3rd International Conference on Engineering, Natural Sciences, and Technological Developments (ICENSTED 2026)

July 3-6, 2026

Application of Quantum Algorithms on Scheduling Problems

Seda Ozmutlu[1], Burcu Caglar Gencosman[1]

[1]Department of Industrial Engineering, Bursa Uludağ University, Bursa, Türkiye
Corresponding author: Burcu Caglar Gencosman (e-mail: burcucaglar@uludag.edu.tr)

Abstract

Scheduling is a prevalent research area in combinatorial optimization and operations research, where exact and heuristic methods are available in literature. However, most scheduling problems are NP-hard, and as the problem size increases with system constraints and an increasing number of jobs and machines, the computation time also increases exponentially. This makes exact methods, and even heuristics, impractical for real-life scheduling problems. Modern research explores using Quantum Computing to solve scheduling problems for faster solutions, and this paper presents a brief review of single-machine and shop-type scheduling problems solved by Quantum methods.

Keywords: Quantum Computing, Single Machine Scheduling Problem, Shop-type scheduling problem

1. INTRODUCTION

The scheduling problem has long been one of the most popular problems in the industrial engineering discipline. There is a myriad of papers on a wide range of problems, ranging from the simplest, such as the single-machine problem, to the more complex job-shop and open-shop problems. Many solution methods have been applied, including mathematical programming, heuristics, supervised learning, and artificial intelligence. Due to the structure of scheduling problems, as the problem size grows, complexity quickly increases to exponential levels, rendering classical optimization techniques computationally expensive or ineffective even for small instances. Conventional techniques are no longer sufficient to solve many practically relevant scheduling problems optimally, despite the continually increasing computing power.

Quantum computing has been a facilitator in solving intractable problems since the 2000s, as it can provide an exponential speedup by employing quantum-mechanical principles, including superposition and entanglement [1]. Quantum computing has been applied to many areas, including cryptography, financial modeling, risk analysis, communication, and combinatorial optimization [2]. In search of a faster alternative, researchers began applying quantum computing to solve scheduling problems.

In this paper, we focus on the application of quantum computing to single-machine and shop-type scheduling problems. The rest of the paper is organized as follows. Section 2 introduces the concept of quantum computing. Section 3 presents literature on quantum methods for single-machine scheduling and shop-type scheduling. Section 4 concludes the paper.

2. QUANTUM COMPUTING AND ALGORITHMS

Quantum computing is a computing technology that originated in quantum physics. Classical computer processor logic is organized in bits with binary number logic. 1 Bit consists of 0-1. Various numbers and characters are created using 0-1s, and operations are performed using 0-1 logic. In classical computers, 0-1 states are defined by whether current passes through transistors. In quantum computing, classical bit logic is replaced by qubits (quantum bits, or qubits). Qubits are also in states 0-1, but they can also exist in a superposition of states at all possible values from 0-1. Superposition is the state in which different states of an object can occur together with certain possibilities. Another fundamental principle of quantum computing is entanglement. Entangled qubits cannot be measured independently of each other; The measurement result of one instantly determines the status of the other. This property is one of the fundamental mechanisms that allow quantum algorithms to evaluate a large number of solutions simultaneously [3].

Calculations that would normally have to be performed separately in many steps with 0s or 1s in a classical device can be completed in a single operation using a qubit in a quantum computer. This can make calculations much

faster. This is the basic accelerating action of quantum computing. However, it should be noted that although a single qubit can exist in the superposition of two classical bits, when one qubit is measured, the measurement actually results in only one classical bit (0-1). Qubits are usually formulated as 0-1 bits multiplied by certain probabilities α and β (Qubit $\psi = \alpha \mid 0 > +\beta \mid 1 >$). A qubit can exist in infinite possible states (superposition) maintaining $\alpha^2 + \beta^2 = 1$.

