Constraint Solving with Quantum Annealing

Tristan Nuck ✉
Chair of Programming Languages and Compiler Construction, Brandenburg University of Technology Cottbus-Senftenberg, Germany

Sven Löffler ✉
Chair of Programming Languages and Compiler Construction, Brandenburg University of Technology Cottbus-Senftenberg, Germany

Petra Hofstedt ✉
Chair of Programming Languages and Compiler Construction, Brandenburg University of Technology Cottbus-Senftenberg, Germany

Abstract

Many real-world problems, like workforce scheduling, routing, and other logistic problems, can be described and solved using constraint programming. Because most of these problems are NP-hard, an exact or optimal solution for large problem instances can only be found using high computational effort, even with the methods developed to date. With the current advancements in quantum computing hardware, the hope rises that these devices can solve such combinatorial problems faster or help accelerate the solution-finding process in cooperation with classical hardware.

This paper gives an overview of our current work-in-progress research on solving constraint problems using quantum hardware and an outlook on future planned topics that are going to be tackled. Solving combinatorial problems of any kind on a quantum annealer requires the problem to be modeled as a quadratic unconstrained binary optimisation problem (QUBO). For this, different novel techniques for QUBO creation from truth tables of single constraints are created and evaluated, which lay the groundwork for future research in that direction. Here, an extension utilizing already embedded QUBO terms is planned.

In addition, different modelling techniques that focus on optimizing quantum annealing-specific metrics were developed and evaluated on the rotating rostering problem (RRP). These two topics lay the foundation for modelling and optimization of more complicated problems like the vehicle routing problem (VRP) with its many extensions, in addition to alternative approaches to constraint solving using hybrid algorithms utilising classical CP solvers in addition to quantum annealing approaches.

2012 ACM Subject Classification Applied computing → Operations research

Keywords and phrases Quantum, Search, Modelling, Operations Research

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

This section gives an overview of quantum annealing and the relevant terms and ideas used to solve combinatorial satisfiability and optimisation problems. Afterwards, a small example is given to show the relevance of the presented energy gap for quantum annealing.

Quantum Annealing

Quantum annealers are a special kind of quantum computing device that uses the quantum adiabatic theorem [3] to compute the optimal solution of a given combinatorial problem. It consists, like any other quantum computing device, of qubits that can be manipulated individually by local biases or in pairs using couplers. Such interactions can be described by an Ising model [4] which has the following form:

$$H(s) = \sum_{i=1}^{n} h_i \cdot s_i \sum_{i,j=1,i<j}^{n} J_{i,j} \cdot s_i \cdot s_j \tag{1}$$

For each of the n spin variables $s_i \in \{-1, +1\}$, we get linear coefficients h_i, and for each pair of two variables, we get quadratic coefficients $J_{i,j}$. In reality, connecting every pair of qubits is difficult, and because of that, the connectivity of quantum annealing devices is limited to 15 to 20 connections for each qubit. Current devices feature up to 5000 qubits, which result in relatively sparse connected graphs. Additionally, the value range of the linear and quadratic coefficients is, because of technical limitations, limited to $h_i \in [-6, +6]$ and $J_{i,j} \in [-2, +1]$. These two factors, connectivity and limited domains, increase the difficulty of solving constraint problems on this type of device.

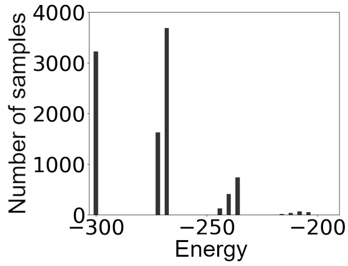
Most hard combinatorial problems require dense connections between the variables, which is the origin of their exponential complexity. So the corresponding Ising model for the problem features more quadratic terms than physical couplers. To solve this issue, an embedding has to be found. Here, multiple physical qubits can be connected in a chain to simulate one logical variable, which has a higher degree of connectivity than a single physical qubit. This is done in a way that for each variable, we achieve the desired connections between logical variables as described in the model. Finding such an Embedding is again an NP-hard problem. To accelerate the process, heuristic approaches [5] are used. These, of course, are not able to find the best possible embedding or sometimes an embedding at all, which additionally increases the difficulty of problem-solving. Additionally, using auxiliary qubits on already size-limited hardware is itself a problem, but doing so also introduces additional opportunities for noise, which can lower the quality of the samples. Meaning that minimising the number of additional qubits used is an important optimisation goal.

