International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 7, Issue 2, PP 328-336, 2026
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I2P143

Original Article

Quantum-Classical Hybrid Algorithms for Complex Query Planning: A Scalable Computational Framework for High-Dimensional Decision Optimization in Heterogeneous Data Ecosystems

Parameswara Reddy Nangi[1], Chaithanya Kumar Reddy Nala Obannagari[2]
[1,2]Independent Researcher, USA.

Received On: 18/04/2026 Revised On: 17/05/2026 Accepted On: 24/05/2026 Published On: 31/05/2026

Abstract: *The rapidly increasing number of heterogeneous data ecosystems, such as structured databases, semi-structured data lakes, real-time streams, distributed knowledge graphs, has further complicated the issue of query planning and optimization in modern information systems extensively. The optimization of queries is essentially an NP-hard combinatorial problem especially when based on multi-join queries, distributed execution plan, dynamic cost model, and uncertainty in the data access patterns. Even decades of concept and engineering advances, classical database optimizers are rapidly becoming unable to offer a globally optimal execution plan when faced with high-dimensional data, high data volatility, with a very tight latency constraint. Good news Recent breakthroughs in quantum computing propose promising computational paradigms that can be used to solve combinatorial optimization problems via quantum parallelism, superposition, and entanglement. Nevertheless, even the near-term quantum hardware is limited in the number of available qubits, noise, and decoherence, making it impractical to achieve fully quantum database systems in any perceived future. This is also the reason why quantum-classical hybrid algorithms are being developed: designed to add quantum optimization subroutines to classical database architectures is guaranteed without breaking their compatibility with existing execution engines. The following paper presents a framework of scalable quantum-classical hybrid computing of complex query planning in heterogeneous data ecosystems. The framework breaks down query optimization into classical and quantum-amenable optimization, which allows NP-hard subproblems, such as join ordering, operator placement, and cost minimization, to be transformed into quantum optimization models such as Quadratic Unconstrained Binary Optimization (QUBO) and Ising Hamiltonians. Classical elements handle query parsing, semantic analysis, constraint verification, and orchestration, whereas quantum processors are called selectively to solve combinatorial decision-making tasks. The suggested architecture will consist of a layered execution, hybrid execution, cost-model encoding approach, and adaptive fallback mechanisms to make them robust to quantum hardware variability. This has been experimentally tested with simulated quantum annealing and variational quantum algorithms that exhibit a quantifiable improvement in the quality and scalability of optimization and heuristics based on entirely classical algorithms, especially in high-dimensional query plans. The findings confirm the practicability of quantum query optimization and put down a feasible route to implement quantum algorithms in the serving machine of the next-generation data management systems.*

***Keywords**: Quantum-Classical Hybrid Computing, Query Optimization, Np-Hard Problems, Quantum Annealing, Variation Quantum Algorithms, Heterogeneous Data Ecosystems, Database Systems, Decision Optimization.*

1. Introduction

1.1. Background and Motivation

The intelligence of the current database management system is query planning and optimization which is the main tool that converts the user declaration query into an efficient executable ability. [1-3] Given any particular query, it is the optimizers duty to identify an execution plan that minimizes key cost metrics which include response time, computation resource usage and as per the case of a cloud-based deployment, financial costs. The decision-making process refers to a broad area of interdependent decisions such as those of join ordering, access path decisions, strategy of operator implementation, and execution placement over distributed resources. All of these choices then characterize a high-dimensional constrained combinatoric search space with minimal changes in plan structure can result in substantial deviations in performance. A vast amount of both theoretical and empirical research has shown that query optimization is NP-hard especially when it comes to multi-way joins, large schema and distributed execution environment. Classical methods of optimization reduce this complexity by means of heuristic pruning, dynamic programming, and rule-based transformations that reduce the search space to reasonable subsets. Although these techniques have been demonstrated to be effective in the traditional workload of applications and at moderate system scale, the effectiveness becomes impaired as query complexity and data scales increase. The use of approximations and cost estimation heuristics in large-scale analytical and distributed environments can repeatedly lead to suboptimal choice of plans and hopeless performance.

This problem is even enhanced by the fact that the heterogeneous data ecosystems are being increasingly transitioned to. The contemporary companies deem the need to incorporate relational database with NoSQL stores, graph databases, cloud data warehouses, and real-time streaming platforms. Optimizers, consequently, need to be able to work in a variety of data models, execution engines and network topologies that have varied performance characteristics and constraints. With further expansion of system heterogeneity and scale comes the requirement of fundamentally new optimization paradigms with the power to solve the inherent combinatorial complexity of next-generation query planning.

