INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 32 (2025) 6–17
DOI: 10.1111/itor.13420

# New advances for quantum-inspired optimization

Yu Du[a,*], Haibo Wang[b,c], Rick Hennig[c], Amit Hulandageri[c], Gary Kochenberger[c] and Fred Glover[c]

[a] *Business School, University of Colorado at Denver, Denver, CO 80217, USA*
[b] *Business School, Texas A&M International University, Laredo, TX 78041, USA*
[c] *Entanglement Inc., Boulder, CO 80302, USA*
*E-mail: yu.du@ucdenver.edu [Du]; haibo@entanglement.ai [Wang]; rick@entanglement.ai [Hennig]; ahulandageri@entanglement.ai [Hulandageri]; gary@entanglement.ai [Kochenberger]; fred@entanglement.ai [Glover]*

Received 22 January 2023; received in revised form 7 December 2023; accepted 7 December 2023

**Abstract**

Advances in quantum computing with applications in combinatorial optimization have evolved at an increasing rate in recent years. The quadratic unconstrained binary optimization (QUBO) model is at the center of these developments and has become recognized as an effective alternative method for representing a wide variety of combinatorial optimization problems. Additional momentum has resulted from the arrival of quantum computers and their ability to solve the Ising spin glass problem, another form of the QUBO model. This paper highlights advances in solving QUBO models and extensions to more general polynomial unconstrained binary optimization (PUBO) models as important alternatives to traditional approaches. Computational experience is provided that compares the performance of unique quantum-inspired metaheuristic solvers—the Next Generation Quantum (NGQ) solver for QUBO models and the NGQ-PUBO solver for PUBO models—with the performance of CPLEX and the Dwave quantum advantage solver. Extensive results, including experiments with a set of large set partitioning problems representing the largest QUBO models reported in the literature to date, along with maximum diversity and max cut problem sets, disclose that our solvers outperform both CPLEX and Dwave by a wide margin in terms of both computational time and solution quality.

*Keywords:* QUBO; combinatorial optimization; integer programming; quantum computing

## 1. Introduction

In recent years, the unifying nature of the QUBO model (whose acronym stands for quadratic unconstrained binary optimization) for modeling and solving many combinatorial problems has taken on heightened importance due to the emergence of quantum computers as documented in

*Corresponding author.

studies such as Kochenberger et al. (2004), Kochenberger et al. (2014), Lucas (2014), and Anthony et al. (2017).

The QUBO model is a mathematical framework used to represent and solve a wide range of real-world combinatorial optimization problems that involve finding the best combination of choices from a large set of possibilities. Examples include the traveling salesman problem, job scheduling, protein folding, and portfolio optimization, among many others.

The quantum computing community emphasizes QUBO models because they are mathematically equivalent to a problem in physics known as the spin glass problem (Zhu et al., 2015). QUBO problems can be directly mapped to quantum annealers, such as those provided by D-Wave Systems, or to gate-based quantum computers, like those from IBM, Google, and others. This makes the QUBO model a natural choice for leveraging the power of quantum computing in optimization tasks as documented in Aramon et al. (2019), Glover et al. (2019), Glover et al. (2020) and Glover et al. (2022a, 2022b).

The QUBO model represents a significant departure from traditional modeling in exploiting the influence of the constraints to shape the search process. Rather than impose restrictions as traditional constraints, QUBO models modify the original objective function with penalties that help guide the search process. From the original constraints, quadratic functions are incorporated into the objective function as penalties causing the objective function to degrade as the search process moves into infeasible regions while having a neutral impact for feasible solutions. Traditional methods sometimes employ Lagrangian techniques to achieve similar outcomes, but these techniques have significant drawbacks not shared by penalty approaches of the QUBO models as observed in Glover et al. (2022a).

When incorporated into the original objective function, the quadratic penalty functions yield a new, unconstrained function to be optimized. It is through these penalties that the influence of the constraints impacts the search process.

Mathematically, the QUBO model is represented by:

$$QUBO : opt\ x_0 = x^t Q x \tag{1}$$

where x is a vector of binary variables, and Q is a symmetric matrix of constants.

There are many hybrid solvers that combine classical and quantum computing resources to solve optimization problems more efficiently. By formulating problems as QUBOs, researchers can develop hybrid algorithms that take advantage of quantum hardware's unique capabilities while also leveraging classical computing. The potential of quantum computing has motivated significant research and development efforts to advance the QUBO model and develop practical algorithms for solving QUBO problems on classical and quantum hardware. Examples of existing quantum algorithms are the variational quantum eigensolver (Peruzzo et al., 2014), the quantum approximate optimization algorithm (Farhi and Goldstone, 2014), and the quantum annealing algorithm (Johnson et al., 2011).

