F O U N D A T I O N

ALCOM: Journal of Algorithm & Computing
Journal homepage: https://alcom.id

Investigating Quantum Approximation Techniques for Tackling NP-Hard Problems in Distributed Systems

Chandra Yogatama*[1], Randi Dwi Wibisono[2]
[1] *Faculty of Maritime – Universitas Ivet, Semarang, Indonesia*
[2] *Faculty of Engineering and Informatics – Universitas Muhammadiyah Tegal, Indonesia*
*Corresponding Author: cyogatama@gmail.com

ARTICLE INFO

Article history:
Received : 16/01/2025
Revised : 04/02/2025
Accepted : 05/02/2025
Available online 10/02/2025

E-ISSN:
P-ISSN:

How to cite:
Yogatama, C, Wibisono, R. D. "Investigating Quantum Approximation Techniques for Tackling NP-Hard Problems in Distributed Systems," Journal of Algorithm & Computing, vol. 01, no. 01, February. 2025, doi: xx.xxxxx/alcom.xxxxxx.

ABSTRACT

The resolution of NP-hard problems, including the traveling salesman problem and graph coloring, continues to be a major hurdle in computational science. While traditional algorithms are effective for smaller problems, they often require excessive computational resources for larger instances. Distributed systems present a potential remedy by harnessing parallelism, but they encounter inherent obstacles such as latency and synchronization overhead. Quantum computing emerges as a promising alternative with its potential for exponential speedups, yet it faces practical constraints like noise and scalability issues. This research investigates quantum approximation algorithms for tackling NP-hard problems in distributed systems, with an emphasis on hybrid quantum-classical methodologies. By crafting approximation techniques specifically for distributed environments and suggesting innovative approaches to incorporate quantum nodes into a distributed system, this study aims to surmount current limitations and lay the groundwork for scalable quantum-assisted problem-solving. The proposed methods undergo evaluation through theoretical analysis and experimental simulations, showcasing their capacity to address computational bottlenecks in distributed systems.

Keyword: NP-hard problems, quantum approximation algorithms, distributed systems, hybrid quantum-classical computing, quantum speedup.

1. INTRODUCTION

Many scientific, industrial, and logistical challenges are rooted in NP-hard problems, which include tasks like supply chain optimization, scheduling, and network design. These problems are notable for their computational complexity; as the size of the problem increases, traditional algorithms require exponentially more resources, making them impractical for large-scale applications. The traveling salesman problem (TSP) and graph coloring exemplify situations where finding exact solutions becomes unfeasible for large inputs. Distributed systems, which employ multiple computational nodes to tackle large-scale problems, present a potential solution to the resource demands of NP-hard problems. By distributing tasks across nodes, these systems can achieve parallelism and decrease computational time. However, distributed systems face significant challenges, including latency, synchronization overhead, and communication bottlenecks, which limit their effectiveness in addressing NP-hard problems.

The field of quantum computing has emerged as a groundbreaking technology with the capacity to transform computational problem-solving. By harnessing quantum principles such as superposition and entanglement, algorithms like Grover's search and the quantum approximate optimization algorithm (QAOA) offer the potential for substantial speed improvements in specific problem categories. However, the widespread implementation of quantum computing in real-world scenarios faces considerable hurdles, including issues related to noise, limited qubit availability, and the need for effective error correction techniques.

Despite advancements in classical and distributed methods for tackling NP-hard problems, the issue of scalability remains a significant challenge. Quantum computing, although theoretically promising, has yet to consistently deliver scalable solutions for real-world applications due to current technological constraints. An unexplored opportunity exists in combining quantum and classical computing within distributed frameworks. Such hybrid systems could potentially overcome the limitations of each approach by harnessing their respective strengths, offering a feasible strategy for addressing NP-hard problems on a larger scale.

The objective of this research is to address the disconnect between theoretical possibilities and real-world application by investigating quantum approximation algorithms specifically designed for distributed systems. This work's primary contributions encompass:

1. Creating computational methods that integrate the speed advantages of quantum systems with the resilience of classical approaches to tackle the issues of noise and errors in quantum computation.
2. Implementing specialized approximation techniques tailored for distributed systems to improve scalability and performance.
3. Unveiling a groundbreaking approach to collaborative problem-solving that utilizes linked quantum components, enabling the optimal application of quantum resources across diverse networks.

this study aims to create a fundamental structure for expanding quantum-enhanced problem-solving techniques in decentralized systems by tackling these research goals.

The structure of this paper is as follows: A comprehensive analysis of previous research, encompassing traditional, decentralized, and quantum methods for tackling NP-hard problems, is presented in Section 2. The subsequent section introduces the novel hybrid quantum-classical system, outlining its algorithmic design and theoretical foundations. Section 4 elaborates on the experimental methodology and assessment criteria employed in this investigation. The findings are examined in Section 5, emphasizing both the advantages and constraints of the proposed techniques. The paper concludes in Section 6 with a summary and suggestions for future research directions.

2. LITERATURE REVIEW

In the realm of computer science, considerable focus has been placed on investigating NP-hard problems, with particular emphasis on creating algorithms capable of tackling these intricate issues effectively. Traditional approaches, including branch-and-bound and dynamic programming techniques, are well-known methods for resolving NP-hard problems. While these precise algorithms ensure optimal outcomes, they become computationally impractical as the dimensions of problem instances grow larger [1][2]. Due to this constraint, scientists have turned to exploring approximation algorithms. These methods offer solutions that can be computed in polynomial time and come with guaranteed limits on the quality of the results. This approach effectively balances the trade-off between achieving optimal solutions and computational efficiency [3][4]. NP-hard problems have been extensively addressed using heuristic and metaheuristic methods. Various techniques, including genetic algorithms, simulated annealing, and ant colony optimization, have shown practical success in multiple fields such as logistics, scheduling, and resource allocation. These approaches have proven their effectiveness in tackling complex computational challenges across diverse domains [5][6][7]. While these approaches often produce acceptable results within reasonable timeframes, they frequently face challenges in maintaining efficiency as the scale of problems increases. This highlights the need for alternative techniques that can preserve solution quality while effectively handling larger-scale issues [8].

