TU Delft

Delft University of Technology

Document Version
Final published version

Citation (APA)
Zielinski, S., Nublein, J., Kolle, M., Gabor, T., Linnhoff-Popien, C., & Feld, S. (2025). Solving Max-3SAT Using QUBO Approximation. In C. Culhane, G. T. Byrd, H. Muller, Y. Alexeev, Y. Alexeev, & S. Sheldon (Eds.), *Technical Papers Program* (pp. 681-691). (Proceedings - IEEE Quantum Week 2024, QCE 2024; Vol. 1). IEEE. https://doi.org/10.1109/QCE60285.2024.00085

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Solving Max-3SAT Using QUBO Approximation

Sebastian Zielinski
Institute for Informatics
LMU Munich
Munich, Germany
sebastian.zielinski@ifi.lmu.de

Jonas Nüßlein
Institute for Informatics
LMU Munich
Munich, Germany
jonas.nuesslein@ifi.lmu.de

Michael Kölle
Institute for Informatics
LMU Munich
Munich, Germany
michael.koelle@ifi.lmu.de

Thomas Gabor
Institute for Informatics
LMU Munich
Munich, Germany
thomas.gabor@ifi.lmu.de

Claudia Linnhoff-Popien
Institute for Infomatics
LMU Munich
Munich, Germany
linnhoff@ifi.lmu.de

Sebastian Feld
Quantum & Computer Engineering
Delft University of Technology
Delft, The Netherlands
s.feld@tudelft.nl

2024 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3315-4137-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/QCE60285.2024.00085

***Abstract*—As contemporary quantum computers do not possess error correction, any calculation performed by these devices can be considered an involuntary approximation. To solve a problem on a quantum annealer, it has to be expressed as an instance of Quadratic Unconstrained Binary Optimization (QUBO). In this work, we thus study whether systematically approximating QUBO representations of the MAX-3SAT problem can improve the solution quality when solved on contemporary quantum hardware, compared to using exact, non-approximated QUBO representations. For a MAX-3SAT instance consisting of a 3SAT formula with n variables and m clauses, we propose a method of systematically creating approximate QUBO representations of dimension $(n \times n)$, which is significantly smaller than the QUBO matrices of any exact, non-approximated MAX-3SAT QUBO transformation. In an empirical evaluation, we demonstrate that using our QUBO approximations for solving MAX-3SAT problems on D-Wave's quantum annealer Advantage_System6.4 can yield better results than using state-of-the-art exact QUBO transformations. Furthermore, we demonstrate that using naive QUBO approximation methods, based on removing values from exact $(n+m) \times (n+m)$-dimensional QUBO representations of MAX-3SAT instances, is ineffective.**

***Index Terms*—Quadratic Unconstrained Binary Optimization, Combinatorial Optimization, Max-3SAT, Approximation**

I. Introduction

Satisfiability problems are at the core of many theoretical and practical computer science applications. In theoretical computer science, they are often used to prove the hardness of other problem classes. In practical domains, they are used to solve problems in the verification of integrated circuits [1], planning [2], [3], dependency resolution [4], and many more. Given a Boolean formula, the satisfiability problem is concerned with deciding whether a satisfying assignment of Boolean values to the formula's variables exists. The optimization version of the satisfiability problem is called the MAX-SAT problem. The MAX-SAT problem also consists of a Boolean formula, but the goal is to find an assignment that satisfies as many clauses as possible. As there is a polynomial-time transformation from any satisfiability problem to a 3SAT problem, which is a satisfiability problem in which each clause consists of at most 3 literals, often 3SAT problems are studied as a canonical representative of satisfiability problems.

Quantum computing is a computational paradigm that promises to speed up the search for solutions to certain problems (e.g., Shor's algorithm [5], Grover's algorithm [6]). With the recent improvements in the availability and capabilities of quantum hardware systems, researchers began to explore the possibilities of solving MAX-3SAT problems on quantum computers. One area of research studies methods of transforming MAX-3SAT problems to instances of Quadratic Unconstrained Binary Optimization (QUBO), as QUBO is the input format for contemporary quantum annealers and for QAOA algorithms on quantum gate systems. In the last decade, numerous methods have been proposed to transform instances of MAX-3SAT problems into instances of QUBO [7]–[12]. It has also been shown that choosing different transformations can significantly impact the solution quality [13], [14]. Consequently, finding new QUBO transformations that potentially increase a solver's capability of finding high-quality solutions for MAX-3SAT problems is an active field of research.

When using contemporary quantum computers as QUBO solvers, one must consider a major challenge these devices currently face: missing error correction. As these devices currently do not possess error correction, calculations on these devices suffer from the influence of noise. Thus, it can be argued that any calculation of a contemporary quantum computer is an involuntary approximation of a given input problem. Hence, as the subtleties of a QUBO instance are lost during the calculation process due to the influence of noise, evaluating the possibilities of purposefully creating QUBO approximations of given problems as an input to quantum computers seems interesting.

To express a given problem as an instance of QUBO often requires many auxiliary variables to precisely express all the constraints and objectives of a real-world problem as a

979-8-3315-4137-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCE60285.2024.00085

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

quadratic problem. Hence, the idea of a QUBO approximation is to create a simpler quadratic representation of a real-world problem that, when minimized, yields good (enough) solutions to the real-world problem. The goal of these simplifications is to decrease the QUBO size or its density (i.e., the dependencies between different variables). A smaller (resp. less dense) QUBO generally needs fewer physical qubits on quantum hardware. Furthermore, it seems reasonable to assume that reducing the dependencies between different variables might reduce the severity of local calculation errors introduced by noise. Thus, the question we want to address in this paper is: Considering the influence of noise on calculations of contemporary quantum hardware, can solving purposefully crafted QUBO approximations of MAX-3SAT problems yield comparable or even better results than solving exact, non-approximated QUBO representations of MAX-3SAT problems on contemporary quantum hardware? In other words, can we craft QUBO approximations for MAX-3SAT instances such that the error we introduce by approximating the problem does not exceed the error the noise introduces into quantum calculations when solving larger, exact QUBO representations of the same instances?

While there are several studies concerning the question of creating QUBO approximations for given problems (e.g. [15], [16]), the motivation for these studies is that either the objective function is unknown and thus the QUBO representation is approximated using a black-box approach, or that the constraints cannot be expressed precisely as a quadratic optimization problem and thus need approximation. Our work, however, is motivated by the study of Sax et al. [17]. In their study, the authors transformed satisfiability problems into instances of QUBO using Choi's method [9]. Then, they employed a simple QUBO approximation method, namely removing entries from a given QUBO matrix, and observed that approximately 70% of the entries could be removed without a significant decrease in the solution quality.

This work proposes a new method to create QUBO approximations of MAX-3SAT problems systematically. Furthermore, we show that under the current circumstances (missing error correction of quantum hardware), we can receive competitive or even better solutions when solving QUBO approximations of MAX-3SAT instances on D-Wave's quantum annealer Advantage_System6.4 compared to solving exact, non-approximated QUBO representations of the same instances.

Our contributions in this paper are:

1. We introduce a new method of systematically creating effective QUBO approximations for MAX-3SAT problems.
2. In an empirical case study performed on D-Wave's quantum annealer Advantage_System6.4, we demonstrate that our method of systematically creating QUBO approximations for MAX-3SAT problems can yield comparable or even better results than non-approximated, exact QUBO transformations.
3. We provide evidence for the scaling of this approach. That is, we show that our QUBO approximation method still yields good results when larger formulas are solved via a classical QUBO solver.
4. We show that naive methods of creating QUBO approximations for the MAX-3SAT problem, based on the removal of values from given $(n+m) \times (n+m)$-dimensional exact QUBO matrices, generally lead to declines in solution quality.