In this paper, we report new advances for solving QUBO problems that are embodied in a specially designed quantum-inspired algorithm called the Next Generation Quantum (NGQ) solver from Entanglement (2023), which has proved extremely successful in solving very large-scale QUBO models. Head-to-head competition with other approaches indicates the effectiveness of the NGQ approach in terms of both solution quality and computational time.

In addition to QUBO models, higher-order polynomial unconstrained binary optimization (PUBO) problems are frequently encountered in the quantum computing community and beyond. Typical examples here include Boolean satisfiability problems, bi-clustering problems, and molecular conformation problems. These higher-order models can be solved as is, or they can be translated as equivalent QUBO models *via* a quadratization transformation process as described by Rosenberg (1972), Boros and Hammer (2002), Boros and Gruber (2014), and Verma et al. (2021). These procedures create an equivalent model at the cost of introducing many new variables. A modified version of our NGQ solver, called NGQ-PUBO, solves PUBO models directly without the conversion to a QUBO model, often with greater efficiency as shown in Section 4.

The rest of this paper is organized as follows. Section 2 reports large-scale QUBO testing on three major classes of problems: set partitioning, maximum diversity problems (MDPs), and max cut problems. Section 3 presents computational experience using a test bed of third- and fourth-order polynomial models with comparisons to results obtained from CPLEX and Dwave Advantage. Section 4 summarizes our findings and draws some conclusions.

## 2. Large-scale QUBO applications

### *2.1. Solving large-scale set partitioning problems in QUBO*

The set partitioning problem can be formulated as

$$\min \sum_{j=1}^{n} c_j x_j, \tag{2}$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j = 1 \; for \; i = 1, \ldots, m. \tag{3}$$

Applying the transformation introduced by Glover et al. (2022a), the set partitioning problem becomes a QUBO problem without introducing new variables, whose objective function takes the following form:

$$\min \sum_{(j=1)}^{n} c_j x_j + P \sum_{i} \Big( \sum_{(j=1)}^{n} a_{ij} x_{ij} - 1 \Big)^2, \tag{4}$$

which can be re-written in the standard QUBO form as

min $x^t Q x$ + an additive constant.

In this section, we give results obtained from CPLEX, our NGQ QUBO solver and the Dwave Advantage solver, which is the leading hybrid quantum QUBO solver. We note that like CPLEX and the Dwave solver, NGQ is a proprietary algorithm. As an overview, NGQ is a quantum-inspired solver that combines different metaheuristic ideas, such as classical tabu search,

Table 1
Next Generation Quantum (NGQ) versus CPLEX versus Dwave Advantage on medium-size problems

| | | CPLEX | | Dwave Advantage | | | NGQ | |
|---|---|---|---|---|---|---|---|---|
| ID | Vars | OFV | Time (s) | OFV | Time (s) | Gap % | OFV | Time (s) |
| SPP01a | 6000 | 10,872 | 7851 | 11,156 | 124 | 2.61% | 10,872 | 53 |
| SPP01c | 6000 | 22,860 | 2598 | 22,917 | 87 | 0.25% | 22,860 | 275 |
| SPP01d | 6000 | 14,793 | 14,115 | 14,917 | 106 | 0.84% | 14,793 | 12 |
| SPP02a | 8000 | 14,959 | 15,348 | 15,332 | 229 | 2.49% | 14,959 | 34 |
| SPP02b | 8000 | 9621 | 18,071 | 10,037 | 253 | 4.32% | 9621 | 74 |
| SPP02c | 8000 | 30,425 | 16,423 | 30,929 | 144 | 1.66% | 30,425 | 95 |
| SPP02d | 8000 | 19,882 | 10,191 | 19,882 | 188 | 0.33% | 19,816 | 19 |

diversification-driven tabu search, adaptive memory programming, scatter search, path relinking, cross-over, and local search (Glover, 1997a; Glover 1997b). This metaheuristic combination strategy has proven to be highly competitive in terms of both solution quality and computational efficiency relative to other approaches for solving the QUBO model as shown in the tables below.

In the following tables, we present results on medium-size, large-size, and very large-size test problems. Except as noted otherwise, all results were obtained on a desktop PC with 11th Gen Intel Core i9-11900H processor at 2.50GHz and 32 GB of RAM. The Dwave Advantage results were obtained directly from the Dwave Leap quantum cloud service Advantage quantum computer.

CPLEX was given a time limit of five hours for each problem, and the CPLEX results were derived from the standard linear model for set partitioning. The NGQ and Dwave Advantage results were obtained from the QUBO formulation of the problems. Times shown in the tables are "times to best solution."

### *2.1.1. Results for medium-sized problems*

As shown in Table 1, both CPLEX and NGQ found optimal solutions for the first six of these modest-sized problems. NGQ obtained a better solution than CPLEX on the last problem and outperformed CPLEX on "time to best" by a wide margin on all eight problems. The Dwave Advantage performance lagged behind that of NGQ and CPLEX, both in terms of solution quality and time to best.

