A Gate-Based Quantum Approach to Optimal University Course Scheduling

Erika Ramirez, Ryan Wang, Nick Pinon

***Abstract*—University course scheduling is a classic NP-Hard combinatorial optimization problem characterized by a factorial explosion in search space size as variables increase. While classical methods such as Integer Linear Programming (ILP) and heuristic solvers currently serve as the industry standard, they face significant computational scaling challenges and often struggle to prioritize soft constraints (preferences) within over-constrained environments. This paper investigates the application of the Quantum Approximate Optimization Algorithm (QAOA) to this domain, pivoting from adiabatic quantum annealing to a universal gate-based model to assess the viability of current Noisy Intermediate-Scale Quantum (NISQ) hardware. We formulate the scheduling problem as a Quadratic Unconstrained Binary Optimization (QUBO) model, incorporating hard constraints for room overlaps and instructor availability, alongside soft constraints for resource minimization. To address the qubit scarcity inherent in today's quantum devices, we introduce a hybrid pre-processing technique utilizing classical domain reduction ("pruning"), which reduced the effective problem size substantially. We benchmark the performance of our QAOA circuit executed on the Amazon Braket SV1 simulator against the Gurobi classical solver. Results indicate that while the classical solver identified the global minimum in < 0.01 seconds with 100% accuracy, the quantum solver failed to converge to a feasible solution, consistently returning high-energy states with multiple hard constraint violations. We attribute this failure to the low density of valid states within the search space. The shallow QAOA circuit struggled to locate the narrow feasible region amidst a vast landscape of high-energy penalties.**

***Index Terms*—Quantum Approximate Optimization Algorithm (QAOA), Constraint Satisfaction Problems (CSP), QUBO Optimization, University Timetabling, NISQ, Hybrid Quantum-Classical Computing, Gurobi.**

I. Introduction

THE University Course Scheduling Problem is a fundamental challenge in operations research and academic administration. Mathematically defined as a combinatorial optimization problem, it involves assigning a set of courses to specific time slots and physical classrooms while satisfying a complex web of constraints. These constraints are categorized into two tiers: "hard constraints," which represent inviolable rules such as preventing double-bookings of rooms or instructors and respecting room capacity limits; and "soft constraints," which represent preferences such as minimizing gaps in student schedules or maximizing the utilization of energy-intensive laboratory spaces. As the number of variables (courses, rooms, and time slots) increases, the computational complexity of the optimization problem grows factorially, classifying it as NP-Hard. In practical terms, this means that for large universities with wide course offerings, the search space of possible schedules becomes astronomically large, rendering brute-force search methods impossible.

The importance of this problem goes beyond solving a tedious and error-prone administrative task. A poor schedule has tangible physical costs: lighting, heating, and cooling empty lecture halls wastes significant energy and budget. Furthermore, these inefficiencies directly hurt students. When core courses overlap, students are often forced to delay graduation, turning a scheduling conflict into a life-altering setback. While classical tools like Integer Linear Programming (ILP) are currently the "gold standard" for checking if a schedule is possible, they tend to be rigid. When a university is over-crowded and no perfect schedule exists, classical software often forces administrators to arbitrarily drop preferences just to get the math to work. We are investigating Quantum Computing, specifically the Quantum Approximate Optimization Algorithm (QAOA), because it handles these "impossible" scenarios differently. By treating constraints as energy levels rather than binary rules, a quantum system can potentially find a "least-bad" compromise that balances conflicting needs, something rigid classical logic struggles to do.

This paper was produced as a part of the 18-619 Introduction to Quantum Computing course at Carnegie Mellon University.

II. The State of Research on the Course Scheduling Problem

The University Course Scheduling Problem (UCSP), frequently referred to in literature as the University Course Timetabling Problem (UCTTP), has been extensively studied in research. It is formally defined as a

multi-dimensional assignment problem where a set of events (courses, exams) must be paired with resources (rooms, time slots, instructors, etc). As the number of variables increases, the search space grows drastically. Important surveys, such as Schaerf (1999) demonstrates how almost all non-trivial variants of this optimization problem are classified as NP-Complete (encompassing NP-Hard), typically by reducing the problem to the classic Graph Coloring Problem where courses correspond to vertices and conflicts to edges [1].