The remainder of the paper is organized as follows: Section II introduces the MAX-3SAT problem and transformations from MAX-3SAT instances to instances of QUBO. In Section III, we state related work. In Section IV we present our method of systematically approximating QUBO transformations for MAX-3SAT problems. In Section V, we perform empirical evaluations on quantum and classical hardware to show the potential of our systematic QUBO approximation method. We conclude the paper in Sec. VI and state future research opportunities.

II. Foundations

A. Satisfiability Problems

Satisfiability problems are concerned with the satisfiability of Boolean formulas. Thus, we will first define a Boolean formula:

Definition 1 (Boolean formula [18])**.** *Let $x_1, ..., x_n$ be Boolean variables. A* Boolean formula *consists of the variables $x_1, ..., x_n$ and the logical operators $\wedge$, $\vee$, $\neg$. Let $z \in \{0,1\}^n$ be a vector of Boolean values. We identify the value 1 as TRUE and the value 0 as FALSE. The vector z is also called an* assignment*, as it assigns truth values to the Boolean variables $x_1, ..., x_n$ as follows: $x_i = z_i$, where z_i is the $i-th$ component of z. If ϕ is a Boolean formula, and $z \in \{0,1\}^n$ is an assignment, then $\phi(z)$ is the evaluation of ϕ when the variable x_i is assigned the Boolean value z_i. If there exists a $z \in \{0,1\}^n$, such that $\phi(z)$ is TRUE, we call ϕ satisfiable. Otherwise, we call ϕ unsatisfiable [18].*

Satisfiability problems are often given in conjunctive normal form, which we will define next:

Definition 2 (Conjunctive Normal Form [18])**.** *A Boolean formula over variables $x_1, ..., x_n$ is in* Conjunctive Normal Form (CNF) *if it is of the following structure:*

$$\bigwedge_i \left(\bigvee_j y_{i_j} \right)$$

Each y_{i_j} is either a variable x_k or its negation $\neg x_k$. The y_{i_j} are called the literals *of the formula. The terms $(\vee_j y_{i_j})$ are called the* clauses *of the formula. A k*CNF *is a CNF formula, in which all clauses contain at most k literals.*

Given a Boolean formula ϕ in *kCNF*, the satisfiability problem is the task of determining whether ϕ is satisfiable or not. This problem was one of the first problems for which NP-completeness has been shown [19]. In this paper, we will especially consider 3*CNF* problems, which we will refer to as

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

3SAT problems.

The generalization of the SAT problem is called MAX-SAT. In the MAX-SAT problem, we are given a Boolean formula ϕ consisting of m clauses. The task is to find an assignment of truth values to the variables of ϕ such that as many clauses as possible are satisfied. Finding an assignment in the MAX-SAT problem that satisfies m clauses is thus equivalent to solving the corresponding satisfiability problem (i.e., determining whether ϕ is satisfiable or not). MAX-SAT is thus NP-hard as well. We emphasize that in a MAX-3SAT instance, we are given a 3SAT formula (not an MAX-3SAT formula) with the task of finding the maximum number of satisfiable clauses.

B. Quadratic Unconstrained Binary Optimization (QUBO)

In this section we will formally introduce quadratic unconstrained binary optimization (QUBO) and related terminology that will be used in the remainder of this paper.

Definition 3 (QUBO [20])**.** *Let $\mathcal{Q} \in \mathbb{R}^{n \times n}$ be a square matrix and let $x \in \{0,1\}^n$ be an n-dimensional vector of Boolean variables. The QUBO problem is defined as follows:*

$$\textit{minimize} \quad H_{QUBO}(x) = x^T \mathcal{Q} x = \sum_i \mathcal{Q}_{ii} x_i + \sum_{i<j} \mathcal{Q}_{ij} x_i x_j \tag{1}$$

We call $H_{QUBO}(x)$ the (QUBO) energy of vector x. The matrix Q will also be called *QUBO matrix*. Representing a QUBO matrix as an upper triangular matrix is customary. Note specifically that a QUBO matrix is just the matrix representation of the quadratic pseudo-Boolean polynomial shown in Eq. 1.

To transform a given MAX-3SAT instance to an instance of QUBO, we will introduce additional variables that do not correspond to any variables of the given MAX-3SAT instance. We will often say that an assignment $\vec{x} = (x_1 := v_0, \ldots, x_n := v_n), v_i \in \{0,1\}$ of Boolean values to the variables $x_1, \ldots, x_n$ of the MAX-3SAT instance has energy E in Q, by which we mean:

$$min\ \{(\vec{x}, y)^T Q (\vec{x}, y) \mid y \in \{0,1\}^m\} = E \tag{2}$$

Here Q is a QUBO matrix and $(\vec{x}, y)$ is an $(n+m)$-dimensional column vector defined as $(\vec{x}, y) = (x_1 = v_0, \ldots, x_n = v_n, y_1, \ldots, y_m)$. The first n values of the vector $(\vec{x}, y)$ are given by the assignment $\vec{x} = (x_1 := v_0, \ldots, x_n := v_n), v_i \in \{0,1\}$ of Boolean values to the variables $x_1, \ldots, x_n$ of the MAX-3SAT instance. The last m entries represent the values of the auxiliary variables $y_1, \ldots, y_m$.

C. MAX-3SAT as QUBO: The General Idea

In this section we first introduce the idea behind most transformations from MAX-3SAT to QUBO, before explaining specific QUBO transformations used in this work. The main observation is that each clause of a 3SAT formula of a MAX-3SAT instance contains either exactly zero, one, two, or three negations. It is well known that each of these four types of clauses can be expressed as a pseudo-Boolean function as follows [12]:

1. Zero negations $(x_i \vee x_j \vee x_k) : f(x_i, x_j, x_k) = -x_i - x_j - x_k + x_i x_j + x_i x_k + x_j x_k - x_i x_j x_k$
2. One negation $(x_i \vee x_j \vee -x_k) : f(x_i, x_j, x_k) = -1 + x_k - x_i x_k - x_j x_k + x_i x_j x_k$
3. Two negations $(x_i \vee -x_j \vee -x_k) : f(x_i, x_j, x_k) = -1 + x_j x_k - x_i x_j x_k$
4. Three negations $(-x_i \vee -x_j \vee -x_k) : f(x_i, x_j, x_k) = -1 + x_i x_j x_k$

Using this approach, whenever a clause is satisfied, $f(x_1, x_2, x_3) = -1$ holds, and in the case the clause is not satisfied, then $f(x_1, x_2, x_3) = 0$. Thus, a satisfying assignment minimizes $f(x_1, x_2, x_3)$ as desired. By applying the correct pseudo-Boolean function to all clauses of the 3SAT formula of the MAX-3SAT instance and summing all resulting pseudo-Boolean polynomials, one receives a pseudo-Boolean polynomial that is minimized by the best solution(s) to the corresponding MAX-3SAT problem.