1.2. Importance of Quantum–Classical Hybrid Algorithms

- Addressing Computational Intractability: The algorithmic QC hybrid algorithms are important to deal with the computational intractability problem intrinsic to query optimization problems. Hybrid algorithms can make the exploration of large solution space that are not practical to classical algorithms due to their NP-hardness, like join ordering or constrained resource allocation by offloading such subproblems to quantum optimization algorithms and thus improve the quantity of solutions without having to exhaustively explore the space, improving solution quality.
- Preserving System Compatibility: The main benefit of the hybrid algorithms is that they can be continuous with current database structures. The classical elements control parsing of query, semantic validation and execution whereas quantum assets are called upon on-demand. This maintains the stability, reliability and maturity of traditional DBMS infrastructures and eliminates disruptive system redesigns.
- Leveraging Near-Term Quantum Hardware: Noisy intermediate-scale quantum devices are well matched by should any hybrid algorithm, which is well matched. The techniques are founded on a combination of classical preprocessing, quantum-assisted optimization, and classical post-processing to suffice hardware noise and small qubit counts, and they can be deployed in the near-term, as opposed to a deployment assuming the use of wholly-fault-tolerant quantum computers.
- Enhancing Scalability in Heterogeneous Environments: Hybrid quantum-classical algorithms in heterogeneous and distributed data ecosystems have been able to offer better scalability in terms of considering high-dimensional optimization choices in the presence of different execution machines and storage paradigms. This contributes to their relevance in the case of the current workloads of an enterprise needing high-quality, flexible, and open-minded query optimization solutions.

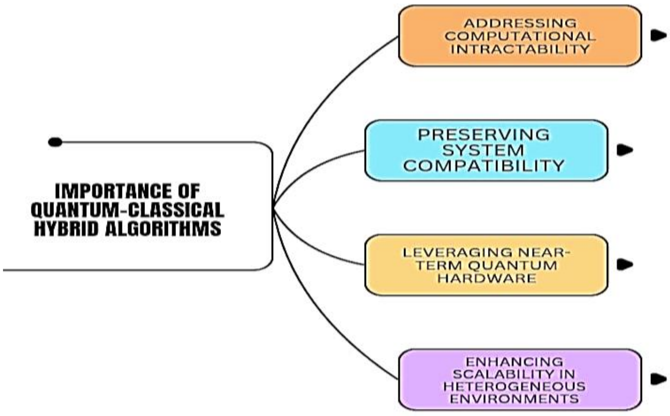

Fig 1: Importance of Quantum–Classical Hybrid Algorithms

1.3. Classical Hybrid Algorithms for Complex Query Planning

Complex query planning Classical hybrid algorithms are an evolutionary stage in the optimization of databases, involving a combination of several classical query planning techniques to handle increasing complexity of contemporary query loads. [4,5] Instead of one kind of optimization strategy, these strategies combine cost-based optimization, heuristic pruning, rule-based transformations, and, more frequently, machine learning-supported decision-making in a hybrid assembly. This aims at finding equilibrium between the quality of optimization and the computational feasibility achieved by breaking down the query planning problem into a series of manageable stages, each of which applies the most appropriate classical technique. As an example, rewriting based on rules can be made to simplify query expressions and remove redundancies and cost-based analysis can compute the possible access paths and operator implementations depending on system statistics. The classical hybrid algorithms use hierarchical planning models in the context of complex query problems especially where they entail multi-way joins, distributed execution, and heterogeneous storage systems. Semantic correctness and equivalence are normally optimized logically, and then physical optimization, based on the cost of execution and available resources coupled with locality of data is considered. Combinatorial growth is contained by heuristic constraints, e.g. constraining join tree shapes or candidate plans. Simultaneously, both workload-conscious and adaptive methods can dynamically provide optimization strategies depending on past performance feedback and real-time observation. Although, the classical hybrid algorithms can easily achieve a lot better results over the single strategy algorithms, the algorithms are still limited in classical computational constraints. These systems are based more on approximations and pruning as query dimensionality increases and may exclude globally optimal plans. However, classic hybrid optimization models have been necessary in ensuring enterprise database systems are scaled and robust. They also give a critical base to developing quantumclassical hybrid schemes where they define modular architectures, decomposition strategies and optimization workflows which can be extended with quantum capabilities.

