European Journal of Operational Research 310 (2023) 518–528

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Application of quantum approximate optimization algorithm to job shop scheduling problem

Krzysztof Kurowski[a,*], Tomasz Pecyna[a], Mateusz Slysz[a], Rafał Różycki[b], Grzegorz Waligóra[b], Jan Węglarz[b]

[a] Poznan Supercomputing and Networking Center, Poznan, IBCH PAS, Poland
[b] Poznan University of Technology, Institute of Computing Science, Poznan, Poland

ARTICLE INFO

Article history:
Received 6 October 2021
Accepted 8 March 2023
Available online 12 March 2023

Keywords:
Scheduling
Computing science
Heuristics
Job shop scheduling problem
Quantum approximate optimization algorithm

ABSTRACT

The Job Shop Scheduling Problem (JSSP) has always been considered as one of the most complex and industry essential scheduling problems. Optimizing the makespan of a given schedule generally involves using dedicated algorithms, local search strategies, or metaheuristics. These approaches, however, heavily rely on classical computational power, which is bounded by the physical limits of microcontrollers and power issues. Inspired by the promising results achieved for Quantum Annealing (QA) based approaches to solve JSSP instances, we propose a new approach that uses gate-model quantum architecture as an alternative to QA. We find that we can make use of the time-indexed JSSP instance representation to build a cost Hamiltonian, which can be embedded into Quantum Approximate Optimization Algorithm (QAOA) to find an optimal solution to a basic JSSP instance. We demonstrate the use of QAOA to solve the JSSP, and we evaluate its efficiency and accuracy for this problem from experimental results, as there is an increased urgency to demonstrate the applicability of quantum optimization algorithms. We also find that optimal variational parameters form patterns that can facilitate computation in bigger quantum circuits. Additionally, we compare the obtained noiseless simulation results of gate-model quantum circuits demonstrating the relationship between two evaluation criteria - makespan and energy. Finally, we analyze and present the overall performance of our approach with the increasing deadline and simulated depth of QAOA circuits.

1. Introduction

The real value of quantum computers can be unlocked only through new applications, especially in operational research, where many difficult problems are of practical interest and hard to solve due to their computational complexity. Inspired by the unique features of the quantum realm, quantum computing promised from the early 1980s to deliver the ability to solve problems beyond classical computers' capability. Despite the impressive progress of quantum research, including recent studies and interesting attempts based on experimental quantum hardware, it is still unclear to many researchers how and to what extent they can benefit from quantum computations. They need to understand how to design, implement, and run a quantum computational experiment to solve a specific combinatorial optimization problem in a reliable and controlled way. It is worth introducing fundamental concepts and existing challenges before diving into our quantum algorithmic and experimental details.

There is a conceptual analogy and mathematical resemblance between the equations of an objective function used in an optimization problem and a *Hamiltonian* formalism. In a nutshell, the Hamiltonian, with its cost value, is an operator for the total energy of a system in quantum mechanics. Historically, it was realized back in the 1980s that simulating quantum dynamics to find the ground state energy of even a small molecule is far too complex for a classical computer as the computational power required to describe a quantum system scales exponentially with the number of components. Thus, the idea of using quantum instead of a classical computer to simulate other quantum systems evolved slowly after new concepts connecting quantum computation and ground states of many-body quantum systems were discussed in Feynman (1982). Additionally, the construction of a microscopic quantum mechanical Hamiltonian model of the computation processes represented by Turing machines was proposed in Benioff (1980). Over the next decade, there have been a lot of theoret-

* Corresponding author.
E-mail address: krzysztof.kurowski@man.poznan.pl (K. Kurowski).

https://doi.org/10.1016/j.ejor.2023.03.013

ical attempts to demonstrate the potential of quantum mechanical computers and their computational properties, including but not limited to concepts of universal and inefficient quantum Turing machine presented initially in Deutsch (1985) and equivalence between quantum Turing machines and uniform quantum circuits in Yao (1993).

Although the tremendous theoretical results sparked a lot of interest among some researchers, many doubts about their practical significance and applications remained. One of the biggest technical challenges towards scalable and reliable quantum computing is decoherence phenomenon. Compared with classical computers, quantum computers are extremely susceptible to noise. Decoherence is the permanent adversary of quantum information processing since it destroys the fragile superpositions where information is stored. Lower decoherence can alleviate the error correction overhead and significantly reduce noise in quantum circuits and processing. Basically, the greater the influence of noise, the shorter the quantum algorithm that can be run before it suffers an error and outputs an incorrect result. Thus, stability and error correction issues remain an obstacle for unlocking quantum supremacy in solving real-world problems today. To describe the current state of the art in the fabrication of quantum circuits the concept of *Noisy Intermediate-Scale Quantum* (NISQ) era was introduced recently in Preskill (2018). It is clear that even today special attention must be paid to quantum error correction as NISQ devices are still not fault-tolerant and contain only a limited number (going into hundreds) of available basic units of quantum information - *qubits*. Thus, it is essential to identify a computing task or problem of interest to operational research that hopefully can be performed more efficiently, with better quality or more cost-effectively using quantum computers. Today, even without access to NISQ devices, advanced analysis and experimental tests of new quantum algorithms relevant for operational research can be performed thanks to quantum simulators widely available, as we demonstrate in this paper. Consequently, new quantum algorithms could be evaluated experimentally, adapted and tuned to constantly improved quantum NISQ devices, and we could have practical applications in the future. Last but not least, quantum computers promise to use much less energy and still vastly outperform supercomputers in the future. Modern classical supercomputers use between one to several megawatts of power on average. In contrast, existing NISQ devices use tens of kilowatts and generate almost no heat. Naturally, energy efficiency will depend on the concrete architecture and thus on available technological solutions. However, we may expect significant progress and a technological breakthrough in this relevant and economic aspect of computations.

The rest of this paper is organized as follows. In Section 2 we give a state of the art and brief review of the related works. In Section 3 we present the problem formulation. In Section 4 we formulate the Ising Hamiltonian representation applicable to the Quantum Approximate Optimization Algorithm (QAOA). Computational experiments and the obtained results are described in Section 5. Section 6 concludes the paper and shows some directions for future work.

2. State of the art

From a conceptual computational complexity perspective, the Deutsch-Jozsa algorithm was the first to show a separation between the quantum and classical difficulty of a problem (Deutsch & Jozsa, 1992). Then, a set of first truly and historically important quantum algorithms were proposed, in particular quantum polynomial-time algorithms for the discrete logarithm and integer factoring problems (Shor, 1994), the first algorithm to solve a promise problem exponentially faster than any classical algorithm (Simon, 1994), or the quantum search algorithm achieving quadratic speedup in unordered database search (Grover, 1996). Although we have some evidence of quantum algorithms performing better than their classical equivalents we are far from precisely establishing the true power of quantum computers. The quantum complexity theory was naturally investigated in Bernstein & Vazirani (1993), Yao (1993), and a new class of computational problems called '*Bounded error, Quantum, Polynomial time*' (BQP) was then introduced in Bernstein & Vazirani (1997). BQP consists of those decision problems that are solvable with bounded probability of error using a polynomial-size quantum circuit, and examples of problems belonging to this class can be the aforementioned integer factorization. We know so far that BQP is contained inside PSPACE class, which is the class of decision problems solvable by a Turing machine in polynomial space, and that it contains the BPP (problems which can be solved using randomized algorithms in polynomial time if bounded by probability error), hence the P class (Fortnow & Rogers, 1999). It is then determined that P $\subseteq$ BPP $\subseteq$ BQP $\subseteq$ PSPACE $\subseteq$ EXP. Since we know that P $\subset$ EXP and do not know which of the inclusion is stricter, and also that we do not know how BQP relates to NP, researchers suspect that there might be some problems outside NP that quantum computers can solve efficiently. Another motivation towards quantum computers is the Solovay Kitaev theorem (Kitaev, 1997) which states that an arbitrary single qubit gate may be approximated to some accuracy using polylogarithmic number of gates from a predefined universal discrete set, so we know that we can construct quantum circuits efficiently.