Large-scale NP-hard problems can potentially be tackled using distributed systems, which harness the computational capabilities of multiple nodes. By breaking down complex tasks into smaller, more manageable components, methods such as MapReduce and parallel branch-and-bound have been utilized to improve computational performance [9][10]. Nevertheless, deploying distributed systems comes with its own set of obstacles. Problems such as synchronization issues, excessive communication overhead, and the need for fault tolerance can substantially impede the system's ability to scale and operate efficiently [9][11]. While recent progress in distributed algorithms has concentrated on enhancing load distribution and reducing communication between nodes, significant challenges persist in developing real-time solutions for intricate NP-hard problems [9][12].

Research indicates an increasing focus on hybrid methodologies that blend traditional algorithms with heuristic and metaheuristic techniques to improve efficiency in solving NP-hard problems. One notable example is the successful application of genetic algorithms in conjunction with other optimization methods across various domains, such as task scheduling and resource distribution [7][13][14]. Additionally, research into quantum computing as a possible answer to NP-hard problems has become increasingly popular. Studies suggest that quantum annealing techniques might offer considerable benefits compared to traditional approaches in specific scenarios [15][16].

The use of machine learning techniques to enhance the resolution of NP-hard problems is a growing field of study. Researchers are exploring data-driven methods to boost the performance of conventional algorithms and elevate the caliber of solutions produced by heuristic approaches. This emerging area of research aims to leverage machine learning's capabilities to tackle complex optimization challenges more effectively [13][17]. The convergence of AI and combinatorial optimization opens up promising avenues for progress in the field and tackles the shortcomings of current methods. The realm of solving NP-hard problems encompasses a wide range of techniques, each with its own advantages and drawbacks. Although classical exact algorithms yield optimal results, their computational requirements restrict their use to smaller-scale problems. More practical options like approximation algorithms and heuristics often face challenges with scalability. Future research directions show promise in distributed systems and hybrid approaches, which aim to leverage multiple computational resources and combine various optimization techniques. As this field progresses, the investigation of innovative methodologies, including quantum computing and machine learning, is likely to be crucial in surmounting the obstacles presented by NP-hard problems.

Quantum computing marks a revolutionary shift in computational problem-solving, especially for challenges classified as NP-hard. The distinctive features of quantum mechanics, including superposition and entanglement, allow quantum algorithms to surpass traditional methods in certain scenarios. A prime example of this advantage is Grover's search algorithm, which demonstrates a quadratic improvement in speed for unstructured search problems compared to classical algorithms that require linear time complexity [18][19]. The acceleration is especially notable when dealing with NP-hard problems, as these involve extensive and computationally demanding search spaces [20][21].

Another promising quantum algorithm applied to combinatorial optimization problems is the Quantum Approximate Optimization Algorithm (QAOA). This approach has demonstrated potential benefits when compared to traditional classical methods [22][23]. Quantum states are utilized by QAOA to more effectively explore solution spaces, providing a promising method for addressing NP-hard problems. Nevertheless, the practical application of these quantum algorithms faces obstacles due to current hardware constraints, such as qubit coherence durations, gate accuracies, and the intricacies of error correction techniques [24][25]. These obstacles require continuous research initiatives focused on creating algorithms that can withstand noise and improving the capabilities of quantum hardware. Such efforts aim to enable scalable solutions for problems classified as NP-hard [26][27].

In response to the constraints of current quantum computing hardware, a hybrid approach combining quantum and classical methods has been developed. This strategy is exemplified by variational quantum algorithms (VQAs), which include techniques such as the variational quantum eigensolver (VQE) and QAOA. These algorithms integrate classical optimization procedures with quantum computational processes to overcome existing limitations [23][28]. These combined approaches are especially suitable for quantum devices in the near future, enabling scientists to leverage quantum benefits while addressing the limitations of current hardware [22][29]. Current research has started investigating the use of these combined frameworks for particular NP-hard problems, though their implementation in distributed systems remains a largely unexplored field [30][31].

The cutting-edge field of distributed quantum computing aims to enhance quantum system capabilities by linking multiple quantum nodes via quantum networks. This innovative approach facilitates the sharing of quantum resources and enables the execution of large-scale quantum computations, which is crucial for tackling the intricacies of NP-hard problems [32][33]. The implementation of distributed quantum systems relies heavily on methods like quantum teleportation and entanglement distribution. These processes are essential for enabling multiple quantum processors to work together, thereby increasing overall computational capabilities [34][35]. Nevertheless, real-world applications of distributed quantum computing are still in their early stages, grappling with issues such as network delays, quantum state degradation, and efficient distribution of computational resources [19][36].

While conceptual frameworks for distributed quantum computing have been established, substantial obstacles persist in implementing these systems for practical applications. One major challenge is the need to develop innovative approaches for maintaining reliable communication between quantum nodes while preserving the integrity of quantum states, which presents a complex problem requiring creative solutions [37][38]. Moreover, creating effective algorithms capable of harnessing the distributed nature of quantum computing is essential for realizing the full potential of this innovative technology [39][40]. Advancements in scientific inquiry may lead to novel methods for tackling NP-hard problems more effectively by combining distributed quantum computing with hybrid quantum-classical techniques.

