International Journal of Quantum Technologies
ISSN: 3068-9090

PRIME OPEN ACCESS

Volume 2, Issue 2

Research Article

Date of Submission: 25 May, 2026
Date of Acceptance: 18 June, 2026
Date of Publication: 02 july, 2026

QAOA Parameter Transfer for Quantum Optimization in Autonomous Mobility

Jeremy Mange* and Paramsothy Jayakumar

US Army Ground Vehicle Systems Center, 6501 E. 11 Mile Rd., Warren, 48397, MI, USA

***Corresponding Author:**
Jeremy Mange, US Army Ground Vehicle Systems Center, 6501 E. 11 Mile Rd., Warren, 48397, MI, USA.

Citation: Mange, J., Jayakumar, P., (2026). QAOA Parameter Transfer for Quantum Optimization in Autonomous Mobility. *Int J Quantum Technol, 2*(2), 01-09.

Abstract
This paper presents an approach for solving complex optimization problems in autonomous mobility through the transfer of optimal parameters in the Quantum Approximate Optimization Algorithm (QAOA) from small to large problem instances. Autonomous mobility problems, such as path planning and resource assignment, can often be formulated as graph problems, enabling the application of QAOA and parameter transfer. We focus on the Max-Cut formulation as a broadly applicable proxy for these graph-based tasks, and demonstrate that QAOA parameters optimized for small graphs can be effectively used for execution on larger graphs with similar graph structures, achieving high-quality solutions with significantly reduced computational effort. Experiments are conducted on both high-performance computing simulators and real quantum hardware (Rigetti Ankaa-2), with graphs ranging from 10 to 80 vertices. Results show that transferred parameters achieve approximation ratios within 3% of full QAOA optimization on average, and in some cases exceed it, confirming the practical viability of this approach for autonomous systems.

Keywords: Quantum Computing, Autonomous Mobility, Qaoa, Quantum Algorithms, Parameter Transfer, Max-Cut

Introduction
Quantum computing offers new paradigms for solving problems that are computationally intractable for classical computers. In recent years, there has been a surge in research exploring the potential of quantum algorithms for combinatorial optimization problems, especially very difficult problem classes (e.g., NP-hard). Variational Quantum Algorithms (VQAs), which leverage both quantum and classical computational resources, have emerged as a particularly promising approach in the Noisy Intermediate-Scale Quantum (NISQ) era, where fully fault-tolerant quantum computers are not yet available.

One of the most widely studied VQAs is the Quantum Approximate Optimization Algorithm (QAOA), introduced by Farhi et al. in 2014 [1]. QAOA is designed to solve discrete optimization problems by constructing a parameterized quantum circuit that alternates between applying problem-specific and mixer Hamiltonians. The goal of this algorithm is to evolve an initial quantum state into one that corresponds to the optimal solution of the target optimization problem.

QAOA has been successfully applied to a variety of problems, including Max-Cut, the traveling salesman problem, and graph coloring [2-4]. Despite its promise, QAOA faces practical challenges, particularly in the optimization of circuit parameters. As the number of circuit layers used for problem solutions increases, the parameter space grows, making classical optimization increasingly difficult due to issues such as barren plateaus and noise in quantum measurements [5].

To address these challenges, researchers have explored methods to initialize QAOA circuits with parameters that are more likely to yield high-quality solutions. One such method is warm-starting, where parameters are ini¬tialized based on classical approximations or prior quantum solutions [6]. Another promising approach is parameter transfer, where in parameters optimized for a small graph are reused for a larger graph with similar structural characteristics. This idea builds on the hypothesis that structurally similar graphs exhibit similar QAOA parameter landscapes, enabling efficient reuse of optimized parameters without full re-optimization.

Previous work has demonstrated that QAOA circuits can exhibit parameter patterns that generalize across instances, and that transferring parameters from low-depth circuits can provide a meaningful head-start for deeper circuit optimization [7,8]. In the context of real-world applications such as autonomous vehicle mobility, where optimization tasks are often graph-based and structurally repetitive, parameter transfer can significantly reduce computational costs and enable scalability. Real-world autonomous systems such as drone swarms, for example, face exactly the kind of large-scale, graph-structured combinatorial challenges – task distribution, inter-agent communication, and resource allocation under energy constraints – for which scalable quantum optimization is well-motivated [9].

