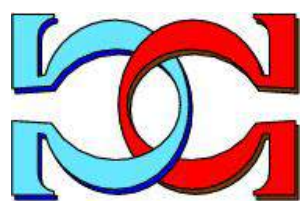
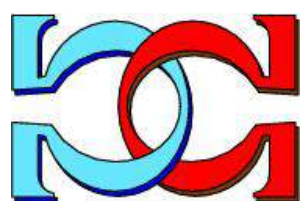
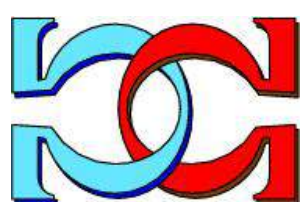


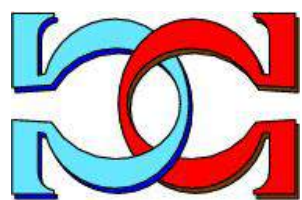
**CDMTCS  
Research  
Report  
Series**



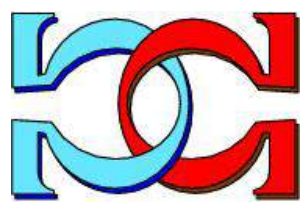
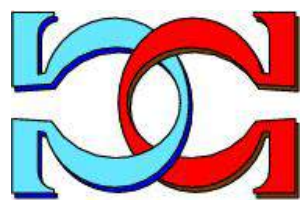
**Proceedings of the PRICAI 2025  
Workshop on Quantum  
Computing for Search and  
Optimization Problems**



**Phillipe Codognet**  
Sorbonne University / CNRS /  
University of Tokyo, Tokyo, Japan



**Cristian S. Calude**  
**Patrice Delmas**  
**Michael J. Dinneen**  
School of Computer Science,  
University of Auckland,  
Auckland, New Zealand



CDMTCS-586  
April 2025

Centre for Discrete Mathematics and  
Theoretical Computer Science

**Proceedings of the PRICAI 2025 Workshop on  
Quantum Computing for Search and Optimization Problems**  
Wellington, New Zealand, 17 November 2025

**QCSOP25**

Organized by:

Philippe Codognet, Sorbonne University / CNRS / University of Tokyo, Tokyo, Japan

Cristian Calude, University of Auckland, New Zealand

Patrice Delmas, University of Auckland, New Zealand

Michael Dinneen, University of Auckland, New Zealand

# Contents

- *Optimizing Qubit Mapping via Spectral Ordering of Input Graphs for QAOA Max-Cut Circuit*  
Devraj, IIT Hyderabad, Telangana, India  
Sankardeep Chakraborty, University of Tokyo, Japan  
Kunihiko Sadakane, University of Tokyo, Japan  
Vorapong Supakitpaisarn, University of Tokyo, Japan
- *Towards a Quantum Resolution of the Knapsack Problem*  
Monsef Hamitouche, ESI, Algiers, Algeria  
Faicel Hnaien, University of Technology of Troyes, France  
Djamel Eddine Menacer, ESI, Algiers, Algeria
- *Quantum Annealing for Computing Bandwidth: QUBO Formulations and Solution Strategies*  
Qinyu Guo, University of Auckland, New Zealand  
Michael J. Dinneen, University of Auckland, New Zealand
- *Quantum Fuzzy Inference Systems: Implementation and a Case Study on Sleep Apnea Detection*  
Samuel Magaz-Romero, Universidade da Coruna, Spain  
Eduardo Mosqueira-Rey, Universidade da Coruna, Spain  
Diego Alvarez-Estevez, Universidade da Coruna, Spain  
Vicente Moret-Bonillo, Universidade da Coruna, Spain
- *Benchmarking Under Fragility in a Shifting Quantum Landscape*  
Nathan Kittichaikoonkij, Chulalongkorn University, Bangkok, Thailand  
Nutthapat Pongtanyavichai, Chulalongkorn University, Bangkok, Thailand  
Poopha Suwananek, Chulalongkorn University, Bangkok, Thailand  
Prabhas Chongstitvatana, Chulalongkorn University, Bangkok, Thailand  
Kamonluk Suksen, Chulalongkorn University, Bangkok, Thailand
- *GRACE: A Grover-Enhanced Actor-Critic Framework for Complex Robotic Manipulation*  
Eirini Panteli, PrioriAnalytica, Adelaide, Australia  
Paulo Santos, PrioriAnalytica, Adelaide, Australia  
Belinda Chiera, University of South Australia, Adelaide  
Josh Chopin, University of South Australia, Adelaide

# Optimizing Qubit Mapping via Spectral Ordering of Input Graphs for QAOA Max-Cut Circuit

Devraj<sup>1</sup>, Sankardeep Chakraborty<sup>2</sup>, Kunihiro Sadakane<sup>2</sup>, and Vorapong Suppakitpaisarn<sup>2</sup>

<sup>1</sup> Indian Institute of Technology Hyderabad, Telangana, India  
es22btech11011@iith.ac.in

<sup>2</sup> The University of Tokyo, Tokyo, Japan  
sankardeep.chakraborty@gmail.com, sada@mist.i.u-tokyo.ac.jp,  
vorapong@is.s.u-tokyo.ac.jp

**Abstract.** The efficiency of quantum circuit execution on near-term quantum processors is significantly influenced by the initial mapping of logical qubits to physical qubits, particularly due to limited connectivity and routing overhead. In this work, we propose a spectral heuristic based on the Fiedler vector — the second eigenvector of the graph Laplacian — to derive a meaningful ordering of logical qubits. This ordering is used to assign logical qubits to physically adjacent qubits on hardware. We demonstrate that this approach improves circuit compilation quality and reduces SWAP overhead for graph-structured problems such as MaxCut on benchmark graphs. Experimental evaluations using QAOA circuits on IBM quantum backends show that Fiedler-based mapping consistently reduces the number of SWAP gates and improves the overall fidelity of output distributions. Our method is classical, deterministic, and integrates seamlessly with existing transpilation workflows, making it a practical tool for NISQ-era quantum applications.

**Keywords:** Qubit mapping · Spectral graph theory · Fiedler vector · QAOA · MaxCut problem.

## 1 Introduction

As quantum computing hardware continues to evolve, near-term quantum processors, often referred to as Noisy Intermediate-Scale Quantum (NISQ) devices, remain constrained by limited qubit connectivity, high gate error rates, and restricted qubit counts [13]. In most hardware architectures, not all qubits are fully connected. Consequently, two-qubit gates (e.g., CNOTs) between non-adjacent qubits require insertion of SWAP gates, introducing additional overhead. These limitations require efficient compilation strategies to reduce circuit depth and error accumulation.

One critical stage in quantum compilation is qubit routing [3], i.e., the process of inserting SWAP gates and reordering qubit assignments in a quantum circuit so that all required two-qubit operations can be executed on a hardware

architecture with restricted qubit connectivity, while minimizing additional circuit depth and error overhead.

Several works have shown that the initial layout—the first assignment of logical qubits to physical qubits before routing—is a major factor in determining the total number of SWAPs and overall fidelity of circuit execution [11,20]. We call the problem of finding an initial mapping of logical qubits to physical qubits in the real hardware architecture, the Qubit Mapping Problem. While heuristic and search-based methods like SABRE [8] and VF2Layout [14] attempt to find favorable mappings dynamically or via subgraph isomorphism, there is still a need for simple, fast, and deterministic strategies that provide good starting layouts for structured problems.

While previous approaches typically focus on generic optimization techniques, we argue that leveraging the known combinatorial structure of the problem can lead to more effective quantum circuit optimization. For example, some works suggest the graph sparsification technique to reduce the number of gates in the QAOA circuit for the Max-Cut problem [10,18].

### 1.1 Our Contributions

In this work, we explore a spectral graph-theoretic approach to the Qubit Mapping problem. Specifically, we leverage the Fiedler vector—the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix of a graph—to determine an ordering of logical qubits in a quantum circuit. The Fiedler vector is widely used in spectral partitioning and layout problems due to its ability to capture global structure and approximate graph distances [1,17]. By assigning logical qubits based on their Fiedler coordinates to a path or tree-shaped subset of the physical topology, we aim to minimize long-range interactions and routing overhead.

We evaluate this method in the context of QAOA [4] circuits for the Max-Cut problem on various classes of graphs, including random regular graphs and Barabasi-Albert graphs. Our experiments on IBM Quantum backends demonstrate that Fiedler-based initial mappings yield lower SWAP counts and improved fidelity compared to SABRE-based layouts.

In Section 2, we discuss briefly about quantum computing, the Qubit Routing Problem, QAOA and spectral graph theory. In Section 3, we go through the intuition behind our method. In Section 4, we show the experimental results of our method in comparison with the other existing mapping strategies. In Section 5, we acknowledge the limitations and threats to validity that our method faces. Finally, in Section 6, we discuss possible further improvements and future research on this topic and conclude.

## 2 Preliminaries

### 2.1 Quantum Computing

Quantum computing [12] harnesses the fundamental principles of quantum mechanics, offering a new paradigm for information processing. Unlike classical bits,

quantum bits (qubits) can exist in superpositions of states and can be entangled, enabling quantum computers to perform certain computations more efficiently than their classical counterparts.

Quantum information is encoded in qubits, which are realized using physical systems such as trapped ions, superconducting circuits, or photonic devices. The state of a qubit is described by a vector in a two-dimensional Hilbert space, and the state of an  $n$ -qubit system is a vector in a  $2^n$ -dimensional space. Quantum operations are implemented via unitary transformations, and quantum algorithms are constructed as sequences of these operations, forming quantum circuits.

In this paper, we focus on gate-based superconducting circuits and conduct our experiments using IBM Quantum devices.

## 2.2 Quantum Circuits and NISQ Constraints

A quantum circuit consists of a sequence of quantum gates applied to qubits, culminating in measurements. In the current era of Noisy Intermediate-Scale Quantum (NISQ) devices [13], the performance of quantum hardware is characterized by limited qubit counts, gate fidelity issues, and sparse qubit connectivity. These limitations make it essential to optimize circuit structure before execution.

Most hardware platforms impose architectural constraints, allowing two-qubit operations (like CNOT gates) only between physically connected qubits. Also, these two-qubit operations are usually quite expensive in terms of error-rates and execution times. Longer execution times make the circuit more susceptible to decoherence, i.e., qubits losing their quantum state over time due to interactions with the environment. Accordingly, minimizing the number of two-qubit operations is essential for optimizing circuit performance on contemporary quantum hardware.

## 2.3 Qubit Mapping and Routing

To execute a quantum circuit on hardware, a compiler must map logical qubits from the circuit to physical qubits on the device. This is called the Initial Mapping. However, even with a good initial mapping, it is often the case that a two-qubit gate in the circuit requires interacting qubits that are not adjacent in the physical topology. In such cases, SWAP gates need to be inserted in the circuit to bring the logical qubits into proximity in the hardware topology. A SWAP gate between two qubits interchanges the quantum states of the two qubits.

Since the SWAP gate is itself a two-qubit gate (equivalent to 3 alternating CNOTs), we can only apply swap gates between pairs of qubits that are physically connected in the hardware architecture.

The Qubit Routing Problem is thus two-fold:

1. Mapping: Assigning logical qubits to physical qubits in the hardware architecture.
2. Routing: Inserting SWAP gates in the logical circuit to allow the execution of two-qubit gates on logical qubits mapped to non-adjacent physical qubits by relocating them to adjacent physical qubits.

The qubit mapping and routing process is known to be NP-complete [3,6,7], and has been the focus of many compiler optimizations, including heuristic and noise-aware strategies [8,19].

As discussed earlier, the SWAP gates that are inserted in the routing process are composed of multiple two-qubit gates and hence, are quite expensive to execute on real quantum hardware. Therefore, minimizing the number of SWAP gates in the routed circuit is a critical objective in the evaluation of qubit routing strategies and has been the subject of substantial research in the field.

The mapping and routing problems are interdependent: a good initial mapping can significantly reduce the routing overhead and the number of SWAPs in the routed circuit.

## 2.4 QAOA for the Max-Cut Problem

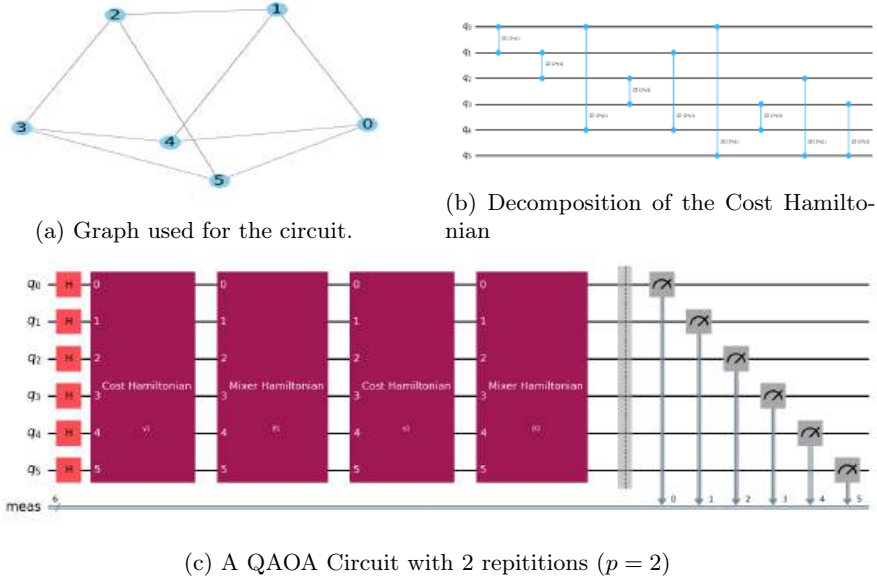


Fig. 1: A QAOA Circuit in Qiskit, a quantum computing development platform of IBM

One promising use case for NISQ devices is the Quantum Approximate Optimization Algorithm (QAOA) [4], which solves combinatorial optimization problems like MaxCut. QAOA consists of executing parameterized quantum gates, namely the Cost and Mixer operators (Figure 1c), on qubits corresponding to nodes of the input graph, with parameters tuned by classical optimization. The

approximation quality of the solution given by QAOA depends on the number of repetitions ( $p$ ) of the Cost and Mixer Operators.

The implementation of QAOA is typically divided into two phases. In the first phase, the Estimator is employed in conjunction with a classical optimizer to determine an optimal set of parameters. The Estimator evaluates the expectation value of the cost Hamiltonian with respect to the parameterized quantum state by executing the quantum circuit multiple times (shots) and averaging the measurement outcomes. This expectation value serves as the objective function for the optimizer, which iteratively updates the parameters to minimize it. Once convergence is achieved, the second phase utilizes the Sampler to obtain a probability distribution over bitstrings by repeatedly executing the optimized quantum circuit. The bitstrings with the highest probabilities correspond to candidate solutions for the Max-Cut problem.

In the QAOA circuits for the Max-Cut problem, each node of the input graph corresponds to a logical qubit in the quantum circuit. Each edge between two nodes of the graph in Figure 1a corresponds to a two-qubit entangling gate (the RZZ gate) of the Cost Hamiltonian between the corresponding logical qubits, as can be seen in Figure 1b. Since RZZ gates are commuting operations, their application order within the circuit does not affect the resulting unitary transformation. Hence, the structure of the QAOA circuit is very intricately tied to the input graph, and this dependency can be leveraged to derive effective initial qubit mappings that reduce SWAP overheads.

## 2.5 Spectral Properties of Graphs and the Fiedler Vector

Spectral graph theory [1] studies the properties of graphs through the eigenvalues and eigenvectors of matrices associated with them, such as the adjacency matrix or the Laplacian matrix.

Given a graph  $G = (V, E)$ , its (unnormalized) Laplacian matrix is defined as  $L = D - A$ , where  $D$  is the matrix, of which the diagonal line is degrees of each node, and  $A$  is the adjacency matrix of the graph. The Laplacian is a symmetric, positive semi-definite matrix, and its eigenvalues contain information about the connectivity and structure of  $G$ .

The Fiedler vector is the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian matrix  $L$ . It solves the following optimization problem:

$$\min_{x \perp \mathbf{1}, \|x\|=1} x^T L x$$

which is equivalent to minimizing the quadratic form:

$$\sum_{(i,j) \in E} (x_i - x_j)^2, \quad \text{subject to } \sum_i x_i = 0, \quad \sum_i x_i^2 = 1$$

where  $x$  is a real-valued vector defined on the nodes of the graph,  $E$  is the set of edges, and  $\mathbf{1}$  is the all-ones vector. This property ensures that two nodes  $i$  and  $j$  that are adjacent or strongly connected typically have similar values of  $x_i$  and  $x_j$ .



The Fiedler vector plays a central role in spectral partitioning and embedding [17], as it captures the most significant mode of variation in the graph's connectivity structure. Intuitively, nodes with similar Fiedler vector values tend to share structural and topological proximity, i.e., they are in similar positions with respect to the graph structure. In other words, the Fiedler vector provides a one-dimensional embedding of nodes according to the spectral properties of the graph.

### 3 Method

We propose a hardware topology-aware initial mapping strategy tailored for QAOA circuits of the Max-Cut problem, which minimizes routing overhead by combining spectral insights from the input graph with structural proximity on the physical qubit topology.

#### 3.1 Properties of the Fiedler Vector

As mentioned earlier, nodes with similar values of the Fiedler vector tend to be in similar positions within the graph. This can be seen in Figure 2 where the nodes that are close to each other seem to be colored in similar colors, i.e., they have similar Fiedler vector values. This seems to be especially true for graphs that have a long, path-like structure, or that are made of multiple such structures, such as in Figure 2a. Thus, one may think of sorting the nodes of the graph according to their corresponding Fiedler vector values to get a promising initial mapping.

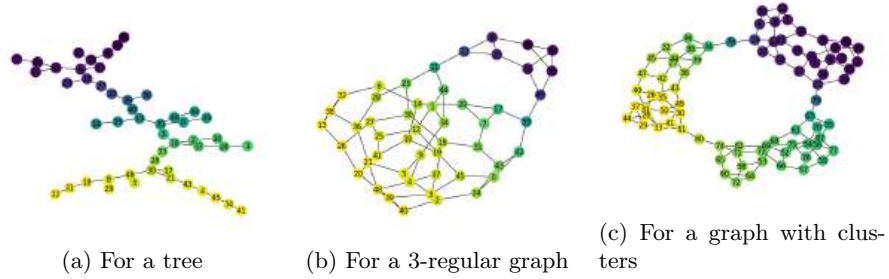


Fig. 2: Heatmap of the nodes of graphs with respect to the Fiedler vector values of the graphs. (Nodes with similar colors have similar Fiedler vector values)

However, this does not mean that nodes having similar Fiedler vector values will necessarily be close to one another. This is evident in Figure 2c where the nodes connecting two clusters (nodes 78 and 79) have similar colors despite being quite far from one another, as the Fiedler vector only provides a one-dimensional

embedding while a given graph may spread out in more than one dimensions. In this case, the Fiedler vector assigns similar values to nodes being in similar positions in the graph structure, even if they are distant.

Therefore, directly using the Fiedler vector of the graph for the initial qubit mapping may be suboptimal, as it can result in assigning graph nodes with similar Fiedler vector values, yet potentially distant in the original topology, to adjacent physical qubits. This misalignment can lead to excessive SWAP operations during routing, thereby degrading performance while execution.

### 3.2 DFS Trees of the Input graph

As we have already discussed, the Fiedler Vector orders the nodes according to their connectivity quite well for graphs with a path-like structure. So, to have such an ordering (nodes sorted according to Fiedler vector values) for graphs that do not have such a structure, we can extract subgraphs of the original graph that are path-like. A Depth-First-Search Spanning Tree (DFS Tree) of a graph is a subgraph that excellently satisfies this requirement. A DFS Tree of a graph is simply a subgraph that is created through the Depth-First-Search Traversal of the original graph until all the nodes of the graph have been visited.

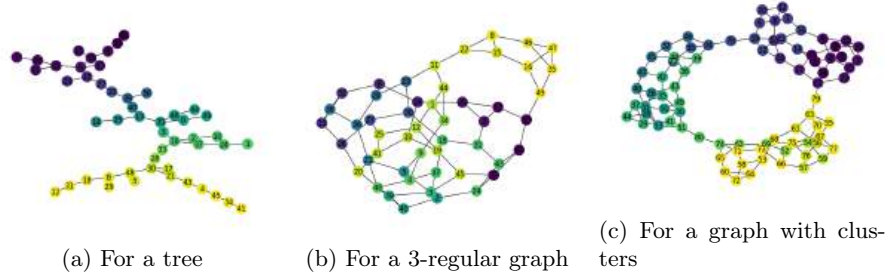


Fig. 3: Heatmap of the nodes of graphs with respect to the Fiedler vector values of a DFS Tree of the graphs.

Now that we have a DFS Tree of the graph, we can order the nodes of the graph according to the Fiedler vector values of the DFS Tree. This will ensure that the nodes that are close to each other in the ordering, i.e., having similar values in the Fiedler vector of the DFS Tree, are actually close to each other in the original graph. This is due to the inherently linear and narrow branching nature of the DFS tree. The Fiedler vector, being a one-dimensional embedding, naturally captures and reflects the hierarchical traversal path. Consequently, nodes with similar Fiedler values are likely to correspond to topologically adjacent or nearby nodes in the original graph.

In Figure 3, we can see that the issue of distant nodes having being close in the ordering is resolved with the use of DFS Trees. One false alarm could be the

abrupt change of color in some pairs of adjacent nodes, such as nodes 11 and 23 in Figure 3b. However, this is acceptable since we only want nodes that are close in the ordering to be actually nearby in the original graph. Achieving the converse—that all nearby nodes in the graph are close in the ordering—is not always feasible. This is because the hardware topologies of real world QPUs with more than 100 qubits are very sparse, and even for moderately dense graphs, for example, 5-regular graphs, we cannot map all the nodes adjacent to a node (5 nodes) to physically adjacent qubits in the hardware (which have a maximum degree of 3 and minimum degree of 1).

### 3.3 Extracting a Connected Subgraph of the Hardware Topology

Now that we have a ordering of nodes of the problem graph, we need a set of physical qubits to map the nodes to. Let  $G = (V, E)$  represent the problem graph,  $H = (Q, C)$  represent the layout of the physical hardware topology and  $S = (s_1, s_2, \dots, s_n)$ ,  $s_i \in V$  denote the  $n$ -tuple ( $n = |V|$ ) of nodes sorted according to their corresponding Fiedler vector value of the DFS Tree of  $G$ . A path of  $n$  qubits in  $H$  may seem like a natural choice, given the one-dimensional embedding nature of the Fiedler vector. However, this would not be optimal since the diameter of the subgraph would be  $n - 1$ , meaning if the first and last nodes in  $S$  happen to be adjacent in  $G$  the distance between them in the subgraph would be  $n - 1$ , which can be improved.

We can try to reduce the diameter of the resulting subgraph by choosing  $n - 1$  nodes of  $H$  that are nearest to a node  $s \in Q$ . We can do this with a Breadth-First-Search(BFS) Traversal of  $H$  with  $s$  as the source. For the regular, lattice-like heavy hex structure of IBM QPUs, BFS is a reliable heuristic for finding subgraphs of low diameter. Thus, we can simply choose the first  $n - 1$  qubits that are visited in the BFS traversal of  $H$  from a random source  $s$ . Let the resulting subgraph be  $H_n = (Q_n, C_n)$ .

We have conducted preliminary experiments in which the source qubit  $s$  was selected based on device-specific calibration data, focusing on regions of the qubit topology with relatively low noise. While no significant improvements were observed under our current experimental constraints, we expect that advantages may become evident for larger and deeper circuits, which are presently impractical due to noise and execution time limitations. However, we do not include this approach in our proposed method due to its current lack of feasibility.

### 3.4 Mapping Logical Qubits to Physical Qubits

We now have the set of physical qubits in  $H$  that the nodes of  $G$  will be mapped to. But, the subgraph  $H_n$  is two-dimensional while the ordering  $S$  is one-dimensional. For this we require a linear ordering of  $H_n$ . We also want that the qubits that are close in  $H_n$  are not very far apart in the linear ordering. We can apply the same idea of extracting path-like structures that was used in Section 3.2 to find an initial mapping. A DFS traversal of  $H_n$ , starting from the qubit in  $Q_n$  corresponding to the source  $s$  of the BFS traversal in Section 3.3, is

expected to yield a linear ordering of qubits that tends to avoid placing qubits, that are close in  $H$ , far apart in the ordering.

We map the nodes of  $G$ , according to  $S$ , to the qubits in  $H_n$  in the order that they are visited by the DFS traversal. Let  $D = (d_1, d_2, \dots, d_n)$ ,  $d_i \in Q_n$  be the nodes in  $Q_n$  in the order that they were visited by the DFS traversal of  $H_n$  with  $s$  as the source. Then, the initial mapping is given by  $M = \{(s_1, d_1), (s_2, d_2), \dots, (s_n, d_n)\}$ .

---

**Algorithm 1** Fiedler-Based Initial Qubit Mapping

---

**Require:** Graph  $G = (V, E)$  with  $n$  nodes, Hardware topology graph  $H = (Q, C)$  with  $|Q| \geq n$

**Ensure:** Initial mapping  $\mathcal{M} : V \rightarrow Q$

- 1: Build a Depth-First Search (DFS) tree  $T_{\text{DFS}}$  from  $G$
  - 2: Compute the Fiedler vector  $\mathbf{f}$  of  $T_{\text{DFS}}$
  - 3: Sort nodes of  $G$  according to their corresponding values in  $\mathbf{f}$  to obtain ordering  $(s_1, s_2, \dots, s_n)$
  - 4: Randomly select a root qubit  $q_0 \in Q$
  - 5: Use Breadth-First Search (BFS) on  $H$  starting from  $q_0$  to find a connected subgraph  $H_n$  of  $n$  qubits
  - 6: Perform a DFS traversal of  $H_n$  starting from  $q_0$  to obtain ordered list of physical qubits  $(d_1, d_2, \dots, d_n)$
  - 7: Assign each node  $s_i$  to qubit  $d_i$  to form the mapping  $\mathcal{M}(s_i) = d_i$
  - 8: **return** Mapping  $\mathcal{M}$
- 

### 3.5 Exploiting the Commutativity of RZZ Gates

As mentioned in Section 2.4, the order in which RZZ gates (corresponding to the same cost operator) are present in the circuit is irrelevant. We can exploit this fact by building circuits tailored to the initial mapping that is being used. An effective approach is to schedule the RZZ gates according to the order of edges in  $E$ , where the edges are sorted by the distance between their associated physical qubits as determined by the initial mapping  $M$ . Specifically, edges whose endpoints correspond to physically proximate qubits under  $M$  are applied first, thereby prioritizing gates that can be executed with minimal routing overhead, which effectively reduces the number of SWAPs in the fully routed circuit.

## 4 Experiments

This section presents a comparative evaluation of our proposed mapping strategy, i.e. Fielder Layout (along with the circuit building strategy in Section 3.5), against SabreLayout [15], a mapping strategy currently employed in Qiskit, and a baseline trivial mapping. The trivial mapping maps logical qubit  $v_1$  to physical qubit  $q_1$ , and so on. SABRE is selected for comparison as it is the current default layout heuristic employed in Qiskit’s transpilation framework and is widely adopted in current quantum compilation workflows.

#### 4.1 Experimental Setup

The experiments were performed with widely-used benchmark graph families, namely random regular graphs and Barabasi-Albert (BA) graphs. Hardware topologies of real-world NISQ devices provided by IBM Quantum, such as IBM Fez, were used in the form of coupling graphs, which represent the connectivity constraints of the device. Each QAOA circuit was constructed with a single repetition of the cost and mixer Hamiltonians ( $p = 1$ ). This setting aligns with standard practice in prior work, as increasing the number of QAOA repetitions significantly increases circuit depth, thereby amplifying the effects of decoherence and gate noise on NISQ devices. To isolate the impact of the initial mapping, all circuits, regardless of the mapping strategy, were routed using SabreSwap [16], which is the default routing pass in Qiskit’s transpiler. The number of nodes/qubits in the comparative graphs has been limited to 100, reflecting current constraints, as the publicly available IBM Quantum hardware currently supports up to 156 qubits. We use Qiskit’s transpiler with optimization level 3 to prepare the circuits for execution. For the default Qiskit transpilation, no explicit initial qubit mapping is specified, whereas for our proposed approach, the Fiedler-based mapping is explicitly provided as the initial mapping.

