Industrial and
Systems Engineering

Characterizing and Benchmarking QUBO Reformulations of the Knapsack Problem

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

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

ISE Technical Report 21T-028

LEHIGH UNIVERSITY

Characterizing and Benchmarking QUBO Reformulations of the Knapsack Problem

Rodolfo A. Quintero*

Luis F. Zuluaga†

Abstract

Constrained combinatorial optimization problems can be reformulated as a quadratic binary unconstrained optimization (QUBO) problems using reformulation and constraint penalization techniques. This allows the use of quantum computers, both gate and annealing based, to obtain approximate solutions of the problems. However, depending on how the reformulation and constraint penalization is done, different penalization constants need to be used; additional binary variables 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 computer solves the aforementioned reformulations. Most of the literature regarding QUBO reformulations for constrained optimization problems is centered around equality constrained problems. Here, we focus on the "simplest" inequality constrained problem in this class: the knapsack problem (KP). Specifically, we derive different QUBO formulations for the KP, characterize the range of their associated penalty constants, as well as computationally benchmark them through experiments using the quantum aproximate optimization algorithm (QAOA) on a gate based quantum computer. As a byproduct, we correct some erroneous results regarding QUBO reformulations for the KP reported in the literature.

1 Introduction

Quantum computing (QC) has the potential of transforming our capabilities to solve difficult optimization problems for which no traditional numerical or theoretical efficient solution algorithms are known to exist [14]. Within this framework, the category of combinatorial optimization problems (COPT) are of special interest; that is, optimization problems whose feasible set $S \subseteq \{0,1\}^n$ (or equivalently, $\{-1,1\}^n$) for some positive integer n [16]. It is well known that many COPT problems are NP-hard to solve [see, e.g., 10]. Here, we will consider one of the "simplest" NP-hard COPT problems: the knapsack problem (KP) [18], since it maximizes a linear objective function subject to a single linear inequality constraint and to binary variable constraints.

The KP together with many famous COPT problems can be reformulated as a *quadratic unconstrained binary optimization* (QUBO) problem, which allows the use of both quantum annealing devices [see, e.g., 11, 13] and quantum gate based devices to solve them [see, 4, 23].

For some COPT feasibility problems (i.e., without an objective) that can be formulated using linear equality constraints, the desired QUBO reformulation can be obtained using any positive

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

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

penalty parameter (to penalize the constraints' violations). For example, consider the QUBO reformulations of the number partitioning problem [11, 15], the graph isomorphism problem [2], the exact cover problem [11], and some planning problems [20], to name a few. However, when the COPT problem formulation requires (or uses) nonlinear constraints and/or an objective function, the desired QUBO reformulation is only guaranteed to be obtained for values of the penalty parameter(s) that are larger than a known, and potentially large, lower bound. For example, consider the QUBO reformulations for the maximum clique problem [11], the traveling salesman problem [11, 15], and the minimax matching problem [11]. Worst, 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 [25], the de-conflicting optimal trajectories problem [24], the traveling salesman problem with time windows [17], and some of the problems discussed in [6]. Additionally, when the COPT problem formulation requires (or uses) linear inequality constraints, a potentially large number of auxiliary (i.e., slack) binary variables need to be introduced to obtain the desired QUBO reformulation. For example, consider the maximum clique QUBO reformulation provided in [11], and the COPT problems considered in [28].

The fact that large (or unknowingly large) penalty parameters, and additional binary variables might be needed to obtain the desired QUBO reformulation can hinder the ability of quantum computers to more efficiently solve COPT problems [see, e.g., 5, 22, 28]. As the results in [7] highlight, this efficiency is key towards the goal of using *noisy intermediate scale quantum* (NISQ) devices to solve COPT problems more efficiently than with classical computers. Not surprisingly, recent articles look beyond obtaining QUBO reformulations of COPT problems such as the graph isomorphism problem as well as tree and cycle elimination problems, to look for *improved* QUBO reformulations of these problems for NISQ devices [see, e.g., 2, 5, 9, 26, 27]. That is, QUBO reformulations that are tailored to be more efficiently used in NISQ devices.

Along this lines, we consider the problem of characterizing the lowest penalty constants that can be used when different QUBO formulations as considered to obtain a QUBO reformulation of the KP. Regarding QUBO reformulations of the KP, in [11, Section 5.2] a QUBO formulation of the KP is given that was used in [19] together with adiabatic computing to solve the KP. In this work, we go deeper into this characterization, since the range of the penalty constants proposed in [19] and [11] does not produce, in general, exact reformulations of the KP. We give penalty constants that correspond with reformulations of the KP, and also provide another characterization of the KP that uses the same amount of binary variables as the one in [11], but with a much smaller input length of the penalty constants and the coefficients in the penalty function.

The rest of the paper is organized as follows. In Section 2, we review the KP and the QUBO reformulation considered in [11] and characterize the penalty constants associated with this QUBO reformulation, as well as other alternative QUBO reformulations. In Section 3, we consider avenues to improve the QUBO reformulations considered in Section 2, by considering unconstrained binary optimization reformulations of the KP without the use of slack variables. To benchmark the QUBO reformulations presented in Section 2, in Section 4, we use them to solve instances of the KP, using the *quantum approximate optimization algorithm* (QAOA) on an IBM quantum simulator.

2 QUBO reformulations for knapsack problem

The KP can be stated as follows. Given n items, each of which has an associated profit $c_i \in \mathbb{Z}_{++}$ and weight $w_i \in \mathbb{Z}_{++}$ for $i \in \{1, \ldots, n\} := [n]$, and a knapsack with weight capacity $W \in \mathbb{Z}_{++}$, where $\mathbb{Z}_{++}$ denotes the set of positive integers; the aim of the KP is to pack as many items as possible to maximize the value of the packed knapsack. The KP can be formulated as:

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

Without loss of generality, we can assume that $w_i \leq W$ for every $i \in [n]$. Also, we can assume that $W < \sum_{i=1}^{n} w_i$, since otherwise the solution to (1) is trivial. Note that with this assumption the binary input length of W cannot grow arbitrary large. For ease of presentation, let us put all the conditions on the data of the KP in a blanket assumption that applies to all the results presented in the article.