In this paper, we apply QAOA and parameter transfer techniques to a class of graph optimization problems relevant to autonomous mobility, using the Max-Cut formulation as a widely applicable proxy for a range of graph-based tasks. We explicitly scope this work to this formulation while acknowledging that extending to other problem types (e.g., assignment and routing) is an important direction for future work. We empirically evaluate the performance of transferred parameters across a range of graph sizes and problem instances, demonstrating the viability and efficiency of this approach on both simulators and real quantum hardware. Our contributions include an in-depth analysis of parameter transfer performance on real quantum hardware and its implications for future quantum-enhanced autonomous systems.

Autonomous Mobility Problems

Autonomous ground vehicles face a range of complex decision-making challenges that are essential for navigation, coordination, and mission execu¬tion. These challenges frequently manifest as optimization problems, often over dynamic environments with constraints such as terrain, obstacles, communication limits, and resource availability. Many of these problems are computationally intensive and can be mapped onto combinatorial or graph-based formulations, making them strong candidates for quantum acceleration.

We highlight below several representative problem classes relevant to autonomous mobility, along with a description of how each maps to graph-based formulations amenable to Max-Cut encoding. While these are far from exhaustive, they illustrate the breadth of problem types and computational challenges that can benefit from quantum algorithms such as QAOA.

Path Planning with Static and Moving Obstacles

Path planning is a fundamental problem in robotics and autonomous systems. In environments with static obstacles, the problem is typically modeled as a shortest path over a weighted graph, where nodes represent navigable positions and edges encode traversal costs. This formulation is solvable in polynomial time, but becomes significantly harder when dynamics such as moving obstacles are introduced. The presence of time-varying constraints transforms the problem into an NP-hard variant, demanding more sophisticated approaches for timely and optimal decisions [10-12].

The underlying graph structure for path planning – a grid or mesh of waypoints – is highly regular. When reasoning about which regions of a map to prioritize or how to partition navigation spaces for multi-vehicle coordi¬nation, the problem can be cast as a graph partitioning or Max-Cut problem over this mesh. Specifically, the Max-Cut formulation identifies a partition of the waypoint graph that maximizes inter-region edge weights, which corresponds to maximizing coverage or information gain at region boundaries –a structure directly useful in surveillance and patrolling scenarios.

Multi-Agent Resource Assignment

Resource assignment involves determining optimal allocations of tasks to vehicles or agents in a distributed system. For example, assigning a fleet of autonomous vehicles to cover surveillance regions or to complete logistics missions can be formulated as a generalized assignment problem or a form of weighted bipartite matching [13,14]. These problems are commonly NP-hard and require heuristic or approximate methods to solve efficiently, especially when temporal and spatial dependencies are involved.

The bipartite graph connecting agents to tasks can be reduced to a Max-Cut formulation by encoding conflict or preference relationships as edge weights. A maximum cut of the agent-task bipartite graph yields a high-quality assignment in which agents on one side are matched to tasks on the other, with maximum total edge weight reflecting maximum compatibility or coverage. This mapping, while an approximation of the full assignment problem, captures the essential combinatorial structure and is well-suited to QAOA.

Clustering and Perception in Sensor Networks

Autonomous vehicles rely on real-time processing of data from a wide array of sensors including LIDAR, radar, and cameras. Clustering is often used to group sensor data for object detection and scene segmentation. The clustering problem, particularly when cast as a Max-Cut formulation, becomes NP-hard [15,16]. Solving such problems rapidly and accurately is critical for timely environmental understanding, particularly in unstructured or off-road environments. In this context, sensor readings or detected objects are represented as graph nodes, with edges weighted by similarity or spatial proximity. A maximum cut partitions the sensor graph into two groups such that dissimilar observations are separated – directly implementing a binary clustering that can be used for foreground/background segmentation or obstacle region identification.

Additional Applications

Beyond the examples listed above, many other critical tasks in autonomous mobility can be cast as combinatorial optimization problems. These include dynamic task reassignment in response to environmental changes, formation control, target tracking, and adaptive routing under uncertainty. Real-world deployments of autonomous swarm systems, such as heterogeneous UAV networks performing reconnaissance or logistics, similarly face large-scale task distribution and communication graph optimization challenges that are naturally structured as Max-Cut or related graph partitioning problems [9].

One recurring theme across these challenges is the prevalence of graph-based formulations that are amenable to quantum algorithms like QAOA. This motivates our investigation into parameter transfer techniques that can help scale QAOA applications across diverse instances in real-world autonomy contexts. The scope of the present paper is limited to the Max-Cut formulation, which we treat as a broadly applicable and theoretically grounded proxy for these graph-based mobility problems; extending parameter transfer to other formulations such as quadratic assignment or vehicle routing is an important avenue for future work.

Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm introduced by Farhi et al. designed to solve combinatorial optimization problems on near-term quantum devices [1]. QAOA operates by preparing a quantum state using a parameterized sequence of unitary operations derived from two Hamiltonians: the problem-specific cost Hamiltonian H_C and a mixing Hamiltonian H_M.

QAOA begins with an initial quantum state $|\psi_0\rangle$, typically chosen to be the uniform superposition over all computational basis states:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle, \qquad (1)$$

where n is the number of qubits and x denotes a bitstring representing a candidate solution.

The QAOA ansatz of depth p applies p alternating layers of unitary op¬erations:

$$|\psi(\gamma,\beta)\rangle = U(H_M,\beta_p)\,U(H_C,\gamma_p)\cdots U(H_M,\beta_1)\,U(H_C,\gamma_1)\,|\psi_0\rangle, \qquad (2)$$

where $\gamma = (\gamma_1, \ldots, \gamma_p)$ and $\beta = (\beta_1, \ldots, \beta_p)$ are variational parameters opti¬mized by a classical algorithm.

For combinatorial problems such as Max-Cut, the cost Hamiltonian is defined as:

$$H_C = \sum_{(i,j)\in E} \frac{1}{2}(1 - Z_i Z_j), \qquad (3)$$

where Z_i and Z_j are Pauli-Z operators acting on qubits i and j, and E is the set of edges in the graph. This formulation ensures that the ground state of HC encodes the optimal solution to the Max-Cut problem.

The mixing Hamiltonian, designed to induce transitions between bit-strings, is typically:

$$H_M = \sum_{i=1}^{n} X_i, \qquad (4)$$

where X_i is the Pauli-X operator on qubit i.

The expectation value of the cost Hamiltonian with respect to the evolved quantum state is:

$$F(\gamma,\beta) = \langle\psi(\gamma,\beta)|H_C|\psi(\gamma,\beta)\rangle. \qquad (5)$$

This objective is evaluated via repeated circuit execution and measurement (sampling), and a classical optimizer is used to update (γ, β) to maximize F. While QAOA is provably optimal in the limit of infinite depth p, practical implementations are constrained to small p due to hardware noise and decoherence. Techniques such as warm-starting, parameter interpolation [8], and machine learning-based parameter prediction have been proposed to accelerate convergence and improve performance [1,6-8].

Barren Plateaus and Local Minima

A particularly important obstacle for QAOA scalability is the barren plateau phenomenon, in which the gradient of the objective function with respect to the variational parameters becomes exponentially small as the number of qubits or layers increases. When the optimizer encounters a barren plateau, gradient-based and even gradient-free methods can stall, consuming many function evaluations without meaningful improvement [5].

Parameter transfer directly addresses this challenge. By initializing the variational parameters for a large-graph QAOA instance from values optimized on a structurally similar small graph, the optimizer begins in a region of parameter space that is already reasonably well-conditioned – reducing the probability of entering a barren plateau. Furthermore, the transferred parameters sometimes outperform those found by a full classical optimization of the large-graph instance. This can occur because the classical optimizer on the large graph, started from a generic initial condition, converges to a local minimum rather than the global optimum. The transferred parameters, by contrast, may land in the basin of attraction of a better solution. This intuition is supported by our experimental results in Section 6 and is consistent with the theoretical analysis of fixed-angle QAOA parameters [17]. Recent benchmarking on quantum hardware shows that QAOA, even with shallow depths and modest qubit counts, can outperform classical heuristics for specific structured problems, further supporting its relevance in ap-plied domains like autonomous mobility [4].

QAOA Parameter Transfer

A key challenge in deploying QAOA to large-scale problems lies in optimizing its variational parameters (γ, β). Since the optimization process for larger problems typically requires significant computational resources, the state of hardware in the NISQ era limits the potential for direct parameter optimization at larger problem sizes. Parameter transfer provides a promising mechanism to address this scalability barrier.

Motivation and Concept

The central idea of QAOA parameter transfer is to leverage the similarity between small and large instances of an optimization problem to reuse parameter values. Rather than performing a full classical optimization for every new instance or problem size, one can first optimize QAOA parameters on a smaller, representative graph and then apply these same parameters to a larger graph with similar structural properties – either directly with a single circuit execution, or as initial warm-start values for further optimization [6]. This approach reduces the need for repeated expensive optimization and can enable near-term devices to handle more practically relevant problem sizes.

This concept is inspired by empirical observations and theoretical studies suggesting that the optimal QAOA parameters exhibit smooth and often pre-dictable trends as a function of graph size and structure [8,7]. For instance, in certain classes of graphs (e.g., regular graphs or grid-like graphs), optimal angles tend to converge or exhibit repetitive patterns as n increases.