Note that the four cases from above create cubic terms $x_i x_j x_k$. As QUBO is concerned with minimizing a quadratic pseudo-Boolean polynomial, we must express the cubic terms as a quadratic polynomial. This can be achieved by quadratic reformulation techniques, which add an extra variable to the problem for each cubic term that gets quadratically reformulated. We refer to [21] for further information on quadratic reformulation techniques. Note that the quadratic reformulation of a cubic term introduces a new auxiliary variable for each cubic term.

D. Chancellor's transformation

The general idea of Chancellor's transformation [8] is to map an arbitrary clause of a satisfiability problem to an instance of QUBO such that all variable assignments that satisfy the clause have the same minimum energy in the QUBO minimization problem. Simultaneously, the single variable assignment, that does not have the minimum energy in the QUBO problem should have a higher energy. By applying QUBO mappings that follow this logic to each clause of the 3SAT formula of a MAX-3SAT instance and adding up the resulting quadratic polynomials, a QUBO $\mathcal{Q}_{instance}$ mapping for the whole MAX-3SAT instance is received. As a consequence of this construction, it is guaranteed that the minimum of $\mathcal{Q}_{instance}$ corresponds to an assignment of Boolean values to the variables of the MAX-3SAT instance, such that the most clauses are satisfied. After some specific logical deductions, which we cannot present in-depth due to space restrictions, Chancellor derives mappings that transform each individual clause of a 3SAT formula to an instance of QUBO. For the four types of clauses defined in Sec. II-C Chancellors mappings are depicted in Table I.

The following example explains the mappings.

Example 1. *Suppose we are given the formula $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee -x_3)$. As the first clause is of type 1 (i.e., has no negations), we apply the transformation shown in Table I(a) and receive the polynomial $P_1 = -2x_1 - 2x_2 - 2x_3 - 2a_1 + x_1x_2 + x_1x_3 + x_1a_1 + x_2x_3 + x_2a_1 + x_3a_1$. Then we*

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Chancellor's QUBO transformations for the four types of clauses

(a) $(x_i \vee x_j \vee x_k)$

	x_i	x_j	x_k	a_l
x_i	-2	1	1	1
x_j		-2	1	1
x_k			-2	1
a_l				-2

(b) $(x_i \vee x_j \vee \neg x_k)$

	x_i	x_j	x_k	a_l
x_i	-1	1		1
x_j		-1		1
x_k				
a_l				-1

(c) $(x_i \vee \neg x_j \vee \neg x_k)$

	x_i	x_j	x_k	a_l
x_i	-1			1
x_j		-1	1	1
x_k			-1	1
a_l				2

(d) $(\neg x_i \vee \neg x_j \vee \neg x_k)$

	x_i	x_j	x_k	a_l
x_i	-1	1	1	1
x_j		-1	1	1
x_k			-1	1
a_l				-1

TABLE III: Nüßlein's QUBO transformations for the four different types of clauses [7]

(a) $(a \vee b \vee c)$

	a	b	c	K
a		2		-2
b				-2
c			-1	1
K				1

(b) $(a \vee b \vee \neg c)$

	a	b	c	K
a		2		-2
b				-2
c			1	-1
K				2

(c) $(a \vee \neg b \vee \neg c)$

	a	b	c	K
a	2	-2		-2
b				2
c			1	-1
K				

(d) $(\neg a \vee \neg b \vee \neg c)$

	a	b	c	K
a	-1	1	1	1
b		-1	1	1
c			-1	1
K				-1

apply the QUBO transformation of Table I(b) to the second clause and receive polynomial $P_2 = -x_1 - x_2 - a_2 + x_1x_2 + x_1a_2 + x_2a_2$. *Note that we have replaced index* l *of variable* a_l *with* 1 *in the first clause and* 2 *in the second clause. As explained in Sec. II-C each clause needs an additional variable to represent cubic terms as a quadratic polynomial. Hence* a_l *is the additional variable of the* l*-th clause. Summing up the polynomials* P_1 *and* P_2 *yields* $P_{final} = P_1 + P_2 = -3x_1 - 3x_2 - 2x_3 - 2a_1 - a_2 + 2x_1x_2 + x_1x_3 + x_1a_1 + x_1a_2 + x_2x_3 + x_2a_1 - x_2a_2 + x_3a_1$ *which can be represented as a QUBO matrix as shown in Tab. II.*

TABLE II: Result of combining the QUBO matrices shown in Tab. Ia) and Tab. Ib).

	x_1	x_2	x_3	a_1	a_2
x_1	-3	2	1	1	1
x_2		-3	1	1	1
x_3			-2	1	
a_1				-2	
a_2					-1

E. Nüßlein's transformation

Nüßlein's transformation [7] employs a similar idea as Chancellor's transformation. Each clause is mapped to an instance of QUBO $\mathcal{Q}$ such that all satisfying assignments of a clause have the same minimal energy in $\mathcal{Q}$. In contrast, the single non-satisfying assignment of the clause has a higher, non-optimal energy. Nüßlein observed that the mapping of the clauses to instances of QUBO could be realized by the mappings depicted in Tab. III.

The QUBO transformations for each of the clause types are applied to the clauses of an MAX-3SAT problem in the same way as demonstrated in Example 1.

F. Pattern QUBO method

As explained in the previous sections, to transform a given MAX-3SAT instance to an instance of QUBO, it suffices to transform each clause of the 3SAT formula associated with the MAX-3SAT problem to an instance of QUBO. All the QUBO instances resulting from the transformation of the clauses will then be combined into a single QUBO instance (as demonstrated in Sec. II-D). The Pattern QUBO method [11] is a meta-approach that can automatically identify all possible transformations from a given clause to an instance of QUBO. As explained in Sec. II-C to transform a clause to an instance of QUBO, an additional variable is needed. Thus, the Pattern QUBO method is given a blank (4×4)-dimensional upper triangular QUBO matrix as input. The user then specifies a set of values that the Pattern QUBO method can use to insert into the blank QUBO matrix. Given this set of values and a specific clause, the Pattern QUBO method exhaustively searches the space of 4×4-dimensional upper triangular QUBO matrices that only consist of values of the user-specified set of values. The goal of this exhaustive search procedure is to find QUBO matrices in which all the minima correspond to satisfying solutions to the given clause. This way, the Pattern QUBO method finds all possible methods of transforming a given clause to an instance of QUBO. Similarly to the previously explained methods by Chancellor and Nüßlein, by applying the respective transformations to the clauses of the 3SAT formula and combining the resulting QUBO matrices, we receive a QUBO representation of the MAX-3SAT instance.

III. Related Work

In a paper by Sax et al. [17], the authors studied the creation of QUBO approximations for several problem classes by randomly removing entries from a given initial QUBO matrix. By removing values from a QUBO matrix, fewer physical qubits are needed to solve the QUBO on a quantum annealer. This study thus researched how much the problem size (resp. the number of needed physical qubits) could be reduced without worsening the solution quality too much. For the class of 3SAT problems, the authors transformed each 3SAT instance to an instance of QUBO using Choi's method [9]. The authors continuously removed values from the initial QUBO created by Choi's method until only diagonal entries were left in the QUBO matrix. They observed that up to 70% of the initial QUBO entries could be removed without observing a significant decline in the solution quality. Apart from this study, QUBO approximation has mostly been studied in contexts

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

where an objective function that is to be expressed as a QUBO instance is unknown or when objectives or constraints of a problem cannot be encoded effectively in a quadratic model. In [15], the authors present a method to approximate the low-energy spectrum of a problem as a QUBO using a black-box approach. This method involves concurrently conducting a regression on the low-energy spectrum and differentiating between high-energy and low-energy states through classification. They demonstrate that this approach yields a significantly more accurate approximation of the low-energy spectrum, resulting in enhanced optimization performance compared to merely performing regression on all sampled states. In a study by Matsumori et al. [16], the authors are concerned with the creation of a QUBO approximation of a design optimization problem, because the objectives and constraints of the design optimization problem cannot be expressed explicitly as a QUBO problem. In their approach, a black-box optimization approach based on the factorization machine [22] is used to create a QUBO approximation of the input problem.