### *2.1.2. Results for large instances*

The quantum solver, Dwave Advantage, exhibited erratic and inferior results on the larger problem instances, failing in many cases to return a solution and otherwise giving substantially inferior results. Accordingly, we left Dwave Advantage out of the next two tables.

Table 2 shows that NGQ quickly found best-known solutions for all 24 problems, while CPLEX was able to find best-known solutions for only 11 of the 24 problems. NGQ had a "time to best" advantage over CPLEX that typically ranged from 1 to 3 orders of magnitude.

### *2.1.3. Results from very large instances*

For the very large problems of Table 3, CPLEX could not find the best-known solution to any of these problems within the time limit of five hours. NGQ quickly provided best-known

Table 2
CPLEX versus NGQ on large-size problems

| | | | | CPLEX | | NGQ | |
|---|---|---|---|---|---|---|---|
| Instance | Vars | Constraints | Density % | OFV | Time (s) | OFV | Time (s) |
| 1 | 10,000 | 1000 | 0.25 | 7292 | 947 | 7292 | 378.2 |
| 2 | 10,000 | 1000 | 50 | 4543 | 1543 | 4543 | 11.6 |
| 3 | 10,000 | 5000 | 25 | 37,968 | 284 | 37,968 | 5.4 |
| 4 | 10,000 | 5000 | 50 | 24,297 | 8683 | 24,297 | 1337 |
| 5 | 15,000 | 1500 | 25 | 10,930 | 66 | 10,930 | 17.5 |
| 6 | 15,000 | 1500 | 50 | 7174 | 2176 | 7047 | 77 |
| 7 | 15,000 | 7500 | 25 | 57,834 | 993 | 57,419 | 706.7 |
| 8 | 15,000 | 7500 | 50 | 37,962 | 2373 | 37,671 | 556.8 |
| 9 | 20,000 | 2000 | 25 | 14,900 | 9833 | 14,900 | 1535.1 |
| 10 | 20,000 | 2000 | 50 | 9412 | 369 | 9412 | 3.6 |
| 11 | 20,000 | 10,000 | 25 | 77,448 | 2119 | 77,198 | 1729.1 |
| 12 | 20,000 | 10,000 | 50 | 50,188 | 4786 | 50,188 | 5.7 |
| 13 | 25,000 | 2500 | 25 | 18,517 | 589 | 18,498 | 10 |
| 14 | 25,000 | 2550 | 50 | 12,008 | 1847 | 11,923 | 813.8 |
| 15 | 25,000 | 12,500 | 25 | 96,690 | 548 | 96,445 | 1474.3 |
| 16 | 25,000 | 12,500 | 50 | 63,173 | 18,903 | 63,156 | 98 |
| 17 | 30,000 | 3000 | 25 | 22,405 | 859 | 22,405 | 14.1 |
| 18 | 30000 | 3000 | 50 | 14,457 | 2507 | 14,457 | 1551.7 |
| 19 | 30,000 | 15,000 | 25 | 115,950 | 16,031 | 115,687 | 410.3 |
| 20 | 30,000 | 15,000 | 50 | 76,276 | 17,393 | 75,684 | 11.5 |
| 21 | 40,000 | 4000 | 25 | 30,592 | 2532 | 30,445 | 894.1 |
| 22 | 40,000 | 4000 | 50 | 19,815 | 7184 | 19,558 | 11.5 |
| 23 | 40,000 | 20,000 | 25 | 155,162 | 19,152 | 155,069 | 393.4 |
| 24 | 40,000 | 20,000 | 50 | 101,835 | 10,286 | 101,835 | 12.9 |

Table 3
CPLEX versus NGQ on very large-size problems

| | | | | CPLEX | | NGQ | |
|---|---|---|---|---|---|---|---|
| ID | Vars | Constraints | Density | OFV | Time (s) | OFV | Time (s) |
| spp50k | 50,000 | 10,000 | 25 | 76,903 | 18,144 | 76,402 | 2315 |
| spp60k | 60,000 | 12,000 | 25 | 92,293 | 16,060 | 91,912 | 2876 |
| spp70k | 70,000 | 14,000 | 25 | 109,168 | 19,084 | 108,112 | 272 |
| spp80k | 80,000 | 16,000 | 25 | 125,139 | 18,600 | 123,890 | 175 |
| spp90k | 90,000 | 18,000 | 25 | 140,223 | 15,278 | 139,269 | 1804 |
| spp100k | 100,000 | 20,000 | 25 | 154,694 | 19,509 | 154,351 | 622 |

solutions for all problems, outperforming CPLEX in terms of solution quality and time. Note that these QUBO models are the largest reported in the literature to date. This study confirms earlier successes of the QUBO model on smaller test problems for SPP reported by Lewis et al. (2008).