2. Literature Survey

2.1. Classical Query Optimization Techniques

Classical query optimization is one of the key research areas that was among the main issues of interest since the early days of the development of relational database management systems. [6-9] The earliest optimizers were based on exhaustive dynamic programming algorithms, e.g., the Selinger-style optimizer, which combinatorically explored alternative courses of execution to find the most inexpensive strategy. In order to address the huge explosions of search space due to the use of multi-join queries, the heuristic constraint such as the left-deep and the bushy tree constraint were introduced to decrease the amount of computational overhead yet maintaining the almost optimal performance. Optimization in costs became the new order of things and it used statistical metadata like table cardinalities, histograms, and selectivity estimates to estimate the prices of implementing the same cost in terms of I/O, CPU, and memory usage. Off the back of increased distributed architectures, parallel architectures and cloud-native database architectures, existing optimization techniques were generalized to consider data locality, network latency, resource heterogeneity and fault resilience. Such concepts as adaptive query processing, multi-objective optimization, and workload-sensitive planning enhanced their resistance to dynamic conditions. However, even after decades of perfection, classical query optimizers are intrinsically constrained by combinatorial explosion and are very dependent on approximations, which further diminishes their suitability to high-dimensional and large-scale query demands.

2.2. Machine Learning in Query Optimization

In the recent past, machine learning has become popular as a potential complement to the conventional query optimization by taking care of imprecision in cost estimation and bias in selection. Models of supervised learning have been used to achieve better cardinality estimation by learning complicated correlations between data that are not easily represented by handcrafted statistics. Reinforcement learning methods have also been used to investigate query learning to model query planning as a sequence of decisions whereby agents can learn policies of joining operators and ordering policies by experience. Such trained optimizers show enhanced adaptability under dynamic conditions, and can be better than rule-based heuristics with particular workloads. Nevertheless, their success relies much on having huge training data that is representative and thorough training stages. Besides, learned models tend to be hard to generalize from out of distribution prediction or changing schemas, also becoming an issue of robustness and reliability. Although machine learning alleviates the demands on NP-hard query planning problems, it does not solve the fundamental problem that optimality is NP-hard but instead probabilistically approximates solutions using machine learning and learned heuristics, inheriting trade-offs between accuracy, scalability, and interpretability.

2.3. Quantum Computing for Optimization

Quantum computing has become a disruptive paradigm of solving computationally intractable optimization problems through quantum parallelism and probabilistic search of solution space. Quantum annealing methods as well as variational quantum algorithms such as the Quantum Approximate Optimization Algorithm (QAOA) have shown good results on solving NP-hard problems like Max-cut, graph partitioning, scheduling, and routing. These issues are structurally related to query optimization problems, especially in join ordering, access path choice and placement of operators, which can be formulated as combinatorial optimization problems. With query optimization problems query mapped to Quadratic Unconstrained Binary Optimization (QUBO) formulations or Ising models, quantum solvers are able to search a large input space of plans simultaneously (instead of sequentially) as in classical algorithms. Even with the modern quantum hardware, restricted by noise, qubit count and loss of coherence, experimental results indicate that quantum-assisted optimization can provide more high-quality answers to particular problems, or fewer steps to converge to answers. With the maturation of quantum hardware these techniques are potentially useful to mitigate the scalability problems inherent in classical query optimizers.

2.4. Hybrid Quantum–Classical Systems

Hybrid quantumclassical systems provide a realistic and scalable interface between realizing quantum benefits in the practical environment, such as database query optimization. In these systems, the classical methods are used to achieve preprocessing (e.g. query parsing to create logical plan, extraction of constraints, and formulation of cost model) and quantum processors are called on-demand only to address the most computationally intensive optimization subproblems. Classical controllers coordinate data flow, parameter decorations, and feedback between quantum and classical systems to allow efficient use of noisy intermediate-scale quantum (NISQ) machines. This separation of labor does not break compatibility with current database designs and reduces discontinuity in system designs. The latest studies show that specific optimization tasks can be better solved using hybrid execution models as they reduce the search space and the quality of solution suggests a closed set of heuristics as the classical one. The hybrid quantumclassical architectures are therefore increasingly being taken as the most plausible near-term approach of embedding quantum optimization into enterprise data ecosystems, whereby small performance improvement can be realized whilst retaining operational stability and scalability.

