

Improving Join Order Optimization on Gate-Based Quantum Computers via Structured Parameter Initialization

Divya Shekar*
Ruokun Wu*
shdivya@umich.edu
ruokun@umich.edu
University of Michigan
Ann Arbor, Michigan
United States

Dhanvi Bharadwaj
University of Michigan
Ann Arbor, Michigan
United States
dhanvib@umich.edu

Gokul Subramanian
Ravi
University of Michigan
Ann Arbor, Michigan
United States
gsravi@umich.edu

Lin Ma
University of Michigan
Ann Arbor, Michigan
United States
linmacse@umich.edu

ABSTRACT

Join Order Optimization (JOO) is one of the most computationally expensive tasks in relational query optimization due to the exponential growth of possible join plans with increasing query size. Recent work has explored quantum and quantum-inspired approaches for solving JOO by reformulating the problem as a Quadratic Unconstrained Binary Optimization (QUBO) problem suitable for optimization using quantum hardware. However, many existing approaches have limited scalability on current gate-based quantum devices. In addition, little work has investigated the role of initialization strategies in improving the performance of gate-based quantum optimization for database workloads. In this work, we investigate gate-based quantum join order optimization using the Quantum Approximate Optimization Algorithm (QAOA) initialized with Scalable Parameter Initialization for QAOA (SPIQ). SPIQ is used to efficiently identify high-quality initial points in the quantum solution landscape for QAOA executed on a gate-based quantum computer. We evaluate the interaction between QUBO encoding, SPIQ initialization, and gate-based optimization on small-scale join ordering problems involving 3 and 4 relations. Our results show that structured initialization improves optimization stability and increases convergence toward high-quality join plans compared to uninformed initialization approaches. Across these small-scale, simulation-based instances, SPIQ increases the sampling frequency of the optimal join order by up to approximately 5× and yields final-state energies significantly lower than a randomly initialized QAOA. Overall, this work enhances existing gate-based quantum optimization while providing an initial proof of concept for applying SPIQ initialization to database query optimization workloads.

VLDB Workshop Reference Format:

Divya Shekar, Ruokun Wu, Dhanvi Bharadwaj, Gokul Subramanian Ravi, and Lin Ma. Improving Join Order Optimization on Gate-Based Quantum Computers via Structured Parameter Initialization. VLDB 2026 Workshop: QC&DKM 2026- 2nd Workshop on Quantum Computing and Data/Knowledge Management.

*Both authors contributed equally to this research.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

VLDB Workshop Artifact Availability:

Github Codebase: <https://github.com/umich-db/quantumdb/tree/spiq-joo>.

1 INTRODUCTION

The performance of modern Relational Database Management Systems (RDBMS) is fundamentally dependent on the quality of query execution plans generated by the optimizer [25]. Among the many components of query planning, Join Order Optimization (JOO) remains one of the most computationally challenging and performance-critical tasks [18, 26]. Given a query involving multiple relations, the optimizer must determine an execution order that minimizes intermediate result sizes and overall execution cost. The number of possible join orders grows exponentially with the number of relations, making the problem NP-hard and increasingly difficult for large enterprise workloads involving tens or hundreds of tables [11]. Classical optimizers traditionally rely on dynamic programming, greedy heuristics, and cost-based pruning strategies using relation statistics and predicate selectivities [8, 25]. However, as modern workloads continue to grow in scale and complexity, classical hardware and exhaustive search methods struggle to efficiently explore the rapidly expanding solution space.

Quantum computing has emerged as a promising alternative paradigm for solving large combinatorial optimization problems [20, 22]. Unlike classical computation, which evaluates candidate solutions sequentially, quantum systems exploit superposition and entanglement to explore multiple states simultaneously. This has motivated the application of quantum algorithms to database optimization problems, particularly join ordering. By representing join order optimization as an energy minimization problem, candidate execution plans can be encoded into a Hamiltonian whose ground state corresponds to the minimum-cost join order. Recent work has demonstrated encouraging proof-of-concept results using Quantum Approximate Optimization Algorithm (QAOA) [5], quantum annealing [13], and quantum-inspired optimization frameworks [16]. These approaches suggest that quantum computing may eventually provide scalable mechanisms for navigating the exponential search space of query optimization more efficiently than classical techniques.