Assumption 1. *Throughout the article, it is assumed that the data of the KP instances satisfies:* $c \in \mathbb{Z}^n_{++}$, $w \in \mathbb{Z}^n_{++}$, $W \in \mathbb{Z}_{++}$, $w_i \leq W$, *for* $i \in [n]$, *and* $W < \sum_{i=1}^{n} w_i$.

Theoretically, it is well known that the decision knapsack problem is NP-complete [see, e.g., 18, 29]. More speficially, the KP complexity is known to be *weakly* NP-hard, since it can be solved in *pseudo-polynomial time* $O(nW)$ using *dynamic programming* [see, e.g., 18]. In practice, large instances of the KP can be solved by exact methods in reasonable time. See [12], for a survey of some techniques to deal with such instances.

Next, we derive the first QUBO reformulation of the KP considered in [see, e.g., 11, Sec. 5.2]. Note that (1) is equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{st.} \quad & \sum_{i=1}^{n} w_i x_i + s = W \\ & x \in \{0,1\}^n, s \in \mathbb{Z}_+. \end{aligned} \tag{2}$$

If we rewrite (2) as follows

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

one can get a QUBO reformulation of (1) by defining a penalty function $g : \{0,1\}^{n+W} \to \mathbb{Z}$ as

$$g(x, y) = \left(\sum_{i=1}^{n} w_i x_i - \sum_{k=1}^{W} k y_k \right)^2 + \left(1 - \sum_{k=1}^{W} y_k \right)^2,$$

and, showing that for a large enough constant C, problem (1) is equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i - C g(x,y) \\ \text{s.t} \quad & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{3}$$

In [11, Sec. 5.2], it is stated that for any constant C greater than $\max\{c_i : i \in [n]\}$, reformulation (3) is equivalent to (1); however, this is not always the case, as illustrated in the next examples:

Example 1. *Consider the following instance of the KP*

$$\begin{aligned} \max \quad & 5x_1 + 3x_2 + 4x_3 + 5x_4 \\ \text{st.} \quad & 2x_1 + x_2 + 2x_3 + 2x_4 \leq 4 \\ & x \in \{0,1\}^4 \end{aligned} \tag{4}$$

whose optimal solution is attained at $x^* = (1,0,0,1)$ *with optimal value* $z^* := 10$*. Now, consider the QUBO formulation*

$$\begin{aligned} \max \quad & 5x_1 + 3x_2 + 4x_3 + 5x_4 - 6\left(2x_1 + x_2 + 2x_3 + 2x_4 - \sum_{k=1}^{4} k y_k\right)^2 - 6\left(1 - \sum_{k=1}^{4} y_k\right)^2 := f(x,y) \\ \text{st.} \quad & x \in \{0,1\}^4, y \in \{0,1\}^4, \end{aligned} \tag{5}$$

and note that

$$f\left((1,1,1,1),(0,0,1,1)\right) = 17 - 6(0) - 6(1) = 11 > 10 = z^*.$$

Then, the QUBO formulation (5) *is not a reformulation of* (4)*, even when the chosen penalty constant* $C = 6$ *is greater than* $\max\{5,3,4,5\} = 5$ *(note that the example still works if one chooses any penalty constant* $5 < C < 7$*).*

Now, let us make a similar analysis for a more general instance of the KP.

Example 2. *For each* $m \in \mathbb{Z}_+$*, let* $\mathcal{A}_m = \{c_1, \ldots, c_{2m+3}\} \subset \mathbb{Z}_+$ *be a set such that* $c_i \leq N$ *for some fixed* $N \in \mathbb{Z}_+$*, and every* $i \in [2m+3]$*. In other words,* $\mathcal{A}_m$ *is a bounded set whose upper bound is independent of* m*. Let*

$$M_m := \max\left\{\sum_{i \in I} c_i : I \subset [2m+3],\ |I| = m+2\right\},$$

and J_m *be a* $(m+2)$*-subset of* $[2m+3]$ *such that* $M_m = \sum_{i \in J_m} c_i$*. By definition,*

$$\begin{aligned} M_m = \max \quad & \sum_{i=1}^{2m+3} c_i x_i \\ \text{st.} \quad & \sum_{i=1}^{2m+3} c_i x_i \leq M_m \\ & x \in \{0,1\}^{2m+3}. \end{aligned} \tag{6}$$

Following (3)*, consider the QUBO formulation*

$$\begin{aligned} z := \max \quad & f(x,y) := \sum_{i=1}^{2m+3} c_i x_i - C\left(\sum_{i=1}^{2m+3} c_i x_i - \sum_{k=1}^{M_m} k y_k\right)^2 - C\left(1 - \sum_{k=1}^{M_m} y_k\right)^2 \\ \text{st.} \quad & x \in \{0,1\}^{2m+3}, y \in \{0,1\}^{M_m}, \end{aligned} \tag{7}$$

with $C = \sum_{i\in H} c_i$*, where* H *is a* m*-subset of* $[2m+3]\setminus J_m$*. By construction of* M_m*, there are positive integers* $\alpha \le M_m$ *and* $\beta \le M_m$ *such that* $\alpha \ne \beta$*, and* $\alpha y_\alpha + \beta y_\beta = \sum_{i=1}^{2m+3} c_i$ *(for instance, we can take* $\alpha = M_m$*, and* $\beta = \sum_{i=1}^{2m+3} c_i - M_m < M_m$*). Therefore, if we let* $\hat{x} = (1,1,\dots,1) \in \mathbb{Z}^{2m+3}$*, and define* $\hat{y} \in \mathbb{Z}^{M_m}$ *as*

$$\hat{y}_k = \begin{cases} 1, & \text{if } h = \alpha \text{ or } h = \beta \\ 0, & \text{otherwise} \end{cases} \text{for all } k \in [M_m].$$

Then, problem (7) *satisfies*

$$z \ge f(\hat{x},\hat{y}) = \sum_{i=1}^{2m+3} c_i - C = \sum_{i=1}^{2m+3} c_i - \sum_{i\in H} c_i = M_m + c_l > M_m$$

where $l = [2m+3] \setminus (J_m \cup H)$*. Thus, we conclude that* (7) *is not a reformulation of problem* (6)*.*

Note that since $\mathcal{A}_m$ *is bounded by* N *for every* $m \in \mathbb{Z}_+$*, and the sum* $C = \sum_{i\in H} c_i \to \infty$ *as* m *goes to infinity, we have just proved that using* N *as a penalty constant in* (7) *is not enough (and neither was* C*), which contradicts the reformulation of the KP shown in [11, Sec. 5.2]. That is, taking* $C = \max\{c_i : i \in [n]\}$ *in* (3) *does not guarantee that* (3) *is a QUBO reformulation of* (1)*.*

Remark 1. *In example 2, we have actually shown something stronger, namely, there is a family of KPs* $\{\mathcal{K}(m) : m \in \mathbb{Z}_+\}$ *such that their associated penalty constant in problem* (3) *goes to infinity as* m *increases to infinity.*

Following up on Examples 1 and 2, we next characterize the range of the penalty constants that ensures that (3) is a QUBO reformulation of (1). In what follows, we use e to denote the vector of all ones in the appropriate dimension.