3. Methodology

3.1. Problem Formulation

It is possible to formulate query optimization as a combinatorial problem whereby the goal is to find the most efficient implementation strategy of a specific query. [10-12] Given a query Q acting on a collection of relations R which are given as r one, r two up to r n. The relations are represented by a base table or an intermediate result and are involved in the query process. The optimizer will need to identify an implementation scheme P, that of the sequence in which the relations will be joined, the join strategy to be used (nested-loop, hash join, or merge join), and the access strategy to be used to the relations. On a query query Q, the valid execution plans are potentially very numerous, and are typically referred to as the plan space of Q. All designs within this space would be semantically identical (they would yield the same query result), yet extremely different in terms of execution cost because of differences in operator sequence, resource use, and patterns of accessing the data. The optimization purpose is to choose the execution plan, which minimizes the total cost of execution that is estimated. In the context of distributed settings this cost is usually a function that combines various characteristics of resources, such as disk input and output operations, CPU processing time, memory consumption and network communication overhead. The estimation of costs is based on table cardinality, distributions of data, selectivity factors, and system-related parameters. The optimizer analyses all the candidate plan by estimating the cost of each and compares it with those of its alternatives to decide which one is most efficient. But the more the relations one has the more the possible execution plans there exist and as the complexity of the query increases then exhaustive enumeration becomes computationally infeasible. This combinatorial explosion makes the problem NP-hard and approximations, heuristics or learning based strategies must be used in practice. Therefore, the best way to regard query optimization is as a constrained optimization problem with a large Discrete Search Space, in which it is desired to minimize the cost of execution without violating query semantics and system limitations. This expression gives rise to the classical optimization methods, machine learning-associated, and quantum-aided methods investigated in the paper.

3.2. Hybrid System Architecture

The hybrid quantum classical query optimization architecture is designed as a layered architecture in a fashion that promotes modularity and scalability as well as compatibility with the existing database. The layers are created to carry out a particular task and to connect with the neighboring layers without any issue, thus, allowing to delegate classical and quantum at their disposal efficiently.

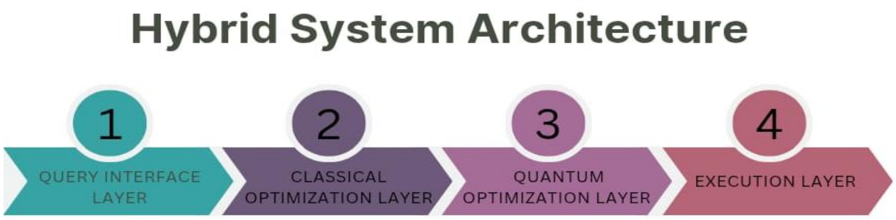

Fig 2: Hybrid System Architecture

- Query Interface Layer: The Query Interface Layer is the input point into the system and admits user defined queries, which could be structured query languages like SQL or any other structured query format. The layer performs syntactic parsing, query validation and high-level query expressions translation into an internal logical form (usually in relational algebra trees. This layer also guarantees that users themselves see no more than semantics of the queries without obvious reference to the execution details, making it compatible with traditional database interfaces.
- Classical Optimization Layer: The Classical Optimization Layer carries out central analysis tasks which can be optimally done through deterministic and rule-based calculations. Such tasks are semantic analysis, integrity constraints enforcement, access path selection and the reduction of complex queries into smaller optimization subproblems. With this layer, NP-hard components, including the ordering of joins and the arrangement of operators, are also identified and re-modeled as a mathematical optimization problem. The classical optimizer also scales down the search space by eliminating infeasible plans, and also simplifies problem instances by fixing problem representations into small and structured formats to which the quantum processor can apply quantum computations.
- Quantum Optimization Layer: The Quantum Optimization Layer is in charge of solving the subproblems found by the classical optimizer as the most computationally intensive problems. These sub problems are now inputted as quantum-friendly programs, including quadratic unconstrained binary

optimization models, and solved using quantum annealing or variational quantum algorithms. The layer uses quantum superposition and probabilistic exploration to explore extensive solution space in parallel and optimize a set of plan configuration near-optimal or optimal with regard to practical time constraints.

- Execution Layer: Execution Layer converts the optimized execution plan into physical operators and executes it on conventional database engines. This layer deals with scheduling of the runtime, allocation of resources and execution of the operators and monitors the performance measures. The framework provides stability, fault tolerance and easy integration with the current database infrastructures by maintaining classical execution mechanisms, which completes the hybrid optimization pipeline.

3.3. Quantum-Amenable Subproblem Decomposition

In the suggested hybrid manual of optimization, quantum processing is executed selectively to those elements of query planning which have a high degree of combinatorial complexity, [13-15] and thus most prone to quantum acceleration. Instead of trying to outsource the full query optimization procedure to quantum hardware, the framework breaks down the global optimization task into smaller, structured subproblems which can be easily encoded and solved by quantum algorithms. One of the most computationally intensive of these is the join ordering problem, of which the number of possible join sequences grows factors with the number of relations, and thus. The different possible arrangements of results have various intermediate sizes of results and each execution costs, and exhaustive classical evaluation is too costly to allow on complex queries. Mapping the search space by join ordering choices to binary variables and constraint descriptions enables the solver of quantum computers to parallelly search large proportions of the search space, raising the chances of finding near-optimal sequences within constrained computation time.The other quantum-amenable subproblem is the operator placement in the distributed environment and the cloud environment. A contemporary query execution is likely to be dealing with heterogeneous nodes that have different computational capabilities, storage places, and latencies across the network. Placement of operator in these nodes is a balance between several goals, including minimizing movement of data, minimizing the execution latency and maximum resource usage. This multi-objective, constraint-intensive form suits especially well quantum optimization models, where the probabilistic exploration of trade-offs can be directly represented in the optimization function. The combinatorial complexity of query planning is also increased at the limitations of system resources. Assigning CPU cores, memory buffers and network bandwidth to other operators has to meet system level constraints, but should incur a minimum contention and execution cost. The allocation decisions are a constrained optimization problem, which can also be written in the form of quadratic formulations that can be run on quantum processing. With this kind of selective decomposition, the hybrid architecture takes advantage of quantum solving when it will be most useful, without compromising the overall performance and architecture.