Most prior work in quantum join order optimization has focused on annealing-based or quantum-inspired optimization methods rather than gate-based quantum computation [24]. While such approaches have demonstrated favorable scalability and reduced encoding complexity, they do not capture the full expressive power

of universal quantum computation. Gate-based quantum computing, by contrast, provides a fundamentally quantum framework that supports parameterized quantum circuits, quantum interference, and variational algorithms such as QAOA [3, 5]. Gate-based methods are therefore both more general and theoretically more powerful than annealing-based systems, making them an important direction for long-term quantum database research.

Despite their theoretical advantages, gate-based quantum algorithms remain difficult to optimize in practice. Current Noisy Intermediate-Scale Quantum (NISQ) devices are constrained by limited qubit counts, short coherence times, restricted qubit connectivity, and high noise levels [22]. In addition, QAOA itself introduces a challenging hybrid optimization problem in which the quality of the final solution depends heavily on the initialization and optimization of circuit parameters [3, 28]. Poor initialization can cause the optimizer to converge to low-quality local minima or produce unstable training behavior, especially for highly non-convex energy landscapes such as those arising in join order optimization [17].

To address this issue, recent work has explored improved initialization strategies for variational quantum algorithms. Techniques such as layer-wise initialization [28], transfer-based initialization, warm-start methods [4], and SPIQ (Scalable Parameter Initialization for QAOA) [2] aim to provide parameter values closer to promising regions of the optimization landscape before training begins. By improving the starting point of the variational optimization process, these methods can potentially accelerate convergence and increase the probability of reaching near-optimal solutions on noisy hardware.

In this work, we investigate the application of SPIQ initialization to gate-based quantum join order optimization using QAOA, represented by a QUBO. By using the compact native encoding from [24], we require only a reasonable number of program qubits. The resulting QUBO is then mapped to an Ising Hamiltonian and optimized using QAOA initialized with SPIQ-generated parameters. We evaluate using noiseless statevector simulation to isolate the impact of SPIQ initialization on optimization without hardware noise effects. While noise-aware simulations could provide insight into real-device performance, they primarily reflect hardware fidelity rather than initialization quality and incur significantly higher computational cost. Evaluating SPIQ under realistic noise models and on physical hardware remains future work.

Our evaluation focuses on small-scale join queries over 3 and 4 relations, with problem size constrained by the cost of classical simulation of quantum systems. Within this regime, we analyze the interaction between the QUBO encoding, SPIQ initialization, and gate-based QAOA optimization. The results show that informed initialization improves convergence toward high-quality candidate join orders and yields more stable optimization behavior than uninformed initialization. With SPIQ-initialized parameters, we achieve at most a 5 $\times$ improvement in optimal cost frequency, and the final state energy is significantly lower than that of the randomly initialized QAOA across the three problems. We expect these benefits to persist at larger problem sizes when executed on real quantum hardware, though qubit error effects will need to be accounted for. The primary contributions of this work are as follows: (1) adapting the encoding from annealing-oriented quantum JOO approaches to

gate-based quantum optimization for better scalability, (2) presenting the first application of SPIQ initialization to gate-based quantum join order optimization, and (3) identifying practical limitations and future research directions for scalable quantum-enhanced query optimization.

2 BACKGROUND AND EXISTING WORK

2.1 Quantum Computing

Quantum computing is an emerging computational paradigm that utilizes quantum mechanical phenomena such as superposition, entanglement, and interference to solve computational problems [20, 22]. Unlike classical bits, which exist only in states 0 or 1, quantum bits (qubits) can exist in linear combinations of both states simultaneously. This enables quantum systems to explore large solution spaces more efficiently than classical architectures for certain classes of optimization and search problems. Current quantum computing approaches can broadly be divided into two categories: gate-based quantum computing and quantum annealing.

Gate-based quantum computing operates using sequences of parameterized quantum gates applied to qubits in a circuit model. Universal quantum computers belong to this category and support algorithms such as Grover’s search algorithm [9], Variational Quantum Eigensolvers (VQE) [21], and the Quantum Approximate Optimization Algorithm (QAOA) [5]. QAOA is particularly relevant for combinatorial optimization because it solves optimization problems by encoding them into a Hamiltonian and variationally minimizing the corresponding energy function through alternating cost and mixing operators. However, current gate-based systems remain limited by noise, decoherence, restricted qubit connectivity, and small qubit counts, making large-scale optimization difficult on present-day Noisy Intermediate-Scale Quantum (NISQ) hardware [3, 22].