The next relevant step was the introduction of quantum fluctuations into the well-known simulated annealing process of optimization problems in Kadowaki & Nishimori (1998). Then, quantum computation by adiabatic evolution was proved in Farhi, Goldstone, Gutmann, & Sipser (2000), and suggested a novel quantum algorithm for solving the satisfiability problem and other combinatorial search problems. Additionally, adiabatic computation has been shown to be polynomially equivalent to conventional quantum computing in the quantum gate model (Aharonov et al., 2007). In principle, the concept of Quantum Annealing (QA) came from the well-known metaheuristic optimization technique called Simulated Annealing (SA), in which the space of admissible solutions to a given optimization problem is penetrated by temperature-dependent random movements. The cost function defines the total energy of the solution space, and the solution space can be efficiently explored thanks to thermal fluctuations. The basic concept of simulating annealing has been adopted and implemented in quantum hardware successfully when quantum fluctuations have replaced thermal fluctuations known from the classical SA approach (Boixo, Albash, Spedalieri, Chancellor, & Lidar, 2013; Humble et al., 2013). Consequently, we have been observing a rapid growth of QA approaches successfully used for solving optimization problems formulated in terms of finding ground states of classical Ising spin Hamiltonians (Lucas, 2014). Today's quantum annealing devices (e.g. D-Wave) can be used to solve small instances of combinatorial optimization problems, including the Job Shop Scheduling Problem (JSSP), as it has been demonstrated in Venturelli, Marchand, & Rojo (2016) and Kurowski, Węglarz, Subocz, Różycki, & Waligóra (2020).

One should note that state of the art quantum computers in the NISQ era can only be easily applied to some problems, but they perform pretty well for some computational problems. If encoded correctly, selected problems can be solved on gate-based quantum computers (e.g. IBM quantum processing unit) thanks to variational hybrid quantum-classical algorithms. In general, efficient variational hybrid quantum-classical approaches include two leading algorithms, namely the Variational Quantum Eigensolver (VQE) (Cerezo et al., 2021; Coveney & Highfield, 2020; Ralli, Love, Tranter, & Coveney, 2021) and Quantum Approximate Optimization Al-

gorithm (QAOA) (Farhi, Goldstone, & Gutmann, 2014). QAOA is a hybrid (quantum-classical) algorithm that approximates the value of the optimal solution of a binary optimization problem with its accuracy controlled by the hyperparameter p, which is a small positive integer. The cost function of the problem is mapped to a Hamiltonian represented by a quantum circuit with depth (length) dependent on p. The quantum circuit that implements the algorithm consists of unitary gates and is evaluated several times on a quantum device with respect to classically precomputed variable parameters, updated with every iteration.

QAOA has already been applied to a few well-known combinatorial optimization problems, such as Max-Cut (Crooks, 2018), Travelling Salesman Problem (Radzihovsky, Murphy, & Mason, 2019), and Graph Coloring (Tabi et al., 2020). The author of the first paper studied the performance of QAOA on the MaxCut problem, optimizing the quantum circuits on a classical computer using automatic differentiation and stochastic gradient descent (Crooks, 2018). It was demonstrated that it is possible to amortize the training cost by optimizing batches of problem instances. The paper shows that QAOA can exceed the performance of the classical polynomial time Goemans-Williamson algorithm with modest circuit depth and that performance with fixed circuit depth is insensitive to problem size. Moreover, MaxCut QAOA can be efficiently implemented on a gate-based quantum computer with limited qubit connectivity using a qubit swap network. The presented results support the prospects that QAOA will be an efficient method for solving complex combinatorial optimization problems on near-term quantum computers. In the second paper the authors implemented and investigated two versions of QAOA for the Travelling Salesman Problem (TSP) (Radzihovsky et al., 2019). They call them Hamiltonian cost implementation and mixer implementation, respectively. The authors showed that their mixer implementation successfully solves the TSP and reproduces, up to cyclic permutations, the solution found by the classical solver. They also found that the mixer approach requires fewer gates than the Hamiltonian cost implementation, whereas the cost Hamiltonian is generally faster than the mixer. The authors stated that their implementations provide a glimpse into what problems quantum computers can solve and the possibility of utilizing quantum supremacy. They concluded that although they could not compare the quantum algorithms against classical algorithms on real-world scale problems, this will be an exciting area of future research as quantum hardware continues to improve. Finally, the authors introduced a novel space-efficient quantum optimization algorithm for the Graph Coloring Problem (Tabi et al., 2020). Their circuits were deeper than the ones of the standard approach. However, through a series of investigations, the authors presented the performance gain of this method. They showed that the required circuit width to embed the colouring problem was exponentially reduced in the number of colours. Although the depth of a single QAOA layer was increased, the number of required layers and optimization iteration steps to reach the optimal solution also decreased. The authors state that the proposed method and comparative study can be extended to a benchmarking framework for such performance gain analyses. Furthermore, they concluded that analogous space-efficient embedding techniques could be used to improve upon other graph-related quantum optimization methods.

Although QAOA is proven to improve the solution quality with increased depth, it is hard to scale this regularity on real NISQ quantum hardware due to errors caused by qubit imperfections such as cross-talk or decoherence, as well as errors caused by faulty gates and measurements. The current references demonstrate that a notable quantum advantage seems likely to be observed on real quantum hardware once the noise decreases by two orders of magnitude (Stilck França & Garcia-Patron, 2021). For this reason, researchers limit themselves in running QAOA experiments only to small circuits (De Palma, Marvian, Rouzé, & França, 2022; Harrigan et al., 2021; Khumalo, Chieza, Prag, & Woolway, 2022; Magann et al., 2021) of size $p = 1$ to $p = 3$. Nonetheless, alternatives that seem more robust to noise have been introduced, e.g. the feedback-based technique discussed in Magann, Rudinger, Grace, & Sarovar (2022).

In general, two main approaches address the issue of noise and errors in existing NISQ devices. The short-term approach is quantum error mitigation techniques used to deal with errors as they occur. Error mitigation techniques operate mainly in the post-processing stage, and therefore, they have limited usage because their usability decreases with circuit length and complexity. The long-term answer is the quantum error correction approach with techniques that can fix errors during circuit execution. With error correction, there exists a concept of a logical qubit, which can be composed of physical qubits, typically in the range between 5 (simple error correction techniques) and 1000 (perfect logical qubit). For more information about quantum error mitigation and correction, we refer the reader to standard textbooks, e.g. Nielsen & Chuang (2002). As our research focused on experimental verification of QAOA using a quantum simulator, not real NISQ quantum hardware, detailed analysis is outside the scope of this paper.

Nevertheless, even if the size of considered problem instances is small for today, we have been observing rapid development of quantum technologies, new error correction techniques, and constant growth of the number and better quality qubits. It will result in the possibility of solving much larger problem instances in the near future. Consequently, it is relevant to develop and verify experimentally new quantum-based approaches, such as QAOA, to complex combinatorial problems today since they will be able to cope with bigger and bigger problem instances until they finally surpass classical algorithms and prove quantum supremacy. Some promising symptoms of that have already been shown in the abovementioned papers.

Thus, this paper is aimed to identify a possible adaptation of QAOA to scheduling problems by conducting a relatively simple but comprehensive series of experiments to pave the way for the scheduling application's development. Due to many practical applications of scheduling in advanced planning, decision support and computer systems, we have selected the JSSP benchmark. In this paper, we show a set of advanced analyses and guide the reader step by step, demonstrating possible opportunities and limits in quantum simulation environments using still significant classical computational power, which can be supported or replaced by improved NISQ devices in the future.

