Quantum Optimization for HPC Cluster Routing: QUBO and LR-QAOA Approach

Rasikh Khan Cheema[2], Sneha Esther Cherian[2], Mohamed Mahmoud Helmy[2], Jeeva Joseph[2], Malmadee S S K Liyanage[2], Asad Raza[2], Muhammad Umair Shaukat[2], Temirlan Yeslamov[2], Stefan Raimund Maschek[1], Erik Schulze[1]
Bull Germany GmbH[1]
Deggendorf Institute of Technology, Germany[2]

Abstract

As **HPC clusters** scale, finding **optimal paths** for thousands of cables while avoiding collisions is a massive mathematical hurdle. We address this using a **Quadratic Unconstrained Binary Optimization (QUBO)** formulation to map routing constraints into a format compatible with the **Quantum Approximate Optimization Algorithm (QAOA)**. This approach aims to identify valid routing paths by prioritizing the shortest cable lengths. We show a Proof of Concept that this method could potentially solve complex wiring challenges for large-scale problem sizes.

Methodology

We map the HPC cluster into a network graph, encoding wiring rules and penalties into a QUBO formulation to **penalize invalid paths**. This problem is addressed using both **Variational QAOA** and **Linear-Ramp QAOA (LR-QAOA)** to identify the shortest valid cable routes. We then compare these two methods, evaluating how each approach handles the solution accuracy and the computational complexity of the routing task.

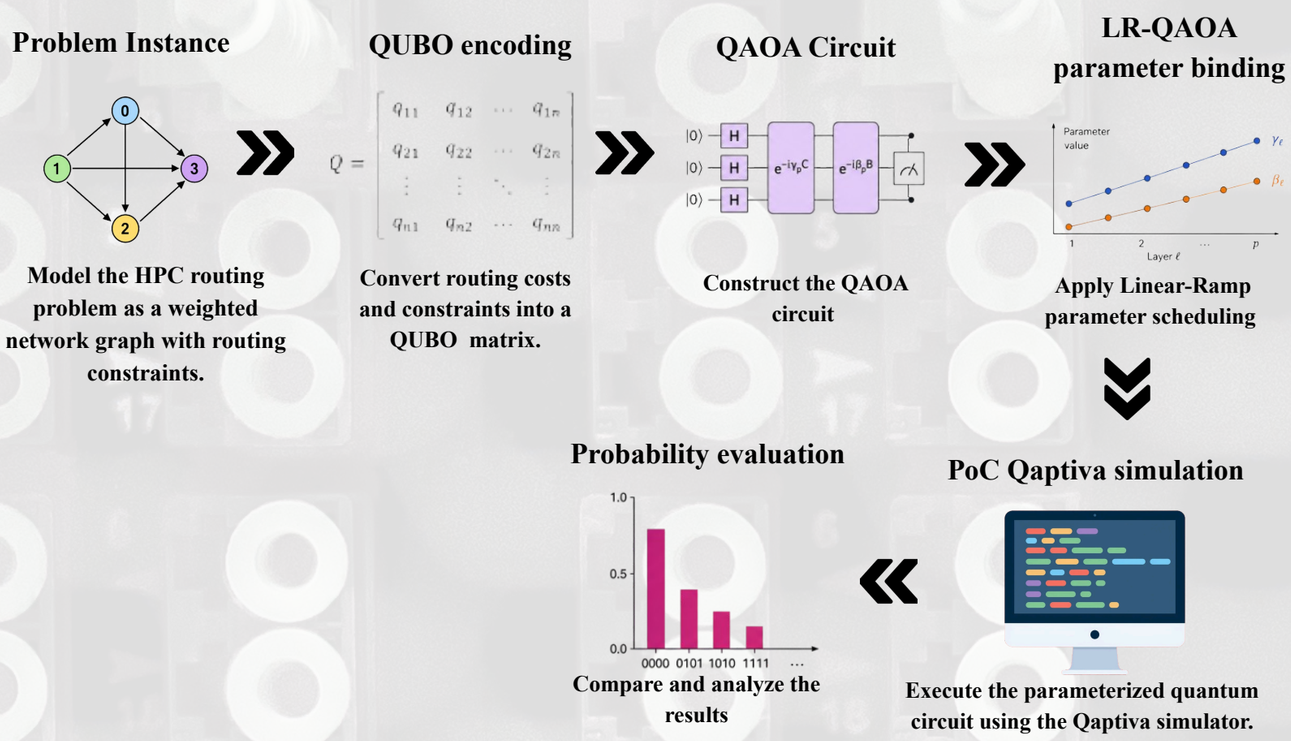

Fig. 1: QUBO-Based Multi-Path Routing Optimization Workflow

QUBO Formulation

Single-Path QUBO Hamiltonian

$$H(x) = \sum_{i \in E} c_i x_i + P_{path} \sum_{n \in V} \left(\sum_{i \in out(n)} x_i - \sum_{i \in in(n)} x_i - b_n \right)^2$$

