Solving the Resource-Constrained Project Scheduling Problem (RCPSP) with Quantum Annealing

Luis Fernando Pérez Armas, Stefan Creemers, Samuel Deleplanque

Solving the Resource-Constrained Project Scheduling Problem (RCPSP) with Quantum Annealing

Luis Fernando PÉREZ ARMAS[1*†], Stefan CREEMERS[1,3†] and Samuel DELEPLANQUE[2†]

[1*]Operations Management, IESEG School of Management, Univ. Lille, CNRS, UMR 9221 - LEM - Lille Economie Management, 3 rue de la digue, Lille, F-59800, Nord, France.
[2]Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520 - IEMN, 41 Bd Vauban, Lille, 59000, France.
[3]ORSTAT KU Leuven, Naamsestraat 69, Leuven, 3000, Belgium.

*Corresponding author(s). E-mail(s): l.perezarmas@ieseg.fr;
Contributing authors: s.creemers@ieseg.fr; stefan.creemers@kuleuven.be; samuel.deleplanque@junia.com;
†These authors contributed equally to this work.

Abstract

Quantum annealing emerges as a viable solution for solving complex problems such as the resource-constrained project scheduling problem (RCPSP). We analyze 12 Mixed Integer Linear Programming (MILP) formulations for solving the RCPSP, and convert the most qubit-efficient formulation into a Quadratic Unconstrained Binary Optimization (QUBO) model. We solve this QUBO model using the D-Wave Advantage 6.3 Quantum Annealing machine and compare its performance with that of classical computer solvers. This pioneering effort marks the first use of quantum annealing for RCPSP, showing promising results, especially for smaller to medium-sized instances.

Keywords: Resource Constrained Project Scheduling Problem, Quantum Optimization, Quantum Annealing

Introduction

In the realm of computational methodologies, quantum computing has emerged as a groundbreaking approach, promising revolutionary solutions to intricate optimization problems. At the forefront of this second quantum revolution is the paradigm of quantum computing, a computational architecture harnessing the principles of quantum mechanics. At the core of quantum computation lies the quantum bit, or qubit, which exhibits unique properties such as superposition and entanglement, paving the way for unprecedented computational capabilities. Currently residing in the NISQ[1] era (Noisy Intermediate-Scale Quantum), quantum computing is marked by its exploration of limited-size and noise-sensitive quantum machines. The NISQ era acknowledges the ongoing challenges in quantum hardware but emphasizes the potential for valuable research and exploration of quantum advantages, even within the constraints of current technology.

In the landscape of optimization, two prominent approaches dominate the quantum computing arena. Universal Quantum computers, operating on the logic circuit gate model, with proven advantages in algorithms such as Shor's [2], Grover's [3] and Creemers et al.[4, 5], but with limited sizes (as of 2023 the biggest Universal Quantum computer has 433 qubits [6]). On the other hand, adiabatic quantum computers, grounded in the principle of adiabatic quantum computation, exhibit promising experimental results, particularly in optimization problems leveraging the ability to map them as energy minimization problems and the natural tendency of the universe to seek states of minimum energy.

Diverging from gate-based quantum computers, quantum annealers carve a niche with a more focused application scope. These systems boast a higher qubit count (+5000) and enhanced noise resilience, fueling extensive research from both industry and academia. The versatility of quantum annealers is illustrated in their application to complex problems like graph partitioning[7, 8], transportation problems[9][10], Job-Shop Scheduling Problem (JSSP) [11, 12], and many others applications[13][14][15][16].

This paper delves into the realm of adiabatic quantum algorithms, with a specific focus on quantum annealing, to tackle one of the most challenging NP-Hard scheduling problems—Resource Constrained Project Scheduling Problem (RCPSP). The significance of this choice arises from the inherent complexity of the problem, its broad applicability in projects' makespan minimization, the non-trivial nature of its formulation as input of the quantum machine. Notably, this work represents the first exploration of applying quantum annealing techniques to the RCPSP.

The primary objective of this research is to present a comprehensive analysis of the potential and limitations of the current Quantum Annealing technology. Our study meticulously outlines a step-by-step approach to solving the RCPSP on the cutting-edge D-Wave Advantage quantum annealer featuring 5640 qubits. Through this exploration, we aim to contribute not only to the theoretical understanding of quantum annealing but also to the practical application of this technology in addressing real-world optimization challenges. Fig. 1 provides a comprehensive visual representation of the methodology and key processes employed in this study.

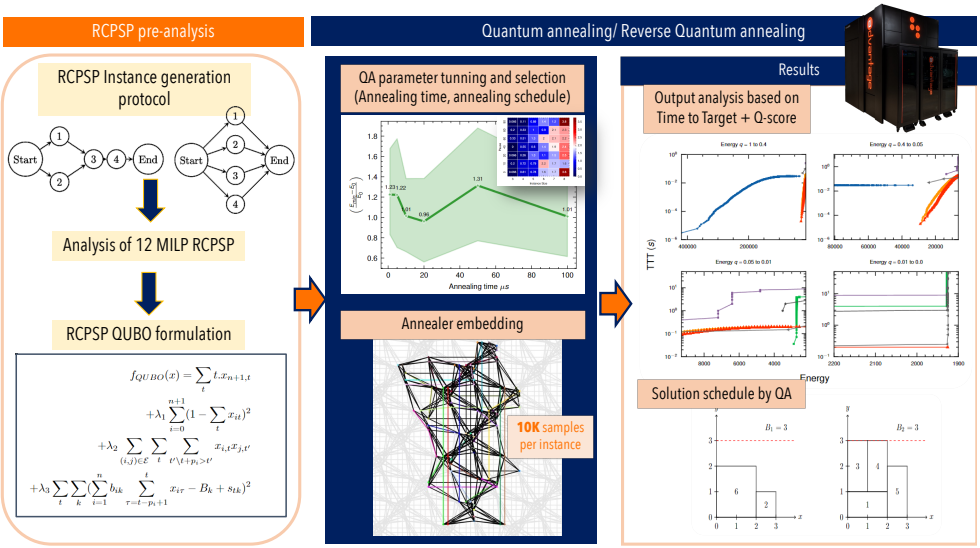


Fig. 1: Visual Abstract: A schematic overview of the methodology employed in this study

This paper provides an in-depth exploration of quantum annealing's theoretical and practical aspects, specifically in relation to the RCPSP. Initially, we examine the quantum annealing fundamentals, focusing on the problem-embedding process into the annealer's qubit graph. We then introduce the RCPSP and explain the methods used in this study, including the selection of instance generators from existing literature, the adoption of metrics like "Time-to-Target"[17] (TTT) and Atos q-score, and strategies for utilizing a quantum annealer for RCPSP (annealing time, shots, reverse schedule). The Results section, preceding the conclusion on open research questions, presents the first QUBO model of RCPSP, derived from an analysis of 12 existing MILP formulations. Subsequently, we conduct an extensive computational experiment to assess the efficacy of quantum annealing in solving RCPSP, discussing key factors such as solution sampling, annealing duration, chain strength, and the use of advanced controls like Reverse Annealing.

Quantum Annealing and D-Wave machines

Quantum Annealing (QA) stands as a quantum metaheuristic designed to tackle combinatorial optimization problems by leveraging the principles of quantum mechanics [18, 19]. This approach draws inspiration from simulated annealing [20], a classical metaheuristic, where a system systematically cools to reach a state of minimum energy. In quantum annealing, quantum phenomena such as superposition and quantum tunneling [21] are harnessed to navigate through local minima efficiently, aiming to pinpoint the global minimum of a cost function. The efficacy of quantum annealing compared to its classical counterpart can vary depending on the specific problem and hardware employed. The literature offers both theoretical proofs [18], highlighting the

advantages of quantum annealing, as well as contrasting perspectives—particularly in comparison to Simulated Annealing [22][23][24].

At the core of quantum annealing lies the profound concept of the natural tendency of the universe to seek systems of minimum energy. This method is grounded in the "adiabatic theorem" of quantum mechanics [25, 26] and the time-dependent Schrödinger equation [27]. It describes the evolution of any quantum system, encapsulating the dynamics of its quantum states. The Hamiltonian, typically represented as a matrix, encapsulates all the information about the quantized energy states available to a quantum system. The adiabatic theorem of quantum mechanics asserts that if a quantum system undergoes a slow and continuous change in its Hamiltonian, it will persist in its instantaneous eigenstate throughout the transformation. In simpler terms, if the system initiates in its ground state and the Hamiltonian changes sufficiently slowly, the system will maintain its ground state.

Applying the adiabatic evolution theorem to optimization problems involves initiating the process with a "simple" initial system represented by the Hamiltonian $\mathcal{H}_0$, for which the ground state can be easily determined. The system is then gradually transformed into the problem Hamiltonian $\mathcal{H}_1$, an energetic mapping of an optimization problem $\mathcal{E}$. At the conclusion of the adiabatic evolution, the system originally described by $\mathcal{H}_0$ would have transitioned into the ground state of the problem Hamiltonian, representing the minimum value solution to the optimization problem $\mathcal{E}$. The QA algorithm is physically implemented using analog control devices to manipulate a collection of qubit states, following a time-dependent Hamiltonian represented as:

$$\mathcal{H}(t) = A(t)\mathcal{H}_0 + B(t)\mathcal{H}_1$$

This algorithm orchestrates a gradual transition from an initial ground state in $\mathcal{H}_0$ to a state described by the problem Hamiltonian $\mathcal{H}_1$. The $\mathcal{H}_1$ Hamiltonian mirrors the energy function of the optimization problem, ensuring that the ground state for $\mathcal{H}_1$ corresponds to a minimum-cost solution to the optimization problem $\mathcal{E}$. Propounded by Farhi et al. [28], Quantum Annealing and in more general the Adiabatic Quantum Model of Computation demonstrates that if the transition is executed slowly enough, the algorithm will, with high probability, converge to a ground state, i.e., an optimal solution.

Quantum Annealing D-Wave implementation

The D-Wave Quantum Annealing processors are purposefully engineered for the specific task of identifying minimum-cost solutions to the Ising Minimization problem or, indirectly, to an isomorphic problem: the QUBO. The Ising problem, defined on a graph $G = (V, E)$, entails the assignment of values from $\{-1, +1\}$ to spin variables s_i with the objective of minimizing the following energy function $\mathcal{H}_1$:

$$\mathcal{H}_1 = \sum_i h_i s_i + \sum_{i,j} J_{i,j} s_i s_j$$

Where $h = h_i : i \in V$ denotes weights, and $J = J_{ij} : (i, j) \in E$ represents coupling constants. In the physical context, spin variables s_i can be construed as magnetic

poles, with negative J_{ij} indicating ferromagnetic interactions and positive values suggesting antiferromagnetic interactions. The optimal configuration of spin variables s_i that minimizes the energy function is denoted as a ground state, while alternative configurations are classified as excited states. Broadening the computational scope, Ising problems effortlessly transformed into Quadratic Unconstrained Binary Optimization (QUBO) problems via the equality $s_i = 2x_i - 1$. This modification involves associating binary variables $x_i \in \{0, 1\}$ with spin variables $s_i \in \{-1, +1\}$.

A D-Wave Quantum Processor Unit (QPU) maintained at temperatures around a few millikelvin exhibits quantum properties such as superposition and quantum tunneling. Despite the presence of a Faraday shield, the QPU remains susceptible to interference which generally reduces the likelihood of attaining a ground state. Consequently, we categorize any D-Wave processor as a heuristic solver, requiring empirical methods for performance analysis. The current most advanced D-Wave processor (Advantage 6.3) has more than 5000 active qubits, a qubit connectivity of 15 qubits (qubits are not filly connected) and 35000 active couplers, which are made of microscopic loops of niobium. These couplers are connected to a sophisticated analog control system through a network of Josephson junctions [29–31]. Table 1 lists the properties of the different topologies made available by D-Wave: Chimera, Pegasus, and Zephyr.

Table 1: Characteristics of available D-Wave Quantum Devices.

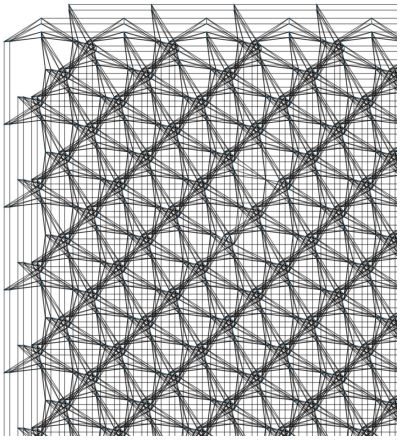
Topology	Chimera	Pegasus	Zephyr
Device Designation	2000Q	Advantage6.3	Advantage2 (Prototype)
Active Qubit Count	2041	5616	563
Connectivity Measure	6	15	20

Connectivity and Embedding

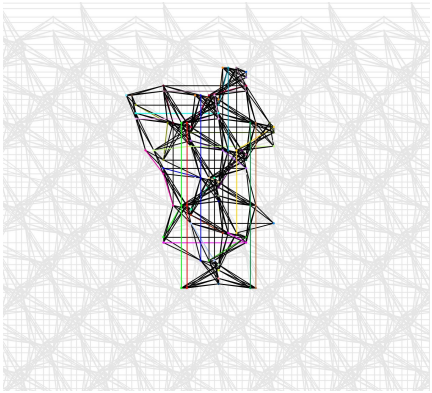
Besides noise and interference, another major challenge is qubit connectivity. In order to solve a problem on a quantum annealer, the problem graph (QUBO/Ising) must be mapped into the physical hardware, which has a limited number of qubits and interconnections. Consequently, it may be necessary to alter the structure of the QUBO/Ising problem to fit within the constrained topology of the quantum annealer. This involves mapping the model onto a larger qubit graph, a critical step that stresses the importance of the machine's inherent topology. This non-trivial process is known as 'minor embedding'. Heuristics exist in the literature and the search for better embedding algorithms remains an active area of research [32][33][34][35]. For this work minor embeddings are going to be found using the available minor-miner D-Wave heuristic [36].

Fig. 2 shows an example of the embedding process with the "Pegasus" Topology. During the minor embedding process when qubits are not directly connected, but the original problem graph requires them to be, then the embedding process requires the creation of a so called "chain". In this scenario, a logical variable is represented by a chain of multiple physical qubits entangled between themselves. To maintain solution consistency, it is imperative that all physical qubits within a chain have the same

value once measured; if not, the chain is considered broken. The chain strength is a parameter introducing a trade-off: if set too low, it may lead to many broken chains, but if too high, it can impede the qubits' ability to change states.



(a) Pegasus topology of the QPU.



(b) RCPSP problem mapped into the QPU pegasus topology.

Fig. 2: Minor embedding process with the pegasus topology.

Anneals and Reverse Annealing

Once the minor embedding process is successfully completed, the annealing process is initiated. While the adiabatic theorem suggests a potentially lengthy annealing duration to ensure the system stays in the ground state, practical constraints arise due to noise, which can elevate the system to higher energy states. Consequently, a brief annealing is conducted and repeated multiple times in a stochastic process that samples from the energy distribution of the problem. Determining the optimal annealing time and the number of samples is contingent on the specific problem and its energy distribution. Typical values fall within the range of 10 to 100 microseconds for annealing times and 1 to 10,000 for the number of samples.