To solve any kind of constraint problem, we first have to convert it into an Ising model representation, where the minimal energy state of the Ising model corresponds to the solutions of our given constraint problem. After finding an embedding for this model, the annealer can determine a minimum and, after reading out the qubits, return a solution to the problem. Finding such an Ising model representation is a problem in itself. While modelling, it is often easier to first represent the problem as a quadratic unconstrained binary optimisation problem (QUBO), which has the following form:

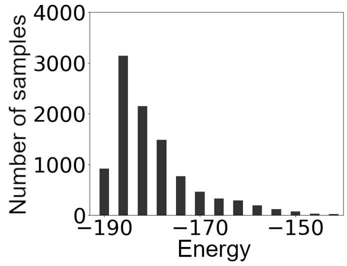
$$Q(x) = \sum_{i,j=1,i\leq j}^{n} Q_{i,j} \cdot x_i \cdot x_j \tag{2}$$

This model is equivalent to the Ising model and can, with a simple variable substitution $s_i = 2 \cdot x_i - 1$, be converted into an Ising model. Some currently standard modelling techniques are describing the problem of finding QUBO expressions as a mixed integer linear program (MILP) using only binary variables. Then, converting the inequalities to equality using slack variables, solving for zero, and then squaring the whole expression, which results in QUBO terms that are zero if and only if the original condition is fulfilled. Adding up all these generated QUBO terms results in the final QUBO model. Another approach is to directly find fitting QUBO expressions for given constraints and add them up. This can often reduce the number of auxiliary variables and quadratic terms needed [10, 1].

Because of the underlying principle of the quantum adiabatic theorem, the model size is not the only relevant feature to take into consideration when optimising a model. To solve the problem via quantum annealing, we first start in a configuration that we know the



(a) Large energy gap



(b) Small energy gap

Figure 1 (a) Anneal results for a large energy gap. (b) Anneal results for a small energy gap.

ground state of and can easily prepare. Afterwards, we use the linear biases and couplers to slowly shift our configuration to our Ising model description. If we do this transition slowly enough, the annealer will stay in the ground state (lowest energy state), and a solution can be read out. The time needed for a successful annealing process depends on the energy gap between the lowest and second-lowest energy state. A larger gap means that less time is required according to the quantum adiabatic theorem. It's important to note that, because current-day quantum hardware is noisy, even with theoretically enough time, the annealer can still land in other undesirable states. Also, increasing the annealing time increases the time the hardware is exposed to noise and therefore also reduces the chance of a successful annealing.

Impact of energy gap

To demonstrate that the impact of the energy gap in quantum annealing is not purely of theoretical interest, an annealing result taken from the master's thesis [14] is shown in Figure 1. The figure shows annealing results for a small graph coloring problem with six vertices and four colors. While in both test runs (a) and (b) the variables were encoded as binary variables using two spin variables $s_{0,i}$ and $s_{1,j}$, the constraint for forbidding two adjacent nodes to have the same color was realised differently.

The x-axis of the plots shows the energy of the found sample, while the y-axis shows the number of times this sample was measured. For the first figure, we were using an Ising expression with a large energy gap, which can be seen by the large gap between the minimum energy state and the second-highest energy state. In contrast, the expression used in the second figure has a smaller energy gap. This also results in drastically different annealing results. Using the expression with the larger energy gap results in roughly 30% of the 10,000 samples reaching the optimum, while using the expression with the smaller energy gap resulted in less than 10% of the samples reaching the optimum. This already shows that the energy gap can have a real impact on annealing performance and is therefore worth optimising for.