The relationships between qubits are not just simple logical operators like AND/OR; and are modeled with more complex gates [1]. The action of current passed/did not pass through the transistor is insufficient to run the qubit logic. Since computational logic changes, different technologies are required to handle this logic; For this reason, quantum computer technologies are still in the development stage. Optical systems, superconductors and molecule-based technologies are being tested. Problem solving with quantum computing is carried out with the following steps [4]: (i) initialization of qubits, (ii) formulation of the problem as a quadratic unconstrained binary optimization (QUBO) model, and (iii) execution of the quantum algorithm. Qubits are typically initialized either to the ground state $|0\rangle$, or to an equal superposition state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, where the $\sqrt{2}$ normalization ensures that the total probability sums to unity.

When a combinatorial optimization problem is intended to be solved using classical methodologies, its mathematical modeling is performed first, and the problem is then solved either through this model or via heuristic algorithms. However, in quantum computing, the problem must first be modeled in a way that can be handled by a quantum processor. The transition between the mathematical models of combinatorial optimization problems and quantum algorithms is achieved through QUBO formulation. When constructing a QUBO model, all problem constraints are relaxed and incorporated into the objective function via penalty terms. For example, an equality constraint g(x)=0 is incorporated into the objective function as $M\cdot(g(x))^2$. Here, M is a sufficiently large penalty coefficient. Inequality constraints can be converted into equality constraints using slack variables where necessary and integrated into the objective function in that manner. The QUBO expression is in the form of $\min_{x\in\{0,1\}^n} x^T Q x$. In this expression, $\in \mathbb{R}^{n\times n}$ is the symmetric QUBO matrix and x is the vector of binary decision variables derived from the mathematical model.

2.1. Implementation of the Quantum Circuit

Following the QUBO transformation, quantum circuits are constructed based on the specific quantum algorithm to be utilized. Most algorithms use Ising Hamiltonians. The Ising Hamiltonian form facilitates the translation of problem conditions into the language of quantum mechanics while guiding the execution of the algorithm [5]. Quantum circuits are then constructed based on this Hamiltonian, enabling the implementation of algorithms such as quantum approximate optimization algorithm (QAOA).

At this stage, unitary gates, which constitute the algorithm's internal logic, are applied. These gates vary depending on the algorithm, and quantum circuits are prepared accordingly. For instance, the QAOA prepares circuits using unitary operators derived from cost (HC) and mixer (HM) Hamiltonians, whereas parametric Ansatz circuits are prepared for the variational quantum eigensolver (VQE) algorithm. Different circuit structures are used in various algorithms. The circuit structures of different quantum algorithms vary in terms of initial states, costs, gates, layer structures, search space exploration patterns, and hyperparameters.

2.2. Measurement

All quantum algorithms conclude with a measurement. When measurement occurs, the superposition collapses, and a bit string is obtained (e.g., 11010). To determine the quantum result, many repetitions (shots) are typically performed. A circuit is usually repeated between 1000 and 10000 shots. The bit string measured most frequently is determined as the final quantum output.

2.3. Classical Processing and Interpretation of Results

Measurement results are processed on a classical computer and repeat if a loop is required. The resulting bit string is converted back into the solution of the original optimization problem—optimizing the relevant objective function, and the value is calculated classically.

3. APPLICATION OF QUANTUM COMPUTING TO SCHEDULING PROBLEMS

3.1. The Application of Quantum Algorithms to Single Machine Scheduling Problems

Scheduling has always been an important area of research, since it is closely related to productivity. A scheduling problem deals with a finite set of tasks (jobs) which have to be performed on a limited number of resources (machines) [6]. One of the well-known scheduling problems, perhaps the simplest, is the single-machine scheduling problem. The problem definition is as follows: There exists a single machine m, and a set of jobs $J = \{1, 2, 3, ..., n\}$ to be processed on machine m. Each job $j \in J = \{1, 2, ..., n\}$ is characterized by a processing time p_j, a due date d_j, and a tardiness weight w_j [7]. There also exists a release time for the jobs r_j. For a permutation schedule $\pi = (\pi(1), \pi(2), ..., \pi(n))$, where $\pi(k)$ denotes the position of job j at position k, the completion time of job j at position k is denoted as $C_{\pi(k)} = \sum_{j=1}^{k} p_{\pi(j)}, k = 1, 2, ..., n.$ For notational convenience, C_j denotes the completion time of job j in the schedule π.