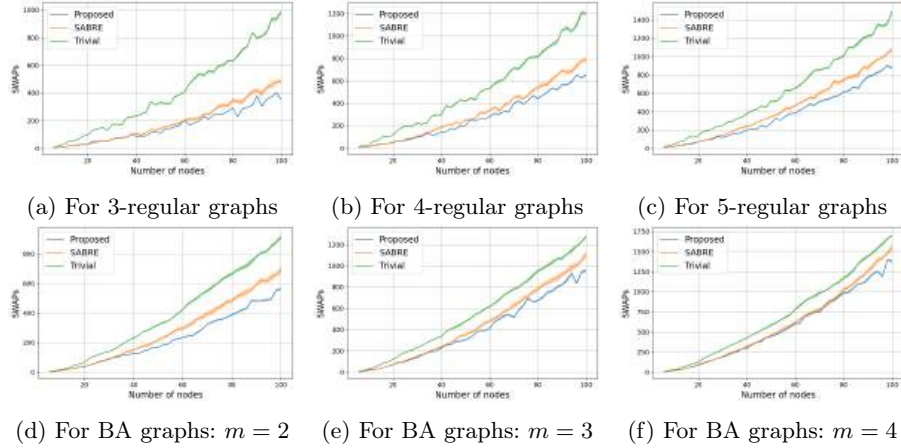


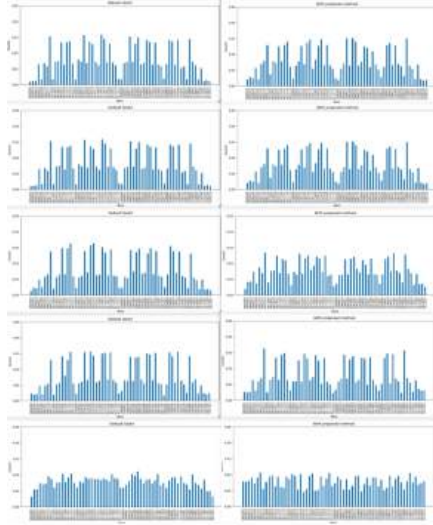
Fig. 4: SWAP counts for regular graphs (top) and Barabási-Albert (BA) graphs (bottom). Solid lines show means; shaded bands indicate  $\pm 1$  SD over 10 runs.

For the local experiments (counting SWAPs and circuit simulations), a laptop having 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz and 16GB memory was used. The experiments on real quantum hardware were conducted using IBM Fez, a 156-qubit device equipped with the Heron r2 processor, which is currently IBM’s most advanced quantum processor.

## 4.2 Reduction in SWAPs

Figure 4 shows the comparison of number of SWAP gates inserted by SabreSwap across 10 instances with initial mappings given by a trivial mapping, SabreLayout and Fiedler Layout, for random regular graphs and BA graphs respectively, targeting the coupling map of IBM Fez. From the figures it is evident that both SabreLayout and Fiedler Layout give a significant improvement over the trivial mapping. Also, for smaller graphs (with vertices fewer than 40), SabreLayout and Fiedler Layout give similar SWAP counts. However, for graphs with more than 40 vertices, Fiedler Layout gives a clear improvement over SabreLayout that only gets more pronounced as the size of the graph increases. For BA graphs, as the graphs become quite dense with the value of the attachment parameter  $m$  equal to 4, there is more overlap between SabreLayout and Fiedler Layout, however, improvement is seen for graphs with more than 80 vertices, as can be seen in Figure 4f.

## 4.3 Performance of Routed Circuit on Real Hardware



| Method          | Mean   | Std. Dev |
|-----------------|--------|----------|
| Qiskit Default  | 8.1024 | 0.4061   |
| Fiedler Mapping | 8.0769 | 0.3586   |
| Random Cut      | 7.5    | 0.0      |

Table 1: Expected Cut Sizes given by the histograms in Figure 5.

Fig. 5: Output distributions on 6 qubits given by a circuit with default qiskit transpilation (left column) and a circuit with the proposed mapping (right column), when run on IBM Fez on five different instances.

The circuits for the experiments were constructed on randomly generated 5-regular graphs with 6 vertices. Each circuit execution involved 1000 shots for both the Estimator and the Sampler components, where “shots” refer to repeated runs of the quantum circuit to compute an expectation value or obtain a measurement outcome distribution. To account for the time-dependent and stochastic nature of noise in quantum hardware, this experimental configuration was executed five times. The chosen circuit depth and graph size were converged upon while keeping expressibility and hardware limitations in mind, as larger graphs or deeper circuits proved infeasible due to increased susceptibility to noise and extended execution times.

Figure 5 presents the histograms obtained by sampling the measurement outcomes of the optimized circuits 1,000 times across five separate executions for each method. As observed, both columns exhibit similar behavior in terms of noise susceptibility, which is reflected in the degree of uniformity within a histogram. This shows that our method performs comparably to the standard Qiskit compilation approach on small-scale instances. No notable improvement is observed likely due to both methods yielding similar SWAP gate counts for small graphs, as shown in Figure 4c. While our approach demonstrates advantages at larger scales—around 40 qubits—these configurations remain impractical to execute on current quantum hardware due to significant noise levels that may obscure any benefits from circuit-level optimization.

#### 4.4 Performance of Routed Circuit on Simulator

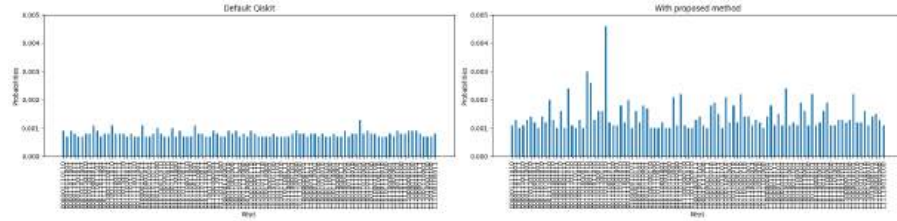


Fig. 6: Mean Output distributions given by a circuit with default qiskit transpilation (left) and a circuit with the proposed mapping (right), when run on FakeBrisbane on 100 different instances.

| Method          | Mean    | Std. Dev |
|-----------------|---------|----------|
| Qiskit Default  | 15.8065 | 0.2019   |
| Fiedler Mapping | 17.0627 | 0.3103   |
| Random Cut      | 15.0    | 0.0      |

Table 2: Expected Cut Sizes given by the histograms in Figure 6.

Simulations were done on FakeBrisbane, a ‘FakeBackend’ simulator [5] provided by Qiskit having configurations, like the coupling graph and noise effects, based on IBM Brisbane. The circuits for the experiment were generated on a 5-regular graph with 12 vertices, with  $p = 1$ . We could not experiment with larger graphs due to the usual constraints of time and noise, and the fact that unlike the linear scaling on quantum computers, execution times for the circuits scale exponentially with respect to the number of qubits for simulators [2]. The configuration of the Estimator and the Sampler were the same as in Section 4.3, with 1000 shots each. The setup was ran for 100 times on the simulator due to varying results on different instances.

Figure 6 presents the mean probability distributions obtained by optimizing and sampling the circuits over 100 independent instances. For clarity, only the top 100 bitstrings (by probability) are displayed for each distribution, as visualizing the complete set would obscure the highest peaks due to scale compression. The impact of noise is more pronounced in the histogram corresponding to the circuit transpiled using the default Qiskit method, as evidenced by the near-uniform distribution of measurement outcomes.

## 5 Limitations and Threats to Validity

This study, while demonstrating practical improvements in qubit mapping strategies for QAOA circuits, is subject to several limitations. First, our method lacks formal theoretical guarantees or proofs regarding optimality or performance bounds. The results are empirical and rely on specific benchmark settings, which may limit generalizability. Second, the current mapping procedure does not incorporate any noise-aware optimization. That is, device-specific calibration data such as readout errors, gate fidelities, or decoherence times are not taken into account, which could significantly impact the performance of the compiled circuits on real hardware.

Moreover, due to resource constraints and the limited number of reliable qubits on current quantum hardware, our experiments were restricted to small graph instances. As a result, we were unable to test the method’s scalability on larger circuits where performance differences may be more pronounced. Another source of limitation is the use of a fixed depth QAOA ansatz with only one layer of cost and mixer operators; deeper circuits may require different optimization strategies and could be more sensitive to mapping choices and noise.

Lastly, while our method is evaluated against a default Qiskit mapping strategy, we do not benchmark against all state-of-the-art qubit routing algorithms. Broader comparisons would provide a more comprehensive assessment of effectiveness.

## 6 Conclusion and Future Studies

In this paper, we presented a Fiedler vector-based initial qubit mapping technique for QAOA circuits, designed to reduce SWAP overheads and improve hard-



ware compatibility on NISQ devices. Our method leverages the spectral properties of the problem graph to derive a ordering of nodes that preserves proximity. This ordering is then aligned with a BFS-DFS traversal of the hardware graph to generate an initial mapping that minimizes expected routing cost.

Comparing our method with SabreLayout in terms of SWAP counts shows significant improvement for sparse graph instances with more than 40 vertices. Experimental results on small random 5-regular graphs suggest that the proposed strategy performs competitively with Qiskit’s default mapping method. While hardware evaluations are currently limited by the number of available qubits and sensitivity to noise, simulations indicate that Fiedler Layout holds potential for improved scalability and circuit efficiency in larger QAOA instances. The competitiveness/improvement shown in terms of noise susceptibility of circuit executions implies that the proposed method integrates seamlessly with the current Qiskit transpilation workflow.

There are several directions for extending this work. First, the current method operates in a noise-agnostic setting; integrating noise-aware heuristics or calibration data from specific hardware backends could further improve performance. Additionally, while our method focuses on static initial mapping, it can be combined with more sophisticated dynamic routing strategies to enhance gate scheduling.

One promising avenue is the incorporation of dynamic circuit building during QAOA compilation. This would involve iteratively constructing the circuit layer by layer [9], inserting only those gates that are currently executable based on the qubit mapping and topology, and applying SWAP layers selectively to unlock additional executable gates. Such an approach could adaptively minimize SWAP depth and gate congestion while preserving the circuit’s logical structure. Exploring this hybrid dynamic-static strategy is a natural progression toward more hardware-efficient QAOA compilation.

Finally, we aim to evaluate this method on deeper QAOA circuits and more structured problem graphs, as well as extend the approach to support other combinatorial optimization problems such as the Traveling Salesman Problem, as well as broader classes of variational quantum algorithms.

## Acknowledgement

This research is partly supported by JST NEXUS Grant Number Y2024L0906031, KAKENHI Grant JP25K00369, and MEXT - Quantum Leap Flagship Program (MEXT Q-LEAP) Development of quantum software by intelligent quantum system design and its applications. The results presented in this paper were obtained using an IBM Quantum computing system as part of the IBM Quantum Hub at The University of Tokyo.

## References

1. Chung, F.R.K.: Spectral Graph Theory, CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society (1997)

2. Cicero, A., Maleki, M.A., Azhar, M.W., Kockum, A.F., Trancoso, P.: Simulation of quantum computers: Review and acceleration opportunities (2025), <https://arxiv.org/abs/2410.12660>
3. Cowtan, A., Dilkes, S., Duncan, R., Krajenbrink, A., Simmons, W., Sivarajah, S.: On the qubit routing problem. In: TQC 2019 (2019)
4. Farhi, E., Goldstone, J., Gutmann, S., Zhou, L.: The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick model at infinite size. *Quantum* **6**, 759 (2022)
5. IBM Quantum: Qiskit IBM Runtime - Fake Provider Documentation (2024), <https://quantum.cloud.ibm.com/docs/en/api/qiskit-ibm-runtime/fake-provider>, accessed: 2025-07-27
6. Ito, T., Kakimura, N., Kamiyama, N., Kobayashi, Y., Okamoto, Y.: Algorithmic theory of qubit routing. In: WADS 2023. pp. 533–546 (2023)
7. Ito, T., Kakimura, N., Kamiyama, N., Kobayashi, Y., Okamoto, Y.: Algorithmic theory of qubit routing in the linear nearest neighbor architectures. *ACM Transactions on Quantum Computing* **6**(3), 1–17 (2025)
8. Li, G., Ding, Y., Xie, Y.: Tackling the qubit mapping problem for NISQ-era quantum devices. In: ASPLOS 2019. pp. 1001–1014 (2019)
9. Matsuo, A., Yamashita, S., Egger, D.J.: A SAT approach to the initial mapping problem in swap gate insertion for commuting gates. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **106**(11), 1424–1431 (2023)
10. Moondra, J., Lotshaw, P.C., Mohler, G., Gupta, S.: Promise of graph sparsification and decomposition for noise reduction in QAOA: Analysis for trapped-ion compilations. *arXiv preprint arXiv:2406.14330* (2024)
11. Murali, P., Linke, N.M., Martonosi, M., Abhari, A.J., Nguyen, N.H., Alderete, C.H.: Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In: ISCA 2019. pp. 527–540 (2019)
12. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information*. Cambridge University Press (2000)
13. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
14. Qiskit Contributors: Vf2layout pass. <https://qiskit.org/documentation/stubs/qiskit.transpiler.passes.VF2Layout.html> (2023)
15. Qiskit Development Team: SabreLayout pass in Qiskit Transpiler. IBM Quantum (2024), <https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.transpiler.passes.SabreLayout>
16. Qiskit Development Team: SabreSwap pass in Qiskit Transpiler. IBM Quantum (2024), <https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.transpiler.passes.SabreSwap>
17. Spielman, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. In: FOCS 1996. pp. 96–105 (1996)
18. Suppakitpaisarn, V., Hao, J.K.: Utilizing graph sparsification for pre-processing in max cut QUBO solver. In: MIC 2024. pp. 219–233 (2024)
19. Zou, H., Treinish, M., Hartman, K., Ivrii, A., Lishman, J.: LightSABRE: A lightweight and enhanced SABRE algorithm (2024), <https://arxiv.org/abs/2409.08368>
20. Zulehner, A., Paler, A., Wille, R.: An efficient methodology for mapping quantum circuits to the ibm qx architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **38**(7), 1226–1236 (2018)

# Towards a Quantum Resolution of the Knapsack Problem

Monsef Hamitouche<sup>1,2</sup>, Faicel Hnaien<sup>2</sup>, and Djamel Eddine Menacer<sup>1</sup>

<sup>1</sup> Higher National School of Computer Science (ESI), Oued Smar, Algiers, 16309, Algeria

`{km_hamitouche,d_menacer}@esi.dz`

<sup>2</sup> University of Technology of Troyes (UTT), LIST3N, 12 Rue Marie Curie, 10300 Troyes, France

`faicel.hnaien@utt.fr`

**Abstract.** Combinatorial optimization problems in NP face exponential growth in feasible solutions, making classical methods like branch-and-bound or dynamic programming quickly intractable. This work introduces a Hybrid Branch-and-Bound (HBB) framework for the binary knapsack problem, combining classical pruning with a Grover-based quantum subroutine. Constraints are encoded via reversible quantum oracles using QFT-based adders and comparators. An enhanced version (EHBB) reduces qubit usage by 53% (from 58 to 27 for 5 items) by reusing ancillae and uncomputing intermediates, without increasing circuit depth. Simulations on Qiskit and Atos myQLM validate the method up to 10 items, showing the promise of translating constraint programming techniques into quantum computing.

**Keywords:** Quantum Computing · Hybrid Algorithms · Combinatorial Optimisation · Constraint Programming · Binary Knapsack Problem

## 1 Introduction

Combinatorial optimisation problems are central to both theoretical computer science and real-world decision-making. These problems require selecting the best solution from a large, often exponentially growing, set of possibilities. As the problem size increases, classical techniques such as branch-and-bound or dynamic programming quickly become impractical [7]. This motivates the search for alternative approaches capable of tackling such complexity.

Quantum computing provides a promising direction in this context. Since the foundational work of Shor [10] and Grover [6], quantum algorithms have shown the potential to outperform classical methods for certain tasks. In particular, Grover’s algorithm provides a quadratic speedup for unstructured search [2,3], and has been widely explored for optimisation [8,4]. However, much of the current research remains theoretical and assumes noiseless, fully connected hardware. In the current NISQ era [9], feasibility must be reassessed under noise, limited connectivity, and gate errors.

In this work, we revisit the classical binary knapsack problem and propose a hybrid quantum approach that combines classical pruning with Grover-based search. Drawing inspiration from constraint programming, we translate feasibility and optimisation constraints into reversible quantum circuits using arithmetic components like QFT-based adders [5] and comparators [1]. We further introduce an enhanced version (EHBB) that reduces qubit usage by over 50% through ancilla reuse and uncomputation, while maintaining circuit depth.

Our main contributions can be summarised as follows:

1. We design modular feasibility and optimisation oracles using Draper QFT adders and QFT comparators.
2. We develop a Hybrid Branch-and-Bound (HBB) framework embedding these oracles into a Grover search, enabling parallel subtree evaluation.
3. We propose an enhanced version (EHBB) that significantly reduces qubit count, improving scalability for NISQ devices.
4. We validate our approach through simulations on instances up to 10 items, and provide a discussion on hardware constraints, potential quantum advantage, and directions for future experimentation.

## 2 Background

This section recalls the foundations required in the remainder of the paper: (i) constraint programming (CP) as a modelling paradigm for discrete optimisation, and (ii) basic concepts in quantum computing that underpin our oracle design.

### 2.1 Constraint Programming

Constraint Programming (CP) provides a declarative framework in which a problem is specified through variables with finite domains, a set of constraints restricting feasible assignments, and, when relevant, an objective function to optimise. CP solvers combine systematic search with constraint propagation to eliminate infeasible states and reduce the exploration space. The 0–1 Knapsack Problem (KP) is a canonical benchmark in this setting; its formal definition is provided in Section 3.

### 2.2 Quantum Computing Basics

Quantum computation encodes information into qubits and evolves states through unitary operations. Core principles include superposition, entanglement, and reversibility. Quantum algorithms are typically structured into three phases: state preparation, unitary evolution, and measurement.

Grover’s search algorithm [6,2,3] provides a quadratic speedup for unstructured search. It relies on a reversible oracle that marks feasible solutions and a diffusion operator that amplifies their amplitudes. For constrained optimisation problems, such oracles require arithmetic modules; in our work, we employ Draper QFT adders [5] and QFT comparators [1].

In the NISQ era [9], hardware noise, decoherence, and limited connectivity impose severe resource constraints. This motivates careful circuit design with ancilla reuse and uncomputation to control qubit count and depth.

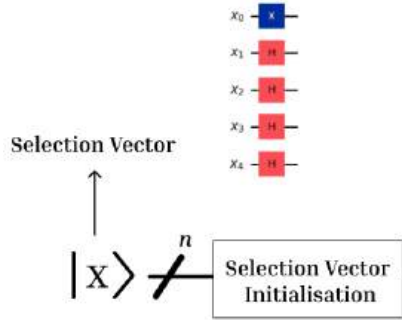
### 3 Problem Formulation

The 0–1 Knapsack Problem (KP) is defined as follows [7]. Given  $n$  items, each with weight  $w_i$  and value  $v_i$ , and a knapsack capacity  $C$ , the task is to select a binary vector  $x \in \{0, 1\}^n$  such that:

$$\max \sum_{i=1}^n v_i x_i, \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i x_i \leq C, \quad x_i \in \{0, 1\}. \quad (2)$$

*Quantum Encoding.* Each decision variable  $x_i$  is mapped to a single qubit: 0 denotes exclusion and 1 inclusion of item  $i$ . Applying Hadamard gates to all  $n$  qubits creates a uniform superposition of all  $2^n$  possible solutions. Figure 1 illustrates the register initialisation. Arithmetic operations and comparators are introduced in Section 4.



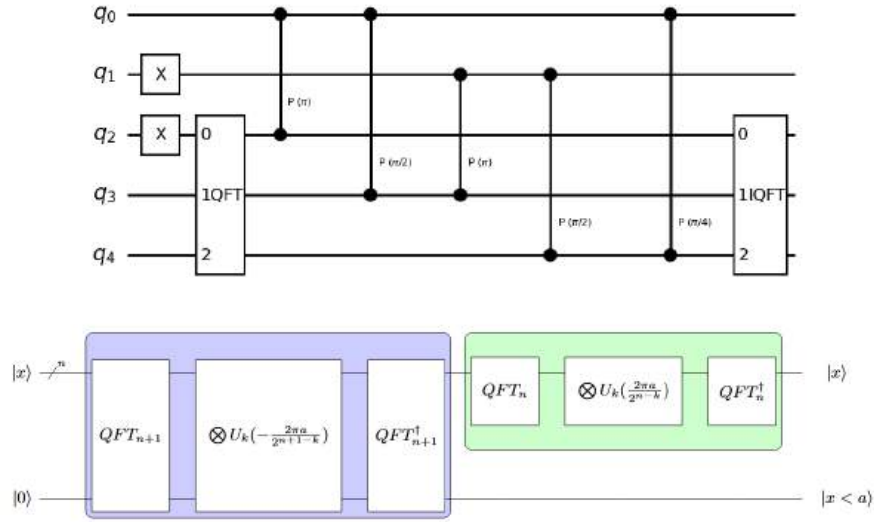
**Fig. 1.** Selection-vector initialisation for the knapsack problem. Each qubit encodes the inclusion or exclusion of an item.

### 4 Quantum Feasibility Oracle

The feasibility oracle is the central component of our Hybrid Branch-and-Bound framework. It encodes the knapsack constraints (capacity and value thresholds) in a reversible quantum circuit that integrates selection, summation, comparison, and validity flagging. This section describes how the oracle is assembled, referencing its building blocks and their composition.

#### 4.1 Arithmetic Components

The oracle relies on modular arithmetic circuits to process weights and values. Specifically, Draper QFT adders are used for efficient quantum addition [5], and QFT-based comparators are used to implement inequality checks [1]. Figure 2 shows their implementation in Qiskit: the left subfigure illustrates the Draper adder, while the right subfigure presents the QFT comparator.



**Fig. 2.** Qiskit implementation of arithmetic components: (Up) Draper QFT adder; (Bottom) QFT-based comparator.

*Adder design choice.* To motivate the use of Draper adders, Table 1 compares a standard carry-ripple adder with the Draper QFT adder for an  $s$ -bit sum. The ripple adder requires linear depth and multiple carry ancillas, while the Draper adder achieves logarithmic depth at the cost of Fourier rotations. In our design, the reduction in ancilla count is critical for NISQ feasibility.

**Table 1.** Comparison of  $s$ -bit adders: ripple vs. Draper QFT. Gate counts are approximate asymptotics; depth refers to circuit depth in  $s$ .

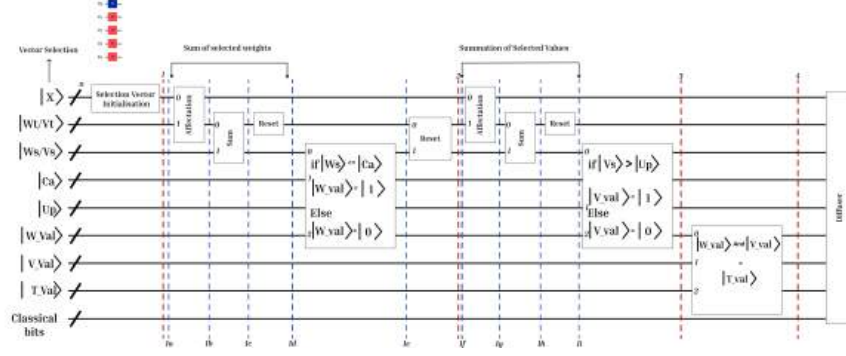
| Metric         | Ripple Adder       | Draper QFT Adder                  |
|----------------|--------------------|-----------------------------------|
| Depth          | $\mathcal{O}(s)$   | $\mathcal{O}(\log s)$             |
| Ancilla qubits | $\mathcal{O}(s)$   | 0 (no extra ancillas)             |
| Gate types     | CNOT, Toffoli      | QFT rotations + CNOT              |
| Gate count     | $\mathcal{O}(s)$   | $\mathcal{O}(s \log s)$           |
| Strength       | Simple, robust     | Shallower depth, ancilla-free     |
| Weakness       | High ancilla usage | Requires high-precision rotations |

This comparison explains our design choice: although Draper adders use more rotation gates, their absence of carry ancillas and reduced depth make them preferable for circuits where qubit footprint is the bottleneck.

## 4.2 Oracle Structure

The oracle integrates these arithmetic submodules into a single reversible circuit. The process is structured as follows (refer to Figure 3):

1. **Selection vector initialization:** Each item is placed in superposition using Hadamard gates, with the first item fixed to break symmetry (Step 1).
2. **Weight accumulation:** Draper adders sum the weights of selected items into a dedicated register (Steps 1a–1c).
3. **Capacity check:** The accumulated weight is compared against the knapsack capacity using the QFT comparator, producing the weight validity flag (Steps 1d–1e).
4. **Value accumulation:** A second Draper adder accumulates the item values in a separate register (Steps 1f–1h).
5. **Value threshold check:** The accumulated value is compared with a user-defined minimum bound to filter trivial solutions (Step 1i).
6. **Final validity flag:** Both conditions (capacity and value) are combined to produce the oracle flag qubit (Step 2), which is then used by the Grover diffuser.



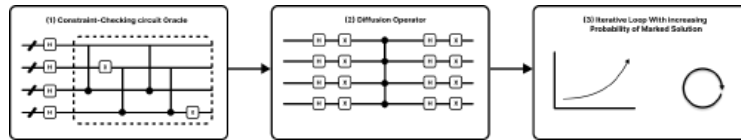
**Fig. 3.** Integrated feasibility oracle for the 0/1 knapsack problem (5-item instance). The schematic shows stepwise construction: Steps 1a–1c: weight accumulation; Steps 1d–1e: capacity check; Steps 1f–1h: value accumulation; Step 1i: value threshold check; Step 2: final validity flag.

## 5 Hybrid Branch-and-Bound Strategy

We couple Grover’s search [6,2,3] with a classical branch-and-bound (B&B) scheme. At a node with fixed prefix  $x_{1:k} = b_{1:k}$ , the oracle of Section 4 marks feasible states among the remaining variables  $x_{k+1:n}$  that also exceed the current threshold  $v_{\min}$ . A short Grover schedule amplifies these states; if an improvement is observed,  $v_{\min}$  is updated and propagated down the tree.

### Grover at a node

For a node with fixed prefix  $x_{1:k} = b_{1:k}$ , we run Grover on the unfixed register  $x_{k+1:n}$ . One iteration consists of: (i) initialisation to the uniform superposition, (ii) the oracle that marks *feasible and improving* assignments (capacity satisfied and value  $> v_{\min}$ ), and (iii) the diffusion about the mean followed by measurement. The complete workflow is shown once in Figure 4.

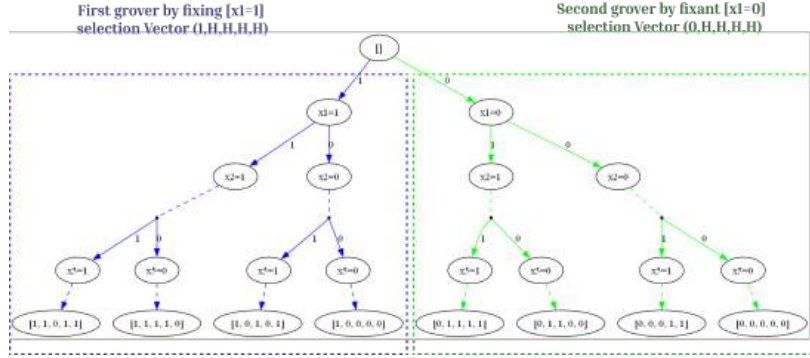


**Fig. 4.** Grover workflow at a B&B node (single illustration): initialisation → oracle marking (*feasible & improving*) → diffusion → measurement. This single figure replaces the previous two-panel layout.



### Branching scheme

We use binary branching on the next decision qubit,  $x_k \in \{1, 0\}$  (include vs. exclude). Each branch runs its own Grover evaluation on  $x_{k+1:n}$  (conceptually in parallel, or sequentially on one device). The split at the root is depicted in Figure 5.



**Fig. 5.** Root split with two Grover evaluations: left subtree fixes  $x_1 = 1$ , right subtree fixes  $x_1 = 0$ . Each subtree calls the oracle on the remaining variables and updates the incumbent when an improvement is found.

### Node routine and threshold update

The node computation combines a classical bound with a short Grover schedule:

---

#### Algorithm 1 Node( $b_{1:k}, v_{\min}$ )

---

- 1: Compute an upper bound  $U(b_{1:k})$  (e.g., fractional-knapsack bound).
  - 2: **if**  $U(b_{1:k}) \leq v_{\min}$  **then**
  - 3:     **return** PRUNE
  - 4: **end if**
  - 5: Prepare  $\frac{1}{\sqrt{2^n - k}} \sum x_{k+1:n} x_{k+1:n}$ .
  - 6: **for**  $r \in \{1, 2, \dots, R_{\max}\}$  **do**
  - 7:     Apply one Grover iteration using the oracle of Section 4.
  - 8:     Measure a candidate  $x^*$ ; classically evaluate  $V(x^*)$  and feasibility.
  - 9:     **if** feasible and  $V(x^*) > v_{\min}$  **then**
  - 10:          $v_{\min} \leftarrow V(x^*)$ ; store  $x^*$  as incumbent; **break**
  - 11:     **end if**
  - 12: **end for**
  - 13: Recurse on  $x_k = 1$  then  $x_k = 0$  (depth-first), passing  $v_{\min}$ .
-

*Iterations  $R$ .* When an estimate of the marked fraction  $M/N$  is available, we follow the standard choice  $R \approx \lfloor \frac{\pi}{4} \sqrt{N/M} \rfloor$ . Otherwise we adopt a short increasing schedule ( $R_{\max} \in [2, 6]$  in our experiments) with early stopping as soon as  $v_{\min}$  improves. This maintains depth under NISQ constraints.