For modern institutions with extensive course catalogs and enrollment, this combinatorial growth creates a search space that effectively prohibits brute-force enumeration. This classification means finding a globally optimal schedule given a set of constraints is often computationally infeasible. Consequently, various automated and semi-automated systems have been developed to address this challenge, shifting the focus from exhaustive search to sophisticated constraint satisfaction techniques.

The literature evaluates scheduling methods based on their ability to satisfy two distinct tiers of constraints. Hard constraints represent inviolable operational requirements such as preventing room double-bookings, respecting room capacity limits, and ensuring instructor availability [1]. A schedule that violates even a single hard constraint is considered operationally infeasible and cannot be deployed. Once feasibility is established, soft constraints are used to assess solution quality. Carter and Laporte (1998) note that soft constraints encode preferences such as minimizing gaps in student schedules, balancing instructor teaching loads, and consolidating courses to reduce energy consumption [2]. The multi-objective nature of these soft constraints, often with competing priorities, adds considerable complexity to the optimization landscape.

Work by Abdullah et al. (2007) has also highlighted the distinction between over-constrained and under-constrained problem instances [3]. In over-constrained scenarios, common at large universities, no feasible solution exists that satisfies all hard constraints simultaneously. Classical exact methods typically fail catastrophically in these cases, forcing administrators to manually relax constraints or remove courses [3]. This operational reality has motivated research into optimization approaches that can navigate infeasible regions and identify compromise solutions.

III. Current Solutions to this Problem

To address the computational complexity of the USCP, researchers and institutions have employed a spectrum of approaches ranging from exact mathematical programming to meta-heuristics.

A. Manual and Semi-Automated Approaches

Despite significant technological advances, Burke et al. (1996) note that many academic institutions continue to rely on manual or semi-automated scheduling processes [4]. These approaches typically involve domain experts using spreadsheet software or specialized scheduling tools to iteratively construct timetables. While such methods leverage human institutional knowledge, they are time-intensive and provide no optimality guarantees [4].

B. Exact Methods (Integer Linear Programming)

Integer Linear Programming (ILP) has emerged as the primary exact method for course scheduling and is widely regarded as the current industry standard for automated approaches. Daskalaki et al. (2004) demonstrate that by formulating the scheduling problem as a system of binary decision variables subject to linear equality and inequality constraints, ILP solvers can guarantee globally optimal solutions when they exist [5]. Modern commercial solvers such as Gurobi, CPLEX, and Xpress employ sophisticated branch-and-bound algorithms with cutting-plane methods to prune the search space efficiently [6]. This mathematical rigor comes with significant computational cost. As the number of decision variables grows corresponding to more courses, rooms, time slots, and students the time required to prove optimality increases exponentially [5]. For large universities with thousands of course sections, ILP solvers may require hours or even days to converge, and in many cases cannot find proven optimal solutions within practical time limits [5]. Furthermore, ILP formulations struggle with over-constrained instances where no strictly feasible solution exists, often terminating without producing any usable schedule [3].

C. Meta-Heuristic Approximations

To overcome the scaling limitations of exact methods, the majority of recent research has focused on meta-heuristic algorithms that trade optimality guarantees for computational tractability. Babaei et al. (2015) provide a comprehensive survey noting that Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search, and hybrid approaches dominate the literature [7]. These methods, as Babaei and colleagues explain, operate by stochastically exploring the solution landscape, using nature-inspired or physics-inspired mechanisms to escape local minima and converge toward high-quality feasible schedules [7].

Despite their empirical success, meta-heuristics suffer from well-documented limitations. A primary drawback

is the tendency to stagnate in local minima or regions of the search space where all immediate neighboring solutions have higher cost, even when superior global solutions exist elsewhere [7]. Algorithm performance is also highly sensitive to parameter tuning (e.g., mutation rates, cooling schedules), and optimal parameter settings vary significantly across problem instances [8]. Additionally, McCollum et al. (2010) emphasize that these methods provide no bounds on solution quality, making it impossible to know how far a returned solution is from the true optimum [9].