The performance measures for single machine scheduling problem (SMSP) are the maximum completion time of the jobs ($C_{\max}$ or makespan), the total flowtime of jobs $F = \sum_{j=1}^{n} F_j$, and lateness and tardiness-related measures [7]. Total tardiness for a schedule can be stated as $T = \sum_{j=1}^{n} T_j$, where $T_j = \max(0, C_j - d_j)$. Weighted performance measures are also sometimes considered. Details of the different variations of the SMSP can be seen in [7].

Most of the variants of the SMSP are solvable in polynomial time. Total Lateness is minimized by shortest procesing time (SPT) dispatching rule, and L_{max} (maximum lateness) and T_{max} (maximum tardiness) are minimized by earliest due date (EDD) dispatching rule [7]. However, more complex variants of the problem also exist, such as those minimizing total tardiness, minimizing total weighted tardiness, and minimizing number of tardy jobs. SMSP with tardiness as the performance measure is known to be NP-hard in the ordinary sense even without release dates or weights, as shown by [8].

Exact algorithms have been proposed for the NP-hard versions of the SMSP. Early research focused on branch-and-bound algorithms, which remain the primary exact solution approach. These methods rely on lower bounds derived from problem relaxations, dominance rules, and partial sequence evaluation [9]. Dynamic programming and pseudo-polynomial approaches have also been explored; however, their applicability is limited to relatively small job sets due to exponential state-space growth [10]. Because of the complexity of exact optimization, research has largely shifted toward heuristic and meta-heuristic methods for medium and large-sized instances. Early constructive heuristics include the apparent tardiness cost (ATC) and modified due date (MDD) heuristics [11], which provide reasonable sequences but rarely achieve near-optimal performance for large problems. Metaheuristics such as tabu search, simulated annealing, genetic algorithms, particle swarm optimization, and ant colony optimization have performed favorably on the NP-hard versions of the SMSP [12]. These approaches are now considered state-of-the-art for large-scale instances.

In recent years, attention has expanded from classical methods to quantum-inspired algorithms. Although still emerging, quantum algorithms have shown promising results for NP-hard scheduling problems. There are a few papers in quantum computing literature studying the SMSP.

Yan et al. [13] investigate the SMSP with uncertainty demand and present a hybrid algorithm, quantum particle swarm optimization (QPSO), combining particle swarm optimization (PSO), and the quantum evolutionary algorithm (QEA) to maximize the total weighting agreement index for all customer orders in a fuzzy environment. Bożejko et al. [14] solve a small-sized SMSP with weighted tardiness using quantum annealing. In a subsequent study, Bożejko et al. [15] address the SMSP with weighted tardiness by determining a lower bound on the objective function using quantum annealing, rather than solving the entire problem, to increase the problem size that can be solved with quantum algorithms. They continue this approach in subsequent studies [16, 17], where they use quantum annealing within a branch-and-bound algorithm to compute lower and upper bounds on the objective function for the SMSP with weighted tardiness.

Wang and Cheng [18] also minimize total weighted tardiness in the SMSP using Grover's quantum search and Trugenberger's quantum optimization algorithms. They have not provided any test problems, so it is not known whether the algorithm provides optimum results. Grange et al. [19] use a combination of Quantum Minimum Finding (obtained from Grover search) and dynamic programming to solve single-machine scheduling problems minimizing total weighted completion time, total weighted completion time with precedence constraints, and total weighted tardiness. They continue the same line of research by adding the total number of late jobs, the total weighted completion time with release dates, and the makespan as performance measures [20]. From this brief yet exhaustive literature review, we observe that the solution to the SMSP using quantum algorithms is in its infancy.

