Addressing Machine Unavailability in Job Shop Scheduling: A Quantum Computing Approach

Riad Aggoune, Samuel Deleplanque

▶ **To cite this version:**

Riad Aggoune, Samuel Deleplanque. Addressing Machine Unavailability in Job Shop Scheduling: A Quantum Computing Approach. 15th Metaheuristics International Conference MIC 2024, Jun 2024, Lorient, France. hal-04532209

HAL Id: hal-04532209
https://hal.science/hal-04532209

Submitted on 4 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Addressing Machine Unavailability in Job Shop Scheduling: A Quantum Computing Approach

Riad Aggoune[1][0000−0003−2622−7097] and Samuel Deleplanque[2][0000−0003−4119−6006]

[1] ITIS department, Luxembourg Institute of Science and Technology, Luxembourg
riad.aggoune@list.lu
[2] CNRS, Centrale Lille, JUNIA, Univ. Lille, Univ. Valenciennes, UMR 8520 IEMN, 41 boulevard Vauban, Lille Cedex 59046, France.
samuel.deleplanque@junia.com

Abstract. We consider solving the Job Shop Scheduling Problem (JSSP) with machine unavailability constraints using an analog quantum machine and running the quantum annealing metaheuristic. We propose a technique to handle these new constraints, whether the unavailability periods are known or variable, in order to integrate them into the same type of disjunctive model processed by the analog machine: Binary, Unconstrained, and Quadratic. We present results on small-scale instances corresponding to what these quantum machines can handle.

Keywords: JSSP, non-availability constraints, quantum computing, QUBO, quantum annealing.

1 Introduction

Quantum optimization, leveraging quantum computers and algorithms to address complex optimization issues, stands as a highly promising area in quantum computing. As in the classical domain, two principal strategies are utilized to solve combinatorial problems in quantum optimization: exact methods like Grover's search algorithm [10] and meta-heuristics such as Quantum Annealing (QA) [11] and the Quantum Approximate Optimization Algorithm (QAOA) [8]. Exact and variational methods like QAOA can be processed on universal gate-based quantum computers, such as IBM machines. In contrast, QA is tailored for analog quantum computers, notably those produced by D-Wave.

Quantum Annealing, a key metaheuristic in quantum optimization, is particularly designed for combinatorial optimization problems, drawing from the principles of quantum mechanics and emulating the process of simulated annealing [12]. It utilizes quantum phenomena, such as superposition and quantum tunneling, to efficiently navigate through local minima and target the global minimum of a cost function. For heuristic approaches like QA, it is often necessary to transform the optimization problem into a format compatible with quantum computers. Quadratic Unconstrained Binary Optimization (QUBO) is generally the preferred format for mapping problems to quantum computers.

In this work, we study the resolution of the job shop scheduling problem with availability constraints by the quantum annealing metaheuristic. We first describe the problem in the following section and review the quantum-based solution methods recently proposed in the literature. Then, the QUBO formulation of the JSSP adapted from [1] is presented. Through minor modifications, we show how this QUBO can be adapted to integrate both fixed and flexible availability constraints. The paper concludes with numerical results obtained using D-Wave's quantum annealing machines and overall conclusions. This synthesis merges the concept of quantum annealing's efficacy with broader quantum optimization approaches and their application to specific problems like JSSP, highlighting the diverse methodologies and quantum computing platforms in use.

2 Problem definition

The Job Shop Scheduling Problem with Availability Constraints (JSSP-AC) can be stated as follows: A set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ has to be processed on a set of m machines $M = \{M_1, \ldots, M_m\}$. Each job J_i consists of a linear sequence of n_i operations $(O_{i1}, O_{i2}, \ldots, O_{in_i})$. Each machine can process only one operation at a time and each operation O_{ij} with a processing time of p_{ij} time units needs exactly one machine. Each job visits the machines according to its own predefined routing. This problem generalizes the flow shop scheduling problem, in which all the jobs are processed following the same routing $(M_1, M_2, \ldots, M_m)$. There are k unavailability periods $\{h_{j1}, h_{j2}, \ldots, h_{jk}\}$ on each machine M_j. Two cases are considered in the paper: either the starting date S_{jk} of unavailability period h_{jk} of duration p'_{jk} is known in advance and fixed, or it is flexible and can vary within a time window. The objective is to determine the starting date of each operation O_{ij} so that the makespan noted C_{max} is minimized. The job shop scheduling problem with availability constraints is strongly NP-hard since the 2-machine flow shop scheduling problem is strongly NP-hard [4].