### 2.2. Solving large-scale MDPs via QUBO

The MDP consists of selecting a subset of $m$ elements from a set of $n$ elements to maximize the sum of the distances between the chosen elements, where the definition of distance between elements is customized to specific applications. Most applications assume that each element can be represented by a set of attributes as follows. Let $s_{ik}$ be the state or value of the $k$th attribute of element $i$, where $k = 1, \ldots, K$. Then the distance between elements $i$ and $j$ may be defined as

$$d_{ij} = \sqrt{\sum_{k=1}^{K} \left(s_{ik} - s_{jk}\right)^2}. \tag{5}$$

In this case, $d_{ij}$ simply constitutes the Euclidean distance between $i$ and $j$. The distance values are then used to formulate the MDP as a quadratic binary problem, where the variable $x_i$ takes the value 1 if element $i$ is selected and 0 otherwise, $i = 1, \ldots, n$:

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j, \tag{6}$$

$$\text{subject to } \sum_{i=1}^{n} x_i = m,$$

$$x_i = 0, 1, 1 \leq i \leq n \tag{7}$$

As before, we can use the transformation introduced by Glover et al. (2022a) to convert this problem to an unconstrained QUBO model. Our testing here was carried out on a set of large instances from the MDPLIB2.0 by Marti et al. (2021). Comparisons highlight the performance of our QUBO solver, NGQ, with that of the opposition-based memetic search (OBMA) method, a state-of-the-art algorithm specifically for the MDP by Zhou et al. (2017), which lacks the ability to solve other problems formulated as QUBO problems.

As shown in Table 4, both NGQ and OBMA produced optimal solutions for each problem instance in this test bed. Times to best were more uniform for OBMA, which is a specialized solver. However, NGQ gave substantially shorter solution times for 15 out of 20 instances.

### 2.3. Solving large-scale maximum cut problems in QUBO

This section reports testing carried out on some large max cut problems, which can be described as follows: Given an undirected graph $G(V, E)$, the maximum cut problem seeks to partition V into two sets such that the number of edges between the two sets (the cut), is as large as possible.

This problem can be modeled as a QUBO by introducing binary variables $x_j = 1$ if vertex $j$ is in one set and $x_j = 0$ if it is in the other set. Viewing a cut as severing edges joining two sets,

Table 4
NGQ versus opposition-based memetic search (OBMA) on large maximum diversity problem (MDP) instances

| File | N | M | OBMA | NGQ | Diff | NGQ time | OBMA time |
|---|---|---|---|---|---|---|---|
| MDG-c_1_n3000_m300.txt | 3000 | 300 | 24,926,344 | 24,926,344 | 0 | 753.5 | 251.3 |
| MDG-c_2_n3000_m300.txt | 3000 | 300 | 24,912,546 | 24,912,546 | 0 | 47.6 | 286.6 |
| MDG-c_3_n3000_m300.txt | 3000 | 300 | 24,905,218 | 24,905,218 | 0 | 658.5 | 239.4 |
| MDG-c_4_n3000_m300.txt | 3000 | 300 | 24,909,710 | 24,909,710 | 0 | 82 | 276.2 |
| MDG-c_5_n3000_m300.txt | 3000 | 300 | 24,895,704 | 24,895,704 | 0 | 83.5 | 212.5 |
| MDG-c_6_n3000_m400.txt | 3000 | 400 | 43,444,948 | 43,444,948 | 0 | 246.3 | 290.8 |
| MDG-c_7_n3000_m400.txt | 3000 | 400 | 43,477,267 | 43,477,267 | 0 | 3.6 | 111.7 |
| MDG-c_8_n3000_m400.txt | 3000 | 400 | 43,465,572 | 43,465,572 | 0 | 45.6 | 163.6 |
| MDG-c_9_n3000_m400.txt | 3000 | 400 | 43,448,137 | 43,448,137 | 0 | 153.8 | 72.5 |
| MDG-c_10_n3000_m400.txt | 3000 | 400 | 43,465,561 | 43,465,561 | 0 | 4.8 | 115.1 |
| MDG-c_11_n3000_m500.txt | 3000 | 500 | 67,021,132 | 67,021,132 | 0 | 3.7 | 335 |
| MDG-c_12_n3000_m500.txt | 3000 | 500 | 67,014,170 | 67,014,170 | 0 | 205.3 | 302.8 |
| MDG-c_13_n3000_m500.txt | 3000 | 500 | 67,024,373 | 67,024,373 | 0 | 40.4 | 380.1 |
| MDG-c_14_n3000_m500.txt | 3000 | 500 | 67,030,190 | 67,030,190 | 0 | 129.2 | 276.3 |
| MDG-c_15_n3000_m500.txt | 3000 | 500 | 67,056,334 | 67,056,334 | 0 | 109.4 | 269 |
| MDG-c_16_n3000_m600.txt | 3000 | 600 | 95,638,929 | 95,638,929 | 0 | 19.6 | 270.8 |
| MDG-c_17_n3000_m600.txt | 3000 | 600 | 95,571,113 | 95,571,113 | 0 | 80 | 312.8 |
| MDG-c_18_n3000_m600.txt | 3000 | 600 | 95,532,141 | 95,532,141 | 0 | 265.6 | 292.1 |
| MDG-c_19_n3000_m600.txt | 3000 | 600 | 95,599,164 | 95,599,164 | 0 | 469.2 | 343.7 |
| MDG-c_20_n3000_m600.txt | 3000 | 600 | 95,584,482 | 95,584,482 | 0 | 46.4 | 299.9 |

