IMPACT OF HYPERPARAMETERS ON QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA) WHILE SOLVING THE SINGLE MACHINE SCHEDULING PROBLEM WITH WEIGHTED TARDINESS AS THE PERFORMANCE

AĞIRLIKLI GECIKMENIN PERFORMANS ÖLÇÜTÜ OLARAK KULLANILDIĞI TEK MAKİNELİ ÇİZELGELEME PROBLEMİNDE HİPERPARAMETRELERİN KUANTUM YAKLAŞIK OPTİMİZASYON ALGORITMASININ (QAOA) PERFORMANSINA ETKISI

Burcu Çağlar Gençosman[1]
Seda Özmutlu[2]

1 Doç.Dr. ,Bursa Uludağ Üniversitesi, Endüstri Mühendisliği Bölümü, 0000-0003-0159-8529
2 Prof.Dr. Bursa Uludağ Üniversitesi Endüstri Mühendisliği Bölümü. 0000-0002-2744-2744

Abstract

Quantum Computing has been a facilitator in solving intractable problems since the 2000s. In search of a faster alternative, researchers began applying quantum computing to solve NP-hard scheduling problems. The Single Machine Weighted Tardiness Problem (SMWTP) is an important NP-hard problem in the scheduling literature and is well-suited to quantum computing solutions. The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm that can be used to solve SMWTP; however, hyperparameter tuning may be important in its application. This study aims to identify the impact of hyperparameters on QAOA's performance to solve the SMWTP. The SMWTP is first modeled as a mathematical program, then converted to a Quadratic Binary Optimization (QUBO) format to be solved by quantum methods. Then, the QUBO format is expressed as Ising-Hamiltonians, i.e., the cost and mixer Hamiltonians. After obtaining the Ising Hamiltonian, the QAOA circuit is executed by alternately applying the cost and mixing Hamiltonians, followed by measurement and subsequent classical evaluation of the weighted tardiness values. Four algorithm parameters for QAOA might affect solution performance: the Cost Hamiltonian parameter, the Mixer Hamiltonian parameter, the number of layers, and the penalty multiplier. To measure the impact of hyperparameters on QAOA performance, test instances in which algorithm parameters are systematically varied for comprehensive experimentation are run. Three metrics are used to measure hyperparameter performance: the distribution of the best results across hyperparameter levels, the average optimality gap, and the percentage of feasible schedules. The hyperparameter analysis reveals that the mixer parameter (β) is the primary determinant of the recovery of feasible solutions. Results indicate that using β=0.1 decreases the optimality gap and increases the percentage of feasible solutions.

Keywords: Quantum computing, QAOA, Single machine scheduling problem, hyperparameter analysis.

Özet

Kuantum hesaplama, 2000'li yıllardan beri zor problemlerin çözümünde kolaylaştırıcı bir rol oynamıştır. Daha hızlı bir alternatif arayışında olan araştırmacılar, NP-zor zamanlama problemlerini çözmek için kuantum hesaplamayı uygulamaya başlamışlardır. Tek Makineli Ağırlıklı Gecikme

Problemi (SMWTP), çizelgeleme literatüründe önemli bir NP-zor problemdir ve kuantum hesaplama çözümlerine oldukça uygundur. Kuantum Yaklaşık Optimizasyon Algoritması (QAOA), SMWTP'yi çözmek için kullanılabilen varyasyonel bir kuantum algoritmasıdır; ancak, hiperparametre seçimleri algoritmanın performansında önemli bir rol oynayabilir. Bu çalışma, hiperparametrelerin QAOA'nın SMWTP'yi çözme performansına etkisini belirlemeyi amaçlamaktadır. SMWTP önce matematiksel programlama ile modellenmiştir, ardından kuantum yöntemlerle çözülmek üzere Karesel İkili Optimizasyon (QUBO) formatına dönüştürülmüştür. Daha sonra, QUBO formatı Ising-Hamiltonyenleri, yani maliyet ve karıştırıcı Hamiltonyenleri olarak ifade edilir. Ising Hamiltonyeni elde edildikten sonra, maliyet ve karıştırma Hamiltonyenlerinin dönüşümlü olarak uygulanmasıyla QAOA devresi yürütülür, ardından ağırlıklı gecikme değerleri ölçülür ve daha sonra klasik olarak değerlendirilir. QAOA için dört algoritma parametresi çözüm performansını etkileyebilir: Maliyet Hamiltonyeni parametresi, Karıştırıcı Hamiltonyen parametresi, katman sayısı ve ceza çarpanı. Hiperparametrelerin QAOA performansı üzerindeki etkisini ölçmek için, algoritma parametrelerinin sistematik olarak değiştirildiği kapsamlı deneyler gerçekleştirilmiştir. Hiperparametre performansını ölçmek için üç ölçüt kullanılmıştır: hiperparametrelerin seviyeleri boyunca en iyi sonuçların dağılımı, ortalama optimalite boşluğu (optimality gap) ve uygun çözümlerin yüzdesi. Hiperparametre analizi, karıştırıcı (β) parametresinin uygulanabilir çözümlerin elde edilmesinde birincil belirleyici olduğunu ortaya koymaktadır. Sonuçlar, β=1 kullanmanın optimale daha yakın sonuçlar ürettiğini ve uygun çözümlerin yüzdesini artırdığını göstermektedir.

Anahtar kelimeler: Kuantum hesaplama, QAOA, Tek makineli çizelgeleme problemi, hiperparametre analizi.

