBO. Thus, the overall benefit of having edges between the vertices in* S^* *is negative (i.e., at most* $1 - C < 0$ *for every edge). As a result, even with* $C > 1$*, problem* (9) *is a QUBO reformulation of the stable set problem* (8) *[see, e.g., 1, 54, for a formal proof]. This same idea basically shows that when* $C = 1$*, problem* (9) *will return the stable number of the graph. However,* $S^* = \{i \in \{1, \dots, n\} : x^*_i = 1\}$ *might not be a stable set, as in this case, the benefit (trade-off) of having an infeasible solution might be zero. As shown in [54], this can be fixed by greedily removing vertices from the set* S^* *that have edges in* E *with other vertices in* S^**.*

The fact that with $C = 1$*, problem* (9) *is a reformulation of the stable set problem is very useful for quantum computing devices as this means that all the coefficients in the objective of the QUBO problem belong to the set* $\{-1, 1\}$*. This is key for the performance of quantum computing devices as their limited size and noisy behavior means that to obtain good practical results, it is better for these coefficients to be integer and in a small range [see, e.g., 63].*

Now, reconsider the knapsack problem discussed in Example 4

Example 7 (The Knapsack Problem (revisited))**.** *Notice that one can go from the inequality constrained formulation of the knapsack problem* (10) *to an equality constrained formulation of the knapsack in a slightly different way than the one done in* (11) *that was exploited to obtain the QUBO reformulation* (12) *in [40]. Namely, one can alternatively formulate* (10) *as* $\max\{c^\top x : w^\top x = e^\top y, (x, y) \in \{0,1\}^{n+W}\}$*. Given this, an* *analogous argument to the one used in Example 4 shows that for any $C > e^\top c$ the QUBO problem*

$$\max_{(x,y)\in\{0,1\}^{n+W}} c^\top x - C\left(w^\top x - \sum_{k=1}^{W} y_k\right)^2 \tag{15}$$

is a reformulation of the knapsack problem. Further analysis of the QUBO (15) (that does not apply to the QUBO (12)) shows that the penalty parameter in (15) can be tightened to be $C = \max\{c_i : i \in \{1,\dots,n\}\}$ [see 55, for details]. The QUBO (15) is more efficient than QUBO (12), as the tightening of the penalty parameter C and the elimination of the constants k, for $k = 1,\dots,W$ in (12), means that the range of the coefficients of the QUBO (15) is tighter than the range of the coefficients in the QUBO (12). More specifically, note that in (15) the coefficients of y_k, $k = 1,\dots,W$ (inside the parenthesis) are all equal to one; wheres in (12), the coefficient of y_k (inside the first parenthesis) is k for $k = 1,\dots,W$. Thus, as W increases, the binary input length of the coefficients of the penalty function in (12) goes to infinity. As mentioned earlier, this is relevant since the binary input length that current quantum computing devices can handle is very limited [see, e.g., 63].

The QUBO (15) can be made more efficient in terms of number of slack binary (y) variables added. This can be done by representing the term $\sum_{k=1}^{W} y_k$ in QUBO (15) using its binary representation. That is, QUBO (15) is equivalent to the following QUBO problem

$$\max_{x\in\{0,1\}^n, y\in\{0,1\}^{M+1}} \sum_{i=1}^{n} c_i x_i - C\left(\sum_{i=1}^{n} w_i x_i - \sum_{k=0}^{M-1} 2^k y_k - \left(W + 1 - 2^M\right) y_M\right)^2, \tag{16}$$

where $M = \lfloor \log W \rfloor$ [see, e.g., 12, 55]. This formulation for the knapsack problem is well-known and has been used in the literature [see, e.g., 12], although in this work, the penalty $C \geq e^\top c$ is used instead of the tighter one discussed above $C = \max\{c_i : i = 1,\dots,n\}$.

Examples 5, 6, and 7 illustrate how to obtain more efficient QUBO reformulations of COPT problems, so that current and near-future quantum computing devices can be better used to address the solution of COPT problems. Similar work has been done for a wealth of COPT problems. For example, consider the work on the max-k-cut problem [13], the maximum k-colorable subgraph problem [54], different tree related COPT problems [16], the graph isomorphism problem [31], the knapsack problem [55], and COPT problems with a fourth degree objective [61], to name a few. Using different numerical experiments in different quantum computing devices, these articles show the positive effect that the improved QUBO reformulations have on the capacity of quantum computing devices to address the solution of COPT problems.