While the traditional Quantum Annealing starts from the ground state of the "Ising Transverse field" Hamiltonian $\mathcal{H}_0$ and thus from an initial uniform superposition and evolves towards the target problem Hamiltonian $\mathcal{H}_1$, a more sophisticated evolution approach can be applied. In Reverse Annealing (RA), the process departs from an already available solution. Operating in reverse, RA reintroduces partial segments of $\mathcal{H}_0$, thereby reinstating a partial quantum superposition. This unique reverse progression is employed to iteratively refine the initial solution [37][38][39][40].

Resource-Constrained Project Scheduling Problem

The RCPSP is perhaps one of the most extensively studied scheduling problems and perhaps one of the easiest to describe; but despite this apparent simplicity, the RCPSP

conceals its true complexity, as demonstrated by Blazewicz et al.[41], who categorized it as an NP-hard problem.

This inherent complexity renders the RCPSP one of the most intractable combinatorial optimization problems, in fact similar to the JSSP [42, 43] the RCPSP falls into the category of problems classified as NP-hard "in the strong sense" [44].

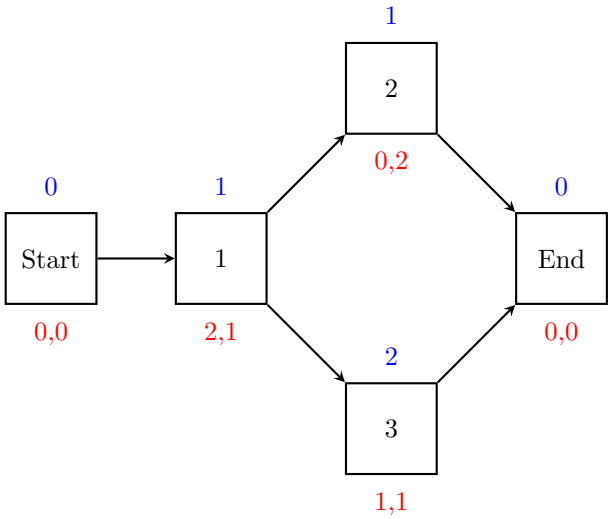
In essence, the RCPSP involves the scheduling of a single project comprising n real activities, subject to precedence and resource constraints, with the overarching objective of minimizing the project makespan – the total time required for the completion of all activities. The RCPSP is typically represented by a graph $G(\mathcal{A}, \mathcal{E})$ with each node $\mathcal{A} = \{0, 1, ..., n+1\}$ that corresponds to the different project activities and each edge $(i, j) \in \mathcal{E}$ equivalent to a straightforward finish-to-start precedence relationship, which means that the start of a successor activity j must await the conclusion of it's predecessor activity i. Nodes 0 and $n+1$ serve as symbolic milestones, representing the "project start" and "project finish," respectively, these milestone activities are often referred as "dummy activities". Each activity $j \in \mathcal{A}$ has a duration p_j and resource consumption's $b_{j,k}$ where k belongs to a set of renewable resource $\mathcal{R}$. Each resource k has a maximum capacity B_k.

A feasible solution to the RCPSP corresponds then to a project schedule $\mathcal{S} = \{S_0, S_1, \ldots, S_{n+1}\}$ comprised of the start times S_i for each activity $j \in \mathcal{A}$, that respects both the precedence and the resources constraints.

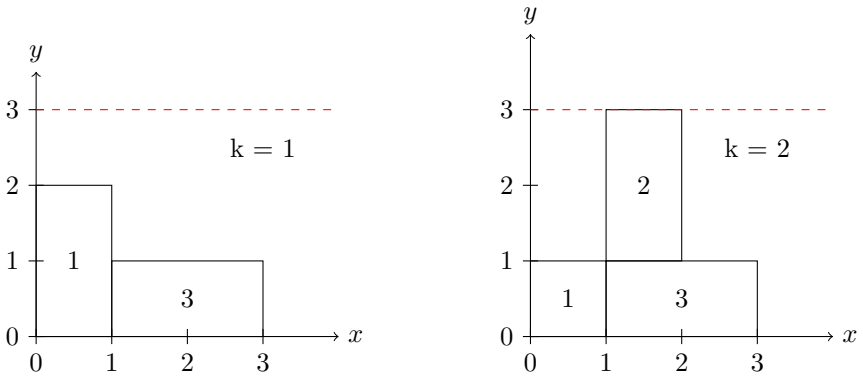
An inherent characteristic of the RCPSP is the non-preemptive nature of activities, signifying that once an activity commences, it cannot be interrupted.

Fig. 3a, depicts the graphical representation of an small RCPSP instance (known as a network diagram) composed of three activities (plus two "dummy" activities) and two renewable resources. The activity duration, denoted as p_j, and resource consumption $b_{j,k}$ are presented above and below each node j, respectively. Both renewable resources have a maximum capacity of three units. Fig. 3b. shows the optimal schedule $\mathcal{S} = \{S_0 = 0, S_1 = 0, S_2 = 1, S_3 = 1, S_4 = 3\}$ for this instance.

(a) Network Diagram.



(b) RCPSP Solution schedule $\mathcal{S}$



(c) Schedule for resource 1.

(d) Schedule for resource 2.

Fig. 3: RCPSP instance example: a) RCPSP instance graph $G(\mathcal{A}, \mathcal{E})$ 3 activities (plus "dummy" Start and End) 2 resources b) Solution Schedule for the instance c) Resource consumption for resource 1 d) resource consumption for resource 2.

Mixed Integer Linear Programming formulations

Given the intrinsic significance of the RCPSP, it is unsurprising that the literature dedicated to Mixed Integer Linear Programing (MILP) formulations for the RCPSP is both prolific and dynamic, offering a myriad of approaches. For a given optimization

problem, multiple formulations can be devised, they can be distinguished primarily by the formulation size, notably concerning the number of variables, secondly by the number of constraints they involve, and thirdly by the strength of their Linear Programming (LP) relaxation. Typically for a given problem a discernible correlation often exists between the problem size and the quality of the LP relaxation. Traditionally it has been the case that to enhance the quality of the LP relaxation, a new extended formulation introduces additional variables and constraints, increasing thus the problem size. With the remarkable advancements of LP algorithms, which can now efficiently solve instances involving millions of variables [45],[46],[47], it comes as no surprise that, in the context of MILP formulations for the RCPSP, a significant emphasis has been placed on advancing and refining the quality of the LP relaxations [48, 49].

Broadly classified, formulations for the RCPSP fall into three distinct families, as illustrated in Fig. 4. The first category encompasses time index formulations [50–56], including works by Pritsker et al. 1969 (PRI69), Christofides et al. 1987 (CHRI87), de Sousa and Wolsey 1997 (SOUWO97), Mingozzi et al. 1998 (MINGZ98), Klein and Kaplan 1998 (KLEIN/KAPL98), Klein 2000 (KLEIN00_1 and KLEIN00_2), Demeulemeester and Herroelen 2002 (DEMHER02), and Bianco and Caramia 2013 (BIACAR13) . This is followed by sequence-based formulations [57, 58], exemplified by Tamarit and Valdés 1993 (TV93) and Artigues et al. 2003 (ART03) . Lastly, the third category comprises event-based formulations [59, 60], including works by Kone et al. 2011 (KONE11) and Artigues et al. 2013 (ART13). An analysis by Kone and Artigues [60, 61] reveals that sequence and event-based formulations tend to be more compact, requiring fewer variables, while time index formulations, although larger in scale, offer superior Linear Programming (LP) relaxations. It is noteworthy that the majority of time index formulations tend to be binary formulations, utilizing exclusively binary variables, while event and sequence-based formulations employ a mix between integer and binary variables.

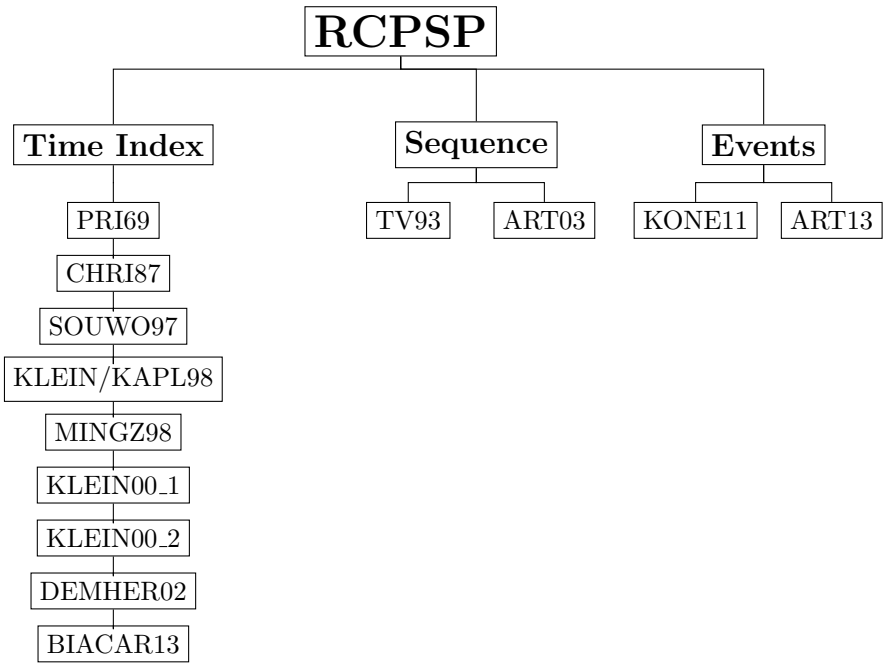


Fig. 4: Multiple RCPSP MILP formulations reported in the literature categorized by their family type

In the context of employing a quantum annealer for solving the RCPSP, a pertinent question emerges: among the various available formulations, which one to choose? Specifically, which formulation aligns most effectively with the capabilities of current commercially accessible quantum annealers? Addressing these inquiries necessitates an investigation into the transformation of these formulations into their corresponding Quadratic Unconstrained Binary Optimization (QUBO) form. This evaluation involves assessing each formulation based on factors such as the resulting QUBO size, the need for additional slack supplementary variables, and the sparsity of the QUBO graph associated with each. These characteristics play a pivotal role in determining the suitability of each formulation for solving the RCPSP, particularly within the confines of Noisy Intermediate-Scale Quantum (NISQ) era quantum annealers. This study aims to provide comprehensive insights to answer these critical questions.

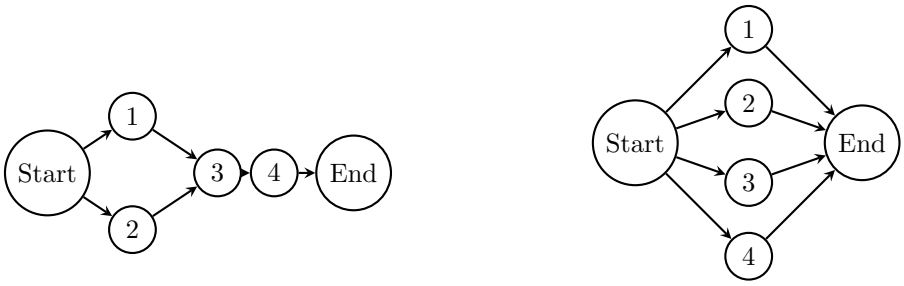
Methods

In the following section we explain different aspects of the experimental methodology followed in this study.

RCPSP instance selection protocol

A classical optimization study related to the RCPSP would normally involved instance froms the well-recognized PSPLIB dataset of Kolisch et al. [62, 63] or perhaps the more recent CV dataset of Coelho and Vanhoucke[64]. Both datasets offer a plethora of hard instances of varying sizes (number of activities, ranging from 20 to 120 activities). However, considering the limitations on the number of available qubits in the D-wave advantage 6.3 quantum annealer, we decided to utilize the Rangen1 instance generator proposed by Demeulemeester, Vanhoucke and Herroelen [65, 66]. This generator allows the creation of random instances with varying levels of difficulty.

The instances used on this work were generated following the well established protocol proposed by Baptiste and Le Pape [67], which involves constructing both disjunctive and cumulative instances. Disjunctive instances are characterized by dominance of precedence constraints, resulting in a highly sequential schedule with limited opportunities for parallel execution. On the other hand, cumulative instances exhibit fewer precedence constraints, providing ample opportunities for parallelism and posing a greater challenge for solving.



(a) Disjunctive. (b) Cummulative.

Fig. 5: Baptiste and Le Pape instance types: a) The left graph corresponds to a disjunctive RCPSP instance, dominated by precedence constraints $\mathcal{E}$ b) Right graph corresponds to a cummulative RCPSP instance, with opportunities for parallelism.

The Rangen1 generator facilitates the generation of both disjunctive and cumulative instances through a parameter called "Order Strength" (OS), ranging from 0 to 1. A value of 0 indicates a fully cumulative instance as in Fig. 5b, while 1 represents a fully disjunctive instance as in 5a. The generator also allows the tuning of resource constraints by specifying the number of resources, along with the level of constraint for renewable resources, this is done via a parameter called "Resource constrainedness" (RC) that again ranges from 0 to 1 where 1 means that all activities consume the maximum resource capacity B_k, while 0 means that there is no resource consumption by the project activities.

In our study, we utilized Rangen1 to produce instances of the RCPSP across various dimensions, determined by the number of activities 3, 4, 5, 6, 7, and 8, excluding dummy activities). Within each size category, we generated three distinct instance types characterized by Order Strength (OS) values of 0.1, 0.5, and 0.9. Each instance

featured two renewable resources ($k = 2$) with a RC set to 0.5. The activity durations p_j, and resource consumption values $b_{j,k}$, were constrained within the range of 1 to 2 units, to further limit the number of qubits required by the quantum annealer. Furthermore, for every instance, the maximum resource capacity B_k was uniformly established at 3 units.

Benchmarking metrics for quantum optimization

Benchmarking metrics play a pivotal role in the evaluation of adiabatic evolution computing algorithms and particularly for quantum annealing. In the context of benchmarking there are inherent challenges that arise from the unique nature of quantum annealers. First QA blends quantum and analog elements, that are devoid of discrete instructions or basic operations that lend themselves to conventional counting, as observed with classical computers. Consequently, resorting to runtime becomes a pragmatic yet intricate solution, given the transient and unstable nature of QA.

A further complication arises from the juxtaposition of hardware-implemented quantum annealing algorithms against their software-implemented counterparts (simulated quantum annealing). Traditional benchmarks for computer platforms, software, and algorithms often fail to account for this mixed scenario, leading to a deficiency in standard guidelines for robust benchmarking.

Unlike classical benchmarks that often distill performance into a single metric, the evaluation of adiabatic evolution algorithms, particularly in the quantum realm, necessitates a more nuanced approach. The performance of a heuristic on a given input is aptly described by a curve delineating the trade-off between computation time and solution quality. This nuanced perspective requires a repertoire of metrics for comprehensive evaluation.

In the empirical evaluation of quantum annealing and adiabatic evolution algorithms, several performance metrics have surfaced in the literature. Notable among these are "Time-to-Solution" [68] (TTS), which centers on the total time required for a solver to identify a ground state (optimal solution) with a sufficiently high probability. One of the biggest disadvantages of TTS, is that it relies on a priori knowledge of the optimal solution, potentially overlooking benefits derived from near-ground state solutions.

Alternatively, "Time-to-Target" (TTT) [17] becomes relevant, focusing on the overall time required by solvers to attain a target solution energy, determined by the energy distribution of the quantum annealing processor. This metric provides a more versatile evaluation, acknowledging solutions that may not align with a predetermined optimal outcome.