2.2 Quantum Computing for Join Order Optimization

The main idea in the quantum line of work for Join Order Optimization (JOO) is to rewrite join ordering as a binary optimization problem that can be solved by quantum or quantum-inspired hardware. Most existing approaches formulate the problem as either a QUBO or Ising model because these representations are compatible with both annealing-based methods and gate-based approaches such as QAOA after suitable transformation [5, 16]. In practice, the primary challenge is not only to represent valid join orders, but also to encode predicate applicability and intermediate-result cost without introducing excessive binary variables or dense pairwise couplings. This tradeoff between model expressiveness and hardware feasibility is a recurring theme throughout the literature.

One of the earlier and most relevant works [23] in this area presents what the authors describe as the first implementation of join order optimization on gate-based quantum hardware. The paper emphasizes co-design between the MILP-derived encoding and hardware limitations while still preserving the semantics of join ordering. Although the work successfully demonstrated the feasibility of gate-based quantum join optimization, it also highlighted major scalability challenges, particularly the rapid increase in qubit requirements due to encoding overhead.

In a closely related follow-up [24], instead of targeting gate-based hardware, the focus is on Fujitsu’s Digital Annealer, a quantum-inspired optimization platform designed for large-scale QUBO problems. The authors propose a native encoding specifically tailored for annealing-style hardware rather than adapting earlier MILP-derived formulations. The reported experiments demonstrate scalability to approximately 50 joined relations, substantially larger than the small-scale instances feasible on gate-based hardware. While this work improves practicality and scalability, it remains quantum-inspired rather than fully gate-based quantum optimization.

More recent work has also explored hybrid and system-level approaches including a hybrid quantum-classical query optimizer [15] integrated into a real database environment and describes itself as the first practical quantum-augmented optimizer within a working database system, constrained quadratic models, hybrid classical-quantum optimization, and alternative encoding strategies, collectively demonstrating that the field is gradually moving toward more scalable and practically usable quantum query optimization frameworks [19, 27]. Another line of work is end-to-end quantum-enhanced query optimization. DEDALUS [14] presents a framework that integrates QUBO-based join order optimization with search-space pruning and cost-aware optimization, while supporting both classical and quantum optimization.

2.3 Initialization Strategies for Quantum Computing

Variational quantum algorithms such as QAOA rely heavily on the initialization of circuit parameters [3]. Since QAOA optimizes a highly non-convex energy landscape using a hybrid classical-quantum optimization loop, poor initialization can lead to convergence toward low-quality local minima, unstable optimization behavior, or barren plateau effects where gradients vanish during training [17]. As a result, initialization strategies have become an important research direction for improving the practical performance of variational quantum algorithms.

In QAOA, the optimization process is controlled by sets of variational parameters, commonly denoted as γ and β , which determine how strongly the quantum circuit applies the problem Hamiltonian and mixing operations at each layer. Traditionally, these parameters are initialized randomly, after which a classical optimizer iteratively updates them to minimize the measured energy of the quantum state. However, because the optimization landscape is highly non-linear and often contains many local minima, random initialization can lead to poor convergence behavior or require a large number of optimization iterations before useful solutions are found. Consequently, recent work has explored structured initialization techniques that attempt to place the initial parameters closer to regions of the landscape associated with lower-energy solutions.

One important class of initialization methods is based on Clifford circuits and Clifford parameter search. Clifford operations are a restricted subset of quantum gates that can be simulated efficiently on classical hardware while still preserving important structural properties of quantum states [7]. In the context of QAOA, Clifford angles refer to special parameter settings for γ and β that cause parts of the quantum circuit to behave similarly to Clifford operations. These angles are useful because they provide stable and structured regions

of the optimization landscape that can be explored efficiently before transitioning to fully variational optimization. Clifford-based search methods therefore attempt to identify promising initialization points by first exploring parameter settings that are easier to analyze and less sensitive to noise.

Several other initialization methods have been explored in the literature. Warm-start methods attempt to incorporate classical optimization solutions into the quantum initialization process [4], while layer-wise training methods progressively optimize deeper QAOA circuits using parameters learned from shallower circuits [28]. Transfer-based initialization methods reuse parameters across related optimization instances to accelerate convergence. More recently, SPIQ [2] has been proposed as a structured initialization approach that attempts to preserve information from the problem Hamiltonian when generating initial QAOA parameters. This makes SPIQ particularly attractive for combinatorial optimization problems such as join order optimization, where the search landscape is highly complex and sensitive to initialization quality.