6. INTRODUCTION

Conventional techniques are no longer sufficient to solve many practically relevant scheduling problems optimally, despite the continually increasing computing power. In search of an alternative, faster solution, researchers began applying quantum computing to solve scheduling problems. Although quantum algorithms are primarily applied to the more familiar and popular JSP in the quantum computing literature, there are also other well-known scheduling problems in the scheduling literature. Single-machine scheduling problems (SMSPs) are a special case of single-stage job scheduling problems, in which only one machine is available. There are NP-hard variants of the SMSP that have been attempted numerous times with Classical Algorithms but rarely studied with Quantum Algorithms. One NP-hard variant of the SMSP is the SMSP with total weighted tardiness as the performance measure. In general, we see little literature on the application of Quantum algorithms to solve the SMSP with tardiness-related measures, which brings us to the research motivation of this paper; i.e., studying the SMSP problem with total weighted tardiness as the performance measure, namely, the Single Machine Weighted Tardiness Problem (SMWTP), with quantum algorithms.

In this paper, the SMWTP is studied using the Quantum Approximate Optimization Algorithm (QAOA) as the quantum algorithm. Hyperparameter optimization is an important issue in application of quantum algorithms. Therefore, the aim of the paper is to analyze the effect of hyperparameters on the performance of QAOA, whilst solving the SMWTP. A through grid study is performed to find the optimal or near-optimal values for the hyperparameters of QAOA.

The rest of the paper is organized as follows. In Section 2, the literature on quantum methods for single-machine scheduling is reviewed. Section 3 presents a brief review of QAOA and a detailed description of the solution methodology and hyperparameter search considered in the paper. Section 4 discusses the results, and Section 5 concludes the paper.

7. LITERATURE REVIEW

In recent years, quantum algorithms have shown promising results for tardiness-based scheduling problems. An extensive review of quantum computing applications on scheduling problems can be seen in (Werner ve Ullinger, 2025). There are several papers in quantum computing literature studying the SMSP. Bożejko, et al. (2022) solve a small-sized SMWTP using quantum annealing. They could optimally solve a problem with n=8 jobs. In a subsequent study, Bożejko, et al. (2023) approach the SMWTP by determining a lower bound on the objective function using quantum annealing. They continue this approach in further studies Bożejko, et al. (2024) and Bożejko, et al. (2025) they use quantum annealing in the branch and bound algorithm to compute the lower and upper bounds of the objective function for the SMWTP.

Wang, et al. (2022) also minimize total weighted tardiness in the SMSP using Grover's quantum search and Trugenberger's quantum optimization algorithms. Grange, et al. (2025) 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 (Grange, et al., 2024).

8. MATERIALS AND METHODS

3.1. The single machine scheduling problem to minimize total weighted tardiness (SMWTP)

The SMWPT considered in this study is defined as follows. There exists a single machine m, and a set of jobs $J = \{1,2,\dots,n\}$ to be processed on machine m. This problem is denoted by $1 \,||\, \sum w_j T_j$. Each job $j \in J = \{1,2,\dots,n\}$ is characterized by a processing time p_j, a due date d_j, and a tardiness weight w_j. There also exists a release time for the jobs r_j. For a permutation schedule $\pi = (\pi(1), \pi(2), \dots, \pi(n))$, where $\pi(k)$ denotes the position of job j at position k, the completion time of job j at position k is denoted as $C_{\pi(k)} = \sum_{j=1}^{k} p_{\pi(j)}, k = 1,2,\dots,n$. For notational convenience, C_j denotes the completion time of job j in the schedule π. Total tardiness for a schedule is also an important performance measure and can be stated as $T = \sum_{j=1}^{n} T_j$, where $T_j = \max\{0, C_j - d_j\}$.

In the SMWPT problem, the objective is to find an optimal schedule π^* that minimizes the total weighted tardiness: $\min_{\pi} \sum_{j=1}^{n} w_j T_j$. We also consider the assumptions that describe the most basic form of the SMWPT: All jobs are available at time zero; each task must be performed on machine m; the machine can perform at most one task at any given time; no preemption is allowed; no idle time is allowed; there are no technological precedence constraints; setup times are considered zero.

The mathematical formulation of SMWTP is a mixed integer program and the decision variables can be defined as:

$$x_{jk} = \begin{cases} 1, & \text{if job } j \text{ is assigned to position } k \text{ in the sequence,} \\ 0, & \text{otherwise.} \end{cases}$$

The mixed-integer programming model is:

$$\min T = \sum_{k=1}^{n} w_k T_k \quad (1)$$

$$\sum_{j=1}^{n} x_{jk} = 1, \quad \forall k = 1,\dots,n \quad (2)$$

$$\sum_{k=1}^{n} x_{jk} = 1, \quad \forall j = 1,\dots,n \quad (3)$$

$$\sum_{j=1}^{n} p_j \sum_{u=1}^{k} x_{ju} - \sum_{j=1}^{n} d_j x_{jk} \leq T_k, \quad \forall k = 1, \dots, n \qquad (4)$$

$$T_k \geq 0, \quad \forall k = 1, \dots, n \qquad (5)$$

$$x_{jk} = \{0,1\}, \quad \forall j, k = 1, \dots, n \qquad (6)$$

The objective function (1) minimizes total weighted tardiness. Constraints (2) and (3) ensure that the sequence–position variables must satisfy the assignment constraints: Constraint (2) denotes that each position is occupied by one job only. Constraint (3) states that each job is assigned to one position, and thus processed once. Constraint (4) is the definitional constraint that captures tardiness, where T_k denotes the tardiness of the job scheduled in position k. Constraint (5) is the non-negativity constraint for the tardiness variable T_k, which will take the value zero if the job finishes before its due date. Constraint (6) defines binary decision variables.

3.2. Quantum Approximate Optimization Algorithm (QAOA)

In this paper, we use the Quantum Approximate Optimization Algorithm (QAOA) to solve the SMWTP. QAOA is a hybrid quantum–classical variational algorithm designed to solve discrete combinatorial optimization problems (Farhi, et al., 2014).