3.2. The Application of Quantum Algorithms to Shop-Type Scheduling Problems

The focus of applying quantum algorithms to scheduling is primarily on the job shop scheduling problem (JSP). JSP is one of the most well-known and challenging combinatorial optimization problems in the field of operations research. It involves scheduling a set of jobs on a set of machines, where each job consists of a specific sequence of operations that must be performed in a predefined order. The problem definition is as follows: There exists m machines $i=\{1,2,...,m\}$, and a set of jobs $J=\{1,2,...,n\}$. Each job $j\in J=\{1,2,...,n\}$ has operations $k=\{1,2,...,p\}$ that have processing times p_{jk}, a due date d_j, and a weight w_j [7]. There also exists a release time for the jobs r_j. A permutation schedule $\pi=(\pi(1),\pi(2),...,\pi(n))$, is created for each machine i where $\pi(l)$ denotes the position of job j at position l, the completion time of job j at position l is denoted as $C_{\pi(l)}=\sum_{j=1}^{l}p_{\pi(j)}$. For notational convenience, C_j denotes the completion time of job j in the schedule π. The operations for the jobs are to be performed within a certain distinct technological route (e.g., operation 1 must finish before operation 2 starts), but each job has a different operation/machine order. Each operation requires a specific machine for a set processing time. Only one operation can be processed on a machine at any given time (no overlaps). The most common goal is to minimize the $C_{\max}$ or makespan, but the total flowtime of jobs $F=\sum_{j=1}^{n}F_j$, and lateness and tardiness-related measures are also popular [7]. Details of the different variations of the JSPs and solution methodologies can be seen in [7]. The flow shop problem is a sub-problem of the JSP, where the technological route of the operations is fixed for all the jobs.

JSP is classified as NP-hard. Thus, larger problems are addressed using heuristics and metaheuristics, such as Genetic Algorithms, Tabu Search, and Simulated Annealing. JSP, although complex in nature, can be easily expressed in terms of binary variables and quadratic constraints, as required by quantum optimization methods. This makes the JSP problem suitable for solution using Quantum Approaches. So, a significant amount of research on solving the JSP with various quantum algorithms has accumulated in the literature. Classified by quantum algorithms, we identify the following studies in the literature on solving the JSP using quantum methods. More details on the methods cannot be included in this paper due to space considerations: QAOA [21–24], Quantum Annealing [3, 25–36], quantum accelerated constraint programming [37], QPSO [38–51], Evolutionary algorithms [52–56], quantum genetic algorithms [57-61], and Variational Quantum Algorithms [21, 62].

4. CONCLUSION

Scheduling problems constitute an important class of problems in the Combinatorial Optimization literature and in the Industrial Engineering discipline. With several important performance measures and a wide variety of system assumptions, even the simplest scheduling problem can be NP-hard, rendering exact methods unproductive, especially in real-time problems. Scheduling problems, whether single-machine or shop-type, have been attempted numerous times with classical algorithms but rarely studied with quantum algorithms. In this paper, we present a brief summary of the application of quantum algorithms to single-machine and shop-type scheduling problems. As future work, we point to applying quantum algorithms to different variations of the scheduling problems.

References

[1] S. K. Sood and Pooja, “Quantum computing review: A decade of research,” IEEE Transactions on Engineering Management, vol. 71, pp. 6662-6676, 2024. doi: 10.1109/TEM.2023.3284689

[2] F. Bova, A. Goldfarb, and R. G. Melko, “Commercial applications of quantum computing,” *EPJ Quantum Technol.*, vol. 8, no. 1, pp. 2, 2021. doi: 10.1140/epjqt/s40507-021-00091-1

[3] A. Ajagekar, K. Al Hamoud, and F. You, “Hybrid classical-quantum optimization techniques for solving mixed-integer programming problems in production scheduling,” *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–16, 2022. doi: 10.1109/TQE.2022.3187367

[4] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz,” *Algorithms*, vol. 12, no. 2, pp. 34, 2019. doi: 10.3390/a12020034

[5] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv*, 2014. doi: 10.48550/arXiv.1411.4028

[6] N. Vakhania, F. Werner, K. J. Ramírez-Fuentes, and V. Pacheco-Valencia, “An overview of some single machine scheduling problems: Polynomial algorithms, complexity and approximability,” *arXiv*, 2024. doi: 10.48550/arXiv.2405.18789

[7] K. R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*. Hoboken, NJ, USA: Wiley, 2018.