4 Ongoing Work

It has been shown that having efficient QUBO formulations of different COPT problems is a key step if algorithms like QAOA or quantum annealing will be used to solve COPT problems in quantum computing devices [see, e.g., 13, 54, 62]. The approach highlighted through the article to obtain such efficient QUBO reformulations is problem centric; that is, every COPT problem is addressed in a way that is particular to the problem. In contrast with this approach, several ideas [see, e.g., 18, 25] have been proposed with the aim to create a general framework to obtain practical QUBO reformulations that do not exploit directly any structure of the original COPT, but instead try to compute, in general, tractable penalty parameters for QUBO reformulations of large-scale COPT instances. Mostly, these ideas apply only to equality constrained COPT problems, and do not scale well when there are inequality constraints, since in order to use them, many binary variables (to model slack variables) and probably large penalty parameters need to be introduced. In [62], the authors try to deal with this problem by partitioning and restricting the feasible domain of the slack variables that need to be added, which allows the use of parallel computing techniques to simultaneously explore the different domains and solve the corresponding QUBOs. This is a topic that still needs to be carefully addressed and could unveil a rich mathematical theory and connections between different areas.

For liner inequality constrained COPT problems, an avenue that is being explored is not only to penalize the quadratic violation of the inequality constraint (after adding adequate slack variables), but also to penalize the linear violation of the inequality constraint. This approach can be described as constructing the QUBO reformulation using an *augmented Lagrangian* [see, e.g., 57] approach. Theoretically, this approach has been considered in [55], and numerically explored in [38, 67].

Now, the idea of eﬀective QUBO reformulation explored in this article might not necessarily be the same for every quantum or classical method to solve COPT problems. For instance, another approach to get eﬀective QUBO reformulations comes from considering a relaxation of the cannonical semidefinite relaxation of the Ising model, and then using Gibbs sampling and suitable *matrix exponents updates* to obtain approximate solutions of the original problem up to a certain additive factor [5]. This factor depends on the binary input length of the QUBO matrix, its sparsity, and the desired accuracy of the solution. A similar approach [52, Ch. 7] tries to find a suitable *quadratic convex relaxation* of the QUBO model whose optimal value is close to that of the QUBO by solving the dual of the cannonical semidefinite relaxation of the proposed QUBO reformulation.

See also

- Augmented Lagrangian methods
- Combinatorial optimization
- Combinatorial test problems and problem generators
- Computational complexity theory
- Graph coloring
- Heuristics for maximum clique and independent set
- Integer programming
- Integer programming, branch and bound methods
- Max cut problem
- NP-complete problems and proof methodology
- Quadratic integer programming
- Quantum aproximate optimization algorithm
- Simulated annealing

References

[1] Abello, J., Butenko, S., Pardalos, P. M., and Resende, M. G. (2001). Finding independent sets in a graph using continuous multivariable polynomial formulations. *Journal of Global Optimization*, 21(2):111–137.

[2] Atamtürk, A. and Narayanan, V. (2007). Cuts for conic mixed-integer programming. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 16–29. Springer.

[3] Bliek, C., Bonami, P., and Lodi, A. (2014). Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report. In *Proceedings of the twenty-sixth RAMP symposium*, pages 16–17.

[4] Bradley, G. H., Hammer, P. L., and Wolsey, L. (1974). Coeﬃcient reduction for inequalities in 0–1 variables. *Mathematical Programming*, 7(1):263–282.

[5] Brandao, F. G. L., Kueng, R., and França, D. S. (2022). Faster quantum and classical SDP approximations for quadratic binary optimization. *Quantum*, 6:625.

[6] Brush, S. G. (1967). History of the Lenz-Ising model. *Reviews of modern physics*, 39(4):883.

[7] Burer, S. and Letchford, A. N. (2012). Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106.

[8] Calude, C. S., Dinneen, M. J., and Hua, R. (2017). QUBO formulations for the graph isomorphism problem and related problems. *Theoretical Computer Science*, 701:54–69.

[9] Choi, V. (2008). Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing*, 7(5):193–209.

[10] Cipra, B. A. (2000). The Ising model is NP-complete. *SIAM News*, 33(6):1–3.

[11] Conforti, M., Cornuéjols, G., Zambelli, G., et al. (2014). *Integer programming*, volume 271. Springer.

[12] de la Grand'rive, P. D. and Hullo, J.-F. (2019). Knapsack problem variants of QAOA for battery revenue optimisation. *arXiv preprint arXiv:1908.02210*.