In what follows we first focus and the Job shop scheduling problem, then we generalize the approach to integrate the availability constraints.

The traditional solution approaches to solve JSSP include heuristics and meta-heuristics as well as exact methods, such as branch-and-bound and constraint programming [5]. The linear disjunctive model [15] for the JSSP can be expressed as follows. The starting times are represented by the integer variable vector, denoted by x. We use z to denote the binary variable vector, which satisfies the following conditions:

$$z_{ijk} = \begin{cases} 1 \text{ if the job } j \text{ precedes job } k \text{ on machine } i, \\ 0 \text{ otherwise.} \end{cases}$$

We note by (σ_1^j, ..., σ_h^j, ..., σ_m^j) the processing order of job j through the machines. The minimization of the objective function (1) forces all the jobs to be finished as soon as possible.

$$\sum_{j \in J} x_{\sigma_m^j j} \tag{1}$$

Another objective extensively discussed and utilized within the literature is the concept of makespan. By adding constraint (2) to the model and replacing the objective function (1) with the minimization of the C_{max} variable, in order to ensure that the last job finishes as early as possible.

$$C_{max} \geq \sum_{j \in J} (x_{\sigma_m^j j} + p_{\sigma_m^j j}) \tag{2}$$

Constraints (3) forbid consecutive operations of one job to start before the previous one is finished.

$$x_{\sigma_h^j j} \geq x_{\sigma_{h-1}^j j} + p_{\sigma_{h-1}^j j} \qquad \forall j \in J, h = 2..m \tag{3}$$

Big $\mathcal{M}$ constraints (4) and (5) forbid to have more than one operation at a time on a given machine.

$$x_{ij} \geq x_{ik} + p_{ik} - \mathcal{M} z_{ijk} \qquad \forall j, k \in J, j < k, i \in M \tag{4}$$

$$x_{ik} \geq x_{ij} + p_{ij} - \mathcal{M}(1 - z_{ijk}) \qquad \forall j, k \in J, j < k, i \in M \tag{5}$$

3 Related works

In the literature, the number of papers dedicated to quantum solutions for hard combinatorial optimization problems is growing fast. In particular, the job shop scheduling problem and its extensions is attracting more and more research works involving quantum computing. Those works can be classified according to the types of quantum computers and algorithms used to solve the problems: analog, universal computers, and simulators. In general, the solution approaches consist in first mapping the decision variables of the considered problems to the qubits of the quantum computer. Then, quantum algorithms are applied to make the qubits value evolve until solutions are found. Solving optimization problems with quantum computers is therefore strongly limited by the number of qubits available, among other hardware constraints.

Since the number of qubits is smaller in universal quantum computers, the studies of the JSSP involving those computers are scarce. The first one was developed by [2]. The authors have proposed four variational quantum heuristics for solving a JSSP with early and late delivery as well as production costs, adapted from a steel manufacturing process. They have compared the performance of the heuristics on two-machine flow shop instances using IBM gate-based computers with 5 to 23 qubits.

Recently, [14] have proposed a QAOA approach to the JSSP with a particular method for updating the parameters of the algorithm. The authors have also investigated the relationship between makespan and energy minimzation.

The number of research works involving quantum annealers and simulators is significantly higher. The first quantum computing approach for solving the JSSP was proposed by Venturelli *et al.* [19]. The authors proposed a time-indexed QUBO formulation and a quantum annealing solution for the makespan minimization. The method was implemented on a D-Wave quantum annealer, with 509 working qubits. The authors also proposed variable pruning techniques, through window shaving and immediate selections, to reduce the number of necessary qubits. The proposed QUBO model has been re-used in several studies, as listed below.