The gap between the mathematical models of combinatorial optimization problems and quantum algorithms is bridged by Quadratic Unconstrained Binary Optimization (QUBO). The classical objective takes the form $\min_{x\in\{0,1\}^n} x^T Q x$, where $Q \in R^{n\times n}$ denotes the symmetric QUBO matrix and x represents the binary decision vector. Each binary variable is associated with a single qubit through the standard mapping to spin variables $s_i \in \{-1,1\}$. $x_i = \frac{1-s_i}{2}$. After the binary-to-spin variable transformation, the quadratic terms of the QUBO formulation naturally translate into pairwise $\hat{Z}_i\hat{Z}_j$ interactions in the quantum model. Promoting the classical spins to Pauli operators, $s_i \to \hat{Z}_i$, leads to the cost Hamiltonian $\widehat{H}_C = \sum_i h_i \hat{Z}_i + \sum_{i<j} J_{ij} \hat{Z}_i\hat{Z}_j$, where the coefficients h_i and J_{ij} are derived from the entries of Q as:

$$J_{ij} = \frac{1}{4} Q_{ij}$$

$$h_i = \frac{1}{2}\left(Q_{ii} - \sum_j Q_{ij} \right)$$

QAOA employs a mixer Hamiltonian $\widehat{H}_M = \sum_i \hat{X}_i$, where $\hat{X}_i$ denotes the Pauli-X operator. The algorithm applies an alternating sequence of unitary operators derived from the cost and mixer Hamiltonians for a fixed number of layers p. The variational parameters $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$, along with the circuit depth p, constitute the hyperparameters of QAOA. A standard QAOA circuit with p layers is defined as:

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \prod_{l=1}^{p} \widehat{U}_M(\beta_l)\widehat{U}_C(\gamma_l)|+\rangle^{\otimes n} = \widehat{U}_M(\beta_p)\widehat{U}_C(\gamma_p) \cdots \widehat{U}_M(\beta_1)\widehat{U}_C(\gamma_1)|+\rangle^{\otimes n}$$

where $|+\rangle^{\otimes n}$ represents the equal superposition state, $\widehat{U}_C(\gamma_l)$ is the cost unitary operator, and $\widehat{U}_M(\beta_l)$ is the mixer unitary operator. QAOA seeks to approximately minimize the expectation value of the cost Hamiltonian $\widehat{H}_C$ by optimizing the variational parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ in a hybrid quantum–classical loop. After preparing the variational state for a given set of parameters, the quantum circuit is executed and the state is measured in the computational basis. Due to the probabilistic nature of quantum mechanics, this process is repeated multiple times (typically $N_{\text{shots}} = 1000$ to 10000 shots).

The measured bitstrings are then decoded into job permutations (schedules) and evaluated classically to compute their weighted tardiness costs. The empirical expectation value is estimated as $\tilde{E}_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$. This solution may then be subjected to classical post-processing like verifying constraint satisfaction or performing local search refinements.

3.3. Computational Experiments

In this section, we present the results of applying QAOA to solve the SMWTP. Problem instances with $n = 5$ and $n = 6$ jobs are solved to demonstrate the applicability of QAOA to the SMWTP. We use the test problems presented in Bożejko, et al. (2024). Fifty problems each with $n = 5$ and $n = 6$ are solved. Warm-start strategy is used in the trials; i.e. the initial quantum state is encoded from a solution obtained using the WSPT heuristic. The problems are then subjected to hyperparameter optimization through an experimental design to explore the QAOA parameter space thoroughly. A comprehensive grid search over the following hyperparameters with the corresponding levels is performed:

- Cost Hamiltonian parameter: $\gamma \in \{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9\}$ (9 values)
- Mixer Hamiltonian parameter: $\beta \in \{0.1,0.2,0.3,0.4,0.5\}$ (5 values)
- Number of layers: $p \in \{1,2\}$ (2 values)
- Penalty multiplier: $M \in \{1,2,5\}$ (3 values)

This results in $9 \times 5 \times 2 \times 3 = 270$ unique hyperparameter configurations per problem instance and for 100 problems (50 for n=5 and 50 for n=6) it results 270,000 trials in total. All configurations are evaluated explicitly in a grid search manner to ensure reproducibility and transparency, and to be able to investigate the effect of the hyperparameters. The number of measurement shots is fixed to 8192 to attain a result in a reasonable amount of time. The solution with the lowest weighted tardiness is selected for each test instance. The results are presented in Table 1 and Table 2.

To evaluate the performance of QAOA, the results are compared to those from an exact solver. To benchmark against the exact solver, the problem instances are mathematically modeled as in Section 3.1 and solved with the Gurobi Optimizer. Gurobi Optimizer Version 11.0.0 is used as the exact solver to obtain optimal or near-optimal solutions. The resulting objective value T_{opt} is used as a benchmark for computing the optimality gap against QAOA. The optimality gap is defined as: $\text{Gap} = \frac{T_{\text{QAOA}} - T_{\text{opt}}}{T_{\text{opt}}} \times 100\%$, where T_{QAOA} is the solution cost obtained by QAOA, and T_{opt} is the optimal cost from Gurobi.

All experiments were conducted using Python 3.9.13, Qiskit 0.45.0, Qiskit Aer 0.13.0, Gurobipy 11.0.0, NumPy 1.24.0, and Pandas 2.0.0. Computations were performed on a Windows 11 Pro (64-bit) system equipped with an Intel Core i9-13900H (2.60 GHz) and 64 GB of RAM. It is important to note that due to the limitations of currently available quantum computing platforms, the proposed quantum optimization algorithm has not been implemented on actual quantum hardware but rather performed as a simulation.