Another recent interesting metric is the "q-score" [69] conceived by the technology consulting company Atos. The q-score measures the maximum number of qubits effectively employed to solve combinatorial optimization problem. Originally the q-score was developed to evaluate the solution of the Max-Cut problem using the Quantum Approximate Optimization Algorithm [70] (QAOA), howver this metric can be readily adapted to assess the performance of quantum annealing as it was shown by Van der Schoot et al. [65]. The q-score of a given problem can be then calculated by:

$$n^* \equiv \max\{n \in \mathbb{N}, \beta(n) > \beta^*\} \tag{1}$$

With $\beta^* = 0.2$ (obtained empirically by studying the behavior of QAOA). and $\beta(n)$ determined by the following ratio:

$$\beta(n) = \frac{\tilde{C}(n) - C_r(n)}{C_{max}(n) - C_r(n)} \tag{2}$$

Where $\tilde{C}(n)$ is the average energy output obtained from QA, $C_r(n)$ is the average output value obtained from solving the QUBO/Ising problem with a random sampling method, and $C_{max}(n)$ equal to the energy of the ground state (optimal solution).

For the specific focus of this paper, the metrics of interest include TTT and the q-score. By concentrating on these metrics, we aim to ascertain the potential for speed-ups and determine the largest instance of the RCPSP, measured in terms of the number of project activities, that can be effectively addressed using the current advantage 6.3 quantum annealing system from D-Wave.

QUBO Penalty selection strategy

Adjusting the values of the multipliers weighing the penalties due to relaxed constraints is an important but difficult step. Given a QUBO problem of energy minimization where $E(x) = x^T Q x$, with a number of n binary variables $x_i, \forall i \in \mathcal{X}$ we can further decompose this problem into two portions $E(x) = v(x) + \lambda c(x)$ where $v(x)$ corresponds to the energy contribution to the objective function and $c(x)$ corresponds to the energy contribution due to the problem constraints and λ equal to the penalty weight. The main goal then is to find the value of λ such that the optimal solution to the penalised objective function is the optimal solution of the original constrained problem. Multiple strategies have been proposed in the literature, each offering a unique approach to determining the most effective penalty coefficients.

One of the earliest and simplest strategies was proposed by Lucas in 2014 [71]. This method involves utilizing the upper bound of the pure objective function, which is mathematically represented as $\lambda = x^T Q x$, where $x_i = 1, \forall i \in \mathcal{X}$. This approach provides a straightforward and easily computable penalty.

Lucas also introduced another possibility for penalty selection, which involves using the maximum QUBO value, denoted as:

$$max(Q_{i,j}) \quad \forall \quad (i,j) \in \mathcal{X}^2 \tag{3}$$

This method takes into account the highest interaction value between the variables, ensuring that the penalty is significant enough to enforce the constraints effectively.

More recently, Verma and Lewis in 2020 [72] proposed a more sophisticated method. Their approach estimates the potential gain or loss in the objective function value that could result from switching a particular bit on or off. This method provides a more nuanced and dynamic way of calculating penalties, potentially leading to more accurate and efficient solutions, especially in complex scenarios where the impact of each binary variable on the objective function is not uniform.

$$W_c = \left\{ -Q_{i,i} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \min\{Q_{i,j}, 0\}, Q_{i,i} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \max\{Q_{i,j}, 0\} \quad \forall i \in \mathcal{X} \right\} \tag{4}$$

$$\lambda = \max_{i=1}^{n} W_{c_i} \tag{5}$$

The array of strategies discussed underscores the dynamic and continuously evolving landscape of penalty selection in QUBO problems. Each approach, distinct in its methodology, brings its own set of strengths and is tailored to suit various problem types. As penalty selection stands as a vibrant and ongoing field of research, our study opts for a more foundational approach. We have chosen to employ a simple penalty selection method where the penalties are just a multiple of the sum of activity duration's p_i, $\lambda = \sum_{j=1}^{n} p_i$, leveraging the upper bound strategy, to ensure clarity and ease of implementation in our analysis.

Experimental setup

The experimental results presented in this study were derived from the utilization of the D-Wave Advantage 6.3 quantum annealer, featuring 5640 qubits and a connectivity of 15 connections per qubit. The Minor embeddings were established employing D-Wave's "minor-miner" heuristic. Each instance from section 3 underwent execution on the QPU with a specific annealing time of 20 µs (the decision to use $20\mu s$ is based on the analysis conducted on section 3 and it is explicitly showed in Fig. 16), and for every instance, 10,000 samples were recorded. It is crucial to note that the mentioned annealing time exclusively pertains to quantum annealing and does not extend to reverse quantum annealing, since RQA uses an specific annealing schedule described below.

Fig. 6 illustrates the implementation of reverse quantum annealing using a 4-point schedule: [[0.0, 1.0], [2.75, 0.45], [82.75, 0.45], [83.025, 1.0]]. The schedule involves a reverse evolution from $s = 1$ to $s = 0.45$ within the initial 2.75 µs, followed by an 80 µs pause. Subsequently, a forward evolution of 1 µs occurs, transitioning from $s = 0.45$ to $s = 1$. Here, the variable s represents the percentage of implementation of the problem Hamiltonian $\mathcal{H}_1$ during the annealing evolution.

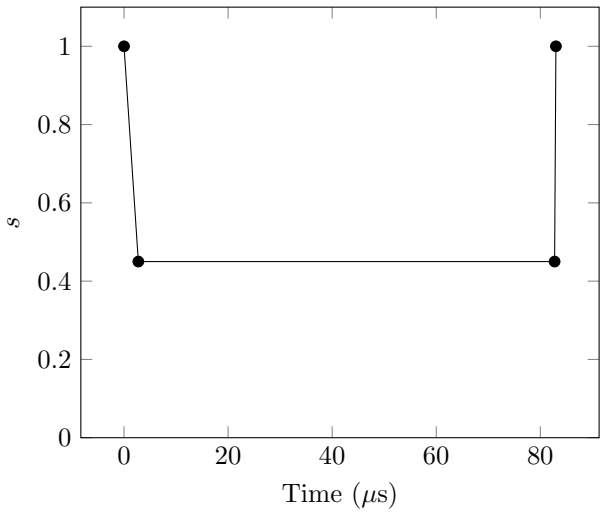


Fig. 6: Reverse Quantum Annealing Schedule used for the RQA numerical results of this work

The impact of including pauses in the annealing schedule on the performance of QA was assessed in Section 3. A annealing time of 20 μs was employed, incorporating pauses of 2, 4, 6, 8, 12, 16, and 18 μs, respectively. The resulting annealing schedules are visualized in Fig. 7. This analysis was comprehensive, covering all instance types and sizes. The evaluation involved recording the relative difference between the energy of the ground state E_0 and the minimum energy value achieved by QA, denoted as E_{min}. This difference is calculated as $\left(\frac{E_{min}-E_0}{E_0}\right)$. For each instance, 1000 samples were recorded. A similar analysis was conducted to evaluate the effect of different annealing times 1,5,10,20,50 and 100 μs in the performance of QA.

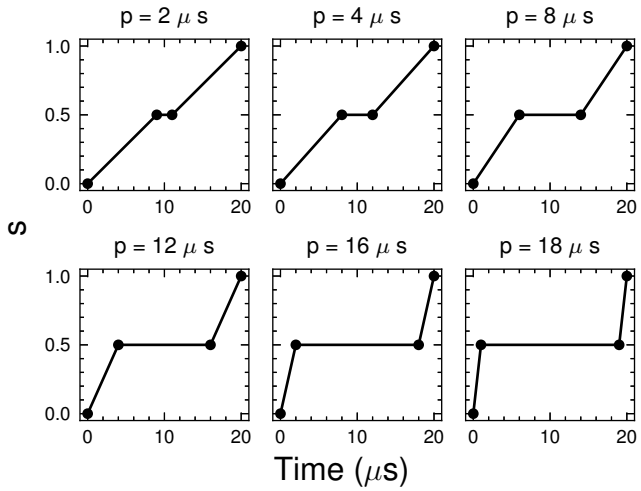


Fig. 7: Annealing Schedules with pauses of different lengths 2,4,8,12,16 and 18 μs for fixed annealing time of 20 μs. These pauses correspond then to 10,20,40,60,80 and 90% pause of the annealing time

To create a comprehensive benchmark for assessing the performance of Quantum Annealing (QA) and Reverse Quantum Annealing (RQA) in this study, we have selected the following classical optimization techniques: Random Sampling (RS) and Simulated Annealing [20] (SA), along with classical solvers including GUROBI, COIN-CBC, and GLPK.

Results

QUBO analysis of MILP formulations for the RCPSP

In this section, we identify the optimal QUBO formulation for the RCPSP, focusing on the one that requires the fewest qubits. The minimal number of qubits is directly related with the number of slack variables, the type of variables, and the sparsity of the QUBO graph. Initially, we examine these crucial elements across 3 types of RCPSP formulations. Subsequently, from the eight formulations we have studied within the selected type, we determine the most effective one.

Analysis of formulation types

Formulations based on Time-Index, Sequence, and Events types (see Fig. 4), once transformed into QUBO, are compared with the data presented in Table B1. The number of qubits required for each formulation is reported in Fig. 8. This Fig. shows that the time index formulation "PRI69" necessitates significantly fewer qubits compared to the sequence-based and event-based formulations. Moreover, the gap between these formulations increases in instance size. These results may appear counter intuitive when considering the original number of variables required by each formulation, as detailed in Table B1. Notably, sequence-based and event-based formulations exhibit a lower number of original variables than time index formulations, this compactness characteristic has been well-documented by Kone [61].

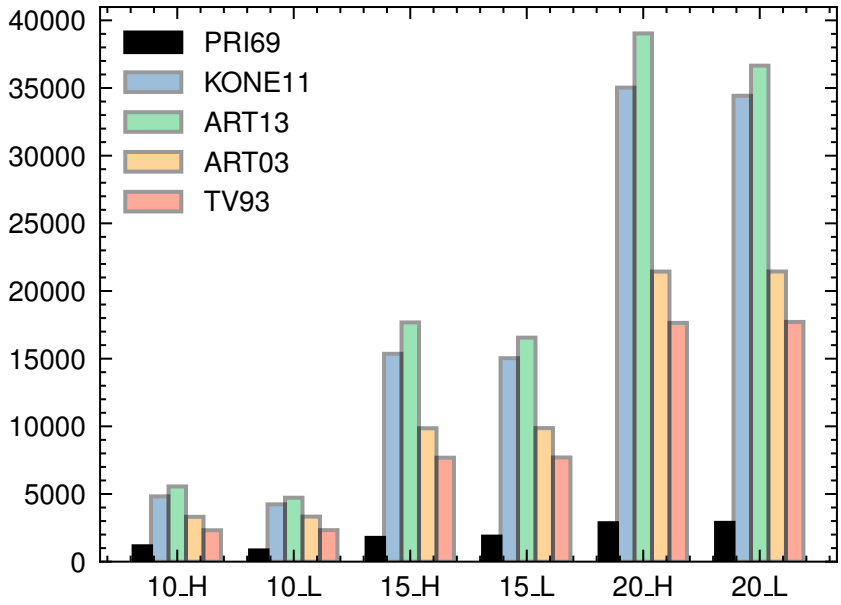


Fig. 8: Number of qubits required by the different families of the RCPSP MILP formulations. H refers to Disjunctive instances of OS = 0.9 and L refers to cummulative instances of OS = 0.1.

The apparent discrepancy in the number of QUBO variables can be elucidated by closely inspecting the nature of the different formulations. The "TV93" formulation necessitates a priori knowledge of "forbidden sets" $\mathcal{F}$. These are sets of activities that share no precedence constraints and, when scheduled in parallel, violate resource constraints. Demeulemeester and Herroelen [73] demonstrated that generating the minimum forbidden sets entails a worst-case complexity of $O(2^n)$. Consequently, it logically demands more qubits compared to "PRI69", specially as the instance size increases, since the size of the forbidden sets $\mathcal{F}$ increases accordingly.

In the "ART03" resource flow formulation, instead of using forbidden sets $\mathcal{F}$, integer variables $\phi_{i,j}^k$ (where $i, j \in V^2$ and $k \in R$) are employed, each necessitating $\lfloor \log_2 \phi_{i,j}^k \rfloor + 1$ binary qubits for transformation. These variables take part of multiple inequalities for maintaining resource constraints, leading to the need for additional slack integer variables, which also require binary transformation. Similarly, the event formulations "KONE11" and "ART13" utilize integer variables t_e to represent event start times, with "KONE11" also incorporating extra integers $b_{e,k}$ for resource consumption. These variables, also involved in several inequalities, necessitate further slack variables for binary conversion.

Analysis of time-index formulations

Taking the same RCPSP instances, Table B2 presents the outcomes resulting from the conversion of 8 distinct RCPSP time-index MILP formulations into QUBO.

The number of qubits per formulation is represented by Fig. 9 which shows the efficiency of the "PRI69" formulation of that aspect compared to the other time-index formulations. This outcome is unsurprising given its inherent simplicity characterized by fewer variables, a diminished count of inequalities, and consequently, a reduced number of slack variables. The next favorable option is the "MINGZ98" formulation; however, this choice entails the prerequisite knowledge of feasible sets $\mathcal{A}$. Similar to "TV93", the determination of these sets $\mathcal{A}$ entails a worst-case complexity of $O(2^n)$.

For the other formulations, the various strategies aimed at optimizing solutions for a classical computer prove to be costly in terms of the number of qubits required. We can highlight a few of them here: the "CHRI87" formulation modifies the "PRI69" formulation by replacing its precedence constraint with additional inequalities. "SOUWO97' introduces 'Step' variables at the beginning, while "KLEIN/KAPL98" considers a 'Step' formulation based on the MILP of a preemptive RCPSP case [55]. The "KLEIN00_1" and "KLEIN00_2" formulations incorporate new 'On/Off' variables. Additionally, "BIACAR13" integrates continuous binary variables indicating the percentage of activity completion while "MINGZ98" , bearing similarities to "TV93", requires prior knowledge of 'feasible sets' A.

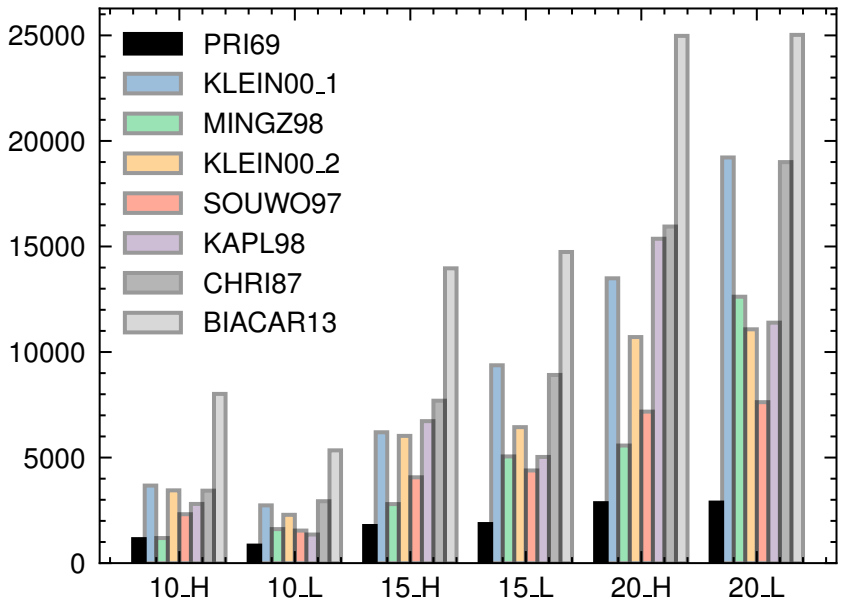


Fig. 9: Number of qubits required by the different time index MILP formulations of the RCPSP. H refers to Disjunctive instances of OS = 0.9 and L refers to cumulative instances of OS = 0.1.