3. Job shop scheduling problem

3.1. Problem formulation

The Job Shop Scheduling Problem (JSSP) has been one of the most studied optimization problems over a few decades. In the problem a set of dedicated (i.e. different) machines is to perform tasks of jobs, i.e. each job is composed of an ordered list of tasks, from among which every task requires a specific machine for a known processing time. There exist several constraints imposed on jobs and machines: (i) tasks are nonpreemptable, (ii) tasks of different jobs are independent, (iii) each task can be performed on one machine at a time, and (iv) each machine can process only one job at a time. The problem is to minimize the makespan, i.e. the maximum completion time of all tasks. The JSSP belongs to the most intractable scheduling problems considered in the literature. It is NP-hard in the strong sense, and only a few particular special cases are efficiently solvable. There are several different formulations of the JSSP (see e.g., Błażewicz, Dror, & Węglarz, 1991 for a survey), whereas a problem instance, as well as a feasible solution,

can be represented by a disjunctive graph or its specialized representation the graph matrix (Błażewicz, Pesch, & Sterna, 2000). In order to solve JSSPs exact methods, heuristic and metaheuristic algorithms have been used over the years (Błażewicz, Domschke, & Pesch, 1996; Jain & Meeran, 1999). The exact approaches have almost entirely been based on branch-and-bound procedures. These, as it is known, rely very much on strong lower bounds in order to cut branches of the enumeration tree as early as possible. Lower bounds for the JSSP have been analyzed in e.g., Brucker & Jurisch (1993) and Carlier & Pinson (1994). As far as approximation algorithms are concerned, many approaches have used priority rules to order the tasks of jobs. For an extended summary and discussion see Haupt (1989). Various schedule generation schemes for the JSSP with sequence-dependent setup times have been analyzed in Artigues, Lopez, & Ayache (2005). One of the most powerful procedures used for the problem is the shifting bottleneck heuristic (Adams, Balas, & Zawack, 1988; Pezzella & Merelli, 2000). Also constraint propagation approach has been widely used to solve the problem (Dorndorf, Pesch, & Phan-Huy, 2002). From among local search methods a great variety of approaches have been applied to the JSSP (see Vaessens, Aarts, & Lenstra, 1996 for a survey). Many metaheuristic algorithms have also been proposed over the years, including simulated annealing, tabu search, genetic algorithms, ant colony optimization, variable depth search, and their hybrids (see Błażewicz et al., 2019 for an extensive review). From among them excellent results are presented in Balas & Vazacopoulos (1998), where a guided local search with shifting bottleneck has been proposed, and in Zhang, Li, Rao, & Guan (2008) in which a simulated annealing/tabu search hybrid is described. In order to compare the efficiency of algorithms various benchmark sets are being used. A review of the current state of bounds on benchmark instances of the JSSP is given in van Hoorn (2018).

In this paper we use the following formulation of the JSSP. There are J jobs $\mathcal{J} = \{j_1, \ldots, j_J\}$ and each of them has O_j operations (tasks) $\mathcal{O}_j = \{o_{j1} \to \cdots \to o_{jO_j}\}$ to be processed in predefined order. Each of these single job operations must be processed on a specified and distinct machine from a set of M machines, $\mathcal{M} = \{m_1, \ldots, m_M\}$, and only one operation can be processed by a machine at a given time. Note that $\forall j\ O_j \leq M$. The objective is to find the minimum makespan, i.e., the earliest completion time of the last running job.

3.2. Problem representation

To effectively apply QAOA to the JSSP we need to find a representation, so that the feasibility constraints can be defined as a sum of binary clauses. Initially, a generic method for the decision version of JSSP together with experimental tests was presented in Venturelli et al. (2016). We have proposed an extension of this method for the optimization version of JSSP. In addition, when comparing two feasible solutions, in our approach, we guaranteed that the one with a shorter makespan would also be of lower energy. Consequently, a new Hamiltonian function responsible for optimizing makespan was proposed, parametrized, and considered during experimental feasibility studies for solving a well-known reference JSSP benchmark (FT06) on the D-Wave 2000Q quantum annealer (Kurowski et al., 2020). We use the time-indexed instance representation and for each operation we assign a set of binary variables representing specific timestamps on which the operation can start, i.e.:

$$x_{k,t} = \begin{cases} 1: & \text{if operation } o_k \text{ starts at time } t \\ 0: & \text{otherwise.} \end{cases} \tag{1}$$

The time t is bounded by an arbitrary deadline T common for all jobs, and the index k is a running index representing a position of an operation in a list that concatenates all operations over all jobs:

$$\left[\underbrace{o_{11}, \ldots, o_{1O_1}}_{j_1}, \underbrace{o_{21}, \ldots o_{2O_2}}_{j_2} \ldots, \underbrace{o_{J1}, \ldots o_{JO_J}}_{j_J} \right] = \left[\underbrace{o_1, \ldots, o_{k_1}}_{j_1}, \underbrace{o_{k_1+1}, \ldots o_{k_2}}_{j_2} \ldots, \underbrace{o_{k_{J-1}+1}, \ldots o_{k_J}}_{j_J} \right]. \tag{2}$$

Note, that in general the deadline T has to be estimated e.g. using some heuristic algorithm. In this paper, however, we restrain ourselves to setting T as a makespan of a random, feasible solution.

Let us now define a set of constraints which assures that the solution is feasible and ends before T. Firstly, the operation must start once and only once, which is expressed by the following formula:

$$\sum_k \left(\sum_t x_{k,t} - 1 \right)^2 = 0. \tag{3}$$

Secondly, there can be only one operation running at a given machine at any time, i.e.:

$$\sum_m \left(\sum_{k,t,k',t' \in R_m} x_{k,t} x_{k',t'} \right) = 0, \tag{4}$$

where R_m is a union of two sets $R_m = A_m \cup B_m$. A_m is a set that constraints operation $o_{k'}$ to start on a machine m if operation o_k is still running on the machine, and B_m is a set that constraints two operations from starting at the same time unless one of their times is 0:

$$\begin{aligned} A_m = \{&(k,t,k',t') : (k,k') \in I_m \times I_m, \\ &k \neq k', 0 \leq t, t' \leq T, 0 < t' - t < l_k\}, \\ B_m = \{&(k,t,k',t') : (k,k') \in I_m \times I_m, \\ &k < k', t' = t, l_k > 0, l_{k'} > 0\}. \end{aligned}$$

In this notation we denote l_k as the processing time of operation k and I_m as the set of all operations that have to be processed on the machine m.

The last constraint is defined so that the original order of the operations is kept for all the operations for every job in a given instance:

$$\sum_{n=1}^{J} \left(\sum_{\substack{k_{n-1} < k < k_n \\ t + l_k > t'}} x_{k,t} x_{k+1,t'} \right) = 0. \tag{5}$$

If we name h_1, h_2, h_3 as the constraint objectives (3), (4) (5) we get:

$$h_1(x) = \sum_k \left(\sum_t x_{k,t} - 1 \right)^2, \tag{6}$$

$$h_2(x) = \sum_m \left(\sum_{k,t,k',t' \in R_m} x_{k,t} x_{k',t'} \right), \tag{7}$$

$$h_3(x) = \sum_{n=1}^{J} \left(\sum_{\substack{k_{n-1} < k < k_n \\ t + l_k > t'}} x_{k,t} x_{k+1,t'} \right), \tag{8}$$

where x is a vector of length R representing all possible variables $x_{k,t}$. If the values of all three objectives are equal 0, then all the

constraints are satisfied, i.e., there is a feasible schedule and the makespan of this schedule is less or equal to T.

The Job Shop Scheduling Problem is not only limited to finding a feasible schedule, but also optimizing its makespan. We can take advantage of the time-indexed representation by deriving an additional term that will put a penalty favouring any optimal schedule over any non-optimal schedule.

Suppose that for our J jobs in a JSSP instance an optimal schedule finishes at a time τ. Since any job finishes when its last operation $o_{k_1}, o_{k_2}, \ldots, o_{k_J}$ is complete, we can penalize only the completion time of the last jobs' operations. Each last operation is therefore given a penalty of the form $base^{last\text{-}operation\text{-}completion\text{-}time}$. Let $t_{k_1}, t_{k_2}, \ldots, t_{k_J}$ be the last operations' completion times in the optimal schedule. Let us also choose the base as $J+1$. We can show now that any optimal schedule will be less penalized that any non-optimal schedule.

The penalty that will be given to any optimal schedule takes the form of:

$$\sum_{n=1}^{J}(J+1)^{t_{k_n}}, \quad t_{k_n} \leq \tau, \tag{9}$$

and the most penalized optimal schedule, i.e., the schedule in which all the last operations finish at time τ will be given the penalty:

$$\sum_{n=1}^{J}(J+1)^{\tau} = J(J+1)^{\tau}. \tag{10}$$

Comparing (9) and (10) we can easily see that, indeed:

$$\sum_{n=1}^{J}(J+1)^{t_{k_n}} \leq J(J+1)^{\tau}. \tag{11}$$

Moreover, from (10) we can also see that:

$$J(J+1)^{\tau} < (J+1)(J+1)^{\tau} = (J+1)^{\tau+1}, \tag{12}$$

which tells us that the most penalized optimal schedule is always less penalized than an operation that completes at a smallest non-optimal time $\tau+1$. If we denote by $t'_{k_1}, t'_{k_2}, \ldots, t'_{k_J}$ the last operations' completion times of this non-optimal schedule (i.e., $\exists n : t'_{k_n} = \tau+1$), we can enhance (12) and write:

$$J(J+1)^{\tau} < (J+1)^{\tau+1} < \sum_{n=1}^{J}(J+1)^{t'_{k_n}}, \tag{13}$$

which ends our proof, that the most penalized optimal schedule is always less penalized than any non-optimal schedule.