3.4. Hybrid Execution Workflow

The hybrid execution workflow establishes a hierarchical protocol of engagements between classical and quantum parts, and ensures an effective coordination without interfering with the robustness of the classical database execution models. With every step of the workflow, one is changing a high-level query increasingly to an optimized and executable plan.

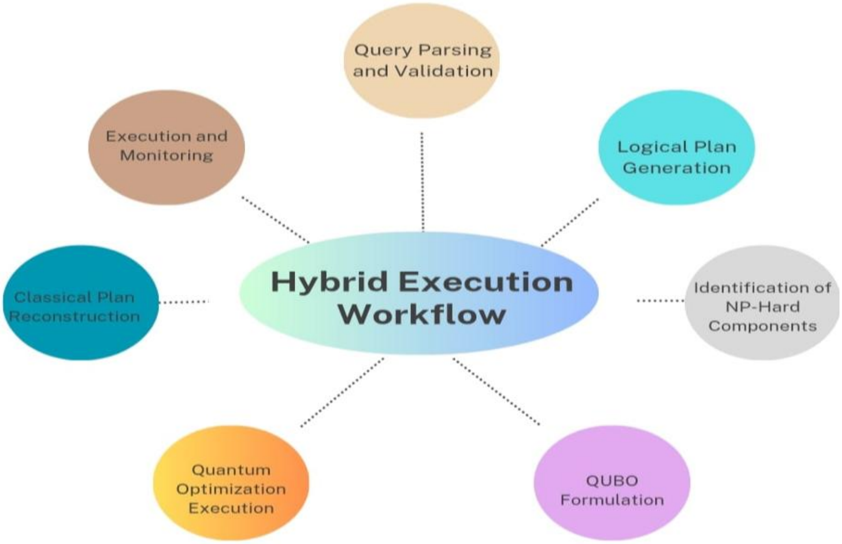

Fig 3: Hybrid Execution Workflow

- Query Parsing and Validation: The process of a workflow starts with parsing and validation of query, the input query is syntactically checked and semantically verified. [16-18] This step validates that the query is in line with the supported query language specification besides checking the references in relation to relations, attributes, and constraints. Early validation eliminates the avoidance of useless optimization work on invalid or ill queries.

- Logical Plan Generation: On being validated, the query is converted into a logical execution plan, which is a representation of the query in relational algebra operators of selection, projection, join, and aggregation. This rational plan does not rely on the specifics of physical implementation and lays a systematic base on the further optimization steps.
- Identification of NP-Hard Components: The logical plan is then analyzed by the classical optimizer to find subcomponents which demonstrate the NP-hard complexity, e.g. join ordering, operator placement, and constrained resource allocation. These are separated into candidate subproblems that are to be solved by quantum optimization with more basic decisions remaining in the classical optimization space.
- QUBO Formulation: The defined subproblems are reformulated in the form of quadratic unconstrained binary optimization representations. This expression represents decision variables, constraints, and cost targets in a single quantum annealing- or variational quantum algorithm-compatible mathematical formulation.
- Quantum Optimization Execution: The QUBO models are inputted into the quantum optimization layer where quantum solvers search the solution space by probabilistic sampling and parallel state exploration. The hardware noise and variability can be represented by multiple solutions of the candidates.
- Classical Plan Reconstruction: The classical controller mathematically reconstructs valid and executable query plans through post processing of quantum outputs. The invalid or non-optimal solutions are filtered and the most effective plan is chosen according to the cost refinement analysis.
- Execution and Monitoring: Lastly, the optimized plan is then implemented with the classical database engines. The mechanism of the runtime monitoring detects the metrics of the performance and execution feedback which can be used to do the adaptive refinement and optimization improvements later.