RCPSP QUBO

Pritsker et al.[50] provided "PRI69", one of the first formulations of the RCPSP. Although it may initially seem counterintuitive, we have chosen this formulation as the basis for writing our QUBO, based on the analysis in the previous section. We use the notations given in Table 2 to introduce the parameters.

Notation	Definition
n	Number of activities (excluding the two dummy activities indexed by 0 and $n+1$.)
$\mathcal{T}$	Maximum number of time periods.
$\mathcal{A}$	Set of Activities. We consider an activity $i \in \mathcal{A} \cup \{0, n+1\}$ if i can be any real activity, the first or the last virtual activity.
$\mathcal{H}$	$\mathcal{H} = 0, 1, ..., T$ is the scheduling horizon.
$\mathcal{E}$	Set of edges representing the sequence from one activity to the next. E.g., if i and j are two activities of $\mathcal{A}$, and if activity i must be finished to start activity j, $(i, j) \in \mathcal{E}$.
$\mathcal{R}$	Set of resources.
p_i	Processing time of the activity $i \in \mathcal{A}$.
B_k	Capacity of the resource $k, k \in \mathcal{R}$.
b_{ik}	Activity i consumption of the resource $k \in \mathcal{R}$ and its capacity B_k.

Table 2: RCPSP parameters used in the "PRI69" time index formulation.

A single type of binary decision variable is considered, denoted as x_{it}, with $i \in \mathcal{A} \cup \{0, n+1\}$ and $t \in H$. This variable is indexed by both the activities and the associated time. Each element x_{it}, $\forall i \in \mathcal{A} \cup \{0, n+1\}$ and $\forall t \in H$, takes a $\{0; 1\}$ value such that:

$$x_{it} = \begin{cases} 1 \text{ if the activity } i \text{ starts at the period } t, \\ 0 \text{ otherwise.} \end{cases}$$

We are revisiting the initial Objective function of [50], i.e., a function to minimize that forces the last virtual activity to finish as soon as possible. This function is noted $f(x)$ and its expression is given by the equality (6). This is the basis of our QUBO which will then be completed with penalties corresponding to the relaxations of the constraints.

$$f(x) = \sum_t t.x_{n+1,t} \tag{6}$$

Moreover we also consider the work of [11, 12] on the JSSP especially for the precedence constraints and the one start constraints reformulation. For the latter, we force each activity to start exactly one time with the following set of constraints (7).

$$\sum_t x_{it} = 1, \qquad i = 0..n+1 \tag{7}$$

We relax these constraints for all activities and turn them into one penalty $P1$ given by the expression (8). Here, to ensure that the search for an optimal solution penalizes any infeasible solution, the expression is squared so that satisfying the constraint gives

no penalty to the QUBO (i.e., equals to the value 0), and that violating the constraint goes in the opposite direction of minimization.

$$P1(x) = \sum_{i=0}^{n+1}(\sum_{t} x_{it} - 1)^2 \tag{8}$$

We can model the precedence constraints of two consecutive activities by the inequalities (9).

$$\sum_{t\in\mathcal{H}} t.x_{j,t} \leq \sum_{t\in\mathcal{H}} t.x_{i,t} + p_i \qquad \forall(i,j) \in \mathcal{E} \tag{9}$$

The reformulation of all precedence constraints into a single penalty, denoted as $P2$, results in the expression shown in (10). Here, there is no need to square them, since any quadratic expression involving two binary variables never yields a negative value. It is easy to understand that the only scenario where such constraint is not satisfied occurs when two consecutive activities erroneously start at the same time.

$$P2(x) = \sum_{(i,j)\in\mathcal{E}} \sum_{t\in\mathcal{H}} \sum_{\substack{t'\in\mathcal{H} \\ t'\setminus t+p_i>t'}} x_{i,t}x_{j,t'} \tag{10}$$

While the JSSP has sharing machine constraints which can also be modeled by simple quadratic expressions giving a similar penalty of $P2$, the RCPSP has another difficulty: the resource constraints. The inequalities (11) express these constraints as modeled in [50].

$$\sum_{i=1}^{n} b_{ik} \sum_{\tau=t-p_i+1}^{t} x_{i\tau} \leq B_k \quad \forall t \in \mathcal{H}, \forall k \in R \tag{11}$$

Since we need a penalty for relaxing the resource constraints, we add slack variables in order to reach an equality for each inequality of (11). Slack variables are noted $s_{tk}, t \in \mathcal{H}, k \in R$, as a non-negative integer for reformulating resource constraints (11) with the quadratic penalties (12).

$$P3(x) = \sum_{t}\sum_{k}(\sum_{i=1}^{n} b_{ik} \sum_{\tau=t-p_i+1}^{t} x_{i\tau} - B_k + s_{tk})^2 \tag{12}$$

Since this reformulation work aims to create a QUBO, the slack variables s_{tk} must be obtained from binaries variables. Here, we can consider the minimum value of s_{tk} as zero and its maximum value as B_k. The related binary expression is given by the equality (13) where y is a binary vector (each y^i takes value in {0,1}) and where the integer function $f(\alpha)$ gives the required maximum power of 2 with α the target integer.

$$s_{tk} = \sum_{i=0}^{i=f(B_k)} 2^i y^i \quad \forall t \in \mathcal{H}, \forall k \in \mathcal{R} \tag{13}$$

We note λ_1, λ_2, and λ_3 the multipliers balancing the penalties $P1$, $P2$, $P3$, respectively. The QUBO of the RCPSP can be formulated by the Objective function

$f_{QUBO}(x)$ with the quadratic constraints (8), (10) and (12) such that $f_{QUBO}(x) = f(x) + \lambda_1 P1(x) + \lambda_2 P2(x) + \lambda_3 P3(x)$ as detailed by the equality (14).

$$\begin{aligned} f_{QUBO}(x) = & \sum_{t} t.x_{n+1,t} \\ & + \lambda_1 \sum_{i=0}^{n+1} (1 - \sum_{t} x_{it})^2 \\ & + \lambda_2 \sum_{(i,j)\in\mathcal{E}} \sum_{t} \sum_{t' \setminus t+p_i > t'} x_{i,t} x_{j,t'} \\ & + \lambda_3 \sum_{t} \sum_{k} (\sum_{i=1}^{n} b_{ik} \sum_{\tau=t-p_i+1}^{t} x_{i\tau} - B_k + s_{tk})^2 \end{aligned} \tag{14}$$

Numerical results

Fig. 10 illustrates the TTT evolution for a range of optimization methods applied to a specific cumulative instance involving six activities (excluding the "dummy" start and end) and two resources. Accompanying this, Fig. 11 presents the network diagram relevant to this instance. The figures are organized into a tripartite panel, demonstrating the progression of the optimization process across different energy quantiles. This progression is depicted from left to right, indicating a transition from higher to lower energy levels.

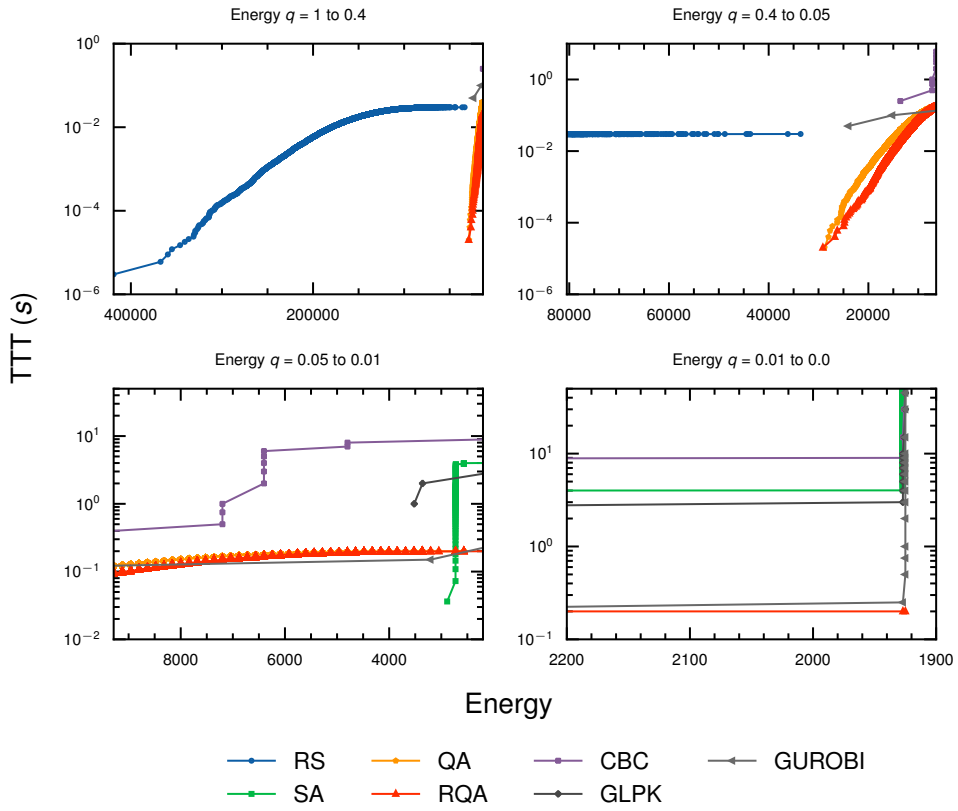


Fig. 10: TTT in seconds for a Cumulative instance (OS = 0.1) of 6 non-dummy activities and 2 resources. The three panels show the evolution for different energy ranges, where energy decreases from left to right. The right most panel shows the energy range for the ground state of this instance.

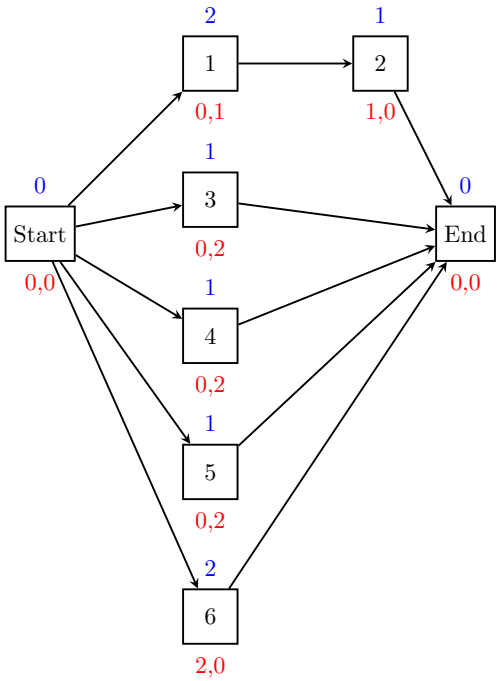
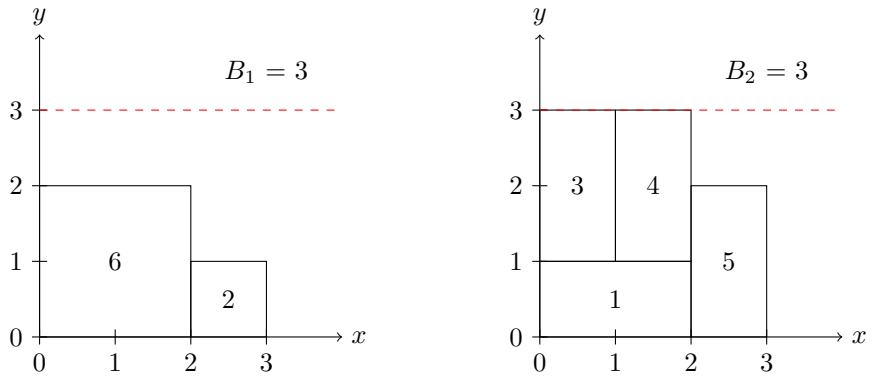


Fig. 11: Network Diagram Cumulative instance (OS = 0.1) with 6 non-dummy activities and 2 resources.

It is noteworthy that QA and RQA exhibit discernible patterns, contrasting with the random sampling behavior observed in the evolution curve of RS. This observation suggests that QA and RQA are not mere random samplers.

In this specific case, RQA demonstrates exceptional performance, outperforming all other optimization methods, including the widely-used commercial solver GUROBI. This superior performance of RQA is particularly noticeable in the third panel of Fig. 10. Here, the distinctive black line marked with 'x' symbols, representing RQA, reaches the ground state energy faster than any other method. Furthermore, Fig. 12 showcases the optimal schedule derived using RQA. This schedule results in an optimal project make span of three days.



(a) Schedule for resource 1.

(b) Schedule for resource 2.

Fig. 12: Optimal schedule obtained from RQA for cumulative instance (OS = 0.1) of 6 non-dummy activities with 2 resources.

Table 3 presents succinct results featuring the mean TTT values for solutions at different energy quantiles (0.9, 0.99 and 0.999). To provide further insight, Fig. 13 visually capture the distinctive mean TTT patterns, highlighting the performance variations among the evaluated optimization methods, for all instance types. Fig. C2, C3 and C4 from Section C offer further insight showcasing the individual TTT performance for the different instance types evaluated in this work (cumulative, middle OS and disjunctive).

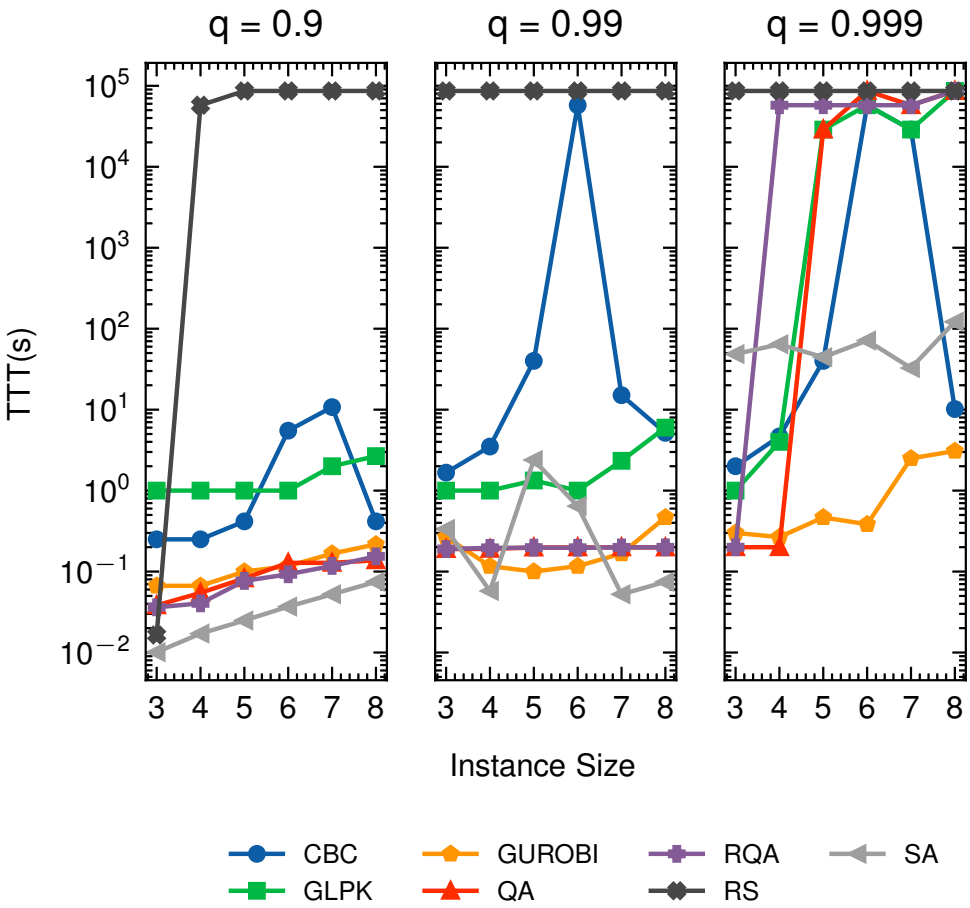


Fig. 13: Mean TTT, reported in seconds for all instance type, OS = 0.1, OS = 0.5 and OS= 0.9.