Formalization

Let $G_k = (V_k, E_k)$ be a small graph instance used to optimize QAOA parameters $\theta_k = (\gamma^{(k)}, \beta^{(k)})$. Given a larger graph $G_l = (V_l, E_l)$, we define a structural similarity function $\phi(G_k, G_l)$ that quantifies how similar the topol¬ogy of G_k is to G_l. If $\phi(G_k, G_l)$ exceeds a threshold ϵ, then the parameters θ_k are transferred to G_l without re-optimization:

$$\theta_l = \theta_k, \quad \text{if } \phi(G_k, G_l) > \epsilon. \tag{6}$$

Alternatively, these parameters can be used as a warm start for partial re-optimization on the larger graph. In the current work, we use the transferred parameters directly in order to isolate and quantify the quality achievable from a single circuit evaluation.

In practice, we evaluated two candidate measures for ϕ. Degree distribution similarity – comparing the histogram of vertex degrees between donor and recipient graphs – served as a simple and effective first-order screen: compatible degree sequences mean the local edge environment seen by each qubit is structurally analogous, which is the primary condition enabling parameter reuse. We also evaluated direct subgraph embedding, checking whether the donor graph appears as an induced subgraph of the recipient; when such an embedding exists, the donor's local structure is literally present within the larger graph, providing a strong theoretical basis for transfer. For the 4-regular graphs used in our hardware experiments, degree distribution is by construction identical across all graph sizes, making this the primary basis for donor-recipient matching. Developing a unified, quantitative similarity score and systematically evaluating its predictive power for transfer quality is an important direction for future work.

Prior Work

Recent studies have explored parameter transfer in various contexts. Zhou et al. showed that low-depth QAOA exhibits regularity in optimal parameters for problem instances of increasing size [7]. Galda et al. demonstrated the convergence of optimal parameters around specific values and applied parameter transfer to random regular and general random graphs, finding less than a 1% reduction in the approximation ratio for certain instances [18]. Shaydulin et al. showed that parameter transfer from unweighted Max-Cut problems also performs well on weighted Max-Cut, with a median decrease in approximation ratio of just 2% compared to a complete QAOA optimization [19]. Montanez-Barrera et al. found that parameter transfer has practical applicability even with a high number of layers (up to p = 10), as well as for a variety of NP-hard problem types including Traveling Salesman, Bin Packing, Knapsack, and Max-Cut [20].

In the present work, we extend these ideas by applying parameter transfer to practical autonomous mobility graphs, validating the approach on both simulators and real quantum devices, and presenting results on graphs of larger size than in most current literature.

Relevance to Autonomous Systems

In autonomous mobility, problem instances such as vehicle coordination or obstacle avoidance often share common structural elements – e.g., grid maps, sensor meshes, or communication networks. These natural similarities make parameter transfer especially well-suited to this domain, where time and computational resources are often constrained.

The present paper focuses on the offline or preprocessing use of QAOA parameter transfer. The classical optimization of donor-graph parameters is performed ahead of time, and the resulting parameters are stored and applied to new problem instances as they arise. This framing is realistic given current quantum hardware constraints: cloud-accessed processors such as the Rigetti Ankaa-2 have latencies on the order of seconds to minutes per circuit execution, which precludes real-time use in the millisecond decision loop of an autonomous vehicle. However, the single-circuit evaluation enabled by parameter transfer represents a drastic reduction in quantum runtime compared to full optimization (which requires hundreds of circuit evaluations), making near-real-time quantum-assisted decision support a plausible trajectory as hardware matures.

Experimental Design

Our experimental design was structured to evaluate the effectiveness of QAOA parameter transfer for solving graph-based optimization problems relevant to autonomous ground vehicle mobility. The design was guided by three key questions: Does parameter transfer preserve solution quality? How do solutions obtained through parameter transfer compare to fully optimized QAOA? Can the technique scale to larger graphs on current quantum processors?

Problem Selection and Encoding

We selected a suite of representative graph problems derived from path planning, resource assignment, and clustering contexts. These problems were encoded using the Max-Cut formulation:

$$H_C = \sum_{(i,j)\in E} \frac{1}{2}(1 - Z_i Z_j), \tag{7}$$

where Z_i is the Pauli-Z operator on qubit i and E denotes the edge set of the graph. The Max-Cut formulation allows us to evaluate solution quality via the approximation ratio with respect to the classical maximum. Graphs ranged in size from 10 to 80 vertices.