9. RESULTS AND DISCUSSION

The computational results for all instances (50 instances for $n = 5$ and 50 instances for $n = 6$) are presented in Table 1 and Table 2, respectively. Tables include the best hyperparameter configuration found for each instance (optimal γ, β, p, and M), along with the weighted tardiness values (T_{QAOA}, T_{opt}).

Table 1 presents the computational results for all 50 instances with $n = 5$ jobs. QAOA achieves the optimal solution (gap = 0%) in 17 out of 50 instances, with an average optimality gap of approximately 11.14% across all instances. It is observed that QAOA can solve the SMWTP problem,

and can solve optimally in 34% of the cases. Table 2 presents the results for $n = 6$ instances, where the problem size increases to 36 qubits (i.e., $n^2 = 36$ binary variables). QAOA again achieves the optimal solution in 17 out of 50 instances with an average optimality gap of approximately 21.92% across all instances.

Table 1: Computational results of QAOA and Gurobi for SMWTP instanced with n=5 jobs

Problem	Best_Gamma	Best_Beta	Best_p	Best_M	T_QAOA	QAOA_Gap	T_opt
wt5_001	0,1	0,2	2	1	44	0,00	44
wt5_002	0,1	0,5	1	1	186	0,00	186
wt5_003	0,1	0,2	2	2	58	0,00	58
wt5_004	0,1	0,4	1	2	126	0,00	126
wt5_005	0,1	0,1	1	1	100	0,00	100
wt5_006	0,1	0,3	1	5	785	6,66	736
wt5_007	0,1	0,4	1	2	432	34,58	321
wt5_008	0,1	0,3	1	1	396	25,32	316
wt5_009	0,9	0,5	2	2	193	6,04	182
wt5_010	0,5	0,5	1	5	665	0,00	665
wt5_011	0,1	0,3	2	1	1248	0,00	1248
wt5_012	0,3	0,3	2	5	1472	21,25	1214
wt5_013	0,1	0,2	2	5	1024	27,36	804
wt5_014	0,8	0,5	1	5	2576	27,27	2024
wt5_015	0,1	0,2	2	2	1409	6,42	1324
wt5_016	0,7	0,3	2	2	1400	14,19	1226
wt5_017	0,1	0,4	1	2	2017	8,32	1862
wt5_018	0,5	0,4	2	2	2556	0,20	2551
wt5_019	0,7	0,5	2	2	515	11,23	463
wt5_020	0,2	0,4	2	1	1039	13,06	919
wt5_021	0,1	0,4	1	2	1390	0,00	1390
wt5_022	0,5	0,5	2	1	1702	7,79	1579
wt5_023	0,8	0,4	2	1	2240	6,62	2101
wt5_024	0,9	0,5	1	5	4250	16,92	3635
wt5_025	0,7	0,5	2	5	5187	4,77	4951
wt5_026	0,9	0,4	1	2	33	0,00	33
wt5_027	0,1	0,3	1	1	0	0,00	0
wt5_028	0,1	0,3	2	5	275	6,59	258
wt5_029	0,1	0,2	2	2	120	30,43	92
wt5_030	0,1	0,1	2	5	87	0,00	87
wt5_031	0,1	0,2	2	5	112	0,00	112
wt5_032	0,1	0,3	2	1	207	0,00	207
wt5_033	0,5	0,4	2	1	853	61,25	529
wt5_034	0,2	0,2	2	2	225	0,00	225
wt5_035	0,6	0,3	2	1	147	0,00	147
wt5_036	0,2	0,3	2	2	372	27,84	291
wt5_037	0,5	0,5	1	1	2160	9,37	1975
wt5_038	0,2	0,3	1	2	463	68,36	275
wt5_039	0,3	0,3	2	5	1239	19,71	1035
wt5_040	0,5	0,5	2	5	423	0,00	423
wt5_041	0,6	0,4	2	5	2125	0,09	2123

wt5_042	0,3	0,3	2	1	1872	13,80	1645
wt5_043	0,3	0,4	1	2	1093	9,19	1001
wt5_044	0,2	0,4	2	5	1588	0,00	1588
wt5_045	0,1	0,3	2	5	2409	20,63	1997
wt5_046	0,8	0,5	2	2	1950	8,88	1791
wt5_047	0,9	0,4	2	1	2475	1,31	2443
wt5_048	0,1	0,3	2	2	3831	14,26	3353
wt5_049	0,9	0,4	1	5	1719	8,94	1578
wt5_050	0,1	0,3	1	1	1230	6,22	1158

To measure the impact of hyperparameters on the performance of QAOA, three metrics are employed, i.e. the distribution of the best results with respect to hyperparameter levels, average optimality gap and the percentage of feasible schedules. Figures 1, 2, 3 and 4 present analyses for hyperparameters, γ, β, p and M respectively.