IV. Our Approach

1. Problem Definition

Let M be the set of meetings (courses), T be the set of time slots, and R be the set of rooms. We define the binary decision variable $x_{m,t,r}$ as:

$$x_{m,t,r} = \begin{cases} 1 & \text{if meeting } m \text{ is scheduled at time } t \text{ in room } r \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2. QUBO Formulation

The Quadratic Unconstrained Binary Optimization (QUBO) problem minimizes the following objective function $O(x)$, composed of Hard Constraints (must be satisfied) and Soft Constraints (optimization goals).

$$O(x) = A \cdot H_{\text{valid}} + A \cdot H_{\text{conflict}} + B \cdot S_{\text{capacity}} \tag{2}$$

Where A and B are penalty weights such that $A \gg B$.

Hard Constraint 1: Completeness: Every meeting m must be assigned to exactly one time slot and one room.

$$H_{\text{valid}} = \sum_{m \in M} \left(\sum_{t \in T} \sum_{r \in R} x_{m,t,r} - 1 \right)^2 \tag{3}$$

Hard Constraint 2: No Overlaps: No two meetings can occupy the same room r at the same time t.

$$H_{\text{conflict}} = \sum_{t \in T} \sum_{r \in R} \sum_{\substack{m_1, m_2 \in M \\ m_1 \neq m_2}} x_{m_1,t,r} x_{m_2,t,r} \tag{4}$$

Soft Constraint: Minimizing Wasted Space: We wish to minimize the unused seats in a room. Let Cap_r be the capacity of room r and En_m be the enrollment of meeting m.

$$S_{\text{capacity}} = \sum_{m \in M} \sum_{t \in T} \sum_{r \in R} (Cap_r - En_m) \cdot x_{m,t,r} \tag{5}$$

3. Mapping to Quantum Hamiltonian (QAOA)

To implement this on a Gate-Based Quantum Computer, we map the binary variables to spin operators using the transformation:

$$x_{m,t,r} \to \frac{1 - Z_i}{2} \tag{6}$$

where Z_i is the Pauli-Z operator acting on qubit i (corresponding to index tuple m, t, r).

The Problem Hamiltonian H_C is derived by substituting this transformation into the QUBO equation:

$$H_C = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j \tag{7}$$

where h_i (local fields) and J_{ij} (couplings) are determined by the expansion of the squared constraints.

The Mixer Hamiltonian H_B, used to drive transitions in the QAOA circuit, is the sum of Pauli-X operators:

$$H_B = \sum_{i=1}^{N} X_i \tag{8}$$

The QAOA ansatz evolves the state $|\psi(\gamma, \beta)\rangle$ by alternating these Hamiltonians:

$$|\psi(\gamma,\beta)\rangle = e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_B} e^{-i\gamma_1 H_C} |+\rangle^{\otimes N} \tag{9}$$

V. QAOA Circuit Construction

Constructing the circuit required implementing both Hamiltonians explicitly. We began by preparing a uniform superposition using Hadamard gates on all qubits so the algorithm would consider every possible schedule from the outset. Once this initialization was complete, the circuit alternated between encoding penalties from the scheduling problem and introducing exploratory dynamics that allowed the state to evolve across different configurations.

Cost Hamiltonian. Single-variable penalties appear in the Ising model as linear terms $h_i Z_i$. Each of these is implemented as a phase rotation

$$R_z(2\gamma h_i),$$

which shifts the phase of the computational basis state depending on whether that scheduling choice violates a constraint. Assignments associated with more severe penalties accumulate larger phase shifts.

Quadratic penalties require an interaction between qubits. A native ZZ gate is not available on Amazon Braket, so we synthesized the interaction

$$e^{-i\gamma J_{ij} Z_i Z_j}$$

using the standard three-step construction

$$\text{CNOT}(i, j), \quad R_z(2\gamma J_{ij}), \quad \text{CNOT}(i, j).$$

Only states that activate both qubits receive the corresponding phase shift, which recreates the quadratic structure of the constraint. To illustrate how these terms were implemented programmatically, we include a representative example.