### Enhanced variant (EHBB)

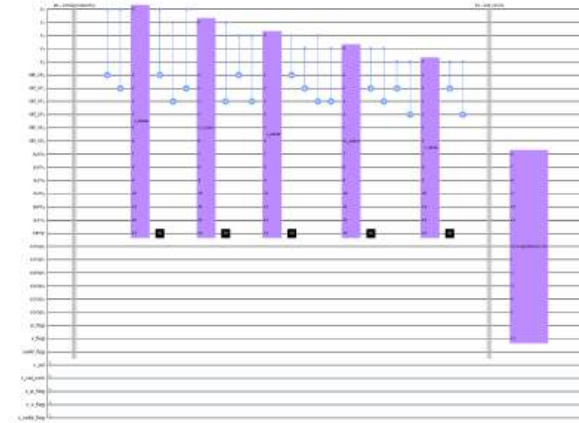
To reduce the qubit footprint, EHBB aggressively reuses ancillas and uncomputes intermediate registers between oracle calls. Only the selection qubits and a small set of flags persist across nodes; weight/value accumulators and comparator ancillas are reset and recycled. This halves the qubit count relative to the baseline (see Section 6) while preserving oracle depth.

### Correctness and pruning

Soundness follows from the oracle semantics: capacity feasibility is enforced by the QFT comparator and improvement by the threshold test; the classical bound prunes subtrees that cannot beat  $v_{\min}$ . Completeness holds because every unpruned prefix is explored by branching on  $x_k \in \{1, 0\}$ ; on termination, the incumbent equals the optimal value for the explored instance.

## 6 Simulations and Results

We validated our modular oracles in Qiskit on Aer/statevector simulators for small knapsack instances ( $n \leq 10$ ). We first verified the integrated feasibility oracle, and then analysed post-measurement distributions obtained after embedding the oracle in a single Grover iteration.



**Fig. 6.** Full feasibility oracle for  $n = 5$ : selection initialisation, weight accumulation, capacity comparison, and validity flag. This matches the annotated schematic in Figure 3.

### Measurement outcomes

**High-level behaviour.** After one Grover iteration, the measurement histograms concentrate probability mass on *feasible* selections (capacity satisfied) and suppress *infeasible* selections (capacity violated). Figure 7 shows two representative summaries for  $n = 5$ : the left panel aggregates counts on the selection register, while the right panel reports the same shots with derived classical features (bitstring, total weight, carry).

| x1 | x2 | x3 | x4 | x5 | Counts |
|----|----|----|----|----|--------|
| 1  | 1  | 0  | 0  | 1  | 51     |
| 0  | 1  | 0  | 0  | 1  | 75     |
| 2  | 1  | 0  | 0  | 0  | 73     |
| 15 | 1  | 0  | 1  | 1  | 70     |
| 4  | 1  | 1  | 1  | 0  | 67     |
| 12 | 1  | 1  | 1  | 1  | 67     |
| 14 | 1  | 1  | 0  | 0  | 67     |
| 8  | 1  | 0  | 1  | 0  | 64     |
| 11 | 1  | 0  | 1  | 0  | 64     |
| 9  | 1  | 0  | 0  | 1  | 63     |
| 7  | 1  | 1  | 0  | 1  | 59     |
| 3  | 1  | 1  | 1  | 1  | 58     |
| 5  | 1  | 1  | 0  | 0  | 58     |
| 6  | 1  | 1  | 1  | 0  | 57     |
| 13 | 1  | 0  | 1  | 1  | 53     |
| 10 | 1  | 1  | 0  | 1  | 48     |

| x bits (MSB-LSB) | wt, no bits | carry  | Total weight | Shots |
|------------------|-------------|--------|--------------|-------|
| 1                | 00011       | 000100 | 3            | 52    |
| 16               | 00011       | 000101 | 5            | 50    |
| 4                | 00101       | 000100 | 8            | 73    |
| 6                | 01001       | 000101 | 7            | 52    |
| 5                | 00111       | 001001 | 8            | 65    |
| 12               | 10011       | 010100 | 10           | 70    |
| 8                | 10001       | 001011 | 11           | 74    |
| 10               | 01101       | 001011 | 11           | 75    |
| 11               | 10011       | 001100 | 14           | 74    |
| 13               | 01111       | 001100 | 14           | 74    |
| 3                | 10101       | 001111 | 15           | 67    |
| 9                | 11001       | 010000 | 16           | 57    |
| 6                | 00111       | 010010 | 18           | 71    |
| 14               | 10011       | 010011 | 19           | 64    |
| 2                | 11101       | 010100 | 20           | 60    |
| 1                | 11111       | 010111 | 23           | 62    |

**Fig. 7.** Representative post-measurement summaries for  $n = 5$ . **Left:** counts on the selection register; **right:** the same shots tabulated with derived totals (weight, carry). Feasible selections dominate.

**Detailed distribution at a fixed capacity.** To make the separation explicit, Figure 8 reports the full distribution for  $C = 14$ . The optimal feasible assignment achieves the highest amplitude; assignments exceeding the capacity appear with visibly smaller probabilities.

| Combinations meeting capacity constraint (Total weight $\leq 14$ ): |              |       |   |
|---------------------------------------------------------------------|--------------|-------|---|
| x bits (MSB-LSB)                                                    | Total weight | Shots |   |
| 6                                                                   | 00011        | 5     | 1 |
| 13                                                                  | 00101        | 6     | 2 |
| 5                                                                   | 01001        | 7     | 4 |
| 2                                                                   | 00111        | 9     | 1 |
| 10                                                                  | 01011        | 10    | 1 |
| 0                                                                   | 01101        | 11    | 1 |
| 12                                                                  | 10001        | 11    | 1 |
| 4                                                                   | 01111        | 14    | 1 |
| 1                                                                   | 10011        | 14    | 3 |

| Combinations exceeding capacity constraint (Total weight $> 14$ ): |              |       |   |
|--------------------------------------------------------------------|--------------|-------|---|
| x bits (MSB-LSB)                                                   | Total weight | Shots |   |
| 7                                                                  | 11001        | 16    | 2 |
| 8                                                                  | 10111        | 18    | 2 |
| 9                                                                  | 11011        | 19    | 3 |
| 11                                                                 | 11101        | 20    | 2 |
| 3                                                                  | 11111        | 23    | 4 |

**Fig. 8.** Detailed post-measurement distribution for a 5-item instance with capacity  $C = 14$ . The optimal feasible assignment attains the largest amplitude.

**Feasibility filtering.** Finally, Figure 9 groups the same outcomes by feasibility: the upper table lists selections with  $\sum_i w_i x_i \leq C$ , while the lower table lists  $\sum_i w_i x_i > C$ . This view confirms that Grover amplification preferentially boosts the feasible group.

| x        | Weight Sum | Value Sum | Weight OK? (Wt ≤ 14) | Value OK? (Val ≤ 0) | Totally Valid? | Shots |
|----------|------------|-----------|----------------------|---------------------|----------------|-------|
| 0 00001  | 2          | 3         | 1                    | 1                   | 1              | 7     |
| 1 00011  | 5          | 7         | 1                    | 1                   | 1              | 5     |
| 2 00101  | 6          | 8         | 1                    | 1                   | 1              | 4     |
| 3 01001  | 7          | 9         | 1                    | 1                   | 1              | 4     |
| 4 00111  | 9          | 12        | 1                    | 1                   | 1              | 3     |
| 5 01011  | 10         | 13        | 1                    | 1                   | 1              | 5     |
| 6 10001  | 11         | 13        | 1                    | 1                   | 1              | 3     |
| 7 01101  | 11         | 14        | 1                    | 1                   | 1              | 5     |
| 8 01111  | 14         | 16        | 1                    | 1                   | 1              | 4     |
| 9 10011  | 14         | 17        | 1                    | 1                   | 1              | 3     |
| 10 10101 | 15         | 18        | 0                    | 1                   | 0              | 1     |
| 11 11001 | 16         | 19        | 0                    | 1                   | 0              | 1     |
| 12 10111 | 18         | 22        | 0                    | 1                   | 0              | 4     |
| 13 11011 | 19         | 23        | 0                    | 1                   | 0              | 3     |
| 14 11101 | 20         | 24        | 0                    | 1                   | 0              | 6     |
| 15 11111 | 23         | 26        | 0                    | 1                   | 0              | 2     |

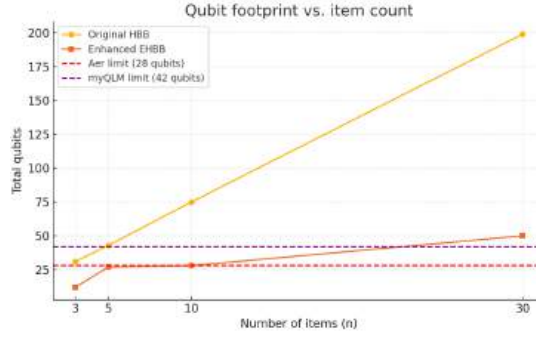
Feasible Solutions (Weight ≤ 14 AND Value > 0):

| x       | Weight Sum | Value Sum | Shots |
|---------|------------|-----------|-------|
| 8 01111 | 14         | 16        | 4     |
| 9 10011 | 14         | 17        | 3     |
| 7 01101 | 11         | 14        | 5     |
| 5 01011 | 10         | 13        | 5     |
| 6 10001 | 11         | 13        | 3     |
| 4 00111 | 9          | 12        | 3     |
| 3 01001 | 7          | 9         | 4     |
| 2 00101 | 6          | 8         | 4     |
| 1 00011 | 5          | 7         | 5     |
| 0 00001 | 2          | 3         | 7     |

**Fig. 9.** Feasibility filtering of measured assignments: **top** – capacity satisfied ( $\sum_i w_i x_i \leq C$ ); **bottom** – capacity violated ( $\sum_i w_i x_i > C$ ).

### Resource usage

We also measured the qubit footprint as a function of the number of items. Figure 10 compares the baseline HBB construction with the enhanced EHBB variant. EHBB reduces qubits by reusing ancillas and uncomputing work registers, achieving a reduction greater than 50% without materially increasing depth—important for NISQ viability.



**Fig. 10.** Qubit footprint vs. item count: baseline HBB vs. EHBB. The dashed lines indicate common simulator limits; EHBB remains within these bounds for larger  $n$ .

### Classical baseline discussion

To contextualise our approach, it is useful to recall how classical branch-and-bound (B&B) solvers scale on the knapsack problem. For  $n$  items, the worst-case complexity remains exponential  $\mathcal{O}(2^n)$ , though practical solvers employ pruning and bounding to reduce average runtime. Dynamic programming variants scale as  $\mathcal{O}(nC)$ , where  $C$  is the capacity, but this becomes prohibitive for large capacities or weights with many bits of precision.

Table 2 contrasts the resource demands of a classical B&B with our quantum HBB and EHBB. The classical solver requires no qubits, but memory/runtime scale quickly with  $n$  and  $C$ . By contrast, the quantum frameworks keep runtime  $\mathcal{O}(2^{n/2})$  oracle calls (due to Grover’s quadratic speedup), at the cost of qubits and circuit depth. The EHBB variant reduces qubits enough to make this trade-off more realistic for NISQ devices.

**Table 2.** Classical B&B vs. Quantum HBB/EHBB: asymptotic scaling and resource demands.

|                 | Classical B&B                 | Quantum HBB                         | Quantum EHBB                        |
|-----------------|-------------------------------|-------------------------------------|-------------------------------------|
| Time complexity | $\mathcal{O}(2^n)$ worst-case | $\mathcal{O}(2^{n/2})$ Grover calls | $\mathcal{O}(2^{n/2})$ Grover calls |
| Memory          | $\text{Poly}(n, C)$           | Qubits $Q_{\text{HBB}}(n)$          | Qubits $Q_{\text{EHBB}}(n)$         |
| Qubits          | 0                             | $\approx 2nb + 3s$                  | $\approx n + b + 3s$                |
| Depth           | —                             | Moderate-High                       | Lower (ancilla reuse)               |
| Strength        | Mature, exact                 | Quantum speedup potential           | NISQ-feasible footprint             |

This comparison underlines that while classical solvers remain superior in absolute runtime for small instances, the quantum design illustrates a path towards polynomial qubit-efficient formulations where quadratic speedup may emerge at scale.

## 7 Resource Analysis and Comparison

*Qubit formulas.* To make the improvement transparent, we report closed-form counts for the original HBB layout versus our enhanced EHBB design. Let  $n$  be the number of items,  $b$  the bit-width used to encode a single item weight/value, and  $s$  the accumulator/comparator width. The original HBB uses multiple dedicated load/mask registers and ripple comparators, yielding

$$Q_{\text{HBB}}(n) = n(1 + 2b) + 3s + 5, \quad (3)$$

whereas EHBB reuses a single load register, adopts Draper adders, and replaces ripple comparators with a compact QFT comparator:

$$Q_{\text{EHBB}}(n) = n + b + 3s + 4. \quad (4)$$

These formulas explain the slower qubit growth of EHBB with  $n$ .

*Register-by-register headcount for  $n = 5$ .* Table 3 details the qubit budget per function block for a 5-item instance. EHBB cuts the total from 58 to 27 qubits by (i) reusing the single load/temporary register across items, (ii) sharing one carry ancilla, and (iii) using a single  $s$ -qubit QFT comparator instead of ripple chains.

**Table 3.** Register-wise qubit count for 5-item knapsack: original HBB vs. enhanced EHBB.

| 2*Register / function          | Original HBB |           | Enhanced EHBB   |           |
|--------------------------------|--------------|-----------|-----------------|-----------|
|                                | Formula      | Qubits    | Formula         | Qubits    |
| Selection bits $x$             | $n$          | 5         | $n$             | 5         |
| Weight <i>load</i> register    | $nb$         | 20        | reuses $Wt\_Vt$ | 6         |
| Value <i>load</i> register     | $nb$         | 20        | reuses $Wt\_Vt$ | 0         |
| AND mask (weights)             | $nb$         | 20        | reuses $Wt\_Vt$ | 0         |
| AND mask (values)              | $nb$         | 20        | reuses $Wt\_Vt$ | 0         |
| Weight sum register $sum_w$    | $s$          | 6         | $s$             | 6         |
| Value sum register $sum_v$     | $s$          | 6         | $s$             | 6         |
| Carry ancilla                  | 2            | 2         | shared carry    | 1         |
| Comparator ancillae            | $s + s$      | 12        | QFT cmp ( $s$ ) | 6         |
| Flags ( $f_w, f_v, f_\Delta$ ) | 3            | 3         | 3               | 3         |
| <b>Total</b>                   | —            | <b>58</b> | —               | <b>27</b> |

*Depth and Grover scheduling.* Depth improves for two reasons: (i) ripple comparators are replaced by a single QFT subtraction-based comparator of width  $s$ , and (ii) the search tree is split into two Grover runs by fixing  $x_1 \in \{0, 1\}$ , which shortens each oracle and enables bound propagation between runs. In our simulations we used a short Grover schedule (one iteration per node) to keep depth compatible with NISQ limits while still amplifying marked states; more iterations can be scheduled following standard amplitude-amplification guidance.

*Scaling picture.* Figure 10 already visualises how  $Q_{\text{EHBB}}$  stays under typical simulator limits far beyond the point where  $Q_{\text{HBB}}$  becomes unsimulable, consistent with the register accounting above.

*Takeaway.* EHBB reduces the qubit footprint by more than 50% for  $n=5$  ( $58 \rightarrow 27$ ) with comparable depth, which is the dominant enabler for running hybrid B&B oracles on today’s simulators and early devices.

## 8 Conclusion

In this work we revisited the 0/1 knapsack problem through the lens of quantum constraint programming and proposed a modular *Hybrid Branch-and-Bound* (HBB) framework. Our design integrates Grover’s search with reversible arithmetic components, including Draper QFT adders and QFT-based comparators, to implement feasibility and optimisation checks in superposition.

We further introduced an enhanced variant (EHBB) that reduces qubit usage by over 50% through ancilla reuse and uncomputation. Simulation results up to ten items demonstrated that the modular oracles behave correctly, that Grover’s amplification concentrates probability mass on feasible assignments, and that the qubit savings of EHBB are substantial for NISQ settings. A comparative analysis confirmed that the new framework achieves the same logical behaviour with far fewer qubits and comparable depth.

**Limitations.** Our current experiments were performed under noiseless simulation. Practical deployment on near-term devices must contend with gate errors, decoherence, and limited connectivity. Moreover, the number of Grover iterations was kept deliberately low to avoid excessive depth, and no state-of-the-art classical baselines were included in runtime comparisons.

**Future work.** Several directions emerge:

1. Extending the framework to other NP-hard combinatorial optimisation problems (e.g. scheduling, graph colouring).
2. Integrating error-mitigation strategies and topology-aware mapping to study NISQ feasibility more realistically.
3. Providing open-source circuit templates for the arithmetic blocks to enhance reproducibility and adoption.
4. Exploring hybrid quantum-classical strategies that combine Grover search with classical heuristics for larger problem instances.

In summary, the enhanced hybrid B&B approach provides a concrete step towards scalable quantum formulations of discrete optimisation problems, narrowing the gap between theoretical quantum speedups and practical implementations on near-term hardware.

## References

1. Ash-Saki, A., Salehi, M., Schuster, D.I., Chong, F.T.: Qft-based comparison circuits with low depth and qubit count (2023), arXiv preprint
2. Boyer, M., Brassard, G., Hoyer, P., Tapp, A.: Tight bounds on quantum searching. *Fortschritte der Physik* **46**(4-5), 493–505 (1998)
3. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. In: *Contemporary Mathematics*, vol. 305, pp. 53–74. American Mathematical Society (2002)
4. Chakrabarti, S., Childs, A.M., Li, Y., Wu, C., Zhang, S.: Quantum algorithms and lower bounds for branch-and-bound. In: *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. pp. 224–252 (2022)
5. Draper, T.G.: Addition on a quantum computer (2000), arXiv preprint
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*. pp. 212–219 (1996)
7. Kellerer, H., Pferschy, U., Pisinger, D.: *Knapsack Problems*. Springer Series in Operations Research, Springer (2004)
8. Montanaro, A.: Quantum walk speedup of backtracking algorithms. *Theory of Computing* **12**(1), 1–24 (2016)
9. Preskill, J.: Quantum computing in the nisq era and beyond. *Quantum* **2**, 79 (2018)
10. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*. pp. 124–134 (1994)



## Response to Reviewers

We would like to thank the reviewers for their constructive feedback. We have carefully revised the manuscript in response to the comments. Below we address each point in detail, indicating the changes made and our rationale.

### Reviewer 1

1. **Limited validation / noise and hardware constraints.** Added explicit discussion of gate errors, decoherence, and connectivity in the *Limitations* paragraph of the Conclusion.
2. **No classical baseline.** Added a new subsection *Classical baseline discussion* in Section 6, with a comparative table contrasting classical B&B, quantum HBB, and EHBB.
3. **Complexity transparency.** Section 7 now includes explicit qubit-count formulas and explains ancilla reuse.
4. **Reproducibility / code availability.** Full source code is available at: <https://github.com/monsefhamitouche/quantum-knapsack-notebooks>.

### Reviewer 2

1. **Target hardware clarity.** Clarified noiseless simulation vs. NISQ assumptions in Introduction/Conclusion.
2. **Conciseness of presentation.** Sections 4–5 rewritten with clearer explanations and integrated figures.
3. **Draper adder details.** Added Table 1 comparing ripple vs. Draper adders.
4. **Circuit reduction (adder + comparator).** Added a note in Section 4 that fusion is non-trivial, left as future work.
5. **Unclear code structure in Section 5.4.** Algorithm 1 rewritten, loop (lines 6–12) now explicit.
6. **Update strategy for  $v_{\min}$ .** Clarified immediate update upon improvement, with early stopping.
7. **B&B structure.** Explicitly shown binary branching with Fig. 5; clarified naming.
8. **Redundancy.** Streamlined Section 5 to avoid repetition.
9. **Experimental scale/runtime outlook.** Added a *Scalability outlook* discussion in Section 6.
10. **Grover iterations.** Specified  $R_{\max} \in [2, 6]$  with early stopping; linked to amplitude amplification formula.
11. **Minor issues.** Fixed headings, removed duplicate equations, corrected notation.

*Closing remark.* We believe these revisions significantly strengthen the manuscript, addressing all major reviewer concerns.

# Quantum Annealing for Computing Bandwidth: QUBO Formulations and Solution Strategies

Qinyu Guo<sup>\*</sup> and Michael J. Dinneen<sup>[0000–0001–9977–525X]</sup>

School of Computer Science, University of Auckland, Auckland, New Zealand  
`qguo731@aucklanduni.ac.nz` and `mjd@cs.auckland.ac.nz`

**Abstract.** The Linear Bandwidth Problem (LBP) is a classical NP-hard graph layout problem with significant applications in sparse matrix reordering and VLSI design. Classical metaheuristic approaches often suffer from scalability issues on large-scale graphs, motivating the exploration of alternative methods such as quantum annealing (QA). Leveraging recent successes of QUBO-based quantum annealing formulations for the cyclic bandwidth problem, this paper presents the first systematic study of QUBO formulations tailored explicitly to the linear bandwidth variant. We propose three novel modeling strategies: (1) a decision-based QUBO with external binary search, (2) an optimization-based QUBO utilizing auxiliary and slack variables for direct objective minimization, and (3) an exponential-penalty QUBO directly encoding bandwidth penalties. We analyze and compare these models regarding variable complexity, matrix sparsity, and compatibility with quantum hardware. Through illustrative examples on small graph instances, we validate the theoretical properties of these formulations and derive practical guidelines for selecting appropriate modeling strategies based on graph characteristics and solver capabilities. Our results not only lay the groundwork for extending quantum combinatorial optimization to LBP but also provide valuable insights for effectively utilizing quantum annealing resources.

**Keywords:** Quantum annealing · QUBO · linear bandwidth.

## 1 Introduction

The *linear bandwidth problem* (LBP) is a classic graph layout problem that aims to assign distinct integer labels to the vertices of a graph such that the largest label difference between adjacent vertices is minimized. This problem arises in various application domains, including sparse matrix reordering [1] and VLSI circuit layout [2].

In recent years, the LBP has received increasing attention due to its practical importance in computational optimization tasks. For instance, minimizing bandwidth directly improves memory access patterns in scientific simulations [1]. Its generalizations, such as the two-dimensional bandwidth minimization problem, have also been explored using both exact and heuristic approaches [3], further

---

<sup>\*</sup> Contact author.

demonstrating the relevance of bandwidth-related models in real-world layout optimization scenarios. However, solving LBP remains notoriously difficult, as it is proven to be NP-hard [4], and even state-of-the-art classical solvers struggle with scalability on large graphs. This highlights the ongoing need for efficient and scalable methods.

Classical algorithms for LBP range from exact solvers to metaheuristics such as simulated annealing and evolutionary algorithms [5]. These methods often involve expensive combinatorial searches and lack guarantees on solution quality or time. As problem sizes increase, their computational inefficiency becomes a critical bottleneck.

Quantum optimization, particularly quantum annealing (QA), has emerged as a promising alternative for solving hard combinatorial problems. Specialized hardware known as Ising machines—including quantum annealers and optical processors—has been explored for efficiently solving Quadratic Unconstrained Binary Optimization (QUBO) and Ising formulations at scale [6]. QA encodes problems into Ising or QUBO models, leveraging quantum tunneling to escape local optima [7, 8]. Among these, recent work Heidari et al. [9] explored quantum annealing for minimization problems in computer vision using QUBO formulations; similarly, Codognot and Monfroy introduced a QUBO-based formulation for the *cyclic bandwidth problem* (CBP) [10], demonstrating the potential of quantum annealing in graph layout optimization. Their method models CBP using a decision-based QUBO formulation, where feasibility is encoded as QUBO constraints and optimization is performed externally via classical binary search.

However, such advances have not yet extended to the linear variant of the problem. Compared to CBP, LBP poses unique challenges due to its unbounded linear structure, which demands different modeling strategies and penalty encodings. This motivates our exploration of tailored QUBO formulations for LBP to bridge this gap and evaluate the suitability of quantum optimization techniques.

To this end, this paper proposes three QUBO-based formulations for the linear bandwidth problem: (1) a decision-based model with repeated feasibility checks, (2) an optimization-based model using auxiliary and slack variables for direct objective minimization, and (3) a compact exponential-penalty model that encodes bandwidth directly with scaled penalties. Each model is analyzed in terms of its variable complexity, matrix sparsity, and penalty structure. We further present guidelines for strategy selection under different problem and hardware conditions.

## 2 Quantum annealing

QA is a metaheuristic optimization technique that exploits quantum mechanical effects to find the global minimum of a given objective function [11]. It operates by encoding the problem into a physical system’s Hamiltonian, where the lowest-energy state represents the optimal solution. The process begins with the system in a simple initial Hamiltonian, whose ground state is straightforward to prepare.

Through a gradual adiabatic evolution, the system transitions to the problem Hamiltonian, ideally remaining in the ground state throughout [12].

In QA, problems are typically formulated as QUBO models. A QUBO problem minimizes an objective function expressed as a quadratic polynomial over binary variables  $x_i \in \{0, 1\}$ :

$$\min_x x^T Q x = \sum_i Q_{ii} x_i + \sum_{i < j} Q_{ij} x_i x_j \quad (1)$$

Here,  $Q$  is an upper-triangular matrix of coefficients, with diagonal terms representing linear contributions and off-diagonal terms capturing quadratic interactions. Constraints are incorporated as penalty terms added to this objective, ensuring that violations increase the energy and guide the system toward feasible solutions. This formulation maps naturally to the Ising model used in QA hardware, enabling efficient exploration of complex search spaces via quantum tunneling [13].

### 3 The Linear Bandwidth Problem

#### 3.1 Problem Definition

The Linear Bandwidth Problem (LBP) is a graph labeling problem defined as follows.

Given an undirected graph  $G = (V, E)$  with  $|V| = n$  vertices, the goal is to find a bijective labeling  $\phi : V \rightarrow \{1, 2, \dots, n\}$  such that the linear bandwidth of  $G$  is minimized:

$$B(G, \phi) = \max_{(u,v) \in E} |\phi(u) - \phi(v)| \quad (2)$$

We assume edges are unordered pairs  $\{u, v\}$ , and for convenience, when referring to  $(u, v) \in E$ , we adopt the convention that  $u < v$  to avoid double-counting (since  $(u, v) \equiv (v, u)$  in undirected graphs).

This problem is known to be NP-hard and has applications in sparse matrix reordering, data layout optimization, and VLSI design.

#### 3.2 Variable Encoding and Common Constraints

To model LBP as a QUBO problem, we adopt a constraint-based formulation. Each solution is encoded as a binary vector, and constraints are embedded as quadratic penalty terms [10].

Let  $x_{v,i} \in \{0, 1\}$  be a binary variable defined as:

$$x_{v,i} = \begin{cases} 1 & \text{if vertex } v \text{ is assigned label } i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

All models share a common permutation constraint to ensure that the labeling is a valid permutation of  $\{1, 2, \dots, n\}$ , requiring:

(i) Each vertex is assigned exactly one label:

$$\sum_{i=1}^n x_{v,i} = 1 \quad \forall v \in V \quad (4)$$

(ii) Each label is assigned to exactly one vertex:

$$\sum_{v \in V} x_{v,i} = 1 \quad \forall i \in \{1, 2, \dots, n\} \quad (5)$$

Following standard QUBO modeling principles [13], we define binary variables  $x_{v,i} \in \{0, 1\}$  indicating whether vertex  $v$  is assigned label  $i$ . To ensure that the labeling forms a valid permutation, we incorporate the one-hot permutation constraints as quadratic penalties:

$$P_{\text{perm}} = \sum_{v \in V} \left( \sum_{i=1}^n x_{v,i} - 1 \right)^2 + \sum_{i=1}^n \left( \sum_{v \in V} x_{v,i} - 1 \right)^2 \quad (6)$$

## 4 Methodology: QUBO Modeling Approaches

### 4.1 Method 1: Decision-based QUBO Model

Our first method is for the decision problem with an additional integer input  $k < n$ .

**Model Description** The complete QUBO objective function is:

$$\min_{x \in \{0,1\}^{n^2}} x^\top Q x = P_{\text{perm}} + \lambda \cdot P_k \quad (7)$$

where  $P_{\text{perm}}$  encodes the permutation constraint,  $P_k$  encodes the linear bandwidth constraint for bandwidth at most  $k$ , and  $\lambda$  is a penalty weighting coefficient.

**Constraint and Penalty Terms** We convert the optimization problem into a satisfiability form  $\text{sat}(G, k)$  that verifies whether all adjacent vertices can be labeled within a maximum difference  $k$ :

$$|\phi(u) - \phi(v)| \leq k \quad \forall (u, v) \in E \quad (8)$$

Let  $L_k$  be the set of allowed label pairs:

$$L_k = \{(i, j) \in [1, n]^2 \mid i \neq j \text{ and } |i - j| \leq k\} \quad (9)$$

The table constraint penalty is then:

$$P_k = \sum_{(u,v) \in E} \sum_{(i,j) \notin L_k} x_{u,i} \cdot x_{v,j} \quad (10)$$

This penalty is minimized when all adjacent vertices are assigned label pairs within the permitted linear distance of at most  $k$ .

## 4.2 Method 2: Optimization-based QUBO Model

**Model Description and Variables** To model LBP as a Quadratic Unconstrained Binary Optimization (QUBO) problem without a fixed  $k$ , we introduce the bandwidth  $K$  as an auxiliary variable and use slack variables to directly minimize  $K$  while enforcing constraints. (See [14, 15] for some other QUBO formulations using slack variables.)