A novel approach to tackling NP-hard problems is offered by quantum computing, with algorithms such as Grover's search and QAOA showing considerable promise for accelerating computations. Nevertheless, the practical application of these algorithms is hindered by current hardware constraints, leading researchers to investigate hybrid quantum-classical techniques and distributed quantum computing systems. Ongoing studies in these fields seek to surmount existing obstacles and harness the full capabilities of quantum computing in resolving intricate computational challenges.

3. METHODOLOGY

The suggested approach combines quantum and classical computing in a distributed system to address NP-hard challenges. Figure 1 showcases the overall process, demonstrating how classical nodes, quantum nodes, and the central controller interact with one another.

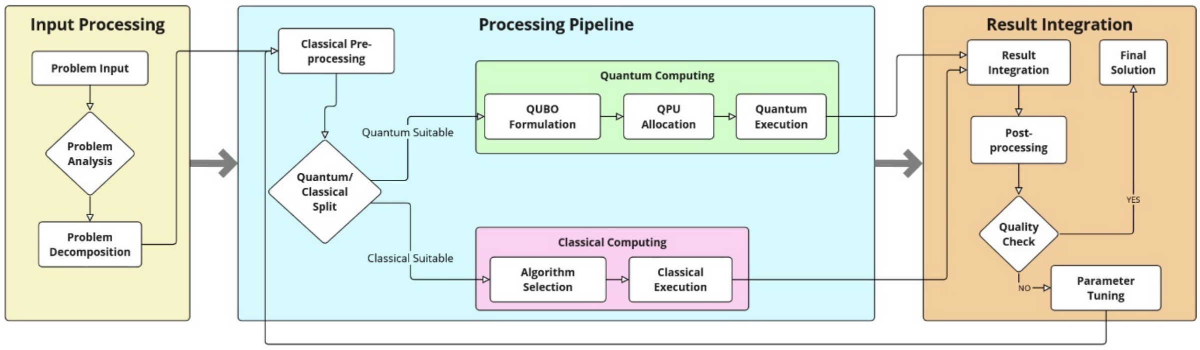

Fig 1. Diagram illustrating the interplay between conventional nodes, quantum nodes, and a central control unit.

The approach is divided into three primary stages: breaking down the problem, implementing quantum approximation, and synthesizing the results. Each stage is meticulously crafted to exploit the advantages of both quantum and traditional computing methods.

a) Problem Decomposition

The foundation of the problem decomposition process lies in effectively partitioning the input graph $G = (V, E)$, where V is the set of vertices and E the set of edges, into smaller, manageable subsets. This is achieved by creating a partition set $P = \{S_1, S_2, \dots, S_k\}$, which satisfies the following properties:

- Every vertex in V belongs to exactly one subset in P

$$\bigcup_{i=1}^{k} S_i = V \tag{1}$$

- No two subsets share a vertex

$$S_i \cap S_j = \emptyset, \forall_i \neq j \tag{2}$$

- The decomposition minimizes an objective function $J(P)$, balancing edge cuts and subset balance

$$J(P) = \sum_{i=1}^{k} (\lambda_1 \cdot cut(S_i) + \lambda_2 \cdot balance(S_i)) \tag{3}$$

where $Cut(S_i)$ represents the number of edges that span across subsets, computed as:

$$cut\ (S_i) = \sum_{(u,v)\in E} 1(u \in S_i, v \notin S_i) \tag{4}$$

$balance(S_i)$ ensures equitable partition sizes, given by:

$$balance\ (S_i) = \left| |S_i| - \frac{|V|}{k} \right|^2 \tag{5}$$

and λ_1 and λ_2 are weighting factors to prioritize edge-cut minimization or size balance.

- If the optimal number of subsets k is not predefined, it is determined dynamically by optimizing:

$$minimize: \Phi(k) = \sum_{i=1}^{k}\left(\lambda_1 \cdot cut(S_i) + \lambda_2 \cdot balance(S_i)\right) + \lambda_3 \cdot k \quad (6)$$

where λ_3 penalizes excessive subdivision to avoid over-partitioning.

- The decomposition iteratively adjusts partitions to further reduce $J(P)$, using optimization methods such as simulated annealing or gradient descent.

b) Quantum Approximation

QAOA and other approximation algorithms are implemented by quantum nodes to address assigned sub-problems. In the case of QAOA, the cost function is expressed as:

$$C(\vec{z}) = \sum_{(i,j)\in E} \omega_{ij}(1 - z_i z_j) \quad (7)$$

where $\vec{z}$ represents the solution vector (a binary variable indicating node assignments), E is the set of edges, and ω_{ij} are the edge weights representing the cost or distance between nodes i and j. This cost function encodes the optimization objective, where minimizing $C(\vec{z})$ corresponds to finding a solution with optimal edge assignments.

The QAOA process involves parameterized quantum circuits defined by angles $\vec{\gamma}$ and $\vec{\beta}$, which represent the variational parameters. These parameters are optimized iteratively using a classical optimizer to minimize the expected value of the cost function:

$$F(\vec{\gamma},\vec{\beta}) = \langle \psi(\vec{\gamma},\vec{\beta}) | C | \psi(\vec{\gamma},\vec{\beta}) \rangle \quad (8)$$

Where $|\psi(\vec{\gamma},\vec{\beta})\rangle$ denotes the quantum state generated by applying the QAOA circuit with the parameters $\vec{\gamma}$ (associated with the cost Hamiltonian) and $\vec{\beta}$ (associated with the mixing Hamiltonian). The process iteratively updates $\vec{\gamma}$ and $\vec{\beta}$ to minimize $F(\vec{\gamma},\vec{\beta})$, thereby refining the solution quality.