Theorem 1. *For any constants* $C_1 \ge \max\{c_i : i \in [n]\}$ *and* $C_2 \ge \sum_{i=1}^n c_i$ *, the knapsack problem* (1) *can be reformulated as the following QUBO problem:*

$$\begin{aligned} \max \quad & f(x,y) := \sum_{i=1}^{n} c_i x_i - C_1\left(\sum_{i=1}^{n} w_i x_i - \sum_{k=1}^{W} k y_k\right)^2 - C_2\left(1 - \sum_{k=1}^{W} y_k\right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{8}$$

Proof. Let $u \in \{0,1\}^n$ be an optimal solution for (1). Note that since u is feasible for (1), there exists $v \in \{0,1\}^W$ such that $\sum_{k=1}^W v_k = 1$ and $w^T u - \sum_{k=1}^W k v_k = 0$, which implies that $f(u,v) = c^T u$. Let $(x^*,y^*) \in \{0,1\}^{n+W}$ be an optimal solution of (8). Then, $f(x^*,y^*) \ge f(u,v)$.

Conversely, we want to prove that $f(x^*,y^*) \le f(u,v)$ by showing that x^* must correspond to a feasible solution for (1), or can be transformed into a feasible solution for (1).

If x^* is feasible for (1), then there exists $y' \in \{0,1\}^W$ such that $w^\intercal x^* = \sum_{k=1}^W k y'_k$ and $e^\intercal y' = 1$. Thus,

$$f(x^*,y^*) = f(x^*,y') = c^\intercal x^* \le c^\intercal u = f(u,v).$$

The first equality above follows from the fact that the $\max\{f(x^*, y) : y \in \{0,1\}^W\}$ is attained at any $y \in \{0,1\}^W$ satisfying $w^\intercal x^* = \sum_{k=1}^W ky_k$, $e^\intercal y = 1$. Further, the inequality above follows since x^* is feasible for (1).

If x^* is not feasible for (1); that is, $w^T x^* - W > 0$, then, it must hold true $y_W^* = 1$ and $y_k^* = 0$ for every $k \in [W-1]$, since at $y = (0, \ldots, 1)$, the difference $w^T x^* - \sum_{k=1}^W ky_k$ is minimized for every $y \in \{0,1\}^W$ such that $e^\intercal y = 1$.

Let $j \in \text{supp}(x^*)$ and define $\hat{x} \in \{0,1\}^n$ by letting

$$\hat{x}_i = \begin{cases} x_j^*, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \quad \text{for all } i \in [n].$$

We consider two cases:

(i) $w^T \hat{x} - W > 0$: Then, by defining $\hat{y} := y^* = (0, \ldots, 1)$, we get that $\sum_{k=1}^W ky_k^* = \sum_{k=1}^W k\hat{y}_k = W$, and:

$$\begin{aligned} f(x^*, y^*) &= c^T x^* - C_1 \left(\sum_{i=1}^n w_i x_i^* - \sum_{k=1}^W k y_k^* \right)^2 - C_2 \left(1 - \sum_{k=1}^W y_k^* \right)^2 \\ &= c^T \hat{x} + c_j - C_1 \left(\sum_{i=1}^n w_i \hat{x}_i + w_j - W \right)^2 \\ &= c^T \hat{x} + c_j - C_1 \left(\sum_{i=1}^n w_i \hat{x}_i - W \right)^2 - 2C_1 w_j \left(\sum_{i=1}^n w_i \hat{x}_i - W \right) - C_1 w_j^2 \\ &< c^T \hat{x} - C_1 \left(\sum_{i=1}^n w_i \hat{x}_i - W \right)^2 = f(\hat{x}, \hat{y}); \end{aligned}$$

which contradicts the optimality of (x^*, y^*) for (8), so this case cannot happen.

(ii) $w^T \hat{x} - W \leq 0$: Then, let $\hat{y} \in \{0,1\}^W$ be defined by

$$\hat{y}_k = \begin{cases} 1, & \text{if } k = w^T \hat{x} \\ 0, & \text{otherwise} \end{cases} \quad \text{for all } k \in [W].$$

Thus,

$$\begin{aligned} f(x^*, y^*) &= c^T x^* - C_1 \left(\sum_{i=1}^n w_i x_i^* - \sum_{k=1}^W k y_k^* \right)^2 - C_2 \left(1 - \sum_{k=1}^W y_k^* \right)^2 \\ &= c^T \hat{x} + c_j - C_1 \left(\sum_{i=1}^n w_i x_i^* - W \right)^2 \\ &\begin{cases} < c^T \hat{x} = c^T \hat{x} - C_1 \left(\sum_{i=1}^n w_i \hat{x}_i - \sum_{i=1}^W k\hat{y}_k \right)^2 = f(\hat{x}, \hat{y}) & \text{if } C_1 < \max\{c_i : i \in [n]\} \\ \leq c^T \hat{x} = c^T \hat{x} - C_1 \left(\sum_{i=1}^n w_i \hat{x}_i - \sum_{i=1}^W k\hat{y}_k \right)^2 = f(\hat{x}, \hat{y}) & \text{if } C_1 = \max\{c_i : i \in [n]\}. \end{cases} \end{aligned}$$

Thus, if $C_1 > \max\{c_i : i \in [n]\}$, this case cannot happen as we reach a contradiction on the optimality of (x^*, y^*) for (8). Therefore, if $C_1 > \max\{c_i : i \in [n]\}$, then x^* is feasible for (1). If $C_1 = \max\{c_i : i \in [n]\}$, either the same conclusion above is reached or $f(x^*, y^*) = f(\hat{x}, \hat{y})$; that is, $(\hat{x}, \hat{y})$ is optimal for (8) too, and since $\hat{x}$ is feasible for (1), $f(x^*, y^*) = c^T \hat{x} \leq c^\intercal u = f(u, v)$.