Let  $p = \lceil \log_2 n \rceil$ . Introduce binary variables  $k_t \in \{0, 1\}$  ( $t = 0, 1, \dots, p-1$ ) to encode  $K$ :

$$K = \sum_{t=0}^{p-1} 2^t k_t \quad (11)$$

For each edge  $e = (u, v) \in E$ , introduce a slack variable  $s_e$  (non-negative integer) encoded with  $p = \lceil \log_2 n \rceil$  binary variables  $s_{e,r} \in \{0, 1\}$  ( $r = 0, 1, \dots, p-1$ ):

$$s_e = \sum_{r=0}^{p-1} 2^r s_{e,r} \quad (12)$$

The total number of variables is  $n^2 + p + |E| \cdot p$ . The complete QUBO objective function is:

$$\min_{x \in \{0,1\}^{n^2 + p + |E| \cdot p}} x^\top Q x = P_{\text{perm}} + A \cdot P_{\text{bw}} + \lambda \cdot P_{\text{minK}} \quad (13)$$

where  $P_{\text{perm}}$  encodes the permutation constraint,  $P_{\text{bw}}$  enforces the bandwidth constraints using slack variables,  $P_{\text{minK}}$  minimizes  $K$ , and  $A, \lambda > 0$  are penalty weighting coefficients (with  $A \gg \lambda$  to prioritize constraint satisfaction).

**Constraint and Penalty Terms** To enforce  $|\phi(u) - \phi(v)| \leq K$  for all  $(u, v) \in E$  without a fixed  $k$ , we introduce slack variables  $s_e$  for each edge  $e = (u, v)$  to convert the inequality constraints into quadratic penalties. This allows the QUBO to directly minimize  $K$  while ensuring feasibility.

The effective distance for edge  $(u, v)$  is:

$$d_e = \sum_{i=1}^n \sum_{j=1}^n |i - j| \cdot x_{u,i} x_{v,j} \quad (14)$$

The constraint  $d_e \leq K$  is reformulated using the non-negative slack  $s_e$  as an equality  $d_e - K + s_e = 0$ , which is enforced via a squared penalty term. The bandwidth penalty is:

$$P_{\text{bw}} = \sum_{(u,v) \in E} (d_e - K + s_e)^2 \quad (15)$$

This penalty is zero if and only if  $d_e \leq K$  for all edges, with  $s_e = K - d_e$  when  $d_e < K$  (absorbing the slack) or  $s_e = 0$  when  $d_e = K$ . If  $d_e > K$ , the penalty is positive, as  $s_e \geq 0$  cannot fully compensate.

Expanding the square yields quadratic terms involving  $x_{u,i}x_{v,j}$ ,  $x_{u,i}k_t$ ,  $x_{u,i}s_{e,r}$ ,  $k_t s_{e,r}$ , and self-terms, which can be incorporated into the QUBO matrix  $Q$ . The matrix  $Q_{\text{bw}}$  aggregates these expansions across all edges, resulting in a block-structured form as illustrated in Figure 1. The blocks correspond to interactions between variable groups: x-x (from  $d_e^2$ ), x-k (from  $-2d_e K$ ), x-s (from  $2d_e s_e$ ), k-k (from  $K^2$ ), k-s (from  $-2K s_e$ ), and s-s (from  $s_e^2$ ). This structure is upper-triangular to align with the QUBO formulation, ensuring efficient computation while maintaining equivalence to the symmetric contributions from the expansions.

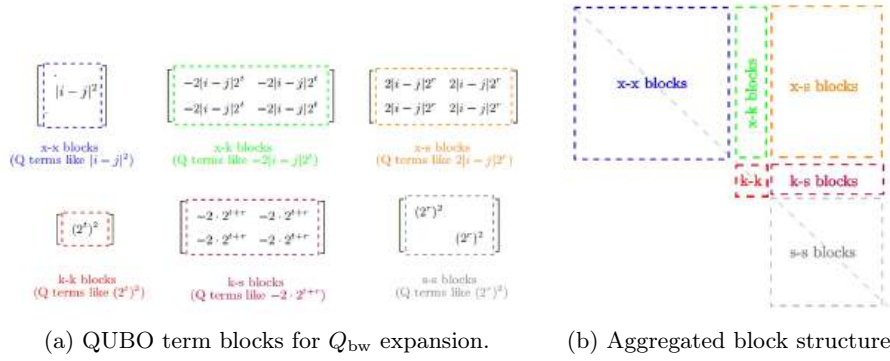


Fig. 1: Schematic illustration of (a) QUBO expansion term blocks and (b) the aggregated  $(n^2 + p + |E| \cdot p) \times (n^2 + p + |E| \cdot p)$  QUBO matrix structure for  $Q_{\text{bw}}$  in method 2.

This formulation ensures the model directly optimizes the minimum bandwidth without external loops over  $k$ . To minimize the bandwidth  $K$ :

$$P_{\text{minK}} = \sum_{t=0}^{m-1} 2^t k_t \quad (16)$$

This linear term (embedded in the QUBO quadratic form via diagonal elements) encourages smaller values of  $K$ .

### 4.3 Method 3: Exponential-penalty QUBO Model

**Model Description** This model modifies the original decision-based QUBO formulation by replacing the fixed- $k$  table constraint  $k$  with an exponential penalty term that encodes the bandwidth directly into the objective without requiring an external search over  $k$ . It uses the same binary variables as the original model, requiring  $n^2$  qubits, but introduces larger constants to distinguish bandwidth values.

The complete QUBO objective function is:

$$\min_{x \in \{0,1\}^{n^2}} x^\top Q x = P_{\text{perm}} + \lambda \cdot P_{\text{exp}} \quad (17)$$

where  $P_{\text{perm}}$  encodes the permutation constraint,  $P_{\text{exp}}$  encodes the exponential penalties for bandwidth violations, and  $\lambda > 0$  is a penalty weighting coefficient.

By minimizing this QUBO, the minimum value achieved (under valid permutations) encodes the minimum bandwidth, which can be decoded from the objective value as explained in the proof of correctness below.

**Exponential Penalty Term** Instead of a fixed  $k$ , we assign exponentially increasing penalties based on the label distance  $|i - j|$  for each edge  $(u, v) \in E$ . The penalty term is:

$$P_{\text{exp}} = \sum_{(u,v) \in E} \sum_{i=1}^n \sum_{j=1}^n 2^{n|i-j|} \cdot x_{u,i} \cdot x_{v,j} \quad (18)$$

Note that when  $i = j$ ,  $|i - j| = 0$ , so  $2^{n \cdot 0} = 1$ , but since the labeling is bijective (enforced by  $P_{\text{perm}}$ ), no two vertices share the same label, and self-loops are not considered in  $E$ . This term is quadratic and can be directly incorporated into the QUBO matrix  $Q$ . The exponential base ensures that larger distances dominate the sum, allowing the minimum bandwidth to be extracted from the total objective value.

**Proof of Correctness** The correctness of this model relies on the property that the minimum value of the QUBO objective (assuming  $\lambda = 1$  for simplicity, and focusing on  $P_{\text{exp}}$  under valid permutations where  $P_{\text{perm}} = 0$ ) uniquely encodes the minimum bandwidth  $B(G) = k_{\min}$ . Specifically, for any valid labeling  $\phi$ , let  $d_{\max} = \max_{(u,v) \in E} |\phi(u) - \phi(v)|$ . Then the value of  $P_{\text{exp}}$  satisfies the bounds that separate different values of  $d_{\max}$ .

Assume that the graph has  $m = |E|$  edges. For a fixed label with  $d_{\max} = k$ : Each edge contributes at most  $2^{nk}$  to the sum (for distances exactly  $k$ ), and there are at most  $m$  edges, so  $P_{\text{exp}} \leq m \cdot 2^{nk}$ . Since  $m \leq \binom{n}{2} < 2^{n-1}$  for  $n \geq 2$  (and strictly  $m \leq \binom{n}{2} \approx n^2/2 < 2^n$  for sufficiently large  $n$ , but the bound holds as  $\binom{n}{2} < 2^n$  for all  $n \geq 1$ ), we have  $P_{\text{exp}} < 2^n \cdot 2^{nk} = 2^{n(k+1)}$ .

If there exists at least one edge with distance  $k + 1$ , that single edge contributes exactly  $2^{n(k+1)}$  (or more if larger distances exist, but at minimum this), and all other edges contribute non-negative terms, so  $P_{\text{exp}} \geq 2^{n(k+1)}$ .



Combining these: If  $d_{\max} \leq k$ , then  $P_{\exp} < 2^{n(k+1)}$ . If  $d_{\max} \geq k+1$ , then  $P_{\exp} \geq 2^{n(k+1)}$ .

Thus, the ranges for different  $k$  are disjoint: the minimal  $P_{\exp}$  over all valid labelings will fall into the range corresponding to  $k_{\min} = B(G)$ , specifically  $1 \leq P_{\exp\min} < 2^{n(k_{\min}+1)}$  (since at least one edge exists in non-trivial graphs, but adjusts for  $m = 0$  trivially). To extract  $k_{\min}$ , compute  $\lfloor \log_2(P_{\exp\min} + \epsilon)/n \rfloor$  (with small  $\epsilon > 0$  to handle bounds), but more precisely, find the smallest  $k$  such that  $P_{\exp\min} < 2^{n(k+1)}$ .

This ensures that the global minimum of the QUBO corresponds to a labeling achieving the minimum bandwidth, and the objective value uniquely identifies  $k_{\min}$  without external iteration over  $k$ . Note that the constants  $2^{n|i-j|}$  grow rapidly, which may pose numerical challenges in classical solvers but is suitable for quantum annealers handling large coefficients via embedding.

**Improved Formulation with Adjusted Base** Although the original exponential penalty with base  $2^n$  effectively distinguishes bandwidth values, it can lead to numerical overflow for larger  $n$  due to excessively large coefficients. To address this, we suggest modifying the model to use an adjusted base  $b = \lfloor \binom{n}{2} \rfloor + 1$ , which is an integer greater than the maximum possible number of edges  $m \leq \binom{n}{2}$ .

The permutation constraint  $P_{\text{perm}}$  remains unchanged. The exponential penalty term is updated as follows:

$$P_{\exp} = \sum_{(u,v) \in E} \sum_{i=1}^n \sum_{j=1}^n b^{|i-j|} \cdot x_{u,i} \cdot x_{v,j} \quad (19)$$

This preserves the quadratic structure and the ability to decode the minimum bandwidth from the objective value, as the proof of correctness holds analogously (with  $2^n$  replaced by  $b > m$ ).

**Reasons for the Modification** The primary reason for this change is to mitigate numerical instability and overflow issues inherent in the original formulation. In the original model, coefficients reach up to  $2^{n(n-1)}$ , which grows as  $2^{O(n^2)}$ . This exponential explosion makes the model impractical for  $n > 20$  on standard computing hardware, as coefficients exceed the representable range of floating-point numbers (e.g., IEEE 754 double precision limits at approximately  $1.8 \times 10^{308}$ ). For instance, at  $n = 50$ ,  $2^{n(n-1)} \approx 2^{2450} \approx 10^{738}$ , which overflows to infinity in most systems.

By choosing  $b \approx n^2/2$ , the model maintains the separation of value ranges for different maximum distances  $k$  (as  $b > m$  ensures disjoint intervals:  $P_{\exp} < b^{k+1}$  if  $d_{\max} \leq k$ , and  $P_{\exp} \geq b^{k+1}$  if  $d_{\max} \geq k+1$ ), while significantly reducing coefficient magnitudes. This adjustment is simple to implement, requiring only a precomputation of  $b$ , and does not increase the number of variables.

**Analysis of Reduced Overflow Risk** The modified base substantially lowers the overflow risk by reducing the growth rate of the maximum coefficient from  $O(2^{n^2})$  to approximately  $O((n^2/2)^{n-1}) = 2^{O(n \log n)}$ . Specifically:

Original maximum coefficient:

$$2^{n(n-1)} = 2^{n^2-n} \quad (20)$$

Modified maximum coefficient:

$$b^{n-1} \approx (n^2/2)^{n-1} = 2^{(n-1) \log_2(n^2/2)} \approx 2^{(n-1)(2 \log_2 n - 1)} \quad (21)$$

For concrete examples: (1) For  $n = 10$ : Original  $\approx 2^{90} \approx 1.2 \times 10^{27}$ ; Modified  $\approx (50)^9 \approx 2^{50.8} \approx 1.7 \times 10^{15}$  (reduction by factor of  $\approx 10^{12}$ ). (2) For  $n = 50$ : Original  $\approx 2^{2450} \approx 10^{738}$ ; Modified  $\approx (1250)^{49} \approx 2^{504} \approx 10^{152}$  (reduction by factor of  $\approx 10^{586}$ ). (3) For  $n = 100$ : Original  $\approx 2^{9900} \approx 10^{2979}$ ; Modified  $\approx (5000)^{99} \approx 2^{(99) \times (\log_2 5000)} \approx 2^{99 \times 12.3} \approx 2^{1218} \approx 10^{367}$  (still large but representable with arbitrary-precision arithmetic, unlike the original).

This slower growth ( $O(n \log n)$  in the exponent vs.  $O(n^2)$ ) ensures the coefficients remain manageable for larger  $n$  (e.g., up to  $n = 100$  with big-integer libraries like Python’s `int`), while avoiding precision loss in quantum annealers or classical solvers. However, for extremely large  $n$ , further optimizations such as layered penalties or normalization may be necessary.

## 5 Practical Usability

### 5.1 Time Complexity Analysis

As summarized in Table 1, the three QUBO modeling approaches exhibit distinct trade-offs in variable count, matrix density, and computational complexity. The decision-based model (Method 1) requires  $O(\log n)$  sequential or binary search iterations over the bandwidth parameter  $k$ , resulting in multiple QUBO solves, but benefits from a relatively sparse matrix structure that is favorable for both classical and quantum solvers. In contrast, the optimization-based model (Method 2) encodes the bandwidth as additional binary variables, allowing the problem to be solved in a single QUBO instance. This approach, however, increases the total number of variables and matrix density, which can present greater challenges for hardware embedding and solution efficiency. The exponential-penalty model (Method 3) similarly requires only a single QUBO solve but generates a fully dense matrix with large coefficients, potentially imposing greater computational and numerical demands on classical solvers, though it remains well-suited for quantum annealing devices capable of handling such dense structures. Overall, the choice of model involves a balance between solution generality, variable count, and practical solvability for a given hardware platform.

Table 1: Time complexity comparison of the three QUBO models.

| Method              | Variable Count              | Matrix Density  | Overall Complexity                |
|---------------------|-----------------------------|-----------------|-----------------------------------|
| Decision-based      | $O(n^2)$                    | Sparse          | $O(T_{\text{QUBO}} \cdot \log n)$ |
| Optimization-based  | $O(n^2 +  E  \cdot \log n)$ | Medium to dense | $O(T'_{\text{QUBO}})$             |
| Exponential-penalty | $O(n^2)$                    | Dense           | $O(T''_{\text{QUBO}})$            |

## 5.2 Illustrative Example: $K_3$ with Pendant

Consider the graph with vertices  $V = \{0, 1, 2, 3\}$  and edges  $E = \{(0, 1), (0, 2), (1, 2), (0, 3)\}$  ( $n = 4$ ). This is a triangle ( $K_3$ ) on vertices  $\{0, 1, 2\}$  with a pendant vertex 3 attached to 0. The optimal bandwidth is 2 (e.g., labeling  $\phi(0) = 2, \phi(1) = 1, \phi(2) = 3, \phi(3) = 4$ ; max difference=2).

We demonstrate the solving process for each method, starting with the common  $P_{\text{perm}}$  and its matrix  $Q_{\text{perm}}$ . Variables are indexed 0-15: rows 0-3 for  $x_{0,1-4}$ , 4-7 for  $x_{1,1-4}$ , 8-11 for  $x_{2,1-4}$ , 12-15 for  $x_{3,1-4}$ .

**Computing  $P_{\text{perm}}$  and  $Q_{\text{perm}}$  Matrix** Substitute  $n = 4, V = \{0, 1, 2, 3\}$  into Eq. 6:

$$P_{\text{perm}} = \sum_{v=0}^3 \left( \sum_{i=1}^4 x_{v,i} - 1 \right)^2 + \sum_{i=1}^4 \left( \sum_{v=0}^3 x_{v,i} - 1 \right)^2$$

Expanding each term: For row  $v = 0$  (variables  $x_{0,1}$  to  $x_{0,4}$ ),

$$\begin{aligned} \left( \sum_{i=1}^4 x_{0,i} - 1 \right)^2 &= \sum_{i=1}^4 x_{0,i}^2 + 2 \sum_{1 \leq i < j \leq 4} x_{0,i} x_{0,j} - 2 \sum_{i=1}^4 x_{0,i} + 1 \\ &= 2 \sum_{1 \leq i < j \leq 4} x_{0,i} x_{0,j} - \sum_{i=1}^4 x_{0,i} + 1 \end{aligned}$$

For column  $i = 1$  (variables  $x_{0,1}$  to  $x_{3,1}$ ), a similar expansion yields  $2 \sum_{0 \leq v < v' \leq 3} x_{v,1} x_{v',1} - \sum_{v=0}^3 x_{v,1} + 1$ .

Summing all (ignoring constants) gives the quadratic coefficient 2 for intra-row/column pairs, linear -2 on diagonals. For example, with assignment  $x_{0,2} = 1, x_{1,1} = 1, x_{2,3} = 1, x_{3,4} = 1$  (valid labeling),  $P_{\text{perm}} = 0$  (each sum=1).

The matrix  $Q_{\text{perm}}$  is derived with quadratic coefficients of 2, according to the standard expansion. For formulations preferring halved coefficients (e.g., to simplify matrix values or align with integer-based solvers, without altering the optimization minima due to proportional scaling), divide all entries by 2, as shown in Figure 2.

**Method 1: Decision-based (for  $k = 2$ )** For  $k = 2$ , substitute into Eq. 9 to compute  $L_2 = \{(i, j) \mid i \neq j, |i - j| \leq 2\}$ , yielding allowed pairs such as (1,2),

$$Q_{\text{perm}} = \begin{matrix} & \begin{matrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{matrix} \\ \begin{matrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{1,4} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{2,4} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \\ x_{3,4} \\ x_{4,1} \\ x_{4,2} \\ x_{4,3} \\ x_{4,4} \end{matrix} & \begin{pmatrix} -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & -1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ & & & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & & -1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ & & & & & & & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ & & & & & & & & & -1 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & & & & & & & & -1 & 1 & 0 & 0 & 1 & 0 \\ & & & & & & & & & & & -1 & 1 & 1 & 1 & 1 \\ & & & & & & & & & & & & -1 & 1 & 1 & 1 \\ & & & & & & & & & & & & & -1 & 1 & 1 \\ & & & & & & & & & & & & & & -1 & 1 \\ & & & & & & & & & & & & & & & -1 \end{pmatrix} \end{pmatrix}$$

Fig. 2: Expanded  $Q_{\text{perm}}$  matrix for  $n = 4$  permutation constraint. Green entries represent row constraints (diagonal -1, off-diagonal 1 within rows); red entries represent column constraints (1 across rows for same label).

(1,3), (2,1), etc., and forbidden pairs (1,4) and (4,1) (since for  $n = 4$ ,  $|i - j| > 2$  only these).

Expanding the table penalty via Eq. 10: For edge (0,1), sum over forbidden (i,j) adds terms  $x_{0,1}x_{1,4} + x_{0,4}x_{1,1}$  (corresponding to positions with coefficient 1 at indices (0,7) and (3,4) in the QUBO matrix, where variables are ordered as  $x_{0,1}$  to  $x_{0,4}$ ,  $x_{1,1}$  to  $x_{1,4}$ , etc.).

For edge (0,2), similarly adds  $x_{0,1}x_{2,4} + x_{0,4}x_{2,1}$  (at (0,11) and (3,8)); for (1,2):  $x_{1,1}x_{2,4} + x_{1,4}x_{2,1}$  (at (4,11) and (7,8)); for (0,3):  $x_{0,1}x_{3,4} + x_{0,4}x_{3,1}$  (at (0,15) and (3,12)).

Summing all gives quadratic coefficients of 1 for each forbidden interaction across edges, resulting in a sparse  $Q_2$  with exactly 8 entries of 1 (2 per edge  $\times$  4 edges). For example, with a valid labeling like  $x_{0,2} = 1$ ,  $x_{1,1} = 1$ ,  $x_{2,3} = 1$ ,  $x_{3,4} = 1$  (no forbidden pairs activated),  $P_2 = 0$ .

The matrix  $Q_2$  is derived with these quadratic coefficients of 1, according to the summation. For formulations preferring adjusted scales (e.g., to align with solver precision), coefficients can be scaled uniformly without altering minima due to proportionality, as shown in Figure 3.

Final  $Q = Q_{\text{perm}} + \lambda Q_{\text{table}}$  ( $\lambda = 10$ , per Eq. 7). Minimization yields objective 0, confirming feasibility; binary search over  $k$  determines the linear bandwidth as 2.

**Method 2: Optimization-based** In this example ( $n = 4$ ,  $|E| = 4$ ), variables are indexed as  $x_{v,i}$  for  $0 \leq v < 4$ ,  $1 \leq i \leq 4$  (0–15),  $k_0, k_1$  (16,17), and slack bits  $s_{e,0}, s_{e,1}$  for each edge  $e$  (18–25).

Key coefficient computations for each block of  $Q_{\text{bw}}$  are illustrated for a representative edge in Figure 4, including x-x from  $d_e^2$  (coefficients  $|i - j|^2$ ), x-k from  $-2d_eK (-2|i - j|2^t)$ , x-s from  $2d_e s_e (2|i - j|2^r)$ , k-k from  $K^2 ((2^t)^2)$ , k-s from  $-2K s_e (-2 \cdot 2^t 2^r)$ , and s-s from  $s_e^2 ((2^r)^2)$ .

$$Q_{\text{table}} = \begin{matrix} & \begin{matrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{matrix} \\ \begin{matrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{1,4} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{2,4} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \\ x_{3,4} \\ x_{4,1} \\ x_{4,2} \\ x_{4,3} \\ x_{4,4} \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & & & 0 & 0 \\ & & & & & & & & & & & & & & & 0 \\ & & & & & & & & & & & & & & & & 0 \end{pmatrix} \end{matrix}$$

Fig. 3: Sparse  $Q_{\text{table}}$  matrix for  $k = 2$ , highlighting penalty coefficients of 1 at positions for forbidden pairs across all edges.

Aggregating over all edges fills the full matrix (e.g.,  $k$ - $k$  scaled by  $|E|$ ). The diagonal for  $k_0, k_1$  incorporates  $P_{\text{minK}}$  from Eq. 16 to favor minimal  $K$ .

Solving the resulting QUBO (e.g., via quantum annealing or classical solvers) yields  $K = 2$  with labeling  $x_{0,2} = 1, x_{1,1} = 1, x_{2,3} = 1, x_{3,4} = 1$  (max  $d_e = 2$ , slacks absorbing differences for minimal energy).

**Method 3: Exponential-penalty** Substitute  $n = 4$  into Eq. 18 to compute  $P_{\text{exp}} = \sum_{(u,v) \in E} \sum_{i=1}^4 \sum_{j=1}^4 2^{4|i-j|} x_{u,i} x_{v,j}$ , yielding penalties like  $2^{4 \cdot 0} = 1$  for  $|i - j| = 0$  (avoided via permutation),  $2^{4 \cdot 1} = 16$  for distance 1,  $2^8 = 256$  for 2,  $2^{12} = 4096$  for 3.

For edge  $(0,1)$ , add  $2^{4|i-j|}$  to positions (e.g.,  $(1,2)$ : 16 at  $(0,5)$ ;  $(1,4)$ : 4096 at  $(0,7)$ ; symmetric for  $(2,1)$ : 16 at  $(1,4)$ ). Similar for other edges, aggregating into dense  $Q_{\text{exp}}$  with 64 non-zero entries (coefficients 16 to 4096), as shown in Figure 5.

For halved coefficients (to align with solvers), divide by 2. Use adjusted base  $b = 7$  (Eq. 19) to reduce max from 4096 to 343.

Final  $Q = Q_{\text{perm}} + \lambda Q_{\text{exp}}$  (Eq. 17). Minimization yields  $P_{\text{exp}} = 1088$  for labeling  $x_{0,2} = 1, x_{1,1} = 1, x_{2,3} = 1, x_{3,4} = 1$  (distances 1,1,2,2; decodes to bandwidth 2, as  $256 \leq 1088 < 4096$  per proof).

## 6 Strategy Selection and Practical Guidelines

Selecting the appropriate QUBO formulation for solving the Linear Bandwidth Problem (LBP) depends significantly on the characteristics of the input graph (size and density) and the constraints posed by computational resources and quantum hardware.

For graphs with small to moderate vertex counts ( $n \lesssim 20$ ), or when computational resources are limited, the **decision-based model** is recommended. Its primary advantage is a sparse matrix and relatively low variable count, enabling

$$\begin{array}{cc}
\begin{array}{c} \text{x-x blocks} \\ \text{(Q terms like } |i-j|^2) \end{array} & \begin{array}{c} \text{x-k blocks} \\ \text{(Q terms like } -2|i-j|2^t) \end{array} \\
\begin{array}{c} \left[ \begin{array}{ccc} |1-2|^2 & |1-3|^2 & |1-4|^2 \\ |2-1|^2 & |2-3|^2 & |2-4|^2 \\ |3-1|^2 & |3-2|^2 & |3-4|^2 \\ |4-1|^2 & |4-2|^2 & |4-3|^2 \end{array} \right] = \left[ \begin{array}{ccc} 1 & 4 & 9 \\ 1 & 1 & 4 \\ 4 & 1 & 1 \\ 9 & 4 & 1 \end{array} \right] \end{array} & \begin{array}{c} \left[ \begin{array}{cc} -2 \cdot 1 \cdot 1 & -2 \cdot 1 \cdot 2 \\ -2 \cdot 2 \cdot 1 & -2 \cdot 2 \cdot 2 \\ -2 \cdot 3 \cdot 1 & -2 \cdot 3 \cdot 2 \\ -2 \cdot 4 \cdot 1 & -2 \cdot 4 \cdot 2 \end{array} \right] = \left[ \begin{array}{cc} -2 & -4 \\ -4 & -8 \\ -6 & -12 \\ -8 & -16 \end{array} \right] \end{array} \\
\begin{array}{c} \text{x-s blocks} \\ \text{(Q terms like } 2|i-j|2^r) \end{array} & \begin{array}{c} \text{k-k blocks} \\ \text{(Q terms like } (2^t)^2) \end{array} \\
\begin{array}{c} \left[ \begin{array}{cc} 2 \cdot 1 \cdot 1 & 2 \cdot 1 \cdot 2 \\ 2 \cdot 2 \cdot 1 & 2 \cdot 2 \cdot 2 \\ 2 \cdot 3 \cdot 1 & 2 \cdot 3 \cdot 2 \\ 2 \cdot 4 \cdot 1 & 2 \cdot 4 \cdot 2 \end{array} \right] = \left[ \begin{array}{cc} 2 & 4 \\ 4 & 8 \\ 6 & 12 \\ 8 & 16 \end{array} \right] \end{array} & \begin{array}{c} \left[ \begin{array}{cc} (1)^2 & (1)(2) \\ (2)(1) & (2)^2 \end{array} \right] = \left[ \begin{array}{cc} 1 & 2 \\ 2 & 4 \end{array} \right] \end{array} \\
\begin{array}{c} \text{k-s blocks} \\ \text{(Q terms like } -2 \cdot 2^t 2^r) \end{array} & \begin{array}{c} \text{s-s blocks} \\ \text{(Q terms like } (2^r)^2) \end{array} \\
\begin{array}{c} \left[ \begin{array}{cc} -2 \cdot 1 \cdot 1 & -2 \cdot 1 \cdot 2 \\ -2 \cdot 2 \cdot 1 & -2 \cdot 2 \cdot 2 \end{array} \right] = \left[ \begin{array}{cc} -2 & -4 \\ -4 & -8 \end{array} \right] \end{array} & \begin{array}{c} \left[ \begin{array}{cc} (1)^2 & (1)(2) \\ (2)(1) & (2)^2 \end{array} \right] = \left[ \begin{array}{cc} 1 & 2 \\ 2 & 4 \end{array} \right] \end{array}
\end{array}$$

Fig. 4: Example coefficient computations for each block in  $Q_{\text{bw}}$  (illustrated for a representative edge).

efficient embedding and quick QUBO solves, despite the iterative search over the bandwidth parameter  $k$ .

The **optimization-based model** should be chosen when reducing the number of QUBO solves is critical and the graph edges ( $|E|$ ) are not extensively dense. This method introduces additional variables (auxiliary bandwidth and slack variables), increasing matrix density. Nevertheless, the direct bandwidth minimization capability, requiring only a single QUBO solve, makes it highly suitable for moderate-density graphs.

The **exponential-penalty model** is optimal for use with quantum annealers or high-capacity classical solvers capable of handling large coefficients and dense matrices. Although it has the lowest variable count and encodes the optimal solution directly, its dense structure and large coefficients pose challenges for classical computational tools.