2 Related Work

There exist different techniques for converting constraint satisfaction problems into QUBO or Ising models. D-Wave has compiled a catalogue of different techniques [8] from simple transformation of linear equations and inequalities, using already known penalty terms, transforming higher order interactions to quadratic ones with auxiliary variables, or using conflict-graph techniques to convert the problem directly into a QUBO representation. All these different techniques work and are quite fast because they follow a strict transformation rule. The problem with this kind of technique is that they don't give any guarantees for the quality of the generated model regarding the energy gap, number of auxiliary variables, and number of quadratic terms. Meaning models generated this way can perform poorly using a quantum annealer, even for already small problem instances. We try to tackle this problem and focus on transformations that try to guarantee good-quality QUBO models.

The creation of optimised QUBO or Ising expressions is something not extensively researched in the literature. Nonetheless, there exists some work on tackling the task of QUBO generation for a given truth table. The heuristic introduced in the paper [16] shows a greedy and fast strategy for finding QUBO expressions that locally maximize the energy gap given a truth table. In contrast, we focus on finding global optimal solutions and optimising additional properties like the number of quadratic terms and coefficient domains.

Using the earlier described method of MILP modelling and afterwards converting to a QUBO expression is currently the most used technique [1, 10, 18]. But there already exist different approaches like unbalanced penalisation [13] which require fewer slack variables but looses exactness of the generated QUBO expression. In comparison, our methods focus on exact models and additional optimization regarding quadratic terms and coefficient domains.

Modelling of different classical constraint problems like the traveling salesman problem, job shop scheduling, or nursing scheduling was already done in the literature [10, 18, 12]. The starting point was nearly always MILP models that were converted to QUBO expressions afterwards. Specialized modelling strategies for such problems were not used. That is something that we are currently working on for the rotating rostering problem (RRP) [11]. But there are, of course, publications on different techniques for improving QUBO modelling. There exists a domain wall encoding [6] that is specialized for encoding finite domain variables into binary variables on a quantum annealer. These encodings will also be used for our problem model descriptions.

Another research direction is the use of classical and quantum computation together in one algorithm. These so-called hybrid quantum computing algorithms could show promising results much quicker than purely quantum approaches. Here we have two obvious ways of combining these two methods. First, using a classical computer as the main solver for the problem and a quantum computer as a fast heuristic sampling device, given hints and bounds to the classical algorithm. D-Wave is already working on this with their own hybrid solver versions [15], but these are currently black boxes and only a little is known about the concrete inner workings. Because of this, a transparent hybrid solver would be interesting.

Another possible option for hybrid computing would be problem decomposition. Here, a given problem can be classically decomposed into subproblems that fit on the quantum annealer. The results can then again be recombined using a classical computer. Of course, this requires good algorithms and heuristics for the best points of decomposition and how to recombine problems, even if some variable assignments don't align. An overview of current existing methods can be found here [2]. Even if there are already many different techniques used, there is still room for improvement, especially on problem-specific decomposition techniques.

3 Annealing Optimised Problem Modelling

In this section, we present two work-in-progress research topics regarding creating an annealing optimised QUBO model for constraint problems. The first part describes the QUBO finder for generating penalty terms using standard boolean truth tables, and the second part describes the rotating rostering and some of the optimisation techniques used to improve on a naive model.

QUBO Finder

To solve any kind of constraint problem on a quantum annealing device, we first have to create a suitable QUBO model of the problem. Here, our work-in-progress QUBO Finder tool comes into play. Given an arbitrary truth table of appropriate size, the QUBO Finder is able to create optimised QUBO expressions that reach a minimum for satisfying variable assignments and larger values than the minimum otherwise. This tool can be used to fully model suitable constraint problems by converting the given constraints into QUBO penalty terms. Summing up all these terms results in a valid model for the given problem.