Results in this paper are presented for 4-regular graphs, which arise naturally in grid-like path planning and communication topologies. All other tested graph structures produced qualitatively similar results; we focus on 4-regular graphs for clarity and to enable direct comparison with prior literature [17,18].

Baseline Definition

To ensure reproducibility and fair comparison, we define the "full QAOA optimization" baseline precisely. For each recipient graph, QAOA parameters (γ, β) were optimized from scratch using the COBYLA optimizer (the same optimizer used for donor-graph parameter search), with random initialization drawn uniformly from [0, π] [0, π/2]. A fixed budget of 200 function evaluations was used for all runs. A single random initialization was used per graph instance (no restarts), reflecting realistic single-shot optimization on NISQ hardware where repeated optimization is expensive. The comparison is therefore between parameter transfer (one circuit evaluation) and a practical single-pass classical optimization (up to 200 evaluations).

HPC Simulation

QAOA parameters (γ, β) were optimized for small graphs of specific structures arising from autonomous mobility problems, using a classical optimization loop integrated with a noiseless Qiskit simulator and COBYLA optimizer. For each graph, we selected a circuit depth of p = 1 or p = 2 to reflect NISQ constraints. Once optimal parameters were found, we applied them directly to larger graphs, using donor-recipient matching via degree-distribution similarity and subgraph embedding as described in Section 4.

Quantum Hardware Execution

Quantum hardware experiments were executed on the Rigetti Ankaa-2 quantum processor via Amazon Braket, using the Python Braket SDK along with Qiskit. Each circuit was executed with 2,048 shots to obtain statistically stable expectation values. Using donor graph orders of 5, 10, and 20 vertices, experiments were run using larger recipient graphs of orders up to 80 vertices. The Rigetti Ankaa-2 uses a fixed qubit connectivity graph, necessitating SWAP-based routing for circuits that require connectivity not natively supported by the hardware. This routing overhead increases circuit depth and introduces additional gate errors. All hardware experiments for a given comparison were conducted within the same calibration window to minimize calibration drift. We observed that hardware noise generally degraded approximation ratios relative to ideal simulation for both methods, but the relative performance between the two was largely preserved, suggesting that the parameter transfer advantage is robust to the noise levels present in current NISQ devices.

Results

The results of our experiments confirm the effectiveness of QAOA parameter transfer as a scalable technique for quantum optimization in autonomous mobility applications. For clarity, results are presented here for 4-regular graphs, although all tested graph types produced similar results.

Transferred QAOA parameters from small donor graphs (5, 10, and 20 vertices) consistently performed well when applied to larger graphs (up to 80 vertices). For all donor graph sizes, the average approximation ratio from a single QAOA circuit execution was less than 3% lower than the result of fully optimized QAOA, with some configurations even exceeding full optimization performance [9,17].

Evaluation Metrics

Results are reported using two complementary metrics. The best-sampled value is the maximum Max-Cut value observed across all 2,048 shots, and represents the best solution found by the algorithm in a single run. The expectation value is the average Max-Cut value across all shots, and characterizes the typical quality of the output distribution. We report both throughout this section, noting explicitly which is shown in each figure.

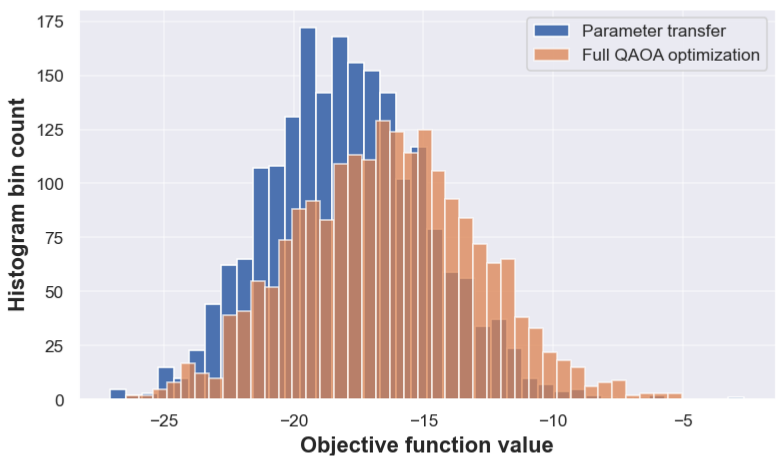

Figure 1: Distribution of Sampled Results From a 40-Vertex Graph, Comparing Parameter Transfer from a 5-vertex Donor Graph (Single Circuit Execution) To A Full QAOA Optimization

Quantum Hardware Results

We executed QAOA single-circuit and full optimization runs on the Rigetti Ankaa-2 84-qubit quantum processor via Amazon Braket. Using donor sizes of $d = 5$, $d = 10$, and $d = 20$, we performed both a single circuit execution using transferred donor parameters and a full QAOA optimization for graphs of each order $n = d + 1, d + 2, \dots, 80$, with 4-regular structure matching the donor graphs.