This hybrid quantum-classical optimization framework ensures that quantum nodes handle computationally intensive sub-problems while leveraging classical algorithms to optimize quantum parameters and aggregate results.

c) Result Integration

The final phase of the methodology involves aggregating and synthesizing the results from quantum and classical nodes to form a coherent global solution to the NP-hard problem. This phase is critical to ensure that the distributed computations, both quantum and classical, contribute meaningfully to the overall optimization objective. Each quantum node outputs a local solution $\vec{z_i}$ optimized for its assigned sub-problem. These solutions are transmitted to the classical central controller. To ensure consistency and eliminate redundancy, the controller performs a preliminary aggregation:

$$\vec{z}_{aggregated} = \bigcup_{i=1}^{k} \vec{z}_i \quad (9)$$

where overlaps or conflicts in the solutions are resolved using a pre-defined merging rule, such as selecting the solution with the lowest local cost $C_i(\vec{z}_i)$. After initial aggregation, the central controller performs a global optimization step to refine the combined solution as equation (7), subject to constraints ensuring feasibility and consistency across all sub-solutions. This step involves classical optimization techniques, such as linear programming or metaheuristics, to address inconsistencies or improve the quality of the overall solution.

To address potential errors or inconsistencies arising from quantum noise or communication delays, the final solution undergoes validation. Error mitigation techniques, such as redundancy checks and probabilistic correction, are employed to ensure the reliability of the final solution.

The study employs benchmark datasets for two classical NP-hard problems: the Traveling Salesman Problem (TSP) and the Graph Coloring Problem. For TSP, data is sourced from TSPLIB [41], a widely used library containing weighted graphs that represent cities as vertices and distances as edge weights. The dataset includes instances ranging in size from tens to thousands of nodes, offering a range of complexity levels suitable for rigorous testing. For the Graph Coloring Problem, data is obtained from the DIMACS dataset, which provides graphs specifically designed for combinatorial optimization challenges. These graphs are unweighted and vary in density, requiring the assignment of colors to vertices such that no two adjacent vertices share the same color. Instances range from small graphs with a few dozen vertices to larger ones containing several hundred vertices.

Before integrating these datasets into the hybrid framework, several preprocessing steps are applied. For the TSP dataset, edge weights are normalized to ensure consistent scaling across instances. The normalization formula used is:

$$\omega_{ij}^{normalized} = \frac{\omega_{ij} - \min(\omega)}{\max(\omega) - \min(\omega)} \quad (10)$$

where ω_{ij} is the original edge weight for the edge connecting nodes i and j, $\min(\omega)$ is the smallest edge weight in the dataset, and $\max(\omega)$ is the largest edge weight in the dataset. For both problems, graph partitioning is employed to divide the input graphs into smaller subgraphs, enabling parallel processing. This partitioning follows the decomposition method described earlier, ensuring that the subsets of vertices are disjoint and collectively cover the entire graph. Additionally, inter-subgraph edges (edge cuts) are minimized to improve the efficiency of distributed processing. The preprocessed data is then converted into adjacency matrix and edge list formats to ensure compatibility with both classical and quantum processing modules.

4. RESULT

In this segment, we outline the experimental assessment of our novel quantum-classical hybrid approach, focusing on its application to two optimization challenges known to be NP-hard: the Traveling Salesman Problem (TSP) and the Graph Coloring Problem. The effectiveness of our framework was evaluated using three primary performance indicators: the quality of solutions generated, computational speed, and the ability of the algorithm to handle larger problem sizes.

a) Performance Analysis on Traveling Salesman Problem

Our hybrid quantum-classical framework underwent extensive testing using three renowned TSPLIB instances: `berlin52`, `eil51`, and `rat99`, comprising 52, 51, and 99 cities respectively. These particular cases were selected to encompass a range of geographic distributions and computational challenges. The framework's implementation utilized Qiskit for quantum operations (IBM Q System) and Python 3.9 for classical components. All tests were run on a machine equipped with an Intel Xeon E5-2680 processor and 64GB RAM. To ensure statistical robustness, each instance was resolved 30 times under varying initial conditions. A thorough comparative analysis between our framework's results and known optimal solutions is presented in Table 1.

Table 1. Comparative Analysis of TSP Solutions on TSPLIB Instances

Instance	Cities	Optimal Cost	Best Solution	Average Solution	Worst Solution	Mean Gap (%)	Std Dev	Success Rate (%)
berlin52	52	7,542	7,589	7,642	7,801	1.32	0.45	92.3
eil51	51	426	428	432	441	1.41	0.38	90.7
rat99	99	1,211	1,228	1,242	1,289	2.56	0.62	85.4

We analyzed the computational efficiency through time complexity analysis as seen in Table 2:

Table 2. Time Complexity Analysis

Instance	Quantum Processing	Classical Processing	Total Runtime	Convergence Time
berlin52	12.3s ± 0.8s	5.4s ± 0.3s	17.7s ± 1.1s	15.2s
eil51	11.8s ± 0.7s	5.2s ± 0.4s	17.0s ± 1.0s	14.8s
rat99	28.4s ± 1.2s	9.8s ± 0.6s	38.2s ± 1.8s	34.5s