To achieve this goal, we are currently testing three different methods for QUBO generation: constraint programming (CP), mixed integer nonlinear programming (MINLP), and a heuristic search approach. Because the coefficients domains are real values, the domains of the CP variables were scaled to emulate multiple decimal places. The CP and MINLP models work in a typical declarative programming way. We describe the necessary constraints to form a valid QUBO expression via equations and inequalities. Afterwards, the solvers (for CP OR-Tools [17] and for MINLP SCIPOpt [9]) are used to find a valid solution. Because it is not clear whether a truth table requires auxiliary variables or not, we first run the models with 0 auxiliary variables and iteratively increase the number by one if no solution is found. This also guarantees that we use the minimum number of auxiliary variables required.

Because these approaches can be quite slow for larger truth tables and multiple required auxiliary variables, we also tried to optimise the model and solver settings. Some used techniques were symmetry breaking for the auxiliary variables, because they can be treated as identical. Adding a variable heuristic for the solver, inspired by the heuristics used in [16]. Also testing different model representations of the problem using the QUBO and Ising formulation. Because these approaches were quite slow for larger problems, another method using a search-focused approach was developed in addition to the other methods.

For the search approach, we take advantage of the fact that finding QUBO expressions without auxiliary variables is an easy LP problem. Only after auxiliary variables are introduced, the problem gets transformed into a hard MINLP problem. Here, in addition to the QUBO coefficients, we also have to find suitable assignments for the auxiliary variables. By iteratively trying different auxiliary assignments for each possible original variable assignment, we iteratively build a LP model that, if all assignments of the truth table are incorporated and a solution can be found, results in a valid QUBO expression. Because we use an LP solver, we are more limited in terms of the objective function and have to work with approximations.

Each method is able to find QUBO expressions that for one maximise the minimum energy gap and also use as few quadratic terms as possible, beating current other implementations like the earlier mentioned heuristic [16] or the D-Wave penalty model [7]. In addition, the resulting expressions are already scaled to fit the maximum domain range of the biases and couplers. Which, especially for QUBO terms, is not trivially done, because these terms have to be converted to Ising terms and then fit into the domain range.

Rotating Rostering Modelling

Generating QUBO expressions from truth tables resulting from the given problem description is only one part of the modelling process of constraint problems for quantum annealing. The variable encoding, choice of constraints, and different model optimisations can drastically reduce the model size and enable previously not embeddable problems to be embedded on a quantum annealer. This is especially relevant, given the current limited hardware size and scalability problems for current-day quantum technologies. Some techniques for optimised problem modelling are shown as an example for the rotating rostering problem [11].

In this problem, n employees have to be assigned shifts in a rotating manner. Meaning employee one uses in the first week his first week schedule, in the second week the schedule of employee two, and so on until it loops around again after n weeks. These concrete problem used four different shifts: early, late, night, and day off, which is the typical shift distribution in most companies. In addition to the rotating nature, there are many constraints for the schedule given by either the company itself or work regulation guidelines. For example, does each day of the week require a certain number of employees in each shift, or each employee work the same shift on each day of the weekend. There are also a minimum and maximum number of times a shift type can repeat itself before a given employee has to be assigned a new shift type. The last constraint ensures that each employee has at least two days off in the span of 14 days.

The most common way in the literature to model such constraint problems is to first create an MILP model using only binary variables and afterwards convert the inequalities and equalitys into QUBO penalty terms, as described in Section 1. This approach often introduces many additional slack variables and quadratic terms, which increases the model size that needs to be embedded on the quantum annealer. Here, a lot of room is left open for improvements to the model. Of course, the encoding of the variables into binary variables is the first step and gives a direction for the following modelling. We used the one hot encoding meaning for each employee i and each weekday j we introduced four new binary variables $x_{i,j,1}, x_{i,j,2}, x_{i,j,3}, x_{i,j,4}$ where always exactly one variable is true if and only if the employee was assigned the corresponding shift on this day (1 = early, 2 = late, 3 = night, 4 = day-off).