Table 3: TTT (s) mean results for Disjunctive, Cumulative and Middle OS instances.

Size	Method	$q = 0.9$	$q = 0.95$	$q = 0.99$	$q = 0.999$	$q = 0.9999$
3	CBC	0.25	0.25	1.67	2	2
3	GLPK	1	1	1	1	1
3	GUROBI	0.07	0.07	0.27	0.3	0.3
3	QA	0.04	0.09	0.19	0.2	0.2
3	RQA	0.04	0.08	0.19	0.2	0.2
3	RS	0.02	57600	86400	86400	86400
3	SA	0.01	0.01	0.34	48.60	59.66
4	CBC	0.25	0.25	3.5	4.67	4.67
4	GLPK	1	1	1	4	28801.33
4	GUROBI	0.07	0.07	0.12	0.27	0.5
4	QA	0.05	0.12	0.19	0.20	28800.13
4	RQA	0.04	0.11	0.20	57600.07	86400
4	RS	57600.01	86400	86400	86400	86400
4	SA	0.02	0.02	0.06	64.20	93.92
5	CBC	0.42	0.67	40	40	45
5	GLPK	1	1	1.33	28802	57601
5	GUROBI	0.1	0.1	0.1	0.47	0.75
5	QA	0.08	0.16	0.20	28800.13	86400
5	RQA	0.08	0.16	0.20	57600.07	86400
5	RS	86400	86400	86400	86400	86400
5	SA	0.02	0.02	2.39	44.41	136.76
6	CBC	5.5	28803.17	57603	57603	57615
6	GLPK	1	1	1	57601	57610
6	GUROBI	0.12	0.12	0.12	0.38	2.17
6	QA	0.13	0.17	0.20	86400	86400
6	RQA	0.09	0.15	0.20	57600.07	86400
6	RS	86400	86400	86400	86400	86400
6	SA	0.04	0.04	0.64	71.46	159.07
7	CBC	10.75	15.08	15.08	28810.08	28810.08
7	GLPK	2	2	2.33	28820	86400
7	GUROBI	0.17	0.17	0.17	2.5	16.17
7	QA	0.13	0.18	0.20	57600.07	86400
7	RQA	0.12	0.17	0.20	57600.07	57600.07
7	RS	86400	86400	86400	86400	86400
7	SA	0.05	0.05	0.05	32.65	108.17
8	CBC	0.42	0.42	5.17	10.17	28800.17
8	GLPK	2.67	2.67	6	86400	86400
8	GUROBI	0.22	0.22	0.47	3.08	5
8	QA	0.14	0.18	0.20	86400	86400
8	RQA	0.15	0.18	0.20	86400	86400
8	RS	86400	86400	86400	86400	86400
8	SA	0.07	0.07	0.07	121.81	441.25

The results presented in Table 3 and Fig. C4, C2, and C3 reveal interesting dynamics. It is evident that QA and RQA face challenges in maintaining superiority, especially when tasked with finding ground states in instances of larger sizes. Despite this, in mere fractions of a second, both QA and, notably, RQA demonstrate

their capability to provide high-quality solutions, approaching the proximity of ground states.

Overall, for the high energy quantiles GUROBI exhibits consistent superiority over SA, QA, RQA, and other free solvers. However, this trend does not extend to certain freely available solvers, such as GLPK and CBC, where the situation is different. Notably, in scenarios involving lower energy quantiles, both QA and RQA demonstrate significant advantages over these free solvers. This is particularly evident in the left (q=0.9) and middle (q=0.99) panels of Fig. 13, where QA and RQA outperform their counterparts in reaching lower energy states more efficiently.

Contrary to the findings reported by Carugno et al. [12] in the context of the JSSP, our results indicate that SA surpasses both QA and RQA in terms of performance across most instance sizes and energy quantiles. In agreement with their observations, however, RQA does tend to show superior performance compared to QA, particularly at lower energy quantiles.

Q-score

Adapting the Q-score methodology for QA, as proposed by Van der Schoot et al. [65], we computed the $\beta(n)$ values for various instance sizes and types examined in our study. Analysis of Fig. 14 reveals that the $\beta(n)$ values for both QA and RQA consistently exceed the critical threshold of 0.2. However, it's crucial to remind the reader that this threshold was empirically established based on the solution of instances of the the Max-Cut problem using the Quantum Approximate Optimization Algorithm (QAOA). Therefore, its validity is based solely on these empirical findings, without additional theoretical support.

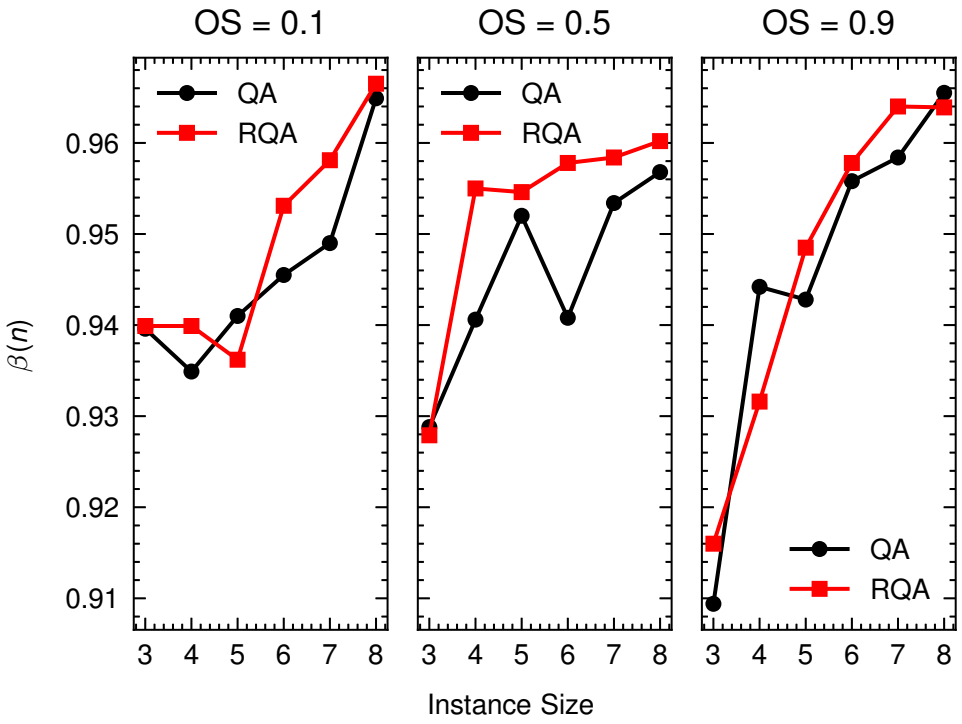


Fig. 14: $\beta(n)$ for all instance size 3,4,5,6,7 and 8 non-dummy activities and all instance types (Cumulative OS = 0.1, Middle OS = 0.5 and Disjunctive OS = 0.9)

We therefore conclude that the estimated maximum instance size of the RCPSP that can be solved using a D-Wave quantum annealer peaks at 7 non-dummy activities. Although the $\beta(n)$ values substantially exceed the critical threshold, we observe a rapid decline in solution quality beyond this activity count. This deterioration is evident when examining the mean 'chain-break' percentage reported by the D-Wave system. For instances with sizes of 8 activities or more, the average chain-break percentage escalates to approximately 74%, indicating a significant breakdown among the logical qubits that were utilized to map the original QUBO graph onto the QPU topology. With such a high percentage of chain breaks, the reliability of the annealer in solving the intended optimization problem becomes questionable, despite the higher $\beta(n)$ values observed.

Anneal time and pausing effects

In the preceding section, our experimental approach was predicated on determining an optimal annealing time for QA. This determination was made empirically, analyzing the performance of QA across a spectrum of annealing times: 1 μs, 5, 10, 20, 50, and 100 μs.

The performance evaluation of QA for each annealing time was determined based on a relative energy difference metric. This metric juxtaposes the minimum energy achieved through QA against the ground state energy for the given instances $\left(\frac{E_{min}-E_0}{E_0}\right)$. Our analysis spanned instance sizes of 3, 4, 5, 6, 7, and 8 non-dummy activities and for instance types (cumulative OS = 0.1, middle OS = 0.5, and disjunctive OS = 0.9). For each configuration we collected 1000 samples.

Insights into the mean behavior of QA across these different annealing times are depicted in Fig. 15. This line graph showcases the relationship between the instance size (x-axis) and the relative energy differences (y-axis), offering a visual representation of the performance variations across instance sizes.

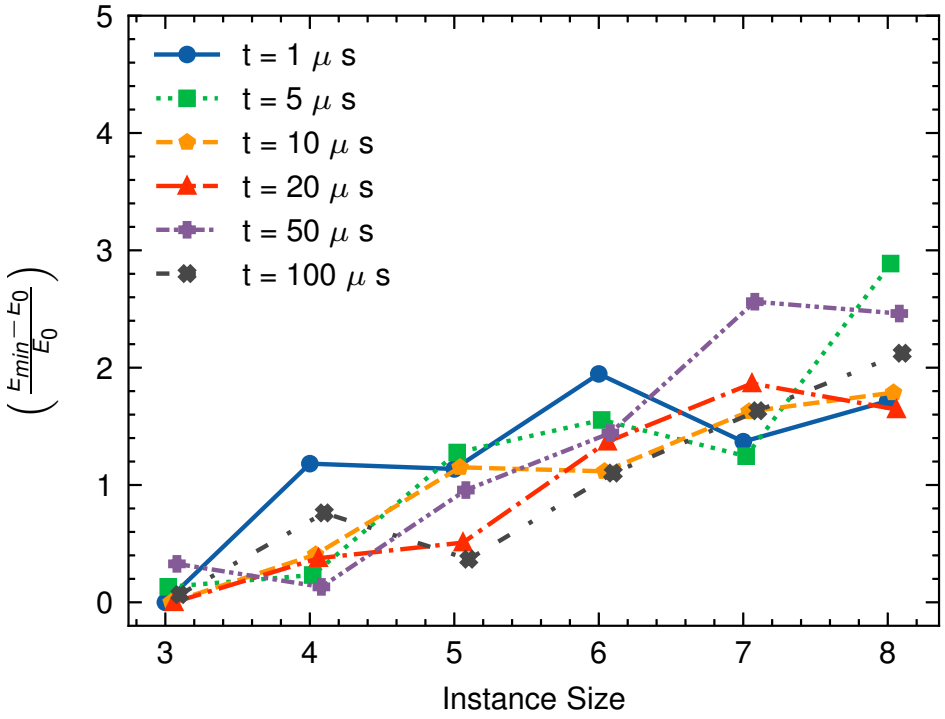


Fig. 15: Mean Effect of annealing time on QA performance, measured as the mean relative deviation from the ground state $\left(\frac{E_{min}-E_0}{E_0}\right)$ for all instance types (OS = 0.1, OS = 0.5 and OS = 0.9).

Complementing this analysis, Fig. 16 presents a holistic view of the mean behavior across all instance sizes and types. Here, the annealing times are mapped along the x-axis, and the corresponding relative energy differences are charted on the y-axis. This graph reveals a noteworthy observation consistent with the findings of King et al.[17] that an annealing time of approximately 20 microseconds emerges as optimal. Beyond this threshold, there is a discernible decline in QA performance.

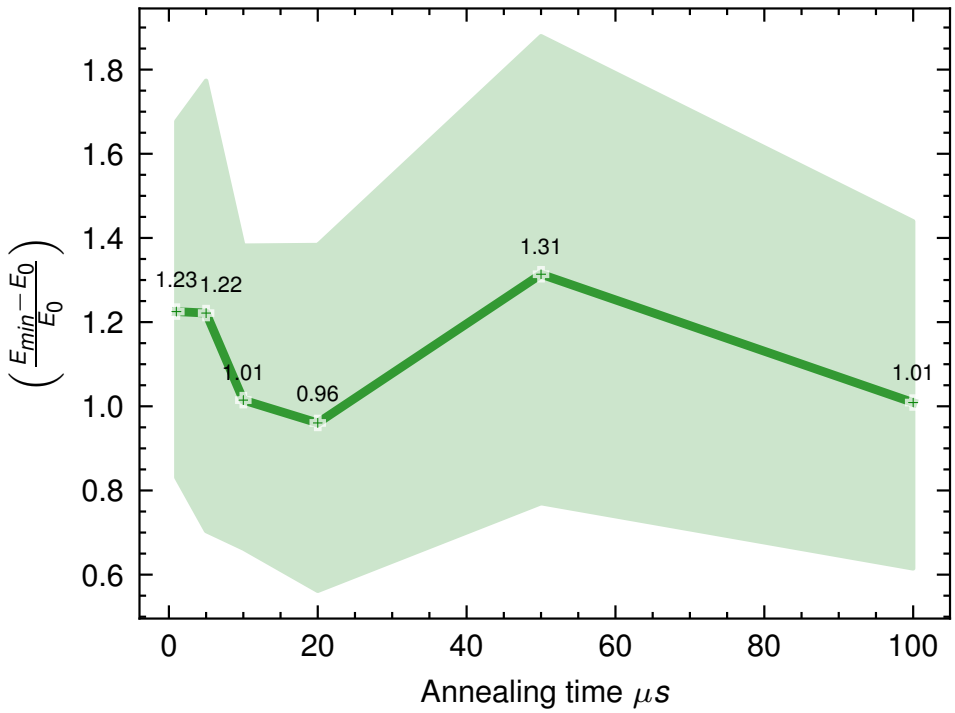


Fig. 16: Overall Effect of annealing time on QA performance, measured as the mean relative deviation from the ground state $\left(\frac{E_{min}-E_0}{E_0}\right)$ for all instance types (OS = 0.1, OS = 0.5 and OS = 0.9) and all sizes.

Given these findings, we elected to conduct the experiments detailed in this work with an annealing time fixed at 20 μs. This decision was guided by the empirical evidence suggesting that this duration strikes an effective balance in optimizing QA performance. Figure D5, D6 and D7 show the effect of annealing times for Cumulative, Middle and Disjunctive instance types.

Annealing pauses

In continuation of our exploration into the multiple parameters involved in QA, we extended the investigation to assess the impact of annealing pauses on QA performance. This analysis aligns with the relative energy difference framework established earlier, comparing the minimum energy achieved via QA to the ground state energy for a particular instance.