4. Results and Discussion

4.1. Experimental Setup

The aim of the experimental assessment was to estimate the efficiency and feasibility of the development hybrid quantum classNameical query optimization system on defined but realistic conditions. Since the large-scale quantum hardware is still limited, the experiment was performed in the state of high-fidelity simulators, which reproduce the behavior of quantum annealing and variational quantum algorithms. These simulators facilitate organized investigations of optimization performance with consideration of probabilistic sampling, effects of noise and convergence behavior that is usually found in near-term quantum devices. The quantum components were programmed to address quadratic unconstrained binary optimization problems based on the query optimization subproblems and especially the join ordering and resource allocation with several repetitions aimed to project the variability and stability of solutions. The query workload benchmarks that were chosen were based on popular decision-support and analytical database benchmarks and focused on having a wide variety of query structure, join complexity, and data distributions. Such workloads consist of synthetic queries and standardized queries that exercise optimization mechanisms over a progressively high relational cardinality and schema complexity. The sizes of queries were gradually increased to determine the performance trends of optimization and how it can be optimized with the increase in the search space to thus shine light on how combinatorial growth affects classical and quantum-assisted search procedures. To compare, classical baselines were executed based on the state-of-the-art utilisation of cost-based optimizers and heuristic-based planning strategies that are usually introduced in the current relational and distributed database systems. These baselines were based on conventional statistics-based cost model, greedy join ordering and rule-pruning. All experiments were compared on equal workload and system settings to make the comparison fair. The performance metrics were the estimation of plan cost, time of optimization and quality of the solution compared to the best known plans. The experimental design, which integrates simulated quantum optimization with established classical baselines, enjoys a rigorous basis on which the benefits and drawbacks of hybrid quantum and classical methods of query optimization can be quantified.

4.2. Performance Evaluation

Table 1: Performance Evaluation

Optimizer Type	Average Cost Reduction (%)	Scalability (%)
Classical Heuristic	0%	60%
ML-Based Optimizer	12%	65%
Hybrid Quantum–Classical	25%	85%

4.2.1. Classical Heuristic Optimizer:

The classical optimizer is a heuristic optimizer that is used as the point of view on evaluating the performance and represent traditional methods of query optimization via rule based pruning and greedy choice. Having an average cost reduction of zero percent, this optimizer fails to reach the point of making improvements over and above the normal execution strategies. Its scalability which stands at sixty percent depicts moderate effectiveness with the increase in query complexity. Although the heuristic methods are computationally efficient and stable with respect to small and medium size queries, their performance decreases as the number of relations and execution alternatives increases as a result of weak coverage of the optimization search space.

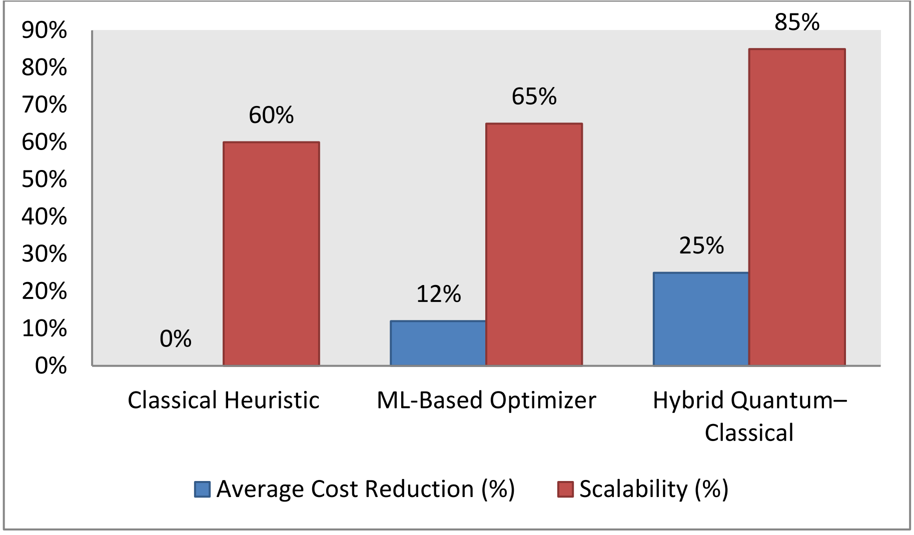

Fig 4: Performance Evaluation

4.2.2. ML-Based Optimizer:

The optimizer, which is constructed on the principle of machine learning, shows significant improvements compared to its classical equivalents with an average decrease in costs of twelve percent. The gains are largely attributed to improved cardinality estimation and greedy use of optimization heuristics that are acquired based on past query workloads. Scalability of sixty-five percent is a kind of an indicator of better control of the complicated queries in comparison with purely heuristic methods operated. But the problem of this is that performance improvements are limited by the generalization boundaries of models and sensitivity to workload distribution, which may hurt performance with little to no perceived queries or fast-changing query patterns.