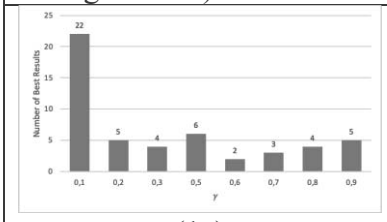
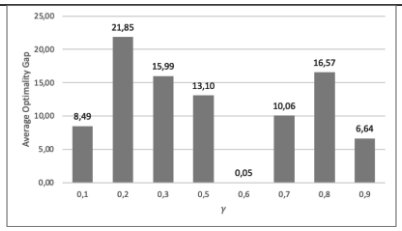
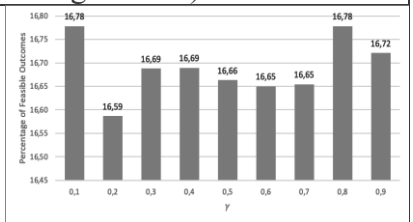
Table 2: Computational results of QAOA and Gurobi for SMWTP instanced with n=6 jobs

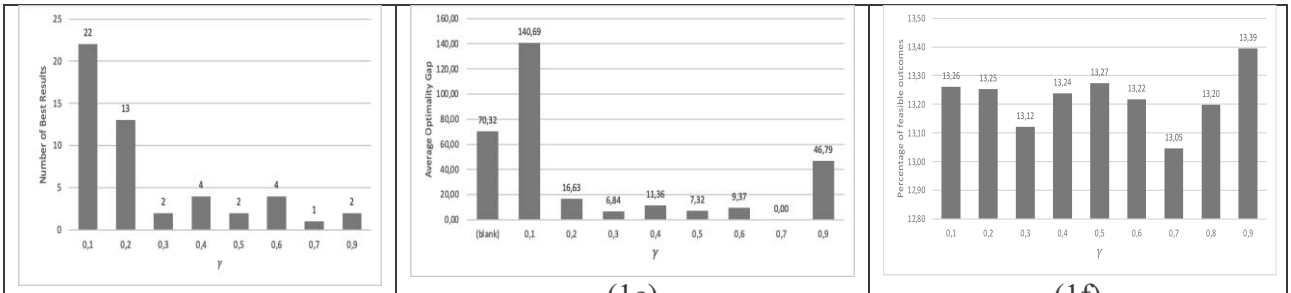
Problem	Best_Gamma	Best_Beta	Best_p	Best_M	T_QAOA	QAOA_Gap	T_opt
wt6_001	0,1	0,2	1	1	198	0,00	198
wt6_002	0,2	0,1	2	5	113	0,00	113
wt6_003	0,4	0,2	2	5	288	45,45	198
wt6_004	0,1	0,1	1	1	52	0,00	52
wt6_005	0,1	0,3	2	1	136	0,00	136
wt6_006	0,1	0,3	2	2	560	117,90	257
wt6_007	0,1	0,3	2	2	937	12,89	830
wt6_008	0,2	0,2	2	2	1173	5,77	1109
wt6_009	0,1	0,2	2	1	876	55,32	564
wt6_010	0,2	0,4	1	2	662	23,51	536
wt6_011	0,9	0,4	1	1	786	69,76	463
wt6_012	0,1	0,3	2	1	712	17,69	605
wt6_013	0,1	0,4	1	5	1156	3,03	1122
wt6_014	0,6	0,2	2	2	1355	0,00	1355
wt6_015	0,1	0,3	2	2	1909	0,00	1909
wt6_016	0,6	0,1	2	2	1305	12,40	1161
wt6_017	0,1	0,4	1	5	5213	59,08	3277
wt6_018	0,4	0,4	1	1	2670	0,00	2670
wt6_019	0,1	0,2	2	5	4151	10,11	3770
wt6_020	0,9	0,3	1	1	3661	23,81	2957
wt6_021	0,1	0,2	1	2	4058	0,00	4058
wt6_022	0,1	0,3	2	2	3881	24,51	3117
wt6_023	0,5	0,3	1	2	3469	8,64	3193
wt6_024	0,2	0,3	2	2	3632	0,00	3632
wt6_025	0,2	0,3	1	2	4058	18,24	3432
wt6_026	0,6	0,4	1	5	48	0,00	0
wt6_027	0,1	0,2	2	1	152	15,15	132
wt6_028	0,4	0,1	2	5	4	0,00	4
wt6_029	0,1	0,4	1	2	126	2420,00	5
wt6_030	0,2	0,3	1	5	9	0,00	9
wt6_031	0,7	0,4	1	5	148	0,00	148
wt6_032	0,1	0,3	1	1	384	52,38	252
wt6_033	0,2	0,2	2	5	237	78,20	133

wt6_034	0,1	0,4	1	5	1164	99,32	584
wt6_035	0,1	0,2	2	5	103	77,59	58
wt6_036	0,2	0,3	2	2	1446	0,21	1443
wt6_037	0,2	0,4	1	2	874	24,86	700
wt6_038	0,4	0,3	1	1	1555	0,00	1555
wt6_039	0,6	0,5	1	1	2310	25,07	1847
wt6_040	0,1	0,5	1	2	2366	8,43	2182
wt6_041	0,2	0,2	2	5	1333	14,42	1165
wt6_042	0,2	0,2	2	1	4749	4,97	4524
wt6_043	0,1	0,3	1	1	2999	4,90	2859
wt6_044	0,1	0,3	2	2	2811	3,04	2728
wt6_045	0,2	0,4	1	1	1632	45,97	1118
wt6_046	0,2	0,5	1	5	2928	0,00	2928
wt6_047	0,5	0,1	2	1	2507	6,00	2365
wt6_048	0,3	0,2	1	2	3673	13,68	3231
wt6_049	0,3	0,2	2	1	2589	0,00	2589
wt6_050	0,1	0,4	1	5	2698	113,79	1262

In all the figures, the analysis is performed for n=5 and n=6 separately to also see the effect of the number of jobs, and the behavior of the performance measures with increasing problem sizes. First, the distribution of the 50 best results with respect to hyperparameter levels are studied to observe whether there is any pattern. Secondly, the optimality gap between QAOA and Gurobi results by hyperparameter levels are studied. The results in Table 1 and 2 are used again for this measure. Thirdly, to further characterize QAOA behavior, the percentage of measurement outcomes that correspond to feasible permutations (Percentage of feasible outcomes-PF) is also tracked. A bitstring is considered feasible if it encodes a valid permutation, i.e., each job is assigned to exactly one position and each position contains exactly one job. The percentage of feasible outcomes is computed not from Tables 1 and 2, which are all feasible apparently; but from the set of all results; i.e. 270*100 = 27,000 runs as PF $= \frac{\text{number of shots yielding a feasible schedule}}{N_{\text{shots}}} \times 100\%$.