Our focus centered on a fixed annealing time of 20 μs (as determined in the previous section), examining the effects of pauses at intervals of 2, 4, 8, 12, 16, and 18 μs please note that these pauses correspond then to 10,20,40,60,80 and 90% of pause of the annealing time. Fig. 7 offers a visual depiction of the annealing schedules, incorporating these pauses.

This investigation was inspired by the findings of Marshal et al. [74], who reported performance enhancements in QA with the inclusion of annealing pauses. Our results, illustrated in Fig. 17, echo this observation. Fig. presents a heat map that encapsulates the QA performance across various instance sizes (3, 4, 5, 6, 7, and 8 non-dummy activities) represented in the x-axis and pause duration's showed in the y-axis as the percentage of pause relative to the total annealing time. Each cell in this heat map represents the average relative energy value derived from 1000 samples across all instance types (cumulative, middle, and disjunctive), providing a comprehensive view of the performance landscape.

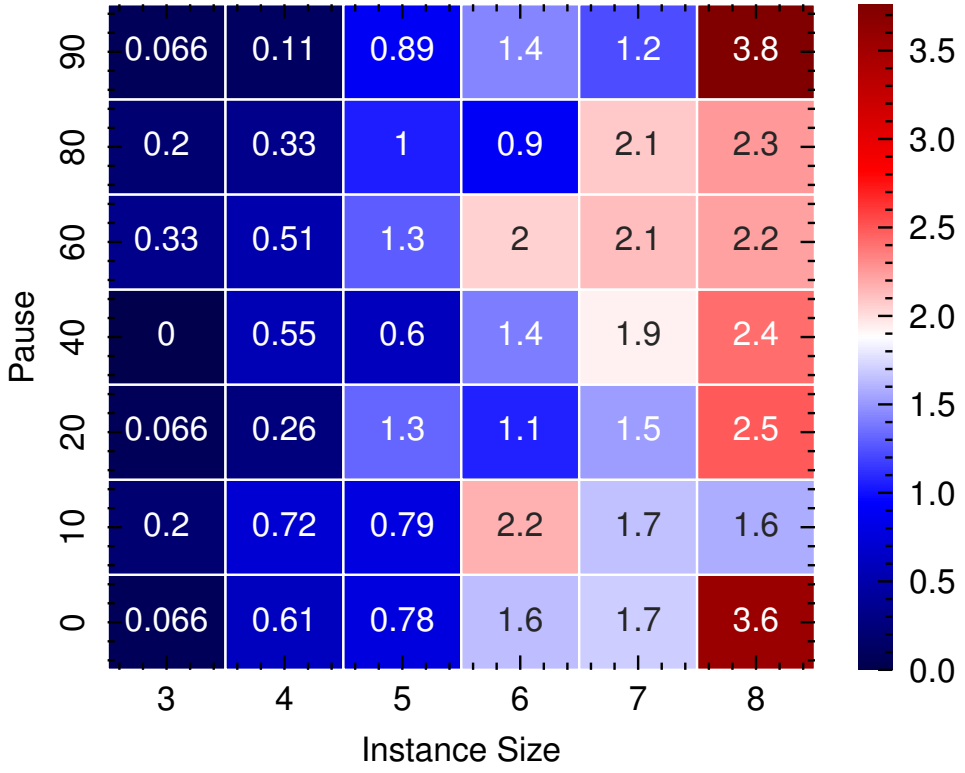


Fig. 17: Heatmap of the effect of pauses included in the annealing schedule vs the relative deviation from the ground state energy $\left(\frac{E_{min}-E_0}{E_0}\right)$ for all instance types (OS = 0.1, OS = 0.5 and OS = 0.9). The y axis shows the percentage of pause in the annealing schedule, as shown in Fig. 7.

Intriguingly, Fig. 18 highlights that annealing pauses constituting 20% to 40% of the overall schedule (equating to 4 μs and 8 μs in this context) yield the most favorable performance. Fig. E8, E9 and E10 show the effect of annealing pauses for Cumulative, Middle and Disjunctive instance types.

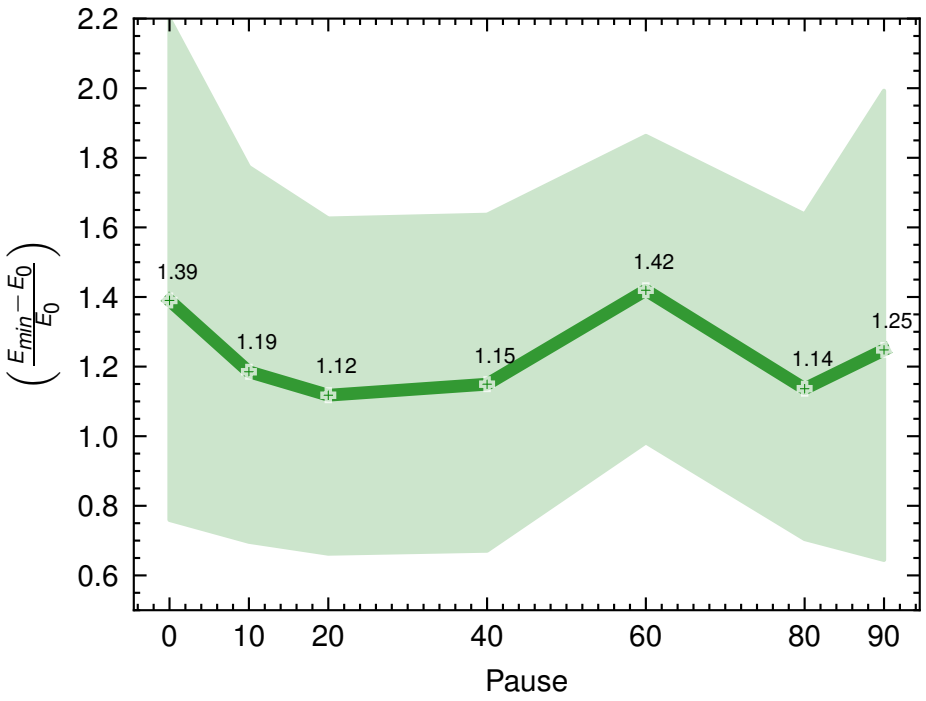


Fig. 18: General effect of pauses included in the annealing schedule vs the relative deviation from the ground state energy $\left(\frac{E_{min}-E_0}{E_0}\right)$ for all instances. The y axis shows the percentage of pause in the annealing schedule, as shown in Fig. 7.

It is important to note, however, that the inherent nature of RQA always incorporates a pause in the annealing schedule. Consequently, the insights gleaned from this analysis predominantly apply to conventional QA rather than RQA.

Discussion

In this work, we have conducted a comprehensive exploration into the practical implementation of the RCPSP using quantum annealing, specifically by leveraging D-Wave's quantum computing technology. Our investigation encompasses a range of aspects, from the basics of quantum annealing to advanced techniques like reverse quantum annealing.

To the best of our knowledge, we are the first to address the RCPSP using a quantum annealer, marking a significant contribution to the integration of quantum computing in Operations Research. Our work underscores the importance of QUBO modeling for solving RCPSP instances on quantum annealing machines. We have conducted a thorough analysis of twelve renowned MILP formulations for RCPSP and

their conversion into QUBO format. This includes identifying the most suitable formulation for quantum annealing, specifically the "PRI69" formulation, and providing the corresponding QUBO model.

It is worthy of highlight that the QUBO model (14) derived on this work in section 3 is flexible enough to be easily adapted to other variations of the RCPSP as in the case of the 1-preemptive resource-constrained project scheduling problem (1_PRCPSP). Adapting the formulation proposed by Ballestín et al. [75], we obtain the QUBO $f(x)^{1\_PRCPSP}_{QUBO}$ (given in F1). The problem does not suffer from the addition of new slack variables. Indeed, the new constraints can be modeled as a quadratic product that must remain zero to be satisfied, similar to precedence constraints.

Other versions of the RCPSP can be based on the QUBO introduced in this study without the need to add many constraints. For example, if resources have periods of unavailability, or if their capacity is reduced (B_k becomes $B_{k,t}$), it's sufficient to treat these periods as distinct activities with fixed variable values as it was shown by Hartmann [76]. This approach can also be applied when the use of resources need to be stopped (e.g., for maintenance operations) without a pre-determined period. In such cases, these periods of unavailability should be considered as activities, but without setting fixed variables. In these scenarios, it's crucial to establish a precedence constraint for each virtual activity to ensure that the identification of the unavailability period is properly included in the optimization process. However, this modification will require the inclusion of additional slack variables associated with the resource constraint.

Continuing with the contributions of this work several instances from three distinct categories, as defined by Baptiste and Le Pape's protocol, have been solved using the RCPSP QUBO (14) on a quantum annealer. Strategies such as developing a reverse quantum annealing schedule were employed. The results were then compared with those obtained from a classical solver. For this comparison, we conducted a detailed exploration of multiple metrics available for evaluating the performance of quantum annealing. Ultimately, we utilized the TTT and the q-score for our evaluation. Additionally, we determined the maximum instance size achievable with a machine equipped with 5k qubits. Another key discovery from our research underscores the potential benefits of quantum annealing, especially in scenarios with constrained time limits, such as those encountered in online Operations Research problems. In addition to our primary research focus, we conducted an extensive and meticulous analysis to understand the impact of annealing times and the role of pauses within annealing schedules on the performance of QA.

While our research offers valuable insights, it's crucial to recognize its limitations. These include using problem instances of limited size, excluding other promising quantum computing techniques like QAOA [70] due to hardware constraints, and the dependence on the "minor-miner" heuristic for the embedding procedure.

In alignment with the work of Venturelli et al. [11] and Carugno et al [12]. on the JSSP, we recognize the potential for significant reduction in time horizons for the RCPSP. This reduction can be achieved by pre-establishing upper bounds, derived from heuristics that are calculable in polynomial time [77], which can considerably minimize the number of required variables required by time-index formulations like

"PRI69". However, in this study, our focus was centered on examining the intrinsic performance of QA and RQA without incorporating these heuristic techniques.

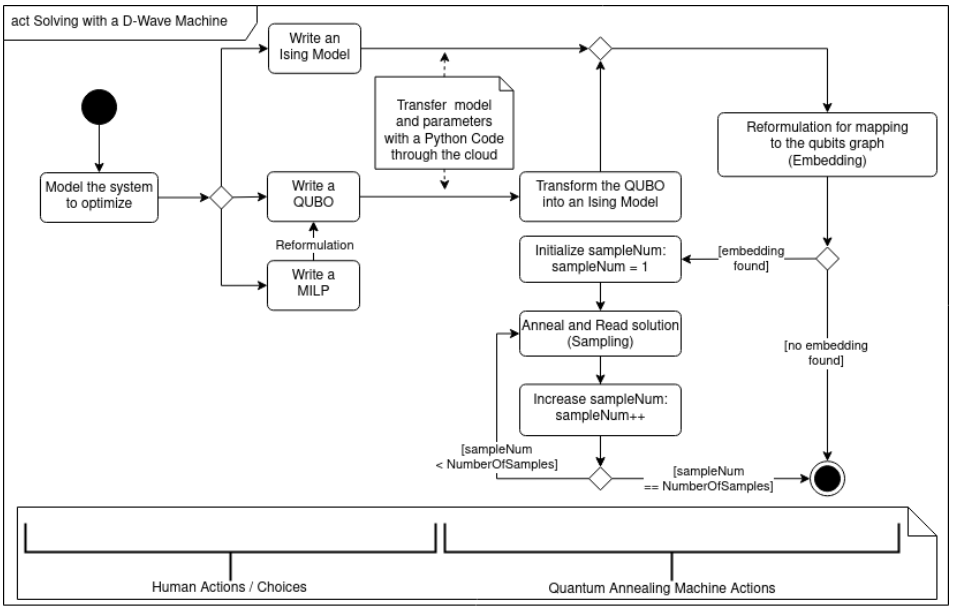
These aspects warrant consideration in future research. Additionally, investigating alternative quantum computing technologies, such as neutral atoms technology by Pasqal[78], is promising and deserves further exploration.

In conclusion, despite the acknowledged limitations, this work serves as a pioneering effort in applying quantum annealing to the RCPSP. We hope that our findings and methodologies will act as a catalyst for future benchmarks, fostering advancements towards practical applications in Operations Research.

Appendix A UML Activity diagram: resolution process with a quantum annealing solver

The UML activity diagram of the Fig. A1 summarizes the process of solving an optimization problem such as RCPSP by a quantum machine implementing QA. The activities are divided into two sets: on the left, human activities corresponding to problem modeling, and on the right, those automatically executed by the machine. The machine takes as input either an Ising model or a QUBO model. If the user only has a MILP, as is the case in this work for 12 different formulations, they must formulate it either as an Ising model or a QUBO model. If the latter is provided, the machine automatically converts it to Ising, as the two models are isomorphic. In addition, the user also provides weights corresponding to the penalties of each series of relaxed constraints (i.e., the multipliers λ_i for each constraint i).

Fig. A1: UML Activity Diagram: Transitioning from the mathematical model to the quantum annealing solution process. Each activity belongs either to the actions to be performed by the user or to the actions to be executed by the machine. The user provides a QUBO or an ISING model to the machine, and it replicates the QA as many times as requested by the user, resulting in as many solutions obtained at the end of the process.



Once the machine has an Ising model to process, the corresponding non-directed graph must be adapted to the machine's topology, that is, the qubits network. Thus,

the Ising model is transformed into another non-directed graph suited to the topology. The resulting graph is generally less dense to match the machine's connectivity, using more qubits for this purpose. This process, called "embedding," solves another optimization problem that the machine resolves heuristically. At the end of this process, if the machine does not find an "embedding," the experiment stops. Otherwise, the QA process is applied to the mapped graph.

The number of QA replications is determined by the number of *anneals* and *reads*, collectively referred to as *samples*. Users must specify two key parameters for each QA experiment conducted on a D-Wave machine: the number of *samples* and the time available for each anneal. These parameters are crucial and significantly influence the outcome of the experiment. Given that each experiment is typically restricted to a duration of about one second, the total time available for quantum machine access is primarily allocated between these two parameters. The optimal balance between them is usually determined empirically and varies depending on the specific optimization problem being addressed.

As output, the user obtains as many solutions as there were anneals executed. Even if all obtained solutions are feasible in the sense that the variable type is respected (binary) and no constraint integrates the QUBO and Ising models, the feasibility of the solutions must be verified through the constraints of the initial MILP model, an operation that can be done in polynomial time due to the complexity of the RCPSP.

Appendix B Evaluation of MILP Formulations for RCPSP

Table B1: Evaluation of MILP Formulations for RCPSP.