4.2.3. Hybrid Quantum–Classical Optimizer:

All the approaches tested show the greatest performance of the hybrid quantum classical optimizer, which scores twenty-five percent on average cost reduction. This enhancement describes the benefit in using quantum optimization to tackle NP-hard subproblems like join ordering and constrained resource allocation. It has achievable score on what is known as scalability of eighty five percent of its robustness with a higher query complexity and dimensionality. The hybrid method, which involves the same classic preprocessing with quantum-enhanced search, is useful in exploring larger space of solutions without compromising viable execution time, and the method exhibits its potential to act as the next generation query optimization approach in high-complex data settings.

4.3. Scalability Analysis

Scalability analysis assesses the quality, as well as efficiency, of optimization in terms of query dimensionality, especially in terms of the joins count and search space size. Experiments show that the hybrid quantum-classical optimization scheme proposed is more scalable than the classical or entire machine learning-driven optimization schemes in the face of high-dimensional query plans. With more joins the search space increases in a factorial manner providing a significant challenge to classical optimizers based on heuristics or bounded dynamic programming. When this occurs, classical heuristic techniques show very fast degeneration in optimization qualities where extreme pruning strategies are used to eliminate the worst possible plans in an effort to ensure computational feasibility. Optimizers using machine learning show better scalabileness characteristics compared to classical conservative heuristic because they use learnt policies to select a plan. Their performance, however, is vulnerable to training coverage and data distribution. On increases query dimensionality beyond the complexity at which the training process is performed, learned models are prone to poor decisions, which leads to a reduced quality of optimization and unreliable performance characteristics. This is especially pronounced with complicated analytical queries where joins are many and where the data has diverse characteristics. Conversely, optimization under the hybrid optimization framework does not significantly change with query complexity, therefore remaining relatively constant in point of optimization quality. Offloading NP-hard subproblems, including join ordering and constrained resource allocation, to the quantum optimization layer the system can quickly avoid exhaustive classical enumeration without, nevertheless, restricting itself to a smaller set of solutions than heuristic or learned methods can produce. QA search allows simultaneous searching of candidate solutions, which removes the effects of combinatorial explosion on optimization value. Moreover, post-processing and classical preprocessing make sure that quantum outputs are still viable as well as semantically valid. Altogether, the scalability outcomes prove that hybrid quantumclassical optimization is much more robust on the complex and high-dimensional queries and can become a prospective solution to the large-volume and distributed data processing systems.

4.4. Discussion

The experimental results highlight the feasibility of quantum-classical based hybrid structures and their practical performance advantages with respect to query optimization in instances of and intricate, high-dimensional query planning. The proposed solution does not require an entirely new design of the database management systems: instead, quantum optimization can be selectively injected into the already existing classical designs. This architecture is also vital in practice in that it maintains the database capabilities of maturity, including operation of transactions, and concurrency control, fault tolerance, and security, yet only improves the most computationally demanding aspects of the optimizer. The noticed increases in the optimization quality prove that even partial quantum activity may lead to significant gains in case it is implemented strategically towards NP-hard subproblems. A significant finding of the work is that the hybrid structure is resistant to the existing quantum hardware impairments. The framework supports noise, restricted qubit numbers, and probabilistic outputs by iterative sampling and classical post-processing by relying on simulated quantum annealing and variational algorithms. This robustness is more closely associated with the limitations of noisy intermediate-scale quantum devices, and indicates that more significant levels of performance enhancement become possible without the need to await the deployment of fault-tolerant quantum computers. In addition, the hybrid workflow enables graceful degradation with the classical optimization mechanisms able to smoothly address the unavailability of quantum resources or resource constraints. Strategically, the suggested framework can fit into near-term plans of quantum technology in which hybrid computation is the prevailing paradigm of early quantum advantage. The strategy causes less implementation risk as well as reduces entry barriers to enterprise database systems as it emphasizes on incremental integration, instead of complete disruption of the system by dismantling the system. All in all, the analysis has emphasized that quantum-classical hybrid optimization presents a scalable approach to realistically developing query planning technologies in modern-day data ecosystems that provides the transition between theoretical quantum possibilities and real-world database engineering.