IV. MAX-3SAT QUBO Approximation Methods

In this section, we will define the concept of QUBO approximation and detail our approach to systematically creating such QUBO approximations for MAX-3SAT instances. We will evaluate these approaches in Sec. V.

Definition 4 (QUBO Approximation for MAX-3SAT)**.** *Let ψ be a MAX-3SAT instance. A QUBO instance $\mathcal{Q}$ is a* QUBO approximation *of ψ if there are optimal solutions of ψ that do not have minimal energy in Q (Sec. II-B defines how to calculate the energy of an assignment).*

Intuitively, this definition states that some of the optimal solutions of a given MAX-3SAT instance cannot be found by minimizing a given QUBO approximation for this MAX-3SAT instance.
In the following sections, we will introduce two strategies for creating QUBO approximations of MAX-3SAT problems. In Sec. IV-A, we will define naive QUBO approximation analogously to Sax et al. [17]. In Sec. IV-B, we present the core contribution of this paper: our method of systematically creating QUBO approximations of MAX-3SAT instances.

A. Naive QUBO Approximation

In their study, Sax et al. [17] created QUBO approximations by removing values from QUBO matrices that resulted from applying Choi's transformation to a set of 3SAT formulas. In this paper, we will use the term *naive QUBO approximation* for creating QUBO approximations for MAX-3SAT problems by removing values from a given, non-approximated QUBO representation. Thus, Sax et al. conducted naive QUBO approximation for Choi's method. Choi's method leads to QUBO matrices of dimension $3m \times 3m$, where m is the number of clauses of a 3SAT formula. In this paper, however, we want to begin our study of QUBO approximation methods for MAX-3SAT problems by applying naive QUBO approximation to a class of QUBO transformations that results in $(n+m) \times (n+m)$-dimensional QUBO matrices, where n is the number of variables and m is the number of clauses of a 3SAT problem. This class contains many thousand different QUBO transformations [11]. Thus, we define two methods of creating naive QUBO approximations analogously to Sax et al. [17]:

1. **Min Pruning**: We are given an initial, exact, non-approximated QUBO representation of an MAX-3SAT instance, consisting of the QUBO matrix $\mathcal{Q}$. To create a naive QUBO approximation, we proceed to remove the N smallest entries of $\mathcal{Q}$.
2 **Random Pruning**: Random Pruning generally works similarly, except that we now remove N randomly chosen values, no matter their sign or magnitude.

B. Systematic QUBO Approximation

In this section, we present the core contribution of our paper: We introduce a new and systematic approach to approximating QUBO representations of MAX-3SAT instances. As explained in Sec. II-C, the general idea of $(n+m) \times (n+m)$-dimensional QUBO transformations for MAX-3SAT instances is to transform each clause of the 3SAT formula of an MAX-3SAT instance into an instance of QUBO and sum up all the resulting QUBO representations of the clauses. This way, we receive a QUBO in which all minima correspond to optimal solutions of the MAX-3SAT instance. To accurately transform a 3SAT clause into an instance of QUBO, the QUBO instance needs to contain an additional auxiliary variable (see Sec. II-B). Thus each QUBO matrix that results from transforming a 3SAT clause into an instance of QUBO is of dimension (4×4) (three variables correspond to the variables of the clause and one auxiliary variable). The idea of our approach to systematically approximate QUBO transformations for MAX-3SAT problems is to create $(n \times n)$-dimensional QUBO representations instead of $(n+m) \times (n+m)$-dimensional ones. In particular, we transform each clause to an instance of QUBO with dimension (3×3) instead of (4×4). Hence, our approach does not contain any auxiliary variables. The following theorem shows that transforming 3SAT clauses into instances of QUBO, which consist of (3×3)-dimensional QUBO matrices, is necessarily a QUBO approximation.

Theorem 1. *Let x_1, x_2, x_3 be the variables of a clause of a MAX-3SAT instance. Let S_{SAT} be the set of all assignments of Boolean values to x_1, x_2, x_3 that satisfy the clause and let S_{UNSAT} be the set of all assignments of Boolean values to x_1, x_2, x_3 that do not satisfy the clause. Thus S_{SAT} and S_{UNSAT} are sets consisting of 3-tuples (k_1, k_2, k_3), $k_1, k_2, k_3 \in \{0, 1\}$, where k_i denotes the value of x_i for $i \in \{1, 3\}$. Let $f(x_1, x_2, x_3) := \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_{12} x_1 x_2 + \alpha_{13} x_1 x_3 + \alpha_{23} x_2 x_3$ be a polynomial in x_1, x_2, x_3 with $\alpha_i \in \mathbb{R}$. Then there do not exist choices of $\alpha_1, \alpha_2, \alpha_3, \alpha_{12}, \alpha_{13}, \alpha_{23}, E \in \mathbb{R}$ such that $f(s) = -E$ and $f(u) > -E$ for each $s \in S_{SAT}$ and for each $u \in S_{UNSAT}$.*

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

Proof. We have to prove Theorem 1 for all four types of clauses (see Sec. II-C). Without loss of generality, we proof Theorem 1 for clauses of type 1 like $(x_i \vee x_j \vee x_k)$. We will use a bitstring of length 3 like **100** to denote the assignment $x_i = 1, x_j = 0, x_k = 0$.
Assume to the contrary, that we can choose $\alpha_1, \alpha_2, \alpha_3, \alpha_{12}, \alpha_{13}, \alpha_{23}, E \in \mathbb{R}$ such that $f(s) = -E$ and $f(u) > -E$ for each $s \in S_{SAT}$ and for each $u \in S_{UNSAT}$. As **100**, **010** and **001** are all satisfying assignments for clauses of type 1, $f(100) = f(010) = f(001) = -E$. Thus, $\alpha_1 = \alpha_2 = \alpha_3 = -E$. Since **110** is a satisfying assignment, it follows that $f(110) = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_{12} x_1 x_2 = -E$. Since $\alpha_1 = \alpha_2 = -E$, we conclude that $\alpha_{12} = E$. Similarly, we show that $\alpha_{13} = \alpha_{23} = E$. But then $f(111) = -E - E - E + E + E + E = 0$, which is a contradiction, because **111** is a satisfying assignment and thus it should hold that $f(111) = -E$. □