Instance	Type	Family	MILP	n MILP	n (QUBO)	S (QUBO)	ρ
10_H	Disjunctive	Time Index	PRI69	570	1136	566	0.1566
10_H	Disjunctive	Sequence	TV93	110	2321	2161	0.0073
10_H	Disjunctive	Sequence	ART03	310	3316	2356	0.0094
10_H	Disjunctive	Event	ART13	230	5559	5219	0.0056
10_H	Disjunctive	Event	KONE11	111	4824	4658	0.0070
10_L	Cumulative	Time Index	PRI69	380	830	450	0.1740
10_L	Cumulative	Sequence	TV93	110	2328	2168	0.0073
10_L	Cumulative	Sequence	ART03	310	3329	2369	0.0093
10_L	Cumulative	Event	ART13	230	4724	4384	0.0068
10_L	Cumulative	Event	KONE11	111	4235	4069	0.0080
15_H	Disjunctive	Time Index	PRI69	990	1758	768	0.1443
15_H	Disjunctive	Sequence	TV93	240	7681	7351	0.0022
15_H	Disjunctive	Sequence	ART03	690	9859	7729	0.0035
15_H	Disjunctive	Event	ART13	495	17678	17003	0.0021
15_H	Disjunctive	Event	KONE11	241	15360	15023	0.0026
15_L	Cumulative	Time Index	PRI69	1035	1851	816	0.1526
15_L	Cumulative	Sequence	TV93	240	7699	7369	0.0022
15_L	Cumulative	Sequence	ART03	690	9869	7739	0.0035
15_L	Cumulative	Event	ART13	495	16553	15878	0.0023
15_L	Cumulative	Event	KONE11	241	15036	14699	0.0027
20_H	Disjunctive	Time Index	PRI69	1780	2840	1060	0.1318
20_H	Disjunctive	Sequence	TV93	420	17645	17105	0.0009
20_H	Disjunctive	Sequence	ART03	1220	21435	17695	0.0017
20_H	Disjunctive	Event	ART13	860	39033	37933	0.0010
20_H	Disjunctive	Event	KONE11	421	35028	34481	0.0012
20_L	Cumulative	Time Index	PRI69	1740	2868	1128	0.1427
20_L	Cumulative	Sequence	TV93	420	17715	17175	0.0009
20_L	Cumulative	Sequence	ART03	1220	21443	17703	0.0017
20_L	Cumulative	Event	ART13	860	36653	35553	0.0011
20_L	Cumulative	Event	KONE11	421	34426	33879	0.0012

n MILP = number of variables original problem, n (QUBO) = number of qubits, S (QUBO) = number of Slack qubits, ρ = Density of the QUBO

Table B2: Evaluation of MILP Formulations for RCPSP

Instance	Type	MILP	n MILP	n (QUBO)	S (QUBO)	ρ
10_H	Disjunctive	PRI69	570	1136	566	0.157
10_H	Disjunctive	CHRI87	570	3432	2862	0.041
10_H	Disjunctive	SOUWO97	570	2316	1746	0.009
10_H	Disjunctive	KLEIN/KAPL98	570	2803	2290	0.009
10_H	Disjunctive	KLEIN00_1	570	3673	3103	0.010
10_H	Disjunctive	KLEIN00_2	1140	3447	2307	0.006
10_H	Disjunctive	BIACAR13	1710	8016	2886	0.016
10_H	Disjunctive	MINGZ98	684	1187	503	0.099
10_L	Cumulative	PRI69	380	830	450	0.174
10_L	Cumulative	CHRI87	380	2932	2552	0.034
10_L	Cumulative	SOUWO97	380	1542	1162	0.010
10_L	Cumulative	KLEIN/KAPL98	380	1354	1012	0.023
10_L	Cumulative	KLEIN00_1	380	2735	2355	0.014
10_L	Cumulative	KLEIN00_2	760	2288	1528	0.013
10_L	Cumulative	BIACAR13	1140	5342	1922	0.023
10_L	Cumulative	MINGZ98	646	1611	965	0.065
15_H	Disjunctive	PRI69	990	1758	768	0.1443
15_H	Disjunctive	CHRI87	990	7694	6704	0.022
15_H	Disjunctive	SOUWO97	990	4065	3075	0.006
15_H	Disjunctive	KLEIN/KAPL98	990	6726	5802	0.009
15_H	Disjunctive	KLEIN00_1	990	6199	5209	0.008
15_H	Disjunctive	KLEIN00_2	1980	6027	4047	0.004
15_H	Disjunctive	BIACAR13	2970	13965	5055	0.013
15_H	Disjunctive	MINGZ98	1320	2796	1476	0.046
15_L	Cumulative	PRI69	1035	1851	816	0.153
15_L	Cumulative	CHRI87	1035	8913	7878	0.020
15_L	Cumulative	SOUWO97	1035	4389	3354	0.006
15_L	Cumulative	KLEIN/KAPL98	1035	5029	4063	0.011
15_L	Cumulative	KLEIN00_1	1035	9372	8337	0.007
15_L	Cumulative	KLEIN00_2	2070	6441	4371	0.008
15_L	Cumulative	BIACAR13	3105	14739	5424	0.012
15_L	Cumulative	MINGZ98	2001	5054	3053	0.055
20_H	Disjunctive	PRI69	1780	2840	1060	0.132
20_H	Disjunctive	CHRI87	1780	15945	14165	0.014
20_H	Disjunctive	SOUWO97	1780	7177	5397	0.004
20_H	Disjunctive	KLEIN/KAPL98	1780	15366	13675	0.005
20_H	Disjunctive	KLEIN00_1	1780	13492	11712	0.005
20_H	Disjunctive	KLEIN00_2	3560	10709	7149	0.003
20_H	Disjunctive	BIACAR13	5340	24977	8957	0.010
20_H	Disjunctive	MINGZ98	2581	5570	2989	0.030
20_L	Cumulative	PRI69	1740	2868	1128	0.143
20_L	Cumulative	CHRI87	1740	18997	17257	0.012
20_L	Cumulative	SOUWO97	1740	7624	5884	0.004
20_L	Cumulative	KLEIN/KAPL98	1740	11391	9738	0.007
20_L	Cumulative	KLEIN00_1	1740	19217	17477	0.004
20_L	Cumulative	KLEIN00_2	3480	11076	7596	0.006
20_L	Cumulative	BIACAR13	5220	25024	9364	0.009
20_L	Cumulative	MINGZ98	5307	12620	7313	0.089

n MILP = number of variables original problem, n (QUBO) = number of qubits, S (QUBO) = number of Slack qubits, ρ = Density of the QUBO

Appendix C TTT Supplementary plots

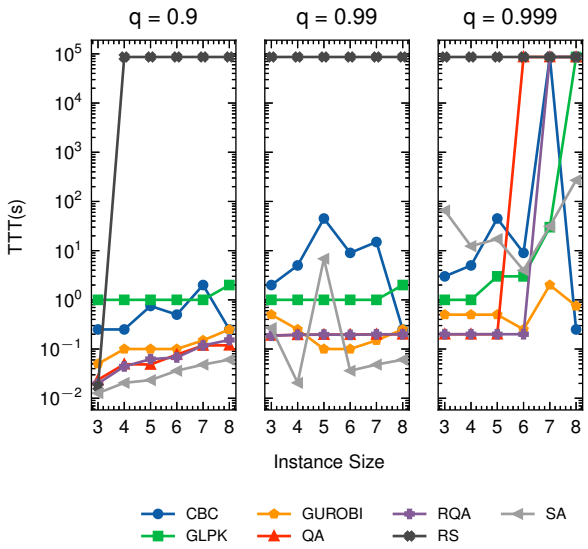


Fig. C2: Time-to-Target, reported in seconds for Cumulative instances with OS = 0.1.

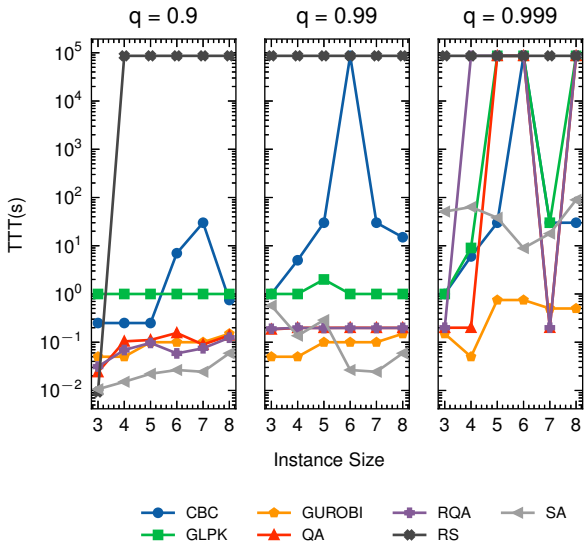


Fig. C3: Time-to-Target, reported in seconds for Middle instances with OS = 0.5.

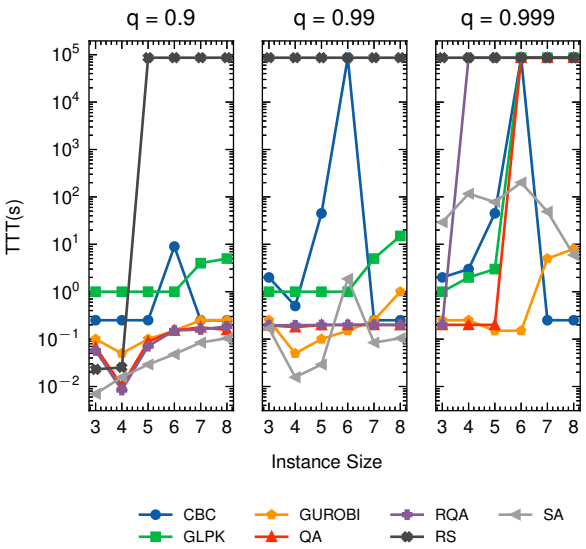


Fig. C4: Time-to-Target, reported in seconds for Disjunctive instances with OS = 0.9.

Appendix D Annealing time Supplementary plots

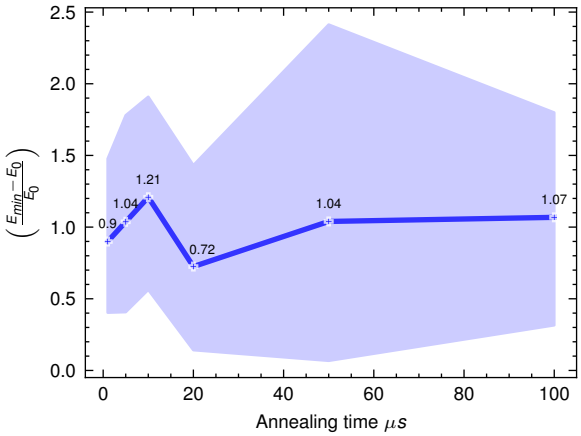


Fig. D5: Overall Effect of annealing time on QA performance, measured as the mean relative deviation from the ground state $\left(\frac{E_{min}-E_0}{E_0}\right)$ for cumulative instances with a OS = 0.1 and all sizes (3,4,5,6,7 and 8 non-dummy activities).

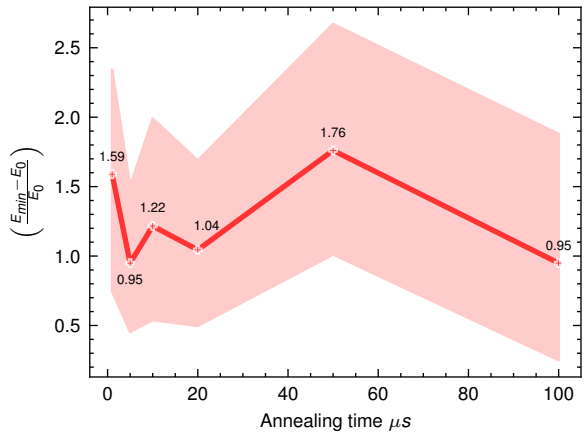


Fig. D6: Overall Effect of annealing time on QA performance, measured as the mean relative deviation from the ground state $\left(\frac{E_{min}-E_0}{E_0}\right)$ for Middle instances with a OS = 0.5 and all sizes (3,4,5,6,7 and 8 non-dummy activities).

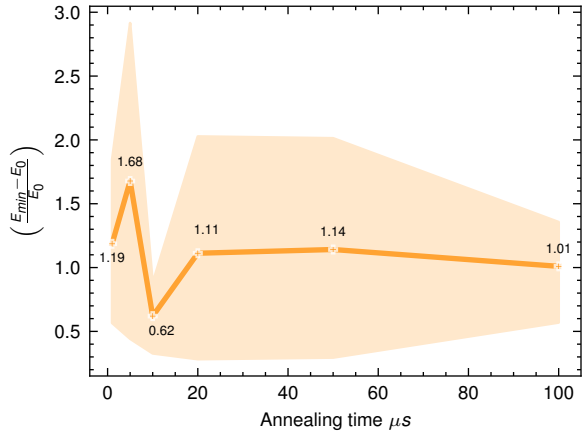


Fig. D7: Overall Effect of annealing time on QA performance, measured as the mean relative deviation from the ground state $\left(\frac{E_{min}-E_0}{E_0}\right)$ for Disjunctive instances with a OS = 0.9 and all sizes (3,4,5,6,7 and 8 non-dummy activities).

Appendix E Annealing pauses supplementary plots

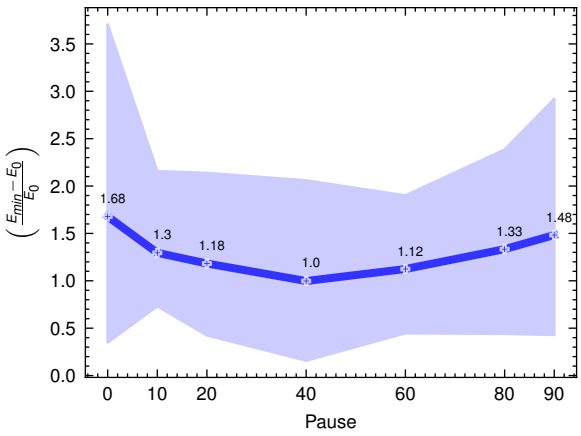


Fig. E8: General effect of pauses included in the annealing schedule vs the relative deviation from the ground state energy $\left(\frac{E_{min}-E_0}{E_0}\right)$ for Cumulative instances with OS = 0.1. The y axis shows the percentage of pause in the annealing schedule, as shown in Fig. 7.

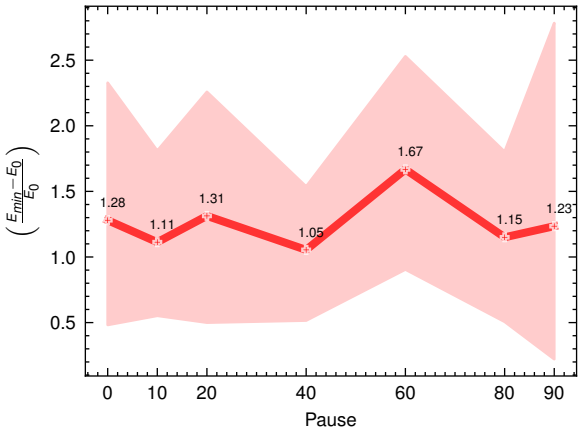


Fig. E9: General effect of pauses included in the annealing schedule vs the relative deviation from the ground state energy $\left(\frac{E_{min}-E_0}{E_0}\right)$ for Middle instances with OS = 0.5. The y axis shows the percentage of pause in the annealing schedule, as shown in Fig.7.