Multi-Path QUBO Hamiltonian

$$H_{total} = \sum_{p=1}^{N} H(x_p) + P_{shared} \sum_{i \in E} \sum_{p_1 < p_2} (x_{i,p_1} \cdot x_{i,p_2})$$

- x_i: Binary edge variable
- $x_{i,p}$: Binary variable for edge i in path p
- c_i : Routing cost (weight) of edge i
- b_n: Path constraint at node n (+1 Source, -1 Sink, 0 intermediate)
- E: Set of all edges in the network
- V: Set of all nodes in the network

Fig. 2 shows the brute-force evaluation used to validate the QUBO formulation. Valid and invalid configurations are separated by their energy values, and the optimal valid path achieves the lowest QUBO energy together with the minimum cable length.

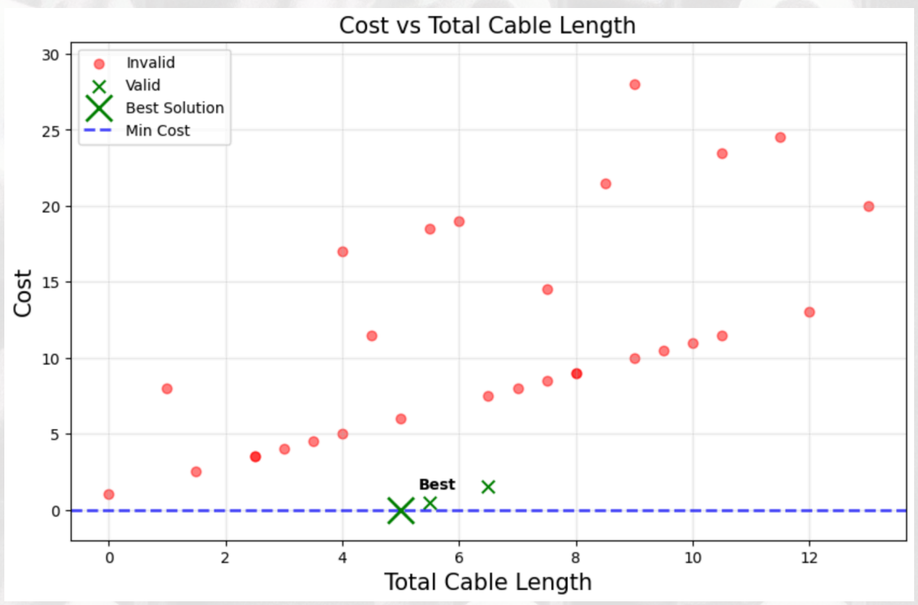

Fig. 2: Analysis of Valid and Invalid Path Costs Via Brute Force Method

Our formulation aims to handle multi-path routing by introducing a shared-edge-penalty whenever multiple paths attempt to use the same network edge simultaneously.

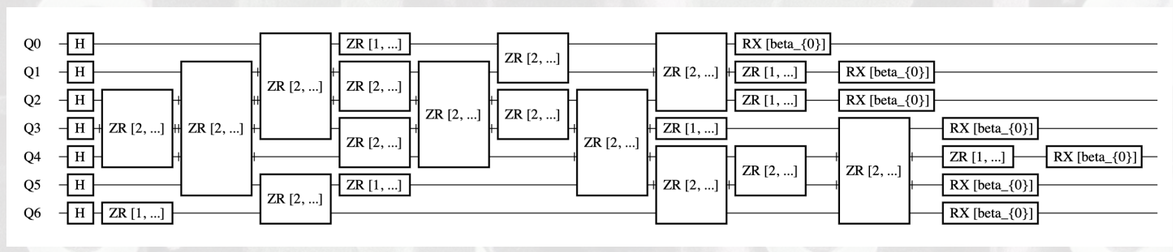

Fig. 3: Single-layer QAOA Quantum Circuit for Optimization

In Fig. 3 you see a single-layer QAOA circuit with 7 qubits. Hadamard (H) gates create a superposition over all possible routing paths. Z-rotation blocks encode QUBO routing costs and constraints, while RX(β0) mixer gates explore different routing configurations. The circuit alternates between cost and mixer operations in the QAOA and contains a total of 35 quantum gates.