In [13] the authors have developed a hybrid quantum annealing heuristic to solve a particular instance of the job shop scheduling problem on the D-Wave 2000Q quantum annealing system that consists of 2041 qubits and a maximum of 6 connections between qubits. The proposed approach includes variable pruning techniques and a processing window heuristic. In [6], job shop instances with unitary operations have been tested on the D-Wave Advantage machine, built upon 5640 qubits and 15 possible connections between qubits. Extensive experiments with the reverse annealing procedure and comparisons with simulated annealing are also described. In [1], we have proposed a QUBO formulation for the minimisation of the total completion time in a job shop. The model was solved using the D-Wave hybrid solver and Advantage quantum annealing computer.

The flexible job shop, which is a generalization of the job shop problem with pools of parallel machines available for processing operations was considered in [7]. The authors proposed a QUBO derived from the one of [19] and an iterative procedure to solve relatively large size instances on a specialized hardware ([3]). Using the QUBO formulations proposed in [7], the authors in [18], also tackle the flexible job shop scheduling problem with the D-Wave solvers comparing various input models. Another QUBO formulation is proposed in [16] for assigning dispatching rules to the machines and scheduling the operations in a flexible job shop system. The problem is solved using the leap hybrid solver. In [17], the authors propose a QUBO formulation for the job shop scheduling with worker assignment considerations. Possible ways to approximate the makespan are discussed and instances solved with the Fujitsu Digital Annealer are described. In the same environment, the authors in [20] efficiently solve large instances of JSSP with a hybrid approach that combines constraints programming and QUBO models for one-machine problems.

The present paper also aims at extending the job shop scheduling model and solution approach, in particular the one proposed in [1], by considering additional constraints that are important in practice. To the best of our knowledge, it is the first study in the quantum optimization literature that integrates availability constraints on the machines of both fixed and flexible types.

4 QUBO formulation

The QA metaheuristic, as executed on a D-Wave quantum machine, takes as input an unconstrained binary model, which can be quadratic. Either an Ising

model ($\{-1;+1\}$ variable values) or a QUBO model ($\{0;1\}$ variable values) can be provided. Since both models are isomorphic, and the machine is capable of converting QUBO into Ising, we focus on the classical binary variables in computing to more easily establish a connection with known MILP models.

We add some notations to those used in the linear formulation of the previous section. We use x to denote the binary variable vector, which, for each i, j and t, with $i = 1..n_i$, $j = 1..n$, $t = 1..T$, satisfies the following conditions:

$$x_{ij}^t = \begin{cases} 1 \text{ if the operation } i \text{ of the job } j \text{ starts in period } t, \\ 0 \text{ otherwise.} \end{cases}$$

The minimization of the Objective function $f(x)$ forces the last operations of all jobs to start globally as soon as possible (see expression (6)). Here, we adapt the objective function (1) from the integer formulation to a binary formulation that we develop for the QUBO:

$$f(x) = \sum_j \sum_t t.x_{n_i j}^t. \tag{6}$$

For optimizing the makespan, it is sufficient to add a virtual job consisting of a single operation that is executed instantly which will be connected to the last operations of the non-virtual jobs by precedence constraints ((10)). It then only remains to exclusively minimize the execution date of the virtual job as in function (7), where n_i of the virtual job $n+1$ is equal to 1 since there is only one operation.

$$f(x) = \sum_t t.x_{(1)(n+1)}^t. \tag{7}$$

To force each operation to start exactly once through a relaxed constraints into the objective function, we apply a penalty $P1$ such that $P1(x) = \sum_i \sum_j P1(x,i,j)$ where each element is given by the expression (8).

$$P1(x,i,j) = (\sum_t x_{ij}^t - 1)^2, \qquad i = 1..n_i, j = 1..n. \tag{8}$$

We note $M_{ij}, i = 1..n_i, j = 1..n$, the required machine for the operation i of the job j. $P2(x)$ is the penalty that forbids to have more than one operation at a time on a given machine, such that $P2(x)$ is the sum of each element calculated by the quadratic expression (9).