Listing 1. Synthesis of Ising interactions within QAOA

```
def build_cost_terms(circ, h, J, gamma):
    n = len(h)

    # Linear terms: Rz rotations
    for i in range(n):
        if h[i] != 0:
            circ.rz(i, 2 * gamma * h[i])

    # Quadratic terms: ZZ interactions
    for i in range(n):
        for j in range(i + 1, n):
            if J[i, j] != 0:
                circ.cnot(i, j)
                circ.rz(j, 2 * gamma * J[
                    i, j])
                circ.cnot(i, j)
```

This routine forms the core of the phase-separation step in every QAOA layer and mirrors the mathematical structure of the QUBO.

Mixer Layer. After the cost Hamiltonian is applied, each qubit receives the rotation

$$R_x(2\beta),$$

which is the action of the mixer Hamiltonian. These rotations permit transitions between assignments by flipping qubits and encourage the algorithm to explore alternative schedules rather than remain near its initial configuration.

Circuit Depth. We experimented with depths $p = 1, 2, 3$. Increasing p introduces repeated applications of both Hamiltonians, which increases expressibility but also circuit depth. For clarity, the figures shown in this paper illustrate the $p = 1$ case, even though deeper circuits were executed during experimentation.

QAOA Circuit Diagram. Figure 1 shows the full QAOA layer produced by our implementation. The diagram visualizes the Hadamard initialization, the sequence of single-qubit R_z penalties, the synthesized ZZ interactions, and the mixer layer.

Preprocessing and Qubit Reduction. The raw QUBO contains many variables corresponding to logically impossible room–time assignments. Removing these entries through classical pruning reduced the circuit to twelve qubits. This reduction lowered the number of ZZ terms and made the circuit small enough to simulate on SV1 while remaining within the limits of current gate-based hardware.

VI. Results

We executed our QAOA circuits on Amazon Braket's noise-free SV1 simulator for depths $p = 1, 2, 3$. The algorithm consistently found configurations with lower energy than random baselines. However, none of the measured bitstrings produced a valid schedule. Conflicts persisted across all runs and included overlapping meetings, rooms used by multiple classes at the same time, missing assignments, and classes appearing in multiple slots within the same hour.

Even as the shot count increased, measurement distributions remained concentrated in infeasible regions. This trend suggests that QAOA was able to reflect the general shape of the energy landscape but did not reach the narrow feasible basin containing correct schedules.

To illustrate the behavior of the algorithm, Table I lists representative outputs from multiple depths.

TABLE I
Representative Schedules Produced by QAOA (All Infeasible)

Depth	Shots	Observed Issues
$p = 1$	500	Room conflicts, overlapping classes
$p = 2$	1000	Duplicate assignments, missing meetings
$p = 3$	3000	Over-capacity rooms, repeated classes

The table emphasizes that increasing depth did not resolve feasibility issues, even in a noise-free environment.

VII. Performance Evaluation and Benchmarking

To quantify the performance of our approach, we evaluated two metrics that are standard in both classical and quantum optimization research: (1) the energy of the returned bitstring under the QUBO objective, and (2) the feasibility of the resulting schedule with respect to the hard constraints. Lower energy provides a direct measure of progress toward the global optimum, while feasibility reflects whether the algorithm achieves a usable timetable rather than merely a low-cost configuration. We compared these outcomes against a classical Gurobi baseline, which achieved the global optimum in under 0.01 seconds for the reduced problem instance. This classical result provides a strong performance reference