As a MAX-3SAT clause consists of three Boolean variables, there are $2^3 = 8$ possible assignments of Boolean values to the variables of the clause. For every clause, exactly seven of these assignments satisfy it, and exactly one assignment does not. Theorem 1 shows that it is not possible to encode all 7 satisfying solutions of a clause as the minima of a QUBO minimization problem, consisting of a (3×3)-dimensional QUBO matrix $\mathcal{Q}$. However, the proof of Theorem 1 shows that it is possible to encode 6 of the 7 satisfying solutions of a clause as a minimum of a QUBO minimization problem. This means that one satisfying solution has a non-optimal energy. By using this approximation approach, we save one auxiliary variable per clause of the 3SAT formula of the MAX-3SAT instance, as well as some quadratic coefficients we would have otherwise had to add in an exact QUBO transformation of dimension $(n + m) \times (n + m)$. Consequently, fewer physical qubits are needed to solve these QUBO approximations on quantum hardware. In return for this gain, we can no longer guarantee that every optimal solution to the MAX-3SAT problem also has minimal energy in the approximated QUBO minimization problem. We will demonstrate that this trade-off is worthwhile in Sec. V-C.
We will now show how to systematically find (3×3)-dimensional QUBOs, for which six of the seven satisfying assignments of a clause have the same energy, while the remaining two assignments have a higher energy in $\mathcal{Q}$. Our approach adapts the Pattern QUBO method [11]. Instead of searching within the space of (4×4)-dimensional QUBO matrices and looking for those where all satisfying assignments have the same minimal energy, we are looking for (3×3)-dimensional QUBO matrices in which six of the seven satisfying assignments of a clause have the same minimal energy. Thus, our task is to assign values to the variables $\alpha_1, \ldots, \alpha_{23}$ of the prototype of a (3×3)-dimensional QUBO matrix shown in Table IV, such that the aforementioned condition is satisfied.
Our algorithm to find (3×3)-dimensional QUBO approximations of 3SAT clauses is as follows: The input to the algorithm

TABLE IV: Prototype of a QUBO approximation for a MAX-3SAT Clause

	x_i	x_j	x_k
x_i	α_1	α_{12}	α_{13}
x_j		α_2	α_{23}
x_k			α_3

is a set S of values that can be assigned to $\alpha_1, \ldots, \alpha_{23}$. In this paper, we will use $S := \{-1, 0, 1\}$. Then, we specify for which type of clause (see Sec. II-C) the search method should find QUBO approximations. This is needed because we need six of the seven satisfying assignments of a clause to have the same optimal energy in the resulting QUBO problem. As different types of clauses are satisfied by a different set of assignments, the algorithm needs the respective clause type as an input. The method then performs an exhaustive search, trying all possible combinations of assignments of values of S to variables $\alpha_1, \ldots, \alpha_{23}$, to find QUBO matrices, for which six of the seven possible satisfying assignments of the specified type of clause have minimal energy. Performing this procedure for all four types of clauses, we receive 4 QUBO approximations for each of the four types of clauses within a few seconds.
Remember, as QUBO is just the matrix representation of a quadratic polynomial, the found QUBO approximations for any of the four types of clauses each define a mapping from a 3SAT clause type to a quadratic polynomial. Fixing one QUBO approximation per clause type thus yields a method of transforming each clause of a 3SAT formula of a MAX-3SAT instance into a quadratic polynomial. Summing up all the polynomials that result from applying QUBO approximations of the correct clause type to the clauses of the 3SAT formula of the MAX-3SAT instances yields a QUBO approximation of the whole MAX-3SAT instance (see example in Sec. II-D). As the search resulted in four QUBO approximations for each clause type, there are $256 = 4 \cdot 4 \cdot 4 \cdot 4$ QUBO approximations for an MAX-3SAT instance. To choose one of these 256 QUBO approximations for an evaluation (see Sec. V), we created a single 3SAT formula, according to the method described in V-A. We then used each of the 256 QUBO approximations to create a QUBO instance corresponding to the 3SAT formula. We solved all 256 resulting QUBO instances with D-Wave's tabu search and chose the best-performing approximation (i.e., the one that satisfied most clauses), which is shown in Tab. V, for practical evaluation.

As explained previously, our systematic QUBO approximation approach cannot guarantee that every solution of a MAX-3SAT problem has minimal energy in the resulting QUBO. Interestingly, despite being an approximation, it is still possible that some optimal solutions of the MAX-3SAT instance have minimal energy in the QUBO approximation. We will show this by proving the following theorem.

Theorem 2. *For any MAX-3SAT instance ψ consisting of n variables and m clauses, there is at least one QUBO*

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE V: Approximated QUBOs for the four different types of clauses

(a) $(x_i \vee x_j \vee x_k)$

	x_i	x_j	x_k
x_i	-1	1	1
x_j		-1	1
x_k			-1

(b) $(x_i \vee x_j \vee \neg x_k)$

	x_i	x_j	x_k
x_i		1	-1
x_j			-1
x_k			1

(c) $(x_i \vee \neg x_j \vee \neg x_k)$

	x_i	x_j	x_k
x_i	1	-1	-1
x_j			1
x_k			

(d) $(\neg x_i \vee \neg x_j \vee \neg x_k)$

	x_i	x_j	x_k
x_i	-1	1	1
x_j		-1	1
x_k			-1

approximation $\mathcal{Q}_{opt}$, that yields the best possible solution to the given MAX-3SAT when minimized.

By QUBO approximation, we refer to the specific QUBO approximation we introduced in this secion.

Proof. Suppose $as_{opt} := (x_1 = as_1, x_2 = as_2, \dots x_n = as_n), as_i \in \{0, 1\}$ for $1 \leq i \leq n$ is an optimal assignment of Boolean values to the variables of ψ such that no assignment satisfies more clauses of ψ than as_{opt}. As explained in Sec. II-C and Sec. II-D, to create a QUBO representation of ψ, it suffices to create QUBO representations of each individual clause of ψ and summing up the resulting quadratic polynomials. To create a QUBO representation for which as_{opt} has minimum energy, we thus only need to guarantee that for each clause of ψ, the assignment of Boolean values to the variables of the clause, which is given by as_{opt}, minimizes the QUBO representation of the clause, if it satisfies the clause. As as_{opt} then minimizes each QUBO representation of each individual clause of ψ as_{opt} satisfies, it follows that as_{opt} minimizes the sum of the QUBO representations (quadratic polynomials) of the clauses, which is the QUBO representation of ψ. Thus, to prove Theorem 2, we only need to show that, for each satisfying assignment of a clause, there is at least one QUBO approximation of that clause, in which the respective assignment has minimal energy. Let $S_{SAT} = \{s_1, s_2, \dots, s_7\}$ be the set of the 7 satisfying assignments for a given clause. As stated in this section, the search method finds four approximations for each clause type that assign minimum energy to six of the seven satisfying solutions for the given clause. We choose one of these four QUBO approximations arbitrarily and call it Q_{a_1}. Assume w.o.l.g that s_1, ..., s_6 have minimal energy in a Q_{a_1} All that is left to do is analyze whether s_7 has minimal energy in any of the remaining three QUBO approximations the search method has found for this clause. It turns out, that there is always at least one QUBO approximation, say Q_{a_2} amongst the four QUBO approximations the search method found for the given clause, such that s_7 has minimal energy in Q_{a_2}. Thus, when the optimal assignment for the given clause (given by as_{opt}) is an element of $\{s_1, s_2, \dots, s_6\}$, we choose Q_{a_1} as the QUBO approximation of the clause, else we choose Q_{a_2}. In either case, the assignment for a clause (defined by as_{opt}) minimizes the QUBO approximation for the clause. □

V. Evaluation

This section provides an empirical evaluation of both approximation approaches detailed in Sec. IV.