While considerable research has been devoted independently to quantum formulations of join order optimization and to initialization strategies for variational quantum algorithms, little work has investigated the interaction between these two areas. To the best of our knowledge, SPIQ initialization has not previously been applied to quantum join order optimization, creating an opportunity to study whether structured initialization can improve convergence and solution quality for database optimization workloads.

3 METHOD

Our proposed approach begins by representing the join order optimization problem as a query graph containing relations, predicates, cardinalities, and predicate selectivities. We formulate the query graph as a QUBO using a compact state-of-the-art encoding proposed by Schonberger [24] for Quantum Annealing approaches and adapt it to be used for our QAOA-based setup. The resulting QUBO is then transformed into an Ising Hamiltonian suitable for QAOA optimization. Before running QAOA, we apply SPIQ initialization to identify promising starting parameter values for the quantum circuit. SPIQ first performs a classical search over a simplified subset of parameter configurations using efficiently simulatable Clifford-based circuits, and then ranks candidate initialization points according to their energy values. The best-performing initialization point obtained from this search is subsequently used as the starting configuration for the full QAOA optimization process. Figure 1 summarizes the overall pipeline, from query-graph construction and QUBO encoding to SPIQ initialization, QAOA optimization, sampling, and final join-order decoding.

3.1 QUBO Formulation

As mentioned in Section 2, QAOA requires the optimization problem to be expressed as a quadratic unconstrained binary optimization (QUBO) problem, whose ground state corresponds to the optimal solution. Join ordering is not naturally in this form, so an encoding step is required. This encoding directly determines the number of binary variables, and therefore the number of qubits needed to represent the problem. The qubit count is the primary factor that bounds the size of solvable instances, since it scales as 2^n in the

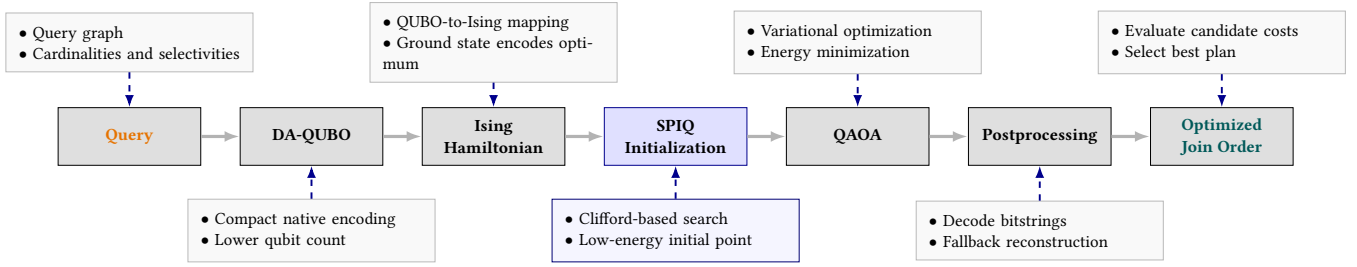


Figure 1: Workflow of SPIQ-initialized QAOA for join order optimization.

number of qubits. The choice of encoding is therefore a central design decision.

We use the state-of-the-art native QUBO encoding proposed by Schonberger, Trummer, and Mauerer [24]. Three design choices make this encoding better suited to join order optimization. The first is how the join tree is represented. It uses a single binary variable per relation per join, and enforces validity based on how many relations are active at each join and preventing a relation from leaving the tree once it has been joined. The second difference lies in how cost is approximated. It uses a quadratic penalty on the logarithmic intermediate cardinality, which captures cost growth continuously. The third follows from the first two. With no inequality constraints anywhere in the formulation, the mandatory variable count is only $(R + P)(J - 1)$ for R relations, P predicates, and J joins.

To adapt this encoding to QAOA, we adopt the native encoding [24], which relies solely on the continuous quadratic cost approximation, so that no auxiliary variables are introduced beyond the $(R+P)(J-1)$ qubits. The model is constructed with docplex and translated, via qiskit-optimization’s docplex translator, into a Qiskit QuadraticProgram. the resulting cost term is the sum over joins of the squared logarithmic intermediate cardinality. We also attempted implementing the MILP-based formulation [23], but found it less suitable than the native encoding for QAOA, because its MILP-style inequality constraints require binary slack variables whose number grows as the JOO problem scales, increasing the qubit count well beyond the native encoding’s $(R+P)(J-1)$.

The QAOA optimization uses Qiskit’s COBYLA optimizer. After performing a comparative study between the classical optimizers COBYLA, SPSA and AQGD, we found that COBYLA performed best with JOO while also achieving the fastest rate of convergence.