## 7 Conclusion

We systematically investigated three quantum annealing-based QUBO formulations for solving the linear bandwidth problem (LBP): the decision-based, optimization-based, and exponential-penalty models. Each approach exhibits distinct trade-offs regarding variable count, matrix sparsity, and numerical complexity.

The decision-based model benefits from a sparse matrix and simpler structure but requires multiple iterations due to the external binary search for band-

| $x_{0,1}$ | $x_{0,2}$ | $x_{0,3}$ | $x_{0,4}$ | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $x_{1,4}$ | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $x_{2,4}$ | $x_{3,1}$ | $x_{3,2}$ | $x_{3,3}$ | $x_{3,4}$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0         |           |           |           | 1         | 16        | 256       | 4096      | 1         | 16        | 256       | 4096      | 1         | 16        | 256       | 4096      |
|           | 0         |           |           | 16        | 1         | 16        | 256       | 16        | 1         | 16        | 256       | 16        | 1         | 16        | 256       |
|           |           | 0         |           | 256       | 16        | 1         | 16        | 256       | 16        | 1         | 16        | 256       | 16        | 1         | 16        |
|           |           |           | 0         | 4096      | 256       | 16        | 1         | 4096      | 256       | 16        | 1         | 4096      | 256       | 16        | 1         |
|           |           |           |           | 0         |           |           |           | 1         | 16        | 256       | 4096      | 1         | 16        | 256       | 4096      |
|           |           |           |           |           | 0         |           |           | 16        | 1         | 16        | 256       | 16        | 1         | 16        | 256       |
|           |           |           |           |           |           | 0         |           | 256       | 16        | 1         | 16        | 256       | 16        | 1         | 16        |
|           |           |           |           |           |           |           | 0         | 4096      | 256       | 16        | 1         | 4096      | 256       | 16        | 1         |
|           |           |           |           |           |           |           |           | 0         |           |           |           | 1         | 16        | 256       | 4096      |
|           |           |           |           |           |           |           |           |           | 0         |           |           | 16        | 1         | 16        | 256       |
|           |           |           |           |           |           |           |           |           |           | 0         |           | 256       | 16        | 1         | 16        |
|           |           |           |           |           |           |           |           |           |           |           | 0         | 4096      | 256       | 16        | 1         |
|           |           |           |           |           |           |           |           |           |           |           |           | 0         |           |           |           |
|           |           |           |           |           |           |           |           |           |           |           |           |           | 0         |           |           |
|           |           |           |           |           |           |           |           |           |           |           |           |           |           | 0         |           |
|           |           |           |           |           |           |           |           |           |           |           |           |           |           |           | 0         |

Fig. 5: **Block structure of  $Q_{\text{exp}}$  for Method 3 ( $n = 4$ ,  $|E| = 4$ ).** Each  $4 \times 4$  block corresponds to variable pairs  $(x_{u,i}, x_{v,j})$  for edges  $(u, v)$  and labels  $i, j$ . Entries are  $2^{4|i-j|}$ ; zeros omitted for clarity. All blocks repeat across edges (see main text for edge/label assignment).

width  $k$ . In contrast, the optimization-based model directly minimizes bandwidth within a single solve but introduces additional variables and denser matrices, potentially complicating hardware embedding. The exponential-penalty model provides a compact single-solve formulation, yet suffers from large, exponentially growing coefficients, posing numerical challenges for classical and quantum solvers at larger scales.

Practical experiments and theoretical analysis suggest selecting the QUBO model according to problem size and hardware constraints: sparse or moderate instances favor decision-based or optimization-based methods, whereas dense or larger graphs might benefit from hybrid classical-quantum strategies or coefficient adjustments.

Future research may focus on developing hardware-optimized QUBO encodings, hybrid quantum-classical algorithms, and extensive benchmarking on quantum hardware to address scalability and applicability to real-world linear bandwidth optimization problems.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Buluç, A., Mattson, T., McMillan, S., Moreira, J., Yang, C.: The GraphBLAS C API Specification. Technical Report, GraphBLAS.org (2017)
2. Wuytack, S., Catthoor, F., De Man, H.: Minimizing the required memory bandwidth in VLSI system realizations. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **7**(4), 433–441 (1999). <https://doi.org/10.1109/92.805752>

3. Rodríguez-García, M.A., Sánchez-Oro, J., Rodríguez-Tello, E., Monfroy, E., Duarte, A.: Two-dimensional bandwidth minimization problem: exact and heuristic approaches. *Knowledge-Based Systems* **214**, 106651 (2021). <https://doi.org/10.1016/j.knosys.2020.106651>
4. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York (1979)
5. Chagas, G.O., de Oliveira, S.L.G.: Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: A systematic review. *Procedia Computer Science* **51**, 211–220 (2015). <https://doi.org/10.1016/j.procs.2015.05.217>
6. Mohseni, N., McMahon, P.L., Byrnes, T.: Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics* **4**, 363–379 (2022). <https://doi.org/10.1038/s42254-022-00440-8>
7. Glover, F., Kochenberger, G., Du, Y.: A Tutorial on Formulating and Using QUBO Models. *arXiv:1811.11538* (2018).
8. Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: Introduction and review. *Reports on Progress in Physics* **85**(10), 106001 (2022). <https://doi.org/10.1088/1361-6633/ac8c54>
9. Heidari, S., Dinneen, M. J., Delmas, P.: Quantum annealing for computer vision minimization problems. *Future Generation Computer Systems* **160**, 54–64 (2024). <https://www.sciencedirect.com/science/article/pii/S0167739X24002590>
10. Codognet, P., & Monfroy, E.: Modeling the Cyclic Bandwidth Problem in QUBO for Quantum Annealing. In: Krzhizhanovskaya, V.V., et al. (eds.) *Computational Science – ICCS 2025*, LNCS, vol. 14485, pp. 165–179. Springer, Cham (2025). [https://doi.org/10.1007/978-3-031-97570-7\\_12](https://doi.org/10.1007/978-3-031-97570-7_12)
11. Albash, T., & Lidar, D. A.: Adiabatic quantum computation. *Reviews of Modern Physics* **90**(1), 015002 (2018). <https://doi.org/10.1103/RevModPhys.90.015002>
12. Kadowaki, T., & Nishimori, H.: Quantum annealing in the transverse Ising model. *Physical Review E* **58**(5), 5355–5363 (1998). <https://doi.org/10.1103/PhysRevE.58.5355>
13. Lucas, A.: Ising formulations of many NP problems. *Frontiers in Physics* **2**, 5 (2014). <https://doi.org/10.3389/fphy..2014.00005>
14. Calude, C. S., Dinneen, M. J.: Solving the broadcast time problem using a D-Wave quantum computer. In Adamatzky, A. (Editor), *Advances in Unconventional Computing*, volume 22 of *Emergence, Complexity and Computation*, chapter 17, pages 439–453. Springer International Publishing Switzerland (2017).
15. Dinneen, M. J., Hua, R.: Formulating graph covering problems for adiabatic quantum computers. In *Proceedings of the Australasian Computer Science Week Multi-conference*, ACSW '17, pages 18:1–18:10, New York, NY, USA, (2017).



# Quantum Fuzzy Inference Systems: Implementation and a Case Study on Sleep Apnea Detection

Samuel Magaz-Romero<sup>[0000–0001–6438–5569]</sup>,  
Eduardo Mosqueira-Rey<sup>[0000–0002–4894–1067]</sup>,  
Diego Alvarez-Estevez<sup>[0000–0001–5790–0577]</sup>, and  
Vicente Moret-Bonillo<sup>[0000–0002–9435–3151]</sup>

Universidade da Coruña (CITIC), Campus de Elviña, 15071 A Coruña, Spain  
`{s.magazr,eduardo.mosqueira,diego.alvareze,vicente.moret}@udc.es`

**Abstract.** Fuzzy Inference Systems (FIS) are a powerful tool for problems whose domain can be formally defined thanks to extracting knowledge from experts. These systems provide certain capabilities that current trend technologies do not, but its use has decreased in the past years due to the popularity of Machine Learning methods. In this paper, we revisit FIS through the perspective of Quantum Computing, an approach to make this model more powerful, thanks to superposition and entanglement. We present a new method for implementing FIS with quantum circuits, named Quantum Fuzzy Inference Systems, providing the definitions for the logical operators and implication. We propose a practical application in sleep medicine diagnosis, more specifically for the detection of apneic events. Our results show that we are able to replicate the behavior of the classical model using the quantum proposal, up to a similarity of 99.97%. We conclude on the contribution of this work towards the development of hybrid algorithms and uncertainty management, and pose some possible lines of future work on extending the inference process and its optimization.

**Keywords:** Quantum Computing · Fuzzy Inference Systems · Uncertainty · Sleep Medicine.

## 1 Introduction

When Lofti A. Zadeh first introduced fuzzy sets in 1965, one of his motivations was to provide a formalism to represent certain kinds of knowledge in a closer way to how humans think [20]. This new proposal was widely accepted, giving rise to different applications, like fuzzy controllers [18] or fuzzy inference [8], marking a milestone in classical Artificial Intelligence (AI) history.

However, in the more recent years, Machine Learning (ML) has taken over the AI ecosystem. With the amount of data available nowadays, it is easier to train a model than to struggle with the knowledge engineering process for the tasks that these models are used for.

Nonetheless, ML methods lack the capability that fuzzy tools provide for uncertainty and explainability. Due to each one’s natures, fuzzy tools are able to clearly state the reasons why a certain output was given (inputs, rules activated) when using uncertain data, while ML methods are too opaque for a simple and clear explanation [15]. Therefore, it may be of interest to revisit fuzzy tools under a new perspective, to develop more powerful tools that work under uncertainty and are explainable.

Parallel to these events, Quantum Computing (QC) has taken an important spot among the cutting-edge technologies with potential for next-generational breakthrough. Although its conception goes decades back [9], recent developments in quantum hardware have posed QC as a real contender for a plethora of meaningful applications, in fields like chemistry, finance or health [16,11,10].

This new computing paradigm, based on the postulates of Quantum Mechanics, provides computing capabilities that are simply not realizable by classical computing. Superposition and entanglement play a fundamental role in these advantages, allowing for a more powerful parallel processing and data correlation, among many other feats [17].

In the recently published white paper [1], quantum reasoning is posed as one of the key directions of AI improvement through QC. In particular, fuzzy logic systems are mentioned as one of the models that could be enhanced with quantum approaches.

Within this context, the objective of this work is to present a method to implement fuzzy inference through quantum circuits, providing a basis that can be already used (as we illustrate with an example) but with room to improve by applying QC techniques. The rest of the paper is structured as follows. Section 2 presents the theoretical basis of the work. In Section 3 we present an experiment with a real application of the proposed method. Finally, Section 4 reflects the conclusions achieved during the realization of this work.

## 2 Materials and method

In this section we present the necessary materials on Fuzzy Inference Systems, as well as the new proposed method.

### 2.1 Fuzzy Inference Systems

Consider a universe of discourse  $U$ , which contains the elements to be worked with. We define a fuzzy set  $S$  in  $U$  by a membership function (MF)  $\mu_S(x)$ , which assigns each element  $x \in U$  a real number in the interval  $[0, 1]$ , with the value of  $\mu_S(x)$  at  $x$  representing the “grade of membership” of  $x$  in  $S$ ; the closer  $\mu_S(x)$  is to 1, the more  $x$  belongs to the fuzzy set  $S$ .

Therefore, a fuzzy set  $S$  contains pairs of both the elements in  $U$  and their grade of membership:

$$S = \{(x, \mu_S(x)) \mid x \in U\}. \quad (1)$$

Using fuzzy sets, we can define an extension of classical logic, known as fuzzy logic. Classical logic is then considered a particular case of fuzzy logic, where the membership functions of each fuzzy set only evaluate to either 0 or 1 (crisp values, rather than fuzzy), but still in the range  $[0, 1]$  of fuzzy logic.

This extension calls for a redefinition of the classical logical operators, as their behavior is only defined for the edge cases where the values are either 0 or 1. This is an interesting characteristic of fuzzy logic, as it allows for different implementations of this operators, as long as they respect certain properties specific to fuzzy logical operators (see Table 1).

Table 1: Fuzzy logical operators' properties

| Property         | Definition                                                     |
|------------------|----------------------------------------------------------------|
| Commutativity    | $f(a, b) = f(b, a)$                                            |
| Monotonicity     | $a \leq b \implies f(a, c) \leq f(b, c)$                       |
| Associativity    | $f(f(a, b), c) = f(a, f(b, c))$                                |
| Identity element | $f(a, 1) = a$ for conjunction<br>$f(a, 0) = a$ for disjunction |

We do not pursue this exposition further as it is not necessary for our method, but it must be kept in mind that these new operators add a layer of complexity, necessary to manage fuzzy variables.

With this approach, we can use fuzzy logic to model uncertainty, a characteristic out of reach for classical logic. We can now define values that are not true nor false, but rather something in between. This is specially useful when considered for the process of inference [6].

There are several forms to implement a fuzzy inference process (similar to fuzzy logical operators), but the process itself always follows the same steps:

1. **Fuzzification:** at first, the information is modeled as crisp values from our universe of discourse; it must be transformed into fuzzy variables, using the membership functions that define the fuzzy sets of our system.
2. **Inference engine:** once the information has been fuzzified, the rules of the system are applied. For each rule, its precedent is evaluated according to the fuzzy logical operators that relate the different variables, and then a value is obtained for the fuzzy set of the output in question.
3. **Defuzzification:** after all the rules have been evaluated, the fuzzy sets of the output are aggregated to generate a crisp value, resulting in the final output of the system.

Fuzzy Inference Systems (FIS) are used to realize this very same process in practice. Their structure is illustrated in Figure 1. Besides the elements that carry out the steps of the process that were described above, FIS also have a knowledge base, which contains knowledge of the problem in question. This

knowledge ranges from constraints that the elements must met, to the rules that relate the elements.

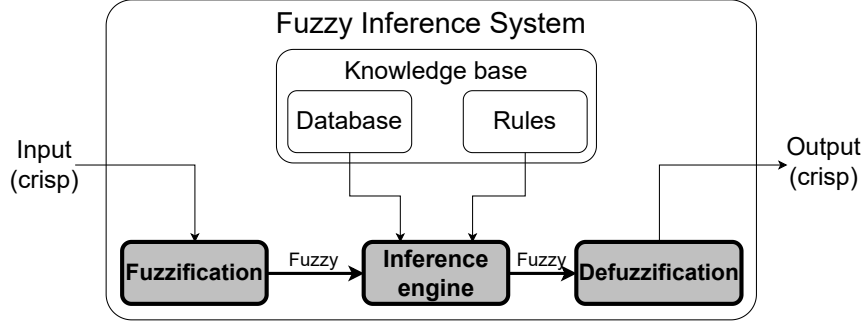


Fig. 1: Fuzzy Inference System structure

In the text that follows, we will focus on the highlighted parts, as the rest of the system remains virtually the same for both classical and quantum paradigms. We provide a formal definition of the elements of FIS that are of interest for this work:

- **Input:** a numeric range must be defined from the universe of discourse  $U$  for each of the variables of our system.

$$x \in [min_x, max_x] \quad (2)$$

- **Fuzzy variable:** a set of fuzzy sets per variable of the system, each of them modeled by a fuzzy set  $X^i$ , with its corresponding membership function.

$$\mu_{X^i}(x) : [min_x, max_x] \rightarrow [0, 1] \quad (3)$$

- **Rules:** the inferential relationships between the fuzzy sets of the inputs to those of the outputs. The operators used for precedents in rules (negation, conjunction, disjunction; example in (4)) and the implication can be implemented through different functions, as long as they abide to the properties of Table 1.

$$\begin{aligned} &\text{IF } (x_0 \text{ is } X_0^i) \in [0, 1] \\ &\quad \wedge (x_1 \text{ is } X_1^j) \in [0, 1] \\ &\quad \dots \\ &\quad \vee (x_n \text{ is } X_n^k) \in [0, 1] \\ &\text{THEN } (y \text{ is } Y^y) \in [0, 1] \end{aligned} \quad (4)$$

We use a shortened notation for fuzzy sets, with  $X_v^s$  being the fuzzy set  $s$  of fuzzy variable  $x_v$ .

- **Output:** once the inference process is carried out, the corresponding values for each of the fuzzy sets of the output are obtained. To aggregate them and generate a crisp output, we use the Takagi-Sugeno-Kang (TSK) inference<sup>1</sup>, as it is a better fit for the quantum encoding used in the proposal, due to its discrete nature:

$$\text{Output (crisp)} = \frac{\sum_{i=0}^{N-1} w_i z_i}{\sum_{i=0}^{N-1} w_i} \quad (5)$$

where  $N$  is the number of rules,  $w_i$  is the value (or weight) of each fuzzy set of the output evaluated through inference, and  $z_i$  is a constant associated to each fuzzy set (similar to the membership functions of the inputs).

## 2.2 Quantum Fuzzy Inference Systems

We now turn to Quantum Computing to define a system equivalent to the classical FIS, which we refer to as Quantum Fuzzy Inference System (QFIS). The goal is to define a process of fuzzy inference through a quantum circuit, using properties as superposition to model the information of the system and quantum operators for the logical operations.

With this purpose, we propose a structure for QFIS as illustrated in Figure 2. Both the input and output of the system remain as crisp values, as well as the processes of fuzzification and defuzzification.

The main difference regarding the classical FIS can be seen in the inference engine, which now is purely quantum. It is divided in the following steps: (1) encoding of the fuzzy variables into quantum states, (2) application of a quantum circuit for inference based on the rules, and (3) measurement of the output qubits to obtain the output fuzzy variable. We proceed to further detail each of these steps in the following lines.

**Encoding** Before delving into details, we want to highlight that this step does not replace the fuzzification process. Once the crisp inputs have been fuzzified, the obtained fuzzy variables are encoded into quantum registers.

In the first place, the qubits used for the inputs are initialized. Each input  $x$  has  $n$  fuzzy sets. Each fuzzy set is mapped to a state of the qubits using amplitude encoding, as it is similar to the idea of fuzzy variables and it is space efficient regarding the amount of qubits (in contrast to other encodings such as basis encoding or angle encoding). This encoding requires  $q = \lceil \log_2 n \rceil$  qubits per input. With  $q$  qubits, we obtain  $2^q$  states, using  $n$  to represent the fuzzy sets:

$$|\psi_x\rangle = \sum_{i=0}^{n-1} \sqrt{\mu_{X^i}(x)} |i\rangle + \sum_{j=n}^{2^q-1} 0 |j\rangle. \quad (6)$$

<sup>1</sup> If required, a normalization operation may be applied to the output obtained, translating the calculated output into a specific range.

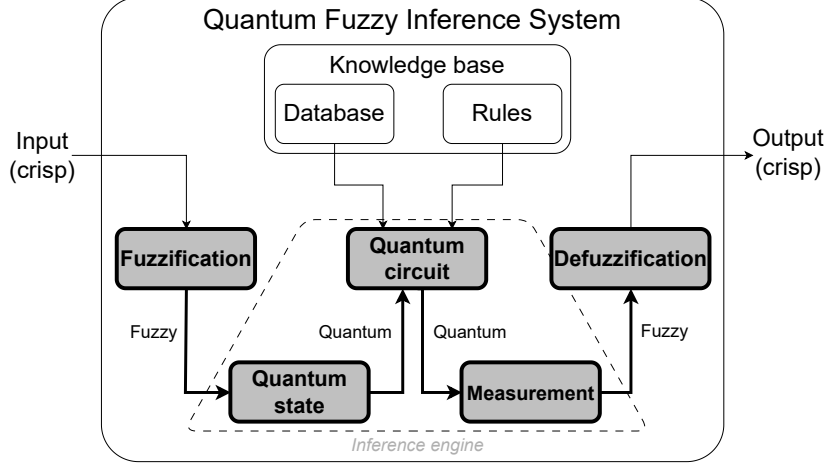


Fig. 2: Quantum Fuzzy Inference System structure

The right part of the sum evaluates to 0; those are the remaining  $2^q - n$  states from the state space, and their amplitude is 0.

Depending on the values of  $\mu_{X^i}(x)$ , a normalization process may be in order, to ensure the correct definition of a quantum state (i.e. ensuring the sum of the modulus squared of the amplitudes equals 1). Let  $amp$  be  $\sum_{i=0}^{n-1} \mu_{X^i}(x)$ . If  $amp = 1$ , no normalization is required. If  $amp < 1$ , we initialize one of the  $2^q - n$  states, say  $g$ , as  $\sqrt{1 - amp} |g\rangle$ , acting as a garbage state. If  $amp > 1$ , we divide each  $\mu_{X^i}(x)$  by  $amp$ , ensuring the normalization.

For the output, we consider a fuzzy variable  $y$  with  $n$  fuzzy sets. We use  $q = \lceil \log_2 n \rceil + 1$  qubits, and initialize them as follows:

$$|\psi_y\rangle = |y_{ex}\rangle \otimes \left( \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle + \sum_{j=n}^{2^q-1} 0 |j\rangle \right). \quad (7)$$

Given that at first we do not know which rules will be activated, we initialize the states associated to the fuzzy sets of the output to an equal superposition, leaving the amplitudes of the remaining states at 0. This configuration allows for an easier implementation of the remaining elements.

The extra qubit  $y_{ex}$  is used to control the activation of the rules. When a rule is activated, we use both the input and the output qubits to control a  $X$  operator on this extra qubit. Therefore the state space is split in two subspaces: the one with the extra qubit in state  $|1\rangle$ , due to the initialization, and the one with the extra qubit in state  $|0\rangle$ , once a rule has been activated and changed this qubit's amplitudes.

**Inference** To define a quantum circuit that realizes the inference process, we must define how to implement the logical operators of the rules' antecedents as well as the implication operator for the rules.

For the precedent of the rules, there may be the case where only a single input is present. In this case, only the qubits of that input are used, and no extra steps are required. However, it is rather common that the precedent of the rule is composed by a combination of multiple inputs.

In the cases where several inputs are used in the precedent of a rule, they are related through the conjunction and disjunction operators. For the conjunction operator, we apply a single quantum operator controlled by the qubits of the inputs, with the control state set to those states that represent the fuzzy sets of the inputs. For the disjunction operator, we apply a combination of controlled quantum operators, each being controlled by the different combinations of states obtained from the inclusion-exclusion principle. These implementations are based on the idea of t-norms and t-conorms, the references for implementing classical fuzzy operators [12]. Figure 3 illustrates both operators with examples.

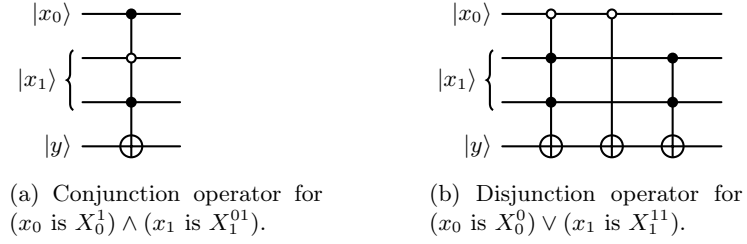


Fig. 3: Examples of quantum operators for conjunction and disjunction.

For the negation of an input, some extra work is required. With the exception of inputs with a single, this operation is non-trivial, as, in general,  $X^{\otimes n} \sqrt{\mu_{X_i}(x)} |i\rangle \neq \sqrt{1 - \mu_{X_i}(x)} |i\rangle$ . Therefore, we employ an extra qubit, initialize it to the state  $|1\rangle$ , and apply a  $X$  operator to it, controlled by the state we want to negate. Then we use this extra qubit (whose amplitude now corresponds to  $\sqrt{1 - \mu_{X_i}(x)}$ ) as a replacement for the original input. Figure 4 illustrates such a case.

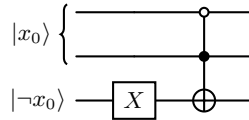


Fig. 4: Negation operator for ( $x_0$  is not  $X_0^{01}$ ).

For the implication, we use the operators defined above plus the output qubits, to control an  $X$  operator on the extra qubit  $y_{ex}$  that we previously specified to divide the state space. This operator modifies the amplitude of the output states proportionally to the amplitudes of the input states, accurately modeling the behavior of classical FIS. Figure 5 illustrates the implementation of an example rule with the elements that we have defined here.

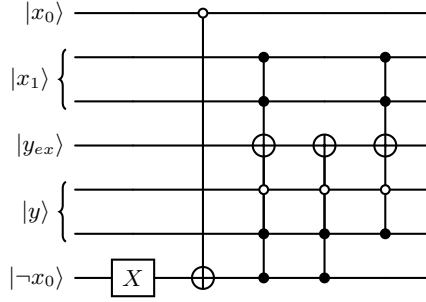


Fig. 5: Quantum circuit for the rule IF  $(x_0 \text{ is not } X_0^1) \vee (x_1 \text{ is } X_1^{11})$  THEN  $(y \text{ is } Y^{01})$

**Measurement** The final step in this new quantum inference engine is measurement. As expected, we measure the output qubits, including both the qubit register for the states and the extra qubit to divide the state space. With these measurements, we obtain bit strings where the first bit corresponds to the extra qubit  $y_{ex}$  that divides the state space, and the remaining bits of the string indicate the measured state on the output qubits, which is related to a specific fuzzy set of the output.

Out of all the shots performed, we use the ones whose first bit is 0 (the ones where the rules were fired, as specified in the encoding of the output), and use the rest of the bits to identify the output qubits' state. The number of shots measured for each state acts as the evaluation of the corresponding fuzzy set of the output, which is later used as the weight in the defuzzification process (see Equation (5)).

With these definitions, we have established a new inference engine for FIS through Quantum Computing, which can be integrated with the classical fuzzification and defuzzification processes, resulting in a hybrid classic-quantum method for our proposal of QFIS.



### 3 Experimentation

In this section, we introduce a real test case scenario by implementing an inference system for the detection of pathological respiratory events in sleep medicine recordings.

#### 3.1 Detection of sleep breathing events

The polysomnographic (PSG) test is the standard diagnostic tool for the diagnosis of sleep disorders. Among the related pathologies, sleep disordered breathing (SDB) is one of the main causes of sleep disruption, characterized by the repeated presence of involuntary respiration pauses during the night, hereafter referred to as *apneic events*. To detect these events, a PSG involves the recording of several biomedical signals, including airflow, thoraco-abdominal movements, and oxygen saturation in arterial blood, among others [19].

Clinical guidelines recommend PSG data to be evaluated by a certified sleep expert. However, this task is time-consuming, and subject to several sources of uncertainty [3]. In this context, (semi-)automatic PSG analysis emerges as a promising approach by assisting clinicians to reduce analysis times and associated inter-rater variability [13,4].

Following this line, we tackle PSG evaluation as a relevant case of study for the methods introduced in this work. More specifically, we take as reference the classical FIS implementation described in past work [7], and implement one of the inference units using the proposed QFIS approach just described.

The chosen inference unit regards the first inference stage, which is used to determine the relevance of a desaturation event in the signal of oxygen saturation ( $SaO_2$ ). This inference unit takes two inputs; the reduction in the saturation ( $SaO_2Red$ ) and the duration of the reduction event ( $SaO_2RedDur$ ). It outputs the degree of severeness of the event ( $Event$ ).

To model this relationship, expert knowledge in the original work was extracted based on the reference clinical guidelines [5]. This knowledge is used to model the definition of the input and output variables and related fuzzy sets, and the corresponding fuzzy IF-THEN rules as presented in Tables 2 and 3 respectively.

#### 3.2 Implementation

After defining the previous knowledge, we can proceed with the implementation of the FIS. We implement it in both classical and quantum paradigms to compare both versions later. Materials for reproducibility of the experiments are accessible online [14].

For the classical implementation, we use MATLAB's Fuzzy Logic Designer, as it is one of the most popular software programs for these systems [2]. It provides out-of-the-box implementations of membership functions, logical operators, and inference tools.