Discussion

1. **Results of Variational and LR-QAOA** - The following graphs represent the routing solution probabilities obtained using Variational QAOA and LR-QAOA with 6 layers.

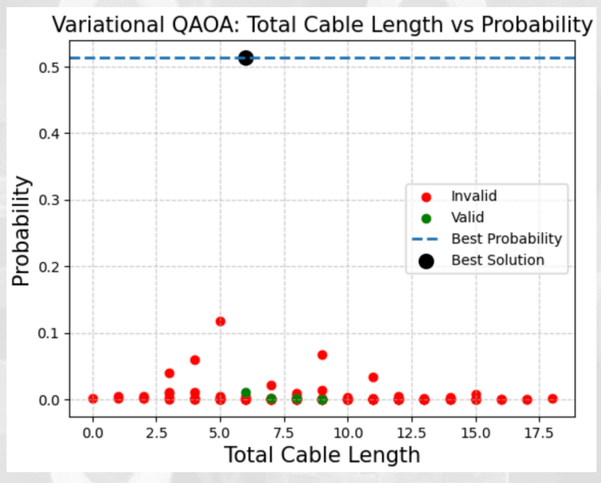

Fig. 4: Variational QAOA: Total cable length vs Probability

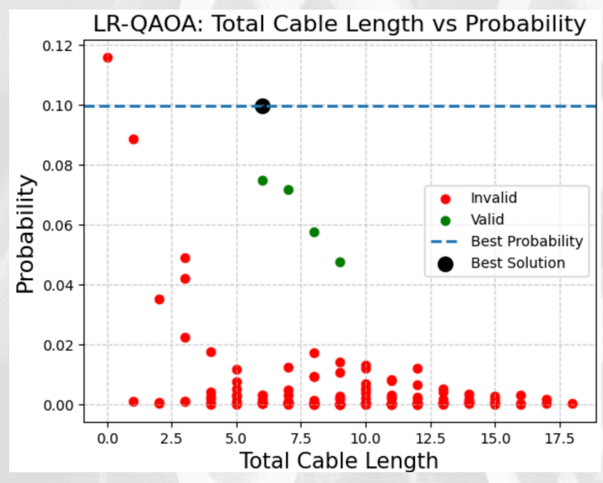

Fig. 5: LR-QAOA: Total cable length vs Probability

2. While the **Variational approach can reach higher success rates**, its results are **inconsistent** across different runs. We switched to the **Linear-Ramp** method because it is **deterministic**, providing **stable and repeatable** results every time the algorithm is executed.

3. **Depth vs. Probability Analysis** - Increasing the circuit depth helps the algorithm pick out the best route, making the shortest valid path more likely to be found as the quantum calculation becomes more precise.

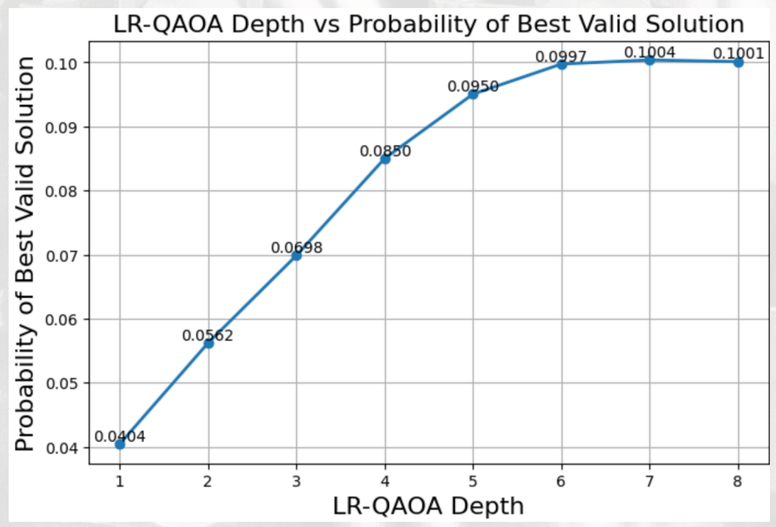

Fig. 6: LR-QAOA Depth vs Best Solution Probability

Conclusion

This study demonstrates that quantum methods can prioritize shortest-path routes, though the **instability of Variational QAOA** highlights the need for **deterministic** approaches like **Linear-Ramp** to ensure **reliable** performance. Future improvements, such as **error mitigation** and **increased circuit depth**, are necessary to bridge the gap between **stability** and **high success probability**. Scaling these refinements is critical for high-stakes applications like **autonomous traffic, train scheduling, or smart grids**, where repeatable, reliable configurations are essential to prevent system failure.

References