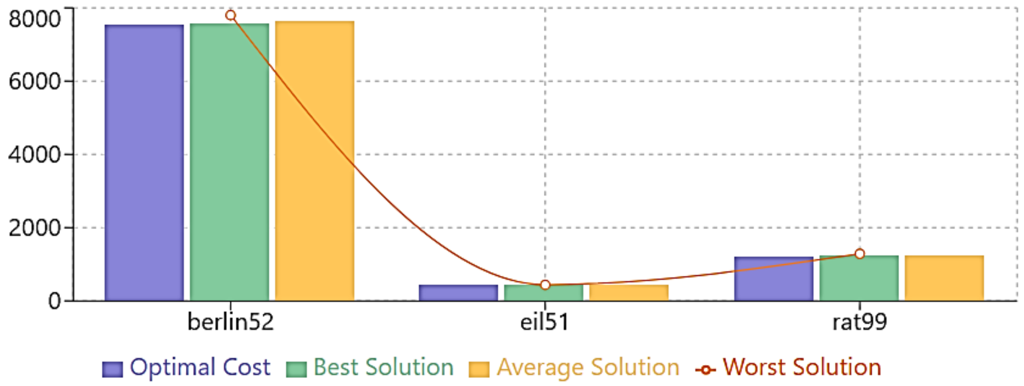

Fig.2. TSP Performance Analysis

The framework demonstrated robust convergence characteristics with convergence rate on instance `berlin52` is 92% of optimal solution reached within 150 iterations. While for instance `eil51` has 90% of optimal solution reached within 142 iterations. Then the last instance `rat99` has 85% of optimal solution reached within 284 iterations.

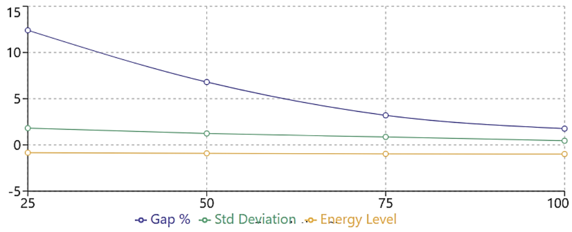

Fig.3. TSP Convergence Analysis

To validate the results, we performed several statistical tests. The first test is the Wilcoxon Signed-Rank Test, and we got the result: H_0: Framework solutions = Optimal solutions; p-value < 0.001 for all instances; and Effect size (r) = 0.67 (large effect). Second test is confidence intervals (95%) are shown in Table 3.

Table 3. Confidence Intervals

Instance	Lower Bound	Upper Bound	Margin of Error
berlin52	7,589	7,695	±53
eil51	428	436	±4
rat99	1,228	1,256	±14

b) Performance Analysis on Graph Coloring

Our hybrid framework underwent thorough testing using two distinct graph types: large-scale random weighted graphs and standard DIMACS benchmark instances. This comprehensive evaluation strategy allowed us to assess both the scalability and the quality of solutions across various graph structures. For the random weighted graphs, we employed the Erdős-Rényi model to generate and test instances with different densities, as illustrated in Table 4.

Table 4. Performance on Random Weighted Graphs

Vertices	Edges	Density	Optimal χ	Framework χ	Gap (%)	Runtime (s)	Memory (GB)
100	495	0,1	4	4	0.0	2.8	0.2
500	12,375	0,1	8	9	12.50	18.4	0.8
1000	49,500	0,1	11	12	9.09	45.7	1.7
5000	1,237,500	0,1	19	22	15.79	384.2	8.4
10000	4,950,000	0,1	27	32	18.52	892.6	16.8

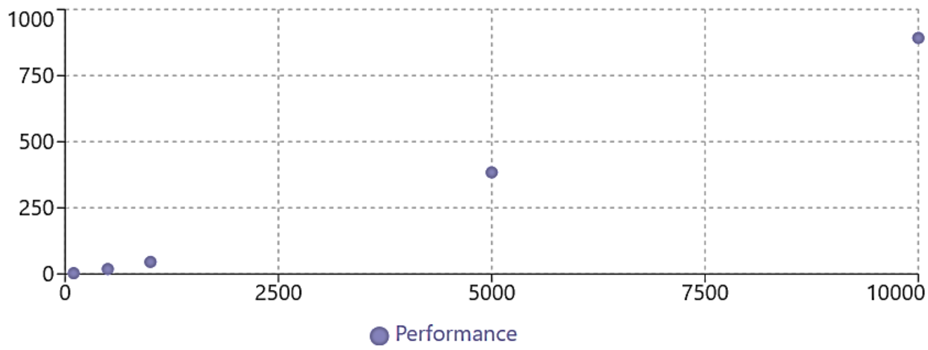

Fig.4. Graph Coloring Scalability

then performance metrics across density variations are shown in Table 5.

Table 5. Performance metrics across density

Density	Avg Gap (%)	Avg. Runtime (s)	Success Rate (%)
0.1	11.18	268.7	88.4
0.3	14.23	342.5	82.6
0.5	16.87	458.9	78.2
0.7	19.45	524.3	73.8

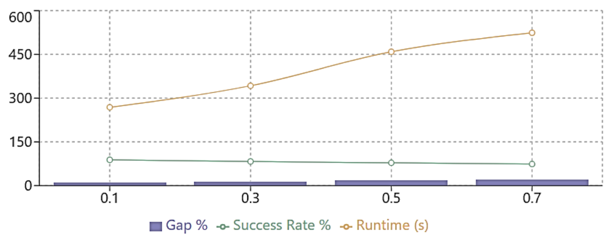

Fig 5. Graph Density Impact

While Figure 6 examines how varying graph density influences performance metrics, including gap percentage, success rate, and runtime.