4 Evaluation

In this section, we present some preliminary results regarding the QUBO Finder and the optimized modelling of the rotating rostering problem. Some results of the QUBO Finder can be seen in Table 1. Here three different examples namely, an allDifferent constraint over the following variables and domains $v_1 \in \{1, 2, 3\}$, $v_2 \in \{2, 3, 6\}$, $v_3 \in \{1, 4, 5\}$, $v_4 \in \{1, 5\}$, $v_5 \in \{4, 6\}$ where each variable is one hot encoded, a boolean expression $((x_1 \oplus x_2) \vee (x_3 \wedge x_4) \Rightarrow x_5 \wedge x_1) \wedge x_6$, and the child game Fizz Buzz were the first four binary variables encode the number $n \in \{0, 1, \dots, 15\}$ and the fifth and sixth variable are 1, if the number n is dividle by 3 (Fizz) or dividble by 5 (Buzz) respectivly. For each problem, the required number of auxiliary variables (Auxil.), the energy gap (Gap), and the number of quadratic terms used (Quad.) are shown. The first three columns are our developed methods consisting of constraint programming (CP), mixed integer nonlinear programming (MINLP), and the search approach (Search). The other two methods are taken from the literature and consist of the greedy heuristic approach (Heur) and the D-Wave penalty model (Pen). Because of technical limitations given by the model from D-Wave to only support up to 8 variables, there was no solution found for the all different constraint.

Because we iteratively try every number of auxiliary variables until infeasibility is confirmed, our approaches always use the minimum number of required variables. In contrast, the greedy heuristic approach can sometimes find no solution for a given number of auxiliary variables even if one exists. Additionally, we try to minimise the number of required quadratic

		CP	MINLP	Search	Heur	Pen
allDifferent	Auxil.	0	0	0	0	-
	Gap	**2**	**2**	**2**	**2**	-
	Quad.	**18**	**18**	19	62	-
	Time	$12s$	$30s$	$1.1s$	**0.3s**	-
Fizz Buzz	Auxil.	**2**	**2**	**2**	**2**	**2**
	Gap	0.516	0.571	**0.84**	0.09	0.43
	Quad.	25.4	**20**	23	28	28
	Time	$200s$	$10s$	$7.86s$	**0.43s**	$0.6s$
Boolean exp.	Auxil.	**1**	**1**	**1**	2	**1**
	Gap	**4**	**4**	**4**	0.6	**4**
	Quad.	**8**	**8**	**8**	24	11
	Time	$1s$	$0.9s$	0.07	$0.19s$	**0.06s**

Table 1 Results of generating optimised QUBO expressions for three example problems using our three developed methods and two methods from the literature.

terms, which the other two methods don't take into consideration. This results in a higher number of quadratic terms. For example, the greedy heuristic uses 62 quadratic terms in contrast to the only 18-19 required with our method. These results look similar for the other problems. For the energy gap, we looked at the scaled QUBO terms that cover the maximum domain range, here our approaches outperformed the other two currently existing methods for the FizzBuzz and Boolean expression example. The only metric where our methods fall behind is the time required to find the solution. But because both methods from the literature are greedy and don't try to find the optimum, this is to be expected.

Some preliminary results for the eight employees rotating rostering problem can be seen in Table 2. The different rows show the number of required variables (Variables), the number of required quadratic terms (Quadratic terms), and the maximum degree (Max. degree), meaning the maximum number of quadratic terms a single variable is involved in. Each column shows a different modelling approach, where the reference naive modelling is shown in the first column (Naive) and each following column displays different novel optimisations to the original model. For the shift repetition constraints (Shift rep.), we were able to reduce the number of variables by 224 and the number of quadratic terms by 1344. The following two optimisations (Rest days and Shift requ.) reduce the number of variables at least a bit, but especially the rest days constraint optimisation drastically reduces the number of quadratic terms and the maximum variable degree. This result should be taken with a grain of salt, because our current optimisation slightly changes the model solution space, for which there are already plans for another optimisation solving this problem. The second last optimization (Day off cons.), which optimised constraints involving the day-off shift variables, improved the model the furthest. The last column combines each of the previous optimisation steps and results therfore in the smallest model, which only requires half of the variables, roughly one third of the quadratic terms, and reduces the maximum degree by over 75%.