Therefore, we have concluded that if (x^*, y^*) is optimal for (8), and u is optimal for (1), then $f(x^*, y^*) = c^\intercal u$ and x^* is feasible for (1), or from x^* one can construct a feasible solution $\hat{x}$ for (1), with and associate $\hat{y} \in \{0,1\}^W$, satisfying $f(x^*, y^*) = f(\hat{x}, \hat{y}) = c^\intercal u$. □

Another QUBO reformulation for the KP (Corollary 1) can be obtained if we write the constraint $w^T x + s = W$ in (2) as $w^T x = \sum_{i=1}^{W} y_k$, for some $y \in \{0,1\}^W$. Note that in this reformulation of the knapsack capacity constraint, the coefficients of y_k are all equal to one, whereas in Theorem 1, each y_k has a coefficient equal to k for every $k \in [n]$. One consequence of this is that as W increases, the binary input length of the coefficients of the penalty function in (8) goes to infinity. This is relevant since the binary input length that current quantum computers can handle is very limited [see, e.g., 28]. Also, as we will see next (Corollary 2), the reformulation of the knapsack capacity constraint allows the use of a lower penalty constant $C = \max\{c_i : i \in [n]\}$, and to reduce the number of logical binary variables used in the QUBO reformulation using a logarithmic representation of the slack variable in the knapsack capacity constraint [see, e.g., 3].

Corollary 1. *For any constant $C \geq \max\{c_i : i \in [n]\}$, the knapsack problem (1) can be reformulated as the following QUBO problem:*

$$\begin{aligned} \max \quad & c^T x - C\left(w^T x - \sum_{k=1}^{W} y_k\right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{9}$$

Proof. The result follows from taking $C_1 = 0$ in Theorem 2. □

In practice, we can decrease the number of binary variables used in the formulation of Corollary 1 by using the binary representation of $\sum_{k=1}^{W} y_k$ [see, e.g., 11, Sec. 2.4] and [3, Sec. 3.2]) . On the other hand, note that contrary to the discussion in [11, Sec. 5.2], one can not take advantage of this type of binary representation in the QUBO reformulation (8). As a consequence of corollary 1, in corollary 2 we see that the penalty constant used in [3] can be significantly tighten from $c^T e$ to $\max\{c_i : i \in [n]\}$.

Corollary 2. *For any constant $C \geq \max\{c_i : i \in [n]\}$, the knapsack problem (1) can be reformulated as the following QUBO problem:*

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i - C\left(\sum_{i=1}^{n} w_i x_i - \sum_{k=0}^{M-1} 2^k y_i - \left(W + 1 - 2^M\right) y_M\right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, y \in \{0,1\}^{M+1}, \end{aligned}$$

where $M = \lfloor \log W \rfloor$.

The QUBO reformulation of the KP in Corollary 2 is well-known and has been used in the literature [see, e.g., 3]. However, Corollary 2 shows that the penalty associated with the QUBO problem can be reduced from the commonly used $C = \sum_{i=1}^{n} c_i$ to $C = \max\{c_i : i \in [n]\}$. Further, it is worth mentioning that the QUBO reformulation in Corollary 1 allows to obtain the reduction on binary variables in Corollary 2 using a binary representation of the slack variable in the knapsack capacity constraint. This type of reduction cannot be obtained using the QUBO reformulation in Theorem 1, as incorrectly pointed out in [11, Sec. 5.2].

The reformulations of the KP in Corollaries 1 and 2 can be cast as special cases of a more general reformulation, an *augmented Lagrangian* QUBO formulation of the KP, where not only the quadratic violation of the knapsack capacity constraint is penalized, but also its linear violation. In classical optimization, this type of constraint relaxation is referred as and augmented Lagrangian of the original problem [see, e.g., 21]. Augmented Lagrangian algorithms have been studied in continuous constrained optimization in detail [see, e.g., 1]. This type of augmented Lagrangian QUBO formulation has been considered in [30] as a means to solve the KP using a quantum annealer. Below, the aim is to characterize the range of penalties that can be used for the augmented Lagrangian QUBO formulation to be indeed a QUBO reformulation of the KP.

Theorem 2. *For any constants $C_1 \in \mathbb{R}$, and $C_2 \geq |C_1| + \max\{c_i : i \in [n]\}$, the knapsack problem (1) can be reformulated as the following QUBO problem:*

$$\begin{aligned} \max \quad & f(x,y) := \sum_{i=1}^{n} c_i x_i + C_1 \left(w^T x - \sum_{k=1}^{W} y_k \right) - C_2 \left(w^T x - \sum_{k=1}^{W} y_k \right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, y \in \{0,1\}^W. \end{aligned} \tag{10}$$

Proof. The proof is similar to that of Theorem 1. For each $x \in \{0,1\}^n, y \in \{0,1\}^W$, we denote by $h(x,y) := w^T x - e^\intercal y$. Note that after completing the square, we can rewrite $f(x,y)$ as

$$f(x,y) = \sum_{i=1}^{n} c_i x_i - C_2 \left(h(x,y) - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2}. \tag{11}$$

Let $u \in \{0,1\}^n$ be an optimal solution for (1). Note that since u is feasible for (1), there exists $v \in \{0,1\}^W$ such that $e^\intercal v = w^T u$, which implies that $f(u,v) = c^T u$. Let $(x^*, y^*) \in \{0,1\}^{n+W}$ be an optimal solution of (10). Then, $f(x^*, y^*) \geq f(u,v)$.

Conversely, we want to prove that $f(x^*, y^*) \leq f(u,v)$ by showing that x^* must correspond to a feasible solution for (1), or can be transformed into a feasible solution for (1).

If x^* is feasible for (1) then there exists $y' \in \{0,1\}^W$ such that $w^\intercal x^* = e^\intercal y'$. Thus,

$$f(x^*, y^*) = f(x^*, y') = c^\intercal x^* \leq c^\intercal u = f(u,v).$$

The first equality above follows from the fact that the $\max\{f(x^*, y) : y \in \{0,1\}^W\}$ is attained at any $y \in \{0,1\}^W$ satisfying $w^\intercal x^* = e^\intercal y$. Further, the inequality above follows since x^* is feasible for (1).