Table 2: Fuzzy sets for the variables of the inference unit.

| Variable                     | Fuzzy set  |                 |                 |               |
|------------------------------|------------|-----------------|-----------------|---------------|
|                              | Name       | Type of MF      | Values          | Quantum state |
| <i>SaO<sub>2</sub>Red</i>    | Low        | Linear Z-shaped | [1 2]           | 000           |
|                              | Normal     | Trapezoidal     | [1.8 2 4 5]     | 001           |
|                              | High       | Trapezoidal     | [4 6 8 10]      | 010           |
|                              | Very High  | Linear S-shaped | [9 12]          | 011           |
| <i>SaO<sub>2</sub>RedDur</i> | Very Short | Linear Z-shaped | [3 4]           | 000           |
|                              | Short      | Trapezoidal     | [3 4 5 6]       | 001           |
|                              | Normal     | Trapezoidal     | [5 6. 535 60]   | 010           |
|                              | Large      | Trapezoidal     | [35 45 120 130] | 011           |
|                              | Very Large | Linear S-shaped | [120 130]       | 100           |
| <i>Event</i>                 | Very Low   | Constant        | [0]             | 000           |
|                              | Low        | Constant        | [0.3]           | 001           |
|                              | Medium     | Constant        | [0.5]           | 010           |
|                              | High       | Constant        | [0.7]           | 011           |
|                              | Very High  | Constant        | [1]             | 100           |

Table 3: Rules of the *SaO<sub>2</sub>* inference unit.

| Rule | Precedent                                                                             | Consequent                |
|------|---------------------------------------------------------------------------------------|---------------------------|
| 1    | <i>SaO<sub>2</sub>Red</i> IS Low                                                      | <i>Event</i> IS Very Low  |
| 2    | <i>SaO<sub>2</sub>Red</i> IS High AND <i>SaO<sub>2</sub>RedDur</i> IS Very Short      | <i>Event</i> IS High      |
| 3    | <i>SaO<sub>2</sub>Red</i> IS High AND <i>SaO<sub>2</sub>RedDur</i> IS Short           | <i>Event</i> IS Medium    |
| 4    | <i>SaO<sub>2</sub>Red</i> IS High AND <i>SaO<sub>2</sub>RedDur</i> IS Large           | <i>Event</i> IS Low       |
| 5    | <i>SaO<sub>2</sub>Red</i> IS High AND <i>SaO<sub>2</sub>RedDur</i> IS Normal          | <i>Event</i> IS Very Low  |
| 6    | <i>SaO<sub>2</sub>Red</i> IS Very High AND <i>SaO<sub>2</sub>RedDur</i> IS Very Short | <i>Event</i> IS Medium    |
| 7    | <i>SaO<sub>2</sub>Red</i> IS Very High AND <i>SaO<sub>2</sub>RedDur</i> IS Low        | <i>Event</i> IS Medium    |
| 8    | <i>SaO<sub>2</sub>Red</i> IS Very High AND <i>SaO<sub>2</sub>RedDur</i> IS Large      | <i>Event</i> IS High      |
| 9    | <i>SaO<sub>2</sub>Red</i> IS Very High AND <i>SaO<sub>2</sub>RedDur</i> IS Normal     | <i>Event</i> IS Very High |
| 10   | <i>SaO<sub>2</sub>Red</i> IS Normal AND <i>SaO<sub>2</sub>RedDur</i> IS Very Short    | <i>Event</i> IS Medium    |
| 11   | <i>SaO<sub>2</sub>Red</i> IS Normal AND <i>SaO<sub>2</sub>RedDur</i> IS Low           | <i>Event</i> IS Medium    |
| 12   | <i>SaO<sub>2</sub>Red</i> IS Normal AND <i>SaO<sub>2</sub>RedDur</i> IS Large         | <i>Event</i> IS High      |
| 13   | <i>SaO<sub>2</sub>Red</i> IS Normal AND <i>SaO<sub>2</sub>RedDur</i> IS Normal        | <i>Event</i> IS Very High |
| 14   | <i>SaO<sub>2</sub>RedDur</i> IS Very High                                             | <i>Event</i> IS Very High |
| 15   | <i>SaO<sub>2</sub>Red</i> IS High AND <i>SaO<sub>2</sub>RedDur</i> IS Very Large      | <i>Event</i> IS Low       |
| 16   | <i>SaO<sub>2</sub>Red</i> IS Very High AND <i>SaO<sub>2</sub>RedDur</i> IS Very Large | <i>Event</i> IS High      |

For the quantum implementation, we use Qiskit to define the quantum circuit according to the proposed method, and simulate the execution using a sampler, as we aim to study the behavior of the proposed model under ideal conditions. We show the resulting circuit in Figure 6, omitting some of the operations for the clarity of its presentation. The initialization of the registers depends on the fuzzification of the values as previously explained, while the operators for the rules and the measurements remain the same.

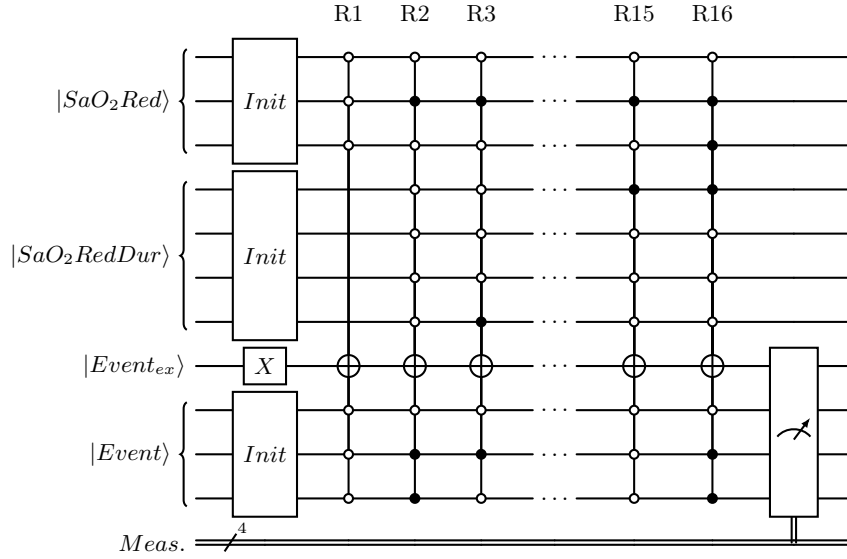


Fig. 6: Quantum circuit for the QFIS of the  $SaO_2$  inference unit.

As illustrated by Figure 6, once given the definitions of the problem for classical FIS, the composition of the quantum circuit for QFIS follows from the definition of the different elements that we provide with the proposed method, making it a powerful framework for the quantum implementation of already-defined classical FIS.

### 3.3 Results

We compare the evaluations obtained with both classical and quantum methods. For the classical FIS, MATLAB provides the results through its Fuzzy Logic Designer, and for QFIS we run the circuit a total of 4096 shots. Figure 7 illustrates the control surface for each method, a 3D plot where the horizontal axes represent the inputs of the system, and the vertical axis represents the corresponding output. This study is not based on datasets, but rather an exhaustive exploration of input ranges.

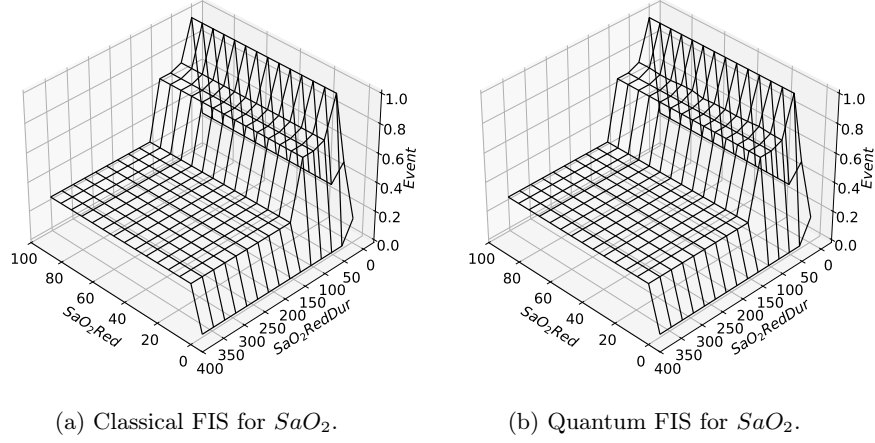


Fig. 7: Control surfaces for classical and quantum FIS.

It can be appreciated how both control surfaces are practically identical, as they present the same behavior when solving the same problem. In fact, the only differences are some slight variations due to the probabilistic nature of Quantum Computing. When compared analytically, we obtain a geometric similarity (shape and proportion) of 99.97% between control surfaces, underlining the success of the experiment.

#### 4 Discussion and conclusions

In this paper, we have presented a new method, named Quantum Fuzzy Inference Systems, for the implementation of Fuzzy Inference Systems under the paradigm of Quantum Computing. Based on the classical elements of FIS, we have provided definitions for their quantum analogues, resulting in a well-defined framework. On top of that, we have tested the proposal with a practical application; the detection of pathological respiratory events in sleep medicine recordings. While the experiment implemented here is only a reasoning unit of a larger system, it showcases the potential of QFIS for these problems.

This new method defines the standard tools of FIS (representation of fuzzy variables, logical fuzzy operators, fuzzy implication) through quantum circuits. These definitions allow for computations as expressive as classical ones, as shown by the results of the experiment. More experimentation is required, to identify possible improvements of the model and to study its behavior with problems of larger scale, but it does not take away from the fact that the model presented here can already be employed for practical applications. In addition, it is worth highlighting the capabilities of QC to represent and manage uncertain information, a feature that will be continued to work on.

As previously explained, the quantum part of QFIS is mostly present on the inference engine, making this method hybrid in nature. The development of such algorithms is currently set to be one of the most prolific branches of QC, and therefore it is of importance to give the proper consideration to this aspect of the proposal. One possible implication that it could present is the potential improvement of energy consumption when compared to classical FIS, although more responsibility relies on the hardware side.

Taking the method presented here, future work will focus on extending the capabilities of QFIS. On the one hand, an important step will be allowing for the chaining of rules, since as of right now it is not possible due to the differences on the encoding of the inputs and outputs. On the other hand, we will research how to apply different techniques of Quantum Computing for optimization, to intend to obtain advantage (when compared to classical FIS) in different aspects, such as computational, energetic, or on runtime. Parallel to these tasks, effort will be dedicated to transversal aspects of developing Quantum Computing solutions, like noise robustness in real hardware and scalability on larger problems (in this case, more inputs and rules).

In conclusion, we hope the work presented here results in a step forward towards the development of Quantum Computing techniques and algorithms, as well as a showcase of the potential of this paradigm to manage uncertain information in real world problems.

**Acknowledgments.** This work is supported by project PID2023-147422OB-I00, funded by MCIU/AEI/10.13039/501100011033 and by the European Regional Development Fund (ERDF) program, and by the Xunta de Galicia (Grant ED431C 2022/44), supported by ERDF. CITIC, as a center accredited for excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Department of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. Additionally, it is co-financed by the EU through the ERDF Galicia 2021-27 operational program (Ref. ED431G 2023/01). SMR has received funding from Xunta de Galicia (grant ED481A 2023/008). DAE has also received support from project RYC2022-038121-I, funded by MCIN/AEI/10.13039/501100011033 and European Social Fund Plus (ESF+), and project ED431F 2025/35 from Xunta de Galicia.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Acampora, G., Ambainis, A., Ares, N., Banchi, L., Bhardwaj, P., Binosi, D., Briggs, G.A.D., Calarco, T., Dunjko, V., Eisert, J., Ezratty, O., Erker, P., Fedele, F., Gil-Fuster, E., Gärttner, M., Granath, M., Heyl, M., Kerenidis, I., Klusch, M., Kockum, A.F., Kueng, R., Krenn, M., Lässl, J., Macaluso, A., Maniscalco, S., Marquardt, F., Michielsen, K., Muñoz-Gil, G., Müssig, D., Nautrup, H.P., Neubauer, S.A., van Nieuwenburg, E., Orus, R., Schmiedmayer, J., Schmitt, M., Slusallek, P., Vicentini, F., Weitenberg, C., Wilhelm, F.K.: Quantum computing and artificial intelligence: status and perspectives (2025). <https://doi.org/10.48550/ARXIV.2505.23860>, <https://arxiv.org/abs/2505.23860>

2. Alcalá-Fdez, J., Alonso, J.M.: A survey of fuzzy systems software: Taxonomy, current research trends, and prospects. *IEEE Transactions on Fuzzy Systems* **24**(1), 40–56 (Feb 2016). <https://doi.org/10.1109/tfuzz.2015.2426212>, <http://dx.doi.org/10.1109/TFUZZ.2015.2426212>
3. Alvarez-Estevez, D.: Challenges of Applying Automated Polysomnography Scoring at Scale. *Sleep Medicine Clinics* **18**(3), 277–292 (Sep 2023). <https://doi.org/10.1016/j.jsmc.2023.05.002>
4. Alvarez-Estevez, D., Rijsman, R.M.: Computer-assisted analysis of polysomnographic recordings improves inter-scorer associated agreement and scoring times. *PLOS ONE* **17**(9), e0275530 (Sep 2022). <https://doi.org/10.1371/journal.pone.0275530>
5. American Academy of Sleep Medicine: The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. American Academy of Sleep Medicine, Westchester, IL (2007), 1st ed.
6. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions. *Fuzzy Sets and Systems* **40**(1), 143–202 (Mar 1991). [https://doi.org/10.1016/0165-0114\(91\)90050-z](https://doi.org/10.1016/0165-0114(91)90050-z), [http://dx.doi.org/10.1016/0165-0114\(91\)90050-Z](http://dx.doi.org/10.1016/0165-0114(91)90050-Z)
7. Álvarez Estevez, D., Moret-Bonillo, V.: Fuzzy reasoning used to detect apneic events in the sleep apnea-hypopnea syndrome. *Expert Systems with Applications* **36**(4), 7778–7785 (May 2009). <https://doi.org/10.1016/j.eswa.2008.11.043>, <http://dx.doi.org/10.1016/j.eswa.2008.11.043>
8. Fazle Azeem, M.: Fuzzy Inference System - Theory and Applications. InTech (May 2012). <https://doi.org/10.5772/2341>, <http://dx.doi.org/10.5772/2341>
9. Feynman, R.P.: Simulating physics with computers. *International Journal of Theoretical Physics* **21**(6–7), 467–488 (Jun 1982). <https://doi.org/10.1007/bf02650179>, <http://dx.doi.org/10.1007/BF02650179>
10. Flöther, F.F.: The state of quantum computing applications in health and medicine. *Research Directions: Quantum Technologies* p. 1–21 (Jul 2023). <https://doi.org/10.1017/qut.2023.4>, <http://dx.doi.org/10.1017/qut.2023.4>
11. Herman, D., Googin, C., Liu, X., Sun, Y., Galda, A., Safro, I., Pistoia, M., Alexeev, Y.: Quantum computing for finance (2023). <https://doi.org/10.48550/ARXIV.2307.11230>, <https://arxiv.org/abs/2307.11230>
12. Hájek, P.: Many-Valued Propositional Calculi, p. 27–61. Springer Netherlands (1998). [https://doi.org/10.1007/978-94-011-5300-3\\_2](https://doi.org/10.1007/978-94-011-5300-3_2), [http://dx.doi.org/10.1007/978-94-011-5300-3\\_2](http://dx.doi.org/10.1007/978-94-011-5300-3_2)
13. Kristiansen, S., Traaen, G.M., Øverland, B., Plagemann, T., Gullestad, L., Akre, H., Nikolaidis, K., Aakerøy, L., Hunt, T.E., Loennechen, J.P., Steinshamn, S., Bendz, C., Anfinsen, O., Goebel, V.: Comparing manual and automatic scoring of sleep monitoring data from portable polygraphy. *Journal of Sleep Research* **30**(2) (May 2020). <https://doi.org/10.1111/jsr.13036>, <http://dx.doi.org/10.1111/jsr.13036>
14. Magaz-Romero, S.: samu-magaz/qfis-pricai-2025: qfis-pricai-2025-0.1 (Jun 2025). <https://doi.org/10.5281/zenodo.15753573>, <https://doi.org/10.5281/zenodo.15753573>
15. Marcinkevičs, R., Vogt, J.E.: Interpretability and explainability: A machine learning zoo mini-tour (2020). <https://doi.org/10.48550/ARXIV.2012.01805>, <https://arxiv.org/abs/2012.01805>
16. Motta, M., Rice, J.E.: Emerging quantum computing algorithms for quantum chemistry. *WIREs Computational Molecular Science* **12**(3) (Dec 2021). <https://doi.org/10.1002/wcms.1580>, <http://dx.doi.org/10.1002/wcms.1580>

17. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (Jun 2012). <https://doi.org/10.1017/cbo9780511976667>, <http://dx.doi.org/10.1017/CB09780511976667>
18. Precup, R.E., Hellendoorn, H.: A survey on industrial applications of fuzzy control. *Computers in Industry* **62**(3), 213–226 (Apr 2011). <https://doi.org/10.1016/j.compind.2010.10.001>, <http://dx.doi.org/10.1016/j.compind.2010.10.001>
19. Rundo, J.V., Downey, R.: Polysomnography, p. 381–392. Elsevier (2019). <https://doi.org/10.1016/b978-0-444-64032-1.00025-4>, <http://dx.doi.org/10.1016/B978-0-444-64032-1.00025-4>
20. Zadeh, L.: Fuzzy sets. *Information and Control* **8**(3), 338–353 (Jun 1965). [https://doi.org/10.1016/s0019-9958\(65\)90241-x](https://doi.org/10.1016/s0019-9958(65)90241-x), [http://dx.doi.org/10.1016/S0019-9958\(65\)90241-X](http://dx.doi.org/10.1016/S0019-9958(65)90241-X)

# Benchmarking Under Fragility in a Shifting Quantum Landscape

Nathan Kittichaikoonkij<sup>1</sup>[0009–0006–0266–6959],  
Nutthapat Pongtanyavichai<sup>1</sup>[0009–0008–2527–3730],  
Poopha Suwananek<sup>1</sup>[0009–0008–5973–9148],  
Prabhas Chongstitvatana<sup>1</sup>[0000–0003–0744–7801], and  
Kamonluk Suksen<sup>1</sup>[0009–0006–1452–6400]\*

Chulalongkorn University, Bangkok, Thailand  
{6538047621, 6532068721, 6532141821}@student.chula.ac.th,  
Prabhas.C@chula.ac.th, kamonluk@cp.eng.chula.ac.th

**Abstract.** Quantum computing holds promise for speedups on many hard problems, especially in optimization, yet the practical evaluation of quantum algorithms is increasingly hindered by the volatility of cloud-based quantum services. In this study, we confront the challenge of reproducibility and infrastructure fragility through a case-based investigation comparing classical solvers (Gurobi, Fixstars) with D-Wave’s quantum annealer. While our original intent was to benchmark solver performance on standard QUBO formulations, repeated disruptions—deprecations of IBM’s Qiskit backends, inconsistencies in quantum API behavior, and unstable parameter mappings—reframed our effort into a study of methodological brittleness. We argue that this infrastructural instability is not an incidental inconvenience but a core research challenge in quantum benchmarking in a rapidly shifting computational landscape.

**Keywords:** Quantum Optimization · QUBO · Quantum Annealing.

## 1 Introduction

Quantum optimization via Quadratic Unconstrained Binary Optimization (QUBO) has become more prevalent due to its potential to outperform classical solvers such as Gurobi and Fixstars on certain combinatorial optimization problems. However, the current landscape of quantum benchmarking remains fragile; cloud services and software stacks evolve rapidly, APIs are frequently deprecated, and platform access models shift. These continual changes often render previously reproducible experiments obsolete, compromising even well-designed studies.

In response to these challenges, the quantum benchmarking literature increasingly calls for standards that go beyond raw speed or accuracy. For instance, Hashim et al. [9] propose a three-tier framework encompassing quantum device characterization, verification of solution validity, and validation of application

---

\* Corresponding author



correctness. However, their focus is primarily at the hardware level and does not address software and platform volatility.

In our experiment, we attempted to benchmark QUBO solvers across three classes of problems—3SAT, Quadratic Assignment (QAP), and the Traveling Salesman Problem (TSP)—using classical (Gurobi, Fixstars) and quantum (D-Wave, Qiskit/QAOA) approaches. Our efforts to deploy QAOA on IBM Qiskit were hindered by backend incompatibilities and API changes<sup>1</sup>.

Despite attempts to mitigate these issues by downgrading to earlier Qiskit versions, we continued to encounter persistent failures executing quantum algorithms as API updates broke existing code and backend services either became unavailable or exhibited undocumented behavior. These recurring disruptions not only made it difficult to maintain a consistent experimental environment but also undermined the reproducibility and reliability of our benchmarking efforts.

This paper argues that such fragility is more than an inconvenience—it represents a substantive research challenge. We reframe our prior benchmark study within this broader context, demonstrating that evaluating QUBO solvers in an evolving quantum computing landscape reveals inherent vulnerabilities in reproducibility, fairness, and methodological integrity.

## 2 Related Work

### 2.1 Benchmarking Classical Solvers for QUBO Problems

Classical methods for solving QUBO problems rely on mature combinatorial optimization techniques. Gurobi is a widely used commercial solver that applies mixed-integer programming (MIP) with presolve strategies, cutting planes, and heuristics to handle binary quadratic forms [6]. Fixstars Amplify provides a GPU-accelerated simulated annealing engine tailored to QUBO problems, offering fast approximate solutions via thermal heuristics [5]. Other works have explored parallel tempering and metaheuristic hybrids for QUBO, although few studies benchmark these methods under shared problem formulations.

Codognet et al. compared digital annealing with D-Wave on QAP instances, showing that classical annealers can outperform quantum hardware depending on problem structure and embedding efficiency [1].

### 2.2 Quantum Annealing Benchmarks

Quantum annealing (QA) has been experimentally evaluated on problems like Max-Cut, 3SAT, and QAP [10]. The D-Wave Advantage series supports Ising and QUBO formats using minor embedding to map logical variables to hardware qubits. However, embedding overhead and chain breakage remain significant limitations [2]. Villar-Rodriguez et al. conducted a large-scale sensitivity

<sup>1</sup> <https://quantum.cloud.ibm.com/docs/en/api/qiskit/release-notes>

study, revealing that performance depends heavily on tuning parameters like `chain_strength`, `annealing_time`, and `schedule` [16].

These findings emphasize the need for controlled benchmarking methodologies that isolate performance factors across parameter sweeps.

### 2.3 Gate-Based Quantum Optimization

Gate-model solvers like QAOA (Quantum Approximate Optimization Algorithm) provide an alternative to annealing, encoding QUBO problems into parameterized quantum circuits [4]. Implementations in Qiskit and Cirq offer access to both simulators and real devices, while CUDA-Q is focused on simulations and does not currently support real quantum devices. However, practical execution of QAOA remains constrained by circuit depth, noise, and calibration drift.

## 3 Background

### 3.1 QUBO and Ising Formulations

Combinatorial optimization problems often admit reformulations into the *Quadratic Unconstrained Binary Optimization* (QUBO) model, a standard mathematical structure in both classical and quantum computing [7]. A QUBO instance is defined by a real symmetric or upper triangular matrix  $Q \in \mathbb{R}^{n \times n}$  and seeks a binary vector  $x \in \{0, 1\}^n$  minimizing the objective function:

$$E(x) = x^T Q x \quad (1)$$

This framework enables encoding of problems such as Max-Cut, 3SAT, QAP, and TSP by transforming constraints into penalty terms. For example, a constrained problem of the form  $Ax = b$  can be absorbed into the QUBO cost using a quadratic penalty term:

$$E(x) = x^T Q x + \lambda \|Ax - b\|^2 \quad (2)$$

where  $\lambda > 0$  controls the weight of the constraint penalty. The resulting function remains quadratic in  $x$ , allowing the entire problem to be expressed as a QUBO.

QUBO is equivalent to the Ising model, commonly used in quantum annealing. The transformation between the binary variable  $x_i \in \{0, 1\}$  and the Ising spin variable  $s_i \in \{-1, +1\}$  is given by:

$$x_i = \frac{1 + s_i}{2} \quad (3)$$

Applying this transformation, the QUBO Hamiltonian becomes:

$$E(\mathbf{x}) = \sum_i Q_{ii} x_i + \sum_{i < j} Q_{ij} x_i x_j, \quad (4)$$

$$E(\mathbf{s}) = \sum_i Q_{ii} \left( \frac{1+s_i}{2} \right) + \sum_{i<j} Q_{ij} \left( \frac{1+s_i}{2} \right) \left( \frac{1+s_j}{2} \right) \quad (5)$$

$$= \sum_i \frac{Q_{ii}}{2} (1+s_i) + \sum_{i<j} \frac{Q_{ij}}{4} (1+s_i)(1+s_j) \quad (6)$$

$$E(\mathbf{s}) = \text{const} + \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j, \quad (7)$$

where  $h_i$  and  $J_{ij}$  are derived from  $Q$ , and the resulting energy landscape corresponds to the Ising Hamiltonian model [12].

### 3.2 Quantum Annealing Process

Quantum annealing (QA) is a metaheuristic that uses quantum tunneling to explore the solution space of discrete optimization problems. The annealing process begins with a driver Hamiltonian  $H_D$  whose ground state is easily prepared (often a transverse field). Over a time-dependent schedule, the system interpolates toward a problem Hamiltonian  $H_P$  encoding the QUBO (or Ising) cost function:

$$H(t) = A(t)H_D + B(t)H_P \quad (8)$$

The coefficients  $A(t)$  and  $B(t)$  define the annealing schedule, which generally satisfies  $A(0) \gg B(0)$  and  $A(T) \ll B(T)$ , where  $T$  is the total annealing time. Under adiabatic conditions, the system remains in its instantaneous ground state throughout evolution [14].

Unlike classical simulated annealing, which relies on thermal noise to escape local minima, QA leverages quantum tunneling, enabling transitions across energy barriers that may trap classical solvers [15].

### 3.3 QAOA on Gate-based Quantum Computers

The Quantum Approximate Optimization Algorithm (QAOA) is a variational algorithm designed for gate-based quantum devices, inspired by adiabatic quantum computation and quantum annealing. QAOA approximates the solution to combinatorial optimization problems, such as QUBO or Ising models, by alternating between the application of a problem Hamiltonian  $H_P$  and a driver Hamiltonian  $H_D$ .

The QAOA circuit consists of  $p$  layers, each applying a unitary evolution under the Hamiltonians  $H_P$  and  $H_D$  with variational parameters  $\gamma_k$  and  $\beta_k$ :

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \prod_{k=1}^p e^{-i\beta_k H_D} e^{-i\gamma_k H_P} |+\rangle^{\otimes n} \quad (9)$$

Here,  $|+\rangle^{\otimes n}$  is the uniform superposition over all computational basis states. The parameters  $\gamma = (\gamma_1, \dots, \gamma_p)$  and  $\beta = (\beta_1, \dots, \beta_p)$  are optimized by a classical outer loop to minimize the expectation value of  $H_P$  [4].

In the limit  $p \rightarrow \infty$ , QAOA can theoretically reproduce the adiabatic trajectory of quantum annealing, but in practice, even shallow circuits often yield good approximate solutions. Unlike analog quantum annealing, QAOA is compatible with near-term devices, noisy gate-model quantum hardware and enables hybrid quantum-classical optimization [8].

## 4 Experiment

### 4.1 Solver Overview

We evaluated solver frameworks across combinatorial optimization problems: Fixstars Amplify, D-Wave Advantage, and IBM Qiskit QAOA. All solvers were accessed via official cloud APIs or SDKs using default configurations, unless otherwise stated.

While additional classical solvers, including Gurobi and brute-force baselines, were included in our full benchmark suite, we will not go into details for all of them. Full comparative results are reported in our prior work [11] and are excluded here for brevity and clarity of discussion.

**Fixstars Amplify.** A GPU-accelerated, quantum-inspired annealer that performs simulated annealing (SA) over QUBO-defined landscapes. It runs entirely on cloud hardware via REST API. Experiments used the default Simulated Annealing engine under the Basic Evaluation Plan.

**D-Wave Quantum Annealing.** D-Wave Quantum Annealing. Experiments were conducted using the Advantage System 6.4 quantum annealer via the D-Wave Leap cloud platform. We used the Amplify SDK and recorded execution times based on `amplify.Result.Solution.time`, which corresponds to the reported QPU access time (including both QPU programming and sampling). Default parameters were employed except for `num_reads`, set to 1000 to ensure sufficient sampling, with each read using the default annealing time of 20  $\mu$ s. The total reported time therefore includes intrinsic QPU operation overheads beyond the raw annealing time, but excludes non-QPU processes such as embedding or network latency.

**IBM Qiskit QAOA.** Due to persistent execution failures, QAOA was excluded from the benchmark comparison and analyzed qualitatively in Section 5.

### 4.2 QUBO Model Formulations

Each combinatorial problem was encoded in QUBO form. For constraint-based problems (QAP and TSP), constraints were incorporated using quadratic penalty terms scaled by a fixed weight  $\lambda$ . The general QUBO structure takes the form:

$$H = H_O + \lambda g(x) \quad (10)$$

where  $H_O$  is the objective and  $g(x)$  is the penalty function enforcing feasibility. Penalty weights were chosen based on empirical calibration to prevent solution distortion while preserving constraint enforcement.

The formulation for each of the problems is from our previous work [11] as follows:

– **3SAT**

$$H_{3SAT} = - \sum_{i=1}^m \left( (1 + w_i)(y_{i1} + y_{i2} + y_{i3}) - y_{i1}y_{i2} - y_{i1}y_{i3} - y_{i2}y_{i3} - 2w_i \right) + K \quad (11)$$

where  $y_{ij}$  are binary variables representing literals ( $x_i$ ),  $w_i$  are the binary variables associated with each clause, and  $K$  is an offset for normalization and is the minimum number of clauses which are satisfied no matter the binary literals. [3]

– **QAP**

$$H_O = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ik} d_{jl} x_{ij} x_{kl} \quad (12)$$

$$g = \sum_{i=1}^n \left( \sum_{j=1}^n x_{ij} - 1 \right)^2 + \sum_{j=1}^n \left( \sum_{i=1}^n x_{ij} - 1 \right)^2 \quad (13)$$

$$H_{QAP} = H_O + \lambda g \quad (14)$$

where  $f_{ij}$  represents the flow (interaction) between facilities  $i$  and  $j$ ,  $d_{ij}$  is the distance between locations  $i$  and  $j$ ,  $x_{ij}$  is a binary variable indicating if facility  $i$  is assigned to location  $j$ , and  $n$  is the number of facilities and number of locations.