Taking advantage of the derived penalties we can define an additional objective:

$$h_4(x) = \sum_{n=1}^{J}(J+1)^{t_{k_n}}, \tag{14}$$

which has the lowest values if the vector x produces an optimal schedule.

4. Quantum computing for the JSSP

4.1. Quantum computing

Preliminary to describing our approach of using QAOA to solve JSSP, it is useful to draft some basic concepts of quantum computing as it is a relatively new field of computer science. In the following subsection we will describe only the necessary ideas, an interested reader is referred to more comprehensive handbooks, e.g. Nielsen & Chuang (2002)

A qubit is a name for any two-level quantum system. A typical representation of a qubit takes the form of a ket

$$|\psi\rangle = a|0\rangle + b|1\rangle \tag{15}$$

by which we mean that we expect the qubit to be in the base state $|0\rangle$ with probability a^2 and to be in the base state $|1\rangle$ with probability b^2. This phenomena is called superposition. The parameters a and b are probability amplitudes of the qubit and satisfy the normalization criterion $a^2+b^2=1$. A system composed of R qubits is represented by

$$a_0|00\ldots00\rangle + a_1|00\ldots01\rangle + \cdots + a_{2^R}|11\ldots11\rangle, \tag{16}$$

where we write $|00\ldots00\rangle$ as an abbreviated form of a tensor product $|0\rangle\otimes|0\rangle\otimes\cdots\otimes|0\rangle\otimes|0\rangle$ and a_r is the probability amplitude of a corresponding base state. A qubit (or a system composed of many qubits) is in a given and prepared state and quantum state preparation is an essential subroutine for quantum computing. The only way to acquire knowledge about the qubit or the quantum system (i.e. there are hundreds of qubits) is to perform a measurement. However, the measurement does not give us complete knowledge about the state because once the qubits are measured, they always collapse into one of their base states. If we can prepare the same state many times, we can alleviate this problem by performing the measurement many times hence estimating the amplitudes up to some error.

Another useful way of describing a quantum state, which origins from quantum mechanics where it describes the total energy of a system, is by specifying its Hamiltonian. In quantum computing, a Hamiltonian is often user-defined and acts as an operator for an expected value measurement for some desired state. The expectation value of a quantum system (often referred to as energy) is mathematically written as

$$E = \langle\psi|H|\psi\rangle. \tag{17}$$

The Hamiltonian H is usually composed of scalar-weighted Pauli gates (Pauli, 1927), which in this case are used as the operators of orthogonal measurements. By its definition, the Hamiltonian can aggregate any function which can be described as a sum of binary clauses (Hadfield, 2021). In many quantum optimization algorithms the strategy is to aggregate into the Hamiltonian functions of interest e.g. functions describing constraints or the cost function and evolve an initial quantum state such that the expected value of the Hamiltonian is minimal. The state evolution is done using so-called quantum gates. Physically they take various forms which depend on quantum computer architecture, but mathematically they can always be described as unitary, linear transformations.

4.2. Quantum approximate optimization algorithm

Having formulated the JSSP, we can now describe the Quantum Approximate Optimization Algorithm and show a straightforward and intuitive transformation which takes the already derived objectives to cost Hamiltonians needed by QAOA.

Looking at the QAOA in a general way, suppose a combinatorial optimization problem is given. If we can define the problem objective function as a sum of finite number of clauses $C_\alpha(x)$ where $x \in \{0,1\}^R$ is a binary string of length R,

$$C(x) = \sum_{\alpha} C_\alpha(x), \tag{18}$$

then we can use QAOA to find an approximate solution for the problem with a given probability. To this end, we need to convert the clauses into quantum Hamiltonians by replacing the binary variables x_r, $r \in \{1,2,\ldots,R\}$, with spin variables s_r,

$$x_r = \frac{1-s_r}{2}, \tag{19}$$

and promoting each spin variable s_r to a Pauli-Z matrix σ_r^z. As a result, we obtain a cost Hamiltonian

$$H_C = C(\sigma^z). \tag{20}$$

The QAOA algorithm alternately applies the cost Hamiltonian and a mixing Hamiltonian H_B to the equal superposition $|+\rangle^{\otimes R}$ state p times, thus achieving a final state ψ:

$$|\psi_p(\vec{\gamma}, \vec{\beta})\rangle = e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_B} e^{-i\gamma_1 H_C} |+\rangle^{\otimes R}, \tag{21}$$

where the variables γ and β play the role of variational parameters to be optimized by a classical algorithm, and the mixing Hamiltonian H_B usually takes the form of a sum of Pauli-X matrices (Farhi et al., 2014):

$$H_B = \sum_{r=1}^{R} \sigma_r^x. \tag{22}$$

The goal of QAOA is to find such optimal parameters γ^* and β^*, so that the expected value

$$F_p(\vec{\gamma}, \vec{\beta}) = \langle \psi_p(\vec{\gamma}, \vec{\beta}) | H_C | \psi_p(\vec{\gamma}, \vec{\beta}) \rangle \tag{23}$$

is maximized (or minimized). In our work we will look for the so-called ground state of the cost Hamiltonian, hence we will aim to minimize the expected value. The expected value of the cost Hamiltonian will be referred to as energy in the following sections.

4.3. QAOA For the JSSP

Let us take the objectives (6) to (8) and (14) we derived beforehand and, using the formula (19), replace all the binary variables from vector x with spin variables s. Subsequently, let us promote the spin variables to Pauli-Z matrices. We can now treat the objective function expressed by the formula:

$$H_C(\sigma^z) = h_1(\sigma^z) + h_2(\sigma^z) + h_3(\sigma^z) + h_4(\sigma^z) \tag{24}$$

as the cost Hamiltonian (20). This cost Hamiltonian can be then embedded into the QAOA to find the solution of the JSSP.

5. Experiments and results

Due to computational power limitations required for quantum simulations, the following analysis is conducted on a toy JSSP instance. Our algorithm was launched using the Atos myQLM quantum simulator framework, allowing users to run quantum noiseless simulations up to several tens of cubits depending on the classical computer performance. The basic instance consists of 3 jobs and each of them contains between 1 and 2 operations with processing times ranging from 1 to 2 units. The number of machines is 3. The makespan for the optimal schedule for this instance is $\tau^* = 3$. We present the instance on a disjunctive graph in Fig. 1.

5.1. Patterns in the parameter space

For a given QAOA depth we can always find such variational parameters in the non-convex space that lie near the global optimum and further optimization gives no significant advantage towards lower energy. Let us call them the optimal variational parameters.

The succeeding optimal variational QAOA parameter sequences $\beta^* = \beta_1^*, \dots, \beta_i^*, \dots, \beta_p^*$ and $\gamma^* = \gamma_1^*, \dots, \gamma_i^*, \dots, \gamma_p^*$ might form patterns, i.e. their values might increase monotonically with increasing i and also, that they should interpolate the space with increasing p as demonstrated in Zhou, Wang, Choi, Pichler, & Lukin (2020). These properties are used in so-called *educated guess strategy* which is known to significantly speed up the process of finding

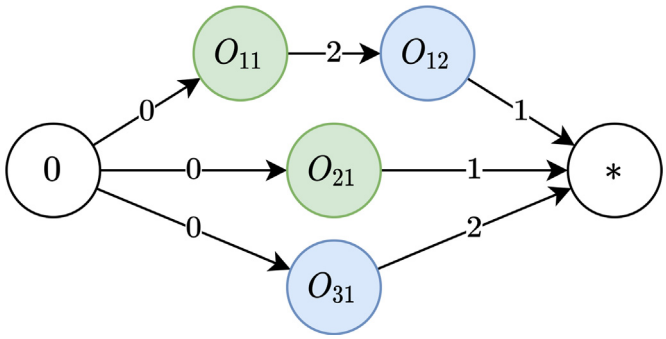

Fig. 1. The basic synthetic JSSP instance - the numbers on the directed edges represent processing tfimes of each operation, and the vertex color coding substitute undirected edges by marking operations that have to processed on the same machine.

optimal parameters (Vikstål et al., 2020; Zhou et al., 2020). The educated guess strategy will be discussed in the next subsection. To confirm that in the case of the JSSP's cost Hamiltonian these properties appear as well, we started our series of experiments by finding optimal parameters for an arbitrary p. Having found the optimal parameters we could plot landscapes of pairs (β_i, γ_i) ranging both β_i and γ_i from -2π to 2π. Fig. 2 shows the energy landscapes with monotonically traveling optimal pairs, which empirically confirms our assumptions.