If x^* is not feasible for (1); that is, $w^T x^* > W$, then note that $\left(w^T x^* - e^T y - \frac{C_1}{2C_2} \right)^2$ viewed as a function of $y \in \{0,1\}^W$ is minimized when $y = e$. Thus, we must have that $y^* = e$.

Let $j \in \text{supp}(x^*)$ and define $\hat{x} \in \{0,1\}^n$ by letting

$$\hat{x}_i = \begin{cases} x_i^*, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \quad \text{for all } i \in [n].$$

We consider two cases:

(i) $w^T\hat{x} - W > 0$: Then, by defining $\hat{y} := y^* = e$, we have that

$$\begin{aligned}
f(x^*, y^*) &= c^T x^* - C_2 \left(w^T x^* - \sum_{k=1}^{W} y_k^* - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} \\
&= c^T \hat{x} + c_j - C_2 \left(w^T \hat{x} + w_j - \sum_{k=1}^{W} \hat{y}_k - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} \\
&= c^T \hat{x} - C_2 \left(w^T \hat{x} - \sum_{k=1}^{W} \hat{y}_k - \frac{C_1}{2C_2} \right)^2 + c_j - 2C_2 w_j \left(w^T \hat{x} - \sum_{k=1}^{W} \hat{y}_k - \frac{C_1}{2C_2} \right) - C_2 w_j^2 + \frac{C_1^2}{4C_2} \\
&< c^T \hat{x} - C_2 \left(w^T \hat{x} - \sum_{k=1}^{W} \hat{y}_k - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} = f(\hat{x}, \hat{y})
\end{aligned}$$

where the last inequality follows from the fact that $c_j \leq C_2 w_j$ and $w^T\hat{x} - e^\intercal\hat{y} - \frac{C_1}{2C_2} > 0$. This implies that $f(x^*, y^*) < f(\hat{x}, \hat{y})$, which contradicts the optimality of (x^*, y^*) for (10). Thus, this case can not happen.

(ii) $w^T\hat{x} - W \leq 0$: Then, let $\hat{y} \in \{0,1\}^W$ be defined by

$$\hat{y}_k = \begin{cases} 1, & \text{if } \sum_{l=1}^{k} l \leq w^T\hat{x} \\ 0, & \text{otherwise} \end{cases} \qquad \text{for all } k \in [W].$$

Thus, using the fact that $w^T x^* - e^T y^* \geq 1$,

$$\begin{aligned}
f(x^*, y^*) &= c^T x^* - C_2 \left(w^T x^* - \sum_{k=1}^{W} y_k^* - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} \\
&\leq c^T \hat{x} + c_j - C_2 \left(1 - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} \\
&= c^T \hat{x} + (c_j - C_2 + C_1) - C_2 \left(\frac{C_1^2}{4C_2^2} \right) + \frac{C_1^2}{4C_2} \\
&\begin{cases} < c^T\hat{x} - C_2 \left(w^T\hat{x} - e^\intercal\hat{y} - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} = f(\hat{x}, \hat{y}) & \text{if } C_2 > |C_1| + \max\{c_i : i \in [n]\} \\ \leq c^T\hat{x} - C_2 \left(w^T\hat{x} - e^\intercal\hat{y} - \frac{C_1}{2C_2} \right)^2 + \frac{C_1^2}{4C_2} = f(\hat{x}, \hat{y}) & \text{if } C_2 = |C_1| + \max\{c_i : i \in [n]\} \end{cases}
\end{aligned}$$

Thus, if $C_2 > |C_1| + \max\{c_i : i \in [n]\}$, this case cannot happen as we reach a contradiction on the optimality of (x^*, y^*) for (10). Therefore, if $C_2 > |C_1| + \max\{c_i : i \in [n]\}$, then x^* is feasible for (1). If $C_2 = |C_1| + \max\{c_i : i \in [n]\}$, either the same conclusion above is reached or $f(x^*, y^*) = f(\hat{x}, \hat{y})$; that is, $(\hat{x}, \hat{y})$ is optimal for (10) too, and since $\hat{x}$ is feasible for (1), $f(x^*, y^*) = c^T\hat{x} \leq c^\intercal u = f(u, v)$.

Therefore, we have concluded that if (x^*, y^*) is optimal for (10), and u is optimal for (1), then $f(x^*, y^*) = c^\intercal u$ and either x^* is feasible for (1), or from x^* one can construct a feasible solution $\hat{x}$ for (1), with and associate $\hat{y} \in \{0,1\}^W$, satisfying $f(x^*, y^*) = f(\hat{x}, \hat{y}) = c^\intercal u$. □

Remark 2. *When $C_2 = C_1 + \max\{c_i : i \in [n]\}$, the proof of Theorem 2 tells us what conditions an optimal solution (x^*, y^*) for (10) must satisfy if x^* is not feasible for (1). Namely, it must hold true that $w^T x^* - e^T y^* = w^T x^* - W = 1$ and $c_i = C_2 - C_1 = \max\{c_i : i \in [n]\}$, for every $i \in \text{supp}(x^*)$, otherwise we can always construct a better optimal solution $(\hat{x}, \hat{y})$ for (10).*

After the characterization of the penalties associated with the QUBO reformulations of the KP discussed in this section, the question is what are the pros and cons of these QUBO reformulations when used to solve the KP using a quantum computer. That analysis is done in Section 4.

3 About removing the slack variable

One common feature of the QUBO reformulations of the KP discussed in the last section is that in order to handle the knapsack capacity inequality constraint in (1), a slack variable is introduced that is then represented as a sum of binary variables in different ways. This actually a common way to handle inequality constraints to obtain QUBO reformulations of inequality constrained combinatorial optimization problems [see, e.g., 5, 11, 24]. However, this approach adds a substantial number of binary variables to the QUBO reformulation, with respect to the number of variables in the original problem, that make the solution of these QUBO formulations, in noisy intermediate-scale quantum (NISQ) devices, very challenging.

One way in which one can avoid the use of the slack variable for the knapsack capacity inequality constraint in (1), is to consider a piecewise linear penalization, instead of a quadratic penalization. The cost however is this way, rather than a QUBO reformulation of the KP, one obtains piece-wise linear unconstrained binary optimization (PLQUBO) of the knapsack problem. Although it is still an open matter how could one efficiently use quantum computers to solve such reformulation, next we characterize the penalty constant associated to such reformulation. In what follows, for any $x \in \mathbb{R}^n$, $\text{supp}(x) = \{i \in [n] : x_i \neq 0\}$.