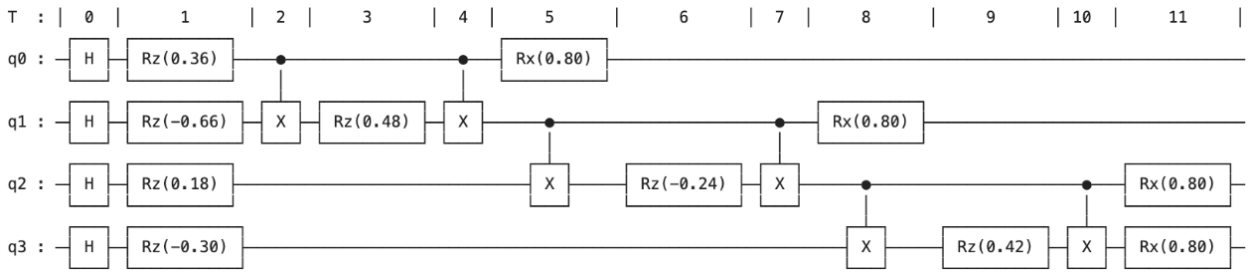

Fig. 1. Gate-level representation of the QAOA circuit for $p = 1$. The top register corresponds to qubit 0 and the bottom register to qubit 3. The diagram illustrates initialization, cost encoding, and the mixer layer.

point, allowing us to quantify the gap between the quantum and classical solutions both in energy and constraint satisfaction.

We also compared our implementation conceptually to recent quantum literature on QAOA-based scheduling, which typically reports that low-depth QAOA struggles on problems with sparse feasible regions. Our results align with these findings: the quantum circuit consistently produced high-energy, infeasible solutions despite parameter optimization and increased measurement counts.

Regarding hardware adequacy, current NISQ processors are not large enough to support the problem sizes where any quantum speedup is theoretically expected. Our reduced twelve-qubit model fits comfortably on existing gate-based platforms, but it is far smaller than the regime (roughly fifty to several hundred qubits) where quantum advantage may emerge for combinatorial optimization. Even with sufficient qubit count, the coherence times and two-qubit gate fidelities available on present hardware would make deep QAOA circuits difficult to execute reliably. For these reasons, our study relies on the Amazon Braket SV1 simulator, which isolates algorithmic limitations from the effects of hardware noise. Larger and more stable processors would be required to meaningfully benchmark QAOA on full-scale course scheduling instances.

VIII. Challenges Encountered

The most fundamental challenge arises from the shape of the search landscape. With twelve binary variables, the problem has $2^{12} = 4096$ possible schedules. Only a very small fraction of these are feasible. As a result, the cost landscape is nearly flat across most of the domain. This leads to a vanishing-gradient phenomenon where small adjustments to γ and β have little influence on the observed energy. The optimizer receives minimal information and often converges prematurely.

A second challenge is circuit expressibility. Real scheduling rules require correlations among several variables. Two-hour meetings require consecutive assignments. Room capacity interacts with instructor availability. Pairwise ZZ interactions can only represent part of this structure, and at shallow depths the circuit cannot synthesize higher-order correlations. This limitation restricts the circuit's ability to represent the true ground state.

The combination of a sparse feasible region and insufficient expressibility produced a situation where the algorithm had difficulty reaching low-energy basins, even though SV1 introduces no physical noise.

IX. Limitations of the Gate-Based Approach

Our initial plan involved running the problem on a quantum annealer. Such devices naturally embed QUBO couplings in hardware and minimize the energy through an analog evolution process. Access constraints required us to adopt a gate-based formulation instead. This alternative requires manual construction of every interaction: each linear term becomes an R_z operation and each quadratic term becomes a CNOT-based synthesis of a ZZ interaction.

This manual construction dramatically increases circuit depth and makes optimization more difficult. Although gate-based QAOA is a flexible algorithm, the scheduling problem demands multi-variable correlations that low-depth circuits cannot easily produce. Consequently, even a noise-free simulator struggled to find feasible solutions.

In short, the combination of a very small feasible region, minimal landscape curvature, limited circuit expressibility, and the constraints of gate-based decomposition explains why QAOA captured the structure of the problem but did not yield operational schedules for this instance.

X. Future Work

Future work should focus on testing larger problem instances on NISQ devices or high-capacity simulators.

The twelve-qubit model used in this study is too small to reveal the scaling behavior where quantum methods may start to diverge from classical solvers. Increasing the number of variables and examining deeper circuits would provide a clearer picture of QAOA's potential.