5.2. Educated guess strategy

The educated guess interpolation strategy proposed in Zhou et al. (2020) assumes patterns in the parameter space. These patterns allow to take the optimal parameter sequences of depth p used by a quantum circuit of depth p, interpolate them to obtain parameter sequences of depth $p+1$ and pass them to the longer circuit of depth $p+1$. This is beneficial because (i) the longer the circuit is, the higher is the probability of obtaining a better solution (i.e., solution with low energy), and (ii) it takes more time to optimize parameters for longer circuits, so starting from initial points which are close to optimal ones can speed up the process of optimization.

In our implementation, we set p to a low value ($p = 3$) and then use multi-start strategy to initialize many random QAOA parameters and pass them as starting points to the local optimizer. The local optimizer was chosen to be the Constrained Optimization BY Linear Approximation (COBYLA) (Powell, 1994) algorithm, implemented in SciPy library, version 1.5.4 (Virtanen et al., 2020), and the function tolerance we set to 0.001. The number of starting points is empirically determined to be 500. After running the optimizer algorithm, we select several (5∼10) points, resulting in the lowest cost Hamiltonian energy. We interpolate them, as it was described in Zhou et al. (2020) to obtain new points of higher dimensionality. These points are then fed as parameters to the QAOA circuit of depth $p+1$ and optimized again by the COBYLA algorithm. We repeat the optimization-interpolation steps until a desired Hamiltonian energy is achieved on one of these points.

We start our series of experiments by defining the cost Hamiltonian as a sum of the feasibility constraints (6) to (8) only. This means that no penalty will be given on non-optimal schedules but the Hamiltonian will be less complex. In Fig. 3 we present a comprehensive visualisation of the behaviour of the variational parameters together with energy probabilities of the Hamiltonian. The data was collected using the aforementioned algorithm to the JSSP, choosing T to equal 4. We can see that both γ and β tend to monotonically increase their values in the domain of a single circuit and that they tend to interpolate the space in the domain of

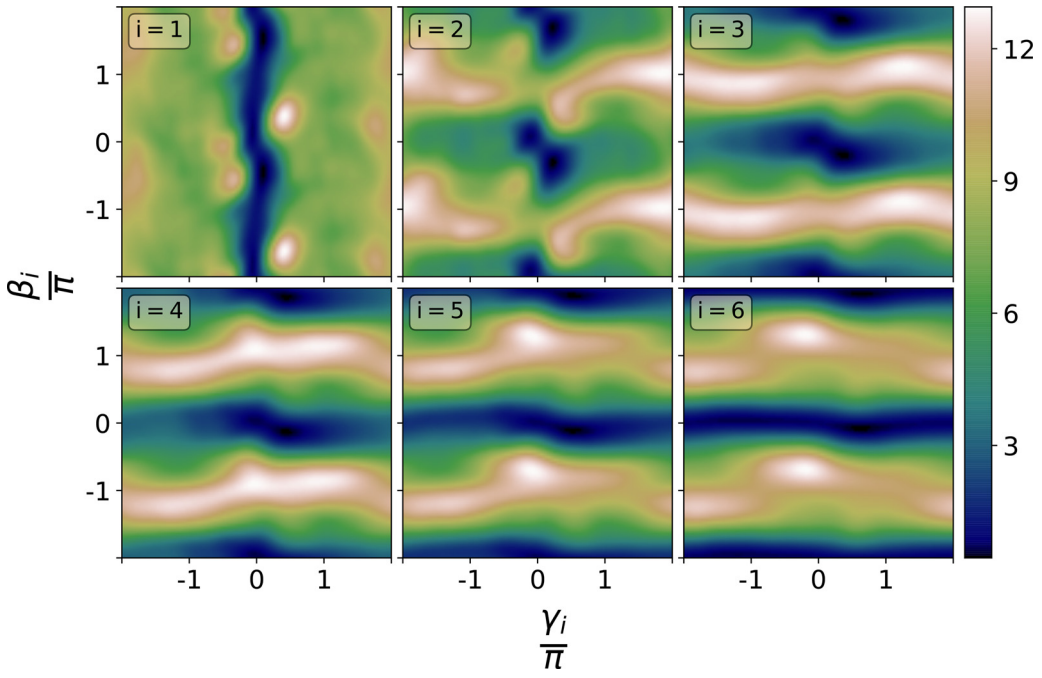

Fig. 2. Energy landscapes for the JSSP in a function of variational parameters β and γ for $p = 6$. On a single subplot we show only selected pairs of parameters (β_i, γ_i), while the remaining $(\beta_j, \gamma_j), i \neq j$ are set to constant, optimal values. We can observe repetition of the landscape energy values, which are the result of symmetries in variational parameters.

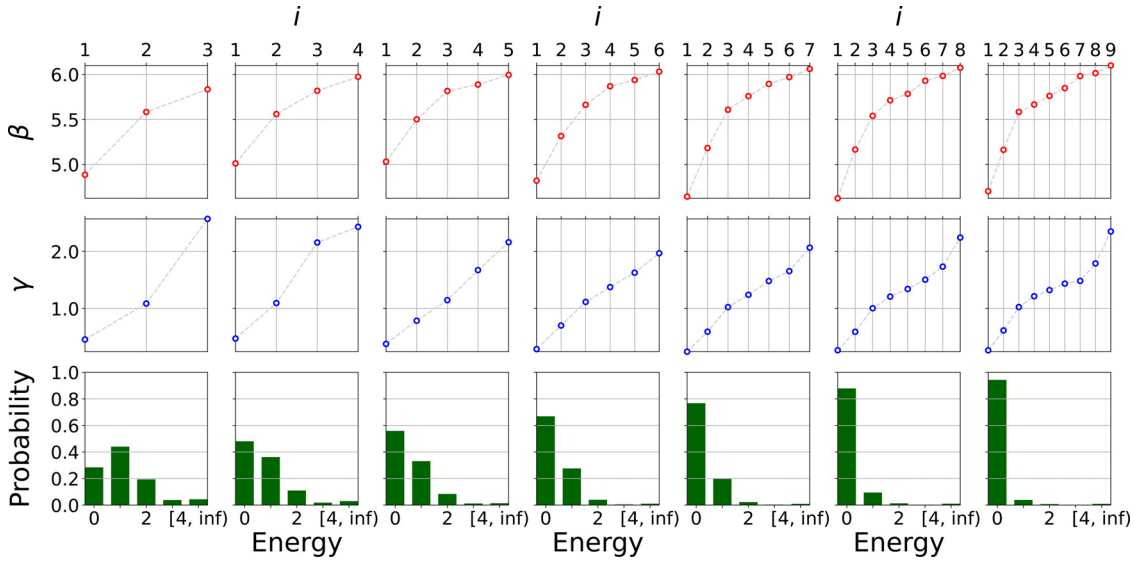

Fig. 3. Results for solving the toy JSSP instance using the educated-guess strategy with the Hamiltonian constructed of feasibility constraints only. The plots show 7 pairs of optimal parameters β and γ and corresponding energy probabilities of the cost Hamiltonian (solutions with energy greater than 4 are aggregated into one bar). Every next column represents one unit longer circuit than the previous one. Every next circuit was fed with previously found, interpolated and then optimized parameters (except the first circuit of depth $p = 3$ plotted on the first column, where the initial parameters were randomly chosen and optimized without interpolation).

increasing circuit depth p. We can also see that, indeed, the circuit depth affects the energy probabilities reaching over 90% chance of measuring the feasible solution when the depth $p = 9$.