$$\begin{aligned} P2(x,i,j,t,i',j',t') = x_{ij}^t x_{i'j'}^{t'}, \\ (i,j,t) \cup (i',j',t') : i,i' = 1..n_i, j,j' = 1..n, (i,j) \neq (i',j'), \\ M_{ij} = M_{i'j'}, (t,t') \in T^2, 0 \leq t' - t < p_{ij}. \end{aligned} \tag{9}$$

The last Penalty which is noted $P3(x)$ forbids consecutive operations to start before the previous one is finished. Each element of $P3(x)$ is calculated by the quadratic expression (10).

$$P3(x,i,j,t,t') = x_{ij}^{t} x_{i+1j}^{t'}, \quad i = 1..(n_i - 1), j = 1..n, (t,t') \in T^2, t + p_{ij} > t'. \tag{10}$$

We can finally express the JSSP quadratically and without constraints through the QUBO formulation of the JSSP with its penalties balanced by 3 multipliers, λ_1, λ_2, and λ_3 (see expression 11) and its detailed form of equality (12) .

$$f^{QUBO}(x) = f(x) + \lambda_1 P1(x) + \lambda_2 P2(x) + \lambda_3 P3(x). \tag{11}$$

$$\begin{aligned} f^{QUBO}(x) &= \sum_j \sum_t t.x_{n_i j}^{t} + \lambda_1 \sum_j \sum_i (\sum_t x_{ij}^{t} - 1)^2 \\ &+\lambda_2 \sum_{(i,j,t)\cup(i',j',t') \in T1} x_{ij}^{t} x_{i'j'}^{t'} + \lambda_3 \sum_{(i,j,t,t') \in T2} x_{ij}^{t} x_{i+1j}^{t'}. \end{aligned} \tag{12}$$

with:

$$T1 = (i,j,t) \cup (i',j',t') : i,i' = 1..n_i, j,j' = 1..n, (i,j) \neq (i',j'),$$
$$M_{ij} = M_{i'j'}, (t,t') \in T^2, 0 \leq t' - t < p_{ij}.$$
$$T2 = (i,j,t,t') : i = 1..(n_i - 1), j = 1..n, (t,t') \in T^2, t + p_{ij} > t'.$$

5 Non Fixed Resource Availability Constraints

Let's consider the problem of resource constraints due to unavailability, whether these are fixed or variable. The management of these resources proves to be intuitive when the problem is formulated as a QUBO. The UML activity diagram shown in Figure (1) illustrates the methodology for developing the QUBO, with a particular emphasis on non-availability constraint management.

When a resource's unavailability is constant over time, it can be treated as a single operation already scheduled. Thus, it becomes possible to spread this constraint throughout all the other operations that cannot simultaneously use the resource. This consideration is expressed through elementary quadratic expressions of the form $x_{ij}^{t} x_{i'j'}^{t'}$, where i and j denote the fictive operation representing the resource's unavailability during a certain period p_{ij}. For any t included in this period, and for all operations characterized by i' and j' that use the same resource, we impose the constraint $x_{ij}^{t} x_{i'j'}^{t'} = 0$ with the related penalty.

When a resource's unavailability has to be scheduled, it should be considered as a unique operation of a project that can be scheduled at any time. If the objective is to minimize the makespan, the virtual operation related to the resource non-availability is integrated in the calculation of this makespan as a last operation of a job. Hence, the unavailability constraints, regardless of their nature, can be sequentially incorporated into the QUBO. We finally obtain a JSSP problem with additional jobs comprising single operations. The QUBO