[13] Fakhimi, R., Validi, H., Hicks, I. V., Terlaky, T., and Zuluaga, L. F. (2021). Quantum-inspired formulations for the max k-cut problem. Technical report, ISE Technical Report 21T-007 Lehigh University. Available at https://engineering.lehigh.edu/sites/engineering.lehigh.edu/files/_DEPARTMENTS/ise/pdf/tech-papers/21/21T_007b.pdf.

[14] Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

[15] Farhi, E. and Harrow, A. W. (2016). Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*.

[16] Fowler, A. (2017). *Improved QUBO formulations for D-Wave quantum computing*. PhD thesis, University of Auckland.

[17] Fuchs, F. G., Kolden, H. Ø., Aase, N. H., and Sartor, G. (2021). E□cient encoding of the weighted max k-cut on a quantum computer using QAOA. *SN Computer Science*, 2(2):1–14.

[18] García, M. D., Ayodele, M., and Moraglio, A. (2022). Exact and sequential penalty weights in quadratic unconstrained binary optimisation with a digital annealer. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO*, volume 22.

[19] Ginsburgh, V. and Van Peeterssen, A. (1969). Un algorithme de programmation quadratique en variables binaires. *Revue française d'informatique et de recherche opérationnelle. Série verte*, 3(V2):57–73.

[20] Glover, F., Kochenberger, G., Hennig, R., and Du, Y. (2022). Quantum bridge analytics I: A tutorial on formulating and using QUBO models. *Annals of Operations Research*, pages 1–43.

[21] Glover, F., Kochenberger, G. A., and Alidaee, B. (1998). Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44(3):336–345.

[22] Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145.

[23] Guerreschi, G. G. and Matsuura, A. Y. (2019). QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific reports*, 9(1):1–7.

[24] Gulati, V., Gupta, S., and Mittal, A. (1984). Unconstrained quadratic bivalent programming problem. *European Journal of Operational Research*, 15(1):121–125.

[25] Gusmeroli, N. and Wiegele, A. (2022). Expedis: An exact penalty method over discrete sets. *Discrete Optimization*, 44:100622.

[26] Hammer, P. L., Johnson, E. L., and Peled, U. N. (1975). Facet of regular 0–1 polytopes. *Mathematical Programming*, 8(1):179–206.

[27] Hammer, P. L. and Rubin, A. A. (1970). Some remarks on quadratic programming with 0-1 variables. *Revue française d'informatique et de recherche opérationnelle. Série verte*, 4(V3):67–79.

[28] Hammer, P. L. and Rudeanu, S. (1968). *Boolean methods in operations research and related areas*, volume 7. Springer Berlin, Heidelberg.

[29] Hammer, P. L. and Rudeanu, S. (1969). Pseudo-boolean programming. *Operations Research*, 17(2):233–261.

[30] Helmberg, C. and Rendl, F. (1998). Solving quadratic (0, 1)-problems by semidefinite programs and cutting planes. *Mathematical programming*, 82(3):291–315.

[31] Hua, R. and Dinneen, M. J. (2020). Improved QUBO formulation of the graph isomorphism problem. *SN Computer Science*, 1(1):19.

[32] Inagaki, T., Haribara, Y., Igarashi, K., Sonobe, T., Tamate, S., Honjo, T., Marandi, A., McMahon, P. L., Umeki, T., Enbutsu, K., et al. (2016). A coherent Ising machine for 2000-node optimization problems. *Science*, 354(6312):603–606.

[33] Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2011). Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198.

[34] Karimi, S. and Ronagh, P. (2017). A subgradient approach for constrained binary optimization via quantum adiabatic evolution. *Quantum Information Processing*, 16(8):1–21.

[35] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

[36] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., and Wang, Y. (2014). The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization*, 28(1):58–81.

[37] Kunzi, H. P. and Oettli, W. (1963). Integer quadratic programming. *Recent advances in mathematical programming*, pages 303–308.

[38] Kuroda, T., Kuwahara, T., Yonaga, K., Osaki, T., Miyama, M. J., and Ohzeki, M. (2021). Quantum annealing for ict system design automation. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 156–165. IEEE.

[39] Li, D., Sun, X., et al. (2006). *Nonlinear integer programming*, volume 84. Springer.

[40] Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in physics*, page 5.

[41] Mahdavi Pajouh, F., Balasundaram, B., and Prokopyev, O. A. (2013). On characterization of maximal independent sets via quadratic optimization. *Journal of Heuristics*, 19(4):629–644.