3.2 Initialization with SPIQ

A major challenge in QAOA is that the algorithm is highly sensitive to its starting parameters [3]. In QAOA, these parameters control how the quantum circuit explores possible solutions during optimization. If the starting point is poor, the optimizer may become stuck in low-quality solutions or fail to improve the result meaningfully [17]. In our join ordering experiments, random initialization often caused the optimizer to converge to suboptimal join plans or produce unstable optimization behavior. As a result, choosing good initial parameters is an important step in improving QAOA performance.

To address this issue, we integrate the Scalable Parameter Initialization for QAOA (SPIQ) framework [2] into our optimization

pipeline. Instead of starting from random parameter values, SPIQ attempts to identify promising starting points before the full QAOA optimization begins. The main idea is to first explore a simplified version of the quantum circuit that can still capture useful structural information about the optimization problem while remaining computationally inexpensive to evaluate.

SPIQ performs this search using a restricted family of quantum operations known as Clifford operations [7]. These operations are important because they can be simulated efficiently on classical hardware, unlike general quantum circuits which are typically very expensive to simulate. SPIQ searches over a small set of special parameter values, often referred to as Clifford angles, corresponding to rotations of 0 , $\frac{\pi}{2}$, π , and $\frac{3\pi}{2}$. Although these parameter settings represent only a simplified subset of all possible QAOA parameters, they still provide useful information about the structure of the optimization landscape.

To explore this parameter space efficiently, SPIQ uses a genetic algorithm [1, 6]. The search evaluates many candidate parameter settings and ranks them according to the energy of the resulting solution. Lower-energy states correspond to potentially better join orders. Rather than selecting parameters randomly, SPIQ therefore produces initialization points that are already biased toward promising regions of the optimization landscape.

Finally, SPIQ selects a diverse set of high-quality initialization points for QAOA optimization. Instead of using only the single best candidate, the method attempts to choose candidates from different regions of the parameter space so that the optimizer does not repeatedly converge to the same local minimum. This improves robustness and increases the likelihood of discovering better join orders during optimization, as illustrated in Figure 2

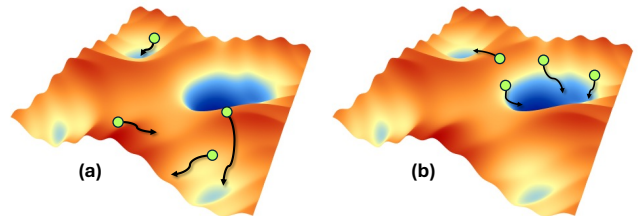


Figure 2: a) depicts points being trapped in sub-optimal or flat regions of the quantum landscape while b) illustrates landscape-aware points more proximal to superior solutions.

The SPIQ initialization integrates into our join ordering workflow as a classical preprocessing stage between QUBO formulation and QAOA execution. The classical SPIQ phase adds no quantum resource overhead and scales polynomially with problem size. Overall, SPIQ provides a structured and problem-aware alternative to random initialization, making QAOA optimization more stable and effective for join order optimization problems.

3.3 Post-Processing

The post-processing stage first converts the output into a relation-join assignment matrix and computes a score vector that reflects how strongly each relation is associated with joins that occur earlier in the order. This ranked ordering is then converted into a candidate join plan whose actual execution cost is evaluated classically using standard join cardinality estimation.

We also employ the same fallback strategy introduced by Schonberger [24] to handle cases where the output may lead to poor or invalid join structures, especially plans containing expensive cross products. Instead of relying entirely on the raw join order output, the fallback mechanism traverses the query graph while respecting the computed ranking order and deliberately avoids disconnected joins whenever possible. In practice, this means that the algorithm tries to construct a connected join sequence following the ranking produced while ensuring that predicates are used before introducing cross products. Finally, both the original candidate plan and the fallback plan are evaluated classically and the lower-cost plan is selected as the most optimal join order.

4 EXPERIMENTAL SETUP

All experiments are executed using Qiskit 0.34.2[12] with the statevector simulator, which provides noiseless, exact quantum state evolution without shot-based sampling noise. The QUBO formulation stage uses Qiskit Optimization 0.3.2 and Gurobi 10.0.1[10] for constructing and validating the optimization model. The SPIQ initialization phase runs in a separate environment using Qiskit Aer 0.13.3 and Qiskit Algorithms 0.3.1 for stabilizer simulation and circuit construction. Since current gate-based quantum hardware imposes strict limits on qubit count, connectivity, and coherence time, simulation allows us to isolate the effects of encoding choice, optimizer selection, and initialization strategy from hardware-induced errors. All classical computation, including QUBO construction, SPIQ search, and QAOA parameter optimization, is performed on a workstation with an Intel Core i9-10940X CPU (14 cores, 28 threads, 3.30 GHz) and 128 GB of RAM.