– **TSP**

$$H_O = \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^{n-1} d_{ij} x_{i,p} x_{j,(p+1)} + \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{i,n} x_{j,1} \quad (15)$$

$$g = \sum_{i=1}^n \left( \sum_{p=1}^n x_{i,p} - 1 \right)^2 + \sum_{p=1}^n \left( \sum_{i=1}^n x_{i,p} - 1 \right)^2 \quad (16)$$

$$H_{TSP} = H_O + \lambda g \quad (17)$$

where  $x_{i,p}$  is Binary Variable indicating whether city  $i$  is being visited at time  $p$ ,  $d_{ij}$  is distance between cities  $i$  and  $j$ , and  $n$  is the total number of cities.

### 4.3 Execution Environment and Protocols

All solvers were evaluated on identical problem instances generated using fixed seeds. D-Wave and Fixstars experiments were run via their respective cloud APIs. Solver performance was measured under the following protocols:

- Each solver was given a maximum runtime of 10 seconds per problem instance.
- Fixstars and D-Wave were used as-is through the Amplify SDK.
- D-Wave’s `num_reads` parameter was set to 1000.
- Each reported result represents the average of 10 independent trials.

Performance was measured in terms of success rate, solution accuracy, and time-to-first-optimum (see Section 4.4 for metric definitions).

### 4.4 Experimental Metrics

To evaluate solver behavior, we recorded:

- **Feasibility Rate:** The proportion of runs returning syntactically valid solutions.
- **Solution Accuracy:** The percentage of solutions matching known optima or baselines.
- **Execution Time:** Average solver runtime per instance (ms).

### 4.5 Experiment Result

The results indicate that Fixstars achieves high accuracy with relatively short runtimes, showing better scalability than traditional solvers. Its performance on sparse problems like 3SAT remains consistently strong, and even dense problems like QAP are solvable up to moderate sizes.

In contrast, the D-Wave quantum annealer currently performs worse overall due to limitations in processing larger problem sizes on its hardware, effectively handling sparse problems like 3SAT but struggling with dense ones such as QAP and TSP because of minor embedding challenges and topology constraints.

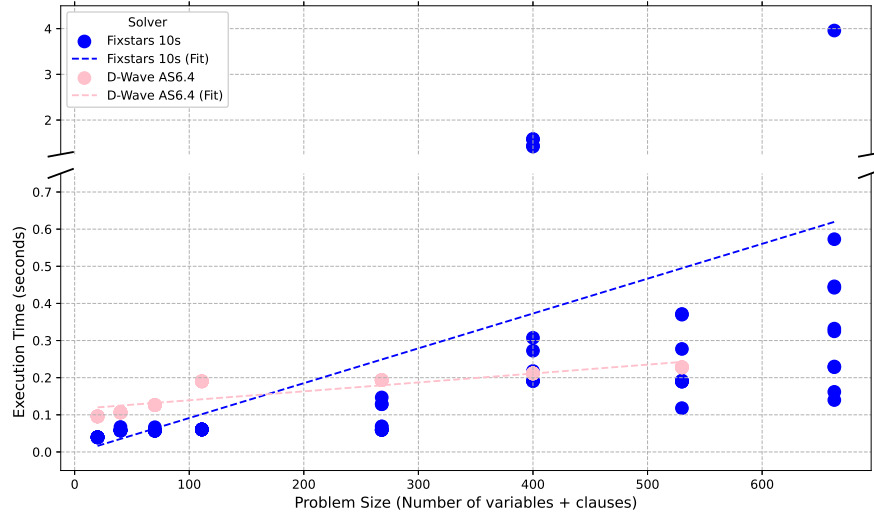
Despite these limitations, D-Wave exhibits promising signs of linear time growth, with only modest increases as problem sizes scale. This trend is evident across all three problem types (Figures 1,2,3), highlighting its potential for future advancements in tackling complex combinatorial optimization efficiently.

These findings underscore the evolving role of quantum annealing in the Noisy Intermediate-Scale Quantum (NISQ) era, emphasizing the need for hybrid approaches to overcome current hardware barriers and achieve broader quantum advantages.

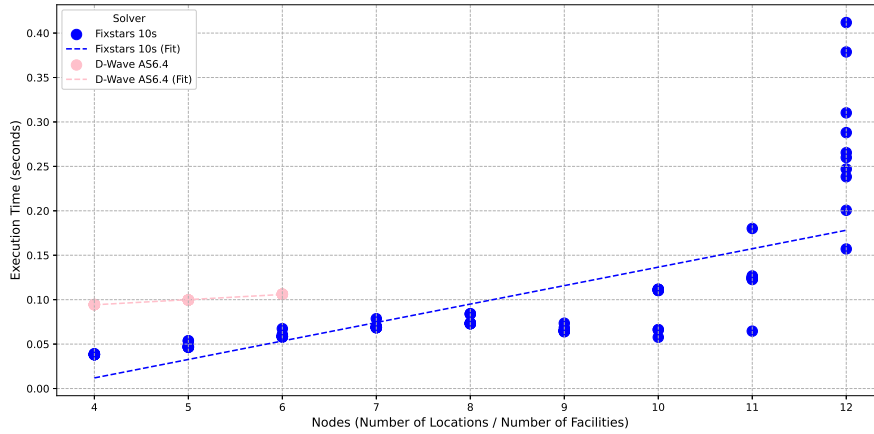
Table 1 is adapted from our previous work [11].

**Table 1.** Summary of D-Wave results for 3SAT, QAP, and TSP

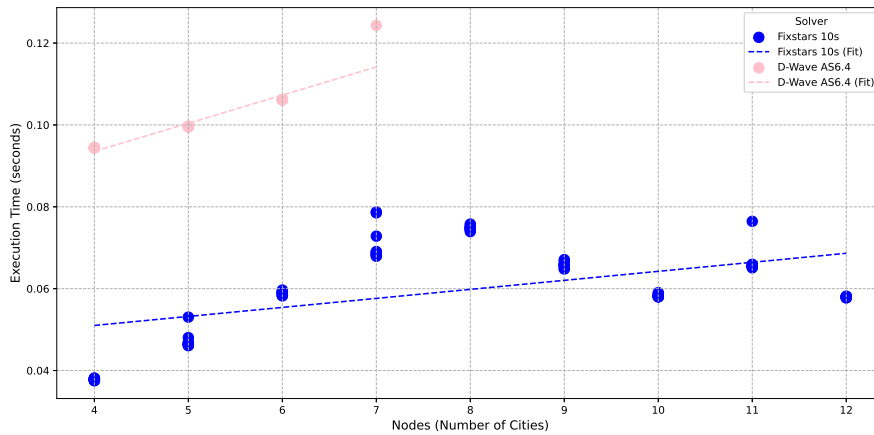
| 3SAT |         |           | QAP  |        |          | TSP  |        |          |
|------|---------|-----------|------|--------|----------|------|--------|----------|
| size | acc(%)  | time (ms) | size | acc(%) | time(ms) | size | acc(%) | time(ms) |
| 20   | 100.000 | 95.803    | 4    | 100    | 94.444   | 4    | 100    | 94.443   |
| 40   | 99.000  | 106.503   | 5    | 70     | 99.603   | 5    | 100    | 99.603   |
| 70   | 97.091  | 126.205   | 6    | 0      | 106.165  | 6    | 20     | 106.164  |
| 111  | 97.033  | 190.226   |      |        |          | 7    | 0      | 124.323  |
| 268  | 97.523  | 193.467   |      |        |          |      |        |          |
| 400  | 95.661  | 211.567   |      |        |          |      |        |          |
| 530  | 95.721  | 227.981   |      |        |          |      |        |          |



**Fig. 1.** 3SAT: Execution Time for Fixstars and D-Wave with Linear Regression Lines. Adapted from [11].



**Fig. 2.** QAP: Execution Time for Fixstars and D-Wave with Linear Regression Lines. Adapted from [11].



**Fig. 3.** TSP: Execution Time for Fixstars and D-Wave with Linear Regression Lines. Adapted from [11].



## 5 Benchmark Fragility

Reproducibility is a well-established concern in quantum algorithm evaluation, but its root causes increasingly stem from infrastructural—not algorithmic—fragility. Mauerer & Scherzinger (2022) explicitly highlight the importance of reproducibility engineering in quantum software experiments. Their approach advocates for packaging code and configuration to remain traceable even when common hardware or vendor platforms change [13].

Our own experience supports this diagnosis. After initial success using D-Wave’s Advantage System 6.4 through the Leap cloud interface, it could not be accessed with the same free-tier account after D-Wave releases a new pricing model. In the case of IBM Qiskit, we attempted to follow the official QAOA MaxCut tutorial<sup>2</sup>, which was functional on simulators. However, adapting the same circuit logic to more general QUBO problems like TSP failed to yield valid results. Even on simulation, QAOA produced nonsensical tours for instances less than 5 nodes. This contrasts sharply with classical solutions obtained using NumPy and Scipy-based optimizers, which matched the known optimum. Furthermore, when transpiled and submitted to real quantum backends, jobs consistently failed to complete due to interactivity timeouts in the Qiskit Runtime environment. These timeouts likely stemmed from session mismanagement or delays exceeding the platform’s job queuing threshold, which disconnects users from backend access during extended idle periods between QAOA iterations.

## 6 Discussion

Our benchmarking results reveal not only the relative performance of quantum and classical QUBO solvers, but also the operational fragility that underlies current quantum computing infrastructure—both at the hardware and software level. This section reflects on the empirical findings, methodological decisions, and broader reproducibility implications, particularly in light of evolving quantum cloud services.

### 6.1 QAOA and Workflow Limitations in General QUBO Problems

Building on the practical failures discussed in Section 5, we now reflect on the broader implications these issues have for solver behavior, quantum benchmarking methodology. Although the Qiskit version (1.3.1) supported QAOA under the `qiskit.algorithms` and `qiskit.optimization` modules, these modules were removed from official qiskit support at the time of experimentation. They remained widely indexed in online documentation and tutorials, leading us to adopt an outdated implementation path unknowingly.

These failures were not due to quantum noise, or algorithmic instability, but to volatility at the software and platform layers. Such infrastructural disruptions

<sup>2</sup> <https://quantum.cloud.ibm.com/docs/en/tutorials/quantum-approximate-optimization-algorithm>

pose a methodological threat to the integrity of longitudinal quantum experiments and must be addressed as a first-class concern in quantum benchmarking research.

## 6.2 Reproducibility and Structural Fragility in QCaaS Workflows

The broader insight from our benchmarking effort is the fragility of quantum workflows under the QCaaS paradigm. Access to D-Wave’s cloud platform was inconsistent: after initial experiments using Advantage 6.4, subsequent team members were unable to access the same solver endpoints under the free-tier account. Similarly, software-layer volatility in Qiskit made it difficult to execute and generalize otherwise functional circuits. These issues were not due to algorithmic shortcomings, but rather to unstable access models, poorly maintained legacy APIs, and weak documentation pathways.

This supports the view that reproducibility in quantum computing must be treated as a systems-level concern. Fragility arises not only from quantum noise or hardware constraints but from the broader ecosystem—SDK evolution, cloud API policies, and the discoverability of supported workflows. Until these structural issues are addressed, benchmarking results must be interpreted within the context of their platform dependencies and temporal validity.

## 6.3 Toward Robust Benchmarking Standards

The fragility observed in quantum benchmarking is not an incidental artifact—it is a reproducible phenomenon that demands formal attention. As quantum computing transitions into an infrastructure-intensive discipline, benchmarking methodologies must evolve to address both algorithmic performance and systems-level resilience.

To that end, we propose a four-pronged framework for designing robust quantum benchmarks that are resilient to evolving APIs, deprecations, and cloud-access variability. This framework aims to standardize practices across research teams and improve reproducibility in future studies.

1. **Standardized Baselines:** Adopt shared, publicly available QUBO formulations with open-source reference implementations and solution sets. These should include canonical problems like Max-Cut, 3SAT, QAP, and TSP across defined sizes (e.g., 4–20 variables). Publishing not just problem definitions but also embedding logic and parameter settings ensures consistent testing.
2. **Snapshot-Aware Experimentation:** Every benchmark run should log the full software and platform stack, including:
  - Backend name and version (e.g., `advantage_6.4`, `ibmq_brisbane`)
  - SDK version (e.g., `qiskit==1.3.1`)
  - API changes or warnings during execution
  - Execution date, user tier (free, pay as you go, premium, etc), and region (when relevant)

These metadata form a "benchmark snapshot" that, while not guaranteeing full reproducibility on real quantum hardware—due to ongoing backend evolution and calibration drift—enable more reliable re-execution in simulators and simplify migration to updated software stacks.

3. **Resilient Methodology:** Benchmarks should emphasize robustness over narrow tuning. Parameter sensitivity (e.g., D-Wave’s `chain_strength`, annealing time) should be reported as distributions or sweeps, not single values. Similarly, use QUBO encodings that are platform-agnostic—avoiding reliance on proprietary transpilation steps when possible.
4. **Benchmarking the Benchmarks:** Core benchmark suites should be re-executed periodically (e.g., quarterly or annually) to measure longitudinal drift in performance or compatibility. This meta-benchmarking helps detect when infrastructure evolution introduces silent errors, regressions, or improvements.

Together, these components shift benchmarking from a one-off evaluation to a reproducible and portable protocol. In future work, we envision a federated benchmarking registry—akin to MLPerf in machine learning—that tracks quantum benchmark scores, version metadata, and known failure modes.

Finally, we recommend that platform-layer failure modes (e.g., runtime errors, silent crashes, or API deprecations) be explicitly reported in benchmark papers, not discarded as outliers. Treating infrastructure behavior as part of the benchmark result will accelerate progress toward both trustworthy evaluations and robust software-hardware co-design.

## 7 Conclusion

This study revisits the problem of benchmarking quantum and classical solvers on combinatorial optimization tasks encoded as QUBO models. Our original goal was to evaluate D-Wave’s quantum annealing system alongside classical baselines and gate-based quantum algorithms. While our experiments did succeed in comparing D-Wave, Fixstars, Gurobi, and Brute Force on standard benchmarks (3SAT, QAP, TSP), a deeper insight emerged: the fragility of quantum benchmarking pipelines is now an intrinsic feature of working in this domain.

D-Wave’s annealer performed reliably within its embedding limits, showcasing low-latency performance and high solution accuracy. Fixstars Amplify, a quantum-inspired classical solver, consistently delivered the fastest results across problem types and sizes. Gurobi remained a strong general-purpose baseline. Yet despite our preparation, efforts to run QAOA on IBM Quantum failed due to systemic shifts in the software ecosystem: deprecated primitives, incomplete V2 support, and mismatched transpilation workflows rendered the platform unusable for our purposes. These failures stemmed from structural volatility, not user or algorithmic error.

The implications of these results extend beyond raw performance. Benchmarking in quantum computing is no longer only a matter of evaluating speed

or scalability—it now requires explicit awareness of platform evolution, software versioning, and service deprecation. As Hashim et al. note [9], reproducibility in quantum computing must grapple with a moving target.

**We emphasize: Benchmarking in quantum computing is now an exercise in infrastructure navigation, reproducibility engineering, and version-aware experimentation. Platform instability—whether through API changes, job failures, or embedding collapse—must be recognized as a primary challenge.**

To move the field forward, we recommend:

- Treating solver configurations and backend versions as first-class experimental parameters.
- Recognizing and reporting platform-level failure modes as benchmark outcomes.

As quantum computing transitions from theoretical promise to practical evaluation, methodological resilience will be as critical as raw performance. Our findings are not a verdict on solver supremacy, but a call for mature, reproducible, infrastructure-aware benchmarking standards in the next phase of quantum algorithm research.

**Acknowledgments.** The first three authors contributed equally to this work.

## References

1. Codognet, P., Diaz, D., Abreu, S.: Quantum and digital annealing for the quadratic assignment problem. In: 2022 IEEE International Conference on Quantum Software (QSW). pp. 1–8 (2022). <https://doi.org/10.1109/QSW55613.2022.00016>
2. D-Wave: Programming the d-wave qpu: Parameters for beginners (2020), <https://www.dwavequantum.com/media/qvbjrzgg/guide-2.pdf>, accessed: 2025-04-17
3. Dinneen, M.J.: Maximum 3-sat as qubo (2016), <https://canvas.auckland.ac.nz/courses/14782/files/574983/download>, lecture notes, CompSci 750, Semester 2, University of Auckland
4. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm (2014), <https://arxiv.org/abs/1411.4028>
5. Fixstars: Fixstars amplify annealing engine (2024), <https://amplify.fixstars.com/en/docs/amplify/v1/clients/fixstars.html>, accessed: 2025-04-17
6. Fixstars: Gurobi optimizer - fixstars amplify sdk documentation (2024), <https://amplify.fixstars.com/en/docs/amplify/v1/clients/gurobi.html>, accessed: 2025-01-05
7. Glover, F., Kochenberger, G., Du, Y.: A tutorial on formulating and using qubo models (2019)
8. Harrigan, M.P., Sung, K.J., Neeley, M., Satzinger, K.J., Arute, F., Arya, K., Atalaya, J., Bardin, J.C., Barends, R., Boixo, S., Broughton, M., Buckley, B.B., Buell, D.A., Burkett, B., Bushnell, N., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Demura, S., Dunsworth, A., Eppens, D., Fowler, A., Foxen, B.,

- Gidney, C., Giustina, M., Graff, R., Habegger, S., Ho, A., Hong, S., Huang, T., Ioffe, L.B., Isakov, S.V., Jeffrey, E., Jiang, Z., Jones, C., Kafri, D., Kechedzhi, K., Kelly, J., Kim, S., Klimov, P.V., Korotkov, A.N., Kostritsa, F., Landhuis, D., Laptev, P., Lindmark, M., Leib, M., Martin, O., Martinis, J.M., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Mohseni, M., Mruczkiewicz, W., Mutus, J., Naaman, O., Neill, C., Neukart, F., Niu, M.Y., O'Brien, T.E., O'Gorman, B., Ostby, E., Petukhov, A., Putterman, H., Quintana, C., Roushan, P., Rubin, N.C., Sank, D., Skolik, A., Smelyanskiy, V., Strain, D., Streif, M., Szalay, M., Vainsencher, A., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Zhou, L., Neven, H., Bacon, D., Lucero, E., Farhi, E., Babbush, R.: Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics* **17**(3), 332–336 (Feb 2021). <https://doi.org/10.1038/s41567-020-01105-y>, <http://dx.doi.org/10.1038/s41567-020-01105-y>
9. Hashim, A., Nguyen, L.B., Goss, N., Marinelli, B., Naik, R.K., Chistolini, T., Hines, J., Marceaux, J.P., Kim, Y., Gokhale, P., Tomesh, T., Chen, S., Jiang, L., Ferracin, S., Rudinger, K., Proctor, T., Young, K.C., Blume-Kohout, R., Siddiqi, I.: A practical introduction to benchmarking and characterization of quantum computers (2024), <https://arxiv.org/abs/2408.12064>
  10. Jain, S.: Solving the traveling salesman problem on the d-wave quantum computer. *Frontiers in Physics* **9** (2021). <https://doi.org/10.3389/fphy.2021.760783>, <https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2021.760783>
  11. Kittichaikoonkij, N., Pongtanyavichai, N., Suwananek, P., Chongstitvatana, P., Suksen, K.: Benchmarking quantum computing for combinatorial optimization. In: 2025 22nd International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). pp. 1–6 (2025). <https://doi.org/10.1109/ECTI-CON64996.2025.11100821>
  12. Mandal, A., Roy, A., Upadhyay, S., Ushijima-Mwesigwa, H.: Compressed quadratization of higher order binary optimization problems (2020), <https://arxiv.org/abs/2001.00658>
  13. Maurer, W., Scherzinger, S.: 1-2-3 reproducibility for quantum software experiments (2022), <https://arxiv.org/abs/2201.12031>
  14. Rajak, A., Suzuki, S., Dutta, A., Chakrabarti, B.K.: Quantum annealing: an overview. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **381**(2241) (dec 2022). <https://doi.org/10.1098/rsta.2021.0417>, <http://dx.doi.org/10.1098/rsta.2021.0417>
  15. Silevitch, D., Rosenbaum, T., Aeppli, G.: Quantum tunneling vs. thermal effects in experiments on adiabatic quantum computing. *The European Physical Journal Special Topics* **224**, 25–34 (2015). <https://doi.org/10.1140/epjst/e2015-02340-6>, <https://doi.org/10.1140/epjst/e2015-02340-6>
  16. Villar-Rodriguez, E., Osaba, E., Oregi, I.: Analyzing the behaviour of d'wave quantum annealer: fine-tuning parameterization and tests with restrictive hamiltonian formulations (2022), <https://arxiv.org/abs/2207.00253>

# GRACE: A Grover-Enhanced Actor-Critic Framework for Complex Robotic Manipulation

Eirini Panteli<sup>1,2</sup>[0009-0008-5747-8378], Paulo E. Santos<sup>1</sup>[0000-0001-8484-0354],  
Belinda Chiera<sup>2</sup>[0000-0001-9274-0943], and Josh Chopin<sup>2</sup>[0000-0002-5012-5827]

<sup>1</sup> PrioriAnalytica, Adelaide, SA, Australia

{eirini.panteli,paulo.santos}@priorianalytica.com

<sup>2</sup> University of South Australia, Adelaide, SA, Australia

{eirini.panteli,belinda.chiera,josh.chopin}@unisa.edu.au

**Abstract.** Autonomous satellite servicing presents unique challenges, including constrained onboard resources, uncertain dynamics, and the critical need for sample-efficient learning. Classical Reinforcement Learning (RL) methods, such as Proximal Policy Optimisation (PPO), often struggle under these conditions, exhibiting slow convergence and high sample complexity. This paper proposes GRACE (Grover-enhanced Reinforcement-learning Actor-Critic Engine), a hybrid quantum-classical RL framework designed to address the limitations of classical RL in high-precision, high-dimensional robotic manipulation tasks. GRACE introduces a Grover-enhanced amplitude amplification mechanism into an Actor-Critic architecture built on Variational Quantum Circuits. This integration leverages quantum properties such as entanglement and superposition to enhance policy expressiveness and exploration focus, while PPO ensures stable training. The framework selectively amplifies the probability amplitudes of favourable actions using Grover-enhanced logic embedded within the quantum policy circuits. Preliminary experiments on a simulated robotic satellite repair task, removing and manipulating components of the Hubble Space Telescope, show improved convergence speed, enhanced training stability, and higher sample efficiency compared to a classical PPO baseline, despite requiring fewer training episodes. These findings highlight the potential of quantum-enhanced RL in solving optimisation problems where classical approaches fall short, particularly in space robotics domains that demand fine-grained control under tight computational constraints. While current quantum hardware remains limited, the GRACE framework establishes a practical algorithmic foundation for future Quantum RL research in highly safety-critical, resource-constrained applications.

**Keywords:** Quantum Reinforcement Learning · Machine Learning · Classical Reinforcement Learning · Robotic Hand · Satellite Servicing.

## 1 Introduction

Autonomous robotic systems are critical for future space missions where communication delays, dynamic environments, and mission duration constraints make

continuous remote control infeasible [3, 19]. These intelligent robotic systems must operate independently, executing complex, multi-step manipulation tasks under tight resource and timing constraints [3, 19]. However, traditional control strategies, including classical Reinforcement Learning (RL), struggle in such conditions, exhibiting slow convergence, poor scalability, and high sample inefficiency in high-dimensional, stochastic environments [16].

To address the challenges of classical RL, we propose GRACE (Grover-enhanced Reinforcement-learning Actor-Critic Engine), a hybrid framework, with both quantum and classical counterparts, that integrates Grover-enhanced amplitude amplification into an Actor-Critic architecture built on Variational Quantum Circuits (VQCs). GRACE is designed to improve exploration efficiency and accelerate policy convergence by leveraging quantum principles to bias action selection toward more promising policies, all within the stability of Proximal Policy Optimisation (PPO).

At its core, RL is an optimisation process where agents iteratively update their policies to maximise cumulative reward [21]. Classical RL algorithms, such as Policy Gradient and Actor-Critic methods, perform this optimisation by interacting with the environment to estimate gradients and adjust policy parameters accordingly [21]. However, classical methods become increasingly inefficient as task complexity and the dimensionality of state-action spaces grow [16]. The GRACE framework addresses these limitations by embedding quantum mechanisms, specifically, trainable Grover-enhanced diffusion and oracle components, within the policy network, where the oracle functions without prior knowledge of which state is correct or optimal. These quantum components selectively amplify favourable action encodings, thereby improving both the efficiency of learning and the quality of decision-making in uncertain, high-dimensional environments.

Quantum Reinforcement Learning (QRL) leverages qubits, which can exist in superpositions and become entangled, enabling compact and parallel representations of policy information [11, 25]. Variational Quantum Circuits (VQCs) take advantage of these quantum properties to model expressive functions with fewer parameters than classical networks [15]. Grover’s algorithm, originally developed for quantum search tasks, provides a natural foundation for structured amplitude amplification techniques [13, 24]. In the GRACE framework, Grover-enhanced logic is employed to guide action selection without requiring oracle access, that is, without prior knowledge of the optimal solution.

This paper contributes the design and preliminary evaluation of the GRACE framework. While focused on robotic satellite servicing as a motivating use case, the GRACE framework is applicable to a broad class of resource-constrained environments that require efficient control, high reliability, and fast policy convergence. Early experiments using the AstroArm simulation environment demonstrate improved convergence speed, enhanced training stability, and reduced sample complexity compared to classical baselines, laying a foundation for future research into scalable and practical quantum-enhanced RL systems.

## 2 Technical Background

To facilitate understanding of the proposed GRACE framework, this section introduces key foundational concepts in quantum computing, reinforcement learning, and recent developments in Quantum Reinforcement Learning. Each concept is briefly explained with emphasis on its relevance to the design of GRACE.

### 2.1 Quantum Computing and Reinforcement Learning Concepts

Quantum computing differs fundamentally from classical computing by operating on qubits instead of classical bits [11, 25]. While classical bits are binary and can hold a value of either 0 or 1 at any given time, qubits can exist in a superposition of both states simultaneously. Such quantum superposition allows quantum systems to represent and evaluate multiple possibilities in parallel, greatly increasing computational efficiency for certain types of problems [11, 25]. Furthermore, qubits can become entangled, meaning the state of one qubit is intrinsically linked to the state of another, regardless of the physical distance between them. Quantum entanglement enables non-classical correlations that allow quantum algorithms to perform distributed, highly parallel computations in ways that are fundamentally impossible with classical systems [11, 25]. Together, superposition and entanglement form the basis of quantum parallelism, enabling powerful algorithmic primitives such as amplitude amplification and interference, tools that offer the potential to solve complex problems more efficiently than classical methods [18].

In light of the qubit state representation, qubits can exist in a superposition of both states simultaneously, therefore exponentially increasing the computing power [11, 25]. Moreover, as a result of the entanglement quantum phenomenon, quantum computers are capable of processing information in highly efficient and novel ways, unlocking unprecedented computational capabilities [11, 25]. In entangled systems, the state of one particle becomes dependent on the state of another, regardless of the distance separating them, allowing qubits to establish correlations that classical bits cannot achieve [11, 25]. Hence, through a series of quantum gates, quantum circuits are able to create entanglement, superpositions, perform computations, and execute quantum-enhanced algorithms to offer exponential state representations, enhanced expressiveness, and exponentially faster convergence than classical computers [11, 25].

Quantum circuits operate using unitary gates, such as  $R_x(\theta)$ ,  $R_y(\theta)$ ,  $R_z(\theta)$ , which perform rotations around the Bloch sphere axes, as well as entangling gates like the controlled-NOT (*CNOT*), which introduce quantum correlations between qubits [9, 25]. These operations enable expressive transformations on high-dimensional quantum states by manipulating the amplitudes and relative phases of superposed states in a reversible and coherent manner [9, 25]. A Variational Quantum Circuit (VQC) is a specific class of quantum circuit that includes parameterised gates, where the gate angles are treated as trainable parameters optimised during learning [9, 25]. These parameters define agent policy or value function when applied to reinforcement learning tasks [9, 25]. The VQC acts



as the core learnable component in hybrid quantum-classical models, enabling compact, flexible representations of complex mappings [9, 25]. The final stage of a VQC involves quantum measurement, which collapses the quantum state into a classical bitstring by projecting the probabilistic superposition onto a computational basis, typically  $|0\rangle$  or  $|1\rangle$ , thereby producing observable outputs for learning and decision-making [9, 25]. The quantum-classical hybrid approach enables the efficient computation of gradients through the quantum circuit, while simultaneously allowing the variational parameters to be updated using classical optimisation algorithms [9, 25].

Early development of Quantum Computing (QC) started with the Grover’s search algorithm, which is a foundational quantum search algorithm that provides a quadratic speedup over classical counterparts [13]. The key operations in Grover’s algorithm are amplitude selection and amplitude amplification. The process works by increasing the probability amplitudes of desired target states while suppressing others through constructive and destructive interference. The mechanism of Grover’s algorithm has inspired a range of quantum-enhanced strategies in optimisation and decision-making, including Quantum Reinforcement Learning [24].