A. Dataset

To evaluate the efficacy of naive QUBO approximation for $(n + m) \times (n + m)$-dimensional QUBO transformations as well as our systematic approach of creating $(n \times n)$-dimensional QUBO approximations of MAX-3SAT instances, we created a dataset of 100 3SAT formulas. We determined experimentally that the 3SAT formulas should not possess more than 500 clauses, as otherwise $(n + m) \times (n + m)$-dimensional QUBO instances that resulted from transforming the 3SAT formulas could no longer be embedded onto D-Wave's QPU architecture. All 100 formulas were created randomly using the Balanced SAT method [23]. We chose this method because it creates 3SAT formulas for which no approach of exploiting their structure is known [23]. The authors of [23] empirically determined that instances generated by this method are potentially hard to solve for state-of-the-art SAT solvers if they possess approximately 3.6 times more clauses than variables. Using a SAT solver, we observed that all formulas we created, consisting of 500 clauses and 3.6 times more clauses than variables, were unsatisfiable. Thus, all of our 100 3SAT formulas consist of 500 clauses and 145 variables ($\approx$ 3.45 times more clauses than variables). Note that this slight deviation of the ratio of the number of clauses and the number of variables does not compromise the difficulty of the formulas.

B. Evaluation of the Naive QUBO Approximation Methods

As described in Sec. IV-A, the naive QUBO approximation methods work by randomly removing values from an exact, non-approximated QUBO transformation method. In this paper, we will use Chancellor's and Nüßlein's transformations as the exact QUBO transformations. For each of the 100 formulas in our dataset, and for each of the transformation methods (Chancellor and Nüßlein), we perform the following steps:

1. Transform the given 3SAT formula to an instance of QUBO $Q_{initial}$.
2. Calculate the initial number of non-zero, non-diagonal values $N_{initial}$. Calculate $N_{10} := 0.1 \times N_{initial}$.
3. a) Min Pruning: Create the QUBO matrix Q_{10} by removing the N_{10} smallest non-diagonal values from $Q_{initial}$. Continue to create Q_{20} by removing N_{10} non-diagonal values from Q_{10}. This procedure is repeated until Q_{100} is created, which is a QUBO matrix that only contains the main diagonal of $Q_{initial}$ and no non-diagonal entries.
   b) Random Pruning: Generally the same as Min Pruning, except that not the N_{10} smallest non-diagonal entries are removed, but N_{10} randomly chosen non-diagonal entries.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

Thus, for each formula and for each of the QUBO transformations we receive 11 QUBO representations ($Q_{initial}$, Q_{10}, ..., Q_{100}). Each of these QUBO matrices is solved 1000 times on D-Wave's quantum annealer Advantage_System6.4.

From these 1000 samples we generated for each formula, each QUBO transformation (Chancellor, Nüßlein), each pruning method (min pruning, random pruning) and each pruning percentage (0%, 10%, ..., 100% non-zero, non-diagonal values) we select the best answer, i.e., the answer that satisfied the highest number of clauses. We then calculate the average of the best answers for each (pruned) QUBO transformation. The results of this evaluation are shown in Fig. 1. It can be

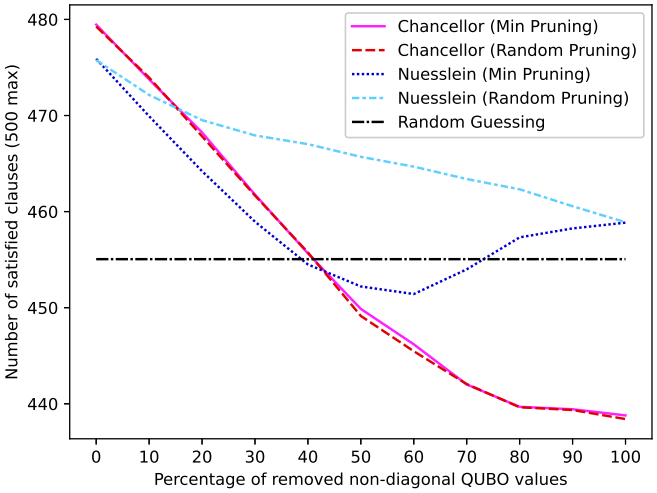


Fig. 1: Average of the best solutions found for each combination of pruning strategy, QUBO transformation and prune percentage.

seen that no method has found a satisfying solution for any formula, as no result yielded 500 satisfied clauses. Observe that randomly guessing solutions, on average, satisfied 455 out of 500 clauses. It is well known that randomly guessing solutions for a MAX-3SAT instance can satisfy approximately 7/8 of all clauses of the instance [24]. Thus, we expected random guessing to find assignments that satisfy approximately 438 of our 500-clause problems. The fact that in our case random guessing solved more than the expected 438 clauses can be attributed to the small size of each formula (500 clauses). Regardless of the used pruning method and regardless of the initial non-approximated QUBO transformation method (Chancellor or Nüßlein), we observe that removing values from the initial QUBO matrix leads to an immediate decline in the solution quality, i.e., in fewer clauses being solved. Additionally, we observe that the two QUBO formulations seem to behave differently when values are removed. In the case of Chancellor's transformation, there is a continuous decline in solution quality, even below the success rate of randomly guessing solutions. In the case of Nüßlein's transformation, we can see that for both, min pruning and random pruning, the decline in the quality of solutions (i.e., the number of clauses satisfied) stops at some point. Furthermore, we can also see for Nüßlein's transformation that using min pruning reduces the quality of the solutions faster compared to random pruning. As both pruning strategies ultimately lead to the same QUBO representation (i.e., Q_{100} contains only the initial main diagonal in any case), we can see that the final solution quality is identical. Although it would be interesting to identify the reason why in Chancellor's case pruning leads to a constant decline in solution quality while the decline in the solution quality of Nüßlein's pruned QUBO representations seems to stop at some point, as well as the difference of random and min pruning in Nüßlein's case, it is beyond the scope of this paper. We leave this analysis for future research. In this work, we are only interested in the observation that the solution quality immediately declines when values are removed. This shows that the results of the study conducted in [17], where 70% of the non-zero quadratic QUBO values could be removed without significant loss in solution quality, only holds for Choi's transformation and is not in general a good strategy for creating QUBO approximations. In contrast to the results of these naive approximation methods, we will demonstrate in the following section that our approach of systematically creating QUBO approximations can even increase the solution quality.

C. Evaluation of the Systematic QUBO Approximation method

In this section, we evaluate our proposed method of systematically creating QUBO approximations, as described in Sec. IV-B. We transformed all 3SAT instances to instances of QUBO using the QUBO approximation shown in Table V (which we call *FullApprox*), Chancellor's transformation, and Nüßlein's transformation. As embedding a QUBO onto D-Wave's QPU is a heuristic process, we generated 10 embeddings for each formula to reduce the influence of particular embeddings (for example different embeddings may require a different number of physical qubits). For each embedding, we then generated 100 samples. Thus, for each formula and each QUBO transformation, we generated 1000 samples, yielding 100,000 D-Wave samples per QUBO transformation. Furthermore, for each formula, we randomly guessed 1000 solutions to compare the D-Wave results against a random guessing baseline. For each formula and each method of solving the MAX-3SAT problem, we used the best of the 1000 received answers to compare with the results of the other methods. The distribution of the best results for each formula and each method is shown in Fig 2a. To determine the relative performance of the QUBO transformation approaches and the random guessing method, we compared the best solution of the FullApprox method with the best solution of all the other methods. That is, for each formula, we calculated the number of satisfied clauses by the best solution of the FullApprox method and subtracted the number of clauses satisfied by the best solution of any other method. Similarly, we compared the best solution for Chancellor's transformation (resp. Nüßlein's transformation) to the best solution for the random method for each formula. The results of this comparison are shown in

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

Fig 2b). A label *A,B* on the x-axis denotes that the respective boxplot shows the results of comparing method A against method B. That is, label *FA, C* denotes that the results of the FullApprox transformation are compared to Chancellor's transformation as described above.