Figure 1 shows the distribution of sampled shot values for a donor graph of 5 vertices and a recipient graph of 40 vertices. Since the problems were modeled as minimization problems, lower values are better. Although the distributions have somewhat different shapes, the best-sampled values were similar: the full QAOA optimization was 2.22% better than the transferred-parameter result for this particular instance.

Figure 2 shows approximation ratios (best-sampled values) for each donor graph size across the full range of recipient graph sizes.

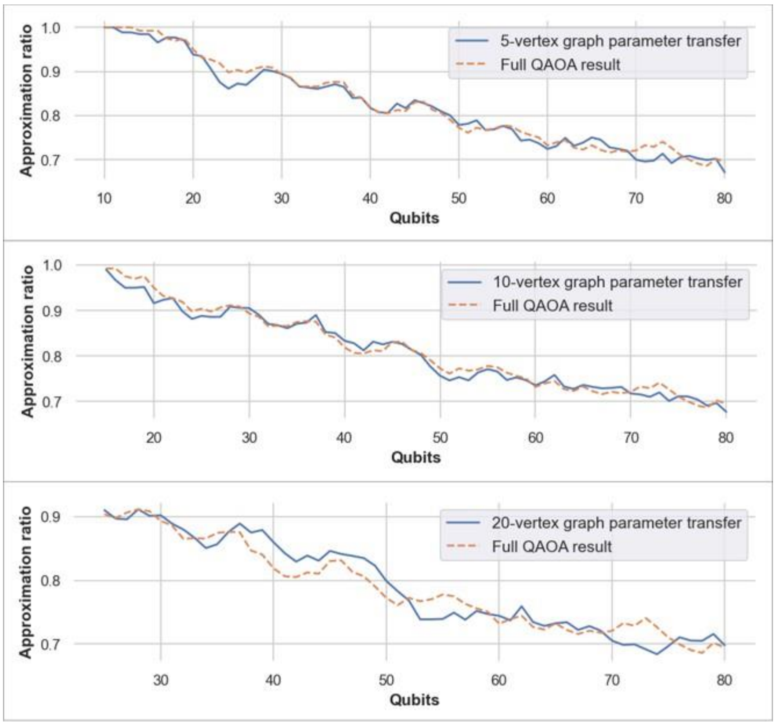

Figure 2: Comparison of Parameter Transfer (Single-Circuit Execution, Best-Sampled Values) with Full QAOA Optimization on the Rigetti Ankaa-2 Quantum Processor For a Range of Graph Sizes And Donor Sizes

The average advantage for full QAOA optimization was 0.598% for the 5-vertex donor, 2.778% for the 10-vertex donor, and 5.659% for the 20-vertex donor, meaning that for the largest donor size the single-circuit execution actually outperformed full QAOA optimization.

Figure 3 shows approximation ratios based on expectation values for a consistent set of problem instances, comparing full QAOA optimization with parameter transfer from all three donor graph sizes.

On average, full QAOA optimization had an advantage of 1.451% over transfer from the 5-vertex graphs, 0.736% over transfer from the 10-vertex graphs, and 0.603% over transfer from the 20-vertex graph. These results indicate that parameter transfer performs well, and that the advantage from using larger donor graph sizes is very slight – almost equal results can be obtained using a donor graph that is much smaller than the recipient

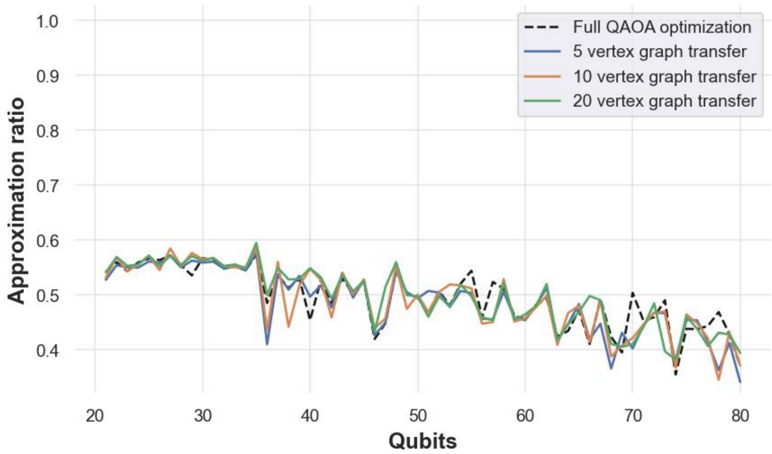

Figure 3: Approximation Ratios Based on Expectation Values for Full QAOA Optimization Compared with Parameter Transfer From Donor Graphs of 5, 10, and 20 Vertices

Figure 4 provides additional distributional information for the 5-vertex donor experiments, displaying inter-quartile ranges alongside expectation values and best-cut values for each recipient graph size.