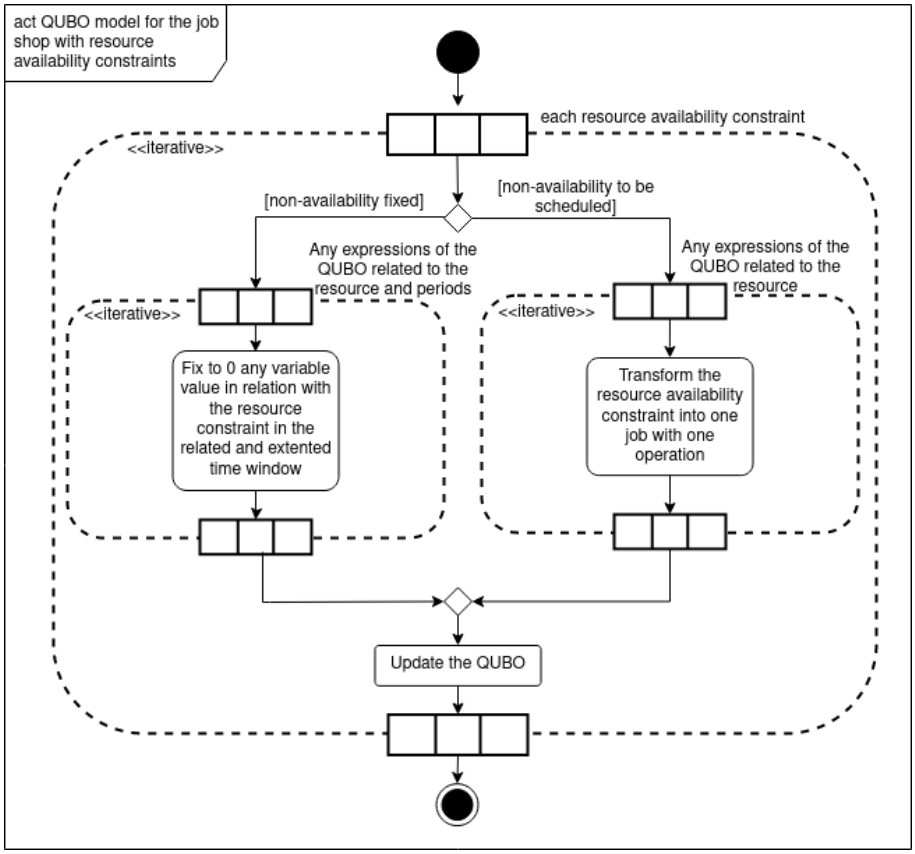


Fig. 1. The non-availability constraints are divided into two subsets: those whose unavailability window is already known, and those where this window is variable. In both cases, we consider each unavailability as a unique operation of a new Job. In the first case, it involves fixing the variables corresponding to these specific operations, and, in the second case, it involves considering the new operations as any other activities where optimization will lead them to finish globally as soon as possible.

model used is thus the same $f^{QUBO}(x)$ as the one given by expression (12) in the previous section.

In Figure 2, we present an example of the Job Shop Scheduling Problem (JSSP) incorporating various types of unavailability constraints. The period labeled 'U1' denotes a time during which Machine 1 is unavailable, specifically in period 3. Conversely, the 'U2' period represents a variable unavailability duration. It is noteworthy that this variable unavailability period is strategically optimized prior to commencing the first operation of Job 3. As depicted, minimizing the makespan necessitates careful scheduling around the ends of these variable unavailability periods.

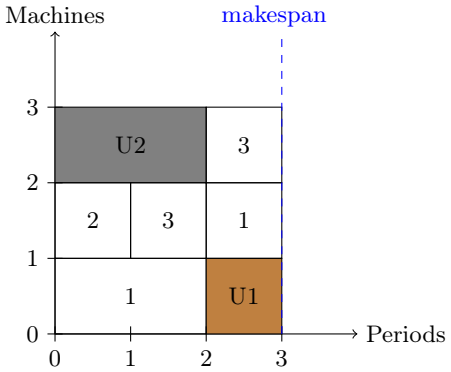


Fig. 2. A solution to JSSP with unavailability constraints, featuring three jobs and a total of six operations. Unavailability constraint U1 is associated with a fixed period of non-availability, whereas the period for U2 is variable.

6 Computational Experiments

In [1], we studied the impact that the number of periods in a JSSP instance has on its resolution. We noticed that iterating different experiments with a reduction in the number of periods until an infeasible solution is obtained showed a significant improvement in results, more than what could be expected from classical calculations, due to the narrowing of the solution space. In this study, we directly considered a relatively small number of periods. We achieved this result empirically and quickly, thanks to a small number of replications initially performed on a QUBO related to the same instance but with a larger number of periods.