the quantity $x_i + x_j - 2x_i x_j$ identifies whether the edge $(i, j)$ is in the cut. Thus, the problem of maximizing the number of edges in the cut can be formulated as

$$\text{Maximize} y = \sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j), \tag{8}$$

which is an instance of $QUBO: \max\ y = x^t Q x$.

Early articles such as those by Boros and Hammer (1991), Kochenberger et al. (2013), Wang et al. (2017), Dunning et al. (2018), and Furini and Traversi (2019) studied the maximum cut problems and suggested the possible use of QUBO model.

The difficulty of the maximum cut problem depends on the regularity and connectivity of the underlying graph. Following an existing trend in the optimization community, we consider the maximum cut on random $d$-regular graphs, where every vertex is connected to exactly $d$ other vertices.

The theoretical cut upper bound for these problems has been shown in Dembo et al. (2017) to be calculated as

$$cut_{ub} = \left(\frac{d}{4} + P^* \sqrt{\frac{d}{4}}\right) n. \tag{9}$$

Here, $P^* = 0.7632$, a universal constant related to the ground state energy of the Sherrington–Kirkpatrick model.

| N | d | Approximation ratio |
|---|---|---|
| 1000 | 3 | 0.978 |
| 2154 | 3 | 0.979 |
| 4642 | 3 | 0.977 |
| 10,000 | 3 | 0.974 |
| 21,544 | 3 | 0.971 |
| 46,416 | 3 | 0.971 |
| 100,000 | 3 | 0.969 |
| 215,443 | 3 | 0.968 |
| 464,159 | 3 | 0.970 |
| 1,000,000 | 3 | 0.968 |
| 1000 | 5 | 0.987 |
| 2154 | 5 | 0.989 |
| 4642 | 5 | 0.987 |
| 10,000 | 5 | 0.987 |
| 21,544 | 5 | 0.985 |
| 46,416 | 5 | 0.985 |
| 100,000 | 5 | 0.983 |
| 215,443 | 5 | 0.982 |
| 464,159 | 5 | 0.984 |
| 1,000,000 | 5 | 0.982 |

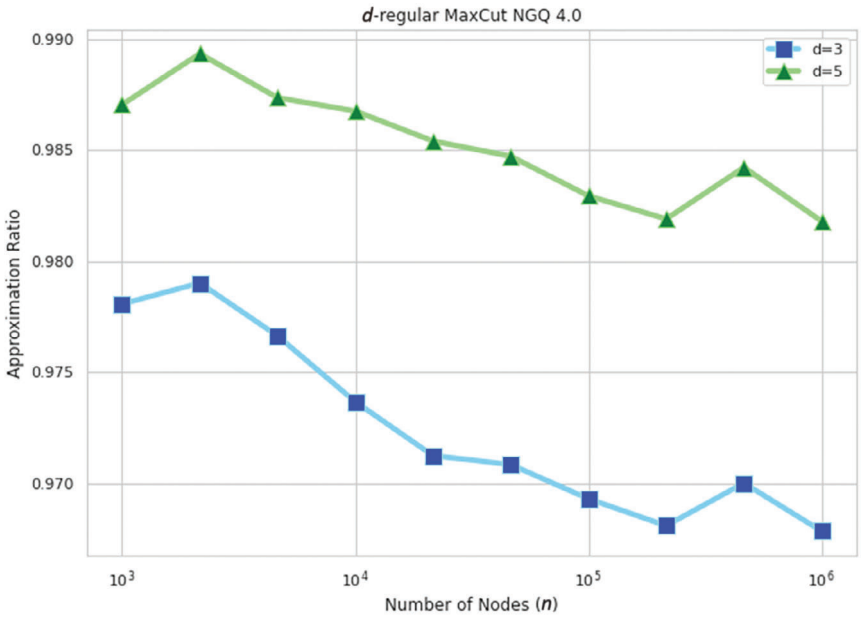


Fig. 1. Next Generation Quantum (NGQ) approximation ratio—10 minutes.