5 PRELIMINARY EVALUATION

5.1 Problem Instances

We are limited to analyzing 3- and 4-table join orders due to resource constraints. To evaluate the impact of SPIQ initialization, we select the following join problems to cover a wide range of predicate selectivities, cardinalities, and cross-product presence:

- (1) **Problem 1:** 3 relation join and 2 predicates with varied predicate selectivities. Cardinalities of the relations are 10, 15, and 20 respectively. There are 2 predicates, one between relations 1 and 2 with selectivity 0.1 and between relations

2 and 3 also with selectivity 0.1. Encoding this problem requires 6 qubits.

- (2) **Problem 2:** 4 relation join and 2 predicates with low predicate connectivity and possible cross products. Cardinalities of the relations are 10, 15, and 20, 30 respectively. There are 2 predicates - between relations 1 and 2 and between relations 3 and 4, each with selectivity 0.1. Encoding this problem requires 12 qubits.
- (3) **Problem 3:** 4 relation join and 6 predicates with high predicate connectivity and no cross products. Cardinalities of the relations are 10, 15, 20, and 30, respectively. There are predicates between every relation with selectivities ranging from 0.1 to 0.6. Encoding this problem requires 20 qubits.

5.2 Evaluation Metrics

We evaluate performance along three axes:

- **Cost ratio:** the fraction of sampled bitstrings that decode to join orders at each of the leading cost levels. For every configuration we report the sampling percentage of the top-3 most-sampled cost levels, ranked by cost, which reveals how probability mass is spread across join orders of differing cost.
- **Optimal ratio:** the fraction of sampled bitstrings that decode to the join order having least cost, compared between the first and final QAOA iterations.
- **Energy convergence:** the trajectory of the expectation value over QAOA iterations, measuring how quickly and reliably optimizer reduces the energy and whether it stabilizes near the ground-state energy.

For each problem, we compare randomly initialized QAOA (random parameter initialization) against SPIQ-initialized QAOA under identical optimizer and iteration settings.

5.3 Observations

Table 1: Top-three final-state cost distributions of evaluation problems

Problem	Randomly Initialized				SPIQ Initialized			
	fallback=false		fallback=true		fallback=false		fallback=true	
	Cost	pct	Cost	pct	Cost	pct	Cost	pct
P1	15	12.1%	15	46.3%	15	49.3%	15	49.4%
	30	56.0%	30	53.7%	30	50.6%	30	50.6%
	200	31.8%	–	–	200	0.127%	–	–
P2	900	18.2%	465	23.2%	465	50.0%	465	50.0%
	960	20.4%	660	24.4%	600	24.5%	960	50.0%
	1350	10.6%	960	42.5%	960	25.4%	–	–
P3	440	7.81%	39	14.1%	39	50.0%	39	74.5%
	510	11.7%	104	14.9%	84	25.5%	84	0.0391%
	600	8.41%	375	25.9%	240	24.4%	104	25.5%

Table 1 reports, for each of the three problems, the three most-sampled final-state cost levels under four settings: random initialization without fallback, random initialization with fallback, SPIQ initialization without fallback, and SPIQ initialization with fallback.