We opted for the quantum quadratic unconstrained binary solver, which is non-hybrid, in contrast to the method of resolution discussed in [1], which relied on D-Wave's hybrid solution for solving constrained binary quadratic problems. The experiment was conducted on an initial instance of the JSSP with 3 jobs

for a total of 7 operations. The size of the base instance was chosen to enable processing by the D-Wave Advantage machine, while also considering the ability to achieve optimality. The instance with a fixed period of unavailability is distinguished by the stopping of machine 2 during the second time period. The third case concerns a variable unavailability of machine 3 over a single time period. We aimed to minimize the makespan by adding a job with only a virtual operation at the beginning and another at the end of the experiment. This was done by adding precedence constraints, focusing the objective on minimizing the start date of the last virtual operation.

For these preliminary results, we achieved the optimal solution in all three cases, where each time 3 periods were necessary to meet all the constraints of the 3 problems. The figures respectively represent the optimal solutions for the JSSP case and for the case with variable unavailability. In the latter, the unavailability period was placed after all other operations, without affecting the makespan. Figure 3 and Figure 4 correspond respectively to Figures 5 and 4. We can observe the impact that the embedding process (mapping of the QUBO graph to the qubit graph) can have on the problem addressed by the quantum machine. This experiment shows that the number of qubits increases from 122 to 208 when moving from the JSSP instance to the instance with variable machine unavailability, where such unavailability is represented by an additional job with a single operation.

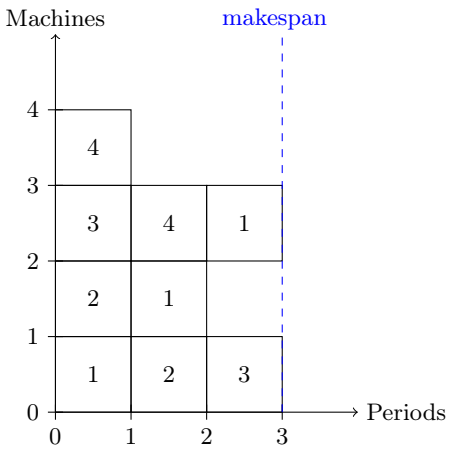


Fig. 3. JSSP without unavailability constraints solution obtained.

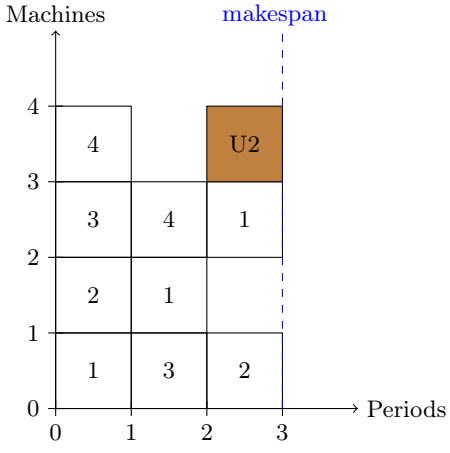


Fig. 4. JSSP with variable unavailability constraints solution obtained.

Fig. 5. Representation of the QUBO graph (left), corresponding to a scenario where all machines are continuously available, and its embedding (right) into the qubit graph of a the Advantage machine. This particular case involves 4 jobs and a total of 9 operations. The QUBO model has 48 variables, while the embedding on the QPU requires 122 qubits. (Visualization created using D-Wave Inspector).