Theorem 3. *For any constant $C \geq \max\{c_i : i \in [n]\}$, the knapsack problem (1) can be reformulated as the following PLUBO problem:*

$$\begin{aligned} \max \quad & f(x) := c^T x + C \min\{0, W - w^T x\} \\ \text{st.} \quad & x \in \{0,1\}^n \end{aligned} \tag{12}$$

Proof. The proof is similar to that of Theorem 1. Let $u \in \{0,1\}^n$ be an optimal solution for (1). Let $x^* \in \{0,1\}^n$ be and optimal solution for (12). Hence, since every feasible solution for (1) is feasible for (12), it follows that $f(x^*) \geq f(u)$.

Conversely, we want to prove that $f(x^*) \leq f(u)$ by showing that x^* must correspond to a feasible solution for (1), or can be transformed into a feasible solution for (1).

If x^* is feasible for (1), it clearly follows that $f(x^*) = c^\intercal x^* \leq c^\intercal u = f(u)$. Now assume that x^* is infeasible for (1). Let $j \in \text{supp}(x^*)$, and define $\hat{x} \in \{0,1\}^n$ by letting

$$\hat{x}_i := \begin{cases} x_i^*, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \text{ for all } i \in [n]. \tag{13}$$

We consider two cases:

(i) $w^T \hat{x} - W > 0$: Then, we have that

$$\begin{aligned} f(x^*) = c^T x^* + C(W - w^T x^*) &= c^T \hat{x} + c_i + C(W - w^T \hat{x} - w_i) \\ &= f(\hat{x}) + c_i - C w_i \begin{cases} < f(\hat{x}) & \text{if } C > \max\{c_i : i \in [n]\} \\ \leq f(\hat{x}) & \text{if } C = \max\{c_i : i \in [n]\}. \end{cases} \end{aligned}$$

(ii) $w^T\hat{x} - W \leq 0$: Then, we have that

$$f(x^*) = c^T\hat{x} + c_i + C(W - w^T x^*) \begin{cases} < c^T\hat{x} = f(\hat{x}) & \text{if } C > \max\{c_i : i \in [n]\} \\ \leq c^T\hat{x} = f(\hat{x}) & \text{if } C = \max\{c_i : i \in [n]\}. \end{cases}$$

Thus, if $C > \max\{c_i : i \in [n]\}$, neither case can happen as we reach a contradiction on the optimality of x^* for (12). Therefore, if $C > \max\{c_i : i \in [n]\}$, then x^* is feasible for (1). If $C = \max\{c_i : i \in [n]\}$, in both cases, either the same conclusion above is reached or $f(x^*) = f(\hat{x})$; that is, $\hat{x}$ is optimal for (12) too. Thus, in case (ii), since $\hat{x}$ is feasible for (1), $f(x^*) = c^T\hat{x} \leq c^\intercal u = f(u)$. In case (i), after repeating as many times as necessary the process (13) after setting $x^* = \hat{x}$ (this algorithm is formally described in Algorithm 1), we arrive eventually (abusing notation) to an optimal solution $\hat{x}$ for (12) that is feasible for (1), and the statement above for case (ii) follows.

Therefore, we have concluded that if x^* is optimal for (12), and u is optimal for (1), then $f(x^*) = c^\intercal u$ and x^* is feasible for (1), or from x^* one can construct a feasible solution $\hat{x}$ for (1), satisfying $f(x^*) = f(\hat{x}) = c^\intercal u$. □

Remark 3. *Similarly, as we did in Remark 2, note that when $C = \max\{c_i : i \in [n]\}$, the proof of Theorem 3 gives us information about what properties any optimal solution x^* for (12) must satisfy if it is not feasible for (1). Namely, if $w^T x^* > W$, it must hold for every $i \in \operatorname{supp}(x)$ that either $c_i = Cw_i$ (case (i) in the proof), equivalently, $c_i = C$ and $w_i = 1$; or, $c_i = C(w^T x^* - W)$, which is equivalent to $c_i = C$ and $w^T x^* - W = 1$ (case (ii) in the proof).*

We note that a PLUBO similar to (12) is considered in [30], in which instead of the penalty function $\min\{0, x\}$, the penalization function is the indicator function $\mathbb{I}_{\mathbb{R}_+}(x)$, defined by $\mathbb{I}_{\mathbb{R}_+}(x) = 1$ if $x \geq 0$, and $\mathbb{I}_{\mathbb{R}_+}(x) = 0$ if $x < 0$.

Although it is likely not possible to obtain a general QUBO reformulation of the KP in which a slack variable associated with the inequality knapsack capacity constraints, it might be possible to obtain such QUBO reformulation for certain classes of KPs. To motivate this possibility, consider the following result, which shows that any optimal solution of the KP has a particular QUBO reformulation, without the use of the slack variable of the knapsack capacity constraint.

Algorithm 1 Knapsack feasibility from (12) to (1)

```
1: Input: (w_1, ..., w_n), W, x ∈ {0,1}^n optimal for (12), supp(x).
2: A → supp(x)
3: while w^T x > W do
4:     Choose i ∈ A
5:     x_i → 0
6:     A → A \ {i}
7: end while
8: Output: x optimal for (12) with x feasible for (1).
```

Theorem 4. *Let $x^* \in \{0,1\}^n$. There exist constants $C_1 \in \mathbb{R}$ and $C_2 \in \mathbb{R}$ such that x^* is an optimal solution of the QUBO problem:*

$$\begin{aligned} \max \quad & f(x) := \sum_{i=1}^{n} c_i x_i + C_1 \left(\sum_{i=1}^{n} w_i x_i - W \right) - C_2 \left(\sum_{i=1}^{n} w_i x_i - W \right)^2 \\ \text{st.} \quad & x \in \{0,1\}^n, \end{aligned} \tag{14}$$

if and only if x^* is an optimal solution of the KP (1).