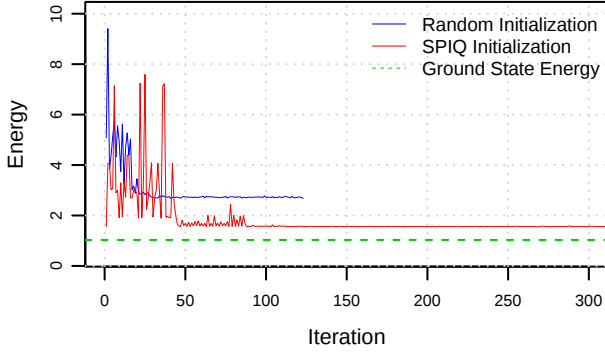


Figure 3: Energy convergence for Problem 1

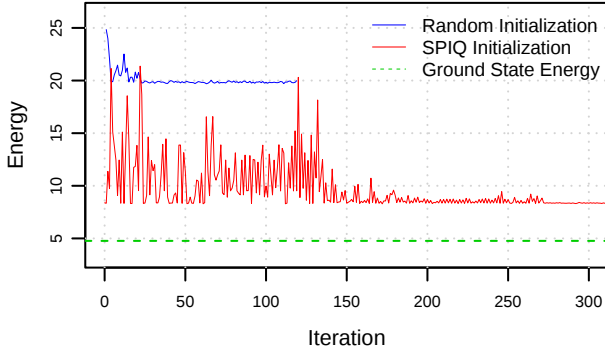


Figure 4: Energy convergence for Problem 2

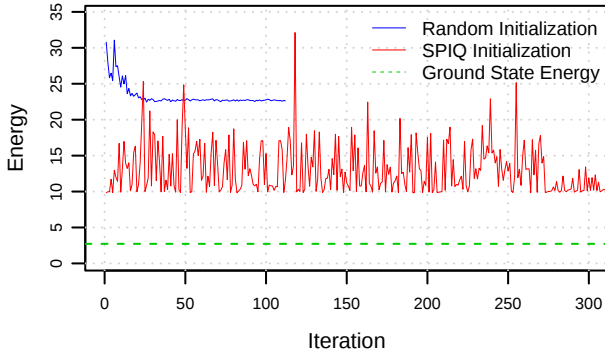


Figure 5: Energy convergence for Problem 3

Because only the top-3 sampled costs are listed, the table characterizes where the bulk of the sampled probability mass concentrates. Across these reported costs, SPIQ initialization generally shifts probability mass toward lower cost outcomes relative to random initialization, increasing the share of the optimal join order among the most-sampled levels.

With fallback disabled, the shift is clearest in Problems 2 and 3. Problem 1 improves more modestly: SPIQ achieves a 4 $\times$ improvement optimal cost frequency and nearly eliminates the high cost level 200 (from 31.8% to 0.127%), but cost 30 remains comparably sampled (50.6%), so the optimal order becomes prominent but not clearly dominating the reported costs.

With fallback enabled, SPIQ again increases the share of the optimal cost among the reported top 3 levels, with the magnitude differing by problem. The effect is strongest in Problem 3, where SPIQ achieves a 5 $\times$ improvement in optimal cost frequency. In Problem 2, SPIQ achieves a 2.2 $\times$ improvement in optimal cost frequency (23.2% to 50.0%). Problem 1 shows the mildest change, consistent with its small search space. Taken together, these results suggest that SPIQ initialization tends to concentrate the most-sampled costs on lower cost join orders.

As shown in Figures 3 and 4, our experimental results show that the energies obtained by QAOA do not yet reach the theoretical ground state, indicating that there is still significant room for improvement in optimization quality, parameter tuning, and hardware scalability. Although SPIQ-initialized QAOA converges to energy values closer to the theoretical ground state than randomly initialized QAOA, the decrease from the initial SPIQ energy to the converged energy is relatively limited. This suggests that SPIQ provides a better starting point, but the optimization process itself does not substantially improve the solution.

6 INSIGHTS AND FUTURE WORK

Our results indicate that SPIQ initialization consistently produces solutions with energies closer to the theoretical ground state than randomly initialized QAOA, suggesting that structured initialization improves optimization stability and guides the algorithm toward higher-quality join plans. However, the energy reduction achieved during the variational optimization itself is limited, as the join ordering landscape remains difficult to optimize in practice.

This points to two complementary directions for future work. On the optimization side, the primary goal is to improve QAOA’s ability to descend below the SPIQ starting point; we are hopeful that this trend will emerge as we move to more complex JOO problems where the variational phase has more room to contribute. On the initialization side, since the starting point is where most of the benefit currently arises, it is worth investigating whether more advanced initialization strategies can further sharpen the starting distribution and ease the subsequent optimization.

ACKNOWLEDGMENTS

This work was supported (in part) by the Google ML and Systems Junior Faculty Award, the Google JAX AI Stack Research Award, and the NVIDIA Academic Grant Program Award. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Number DE-SC0025633. This research was, in part, funded by the U.S. Government. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