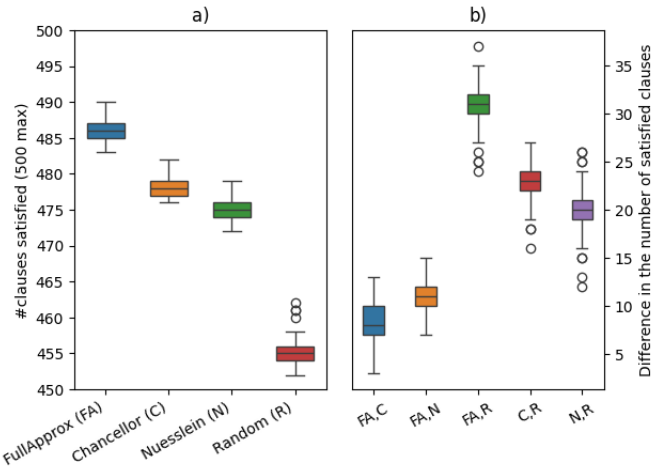


Fig. 2: Distribution of the best solutions found with different QUBO transformations (left). Difference of the best solutions found by one approach compared to the best solutions found by another approach (right).

From Fig. 2a we can see that no satisfying solution was found for either of the approaches since no answer of any approach yielded 500 satisfied clauses. However, as we are solving MAX-SAT, we are interested in the approach that satisfies the largest number of clauses. The best results, i.e., the most satisfied clauses, were obtained using our FullApprox method. The exact QUBO transformations by Chancellor and Nüßlein and the random method yielded fewer satisfied clauses for each formula. The direct comparison (Fig. 2b) shows that the best solution for all the formulas was found by the FullApprox method, as Chancellor's transformation yielded between 3 and 13 less satisfied clauses, Nüßlein's transformation between 7 and 15 less satisfied clauses, and the random baseline between 24 and 37 less satisfied clauses for each formula compared to the FullApprox method. As explained in Sec. 2, the results of random guessing are as expected, as it is possible to satisfy approximately 7/8 of all clauses by randomly guessing [24]. This emphasizes an important point about the interpretation of the results: comparing the absolute number of satisfied clauses of solutions generated by different methods does not yield a good indication of the performance of different methods. For example, a perfect solver will find a solution that satisfies all the clauses, while randomly guessing satisfies *only* 12.5% (=1/8) fewer clauses. We thus argue that the number of clauses satisfied by the random guessing method should be the baseline for each formula. Using the number of satisfied clauses found by the random guessing method for each formula as the baseline, detailed data analysis showed that the FullApprox method yields between 12% and 59% more satisfied clauses than Chancellor's method. A similar analysis showed that the FullApprox method yields between 28% and 125% more satisfied clauses than Nüßlein's method.

TABLE VI: Embedding sizes (i.e., number of physical qubits) on D-Wave's Advantage_System6.4 (5,614 available qubits)

QUBO Transformation	Min Size	Max Size	Avg Size
FullApprox	2,203	2,819	2,385
Chancellor	4,653	4,937	4,769
Nüßlein	4,640	4,911	4,763

As the FullApprox approach is an approximation, it needs a significantly smaller amount of physical qubits on a quantum computer than the exact QUBO transformations by Chancellor and Nüßlein, to solve the same MAX-3SAT problems. Table VI shows the embedding sizes (i.e., the number of physical qubits needed) on D-Wave's Quantum Annealer *Advantage_System6.4* for the formulas and QUBO transformations of the previously conducted evaluation.

It is notable that our QUBO approximation approach *FullApprox* only needed half of the number of physical qubits on D-Wave's Advantage_System6.4 than the other approaches, while at the same time yielding better results (as shown in Fig. 2). Note that the Advantage_System6.4 possesses 5,612 physical qubits. Considering the embedding sizes in Table VI, it is clear that when using exact approaches (like Chancellor's and Nüßlein's transformations), one cannot solve formulas with much more than 500 clauses on this machine. However, using the approximation approach, we have been able to embed formulas with 750 clauses on the Advantage_System6.4, which used approximately 4,500 qubits. Thus, by using our systematic QUBO approximation method, one can embed 50% larger formulas on the quantum annealer compared to non-approximated $(n + m) \times (n + m)$-dimensional QUBO transformations (like Chancellor's and Nüßlein's method).

D. Scaling of the Systematic QUBO Approximation Method

As seen in the previous section, our systematic QUBO approximation method can yield even better results than exact QUBO transformations. In this section, we want to address the scaling behavior of this approach. We will show that this approach does not only work when the formulas are small but can still yield good solutions as the size of the MAX-3SAT problems grows.

For this section, we used the Balanced SAT [23] method again to create 100 3SAT formulas each consisting of 10,000 clauses and 2,780 variables, which amounts to a ratio of clauses to variables of approximately 3.6. This is the empirically derived clauses-to-variable ratio for hard-to-solve 3SAT instances of the Balanced SAT method. As hard formulas of this size can take multiple days (or even longer) to get solved, we do not know whether these formulas are satisfiable (i.e., the maximum number of satisfiable clauses is 10,000) or not (i.e., the maximum number of satisfiable clauses is smaller than 10,000). As we are only interested in the relative results between different approaches, this is not a limitation.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

We transformed each of these 100 formulas to instances of QUBO using the previously described FullApprox method and Chancellor's and Nüßlein's transformation. Each QUBO will be solved on a classical (i.e., non-quantum) computer 100 times using D-Wave's tabu search implementation. Thus, for each formula and each QUBO transformation method, we have generated 100 samples.

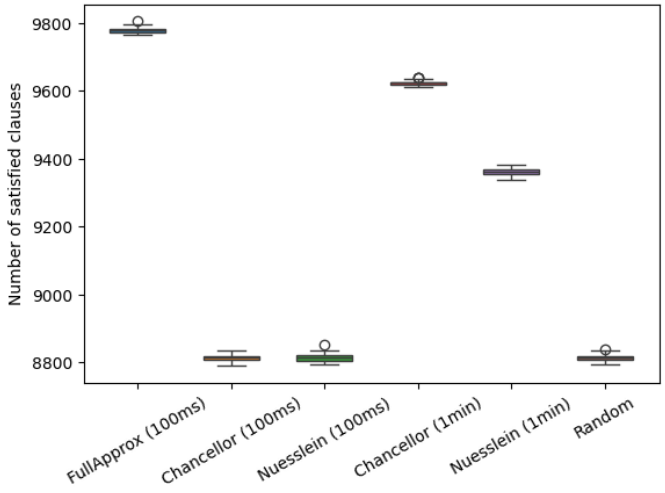


Fig. 3: Distribution of the best solutions found for each combination of a QUBO transformation and tabu solver time limit.

The results shown in Fig. 3 reveal two interesting insights. Firstly, we observe that randomly guessing solutions leads to approximately 8,800 satisfied clauses, which is close to the expected value of $7/8 \times 10,000$ clauses. Even though the size of the created formulas is 20 times larger than the size of the formulas created in Sec. II-C, our approximation method (FullApprox) still yields good solutions. For each formula, our FullApprox approach found solutions that satisfy approximately 98% of all clauses. Secondly, to put these results into perspective, we compare the results of the FullApprox method to the results of Chancellor's and Nüßlein's method, also shown in Fig. 3. We can see that for our dataset, the FullApprox method still yielded the best results.