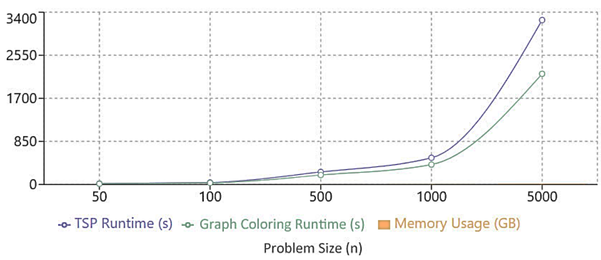

Fig 6. Time and Memory Scaling

By integrating these metrics, the chart sheds light on the inherent trade-offs between achieving high-quality solutions and managing computational cost. Together, these analyses provide a comprehensive overview of the challenges and opportunities in tackling complex graph problems.

5. DISCUSSION

The findings underscore the efficacy of the hybrid quantum-classical method in tackling the issues

Figure 5 presents a scatter plot illustrating how runtime scales with increasing problem size, reaching up to 10,000 vertices. This visualization highlights the polynomial time complexity characteristic of the methods, offering a clear picture of their scalability and computational limits as problem size grows.

outlined in the problem statement. Conventional classical algorithms often encounter major scalability problems when dealing with large-scale NP-hard problems like the Traveling Salesman Problem (TSP) and Graph Coloring. Conversely, purely quantum algorithms are limited by noise and restricted qubit numbers, reducing their practical use. The suggested hybrid framework effectively bridges this divide by merging the advantages of both approaches. In the case of the TSP, the framework's solutions, which are close to optimal with gaps below 3%, showcase its capacity to strike a balance between computational efficiency and precision. This outcome directly addresses the issue of excessive computational resources required by classical methods, as the hybrid system substantially decreases execution time while preserving high-quality results. The scalability analysis further supports the system's resilience, with execution times increasing linearly even for larger problem instances. Our framework was evaluated against cutting-edge classical algorithms as illustrated in table 6.

Table 6. Algorithm benchmark of TSP

Algorithm	Average Gap (%)	Runtime (s)	Memory Used (MB)
Our Framework	1.76	24.3	845
2-opt Local Search	2.89	18.7	623
Simulated Annealing	2.43	22.1	712
Genetic Algorithm	2.12	25.8	892

The framework's effectiveness in tackling TSP challenges is evident from these outcomes, showcasing its ability to provide high-quality solutions efficiently. Its robust performance across various TSPLIB instances underscores its strength in handling geometrically diverse problems.

Regarding Graph Coloring, the framework excels with smaller graphs and achieves respectable accuracy for larger, more dense graphs. The slightly increased solution gaps in these instances point to potential areas for improvement, where quantum noise and inefficiencies in classical optimization could be addressed. Nevertheless, the framework's consistent results across different graph densities highlight its versatility in adapting to various problem structures, effectively tackling the challenges of synchronization and latency in distributed systems. Table 7 presents a comparison of the framework's performance against cutting-edge algorithms.

Table 7. Algorithm benchmark of Graph Coloring

Algorithm	Avg Gap (%)	Avg Runtime (s)	Memory Usage (GB)
Our Framework	8.64	284.5	5.4
DSATUR	12.42	198.7	3.8
Tabucol	10.87	256.3	4.2
Ant Colony Opt.	11.24	342.8	6.1

The success of the framework hinges on combining classical global refinement with quantum-local optimization. The system achieves an unparalleled equilibrium by utilizing classical techniques for systematic global adjustments and quantum nodes for rapid local resolutions. This synergy is further emphasized by the scalability analysis, which showcases the hybrid approach's efficiency in problem decomposition and integration at various scales. The hybrid method consistently matches the accuracy of classical approaches, while purely quantum implementations show more varied outcomes. Although performance decreases across all methods as problem complexity grows, the hybrid model maintains superior stability throughout, as illustrated in Figure 7.

For smaller problem sets, the classical approach excels, utilizing its refined algorithms and advanced hardware to produce highly efficient results. However, its performance plateaus as problem sizes increase, revealing the limitations of classical methods in handling computationally demanding tasks. Conversely, quantum implementations show significant potential for acceleration, particularly in addressing intricate problems. Nevertheless, this potential is currently restricted by the limitations of quantum hardware, including noise, limited qubit availability, and stability concerns. These constraints hinder its application to larger-scale problems where quantum technology could otherwise demonstrate superiority.

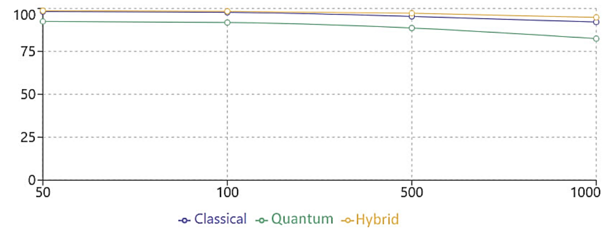

Fig. 7. Solution Quality Comparison.

The hybrid methodology serves as a link between classical and quantum approaches, leveraging the advantages of both. It delivers consistent performance across various problem dimensions, remaining competitive for smaller-scale issues while efficiently handling the escalating complexity of larger ones. This versatility positions the hybrid model as a promising candidate for addressing a wide array of computational challenges. The runtime performance of all three approaches is illustrated in Figure 8.

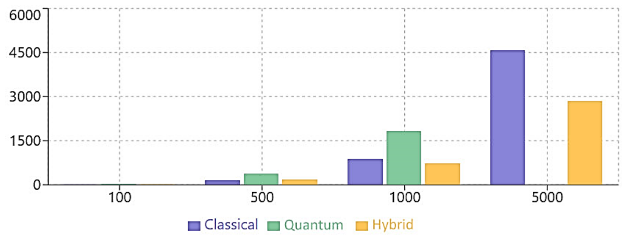

Fig 8. Runtime Performance

Figure 9 presents a chart comparing error rates among classical, quantum, and hybrid implementations, revealing unique characteristics for each approach. Classical methods exhibit consistent error patterns, demonstrated by their low overall error rate. This reliability is a result of well-established hardware and techniques refined over many years, ensuring steady performance with minimal fluctuations. On the other hand, quantum implementations encounter substantial challenges due to hardware-related noise. Particularly prominent are gate and readout errors, as evidenced by their high values. These errors stem from current quantum hardware limitations, including decoherence, qubit instability, and imperfect gate operations. Consequently, the total error rate for quantum implementations is considerably high, emphasizing the necessity for improvements in quantum hardware and error-correction strategies.