A second direction is to evaluate the scheduling QUBO on quantum annealers. These architectures encode pairwise couplings directly in hardware and may perform better on constraint-heavy optimization problems. Prior studies suggest that annealers sometimes succeed where gate-based QAOA struggles.

Finally, there are several ways to improve the expressiveness of the algorithm. Structured mixers, warm starts, and alternative encodings may help QAOA navigate the sparse feasible regions that characterize real scheduling tasks. Exploring these design choices would help determine whether quantum optimization can play a meaningful role in timetabling at larger scales.

XI. Conclusion

This project examined the feasibility of applying QAOA to the university course scheduling problem. We formulated the task as a QUBO, reduced its dimensionality through classical preprocessing, and implemented the resulting Ising Hamiltonian on a gate-based quantum circuit. While the algorithm was able to capture aspects of the problem structure, it consistently failed to reach feasible schedules, even in a noise-free simulation environment.

Our results suggest that the combination of a sparse feasible region, shallow circuit depth, and limited hardware resources prevents current gate-based QAOA from solving timetabling problems of this form. Classical solvers continue to outperform quantum approaches by a wide margin. Nonetheless, the study provides a useful baseline for future experiments on larger quantum systems and motivates continued exploration of annealing-based hardware and more expressive quantum ansätze.

References

[1] A. Schaerf, "A survey of automated timetabling," Artificial Intelligence Review, vol. 13, no. 2, pp. 87-127, 1999.

[2] M. W. Carter and G. Laporte, "Recent developments in practical course timetabling," Practice and Theory of Automated Timetabling II, pp. 3-19, 1998.

[3] S. Abdullah, E. K. Burke, and B. McCollum, "Using a randomised iterative improvement algorithm with composite neighbourhood structures for university course timetabling," Metaheuristics, pp. 153-169, 2007.

[4] E. K. Burke, D. G. Elliman, P. H. Ford, and R. F. Weare, "Examination timetabling in British universities: A survey," Practice and Theory of Automated Timetabling, pp. 76-90, 1996.

[5] S. Daskalaki, T. Birbas, and E. Housos, "An integer programming formulation for a case study in university timetabling," European Journal of Operational Research, vol. 153, no. 1, pp. 117-135, 2004.

[6] H. Mittelmann, "Decision tree for optimization software," INFORMS Journal on Computing, 2023.

[7] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," Computers & Industrial Engineering, vol. 86, pp. 43-59, 2015.

[8] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," OR Spectrum, vol. 30, no. 1, pp. 167-190, 2008.

[9] B. McCollum et al., "Setting the research agenda in automated timetabling," INFORMS Journal on Computing, vol. 22, no. 1, pp. 120-130, 2010.

[10] Kurowski, K., Pecyna, T., Slysz, M., Różycki, R., Waligóra, G., and Wglarz, J., "Application of Quantum Approximate Optimization Algorithm to Job Shop Scheduling Problem," *European Journal of Operational Research*, vol. 310, no. 2, pp. 518–528, 2023. doi: 10.1016/j.ejor.2023.03.013.

[11] Tirado-Domínguez, J. A., Gutiérrez, E., and Plata, O., "QTIS: A QAOA-Based Quantum Time Interval Scheduler," *arXiv preprint arXiv:2511.15590*, 2025. Available: https://arxiv.org/abs/2511.15590.

[12] Rudová, H., and Murray, K., "University Course Timetabling with Soft Constraints," in *Practice and Theory of Automated Timetabling IV*, Lecture Notes in Computer Science, vol. 2740, Springer, pp. 310–328, 2003.

[13] Amazon Braket Team, "QAOA Example Notebook," Available: https://github.com/amazon-braket/amazon-braket-examples/blob/main/examples/hybrid_quantum_algorithms/QAOA/QAOA_braket.ipynb. Accessed 2025.

[14] Farhi, E., Goldstone, J., and Gutmann, S., "A Quantum Approximate Optimization Algorithm," *arXiv preprint arXiv:1411.4028*, 2014.