Performance Boundaries and Degradation Cases

The 80-vertex ceiling in our experiments is imposed by the qubit count of the Rigetti Ankaa-2 processor (84 qubits, with some reserved for routing overhead) and the depth limitations of NISQ circuits at this scale. In principle, parameter transfer should continue to function beyond 80 qubits as hardware scales, provided that structural similarity between donor and recipient graphs is maintained. We expect parameter transfer to degrade in quality when donor and recipient graphs differ substantially in structure –for example, when transferring parameters optimized on regular graphs to highly irregular or skewed graphs such as scale-free or power-law networks.

Relationship to Classical Heuristics

A complete benchmark against classical heuristics is beyond the scope of this paper. Nevertheless, it is instructive to situate QAOA within the theoretical landscape of classical approximation algorithms. The Goemans–Williamson (GW) semidefinite programming algorithm achieves a prov-able approximation ratio of at least 0.878 for general Max-Cut; simple greedy [21].

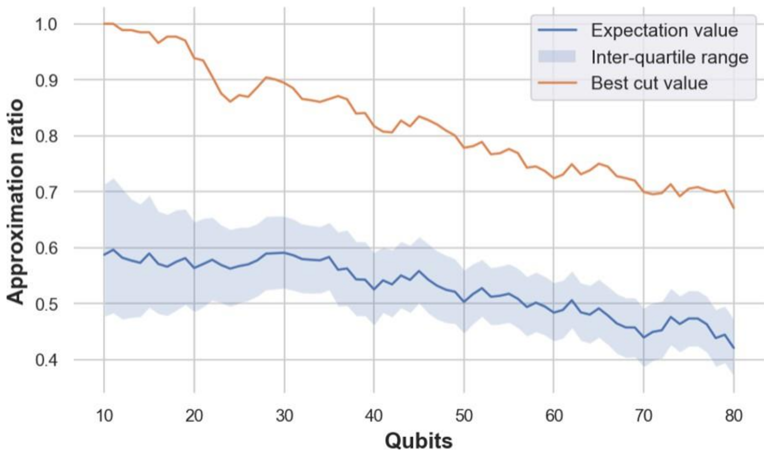

Figure 4: Expectation Values, Inter-Quartile Ranges, And Best Cut Values For A Range Of Graph Problem Sizes Using Parameter Transfer From A 5-Vertex Donor Graph

algorithms achieve ratios in the range 0.5–0.75. QAOA at depth p = 1 is known to achieve an approximation ratio of at least 1/2 for general graphs, with performance improving for structured graph classes and larger p. Crucially, our results show that parameter transfer does not degrade this performance: transferred parameters yield approximation ratios within 3% of fully optimized QAOA on average, reducing the quantum computational overhead to a single circuit evaluation [1]. A rigorous, timed empirical comparison against GW and other heuristics is an important direction for future work.

Implications for Autonomous Systems

The results demonstrate that QAOA parameter transfer provides a viable path to scaling quantum optimization for real-world autonomous mobility applications. The fact that a single circuit evaluation using parameter transfer can produce results similar to – and in some cases better than – a full QAOA optimization indicates that this approach has great promise to reduce quantum computation time for these types of problems by orders of magnitude.

Conclusions

This work introduces and validates the QAOA parameter transfer methodology aimed at addressing the scalability challenges of quantum optimization for autonomous mobility. By leveraging parameter optimization on small, structurally representative graphs, our approach allows for efficient application of of QAOA to larger graphs, significantly reducing the computational burden required for QAOA-based solutions to autonomous mobility problems.