We now proceed with the experiments by adding the fourth objective (14), which penalizes non-optimal schedules, to the Hamiltonian. Looking at the Fig. 4 we can see that the results are similar to the ones presented in Fig 3. Variational parameters make twin landscapes and they form similar patterns that interpolate the space with increasing circuit depth. Since now, the JSSP operations are penalized on their completion time, the minimal energy is almost always greater than 0 (unless all the processing times are 0, which is not the case in the instance). Comparing the energy probabilities of circuits with the same depth from both figures, we can see that they are roughly similar, e.g., the summed probability of measuring energies in range $[0.0, 0.5)$ and $[0.5, 1.0)$ in Fig. 3 with circuit of depth $p = 6$ is approximately equal to 0.64, which is comparable to probability of measuring energy value of 0 in Fig. 4 with circuit of depth $p = 6$. This leads to a conclusion that using the more complex Hamiltonian, seems to have little effect on the probabilities of measuring a solution with a low energy. This is good news since it also means that we can obtain low-makespan solution with no additional computational effort. The relation between energy and makespan will be the matter of the next subsection.

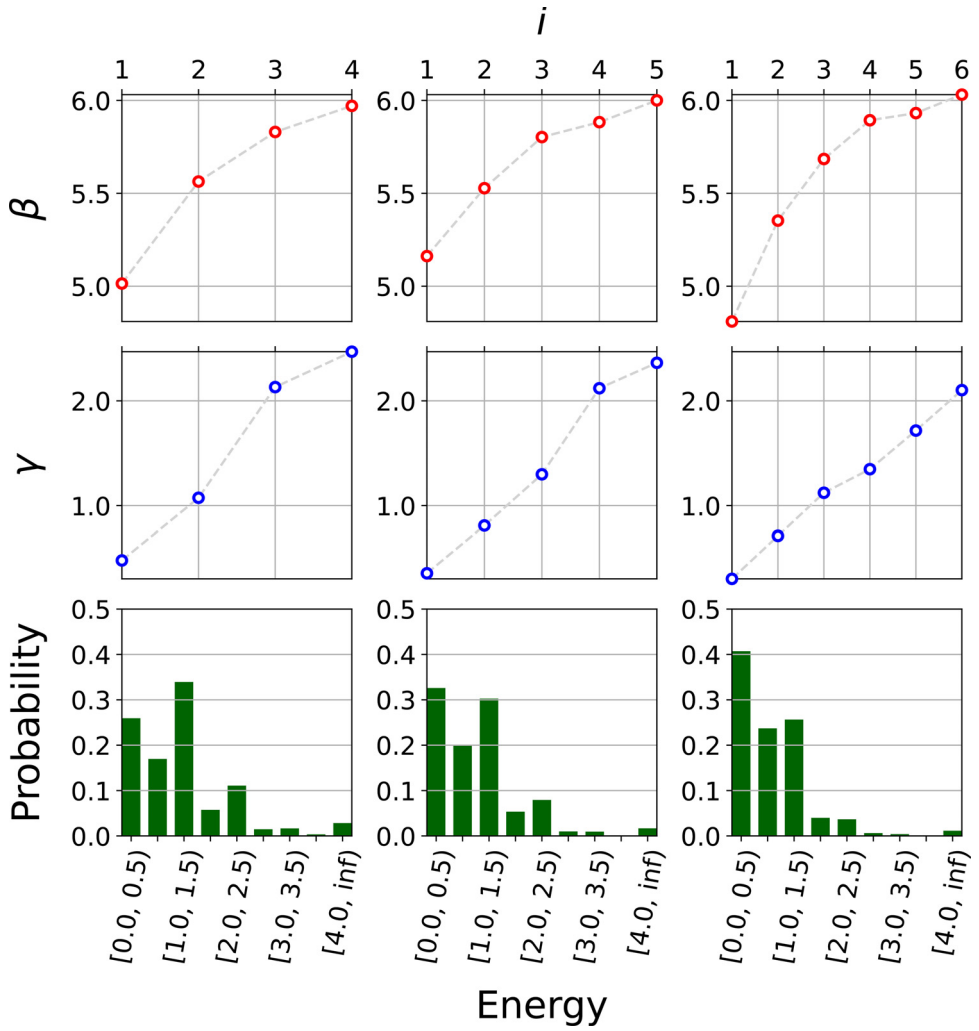

Fig. 4. Results for the JSSP instance using the educated-guess strategy with the Hamiltonian also penalizing solutions with high makespan. The subplots show 3 pairs of optimal parameters β and γ as well as corresponding energy probabilities of the cost Hamiltonian (solutions with close energy values are aggregated into single bins of range of 0.5).

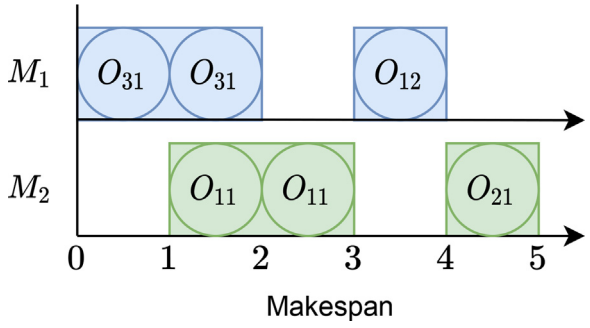

Fig. 5. A randomly sampled solution of the toy JSSP instance plotted on a Gantt chart. The operations O_{31} and O_{11} have processing time equal 2. The schedule generates energy 1.27 with the final makespan 5.

5.3. Performance analysis

We start the following analysis by randomly sampling a feasible solution for our toy instance. We plot the solution in Fig. 5. Its energy is approximately 1.27 and its makespan is 5. Based on this random solution, in the following series of experiments we set the deadline $T = 5$ and launch our energy-minimization algorithm to see how the mean energy decrease affects the probability of obtaining a solution with different (preferably lower) makespan. Afterwards, we lower the deadline to $T = 4$ in order to investigate the impact that hardening the constraints has on the probabilities of obtaining solutions with given energies.

We present the relation between energy and makespan in Fig. 6. The circuit used to produce data for this figure had depth $p = 5$. We can see a clear pattern that the lower the energy is, the higher is the probability of obtaining a feasible solution with a low makespan. Moreover, only the lowest range of energies correspond with the optimal solution (makespan 3). With circuit depth $p = 5$ approximately half of the solutions are infeasible, however. This is an expected behaviour that comes from the nature of quantum computing and can be reduced by iterative increasing the depth of the circuit, as was described in Section 5.2.

Interestingly enough, restricting the deadline to $T = 4$ resulted in rise in probability of measuring solutions with low energy, therefore lower makespan. This property might be advantageous when using any strategy based on repeated querying the circuit, e.g., starting with large T, measuring any feasible solution with makespan lower than T, and then setting this makespan value as T for the next iteration. Consequently, this would, again, accelerate the process of finding the optimal solution.

When considering performance we cannot underestimate the importance of time consumption, so in Fig. 7 we show the opti-