Proof. Let $x^* \in \{0,1\}$ be an optimal solution for (1). Also, let C_2 be a constant satisfying the property that for every $z \in \{0,1\}^n$ and $y \in \{0,1\}^n$ such that $w^Tz < w^Tx^*$ and $w^Ty > w^Tx^*$,

$$(w^Ty - w^Tz)C_2 \geq \frac{c^Tx^* - c^Tz}{w^Tz - w^Tx^*} - \frac{c^Tx^* - c^Ty}{w^Ty - w^Tx^*} \tag{15}$$

and define the following constants

$$a := \max\left\{\frac{c^Tx^* - c^Tz}{w^Tz - w^Tx^*} + C_2(w^Tx^* + w^Tz - 2W) : z \in \{0,1\}^n,\ w^Tz < w^Tx^*\right\}$$

and

$$b := \min\left\{\frac{c^Tx^* - c^Ty}{w^Ty - w^Tx^*} + C_2(w^Tx^* + w^Ty - 2W) : y \in \{0,1\}^n,\ w^Ty > w^Tx^*\right\}$$

By construction of C_2, we have that $a \leq b$. If we take $C_1 \in [a,b]$ and C_2 satisfying (15), we claim that $x^* \in \text{argmax}\{f(x) : x \in \{0,1\}^n\}$. Let $x \in \{0,1\}^n$, we consider three cases

(i) If $w^Tx = w^Tx^*$, we have that

$$\begin{aligned} f(x^*) &= c^Tx^* + C_1(w^Tx^* - W) - C_2(w^Tx^* - W)^2 = c^Tx^* + C_1(w^Tx - W) - C_2(w^Tx - W)^2 \\ &\geq c^Tx + C_1(w^Tx - W) - C_2(w^Tx - W)^2 = f(x) \end{aligned}$$

(ii) If $w^Tx < w^Tx^*$, we have that

$$\begin{aligned} & f(x^*) - f(x) = c^Tx^* + C_1(w^Tx^* - W) - C_2(w^Tx^* - W)^2 - \left(c^Tx + C_1(w^Tx - W) - C_2(w^Tx - W)^2\right) \\ &= c^Tx^* - c^Tx + C_1(w^Tx^* - w^Tx) - C_2(w^Tx^* - w^Tx)(w^Tx^* + w^Tx - 2W) \\ &= c^Tx^* - c^Tx + (w^Tx^* - w^Tx)\left(C_1 - C_2(w^Tx^* + w^Tx - 2W)\right) \\ &\geq c^Tx^* - c^Tx + (w^Tx^* - w^Tx)\left(\frac{c^Tx^* - c^Tx}{w^Tx - w^Tx^*} + C_2(w^Tx^* + w^Tx - 2W) - C_2(w^Tx^* + w^Tx - 2W)\right) \\ &= c^Tx^* - c^Tx - (c^Tx^* - c^Tx) = 0 \end{aligned}$$

where the inequality follows from the fact that $w^Tx^* - w^Tx > 0$ and $C_1 \geq a \geq \frac{c^Tx^* - c^Tx}{w^Tx - w^Tx^*} + C_2(w^Tx^* + w^Tx - 2W)$.

(iii) If $w^Tx > w^Tx^*$, we have that

$$\begin{aligned} & f(x^*) - f(x) = c^Tx^* + C_1(w^Tx^* - W) - C_2(w^Tx^* - W)^2 - \left(c^Tx + C_1(w^Tx - W) - C_2(w^Tx - W)^2\right) \\ &= c^Tx^* - c^Tx + C_1(w^Tx^* - w^Tx) - C_2(w^Tx^* - w^Tx)(w^Tx^* + w^Tx - 2W) \\ &= c^Tx^* - c^Tx + (w^Tx^* - w^Tx)\left(C_1 - C_2(w^Tx^*hf + w^Tx - 2W)\right) \\ &\geq c^Tx^* - c^Tx + (w^Tx^* - w^Tx)\left(\frac{c^Tx^* - c^Tx}{w^Tx - w^Tx^*} + C_2(w^Tx^* + w^Tx - 2W) - C_2(w^Tx^* + w^Tx - 2W)\right) \\ &= c^Tx^* - c^Tx - (c^Tx^* - c^Tx) = 0 \end{aligned}$$

where the inequality follows from the fact that $w^Tx^* - w^Tx < 0$ and $C_1 \leq b \leq \frac{c^Tx^* - c^Tx}{w^Tx - w^Tx^*} + C_2(w^Tx^* + w^Tx - 2W)$.

This shows that x^* is an optimal solution of (14). Finally, note that if we take C_2 so that $a < b$ and $C_1 \in (a, b)$, then, $f(x) = f(x^*)$ if and only if $w^T x = w^T x^*$ and $c^T x = c^T x^*$, if and only if x is an optimal solution of (1). □

Note that Theorem 4 would provide the desired QUBO reformulation of the KP, without the use of a slack variable for the knapsack capacity constraint, if one could find appropriate bounds for the constants a and b that are independent of an optimal solution of the KP. Although it is likely impossible to do this for all instances of the KP, it might be possible to obtain such bounds for certain classes of KP problems.

4 Benchmarking

Below, in Table 1, we can see a comparison of the number of binary variables used by each of the QUBO reformulations, and the range of the coefficients in their objective functions, where we assume without loss of generality that $w_1 \leq \cdots \leq w_n$.

Reformulation	Logical Qubits	Range of Coefficients (approx.)
Theorem 1	$n + W$	$[c_1 - C_1 W^2 - C_2, 2C_1 w_n W]$
Corollary 1	$n + W$	$[-c_n(w_n^2 - 1), 2w_n]$
Corollary 2	$n + \log_2 W$	$\left[-c_n(w_n^2 - 1), 2w_n \max\{2^{M-1}, W + 1 - 2^M\}\right]$
Theorem 2	$n + W$	$\left[\min\{c_1 + C_1 w_1 - C_2 w_n^2, -C_1 - C_2\}, 2w_n C_2\right]$

Table 1: Benchmarking of QUBO reformulation in terms of binary variables and coefficient ranges.