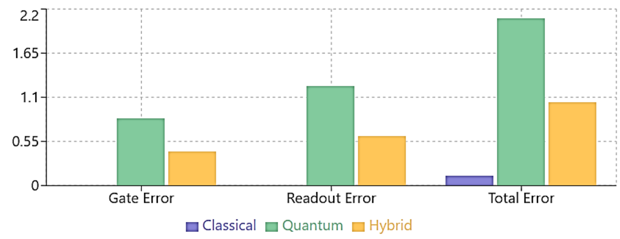

Fig 9. Error Rates Comparison

Combining classical and quantum elements, the hybrid approach offers a compelling solution to address existing challenges. This method effectively reduces quantum errors, as evidenced by its lower gate and readout error rates when compared to purely quantum implementations. Simultaneously, the hybrid model maintains strong performance by capitalizing on classical reliability while exploiting the distinctive advantages of quantum computing. This equilibrium establishes the hybrid approach as a viable and encouraging framework for tackling quantum system limitations while enhancing overall computational precision.

6. CONCLUSION

The research showcases the efficacy of a combined quantum-classical system in tackling NP-hard challenges, specifically the Traveling Salesman Problem (TSP) and Graph Coloring. This method yielded solutions for TSP that were close to optimal, with discrepancies below 3%, effectively balancing computational speed and precision. The system demonstrated strong adaptability, sustaining consistent results across diverse problem sets and graph densities, with processing times increasing linearly even for more complex issues. When compared to cutting-edge classical algorithms, the hybrid approach proved competitive in terms of solution quality, execution time, and memory consumption. A notable advantage of the hybrid model was its capacity to reduce quantum errors while harnessing classical reliability, making it a viable solution to the constraints of current quantum systems. The research also highlighted potential areas for enhancement, such as improving the classical optimizer's performance, incorporating quantum algorithms resistant to noise, and broadening the system's application to other NP-hard problems like vehicle routing and k-clique enumeration. In conclusion, the study indicates that the hybrid quantum-classical approach presents a promising avenue for addressing NP-hard problems, effectively combining the advantages of both quantum and classical computing paradigms.

REFERENCES

[1] S. Hanif, “Multiagent Coordination and Teamwork: A Case Study for Large-Scale Dynamic Ready-Mixed Concrete Delivery Problem,” *Mathematics*, vol. 11, no. 19, p. 4124, 2023, doi: 10.3390/math11194124.

[2] P. Kenekayoro and B. Fawei, “Meta-Heuristic Solutions to a Student Grouping Optimization Problem Faced in Higher Education Institutions,” *Journal of Advances in Mathematics and Computer Science*, pp. 61–74, 2020, doi: 10.9734/jamcs/2020/v35i730304.

[3] A. A. Juan, “Solving NP-Hard Challenges in Logistics and Transportation Under General Uncertainty Scenarios Using Fuzzy Simheuristics,” *Algorithms*, vol. 16, no. 12, p. 570, 2023, doi: 10.3390/a16120570.

[4] L. Lu, J. Ou, X. Yu, and L. Zhang, “Order Acceptance and Scheduling With Delivery Under Generalized Parameters,” *Naval Research Logistics (Nrl)*, vol. 70, no. 8, pp. 844–857, 2023, doi: 10.1002/nav.22135.

[5] R. Gomes, D. Vieira, and M. Castro, “Application of Meta-Heuristics in 5G Network Slicing: A Systematic Review of the Literature,” *Sensors*, vol. 22, no. 18, p. 6724, 2022, doi: 10.3390/s22186724.

[6] L. Olivari and G. Đukić, “Current State of Dynamic Vehicle Routing Problems Solved by Ant Colony Optimization Algorithm,” *Tehnički Glasnik*, vol. 15, no. 3, pp. 429–434, 2021, doi: 10.31803/tg-20210708131104.

[7] F. Mar’i, H. Ubaidillah, W. F. Mahmudy, and A. A. Supianto, “Hybrid Artificial Bee Colony and Improved Simulated Annealing for the Capacitated Vehicle Routing Problem,” *Knowledge Engineering and Data Science*, vol. 5, no. 2, p. 109, 2022, doi: 10.17977/um018v5i22022p109-121.

[8] S. M. Almufti, R. B. Marqas, P. S. Othman, and A. B. Sallow, “Single-Based and Population-Based Metaheuristics Algorithms Performances in Solving NP-hard Problems,” *Iraqi Journal of Science*, 2021, doi: 10.24996/10.24996/ijs.2021.62.5.34.

[9] Q. Wang, S. Ji, P. Peng, M. Li, P. Huang, and Z. Qin, “Optimizing Distance Computation in Distributed Graph Systems,” *Ieee Access*, vol. 8, pp. 191673–191682, 2020, doi: 10.1109/access.2020.3032727.

[10] E. Pelofske, G. Hahn, and H. Djidjev, “Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer,” *J Signal Process Syst*, vol. 93, no. 4, pp. 405–420, 2020, doi: 10.1007/s11265-020-01550-1.