The hyperparameter γ is examined in Figure 1. Figures 1(a) and 1(d)shows the distribution of the γ values that yield the best results, across 50 instances for n=5 and n=6, respectively, with $\gamma = 0.1$ standing out as the best configuration in 22 out of 50 instances for both n=5 and n=6. In terms of optimality gap, it is observed that medium values tend to provide better results, for n=5 $\gamma = 0.6$ provides the best optimality gap (0,05%) in Figure 1(b) and for n=6 $\gamma = 0.7$ provides the best optimality gap (0%) in Figure 1(e), thus mid-range values for γ can be used. In Figures 1(c) and 1(f), changing γ values does not seem to make much difference for PF, for n=5 PF is around 16% and for n=6 around 13%, regardless of the γ value. So, $\gamma = 0.6$ or $\gamma = 0.7$ can be used for the SMWTP to minimize the optimality gap in the solutions.

Distribution of best 50 results by γ (best configurations)	Average Optimality Gap by γ (best configurations)	Percentage of feasible outcomes by γ (all configurations)
 (1a)	 (1b)	 (1c)
n=5		



(1d) (1e) (1f)

n=6

Figure 1: Effect of hyperparameter γ on various performance measures for QAOA

Figure 2 presents analyses for hyperparameter β. Figure 2(a) and Figure 2(d) show the distribution of β values that yielded the best results. It is observed that $\beta = 0.3$ is the β value that yields the highest number of best results for both n=5 and n=6 problems. For optimality gap, it is noteworthy that the best β performances shifts to lower values in Figures 2(b) and 2(e). At $\beta = 0.1$, the average optimality gap is 0 for n=5 and at its lowest with 3,68 % for n=6. Thus, for both problem sizes $\beta = 0.1$ yielded the best results with the lowest optimality gap values. Figure 2(c) and 2(f) presents the average PF values calculated over all configurations as a function of β. The graphs clearly show that β has an effect on the percentage of feasible solutions. As β increases, PF decreases for both n=5 and n=6. Highest PF values are observed for $\beta = 0.1$. These findings indicate that β controls the exploration/optimality balance in the QAOA circuit, lower β values offer more feasible solutions, and thus better optimality gaps for both problem sizes. $\beta = 0.1$ can be preferred in subsequent studies to produce more feasible solutions with better results.

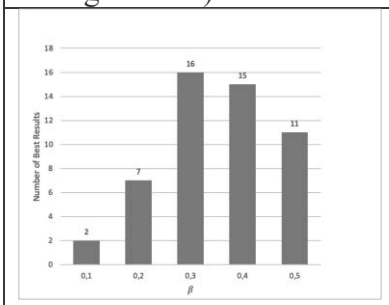
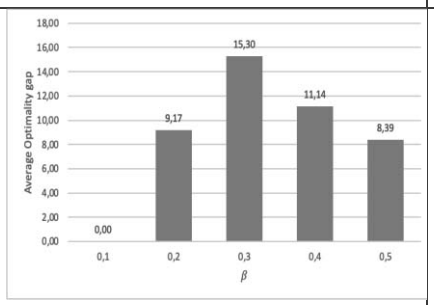
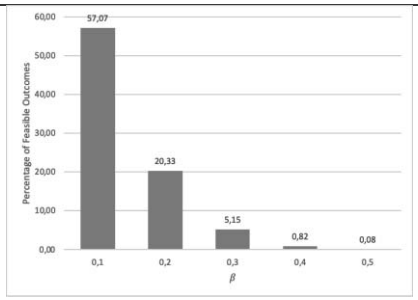
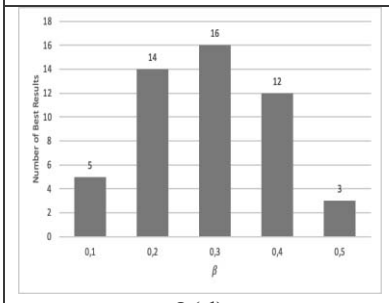
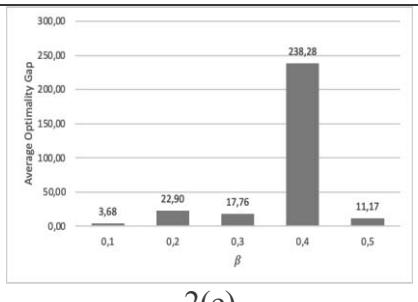
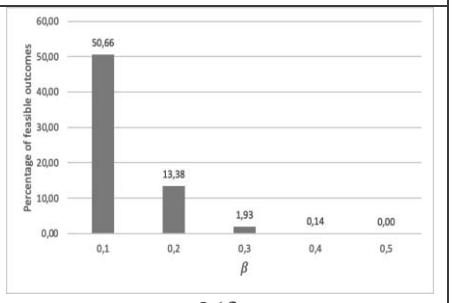
Distribution of best 50 results by β (best configurations)	Average Optimality Gap by β (best configurations)	Percentage of feasible outcomes by β (all configurations)
		
2(a)	2(b)	2(c)
n=5		
		
2(d)	2(e)	2(f)
n=6		

Figure 2: Effect of hyperparameter β on various performance measures for QAOA

The hyperparameter p is examined in Figure 3. Figures 3(a) and 3(d)show the distribution of the p values that yield the best results, however does not show any pattern. In terms of optimality gap, p=2 tends to provide better results. For n=5 $p = 2$ provides 20,9% optimality gap in Figure 3(b), and for n=6 $p = 2$ provides 10,12% optimality gap in Figure 3(e). However, in Figures 3 (c) and 3(f), $p = 1$ seems to provide a higher percentage of feasible schedules for both n=5 and n=6. These results make the choice of p ambiguous. Further tests can be done for firmer results, however efforts are truncated at this point for an exploratory initial study.