[8] J. Du and J. Y. T. Leung, “Minimizing total tardiness on one machine is NP-hard,” *Math. Oper. Res.*, vol. 15, no. 3, pp. 483–495, 1990. doi: 10.1287/moor.15.3.483
[9] C. N. Potts and L. N. Van Wassenhove, “A branch and bound algorithm for the total weighted tardiness problem,” *Oper. Res.*, vol. 33, no. 2, pp. 363–377, 1985. doi: 10.1287/opre.33.2.363
[10] E. L. Lawler, “A 'pseudopolynomial' algorithm for sequencing jobs to minimize total tardiness,” *Ann. Discrete Math.*, pp. 331–342, 1977. doi: 10.1016/S0167-5060(08)70742-8
[11] K. R. Baker and J. Bertrand, “A dynamic priority rule for scheduling against due-dates,” *J. Oper. Manage.*, vol. 3, no. 1, pp. 37–42, 1982. doi: 10.1016/0272-6963(82)90020-1
[12] U. Bilge, M. Kurtulan, and F. Kirac, “A tabu search algorithm for the single machine total weighted tardiness problem,” *Eur. J. Oper. Res.*, vol. 176, no. 3, pp. 1423–1435, 2007. doi: 10.1016/j.ejor.2005.10.030
[13] P. Yan, M. Jiao, and X. Yao, “Solving a single machine scheduling problem with uncertain demand using QPSO algorithms,” in *Proc. 25th Chinese Control Decision Conf. (CCDC)*, 2013, pp. 2741–2745. doi: 10.1109/CCDC.2013.6561408
[14] W. Bożejko, J. Pempera, M. Uchroński, and M. Wodecki, “Distributed quantum annealing on D-Wave for the single machine total weighted tardiness scheduling problem,” in *Proc. Int. Conf. Comput. Sci. (ICCS)*, 2022, pp. 171–178. doi: 10.1007/978-3-031-08760-8_15
[15] W. Bożejko, J. Pempera, M. Uchroński, and M. Wodecki, “An exact quantum annealing-driven branch and bound algorithm for maximizing the total weighted number of on-time jobs on a single machine,” in *Advanced, Contemporary Control*, M. Pawelczyk, D. Bismor, S. Ogonowski, and J. Kacprzyk, Eds. Cham, Switzerland: Springer, 2023, pp. 73–83. doi: 10.1007/978-3-031-35173-0_8
[16] W. Bożejko, R. Klempous, J. Pempera, J. Rozenblit, C. Smutnicki, M. Uchroński, and M. Wodecki, “Optimal solving of a scheduling problem using quantum annealing metaheuristics on the D-Wave quantum solver,” *IEEE Trans. Syst., Man, Cybern., Syst.*, 2024. doi: 10.1109/TSMC.2024.3458873
[17] W. Bożejko, S. Trotskyi, M. Uchroński, and M. Wodecki, “Optimizing two-machine scheduling in flexible manufacturing systems using autonomous AI and quantum computing,” *Neurocomputing*, vol. 629, art. no. 132067, 2025. doi: 10.1016/j.neucom.2025.132067
[18] Y. S. Wang and J. Cheng, “A quantum optimization algorithm for single machine total weighted tardiness minimization,” *arXiv*, 2022. doi: 10.48550/arXiv.2209.13712