5 Conclusion and Future Work

In this paper, we presented two of our current work-in-progress research topics, namely the QUBO Finder with the goal of generating optimised QUBO expressions for quantum

	Naive	Shift rep.	Rest days	Shift requ.	Day off cons.	Combined
Variables	896	672	840	896	656	488
Quadratic terms	7896	6552	5432	7756	3984	2780
Max. degree	99	94	52	94	30	23

Table 2 Results of the optimised QUBO modelling for the rotating rostering problem, displaying the number of variables, quadratic terms, and the maximum degree of variables for each different optimisation technique.

annealing from truth tables, and the topic of optimised modelling of constraint problems on the example of the rotating rostering problem. The preliminary results for the QUBO Finder are already promising, outperforming current tools in quality but struggling with computation time. Here, especially, the topic of scalability is going to be a future topic. The current global approaches work well for a small number of variables and smaller truth tables, but take too much time for larger problem sizes. Here, more locally oriented techniques are required to solve the problem in a moderate amount of time. Additionally, some plans regarding an expansion of the QUBO Finder are already planned. Because even with optimised QUBO expressions, the problem of embedding such a model is still NP-hard, we try to lower the difficulty of this problem by generating already optimal or nearly optimal embedded penalty terms. These then only have to be placed on the annealer, and penalty terms using the same problem variable have to be connected via qubit chains. The idea here is similar to the placement of computing units on chips that then have to be connected via wires.

Looking at the preliminary results of the optimised QUBO model for the rotating rostering, we can also already see promising results for future research direction. Improving on the current standard modelling of using MILP models as a base can drastically reduce the model size. Here, extensive tests with other variable encodings and also other problems are planned. One economically relevant and interesting problem to look at, for example, would be the vehicle routing problem, which has many possible variations and extensions. Additionally, other modelling techniques are also planned for future investigation. Current-day constraint modelling can profit from the use of global constraints. A possible adaptation of this would be to develop an optimised QUBO transformation from commonly used constraints, which, for one, would facilitate QUBO model creation and improve model performance and size.

Currently, both the QUBO expression optimisation using the QUBO Finder and the optimised modelling for the rotating rostering problem are evaluated purely theoretically. It is planned to perform practical tests on real quantum devices and evaluate metrics like time to target and success probability to analyse the impact of the different optimization techniques developed properly.

The last topic regarding future work are hybrid approaches. Because current-day computer hardware has had multiple decades more time to develop, current-day quantum hardware is lacking behind in comparison. But to already use the potential and new computational methods that quantum computing can deliver, we can think about hybrid approaches that combine, for example, classical CP approaches with the more heuristic and sample-based nature of quantum annealing. Here, a classical computer solving the full problem via constraint programming and using the annealer as a support tool to get early and quick lower bounds for certain sub-problems or problem variations would be imaginable. Of course, other hybrid approaches could include the classical splitting of problems into smaller subproblems that fit onto current quanntum annealers. These sub-problems then have to be recombined in a reasonable manner again via a classical computer.

References

1 Luis Armas, Stefan Creemers, and Samuel Deleplanque. Solving the resource constrained project scheduling problem with quantum annealing. *Scientific Reports*, 14, 07 2024. `doi:10.1038/s41598-024-67168-6`.

2 Gideon Bass, Max Henderson, Joshua Heath, and Joseph Dulny III. Optimizing the optimizer: Decomposition techniques for quantum annealing, 2020. URL: `https://arxiv.org/abs/2001.06079`, `arXiv:2001.06079`.