For very large graphs up to a million variables, numerical benchmarks are unavailable. As a result, we use an approximation ratio to measure NGQ performance on regular graphs with $d = 3$ and $d = 5$ with 10 runs of 10 minutes each.

Figure 1 shows that NGQ achieves a mean approximation ratio of 0.972 for $d = 3$ and 0.985 for $d = 5$ for graphs of up to a million nodes, which surpasses the performance reported in the literature, such as Schuetz et al. (2022), using physics-inspired graph neural networks.

## 3. PUBO problems: going beyond QUBO

While QUBO models have become widely adopted in recent years as a useful form of problem representation, many important problems are more naturally modeled as higher-order polynomials. We refer to such models as PUBO models. Formally, these Boolean nonlinear models can be expressed as

$$\min x_0 = c_0 + \sum \left( c_p F_p : p \in P \right), \tag{10}$$

where $F_p$ is a product of components of the $x$ vector given by

$$F_p = \prod \left( x_i : i \in N_p \right), \; N_p \subset N, \tag{11}$$

Table 5
A comparison of NGQ-polynomial unconstrained binary optimization (PUBO) and CPLEX on some third-order minimization problems

| | | | | NGQPUBO | | CPLEX* | |
|---|---|---|---|---|---|---|---|
| *d* | *n* | *m* | Q size | Obj | Time | Obj | Time |
| 3 | 200 | 1000 | 825 | **−1068.94** | 3.12 | −1062.14 | 39.87 |
| 3 | 200 | 1000 | 836 | **−1165.5** | 1.706 | −1156.92 | 19.31 |
| 3 | 200 | 1000 | 843 | **−1083.14** | 2.084 | −1073.79 | 194.22 |
| 3 | 200 | 1000 | 829 | **−1037.82** | 2.97 | −1036.64 | 132.88 |
| 3 | 200 | 1000 | 824 | **−1065** | 2.64 | −1049.49 | 16.99 |

where $x = (x_1, x_2, \ldots, x_n)$ is composed of binary variables, $x_i \in \{0, 1\}$ for $i \in N = \{1, \ldots, n\}$. The coefficients $c_p$ for $p \in P$ are non-zero scalars. Each variable $x_i$ in the product-defining $F_p$ appears only once, noting that $x_i^h = x_i$ for $x_i$ binary, which renders powers $h$ of $x_i$ other than $h = 1$ irrelevant (Glover et al., 2011; Boros and Gruber, 2014).

Due to the limited availability of PUBO solvers, higher-order PUBO problems are typically solved by first converting them to an equivalent QUBO model. There are many discussions in the literature on the translation from the PUBO model to a QUBO model, notably including (Rosenberg, 1972; Rodriguez-Heck 2018; Boros et al., 2020). However, the conversion comes at the cost of introducing many new variables. For the results presented in this section, the conversion to an equivalent QUBO model was accomplished by using the well-known Rosenberg procedure. For example, to convert a qubic function in binary variables to an equivalent quadratic function, a pair of variables, $x_i$ and $x_j$, are replaced by a new binary variable, $y_{ij}$ , in each cubic term containing the product $x_i x_j$ and a penalty term of the form $P(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ added to the objective function, where $P$ is a suitably large scalar penalty. In this manner, each cubic term can be converted to a quadratic term, yielding an equivalent QUBO model. Higher-order problems can be converted to QUBOs by repeating this procedure.

As previously noted, the modified version of our NGQ solver, NGQ-PUBO, solves PUBO models directly without the conversion to a QUBO model. Tables 5 and 6 show that this direct PUBO solution algorithm often solves PUBO problems with greater efficiency. Computational experiments were conducted using the PUBO/QUBO instances introduced by Verma and Lewis (2020). The instances were generated using the number of variables $n$, the number of monomials $m$, and the degree of the PUBO, $d$. The variables that belong to each monomial are chosen uniformly and independently among the available $n$ variables. The coefficient of each monomial is determined independently and uniformly from the interval [−10,10].

In Table 5 below, we provide results for some third-degree polynomial problems with 200 variables and 1000 terms, showing the comparative performance of our solver with that of CPLEX. The CPLEX results were derived from the QUBO equivalent version of each third-degree problem. Note that the table also shows the number of variables in the resulting QUBO-converted instances.

While none of these problems proved to be difficult, Table 5 shows that across the five problems, NGQ-PUBO outperformed CPLEX in terms of both solution quality and solution time.