[19] C. Grange, M. Poss, E. Bourreau, V. T'kindt, and O. Ploton, “Moderate exponential-time quantum dynamic programming across the subsets for scheduling problems,” *Eur. J. Oper. Res.*, vol. 320, no. 3, pp. 516–526, 2025. doi: 10.1016/j.ejor.2024.09.005
[20] C. Grange, M. Poss, E. Bourreau, V. T'kindt, and O. Ploton, “Companion paper: Moderate exponential-time quantum dynamic programming across the subsets for scheduling problems,” *Hal Open Science*, 2024.
[21] D. Amaro, M. Rosenkranz, N. Fitzpatrick, K. Hirano, and M. Fiorentini, “A case study of variational quantum algorithms for a job shop scheduling problem,” *EPJ Quantum Technol.*, vol. 9, no. 1, pp. 5, 2022. doi: 10.1140/epjqt/s40507-022-00123-4
[22] K. Kurowski, T. Pecyna, M. Slysz, R. Różycki, G. Waligóra, and J. Weglarz, “Application of quantum approximate optimization algorithm to job shop scheduling problem,” *Eur. J. Oper. Res.*, vol. 310, no. 2, pp. 518–528, 2023. doi: 10.1016/j.ejor.2023.03.013
[23] L. Palackal, L. Richter, and M. Hess, “Graph-controlled permutation mixers in QAOA for the flexible job-shop problem,” *arXiv*, 2023. doi: 10.48550/arXiv.2311.04100
[24] E. Bourreau, G. Fleury, and P. Lacomme, “Indirect job-shop coding using rank: Application to QAOA (IQAOA),” *arXiv*, 2024. doi: 10.48550/arXiv.2402.18280
[25] D. Venturelli, D. J. Marchand, and G. Rojo, “Quantum annealing implementation of job-shop scheduling,” *arXiv*, 2015. doi: 10.48550/arXiv.1506.08479
[26] K. Kurowski, J. Weglarz, M. Subocz, R. Różycki, and G. Waligóra, “Hybrid quantum annealing heuristic method for solving job shop scheduling problem,” in *Proc. Int. Conf. Comput. Sci. (ICCS)*, 2020, pp. 502–515. doi: 10.1007/978-3-030-50433-5_39
[27] M. R. Zielewski, M. Agung, R. Egawa, and H. Takizawa, “Improving quantum annealing performance on embedded problems,” *Supercomput. Front. Innov.*, vol. 7, no. 4, pp. 32–48, 2020. doi: 10.14529/jsfi200403
[28] C. Carugno, M. Ferrari Dacrema, and P. Cremonesi, “Evaluating the job shop scheduling problem on a D-Wave quantum annealer,” *Sci. Rep.*, vol. 12, no. 1, art. no. 6539, 2022. doi: 10.1038/s41598-022-10169-0
[29] M. Geitz, C. Grozea, W. Steigerwald, R. Stöhr, and A. Wolf, “Solving the extended job shop scheduling problem with AGVs—classical and quantum approaches," in *Proc. Int. Conf. Integr. Constraint Program., Artif. Intell., Oper. Res. (CPAIOR)*, 2022, pp. 120–137. doi: 10.1007/978-3-031-08011-1_10
[30] P. U. Rao and B. Sodhi, “Scheduling with multiple dispatch rules: A quantum computing approach,” in *Proc. Int. Conf. Comput. Sci. (ICCS)*, 2022, pp. 233–246.
[31] R. Aggoune and S. Deleplanque, “A quantum annealing solution to the job shop scheduling problem,” in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, 2023, pp. 421–428.