Our experimental evidence, gathered through both simulation and execution on real quantum devices, indicates that this technique provides a practical means of achieving high-quality solutions without the computational burden of full parameter optimization. In some cases, a single-circuit QAOA execution using transferred parameters even outperforms a full QAOA optimization, which we attribute to the avoidance of local minima consistent with the theory of barren plateaus and fixed-angle QAOA parameters [5,17].Theoretically, QAOA at current shallow depths does not claim to surpass the best classical approximation algorithms such as Goemans–Williamson; however, parameter transfer drastically reduces the quantum resource cost of QAOA without sacrificing solution quality relative to the fully optimized quantum baseline, setting the stage for meaningful quantum-classical com-parisons as hardware and circuit depth continue to improve.

In future work we hope to extend this strategy to problems in granular terrain modeling, to evaluate parameter transfer on irregular and heterogeneous graph structures, and to more fully explore specific graph structural similarity measures for potential gains in parameter transfer accuracy. A rigorous benchmark against classical heuristics, and an extension to problem formulations beyond Max-Cut, are also important future directions.

References

1. Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.
2. Crooks, G. E. (2018). Performance of the quantum approximate optimization algorithm on the maximum cut problem. arXiv preprint arXiv:1811.08419.
3. Willsch, M., Willsch, D., Jin, F., De Raedt, H., & Michielsen, K. (2020). Benchmarking the quantum approximate optimization algorithm. Quantum Information Processing, 19(7), 197.
4. Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., ... & Babbush, R. (2021). Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nature Physics, 17(3), 332-336.
5. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. Nature communications, 9(1), 4812.
6. Egger, D. J., Mareček, J., & Woerner, S. (2021). Warm-starting quantum optimization. Quantum, 5, 479.
7. Zhou, L., Wang, S. T., Choi, S., Pichler, H., & Lukin, M. D. (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Physical Review X, 10(2), 021067.
8. Alam, M., Ash-Saki, A., & Ghosh, S. (2020). Accelerating quantum approximate optimization algorithm using machine learning. arXiv preprint arXiv:2002.01089.
9. Bakirci, M. (2023). A novel swarm unmanned aerial vehicle system: Incorporating autonomous flight, real-time object detection, and coordinated intelligence for enhanced performance. Traitement du Signal, 40(5), 2063-2078.
10. Canny, J., & Reif, J. (1987, October). New lower bound techniques for robot motion planning problems. In 28th Annual Symposium on Foundations of Computer Science (sfcs 1987) (pp. 49-60). IEEE.
11. Frederick, P., Kania, R., Rose, M. D., Ward, D., Benz, U., Baylot, A., ... & Yamauchi, H. (2005, October). Spaceborne path planning for unmanned ground vehicles (UGVs). In MILCOM 2005-2005 IEEE Military Communications Conference (pp. 3134-3141). IEEE.
12. Dallas, J., Cole, M. P., Jayakumar, P., & Ersal, T. (2021). Terrain adaptive trajectory planning and tracking on deformable terrains. IEEE Transactions on Vehicular Technology, 70(11), 11255-11268.
13. Kulatunga, A. K., Liu, D. K., Dissanayake, G., & Siyambalapitiya, S. B. (2006, June). Ant colony optimization based simultaneous task allocation and path planning of autonomous vehicles. In 2006 IEEE Conference on Cybernetics and Intelligent Systems (pp. 1-6). IEEE.
14. Farinelli, A., Iocchi, L., & Nardi, D. (2017). Distributed on-line dynamic task assignment for multi-robot patrolling. Autonomous Robots, 41(6), 1321-1345.
15. Wigness, M., Eum, S., Rogers, J. G., Han, D., & Kwon, H. (2019, November). A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5000-5007). IEEE.
16. Maturana, D., Chou, P. W., Uenoyama, M., & Scherer, S. (2017, November). Real-time semantic mapping for autonomous off-road navigation. In Field and Service Robotics: Results of the 11th International Conference (pp. 335-350). Cham: Springer International Publishing.
17. Brandao, F. G., Broughton, M., Farhi, E., Gutmann, S., & Neven, H. (2018). For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. arXiv preprint arXiv:1812.04170.
18. Galda, A., Liu, X., Lykov, D., Alexeev, Y., & Safro, I. (2021, October). Transferability of optimal QAOA parameters between random graphs. In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE) (pp. 171-180). IEEE.
19. Shaydulin, R., Lotshaw, P. C., Larson, J., Ostrowski, J., & Humble, T. S. (2023). Parameter transfer for quantum approximate optimization of weighted maxcut. ACM Transactions on Quantum Computing, 4(3), 1-15.
20. Montanez-Barrera, J. A., Willsch, D., & Michielsen, K. (2025). Transfer learning of optimal QAOA parameters in combinatorial optimization: JA Montañez-Barrera et al. Quantum Information Processing, 24(5), 129.
21. Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM), 42(6), 1115-1145.