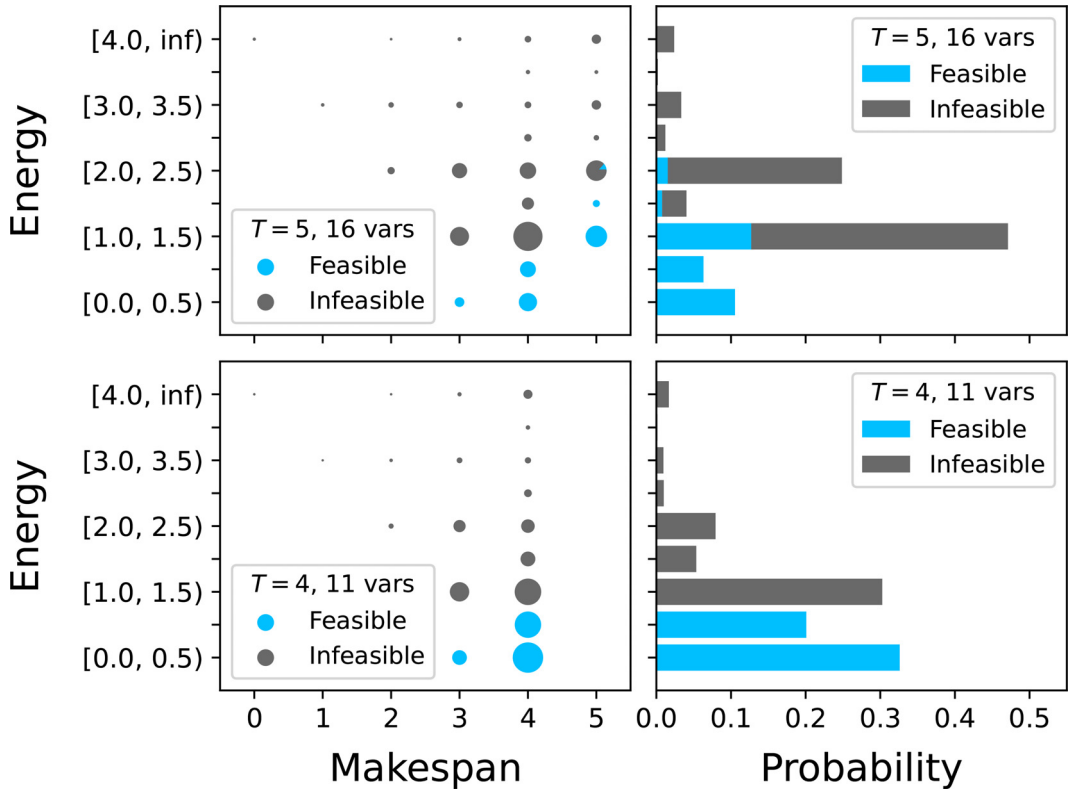

Fig. 6. Left hand-side: relationship between solution makespan and energy. Size of the markers are proportional to the probability of measuring a solution with given makespan and energy range. Right hand-side: marginal distribution of energy over the makespan. The top row present the results for deadline $T = 5$, while the bottom row: $T = 4$.

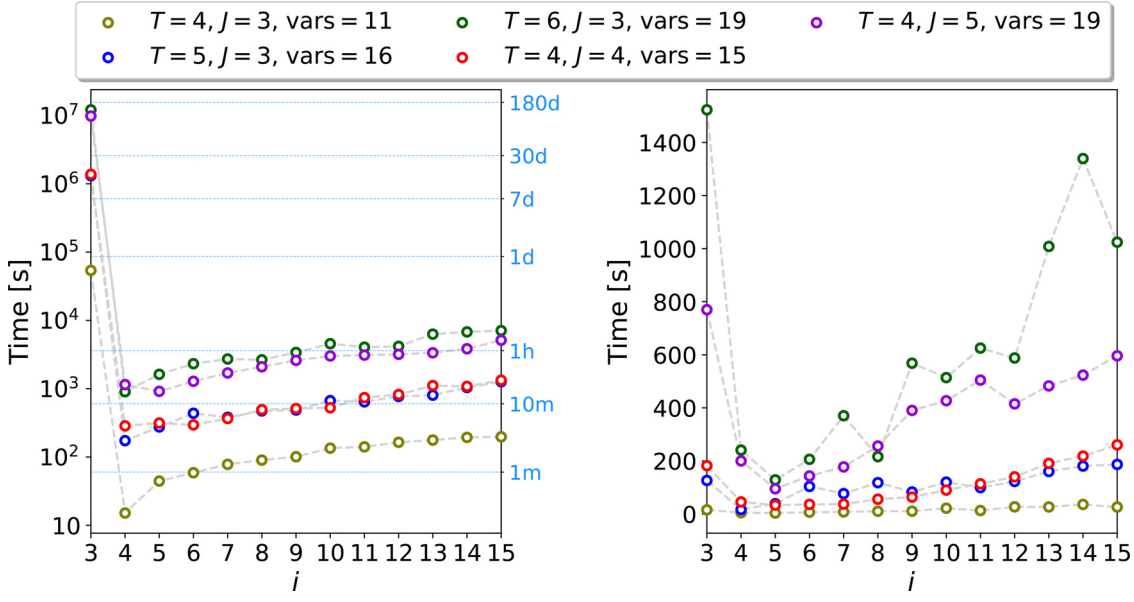

Fig. 7. Time needed for optimization of the variational parameters γ and β in the function of depth of simulated quantum gate-model circuit. The left hand-side shows the summed optimization time for all initial β and γ points per ith steps over all threads which were used. The right hand-side shows the optimization time of a single initial point, which resulted in the optimal solution.

mization time of the toy instance and its slight modifications (e.g. increased number of jobs). Then, we have enhanced the experiments up to 19 variables. We can observe an advantage of using the educated-guess strategy to solve the JSSP over a regular QAOA. Finding the optimal parameters is hard even for low-depth circuits, but if the optimal parameters are found and interpolated, the time needed to optimize them decreases significantly.

The second most apparent observation is the height of the overall duration time of the algorithm. For the toy instance, which is easily solvable by hand in several minutes by humans, it takes even days for a computer to solve, which might seem disappointing. Note, however, that in this paper, all the experiments have been made using a quantum simulator, meaning that all possible states (which number grows exponentially with the number of variables)

had to be processed sequentially on a classical computer. Thus, experimenting with only toy scheduling problem instances is necessary today when working with quantum simulators since quantum evolution requires the multiplication of quantum states by large matrices 2^{2n}. If we were to analyze with benchmarks from scientific literature, e.g. the FT06 benchmark for JSSP as we have demonstrated on a real quantum annealer, we would need at least hundreds of variables, which means that we would need RAM of around $2^{2 \cdot 100 \cdot 4}$ bytes on a classical computer to keep the quantum evolution matrices. Nevertheless, nowadays, several real gate-based NISQ devices with up to hundreds of qubits, e.g. recent IBM Q systems, are capable of handling this complexity with quantum computations.

We can also see, that the optimization time not only depends on the number of variables, but also depends on T or the instance parameters such as number of jobs. The reason for that is that QAOA's circuit depth is no longer dependent on the instance size n, but instead it is a function of the parameter p (Zhou et al., 2020). Even though these are not directly comparable, it is shown in Farhi & Harrow (2019) that it is enough to obtain quantum advantage in some cases.

6. Conclusions

This paper presented how to successfully implement a widely studied QAOA to solve a reference scheduling problem. We demonstrated how to efficiently apply QAOA to the well-known JSSP to find the optimal solution and pave the way for solving more complex scheduling problems in the future with more powerful and hybrid classic-quantum NISQ devices. Additionally, we investigated the behaviour of our energy-minimization algorithm in a series of experiments and demonstrated the relation between energy and makespan in achieving feasible and infeasible solutions. We discussed our experiences gained from real computational experiments and showed appropriate technical steps for quantum application developers interested in quantum simulation environments. All the presented results can be quickly adopted and extended by gate-model application developers, especially for initial testing and experimental verification of new quantum-based approaches for intensive quantum simulations. This paper can also be seen from the performance evaluation perspective of classical high-performance computer (HPC) systems and requirements for computing resources as new quantum-inspired and hybrid Quantum-HPC methodologies emerge.

Acknowledgments

This research has been partially supported by the statutory funds of Poznan University of Technology, Poznan Supercomputing and Networking Center affiliated to Institute of Bioorganic Chemistry Polish Academy of Sciences, and grant QATM, DOB-SZAFIR/01/B/023/01/2021. The calculations were performed at the Poznan Supercomputing and Networking Center. We express our gratitude to three anonymous referees whose valuable comments helped us significantly improve the quality of our paper.

References

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science, 34*(3), 391–401.

Aharonov, D., Van Dam, W., Kempe, J., Landau, Z., Lloyd, S., & Regev, O. (2007). Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Journal on Computing, 37*(1), 166–194. https://doi.org/10.1137/S0097539705447323.

Artigues, C., Lopez, P., & Ayache, P. (2005). Schedule generation schemes for the job-shop problem with sequence-dependent setup times: Dominance properties and computational analysis. *Annals of Operations Research, 138*, 21–52.

Balas, E., & Vazacopoulos, A. (1998). Guided local search with shifting bottleneck for job shop scheduling. *Management Science, 44*(2), 262–275.

Benioff, P. (1980). The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics, 22*(5), 563–591. https://doi.org/10.1007/BF01011339.

Bernstein, E., & Vazirani, U. (1993). Quantum complexity theory. In *Proceedings of the twenty-fifth annual acm symposium on theory of computing*. In *STOC '93* (pp. 11–20). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/167088.167097.