[42] Mohseni, N., McMahon, P. L., and Byrnes, T. (2022). Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379.

[43] Nannicini, G. (2019). Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. *Physical Review E*, 99(1):013304.

[44] Palmer, S., Karagiannis, K., Florence, A., Rodriguez, A., Orus, R., Naik, H., and Mugel, S. (2022). Financial index tracking via quantum computing with cardinality constraints. *arXiv preprint arXiv:2208.11380*.

[45] Pan, S., Tan, T., and Jiang, Y. (2008). A global continuation algorithm for solving binary quadratic programming problems. *Computational Optimization and Applications*, 41(3):349–362.

[46] Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

[47] Papalitsas, C., Andronikos, T., Giannakis, K., Theocharopoulou, G., and Fanarioti, S. (2019). A QUBO model for the traveling salesman problem with time windows. *Algorithms*, 12(11):224.

[48] Pardalos, P. M., Prokopyev, O. A., and Busygin, S. (2006). Continuous approaches for solving discrete optimization problems. In *Handbook on modelling for discrete optimization*, pages 39–60. Springer.

[49] Pardalos, P. M. and Rodgers, G. P. (1990). Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45(2):131–144.

[50] Pokharel, B. and Lidar, D. A. (2022). Demonstration of algorithmic quantum speedup. *arXiv preprint arXiv:2207.07647*.

[51] Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, 2:79.

[52] Punnen, A. P. (2022a). The quadratic unconstrained binary optimization problem. Technical report, Springer.

[53] Punnen, A. P. (2022b). *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications*. Springer Nature.

[54] Quintero, R., Bernal, D., Terlaky, T., and Zuluaga, L. F. (2022). Characterization of QUBO reformulations for the maximum k-colorable subgraph problem. *Quantum Information Processing*, 21(3):1–36.

[55] Quintero, R. A. and Zuluaga, L. F. (2021). Characterizing and benchmarking QUBO reformulations of the knapsack problem. Technical Report ISE Technical Report 21T-028, Lehigh University. Available at https://engineering.lehigh.edu/sites/engineering.lehigh.edu/files/_DEPARTMENTS/ise/pdf/tech-papers/21/21T_028.pdf.

[56] Rie□el, E. G., Venturelli, D., O'Gorman, B., Do, M. B., Prystay, E. M., and Smelyanskiy, V. N. (2015). A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14(1):1–36.

[57] Rockafellar, R. T. (1974). Augmented lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12(2):268–285.

[58] Sefair, J. A., Méndez, C. Y., Babat, O., Medaglia, A. L., and Zuluaga, L. F. (2017). Linear solution schemes for mean-semivariance project portfolio selection problems: An application in the oil and gas industry. *Omega*, 68:39–48.

[59] Stollenwerk, T., O?Gorman, B., Venturelli, D., Mandrà, S., Rodionova, O., Ng, H., Sridhar, B., Rie□el, E. G., and Biswas, R. (2019). Quantum annealing applied to de-conflicting optimal trajectories for air tra□c management. *IEEE transactions on intelligent transportation systems*, 21(1):285–297.

[60] Venturelli, D., Marchand, D., and Rojo, G. (2016). Job shop scheduling solver based on quantum annealing. In *Proc. of ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS)*, pages 25–34.

[61] Verma, A. and Lewis, M. (2020a). Optimal quadratic reformulations of fourth degree pseudo-boolean functions. *Optimization Letters*, 14(6):1557–1569.

[62] Verma, A. and Lewis, M. (2020b). Penalty and partitioning techniques to improve performance of QUBO solvers. *Discrete Optimization*, page 100594.

[63] Vyskočil, T., Pakin, S., and Djidjev, H. N. (2019). Embedding inequality constraints for quantum annealing optimization. In *International workshop on quantum technology and optimization problems*, pages 11–22. Springer.

[64] Wang, F. and Xu, Z. (2013). Metaheuristics for robust graph coloring. *Journal of Heuristics*, 19(4):529–548.

[65] Wang, Z., Hadfield, S., Jiang, Z., and Rieffel, E. G. (2018). Quantum approximate optimization algorithm for maxcut: A fermionic view. *Physical Review A*, 97(2):022304.

[66] Williams, H. P. (1985). Model building in linear and integer programming. *Computational mathematical programming*, pages 25–53.

[67] Yonaga, K., Miyama, M. J., and Ohzeki, M. (2020). Solving inequality-constrained binary optimization problems on quantum annealer. *arXiv preprint arXiv:2012.06119*.