Table 6
A comparison of NGQ-PUBO and the Dwave Advantage solver on some third- and fourth-order minimization problems

| | | | | NGQPUBO | | DWave Kerberos | | |
|---|---|---|---|---|---|---|---|---|
| *d* | *n* | *m* | Q size | Obj | Time | Obj | Time | Gap % |
| 3 | 400 | 1000 | 1333 | **−1556.24** | 4.478 | −1284.06 | 17.549 | −17.4893 |
| 3 | 400 | 3000 | 2959 | **−2713.74** | 2.526 | −1975.54 | 25.98 | −27.2023 |
| 3 | 400 | 5000 | 4382 | **−3605.82** | 13.828 | −2601.48 | 56.344 | −27.8532 |
| 4 | 400 | 1000 | 2202 | **−1326.15** | 9.512 | −798.419 | 17.611 | −39.7942 |
| 4 | 400 | 3000 | 5308 | **−2486.05** | 41.523 | −1198.99 | 48.585 | −51.7712 |
| 4 | 400 | 5000 | 7995 | **−3047.88** | 11.204 | −1222.88 | 47.275 | −59.8776 |

To provide further results and comparisons on PUBO problems, we ran some third- and fourth-degree polynomial problems, providing outcomes derived from the NGQ-PUBO solver and the quantum solver, Dwave Advantage. Again we note that NGQ-PUBO solved the third- and fourth-order models directly, while Dwave Advantage solved the QUBO version of each problem created by using the Rosenberg (1972) transformation. As also remarked previously, the re-casting into a QUBO model greatly increases the size of the problem to be solved. Our results of this testing are shown in Table 6.

As seen in Table 6, NGQ-PUBO quickly found high-quality solutions, outperforming the quantum solver, Dwave Advantage, in terms of both solution quality and solution time by a wide margin on all six problems. The Dwave Advantage solution quality is off by double digits across the board and in the last problem by more than 50 %. These results highlight the potential advantage of solving PUBO problems directly rather than resorting to a QUBO reduction.

## 4. Summary and Conclusion

The QUBO model, with a new quantum-inspired solver NGQ, has proven to be very successful for many important problems in industry and government. We demonstrate in this paper that it is capable of producing best-known solutions for very large set partitioning problems, max diversity problems and max cut problems. Across the board, the NGQ solver quickly found best-known solutions for all problems tested. We find similar results in other problem domains like clique partitioning problems, coloring problems, scheduling problems, classification problems, and many more. As computers and solvers, both conventional and quantum, continue to improve, the successes of the QUBO approach will likely continue to grow.

PUBO models, solved directly by metaheuristic methods rather than reformulating them as QUBO models, can yield significantly better outcomes than attempting to solve their QUBO counterparts, which in general are substantially larger and challenging to solve. Our work here employed the commonly used Rosenberg conversion for reducing higher-order models to equivalent QUBO models. Other methods for such conversions, as referenced earlier, require the introduction of fewer variables, which may to some extent reduce the advantage we currently observe of solving PUBOs directly rather than solving the equivalent QUBO model. Exploring these alternative conversions is part of our work in progress.

## Acknowledgments

The authors are grateful to the associate editor and two anonymous referees for valuable comments on an earlier version of the paper.

## References

Anthony, M., Boros, E., Crama, Y., Gruber, A., 2017. Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming* 162, 1-2, 115–144.

Aramon, M., Rosenberger, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G., 2019. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics* 7, 48.

Boros, E., Crama, Y., Rodríguez-Heck, E., 2020. Compact quadratizations for pseudo-Boolean functions. *Journal of Combinatorial Optimization* 39, 3, 687–707.

Boros, E., Gruber, A., 2014. On quadratization of pseudo-Boolean functions. arXiv preprint arXiv:1404.6538

Boros, E., Hammer, P., 1991. The max-cut problem and quadratic 0–1 optimization: polyhedral aspects, relaxations and bounds. *Annals of Operations Research* 33, 3, 151–180.

Boros, E., Hammer, P., 2002. Pseudo-Boolean optimization. *Discrete Applied Mathematics* 123, 1, 155–225.

Dembo, A., Montanari, A., Sen, S., 2017. Extremal cuts of sparse random graphs. *The Annals of Probability* 45, 1190.

Dunning, L., Gupta, S., Silberholz, J., 2018. What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO. *INFORMS Journal on Computing* 30, 3, 608–624.

Dwave Quantum Advantage hybrid solver, 2023. https://www.dwavesys.com/solutions-and-products/cloud-platform/ (accessed October 2022).

Entanglement., 2023. https://www.entanglement.com (accessed October 2023).

Farhi, E., Goldstone, F., 2014. A Quantum Approximate Optimization Algorithm, https://arxiv.org/pdf/1411.4028.pdf

Furini, F., Traversi, E., 2019. Theoretical and computational study of several linearisation techniques for binary quadratic problems. *Annals of Operations Research* 279, 387–411.