Observe, that for the FullApprox method, the tabu search method was given 100 ms of computation time. In contrast, the tabu search method was allowed 1 minute of computation time for the non-approximated exact QUBO transformations by Chancellor and Nüßlein. This is a consequence of D-Wave's implementation of the tabu search method. As the QUBOs of the non-approximated, exact QUBO transformations by Chancellor and Nüßlein are approximately 4.6 times as large as the QUBOs generated by our approximation method FullApprox, D-Wave's specific implementation of the tabu search method needs more time to generate reasonable results. We thus allowed the tabu search method 600 times more time to solve the 4.6 times larger QUBO matrices resulting from Chancellor's and Nüßlein's transformation. We want to emphasize that we included the specific time parameters for reproducibility in this paper. This is not meant as a performance comparison. By providing a different implementation of the tabu search method, one could reduce the solving times for larger QUBOs. The sole intention of this section, and especially the comparison of our approximated method with the non-approximated methods, is to show that our QUBO approximation for MAX-3SAT problems still yields comparably good results, even if the problem size increases.

VI. Conclusion

Due to the missing error correction of contemporary quantum hardware, any quantum computer calculation can be considered an involuntary solution approximation. In this work, we thus studied whether it is possible to improve the solution quality when solving problems on D-Wave's quantum annealer Advantage_System6.4 by using systematic QUBO approximation of MAX-3SAT problems instead of exact, non-approximated QUBO representations as an input for the quantum annealer. For a MAX-3SAT instance consisting of a 3SAT formula with n variables and m clauses, our proposed QUBO approximation method yields $(n \times n)$-dimensional QUBO matrices, which is considerably smaller than the QUBO matrices that result from any exact, non-approximated QUBO transformation. The method is based on an adaption of the creation method of exact QUBO transformations that result in QUBO matrices of dimension $(n + m) \times (n + m)$. In an empirical evaluation, we demonstrated that our QUBO approximations can yield comparable or even better results than exact, non-approximated QUBO transformations when solved on D-Wave's quantum annealer Advantage_System6.4. Furthermore, 50% larger MAX-3SAT instances can be solved on the quantum annealer due to our approximation method's reduced need for physical qubits. Additionally, we empirically showed that naive methods of creating QUBO approximations for MAX-3SAT problems using $(n+m)\times(n+m)$-dimensional QUBO matrices as initial QUBOs are not effective.

In the future, we would like to explore the cause for the different behavior of the naive QUO approximation methods *min pruning* and *random pruning*. Furthermore, it is also interesting to investigate whether systematic QUBO approximation can be beneficial for solving other classes of hard problems on D-Wave's quantum annealer.

Acknowledgement

This paper was partially funded by the German Federal Ministry of Education and Research through the funding program "quantum technologies — from basic research to market" (contract number: 13N16196).

References

[1] J. P. Marques-Silva and K. A. Sakallah, "Boolean satisfiability in electronic design automation," in *Proceedings of the 37th Annual Design Automation Conference*, 2000, pp. 675–680.

[2] J. Rintanen, K. Heljanko, and I. Niemelä, "Planning as satisfiability: parallel plans and algorithms for plan search," *Artificial Intelligence*, vol. 170, no. 12-13, pp. 1031–1080, 2006.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.

[3] Q. Duan, S. Al-Haj, and E. Al-Shaer, "Provable configuration planning for wireless sensor networks," in *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*. IEEE, 2012, pp. 316–321.

[4] P. Abate, R. Di Cosmo, G. Gousios, and S. Zacchiroli, "Dependency solving is still hard, but we are getting better at it," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2020, pp. 547–551.

[5] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[7] J. Nüßlein, S. Zielinski, T. Gabor, C. Linnhoff-Popien, and S. Feld, "Solving (max) 3-sat via quadratic unconstrained binary optimization," *arXiv preprint arXiv:2302.03536*, 2023.

[8] N. Chancellor, S. Zohren, P. A. Warburton, S. C. Benjamin, and S. Roberts, "A direct mapping of max k-SAT and high order parity checks to a chimera graph," *Scientific reports*, vol. 6, no. 1, p. 37107, 2016.

[9] V. Choi, "Adiabatic quantum algorithms for the np-complete maximum-weight independent set, exact cover and 3sat problems," *arXiv preprint arXiv:1004.2226*, 2010.

[10] Z. Bian, F. Chudak, W. Macready, A. Roy, R. Sebastiani, and S. Varotti, "Solving sat (and maxsat) with a quantum annealer: Foundations, encodings, and preliminary results," *Information and computation*, vol. 275, p. 104609, 2020.

[11] S. Zielinski, J. Nüßlein, J. Stein, T. Gabor, C. Linnhoff-Popien, and S. Feld, "Pattern qubos: Algorithmic construction of 3sat-to-qubo transformations," *Electronics*, vol. 12, no. 16, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/16/3492

[12] A. Verma, M. Lewis, and G. Kochenberger, "Efficient qubo transformation for higher degree pseudo boolean functions," *arXiv preprint arXiv:2107.11695*, 2021.

[13] S. Zielinski, J. Nüßlein, J. Stein, T. Gabor, C. Linnhoff-Popien, and S. Feld, "Influence of different 3sat-to-qubo transformations on the solution quality of quantum annealing: A benchmark study," *arXiv preprint arXiv:2305.00720*, 2023.

[14] T. Krüger and W. Mauerer, "Quantum annealing-based software components: An experimental case study with sat solving," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 2020, pp. 445–450.

[15] J. Nüßlein, C. Roch, T. Gabor, J. Stein, C. Linnhoff-Popien, and S. Feld, "Black box optimization using qubo and the cross entropy method," in *International Conference on Computational Science*. Springer, 2023, pp. 48–55.

[16] T. Matsumori, M. Taki, and T. Kadowaki, "Application of qubo solver using black-box optimization to structural design for resonance avoidance," *Scientific Reports*, vol. 12, no. 1, p. 12143, 2022.

[17] I. Sax, S. Feld, S. Zielinski, T. Gabor, C. Linnhoff-Popien, and W. Mauerer, "Approximate approximation on a quantum annealer," in *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 2020, pp. 108–117.

[18] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambrtidge University Press, 2009.

[19] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the third annual ACM symposium on Theory of computing*, 1971, pp. 151–158.

[20] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," *arXiv preprint arXiv:1811.11538*, 2018.

[21] M. Anthony, E. Boros, Y. Crama, and A. Gruber, "Quadratic reformulations of nonlinear binary optimization problems," *Mathematical Programming*, vol. 162, pp. 115–144, 2017.

[22] S. Rendle, "Factorization machines," in *2010 IEEE International conference on data mining*. IEEE, 2010, pp. 995–1000.

[23] I. Spence, "Balanced random sat benchmarks," *SAT COMPETITION 2017*, p. 53, 2017.

[24] J. Håstad, "Some optimal inapproximability results," *Journal of the ACM (JACM)*, vol. 48, no. 4, pp. 798–859, 2001.

Authorized licensed use limited to: TU Delft Library. Downloaded on February 26,2025 at 11:07:49 UTC from IEEE Xplore. Restrictions apply.