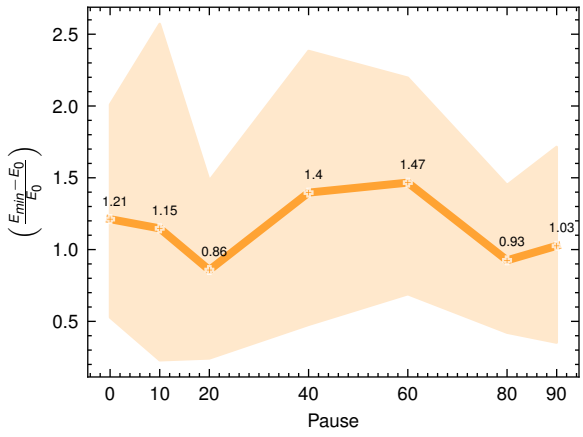


Fig. E10: General effect of pauses included in the annealing schedule vs the relative deviation from the ground state energy $\left(\frac{E_{min}-E_0}{E_0}\right)$ for Disjunctive instances with OS = 0.9. The y axis shows the percentage of pause in the annealing schedule, as shown in Fig.7.

Appendix F QUBO of the 1_PRCPSP

$$
\begin{aligned}
f(x)^{1\_PRCPSP}_{QUBO} = & \sum_{t} t.x_{n+1,t} \\
& + \lambda_1^A \sum_{i=0}^{n+1} (1 - \sum_{t} x_{iAt})^2 \\
& + \lambda_1^B \sum_{i=0}^{n+1} (1 - \sum_{t} x_{iBt})^2 \\
& + \lambda_2^{BB} \sum_{(i,j)\in\mathcal{E}} \sum_{t} \sum_{t'\setminus t+p_{iB}>t'} x_{iB,t} x_{jB,t'} \\
& + \lambda_2^{AB} \sum_{i\in\mathcal{A}} \sum_{t} \sum_{t'\setminus t+p_{iA}>t'} x_{iA,t} x_{iB,t'} \\
& + \lambda_3^C \sum_{t} \sum_{k} (\sum_{i=1}^{n} b_{ik} \sum_{\tau=t-p_{iA}+1}^{t} x_{iA\tau} + x_{iB\tau} - B_k + s_{tk})^2
\end{aligned}
\tag{F1}
$$

We adapt the notation of the multipliers according to the activity and the type of constraints (e.g. "λ_1^B"). Each original variable $x_{i,t}$ is now divided into two parts $x_{iA,t}$ and $x_{iB,t}$ that must comply with:

$$
\sum_{t\in\mathcal{H}} t.x_{iA,t} \leq \sum_{t\in\mathcal{H}} t.x_{iB,t} + p_A \qquad \forall i \in \mathcal{A}
$$

References

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

[2] Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). Ieee

[3] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)

[4] Creemers, S., Perez, L.: Discrete optimization: A quantum revolution (part i). Available at SSRN 4198077 (2023) https://doi.org/10.2139/ssrn.4198077

[5] Creemers, S., Perez, L.: Discrete optimization: A quantum revolution (part ii). Available at SSRN 4494506 (2023) https://doi.org/10.2139/ssrn.4494506

[6] MIT Technology Review, M.I.o.T.: IBM wants to build a 100,000-qubit quantum computer. https://www.technologyreview.com/2023/05/25/1073606/ibm-wants-to-build-a-100000-qubit-quantum-computer/#:~:text=Late%20last%20year%2C%20IBM%20took,blocks%20of%20quantum%20information%20processing. 11-2023 (2023)

[7] Ushijima-Mwesigwa, H., Negre, C.F., Mniszewski, S.M.: Graph partitioning using quantum annealing on the d-wave system. In: Proceedings of the Second International Workshop on Post Moores Era Supercomputing, pp. 22–29 (2017)

[8] Bauckhage, C., Piatkowski, N., Sifa, R., Hecker, D., Wrobel, S.: A qubo formulation of the k-medoids problem. In: LWDA, pp. 54–63 (2019)

[9] Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., Parney, B.: Traffic flow optimization using a quantum annealer. Frontiers in ICT **4**, 29 (2017)

[10] Stollenwerk, T., O'Gorman, B., Venturelli, D., Mandra, S., Rodionova, O., Ng, H., Sridhar, B., Rieffel, E.G., Biswas, R.: Quantum annealing applied to de-conflicting optimal trajectories for air traffic management. IEEE transactions on intelligent transportation systems **21**(1), 285–297 (2019)

[11] Venturelli, D., Marchand, D.J., Rojo, G.: Quantum annealing implementation of job-shop scheduling. arXiv preprint arXiv:1506.08479 (2015)

[12] Carugno, C., Ferrari Dacrema, M., Cremonesi, P.: Evaluating the job shop scheduling problem on a d-wave quantum annealer. Scientific Reports **12**(1), 6539 (2022)

[13] Venturelli, D., Mandrà, S., Knysh, S., O'Gorman, B., Biswas, R., Smelyanskiy, V.:

Quantum optimization of fully connected spin glasses. Physical Review X **5**(3), 031040 (2015)

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

[15] Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., De Prado, M.L.: Solving the optimal trading trajectory problem using a quantum annealer. In: Proceedings of the 8th Workshop on High Performance Computational Finance, pp. 1–7 (2015)

[16] King, A.D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., Reis, M., Lanting, T., Harris, R., Altomare, F., *et al.*: Observation of topological phenomena in a programmable lattice of 1,800 qubits. Nature **560**(7719), 456–460 (2018)

[17] King, J., Yarkoni, S., Nevisi, M.M., Hilton, J.P., McGeoch, C.C.: Benchmarking a quantum annealing processor with the time-to-target metric. arXiv preprint arXiv:1508.05087 (2015)

[18] Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Physical Review E **58**(5), 5355 (1998)

[19] Albash, T., Lidar, D.A.: Adiabatic quantum computation. Reviews of Modern Physics **90**(1), 015002 (2018)

[20] Kirkpatrick, S., Gelatt Jr, C.D., Vecchi, M.P.: Optimization by simulated annealing. science **220**(4598), 671–680 (1983)

[21] Harris, R., Johansson, J., Berkley, A., Johnson, M., Lanting, T., Han, S., Bunyk, P., Ladizinsky, E., Oh, T., Perminov, I., *et al.*: Experimental demonstration of a robust and scalable flux qubit. Physical Review B **81**(13), 134510 (2010)

[22] Santoro, G.E., Martonák, R., Tosatti, E., Car, R.: Theory of quantum annealing of an ising spin glass. Science **295**(5564), 2427–2430 (2002)

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

[24] Vert, D., Sirdey, R., Louise, S.: Benchmarking quantum annealing against "hard" instances of the bipartite matching problem. SN Computer Science **2**, 1–12 (2021)

[25] Born, M., Fock, V.: Beweis des adiabatensatzes. Zeitschrift für Physik **51**, 165–180 (1928) https://doi.org/10.1007/BF01343193

[26] Kato, T.: On the adiabatic theorem of quantum mechanics. Journal of the

Physical Society of Japan **5**(6), 435–439 (1950)

[27] Schrödinger, E.: An undulatory theory of the mechanics of atoms and molecules. Physical review **28**(6), 1049 (1926)

[28] Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D.: A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. Science **292**(5516), 472–475 (2001)

[29] Josephson, B.D.: Possible new effects in superconductive tunnelling. Physics letters **1**(7), 251–253 (1962)

[30] Kwon, S., Tomonaga, A., Lakshmi Bhai, G., Devitt, S.J., Tsai, J.-S.: Gate-based superconducting quantum computing. Journal of Applied Physics **129**(4), 041102 (2021)

[31] Krantz, P., Kjaergaard, M., Yan, F., Orlando, T.P., Gustavsson, S., Oliver, W.D.: A quantum engineer's guide to superconducting qubits. Applied physics reviews **6**(2), 021318 (2019)

[32] Bernal, D.E., Booth, K.E., Dridi, R., Alghassi, H., Tayur, S., Venturelli, D.: Integer programming techniques for minor-embedding in quantum annealers. In: Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 17th International Conference, CPAIOR 2020, Vienna, Austria, September 21–24, 2020, Proceedings 17, pp. 112–129 (2020). Springer

[33] Okada, S., Ohzeki, M., Terabe, M., Taguchi, S.: Improving solutions by embedding larger subproblems in a d-wave quantum annealer. Scientific reports **9**(1), 2098 (2019)

[34] Abbott, A.A., Calude, C.S., Dinneen, M.J., Hua, R.: A hybrid quantum-classical paradigm to mitigate embedding costs in quantum annealing. International Journal of Quantum Information **17**(05), 1950042 (2019)

[35] Serra, T., Huang, T., Raghunathan, A.U., Bergman, D.: Template-based minor embedding for adiabatic quantum optimization. INFORMS Journal on Computing **34**(1), 427–439 (2022)

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

[37] Venturelli, D., Kondratyev, A.: Reverse quantum annealing approach to portfolio optimization problems. Quantum Machine Intelligence **1**(1-2), 17–30 (2019)

[38] Ikeda, K., Nakamura, Y., Humble, T.S.: Application of quantum annealing to nurse scheduling problem. Scientific reports **9**(1), 12837 (2019)

[39] Rocutto, L., Destri, C., Prati, E.: Quantum semantic learning by reverse annealing of an adiabatic quantum computer. Advanced Quantum Technologies **4**(2), 2000133 (2021)

[40] Golden, J., O'Malley, D.: Reverse annealing for nonnegative/binary matrix factorization. Plos one **16**(1), 0244026 (2021)

[41] Blazewicz, J., Lenstra, J.K., Kan, A.R.: Scheduling subject to resource constraints: classification and complexity. Discrete applied mathematics **5**(1), 11–24 (1983)

[42] Sahni, S., Cho, Y.: Complexity of scheduling shops with no wait in process. Mathematics of Operations Research **4**(4), 448–457 (1979)

[43] Gromicho, J.A., Van Hoorn, J.J., Saldanha-da-Gama, F., Timmer, G.T.: Solving the job-shop scheduling problem optimally by dynamic programming. Computers & Operations Research **39**(12), 2968–2977 (2012)

[44] Garey, M.R.: Computers and intractability: A guide to the theory of np-completeness, freeman. Fundamental (1997)

[45] Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Oliveira, F.: A parallelizable augmented lagrangian method applied to large-scale non-convex-constrained optimization problems. Mathematical Programming **175**, 503–536 (2019)

[46] Castro, J., Nasini, S., Saldanha-da-Gama, F.: A cutting-plane approach for large-scale capacitated multi-period facility location using a specialized interior-point method. Mathematical programming **163**(1-2), 411–444 (2017)

[47] Castro, J., Nasini, S.: A specialized interior-point algorithm for huge minimum convex cost flows in bipartite networks. European Journal of Operational Research **290**(3), 857–869 (2021)

[48] Damay, J., Quilliot, A., Sanlaville, E.: Linear programming based algorithms for preemptive and non-preemptive rcpsp. European Journal of Operational Research **182**(3), 1012–1022 (2007)

[49] Brucker, P., Knust, S.: A linear programming and constraint propagation-based lower bound for the rcpsp. European Journal of Operational Research **127**(2), 355–362 (2000)

[50] Pritsker, A.A.B., Waiters, L.J., Wolfe, P.M.: Multiproject scheduling with limited resources: A zero-one programming approach. Management science **16**(1), 93–108 (1969)

[51] Christofides, N., Alvarez-Valdés, R., Tamarit, J.M.: Project scheduling with

resource constraints: A branch and bound approach. European journal of operational research **29**(3), 262–273 (1987)

[52] De Souza, C., Wolsey, L.: Scheduling projects with labour constraints. Relatório Técnico IC-P7-22. Instituto de Computação, Universidade Estadual de Campinas (1997)

[53] Mingozzi, A., Maniezzo, V., Ricciardelli, S., Bianco, L.: An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. Management science **44**(5), 714–729 (1998)

[54] Klein, R.: Scheduling of Resource-constrained Projects vol. 10. Springer, ??? (2000)

[55] Kaplan, L.A.: Resource-constrained Project Scheduling with Preemption of Jobs. University of Michigan, ??? (1988)

[56] Bianco, L., Caramia, M.: A new formulation for the project scheduling problem under limited resources. Flexible Services and Manufacturing Journal **25**, 6–24 (2013)

[57] Olaguibel, R.A.-V., Goerlich, J.T.: The project scheduling polyhedron: Dimension, facets and lifting theorems. European Journal of Operational Research **67**(2), 204–220 (1993)

[58] Artigues, C., Michelon, P., Reusser, S.: Insertion techniques for static and dynamic resource-constrained project scheduling. European Journal of Operational Research **149**(2), 249–267 (2003)

[59] Koné, O., Artigues, C., Lopez, P., Mongeau, M.: Event-based milp models for resource-constrained project scheduling problems. Computers & Operations Research **38**(1), 3–13 (2011)

[60] Artigues, C., Brucker, P., Knust, S., Koné, O., Lopez, P., Mongeau, M.: A note on "event-based milp models for resource-constrained project scheduling problems". Computers & Operations Research **40**(4), 1060–1063 (2013)

[61] Koné, O., Artigues, C., Lopez, P., Mongeau, M.: Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. Flexible Services and Manufacturing Journal **25**, 25–47 (2013)

[62] Kolisch, R., Sprecher, A.: Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. European journal of operational research **96**(1), 205–216 (1997)

[63] Kolisch, R., Sprecher, A.: PSPLIB. https://www.om-db.wi.tum.de/psplib/data.

html. 11-2023 (1996)

[64] Coelho, J., Vanhoucke, M.: Going to the core of hard resource-constrained project scheduling instances. Computers & Operations Research **121**, 104976 (2020)

[65] Schoot, W., Leermakers, D., Wezeman, R., Neumann, N., Phillipson, F.: Evaluating the q-score of quantum annealers. In: 2022 IEEE International Conference on Quantum Software (QSW), pp. 9–16 (2022). IEEE

[66] Demeulemeester, E., Vanhoucke, M., Herroelen, W.: Rangen. https://www.projectmanagement.ugent.be/research/data/RanGen. 08-2023 (2003)

[67] Baptiste, P., Pape, C.L.: Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. Constraints **5**, 119–139 (2000)

[68] Lidar, D.A., Wecker, D., Martinis, J.M., Job, J., Troyer, M., Isakov, S.V., Boixo, S., Wang, Z., et al.: Defining and detecting quantum speedup. Science (2014)

[69] Martiel, S., Ayral, T., Allouche, C.: Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way. IEEE Transactions on Quantum Engineering **2**, 1–11 (2021)

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

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

[72] Verma, A., Lewis, M.: Penalty and partitioning techniques to improve performance of qubo solvers. Discrete Optimization **44**, 100594 (2022)

[73] Demeulemeester, E.L., Herroelen, W.S.: Project Scheduling: a Research Handbook vol. 49. Springer, ??? (2006)

[74] Marshall, J., Venturelli, D., Hen, I., Rieffel, E.G.: Power of pausing: Advancing understanding of thermalization in experimental quantum annealers. Physical Review Applied **11**(4), 044083 (2019)

[75] Ballestín, F., Valls, V., Quintanilla, S.: Pre-emption in resource-constrained project scheduling. European Journal of Operational Research **189**(3), 1136–1152 (2008)

[76] Hartmann, S.: Project scheduling with resource capacities and requests varying with time: a case study. Springer (2013)

[77] Kolisch, R., Hartmann, S.: Experimental evaluation of heuristics for the resource-constrained project scheduling problem: An update. European Journal of

Operational Research (2005)

[78] Sadeghi, H.: Quantum computing with neutral atoms with applications in energy. In: EAGE Seventh High Performance Computing Workshop, vol. 2023, pp. 1–3 (2023). European Association of Geoscientists & Engineers