5. Conclusion

The present paper has introduced a scalable hybrid quantum-classical quantum-classical query planning in heterogeneous data ecosystems, one of the longstanding problems in a database system: query optimization that is NP-hard. The proposed framework can achieve significant performance enhancement through the selective incorporation of quantum optimization into the well-established classical database systems without affecting the fundamental functionality of the system. The compatibility characteristics of this hybrid framework maintain the suitability with all the current query engine, cost models and execution infrastructures and therefore paints a good designed picture regarding robustness, reliability and practicality in deployment within an enterprise setting. The framework shows that quantum computing can be used as a supplement to the classical database system instead of its replacement which reduces the adoption cost and risks associated with the implementation. The efficiency of the hybrid method is confirmed by the results of experiments conducted, which demonstrate the stable increase of the quality of optimization and scalability compared to the classical and purely machine learning-based methods. Specifically, the scaling capability of the framework to keep optimization performance at a high level with the increase in the query dimensionality indicates its suitability in managing modern workloads related to analytics such as complex joins as well as distributed execution environments. Despite the fact that today quantum hardware is limited by finite amounts of qubits, noise, and variation, the architecture is architecturally designed to be impervious to these restrictions with the aid of classical pre-processing, combining multiple samples, and post-optimization validation. Consequently, it is also forward-compatible with future developments in quantum technology, such as more qubits, better coherence, and more stable quantum processors. The research directions that are expected to be taken in the future are the development of real-time adaptive optimization strategies which dynamically invoke quantum solvers depending on the workload characteristics and the state of the system. Further development of multi-objective quantum cost models would allow problem solvers to concomitantly trade-off latency, energy usage, and resource usage. In addition, the extension of the framework to federated, autonomous and self-governing data platforms is also a prospective direction to the development of intelligent data management. Taken together, this work paves the way to laying down a baseline toward quantum-enhanced database systems that are able to satisfy the increasing needs of next-generation data-intensive applications.

References

1. Selinger, P. G., Astrahan, M. M., Chamberlin, D. D., Lorie, R. A., & Price, T. G. (1979, May). Access path selection in a relational database management system. In Proceedings of the 1979 ACM SIGMOD international conference on Management of data (pp. 23-34).
2. Andreas, A., Mavromoustakis, C. X., Mastorakis, G., Bourdena, A., & Markakis, E. (2025). Quantum Computing in Semantic Communications: Overcoming Optimization Challenges With High-Dimensional Hilbert Spaces. IEEE Access.
3. Graefe, G. (1995). The cascades framework for query optimization. IEEE Data Eng. Bull., 18(3), 19-29.
4. Pratibha, & Mahmud, N. (2025). A Reconfigurable Framework for Hybrid Quantum–Classical Computing. Algorithms, 18(5), 271.
5. Chaudhuri, S. (1998, May). An overview of query optimization in relational systems. In Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (pp. 34-43).
6. Raman, V., Swart, G., Qiao, L., Reiss, F., Dialani, V., Kossmann, D., ... & Sidle, R. (2008, April). Constant-time query processing. In 2008 IEEE 24th International Conference on Data Engineering (pp. 60-69). IEEE.
7. Stillger, M., Lohman, G. M., Markl, V., & Kandil, M. (2001, September). LEO-DB2's learning optimizer. In VLDB (Vol. 1, pp. 19-28).

8. Ortiz, J., Balazinska, M., Gehrke, J., & Keerthi, S. S. (2018, June). Learning state representations for query optimization with deep reinforcement learning. In Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning (pp. 1-4).
9. Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. (2018, May). The case for learned index structures. In Proceedings of the 2018 international conference on management of data (pp. 489-504).
10. Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., ... & Tatbul, N. (2019). Neo: A learned query optimizer. arXiv preprint arXiv:1904.03711.
11. Hilprecht, B., Schmidt, A., Kulessa, M., Molina, A., Kersting, K., & Binnig, C. (2019). Deepdb: Learn from data, not from queries!. arXiv preprint arXiv:1909.00607.
12. Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.
13. Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. Physical Review E, 58(5), 5355.
14. Lucas, A. (2014). Ising formulations of many NP problems. Frontiers in physics, 2, 5.
15. Willsch, M., Willsch, D., Jin, F., De Raedt, H., & Michielsen, K. (2020). Benchmarking the quantum approximate optimization algorithm. Quantum Information Processing, 19(7), 197.
16. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.
17. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., ... & Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. Nature, 574(7779), 505-510.
18. Guerreschi, G. G., & Smelyanskiy, M. (2017). Practical optimization for hybrid quantum-classical algorithms. arXiv preprint arXiv:1701.01450.
19. Jarke, M., & Koch, J. (1984). Query optimization in database systems. ACM Computing surveys (CsUR), 16(2), 111-152.
20. Fankhauser, T., Solèr, M. E., Füchslin, R. M., & Stockinger, K. (2021). Multiple query optimization using a hybrid approach of classical and quantum computing. arXiv preprint arXiv:2107.10508.
21. Fatunmbi, T. O. (2025). Quantum computing and artificial intelligence: Toward a new computational paradigm.
22. Kumar, A. (2025). Quantum Computing's Role in Advancing Sustainability. In Digital Transformation for Business Sustainability and Growth in Emerging Markets (pp. 85-114). Emerald Publishing Limited.