Bernstein, E., & Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal on Computing, 26*(5), 1411–1473.

Błażewicz, J., Domschke, W., & Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research, 93*(1), 1–33.

Błażewicz, J., Dror, M., & Węglarz, J. (1991). Mathematical programming formulations for machine scheduling: A survey. *European Journal of Operational Research, 51*(3), 283–300.

Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., Sterna, M., & Węglarz, J. (2019). *Handbook on scheduling: From theory to practice*. Cham: Springer.

Błażewicz, J., Pesch, E., & Sterna, M. (2000). The disjunctive graph machine representation of the job shop problem. *European Journal of Operational Research, 127*(2), 317–331.

Boixo, S., Albash, T., Spedalieri, F., Chancellor, N., & Lidar, D. (2013). Experimental signature of programmable quantum annealing. *Nature Communications, 4*, 2067.

Brucker, P., & Jurisch, B. (1993). A new lower bound for the job shop scheduling problem. *European Journal of Operational Research, 64*(2), 156–167.

Carlier, P., & Pinson, E. (1994). Adjustment of heads and tails for the job shop problem. *European Journal of Operational Research, 78*(2), 146–161.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., … Coles, P. J. (2021). Variational quantum algorithms. *ArXiv* abs/2012.09265.

Coveney, P. V., & Highfield, R. R. (2020). From digital hype to analogue reality: Universal simulation beyond the quantum and exascale eras. *Journal of Computational Science, 46*, 101093. https://doi.org/10.1016/j.jocs.2020.101093. https://www.sciencedirect.com/science/article/pii/S1877750320300545

Crooks, G. E. (2018). Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv preprint* arXiv:1811.08419.

De Palma, G., Marvian, M., Rouzé, C., & França, D. S. (2022). Limitations of variational quantum algorithms: Aquantum optimal transport approach. *arXiv preprint* arXiv:2204.03455.

Deutsch, D. (1985). Quantum theory as a universal physical theory. *International Journal of Theoretical Physics, 24*(1), 1–41. https://doi.org/10.1007/BF00670071.

Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond. A, 439*, 553–558. https://doi.org/10.1098/rspa.1992.0167.

Dorndorf, U., Pesch, E., & Phan-Huy, T. (2002). Constraint propagation and problem decomposition: A preprocessing procedure for the job shop problem. *Annals of Operations Research, 115*, 125–145.

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv:1411.4028: Quantum Physics*.

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. https://arxiv.org/abs/quant-ph/0001106. 10.48550/ARXIV.QUANT-PH/0001106.

Farhi, E., & Harrow, A. W. (2019). Quantum supremacy through the quantum approximate optimization algorithm. *arXiv:1602.07674: Quantum Physics*.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics, 21*(6), 467–488. https://doi.org/10.1007/BF02650179.

Fortnow, L., & Rogers, J. (1999). Complexity limitations on quantum computation. *Journal of Computer and System Sciences, 59*(2), 240–252.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. https://arxiv.org/abs/quant-ph/9605043. doi:10.48550/ARXIV.QUANT-PH/9605043.

Hadfield, S. (2021). On the representation of boolean and real functions as hamiltonians for quantum computing. *ACM Transactions on Quantum Computing, 2*(4), 1–21.

Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., … Boixo, S., et al., (2021). Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics, 17*(3), 332–336.

Haupt, R. (1989). A survey of priority-rule based scheduling. *OR Spektrum, 11*(1), 3–16.

van Hoorn, J. (2018). The current state of bounds on benchmark instances of the job-shop scheduling problem. *Journal of Scheduling, 21*(1), 127–128.

Humble, T., McCaskey, A., Bennink, R., Billings, J., D'Azevedo, E., Sullivan, B. D., … Seddiqi, H. (2013). An integrated programming and development environment for adiabatic quantum optimization. *Computational Science & Discovery, 7*, 015006.

Jain, A., & Meeran, S. (1999). Deterministic job shop scheduling: Past, present and future. *European Journal of Operational Research, 113*(2), 390–434.

Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Physical Review E, 58*(5), 5355–5363. https://doi.org/10.1103/physreve.58.5355.

Khumalo, M. T., Chieza, H. A., Prag, K., & Woolway, M. (2022). An investigation of ibm quantum computing device performance on combinatorial optimisation problems. *Neural Computing and Applications*, 1–16.

Kitaev, A. Y. (1997). Quantum computations: Algorithms and error correction. *Russian Mathematical Surveys, 52*(6), 1191.

Kurowski, K., Węglarz, J., Subocz, M., Różycki, R., & Waligóra, G. (2020). Hybrid quantum annealing heuristic method for solving job shop scheduling problem. In V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, & J. Teixeira (Eds.), *Computational science – iccs 2020* (pp. 502–515). Cham: Springer International Publishing.

Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics, 2*, 5. https://doi.org/10.3389/fphy.2014.00005. https://www.frontiersin.org/article/10.3389/fphy.2014.00005.

Magann, A. B., Arenz, C., Grace, M. D., Ho, T.-S., Kosut, R. L., McClean, J. R., … Sarovar, M. (2021). From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms. *PRX Quantum, 2*(1), 010101.

Magann, A. B., Rudinger, K. M., Grace, M. D., & Sarovar, M. (2022). Lyapunov-control-inspired strategies for quantum combinatorial optimization. *Physical Review A, 106*(6), 062414.

Nielsen, M. A., & Chuang, I. (2002). Quantum computation and quantum information.

Pauli, W. (1927). Zur quantenmechanik des magnetischen elektrons. *Zeitschrift für Physik A Hadrons and Nuclei, 43*(9), 601–623.

Pezzella, E., & Merelli, E. (2000). Tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research, 120*(2), 297–310.

Powell, M. J. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis* (pp. 51–67). Springer.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79. https://doi.org/10.22331/q-2018-08-06-79.

Radzihovsky, M., Murphy, J., & Mason, S. (2019). A qaoa solution to the traveling salesman problem using pyquil. https://cs269q.stanford.edu/projects2019/radzihovsky_murphy_swofford_Y.pdf.

Ralli, A., Love, P. J., Tranter, A., & Coveney, P. V. (2021). Implementation of measurement reduction for the variational quantum eigensolver. *Physical Review Research, 3*(3), 033195. https://doi.org/10.1103/PhysRevResearch.3.033195.

Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science* (pp. 124–134). https://doi.org/10.1109/SFCS.1994.365700.

Simon, D. R. (1994). On the power of quantum computation. In *Proceedings of the 35th annual symposium on foundations of computer science*. In *SFCS '94* (pp. 116–123). USA: IEEE Computer Society. https://doi.org/10.1109/SFCS.1994.365701.

Stilck França, D., & Garcia-Patron, R. (2021). Limitations of optimization algorithms on noisy quantum devices. *Nature Physics, 17*(11), 1221–1227.

Tabi, Z., El-Safty, K. H., Kallus, Z., H'aga, P., Kozsik, T., Glos, A., & Zimbor'as, Z. (2020). Quantum optimization for the graph coloring problem with space-efficient embedding. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 56–62.

Vaessens, R., Aarts, E., & Lenstra, J. (1996). Job shop scheduling by local search. *INFORMS Journal of Computing, 8*(3), 302–317.

Venturelli, D., Marchand, D. J. J., & Rojo, G. (2016). Quantum annealing implementation of job-shop scheduling. *arXiv:1506.08479: Quantum Physics*.

Vikstål, P., Grönkvist, M., Svensson, M., Andersson, M., Johansson, G., & Ferrini, G. (2020). Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Physical Review Applied, 14*(3), 034009. https://doi.org/10.1103/PhysRevApplied.14.034009.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … SciPy 1. 0 Contributors (2020). Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods, 17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

Yao, A. C.-C. (1993). Quantum circuit complexity. In *Proceedings of 1993 ieee 34th annual foundations of computer science*. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science* (pp. 352–361). https://doi.org/10.1109/SFCS.1993.366852.

Zhang, C., Li, P., Rao, Y., & Guan, Z. (2008). A very fast TS/SA algorithm for the job shop scheduling problem. *Computers & Operations Research, 35*(1), 282–294.

Zhou, L., Wang, S.-T., Choi, S., Pichler, H., & Lukin, M. D. (2020). Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *Physical Review X, 10*(2), 021067. https://doi.org/10.1103/PhysRevX.10.021067.