Figure 4 presents the results of the analysis for hyperparameter M. The results resemble those of p. Figures 4(a) and 4(d) does not show any pattern. In terms of optimality gap, M=1 tends to provide better results. For n=5 $M = 1$ provides 9,05% optimality gap in Figure 4(b), and for n=6 $M = 1$ provides 18,89% optimality gap in Figure 4(e). However, in Figure 4 (c) $M = 1$ seems to provide a higher percentage of feasible schedules for n=5, but in Figure 4(f) $M = 5$ seems to provide a higher percentage of feasible schedules for n=6. These results make the choice of M ambiguous, but $M = 1$ can be preferred to favor optimality gap.

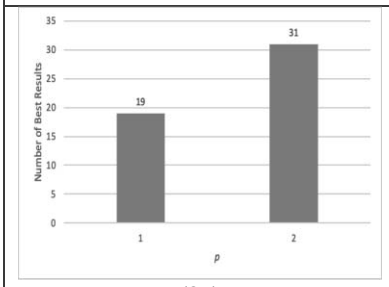
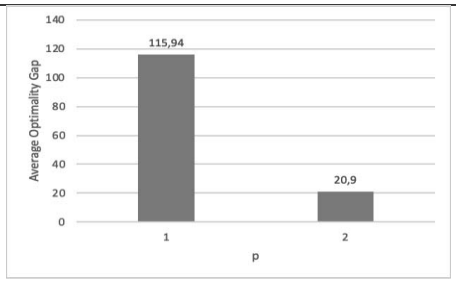
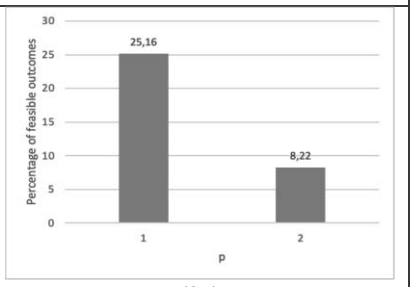
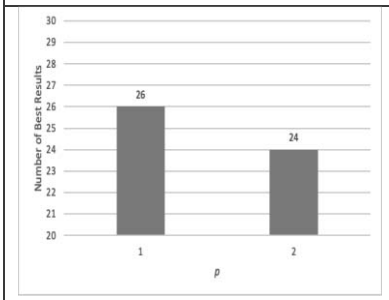
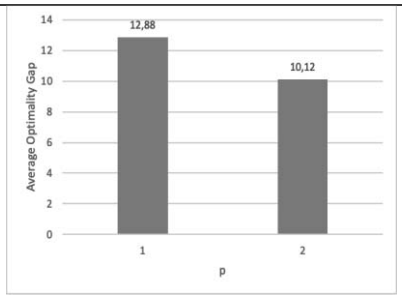
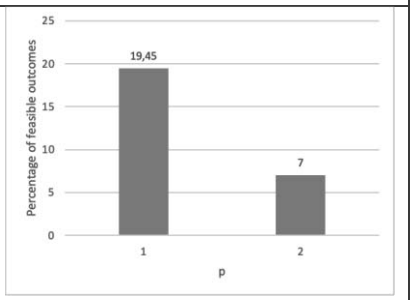
Distribution of best 50 results by p (best configurations)	Average Optimality Gap by p (best configurations)	Percentage of feasible outcomes by p (all configurations)
		
(3a)	(3b)	(3c)
n=5		
		
(3d)	(3e)	(3f)
n=6		

Figure 3: Effect of hyperparameter p on various performance measures for QAOA

Distribution of best 50 results by p (best configurations)	Average Optimality Gap by p (best configurations)	Percentage of feasible outcomes by p (all configurations)