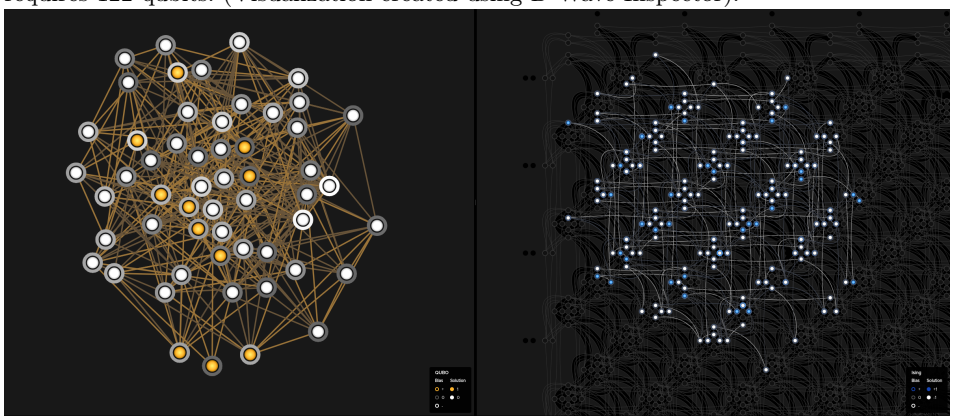
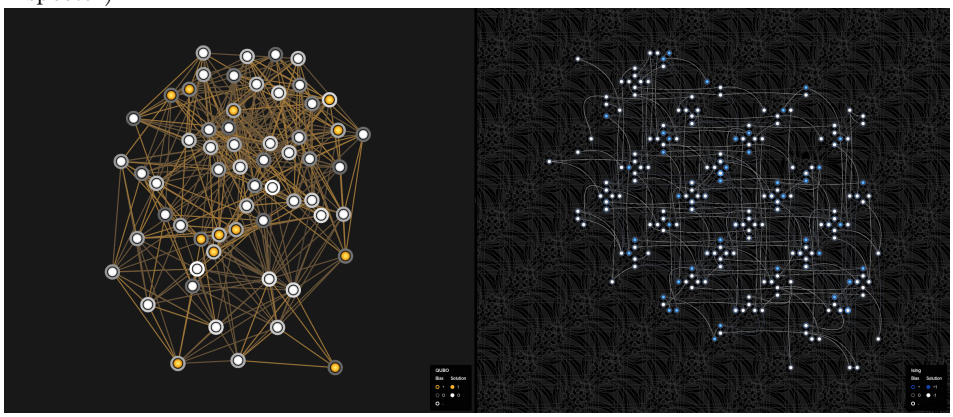


Fig. 6. Representation of the QUBO graph (left) illustrating a scenario in which a machine is unavailable during a specific period, and its corresponding embedding (right) into the qubit graph of the Advantage machine. This instance involves managing 4 jobs with a total of 9 operations. The QUBO model comprises 60 variables, and its embedding into the QPU utilizes 208 qubits. (Visualization created using D-Wave Inspector).



7 Discussion

In our study, we conducted an analysis of recent quantum computing strategies for the JSSP, particularly emphasizing QUBO models and their role in effectively

incorporating practical constraints. We apply this method to a JSSP with machine unavailability constraints. These constraints can be either variable or fixed. Starting from a QUBO that models the JSSP and is well-suited for quadratic modeling, we treat these new constraints as unique operations of jobs that are added to the initial ones. Thus, the same QUBO formulation could be implemented to describe instances for the quantum machine.

Our research included the development and testing of this QUBO model using D-Wave's quantum annealing technology. We noted that current quantum hardware has limitations in managing the volume of variables produced by such models. Despite this, it's crucial to continue refining the modeling of real-world problems, as quantum approaches, while not currently outperforming classical methods, hold potential for future advancements, especially with the anticipated increase in available qubits. Our future research aims to explore methods to minimize variable count while still efficiently embedding necessary constraints. Additionally, the recent increase in the number of qubits, surpassing the 1000-qubit threshold in machines from IBM or Atom Computing, now allows for the consideration of solving small-scale instances like those addressed in this work. This is made possible through quantum algorithms such as QAOA implemented on these discrete (gate-based) machines.