REFERENCES

- [1] Tanweer Alam, Shamimul Qamar, Amit Dixit, and Mohamed Benaïda. 2020. Genetic Algorithm: Reviews, Implementations, and Applications. *arXiv preprint arXiv:2007.12673* (2020).
- [2] Dhanvi Bharadwaj, Yuewen Hou, Guang-Yi Li, and Gokul Subramanian Ravi. 2026. Scalable Clifford-Based Classical Initialization for the Quantum Approximate Optimization Algorithm. *arXiv preprint arXiv:2602.14327* (2026).
- [3] M. Cerezo, A. Arrasmith, R. Babbush, et al. 2021. Variational Quantum Algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.
- [4] Daniel J. Egger, Jakub Marecek, and Stefan Woerner. 2021. Warm-starting Quantum Optimization. *Quantum* 5 (2021), 479.
- [5] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm. *arXiv preprint arXiv:1411.4028* (2014).
- [6] Ahmed Fawzy Gad. 2021. PyGAD: An Intuitive Genetic Algorithm Python Library. *arXiv preprint arXiv:2106.06158* (2021).
- [7] Daniel Gottesman. 1998. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006* (1998).
- [8] Goetz Graefe. 1993. Query Evaluation Techniques for Large Databases. *Comput. Surveys* 25, 2 (1993), 73–169.
- [9] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. 212–219.
- [10] Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- [11] Yannis E. Ioannidis. 1996. Query Optimization. *Comput. Surveys* 28, 1 (1996), 121–123.
- [12] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. 2024. Quantum computing with Qiskit. <https://doi.org/10.48550/arXiv.2405.08810> arXiv:2405.08810 [quant-ph]
- [13] Tadashi Kadowaki and Hidetoshi Nishimori. 1998. Quantum Annealing in the Transverse Ising Model. *Physical Review E* 58, 5 (1998), 5355–5363.
- [14] Emmanouil Limnios, Markos Stergiopoulos, George T. Stamatiou, Efthymios Papageorgiou, Vasilis Efthymiou, Dimitrios Loupas, Dimitrios Tsourounis, Kostas Blekos, Aggelos Tsikas, Dimitris Plexousakis, Kostas Magoutis, Yannis Tzitzikas, and Haridimos Kondylakis. 2026. DEDALUS: A Quantum-Enhanced End-to-End Framework for Cost-Aware Join Order Optimization with Search Space Pruning. In *Proceedings of the 29th International Conference on Extending Database Technology (EDBT)*. 716–719. <https://doi.org/10.48786/EDBT.2026.63>
- [15] Yu Liu et al. 2025. A Demonstration of Q2O: Quantum-augmented Query Optimizer. *Proceedings of the VLDB Endowment* 18, 12 (2025).
- [16] Andrew Lucas. 2014. Ising Formulations of Many NP Problems. *Frontiers in Physics* 2 (2014), 5.
- [17] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, et al. 2018. Barren Plateaus in Quantum Neural Network Training Landscapes. *Nature Communications* 9, 4812 (2018).
- [18] Guido Moerkotte and Thomas Neumann. 2006. Analysis of Two Existing and One New Dynamic Programming Algorithm for the Generation of Optimal Bushy Join Trees without Cross Products. *Proceedings of the 32nd International Conference on Very Large Data Bases* (2006), 930–941.
- [19] Mohammad Mohammadi et al. 2022. QUBO Formulations for Combinatorial Optimization Problems: A Survey. *arXiv preprint arXiv:2202.03036* (2022).
- [20] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information*. Cambridge University Press.
- [21] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, et al. 2014. A Variational Eigenvalue Solver on a Photonic Quantum Processor. *Nature Communications* 5, 4213 (2014).
- [22] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.
- [23] Michael Schönberger, Immanuel Trummer, et al. 2023. Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware. *Proceedings of the ACM on Management of Data* 1, 2 (2023).
- [24] Michael Schönberger, Immanuel Trummer, et al. 2024. Quantum-Inspired Digital Annealing for Join Ordering. *Proceedings of the VLDB Endowment* 17, 3 (2024).
- [25] Patricia Griffiths Selinger, Morton M. Astrahan, Donald D. Chamberlin, Raymond A. Lorie, and Thomas G. Price. 1979. Access Path Selection in a Relational Database Management System. *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data* (1979), 23–34.
- [26] Matthias Steinbrunn, Guido Moerkotte, and Alfons Kemper. 1997. Heuristic and Randomized Optimization for the Join Ordering Problem. *The VLDB Journal* 6, 3 (1997), 191–208.
- [27] Davide Venturelli and Alex Kondratyev. 2019. Reverse Quantum Annealing Approach to Portfolio Optimization Problems. *Quantum Machine Intelligence* 1, 1 (2019), 17–30.
- [28] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, et al. 2020. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Physical Review X* 10, 2 (2020), 021067.