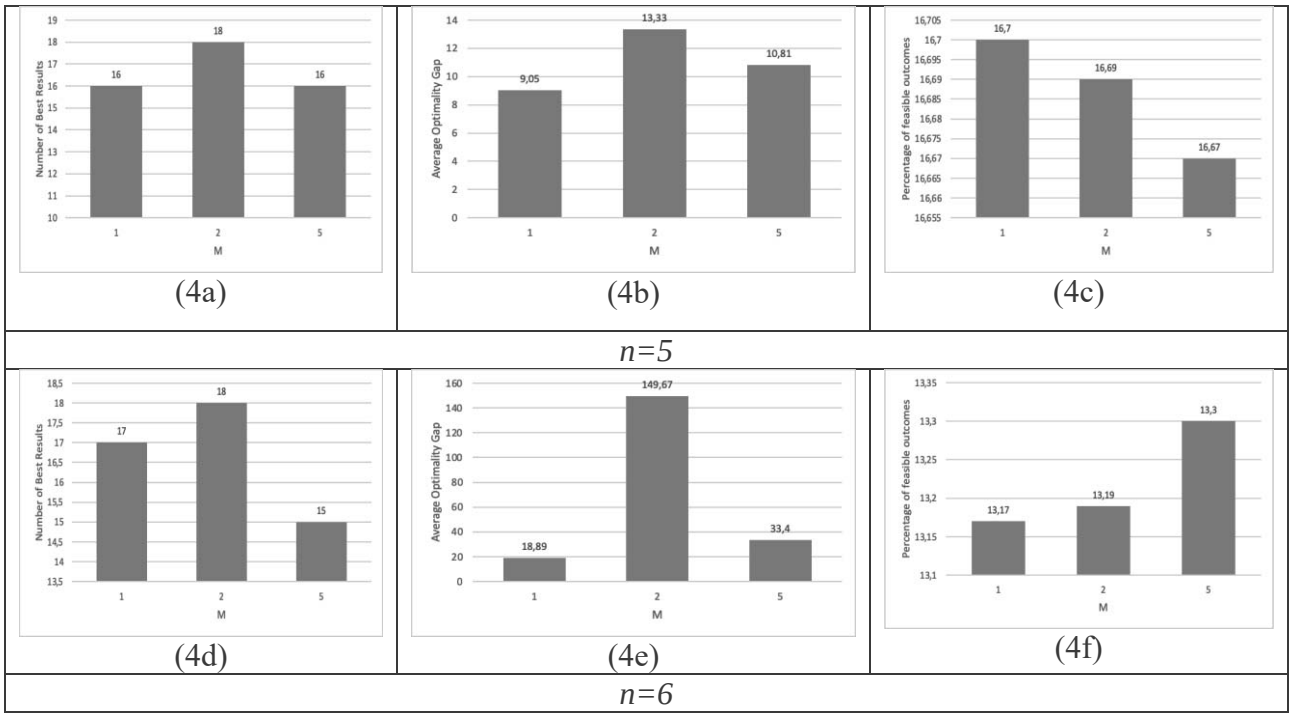


Figure 4: Effect of hyperparameter M on various performance measures for QAOA

10. CONCLUSION

The Single Machine Weighted Tardiness Problem (SMWTP) is an important NP-hard problem in the scheduling literature that has been extensively studied with classical algorithms but rarely approached with quantum methods. In this paper, we investigate the applicability of QAOA to the SMWTP. The SMWTP is formulated as a QUBO problem. A warm-start strategy based on the WSPT heuristic is employed to initialize the quantum state; and a comprehensive grid search is conducted over circuit depth, penalty scaling, and variational parameters to investigate the effect of hyperparameters on quantum computing performance. Results are benchmarked against Gurobi (exact solver) across 50 instances each for $n = 5$ and $n = 6$ jobs.

The experimental results offer several insights into QAOA behavior for scheduling problems. For $n = 5$, QAOA achieves the optimal solution in 17 out of 50 instances with an average optimality gap of 11.1%, and For $n = 6$, the average gap increases substantially, reflecting the exponential growth of the search space from 2^{25} to 2^{36} basis states. The hyperparameter analysis reveals that the mixer parameter β is the primary determinant of feasible solution recovery for both n=5 and n=6 problem sizes. Results indicate using $\beta = 0{,}1$ for n=5 and n=6 problem sizes, both to decrease optimality gap and to increase the percentage of feasible solutions. As for hyperparameter γ, $\gamma = 0.6$ or $\gamma = 0.7$ can be used for the SMWTP to decrease the optimality gap in the solutions. γ does not seem to have any specific effect on the percentage of feasible solutions, as well as p not having a specific effect on neither average optimality gap, nor the percentage of feasible solutions. M=1 provided favorable results for optimality gap, but no clear effect was observed for the percentage of feasible solutions.

REFERENCES

Bozejko, W., Klempous, R., Pempera, J., Rozenblit, J., Smutnicki, C., Uchroński, M. Wodecki, M., (2024). *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* DOI 10.1109/TSMC.2024.3458873.

Bozejko W., Pempera J., Uchronski M., Wodecki M. (2022). Distributed quantum annealing on d-wave for the single machine total weighted tardiness scheduling problem. *International Conference on Computational Science,* 171–178 (Springer), URL http://dx.doi.org/10.1007/978-3-031-08760-8_15.

Bozejko W., Pempera J., Uchronski M., Wodecki M. (2023). An exact quantum annealing-driven branch and bound algorithm for maximizing the total weighted number of on-time jobs on a single machine. Pawelczyk, M., Bismor, D., Ogonowski, S., Kacprzyk, J., eds., *Advanced, Contemporary Control*, (709) *Lecture Notes in Networks and Systems* (Cham: Springer), URL http://dx.doi.org/10.1007/978-3-031-35173-0_8.

Bozejko W., Trotskyi S., Uchronski M., Wodecki M. (2025). Optimizing two-machine scheduling in flexible manufacturing systems using autonomous ai and quantum computing. *Neurocomputing* 132067, URL http://dx.doi.org/10.1016/j.neucom.2025.132067.

Farhi E., Goldstone J., Gutmann S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 URL http://dx.doi.org/10.48550/arXiv.1411.4028.

Grange C., Poss M., Bourreau E., T'kindt V., Ploton O. (2024). Moderate exponential-time quantum dynamic programming across the subsets for scheduling problems. URL https://hal. science/hal-04296238v3, hAL Id: hal-04296238.

Grange C., Poss M., Bourreau E., T'kindt V., Ploton O. (2025). Moderate exponential-time quantum dynamic programming across the subsets for scheduling problems. *European Journal of Operational Research*, 320(3):516–526, URL http://dx.doi.org/10.1016/j.ejor.2024.09.005.

Wang, Y.S, Cheng, J. (2022). A quantum optimization algorithm for single machine total weighted tardiness minimization. arXiv preprint arXiv:2209.13712 URL http://dx.doi.org/10.48550/arXiv.2209.13712.

Werner, T., Ullinger, F. (2025). Quantum algorithms for scheduling problems: A survey. https://www.researchgate.net/profile/Tino-Werner-2/publication/393228512_Quantum_ algorithms_ for_scheduling_ problems_A_survey/links/6863bc36b991270ef3008d6e/ Quantum-algorithms-for-scheduling-problems-A-survey.pdf, accessed: 2026-02-07.