References

1. Aggoune, R. and S. Deleplanque. A Quantum Annealing Solution to the Job Shop Scheduling Problem. *International Conference on Computational Science and Its Applications*, LNCS, Springer. *2*023. ⟨hal-04172854⟩
2. Amaro, D. and Rosenkranz, M. and Fitzpatrick, N. and Hirano, K. and M. Fiorentini. A case study of variational quantum algorithms for a job shop scheduling problem. *EPJ Quantum Technol*, 9:100–114, 2022.
3. Aramon, Maliheh and Rosenberg, Gili and Valiante, Elisabetta and Miyazawa, Toshiyuki and Tamura, Hirotaka and Katzgraber, Helmut G. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics*, 7: 48, 2019.
4. Blazewicz and J. Breit and P. Formanowicz and W. Kubiak and G. Schmidt. Heuristic algorithms for the two-machine flowshop Problem with limited machine availability. *Omega Journal*, 29:599–608, 2001.
5. Da Col, G. and E. C. Teppan. Industrial-size job shop scheduling with constraint programming. *Operations Research Perspectives*, 9, 2022.
6. Carugno, C. and Ferrari Dacrema, M. and Cremonesi, P. Evaluating the job shop scheduling problem on a D-wave quantum annealer. *Sci Rep*, 12:6539, 2022. https://doi.org/10.1038/s41598-022-10169-0.
7. Denkena Berend and Schinkel Fritz and Pirnay Jonathan and Sören Wilmsmeier. *Quantum algorithms for process parallel flexible job shop scheduling. CIRP Journal of Manufacturing Science and Technology*, 12142, 2020.
8. Farhi, Edward and Goldstone, Jeffrey and Gutmann, Sam. *A Quantum Approximate Optimization Algorithm. 10.48550/arxiv.1411.4028*, 2014.
9. Garey, Michael R. and Johnson, David S. *Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., USA*, 1979.

10. Grover, Lov K. *A Fast Quantum Mechanical Algorithm for Database Search. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 212–219, 1996.
11. Kadowaki Tadashi and Hidetoshi Nishimori. *Quantum annealing in the transverse ising model. Physical Review E*, 58(5):5355, 1998.
12. Kirkpatrick, S. and Gelatt Jr, C D. and Vecchi, M. P. *Optimization by simulated annealing science, 220, 4598*, 1983
13. Kurowski, K. and Węglarz, J. and Subocz, M. and Różycki, R. and Waligóra, G. *Hybrid Quantum Annealing Heuristic Method for Solving Job Shop Scheduling Problem. ICCS 2020. Lecture Notes in Computer Science, Springer*, 33:100–114, 2021.
14. Kurowski, K. and Pecyna T. and Slysz R. and Różycki, R. and Waligóra, G. and Węglarz, J. *Application of quantum approximate optimization algorithm to job shop scheduling problem. European Journal of Operational Research*, 310:518–528, 2023. https://doi.org/10.1016/j.ejor.2023.03.013.
15. Manne, Alan S. *On the Job-Shop Scheduling Problem. Operations Research*, 8(2):–223. 1960. https://doi.org/10.1287/opre.8.2.219
16. Poojith U Rao and Baldwinder Sodhi. *Scheduling with Multiple Dispatch Rules: A Quantum Computing Approach. ICCS 2022. Lecture Notes in Computer Science, Springer*, 13353:233–246, 2022.
17. Shimada, D.and T. Shibuya and T. Shibasaki. *A Decomposition Method for Makespan Minimization in Job-Shop Scheduling Problem Using Ising Machine. 2021 IEEE 8th International Conference on Industrial Engineering and Applications*, 307–314, 2021.
18. Schworm, P., Wu, X., Glatt, M. and J. C. Aurich. *Solving flexible job shop scheduling problems in manufacturing with Quantum Annealing.. Prod. Eng. Res. Devel.*, 17: 105-115, 2023. https://doi.org/10.1007/s11740-022-01145-8
19. Venturelli, D. and Marchand, D. J. and Rojo, G. *Quantum annealing implementation of job-shop scheduling. arXiv preprint:1506.08479*, 2015.
20. Zhang, J., Lo Bianco, G., and Beck, J. C. *Solving Job-Shop Scheduling Problems with QUBO-Based Specialized Hardware. Proceedings of the International Conference on Automated Planning and Scheduling*, 32(1) 404–412, 2022.