Glover, F., 1997a. Tabu search and adaptive memory programming—advances, applications and challenges. In Barr, R.S., Helgason, R.V., Kennington, J.L. (eds) *Interfaces in Computer Science and Operations Research*. Springer, Boston, pp. 1–75.

Glover, F., 1997b. A template for scatter search and path relinking. In Hao, J.K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds) *European Conference on Artificial Evolution*. Springer, Berlin, pp. 1–51.

Glover, F., Hao, J.-K., Kochenberger, G., 2011. Polynomial unconstrained binary optimization—part 1. *International Journal of Metaheuristics* 1, 3, 232–256.

Glover, F., Kochenberger, G., Du, Y., 2019. Quantum bridge analytics I: a tutorial on formulating and using QUBO models. *4OR Quarterly Journal of Operations Research* 17, 335–371.

Glover, F., Kochenberger, G., Hennig, R., Du, Y., 2022a. Quantum bridge analytics I: a tutorial on formulating and using QUBO models. *Annuals of Operations Research* 314, 141–183.

Glover, F., Kochenberger, G., Du, Y., 2022b. Applications and computational advances for solving the QUBO model. In Punnen, A.P. (ed.) *The Quadratic Unconstrained Binary Optimization Problem*. Springer, Cham, pp. 39–56.

Glover, F., Kochenberger, G., Ma, M., Du, Y., 2020. Quantum bridge analytics II: QUBO-plus, network optimization and combinatorial chaining for asset exchange. *4OR* 18, 387–417.

Glover, F., Lü, Z., Hao, J.K., 2010. Diversification-driven tabu search for unconstrained binary quadratic problems. *4OR* 8, 3, 239–253.

Johnson, M.W., Amin, M.H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., Chapple, E.M., Enderud, C., Hilton, J.P., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C., Thom, M.C., Tolkacheva, E., Truncik, C.J., Uchaikin, S., Wang, J., Wilson, B., Rose, G., 2011. Quantum annealing with manufactured spins. *Nature*. 473, 7346, 194–198.

Kochenberger, G., Glover, F., Alidaee, B., Rego, C., 2004. A unified modeling and solution framework for combinatorial optimization problems. *OR Spectrum* 26, 2, 237–250.

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lu, Z., Wang, H., Wang, Y., 2014. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization* 28, 1, 58–81.

Kochenberger, G., Hao, J.-K, Lu, S., Wang, H., Glover, F., 2013. Solving large scale Max cut problems via tabu search. *Journal of Heuristics* 19, 4, 565–571.

Lewis, M., Kochenberger, G., Alidaee, B., 2008. A new modeling and solution approach for the set partitioning problem. *Computers and OR* 35, 3, 807–813.

Lucas, A., 2014. Ising formulations of many NP problems. *Frontiers in Physics* 2, https://doi.org/10.3389/fphy.2014.00005.

Marti, R., Duarte, A., Martinez-Gavara, A., 2021. *The MDPLIB 2.0 library of benchmark instances for diversity problems*. University of Valencia. Available at https://www.uv.es/rmarti/paper/mdp.html (accessed October 2022).

Peruzzo, A., McClean, J., Shadbolt, P. et al. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communicat* 5, 4213. https://doi.org/10.1038/ncomms5213

Rodrıguez-Heck, E., 2018. Linear and quadratic reformulations of nonlinear optimization problems in binary variables. *4OR* 17, 221–222.

Rosenberg, G., 1972. 0–1 Optimization and non-linear programming. *Revue Francaise D'automatique, Informatique Recherche Operationnelle (Série Bleue)* 2, 1972, 95–97.

Schuetz, M., Brubaker, J.K., Katzgraber, H.G., 2022. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence* 4, 4, 367–377.

Verma, A., Lewis, M.W., Kochenberger, G., 2021. Efficient QUBO transformation for Higher Degree Pseudo Boolean Functions. ArXiv, abs/2107.11695.

Verma, A., Lewis, M., 2020. Optimal quadratic reformulations of fourth degree pseudo-Boolean functions. *Optimization Letters* 14, 6, 1557–1569.

Wang, Z., Hadfield, S., Jiang, Z., Rieffel, E.G., 2017. The quantum approximation optimization algorithm for MaxCut: a fermionic view. arXiv:1706.02998. https://arxiv.org/abs/1706.02998

Zhou, Y., Hao, J.-K., Duval, B., 2017. Opposition-based memetic search for the maximum diversity problem. *IEEE Transactions on Evolutionary Computation* 21, 5, 731–745.

Zhu, Z., Ochoa, A.J., Katzgraber, H.G., 2015. Efficient cluster algorithm for spin glasses in any space dimension. *Physical Review Letters* 115, 7, 077201.