Grover-enhanced logic in GRACE refers to a modified form of Grover’s algorithm adapted for quantum learning circuits. Instead of performing a full quantum search, the adapted algorithm selectively amplifies favourable quantum encodings, such as actions, using trainable selection and diffusion components. These components are parameterised and integrated within variational quantum layers to enhance the likelihood of selecting high-reward actions in reinforcement learning tasks. Unlike the original Grover algorithm, which is used for searching unstructured databases, Grover-enhanced logic in GRACE is embedded within the actor and critic circuits to improve exploration bias during training.

Reinforcement Learning is a learning paradigm where an agent interacts with an environment to maximise cumulative rewards over time. The decision-making process is formalised as a Markov Decision Process, aiming to find an optimal policy that maps states to actions [21]. Within RL, the optimisation objective is to adjust the parameters of a policy or value function to maximise expected return. Actor-Critic methods decompose the learning task into two parts: the actor selects actions based on the current policy, and the critic estimates the value of states or actions to inform policy updates. Proximal Policy Optimisation (PPO) is a policy-gradient algorithm that improves training stability through clipped updates and advantage-weighted gradient estimates [23]. In high-dimensional or resource-constrained environments such as robotics in space, RL faces challenges due to large state-action spaces, delayed convergence, and inefficient exploration strategies [6, 26]. These challenges provide strong motivation for hybrid quantum-classical approaches like GRACE, which exploit the expressiveness of quantum circuits and Grover-enhanced amplitude amplification to improve sample efficiency, policy convergence, and learning robustness in demanding environments.

## 2.2 Prior Work in Quantum Reinforcement Learning

Quantum Reinforcement Learning combines quantum computing principles with reinforcement learning techniques to enable more efficient exploration, policy representation, and convergence [16]. QRL can represent complex policies with fewer parameters using VQCs, which include parameterised gates and entanglement layers [10, 15]. Measurement collapses the probabilistic quantum state into classical outputs suitable for learning and control.

Recent research in QRL focuses on employing VQCs as function approximators, leading to the development of QRL variants. A study in 2022 presented two frameworks for deep QRL tasks using gradient-free evolutionary optimisation [4]. The amplitude encoding scheme was first applied to the Cart-Pole problem, a classic control task where the objective is to balance a pole upright on a moving cart, demonstrating the quantum advantage of parameter saving using amplitude encoding [4]. A hybrid framework was later proposed where the QRL agent was equipped with a hybrid Tensor Network-VQC (TN-VQC) architecture to handle inputs of dimensions exceeding the number of qubits, allowing the performance of QRL in the MiniGrid environment with 147-dimensional inputs [4]. The hybrid TN-VQC framework offers an intuitive and effective method for compressing input dimensions, making the architecture well-suited for advancing QRL on Noisy Intermediate-Scale Quantum devices [4]. In another paper [5], a framework for analysing the performance of QRL via policy iteration on OpenAI Gym environment was studied. Key to the study was building quantum states that approximately encode the value function of a policy  $\pi$  to construct quantum policy evaluation methods for infinite-horizon discounted problems, as well as quantum policy improvement methods by post-processing measurement outcomes on the quantum states [5].

Additionally, the application of Quantum Reinforcement Learning in energy-efficient scenarios was actively explored, where both the benefits and limitations of QRL are examined in contexts such as HVAC (Heating, Ventilation, Air-Conditioning) control, electric vehicle energy management, and building energy optimisation [1]. As a testbed, the authors utilised existing energy-efficiency-focused RL simulators to compare classical RL algorithms with the proposed quantum-based approach. The results, across use cases like HVAC systems, electric vehicle fuel consumption, and profit optimisation for electrical charging stations, indicated that the quantum solution outperforms classical RL methods. Their proposed approach achieved higher cumulative rewards while requiring fewer parameters to be learned [1].

A recent paper, focusing on the space and satellite domain, proposed and highlighted the potential of QRL for coordinated mobility and satellite systems [14]. The appeal of QRL in the space context stems from its unique advantages including faster convergence rates and high scalability. The paper demonstrates that QRL can effectively address several limitations of conventional reinforcement learning, particularly in environments characterised by large and continuous action spaces, high-dimensional state representations, and the need for real-time decision-making under uncertainty. Furthermore, the study pro-

poses the use of multi-agent QRL strategies to support a variety of mobility system applications [14].

In light of the growing research supporting QRL in complex and constrained settings, its application to in-orbit robotic-hand satellite manipulation is a natural extension. Robotic manipulation tasks in orbital settings are characterised by uncertain dynamics, complex spatial constraints, and limited onboard computational resources, making them ideal candidates for quantum-enhanced RL models such as GRACE [1, 12, 16].

### 3 Motivation for QRL in Resource-Constrained Systems

Classical RL methods, including Deep RL variants, have achieved impressive results across numerous domains [17]. However, when deployed in real-world systems that operate under strict memory, power, or computational constraints, such as space-robotic manipulators, classical approaches encounter obstacles. In such environments, the size of the state and action spaces increases rapidly with task complexity, leading to inefficient exploration and limited generalisation [17]. Such limitations are especially critical in precision-dependent, multi-step manipulation tasks, where poor sample efficiency and slow convergence can render classical RL approaches impractical [17].

Moreover, classical RL agents are fundamentally limited by the sequential nature of classical hardware [17]. Even with the use of deep networks, policy updates rely on gradient descent over large batches of experience, requiring high computational power and memory access [17]. In space contexts, where on-board computing is tightly constrained and real-time decisions are required, these demands create a bottleneck classical architectures struggle to overcome [17].

Quantum Reinforcement Learning offers a potentially transformative alternative [1]. Quantum circuits allow for the parallel encoding and manipulation of multiple solution candidates via superposition, while entanglement enables compact representation of complex dependencies between input variables [1, 12, 16]. Together, these properties offer the prospect of faster policy search, more efficient exploration, and scalable value propagation over high-dimensional spaces [25]. In particular, quantum amplitude amplification techniques can bias sampling toward high-reward actions earlier in training, something classical agents typically require many episodes to discover through trial and error [25].

Despite these advantages, most existing QRL approaches remain in early stages, often limited to discrete action spaces, shallow environments, or minimal circuit depths due to current hardware constraints [25]. Furthermore, few architectures explicitly address the challenge of continuous control or the integration of quantum search mechanisms into policy improvement routines [25]. These limitations strongly motivate the development of a more robust, scalable, and expressive QRL agents, designed from the ground up to tackle control problems in high-stakes, real-time domains.

To this end, we propose the GRACE framework, a Grover-enhanced Reinforcement-learning Actor-Critic Engine. As discussed in Section 2, GRACE

integrates Grover-enhanced amplitude amplification within a hybrid Actor-Critic architecture, powered by Variational Quantum Circuits and Proximal Policy Optimisation. By uniting quantum search logic with robust policy gradient training, GRACE is designed to accelerate convergence and improve action selection in high-dimensional continuous control settings. The next section presents the architecture in detail.

#### 4 Proposed Framework: GRACE – Grover-enhanced Reinforcement-learning Actor-Critic Engine

Classical RL agents, including deep and actor-critic variants, often face limitations in high-dimensional, resource-constrained environments, such as space robotics, due to slow convergence, inefficient exploration, and the inability to scale efficiently across large state-action spaces [17]. Section 3 outlined how quantum mechanisms such as superposition, entanglement, and amplitude amplification can address these challenges. Building upon this, we present the GRACE framework—Grover-enhanced Reinforcement-learning Actor-Critic Engine, which integrates Grover-enhanced logic into a hybrid quantum-classical Actor-Critic architecture trained using Proximal Policy Optimisation.

GRACE introduces a novel synthesis of quantum search mechanisms with classical reinforcement learning policy training. Specifically, the framework embeds Grover-enhanced amplitude amplification within both the Actor and Critic quantum circuits, using Variational Quantum Circuits and PPO to perform quantum-enhanced function approximation. The goal is to guide action selection and value estimation more effectively by amplifying desirable solution paths in the quantum latent space.

The GRACE pipeline, as illustrated in Figure 1, is as follows:

- **Input:** The agent receives an observation of the current environment state, including joint angles and end-effector position, from the AstroArm simulation environment (Figure 1, left).
- **Actor circuit:** The input state is encoded using *Angle Embedding*, which transforms classical inputs into quantum rotation gate parameters and is passed through a parameterised VQC (Figure 1, upper-central block), composed of strongly entangling layers. The encoded quantum state is then processed by a Grover-enhanced oracle and diffusion block that amplifies favourable action amplitudes. The resulting quantum state is measured, producing outputs that are linearly mapped to action logits, which are real-valued scores that indicate the model’s relative preference for each possible action before applying a sampling step. A Normal distribution is then sampled from these logits to produce an action. The selected action is then passed back to the simulation environment, where the action is executed and the corresponding reward is computed.
- **Critic circuit:** The same input quantum state is sent through a structurally similar quantum circuit (Figure 1, lower-central block). Using the same angle

embedding and Grover-enhanced amplification, the critic estimates the scalar value of the current state under the policy.

- **Policy Update:** The value output is used to compute the advantage function. PPO calculates a clipped objective that balances policy improvement and stability. The full loss includes policy loss, value loss, and entropy regularisation. Gradients are propagated using PennyLane’s automatic differentiation engine with adjoint methods (Figure 1, right).

The GRACE architecture offers several advantages: the Grover-enhanced blocks amplify quantum states corresponding to high-reward actions, improving the signal for policy optimisation; entanglement layers support richer policy modelling than shallow classical architectures; and quantum superposition allows parallel exploration in the policy search space. By embedding amplitude amplification into the training loop, GRACE biases updates toward desirable outcomes more efficiently than classical counterparts.

Although Grover’s original algorithm was designed for database search, the GRACE framework adapts its core idea, namely amplifying solution amplitudes, into a differentiable, trainable structure suited for reinforcement learning. Unlike prior QRL frameworks, GRACE applies this amplification not only to the policy network but also to the critic, enabling coherent quantum-enhanced estimation of both action and value functions.

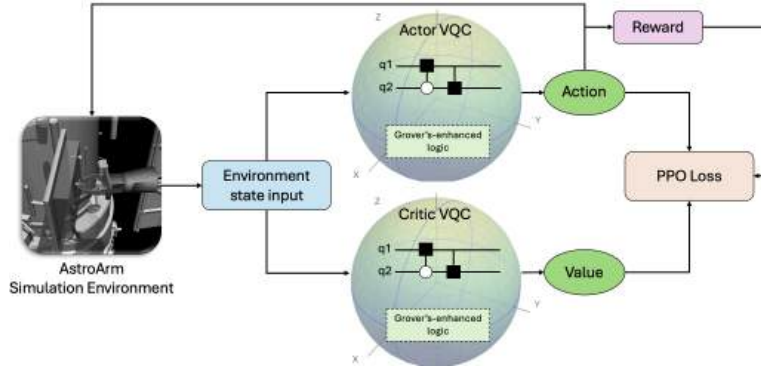


Fig. 1: GRACE framework architecture: a hybrid quantum-classical Actor-Critic model integrating VQC with Grover-enhanced amplitude amplification for enhanced policy representation and exploration.

Figure 1 highlights the bidirectional flow of information between the environment, Actor and Critic quantum circuits, and the PPO loss computation. While the full theoretical guarantee of quantum advantage requires robust quantum error correction, GRACE is designed with near-term, hybrid execution in mind and can be evaluated using simulators such as *lightning.qubit*.

## 5 Use Case: Training a Dexterous Robotic Hand for Satellite Repair

Building on the previously established work in [20], all experiments were conducted in the AstroArm simulation environment, which includes a 3D model of the Hubble Space Telescope (HST), the Science Instrument Control and Data Handling (SI C&DH) unit, and a dexterous robotic hand. The HST was chosen for its modular design featuring Orbital Replaceable Units (ORUs), which include instruments and batteries accessible via mechanical interfaces such as doors and fasteners [20].

The SI C&DH unit was selected as the manipulation target, following NASA’s EVA replacement sequence [20]. The documented sequence offers a realistic benchmark for simulating precise robotic interaction in a space-based context [20]. Integrating the SI C&DH replacement procedure into the AstroArm environment enables the evaluation of quantum-enhanced control architectures in a physically grounded and operationally relevant scenario [20].

Figure 2a shows the simulation setup with the satellite, robotic hand, and door assembly [20]. Figures 2b and 2c provide close-ups of the SI unit, which includes a handle, screw, nut, and the robotic hand. The hand features two grippers and eight degrees of freedom (joints) for manipulation and movement [20]. Nonessential joints are deactivated based on task requirements to reduce action space and improve training efficiency [20].

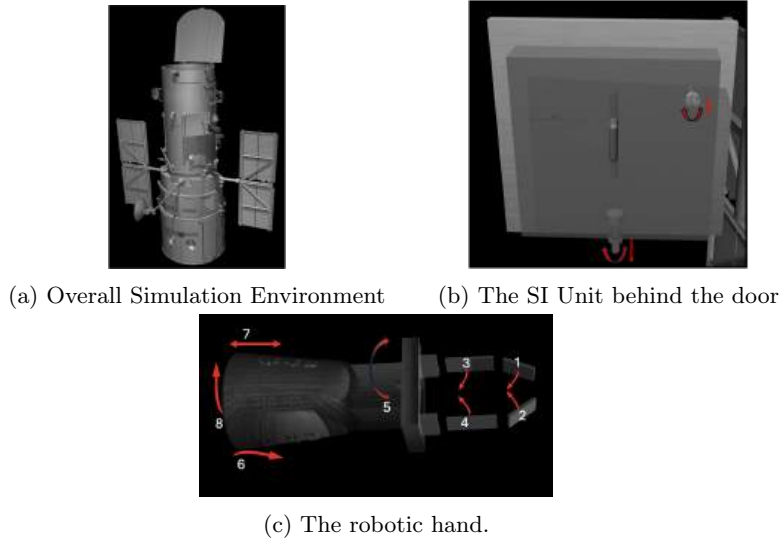


Fig. 2: Components of AstroArm simulation environment

To enable autonomous manipulation of the SI C&DH components, an RL policy was trained to control the robotic hand in the simulation environment [20]. The policy observed joint positions, end-effector pose, and task-relevant feed-

back, and outputs continuous motor actions to operate the grippers and joints. The goal was to learn precise movement sequences to complete various manipulation tasks within geometric and action space constraints [20]. The control problem was framed within a classical Actor-Critic RL architecture, where the Actor selected actions and the Critic estimated state values to stabilise updates. PPO was used for its robustness in high-dimensional continuous control and its clipped update mechanism [20]. However, the complexity of spatial interactions and the need for fine-grained precision pose challenges for classical RL Actor-Critic PPO, motivating the use of quantum-enhanced methods [20].

The proposed GRACE framework, which integrates Grover-enhanced logic into Actor-Critic VQCs within a PPO setting, is trained within the AstroArm simulation environment to carry out the task of SI\_Unit removal. The objective is to enable and enhance the robotic hand to learn a sequence of precise manipulations required to detach the SI\_Unit components, such as the door, screw, and nut, under the complex spatial constraints of the satellite assembly, by leveraging the benefits of the quantum model. Once the training is complete, the performance of the quantum-enhanced model is systematically evaluated and compared against that of a classical Actor-Critic PPO baseline, focusing on key metrics such as task success rate, convergence speed, and sample efficiency. This comparison aims to assess the potential advantages of QRL in domains requiring fine-grained control and efficient policy learning.

## 6 Preliminary Results

To assess the viability of the GRACE framework, a set of preliminary experiments was performed using the AstroArm simulation environment, focusing on the specific task of opening the door of the SI C&DH unit. These experiments aim to explore whether Grover-enhanced amplitude amplification improves action selection and learning efficiency when integrated into a hybrid Actor-Critic architecture with PPO optimisation. The following results offer early-stage insights into the behaviour and training dynamics of the GRACE agent, even under constraints such as limited circuit depth and a small number of qubits.

As illustrated in Figures 3a and 3b, the quantum-enhanced agent demonstrates a more stable training process and higher peak rewards compared to the classical baseline. In the classical agent training curve (Figure 3a), reward values mostly remain below 8 and exhibit long stretches of stagnation, particularly in the middle episodes. In contrast, the GRACE agent (Figure 3b) maintains higher variability and more frequent high-reward spikes throughout the training phase, even though the agent was only trained for 100 episodes, compared to 250 episodes for the classical model, suggesting improved sample efficiency and potentially faster convergence.

Additional insight is offered by the test-phase results shown in Figures 3c and 3d. Across 10 test episodes, the classical agent completed 3 tasks successfully, while the quantum agent completed 4. Although the difference is small in absolute terms, the improved performance was achieved with reduced training

time, suggesting that the quantum-enhanced model may achieve comparable or better generalisation using fewer computational resources.

Overall, the early results indicate that the Grover-enhanced amplitude amplification mechanism embedded within GRACE may support more focused exploration, helping the agent discover high-reward trajectories more efficiently. Additionally, the reduced training duration implies potential gains in sample efficiency, which is crucial for robotics scenarios where simulation or real-world interaction is expensive. While further evaluation is necessary to establish robustness across environments and hyperparameter settings, these findings serve as an encouraging starting point for future investigation into quantum-classical RL architectures.

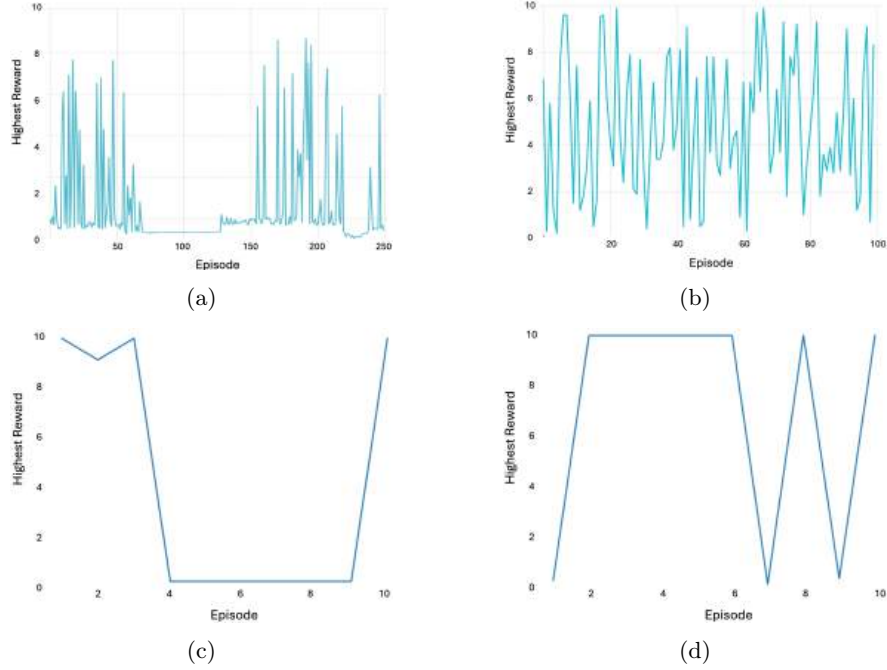


Fig. 3: Training and testing performance for the SI C&DH door-opening task in AstroArm. (a) Classical training, (b) GRACE training, (c) Classical testing, (d) GRACE testing. The GRACE agent was trained for 100 episodes, compared to 250 for the classical agent.

## 7 Position Justification and Assumptions

The proposed GRACE framework introduces a Quantum-enhanced Actor-Critic architecture that incorporates Grover-enhanced Variational Quantum Circuits to strengthen exploration and sample efficiency in Reinforcement Learning. While



classical RL algorithms, such as Proximal Policy Optimisation, have achieved state-of-the-art performance across a number of domains, such methods continue to face challenges related to suboptimal exploration and local policy convergence, particularly in high-dimensional robotic manipulation tasks with complex dynamics [22]. By embedding quantum amplitude amplification into the policy circuit, our approach aims to bias the agent toward high-reward actions by amplifying their associated quantum amplitudes. The targeted selection process enhances exploration efficiency and helps guide the policy updates more effectively, which can lead to faster convergence and increased robustness in high-dimensional or stochastic environments.

However, acknowledging several valid and anticipated concerns is important at this stage. Important limitations of quantum computing include the number of qubits, noise, errors, and decoherence/coherence times, indicating that current devices struggle to maintain stable quantum states over long periods, or execute interference-free operations [7]. Moreover, prior work highlights that quantum systems can suffer from poor error correction methods and limited inter-qubit connectivity, both of which affect scalability [7]. Other persistent problems, such as limited qubit fidelity, state preparation and efficiency in the application of quantum logic gates, also show that the manipulation of quantum information can still be inaccurate [7]. Lastly, operational limitations include restricted access to quantum devices, caused by high demand and limited availability, which slows down the research progress [7].

We acknowledge these concerns and limitations as valid within the context of today’s hardware constraints. Current quantum devices remain noisy, limited in qubit count, and difficult to scale for practical reinforcement learning applications [7]. However, our position is grounded in a long-term vision rather than a short-term deployment claim. The advocated GRACE framework is developed as a forward-looking architecture designed to explore how hybrid quantum-classical learning systems may benefit from quantum expressivity, parallelism, and amplitude-based optimisation in high-dimensional control spaces once the supporting hardware matures. By proactively addressing architectural design now, GRACE lays the foundation for future RL systems capable of operating in complex, resource-constrained environments, such as robotics, edge computing, and satellite space-based missions, where classical methods struggle to scale, enabling real-time quantum-enhanced decision-making for challenging operations, such as autonomous satellite servicing.

Simulation results conducted through software framework such as PennyLane [2] and Qiskit [20] demonstrate the feasibility of not only quantum algorithms, such as hybrid Variational Quantum Circuits, but also the incorporation of Grover-enhanced logic [2, 12]. Moreover, hybrid quantum-classical methods, as advocated in this work, are specifically designed to mitigate the shortcomings of current QPUs by switching between classical processors (CPUs or GPUs) for pre-processing and postprocessing, and quantum processors (QPUs) for components where entanglement and quantum representation offer a unique advantage [2, 12].

Prior studies have demonstrated that VQCs can be effectively scaled to sim-

ple robotic control tasks, yielding competitive results in terms of policy learning and sample efficiency [10]. Building on this foundation, our preliminary experiments show that incorporating amplitude amplification mechanisms, such as Grover’s-enhanced logic, can further enhance action selection in more complex, high-dimensional environments. The observed improvement is particularly notable in tasks where precise action selection and exploration are critical for achieving stable learning performance and successful task completion.

The position adopted here is based on the assumption that practical access to quantum devices will become increasingly feasible in the near future, driven by ongoing advancements in both academic research and industry [8]. We do not claim that quantum advantage is achievable today, rather, we argue that foundational research in architecture and algorithm design must precede the arrival of mature, large-scale quantum hardware. By investigating QRL frameworks now, we aim to position the RL community to take full advantage of emerging quantum capabilities as those technologies become practically viable. Such a forward-looking strategy ensures that future applications are not constrained by a lack of algorithmic readiness when quantum technologies reach sufficient scale, reliability, and integration into real-world systems.

## 8 Conclusion

This paper proposed a paradigm shift in RL for high-precision space robotics via the GRACE framework, a hybrid quantum-classical architecture. By integrating Grover-enhanced amplitude amplification within an Actor-Critic architecture powered by Variational Quantum Circuits and Proximal Policy Optimisation, the proposed approach aims to overcome core limitations of classical RL methods in high-dimensional, resource-constrained, and uncertain environments.

Preliminary results, obtained in a simulated robotic satellite repair scenario, suggest that the quantum-enhanced agent demonstrates improved training stability, faster convergence, and better sample efficiency relative to its classical counterpart, even when trained with fewer episodes. The early findings support the hypothesis that quantum policy representations, especially those incorporating structured amplitude amplification, can yield more targeted and effective exploration in RL.

We acknowledge the hardware limitations inherent in the quantum devices of today, including decoherence, gate depth constraints, and limited qubit fidelity. Nonetheless, this work is situated within a long-term vision, to proactively establish the architectural and algorithmic foundations required to unlock the full potential of quantum RL as QPUs continue to mature.

Ultimately, the present study calls for expanded research into QRL frameworks in safety-critical, high-dimensional domains such as space robotics. The compounded challenges of real-time decision-making, constrained onboard computation, and the need for highly sample-efficient learning make space robotics and similar applications not only suitable testbeds but also urgent and strategically important use cases for advancing quantum-enhanced learning systems.

[6, 26]. Since RL is inherently an optimisation problem, where the agent seeks to maximise expected cumulative reward through iterative policy improvement [21], enhancing the learning and optimisation cycle with quantum mechanisms presents a natural and promising direction. The GRACE framework offers a compelling and forward-looking path for such advancements.

## References

1. Andrés, E., Cuéllar, M.P., Navarro, G.: On the use of quantum reinforcement learning in energy-efficiency scenarios. *Energies* **15**(16), 6034 (2022)
2. Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Alam, M.S., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., et al.: PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968* (2018)
3. Castano, R., Vaquero, T., Rossi, F., Verma, V., Van Wyk, E., Allard, D., Huffmann, B., Murphy, E.M., Dhamani, N., Hewitt, R.A., et al.: Operations for autonomous spacecraft. In: 2022 IEEE Aerospace Conference (AERO). pp. 1–20. IEEE (2022)
4. Chen, S.Y.C., Huang, C.M., Hsing, C.W., Goan, H.S., Kao, Y.J.: Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology* **3**(1), 015025 (2022)
5. Cherrat, E.A., Kerenidis, I., Prakash, A.: Quantum reinforcement learning via policy iteration. *Quantum Machine Intelligence* **5**(2), 30 (2023)
6. Crespo Márquez, A.: The curse of dimensionality. In: *Digital Maintenance Management: Guiding Digital Transformation in Maintenance*, pp. 67–86. Springer (2022)
7. Desdentado, E., Calero, C., Moraga, M., García, F.: Quantum computing software solutions, technologies, evaluation and limitations: a systematic mapping study. *Computing* **107**(5), 1–60 (2025)
8. Gill, S.S., Cetinkaya, O., Marrone, S., Claudino, D., Haunschild, D., Schlote, L., Wu, H., Ottaviani, C., Liu, X., Machupalli, S.P., et al.: Quantum computing: Vision and challenges. In: *Quantum Computing*, pp. 19–42. Elsevier (2025)
9. Gong, L.H., Pei, J.J., Zhang, T.F., Zhou, N.R.: Quantum convolutional neural network based on variational quantum circuits. *Optics Communications* **550**, 129993 (2024)
10. Hohenfeld, H., Heimann, D., Wiebe, F., Kirchner, F.: Quantum deep reinforcement learning for robot navigation tasks. *IEEE Access* (2024)
11. Hughes, C., Isaacson, J., Perry, A., Sun, R.F., Turner, J.: What is a qubit? In: *Quantum computing for the quantum curious*, pp. 7–16. Springer (2021)
12. Kanazawa, N., Egger, D.J., Ben-Haim, Y., Zhang, H., Shanks, W.E., Aleksandrowicz, G., Wood, C.J.: Qiskit experiments: A python package to characterize and calibrate quantum computers. *Journal of Open Source Software* **8**(84), 5329 (2023)
13. Khanal, B., Rivas, P., Orduz, J., Zhakubayev, A.: Quantum machine learning: A case study of grover’s algorithm. In: 2021 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 79–84. IEEE (2021)
14. Kim, G.S., Chen, S.Y.C., Park, S., Kim, J.: Quantum reinforcement learning for coordinated satellite systems. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
15. Kim, J.: Quantum reinforcement learning: Concepts, models, and applications. In: *International Conference on Intelligence of Things*. pp. 3–11. Springer (2024)

16. Kölle, M., Hgog, M., Ritz, F., Altmann, P., Zorn, M., Stein, J., Linnhoff-Popien, C.: Quantum advantage actor-critic for reinforcement learning. arXiv preprint arXiv:2401.07043 (2024)
17. Li, S.E.: Deep reinforcement learning. In: Reinforcement learning for sequential decision and optimal control, pp. 365–402. Springer (2023)
18. Markidis, S.: What is quantum parallelism, anyhow? In: ISC High Performance 2024 Research Paper Proceedings (39th International Conference). pp. 1–12. Prometheus GmbH (2024)
19. Nesnas, I.A., Fesq, L.M., Volpe, R.A.: Autonomy for space robots: Past, present, and future. *Current Robotics Reports* **2**(3), 251–263 (2021)
20. Panteli, E., Chopin, J., Chiera, B.: Astroarm: Robotic hand simulation environment for satellite servicing. In: Proceedings of AusDM24 (2024), to appear
21. Powell, K.M., Machalek, D., Quah, T.: Real-time optimization using reinforcement learning. *Computers & Chemical Engineering* **143**, 107077 (2020)
22. Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., Levine, S.: Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. arXiv preprint arXiv:1709.10087 (2017)
23. Rivera, A., Stewart, A.: Study of spacecraft deployables failures. In: 19th European Space Mechanisms and Tribology Symposium (ESMATS) 2021 (2021)
24. Singleton Jr, R.L., Rogers, M.L., Ostby, D.L.: Grover’s algorithm with diffusion and amplitude steering. arXiv preprint arXiv:2110.11163 (2021)
25. Taghandiki, K.: Quantum machine learning unveiled: A comprehensive review. *Journal of Engineering and Applied Research* **1**(2), 29–48 (2024)
26. Zhang, T., Mo, H.: Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems* **18**(3), 17298814211007305 (2021)