[32] J. R. Jiang and C. W. Chu, “Classifying and benchmarking quantum annealing algorithms based on quadratic unconstrained binary optimization for solving NP-hard problems,” *IEEE Access*, vol. 11, pp. 104165–104178, 2023.
[33] T. Chinnusamy, V. Raja, T. Gopalakrishnan, and R. Manikandan, “A quantum annealing-based approach for multi-objective optimization in flexible job shop scheduling using quantum-adaptive flex scheduler,” *Res. Square*, 2024 (preprint).
[34] S. Inoue, T. Qin, S. Fujita, S. Tajima, and H. Yoshioka, “Practical approach to flexible job shop scheduling with tool switching constraints using quantum annealing,” *J. Adv. Mech. Des., Syst., Manuf.*, vol. 18, no. 2, art. no. JAMDSM0009, 2024.
[35] P. Schworm, X. Wu, M. Klar, and J. C. Aurich, “Evaluation of quantum annealing-based algorithms for flexible job shop scheduling,” *arXiv*, 2024. doi: 10.48550/arXiv.2408.15671
[36] L. Toma, M. Zajac, and U. Störl, “Solving distributed flexible job shop scheduling problems in the wool textile industry with quantum annealing,” *arXiv*, 2024. doi: 10.48550/arXiv.2403.06699
[37] K. E. Booth, B. O’Gorman, J. Marshall, S. Hadfield, and E. Rieffel, “Quantum-accelerated constraint programming,” *Quantum*, vol. 5, pp. 550, 2021. doi: 10.22331/q-2021-09-28-550
[38] S. Jinfeng, F. Bin, and S. Jun, “Solving job-shop scheduling problem with QPSO algorithm with mutation operator,” *Comput. Eng. Appl.*, vol. 44, no. 8, pp. 49–52, 2008.
[39] J. Shi, B. Feng, and J. Sun, “Quantum-behaved particle swarm optimization for solving job-shop scheduling problem,” *Appl. Res. Comput.*, vol. 25, no. 3, pp. 684–686, 2008.
[40] Z. Fu, X. Liu, H. Chen, and M. Zhou, “Hybrid QPSO algorithm for job shop scheduling problems,” *J. Hefei Univ. Technol.*, vol. 32, no. 3, pp. 369–373, 2009.
[41] B. Jiao and S. Yan, “A cooperative co-evolutionary quantum particle swarm optimizer based on simulated annealing for job shop scheduling problem,” *Int. J. Artif. Intell.*, vol. 7, pp. 232–247, 2011.
[42] M. R. Singh, S. Mahapatra, and R. Mishra, “Robust scheduling for flexible job shop problems with random machine breakdowns using a quantum behaved particle swarm optimisation,” *Int. J. Serv. Oper. Manage.*, vol. 20, no. 1, pp. 1–20, 2015. doi: 10.1504/IJSOM.2015.065969
[43] M. R. Singh and S. S. Mahapatra, “A quantum behaved particle swarm optimization for flexible job shop scheduling,” *Comput. Ind. Eng.*, vol. 93, pp. 36–44, 2016.
[44] J. Xuan and H. Ming, “An improved quantum particle swarm optimization and its application,” in *Proc. 6th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, 2017, pp. 28–31. doi: 10.1109/ICCSNT.2017.8343471
[45] X. Liang, L. Li, and M. Huang, “Application of hybrid quantum behaved particle swarm optimization in multi-objective flexible job shop scheduling,” in *Proc. 3rd IEEE Int. Conf. Cloud Comput. Internet Things (CCIOT)*, 2018, pp. 78–82.
[46] S. Yu, “Differential evolution quantum particle swarm optimization for solving job-shop scheduling problem,” in *Proc. Chinese Control Decision Conf. (CCDC)*, 2019, pp. 559–564. doi: 10.1109/CCDC.2019.8833361
[47] Q. Zhang and S. Hu, “An improved hybrid quantum particle swarm optimization algorithm for FJSP,” in *Proc. 11th Int. Conf. Mach. Learn. Comput. (ICMLC)*, 2019, pp. 246–252. doi: 10.1145/3318299.3318359
[48] B. Denkena, F. Schinkel, J. Pirnay, and S. Wilmsmeier, “Quantum algorithms for process parallel flexible job shop scheduling,” *CIRP J. Manuf. Sci. Technol.*, vol. 33, pp. 100–114, 2021. doi: 10.1016/j.cirpj.2021.03.006
[49] L. Guan, Y. Ma, B. Jiang, Y. Huang, J. Zhou, and H. Zhang, “Multi-objective flexible job shop scheduling based on Pareto quantum particle swarm optimization algorithm,” in *Proc. 36th Chinese Control Decision Conf. (CCDC)*, 2024, pp. 2626–2631. doi: 10.1109/CCDC62350.2024.10588114
[50] Y. Xu, D. Wang, M. Zhang, M. Yang, and C. Liang, “Quantum particle swarm optimization with chaotic encoding schemes for flexible job-shop scheduling problem,” *Swarm Evol. Comput.*, vol. 93, art. no. 101836, 2025. doi: 10.1016/j.swevo.2024.101836
[51] Y. Xu, M. Zhang, M. Yang, and D. Wang, “Hybrid quantum particle swarm optimization and variable neighborhood search for flexible job-shop scheduling problem,” *J. Manuf. Syst.*, vol. 73, pp. 334–348, 2024. doi: 10.1016/j.jmsy.2024.02.007
[52] M. Fiasché, D. E. Liberati, S. Gualandi, and M. Taisch, “Quantum-inspired evolutionary multiobjective optimization for a dynamic production scheduling approach,” in *Multidisciplinary Approaches to Neural Computing*, Cham, Switzerland: Springer, 2018, pp. 191–201.
[53] H. Zhu, N. Luo, and X. Li, “A quantum-inspired cuckoo co-evolutionary algorithm for no-wait flow shop scheduling,” *IET Collab. Intell. Manuf.*, vol. 3, no. 2, pp. 105–118, 2021. doi: 10.1049/cim2.12002
[54] J. Zhu, Z. Shao, and C. Chen, “An improved whale optimization algorithm for job-shop scheduling based on quantum computing,” *Int. J. Simul. Model.*, vol. 18, no. 3, pp. 521–530, 2019. doi: 10.2507/IJSIMM18(3)CO13