Two clear conclusions can be drawn from Table 1. First, it is clear that the range of the objective coefficients in the QUBO reformulation given in Corollary 1 is smaller than the range of the objective coefficients in the QUBO reformulation given in Theorem 1. It is thus expected that the QUBO reformulation given in Corollary 1 will produce better results than the QUBO reformulation given in Theorem 1 when one tries to solve the KP via these QUBO reformulations in a NISQ device. Second, it is clear that the range of the objective coefficients in the QUBO reformulation given in Corollary 1 is smaller than the range of the objective coefficients in the augmented Lagrangian QUBO reformulation given in Theorem 2. It is thus expected that the QUBO reformulation given in Corollary 1 will produce better results than the QUBO reformulation given in Theorem 2 when one tries to solve the KP via these QUBO reformulations in a NISQ device. Thus, from the point of view of range of coefficients, it doesn't seem to be useful to consider an augmented Lagragian approach to construction a QUBO reformulation for the KP.

Next we benchmark the different QUBO reformulations presented in the article using some preliminary numerical results. As stated above, when running the numerical experiments, we assume without loss of generality that the items are sorted by weight, namely, $w_1 \leq w_2 \leq \ldots w_n$, so, any optimal solution (x^*, s^*) of (2) satisfies $s^* \leq W - w_1$, since $w_1 + s^* \leq w^T x^* + s^* \leq W$. Moreover, since $W \leq w_1 + \ldots + w_n$, we must have $s^* \leq w_n$, otherwise, we can construct a new feasible solution $(\hat{x}, \hat{s})$ from (x^*, s^*) that attains a better objective value than (x^*, s^*) by adding an item $j \in [n] \setminus \text{supp}(x)$ to the knapsack. As a result, it can be concluded that $s^* \leq \min\{W - w_1, w_n\}$. Thus, instead of using W slack variables in Theorems 1, and Corollaries 1 and 2, we only use $\min\{W - w_1, w_n\}$. A similar analysis for bounding s can be found in [8].

We consider the following basic instance of the Knapsack problem:

$$\begin{aligned} \max \quad & 6x_1 + 2x_2 + 4x_3 + 5x_4 \\ \text{subject to} \quad & 4x_1 + 1x_2 + 2x_3 + 2x_4 \leq 4 \\ & x \in \{0,1\}^4. \end{aligned} \tag{16}$$

After running ten times the QAOA algorithm on this instance, we get the following results:

	Objective value	Optimal Sol (x)	Mean time to solution (sec)
Thm. 1	9.0	$[0,0,1,1]$	32.9354
Cor. 1	9.0	$[0,0,1,1]$	38.1678
Cor. 2	9.0	$[0,0,1,1]$	27.3654

If we change the coefficients of the objective function in (16) by larger ones, for instance, if we try to solve

$$\begin{aligned} \max \quad & 16x_1 + 10x_2 + 12x_3 + 13x_4 \\ \text{subject to} \quad & 4x_1 + 1x_2 + 2x_3 + 2x_4 \leq 4 \\ & x \in \{0,1\}^4, \end{aligned} \tag{17}$$

we obtain the following results after five runs:

	Objective value	Optimal Sol	Mean time to solution (sec)
Thm. 1	25.0	$[0,0,1,1,0,0,0,1]$	161.3289
Cor. 1	25.0	$[0,0,1,1,1,1,1,1]$	152.8471
Cor. 2	25	$[0,0,1,1,1,1,1]$	144.4432

Now, if we change slightly the weigths of the items in the Knapsack problem, for instance to

$$\begin{aligned} \max \quad & 6x_1 + 2x_2 + 4x_3 + 5x_4 \\ \text{subject to} \quad & 4x_1 + 1x_2 + 3x_3 + 2x_4 \leq 8 \\ & x \in \{0,1\}^4, \end{aligned} \tag{18}$$

we obtain the following results

	Objective value	Optimal Sol	Time to solution (sec)
Thm. 1	$--$	$--$	$--$
Cor. 1	13.0	$[1,1,0,1]$	1306.6719
Cor. 2	13.0	$[1,1,0,1]$	113.13737

To draw conclusions on the numerical behavior of the different QUBO reformulations when NISQ devices are used to solve the KP, substantial numerical experiments need to and will be done in the near future.

Acknowledgements

This project has been carried out thanks to funding by the Defense Advanced Research Projects Agency (DARPA), ONISQ grant W911NF2010022, titled The Quantum Computing Revolution and Optimization: Challenges and Opportunities.

References

[1] Birgin, E. G. and Martínez, J. M. (2014). *Practical augmented Lagrangian methods for constrained optimization*. SIAM.

[2] 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.

[3] 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*.

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

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

[6] Glover, F., Kochenberger, G., and Du, Y. (2019). Quantum bridge analytics I: a tutorial on formulating and using QUBO models. *4OR*, 17(4):335–371.

[7] Guerreschi, G. G. and Matsuura, A. Y. (2019). QAOA for max-cut requires hundreds of qubits for quantum speed-up. *Scientific reports*, 9(1):6903.

[8] Herrman, R., Ostrowski, J., Humble, T. S., and Siopsis, G. (2021). Lower bounds on circuit depth of the quantum approximate optimization algorithm. *Quantum Information Processing*, 20(2):1–17.

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

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

[11] Lucas, A. (2014). Ising formulations of many np problems. *Frontiers in Physics*, 2:5.

[12] Martello, S., Pisinger, D., and Toth, P. (2000). New trends in exact algorithms for the 0–1 knapsack problem. *European Journal of Operational Research*, 123(2):325–332.

[13] McGeoch, C. C. (2014). Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93.

[14] Montanaro, A. (2016). Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8.

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

[16] Onn, S. (2010). Nonlinear discrete optimization. *Zurich Lectures in Advanced Mathematics, European Mathematical Society*.

[17] 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.

[18] Pisinger, D. and Toth, P. (1998). Knapsack problems. In *Handbook of combinatorial optimization*, pages 299–428. Springer.

[19] Pusey-Nazzaro, L. et al. (2020). Adiabatic quantum optimization fails to solve the knapsack problem. *arXiv preprint arXiv:2008.07456*.

[20] Rieffel, 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.

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

[22] Ruan, Y., Marsh, S., Xue, X., Li, X., Liu, Z., and Wang, J. (2020). Quantum approximate algorithm for np optimization problems with constraints. *arXiv preprint arXiv:2002.00943*.

[23] Singh, S. P. (2020). The ising model: Brief introduction and its application. In *Solid State Physics-Metastable, Spintronics Materials and Mechanics of Deformable Bodies-Recent Progress*. IntechOpen.

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

[25] 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.

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

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

[28] 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.

[29] Wolsey, L. A. (2020). *Integer programming.* John Wiley & Sons.

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