3 M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift für Physik*, 51:165–180, mar 1928. `doi:10.1007/BF01343193`.

4 STEPHEN G. BRUSH. History of the lenz-ising model. *Rev. Mod. Phys.*, 39:883–893, Oct 1967. URL: `https://link.aps.org/doi/10.1103/RevModPhys.39.883`, `doi:10.1103/RevModPhys.39.883`.

5 Jun Cai, William G. Macready, and Aidan Roy. A practical heuristic for finding graph minors, 2014. URL: `https://arxiv.org/abs/1406.2741`, `arXiv:1406.2741`.

6 Nicholas Chancellor. Domain wall encoding of discrete variables for quantum annealing and qaoa. *Quantum Science and Technology*, 4(4):045004, August 2019. URL: `http://dx.doi.org/10.1088/2058-9565/ab33c2`, `doi:10.1088/2058-9565/ab33c2`.

7 D-Wave. D-wave penalty model. `https://github.com/dwavesystems/penaltymodel`. last visited 2026-05-11.

8 D-Wave. Reformulating a problem. `https://docs.dwavequantum.com/en/latest/quantum_research/reformulating.html#qpu-reformulating`. last visited 2026-05-11.

9 Christopher Hojny, Mathieu Besançon, Ksenia Bestuzheva, Sander Borst, Antonia Chmiela, João Dionísio, Leon Eifler, Mohammed Ghannam, Ambros Gleixner, Adrian Göß, Alexander Hoen, Rolf van der Hulst, Dominik Kamp, Thorsten Koch, Kevin Kofler, Jurgen Lentz, Stephen J. Maher, Gioni Mexi, Erik Mühmer, Marc E. Pfetsch, Sebastian Pokutta, Felipe Serrano, Yuji Shinano, Mark Turner, Stefan Vigerske, Matthias Walter, Dieter Weninger, and Liding Xu. The SCIP Optimization Suite 10.0. Technical report, Optimization Online, November 2025. URL: `https://optimization-online.org/2025/11/the-scip-optimization-suite-10-0/`.

10 Kazuki Ikeda, Yuma Nakamura, and Travis Humble. Application of quantum annealing to nurse scheduling problem. *Scientific Reports*, 9, 09 2019. `doi:10.1038/s41598-019-49172-3`.

11 Sven Löffler, Ilja Becker, and Petra Hofstedt. CSPLib problem 087: Rotating rostering problem. `http://www.csplib.org/Problems/prob087`. last visited 2026-05-11.

12 Roman Martoňák, Giuseppe E. Santoro, and Erio Tosatti. Quantum annealing of the traveling-salesman problem. *Physical Review E*, 70(5), November 2004. URL: `http://dx.doi.org/10.1103/PhysRevE.70.057701`, `doi:10.1103/physreve.70.057701`.

13 J A Montañez-Barrera, Dennis Willsch, A Maldonado-Romo, and Kristel Michielsen. Unbalanced penalization: a new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms. *Quantum Science and Technology*, 9(2):025022, apr 2024. `doi:10.1088/2058-9565/ad35e4`.

14 Tristan Nuck. Modellierung und untersuchung von constraint-problemen für quanten-annealer. `https://github.com/Nidoll/QuantumAnnealing/blob/develop/Masterarbeit.pdf`. last visited 2026-06-08.

15 Eneko Osaba and Pablo Miranda-Rodriguez. D-wave's nonlinear-program hybrid solver: Description and performance analysis, 2024. URL: `https://arxiv.org/abs/2410.07980`.

16 Scott Pakin. A simple heuristic for expressing a truth table as a quadratic pseudo-boolean function. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 218–224, 2021. `doi:10.1109/QCE52317.2021.00039`.

17 Laurent Perron and Frédéric Didier. Cp-sat. URL: `https://developers.google.com/optimization/cp/cp_solver/`.

18 Davide Venturelli, Dominic J. J. Marchand, and Galo Rojo. Quantum annealing implementation of job-shop scheduling, 2016. URL: `https://arxiv.org/abs/1506.08479`, `arXiv:1506.08479`.