[55] Y. Luo, Q. Zhang, and L. Lin, “A cooperative hybrid evolutionary algorithm for flexible scheduling with AGVs,” in *Proc. Int. Conf. Sens., Meas. Data Anal. Era Artif. Intell. (ICSMD)*, 2023, pp. 1–7. doi: 10.1109/ICSMD60522.2023.10490770
[56] X. Wu and S. Wu, “An elitist quantum-inspired evolutionary algorithm for the flexible job-shop scheduling problem,” *J. Intell. Manuf.*, vol. 28, pp. 1441–1457, 2017. doi: 10.1007/s10845-015-1060-6
[57] J. Gu, X. Gu, and M. Gu, “A novel parallel quantum genetic algorithm for stochastic job shop scheduling,” *J. Math. Anal. Appl.*, vol. 355, no. 1, pp. 63–81, 2009.
[58] D. W. Li, “To solve the job shop scheduling problem with the improved quantum genetic algorithm,” in *Proc. 3rd Global Congr. Intell. Syst. (GCIS)*, 2012. doi: 10.1109/GCIS.2012.98
[59] T. Ning, H. Jin, and X. Song, “An improved quantum genetic algorithm based on MAGTD for dynamic FJSP,” *J. Ambient Intell. Humanized Comput.*, vol. 9, pp. 931–940, 2018. doi: 10.1007/s12652-017-0486-4
[60] S. Zhang, H. Du, S. Borucki, S. Jin, T. Hou, and Z. Li, “Dual resource constrained flexible job shop scheduling based on improved quantum genetic algorithm,” *Machines*, vol. 9, no. 6, art. no. 108, 2021.
[61] K. S. N. Ripon and A. Singh, “Quantum representation based job shop scheduling,” in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2023, pp. 1227–1233.
[62] M. Schmid, S. Braun, R. Sollacher, and M. J. Hartmann, “Highly efficient encoding for job-shop scheduling problems and its application on quantum computers,” *Quantum Sci. Technol.*, vol. 10, no. 1, pp. 15–51, 2024.