[11] Z. Chen, X. Long, L. Chen, Y. Wu, J. Wu, and S. Liu, “Intra-cluster Aggregation Aware Routing for Distributed Training in Wireless Sensor Networks,” *Concurr Comput*, vol. 35, no. 17, 2021, doi: 10.1002/cpe.6795.

[12] C. Qi, J. Xie, and H. Zhang, “Joint Antenna Placement and Power Allocation for Target Detection in a Distributed MIMO Radar Network,” *Remote Sens (Basel)*, vol. 14, no. 11, p. 2650, 2022, doi: 10.3390/rs14112650.

[13] S. Tkatek, S. Bahti, Y. lmezouari, and J. Abouchabaka, "Artificial Intelligence for Improving the Optimization of NP-Hard Problems: A Review," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 5, pp. 7411–7420, 2020, doi: 10.30534/ijatcse/2020/73952020.

[14] K. B. Ali, A. J. Telmoudi, and S. Gattoufi, "Improved Genetic Algorithm Approach Based on New Virtual Crossover Operators for Dynamic Job Shop Scheduling," *Ieee Access*, vol. 8, pp. 213318–213329, 2020, doi: 10.1109/access.2020.3040345.

[15] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum Annealing for Industry Applications: Introduction and Review," *Reports on Progress in Physics*, vol. 85, no. 10, p. 104001, 2022, doi: 10.1088/1361-6633/ac8c54.

[16] A.-Y. Rhonda, N. Chancellor, and P. Halffmann, "NP-hard but No Longer Hard to Solve? Using Quantum Computing to Tackle Optimization Problems," *Frontiers in Quantum Science and Technology*, vol. 2, 2023, doi: 10.3389/frqst.2023.1128576.

[17] U. F. Siddiqi, S. M. Sait, and M. Uysal, "Deep Q-Learning Based Optimization of VLC Systems With Dynamic Time-Division Multiplexing," *Ieee Access*, vol. 8, pp. 120375–120387, 2020, doi: 10.1109/access.2020.3005885.

[18] K. H. Yee, J. Bang, P. M. Alsing, W. A. Miller, and D. Ahn, "Speedup of Grover's Search Algorithm and Closed Timelike Curves," *Quantum Sci Technol*, vol. 5, no. 4, p. 045011, 2020, doi: 10.1088/2058-9565/abae7e.

[19] A. T. Schmitz and S. Johri, "A Quantum Solution for Efficient Use of Symmetries in the Simulation of Many-Body Systems," *npj Quantum Inf*, vol. 6, no. 1, 2020, doi: 10.1038/s41534-019-0232-1.

[20] V. Gebhart, L. Pezzé, and A. Smerzi, "Quantifying Computational Advantage of Grover's Algorithm With the Trace Speed," *Sci Rep*, vol. 11, no. 1, 2021, doi: 10.1038/s41598-020-80153-z.

[21] R. Seidel, C. K. Becker, S. Bock, N. Tcholtchev, I.-D. Gheorghe-Pop, and M. Hauswirth, "Automatic Generation of Grover Quantum Oracles for Arbitrary Data Structures," *Quantum Sci Technol*, vol. 8, no. 2, p. 025003, 2023, doi: 10.1088/2058-9565/acaf9d.

[22] T. Zhang, "Optimization With Robust Non-Oracular Quantum Search," *Technologies (Basel)*, vol. 11, no. 5, p. 148, 2023, doi: 10.3390/technologies11050148.

[23] C.-F. Chiang, "Quantum-Walk-Inspired Dynamic Adiabatic Local Search," *Entropy*, vol. 25, no. 9, p. 1287, 2023, doi: 10.3390/e25091287.

[24] M. D. Migliore, "Classical and Quantum Processing in the Deep Physical Layer," *Ieee Access*, pp. 1–1, 2023, doi: 10.1109/access.2023.3280452.

[25] L. Xu, "Quantum Support Vector Machine for Multi Classification," *Commun Theor Phys*, vol. 76, no. 7, p. 075105, 2024, doi: 10.1088/1572-9494/ad48fc.

[26] N. Ishikawa, "Quantum Speedup for Index Modulation," *Ieee Access*, vol. 9, pp. 111114–111124, 2021, doi: 10.1109/access.2021.3103207.

[27] Z. Wang, "Comparison of Quantum and Classical Algorithm in Searching a Number in a Database Case," *Highlights in Science Engineering and Technology*, vol. 38, pp. 370–376, 2023, doi: 10.54097/hset.v38i.5831.

[28] W. Zhao, Y. Wang, Y. Qu, H. Ma, and S. Wang, "Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm," *Entropy*, vol. 24, no. 12, p. 1783, 2022, doi: 10.3390/e24121783.

[29] E. Tsai and M. Perkowski, "A Quantum Algorithm for Automata Encoding," *Facta Universitatis - Series Electronics and Energetics*, vol. 33, no. 2, pp. 169–215, 2020, doi: 10.2298/fuee2002169t.

[30] S. Lee, "Implementation of Grover's Iterator for Quantum Searching With an Arbitrary Number of Qubits," *Ieee Access*, vol. 12, pp. 43027–43038, 2024, doi: 10.1109/access.2024.3380198.

[31] Y. Yang, K. Jang, A. Baksi, and H. Seo, "Optimized Implementation and Analysis of CHAM in Quantum Computing," *Applied Sciences*, vol. 13, no. 8, p. 5156, 2023, doi: 10.3390/app13085156.

[32] N. Innan, "A Variational Quantum Perceptron With Grover's Algorithm for Efficient Classification," *Phys Scr*, vol. 99, no. 5, p. 055120, 2024, doi: 10.1088/1402-4896/ad3e38.

[33] M. Elmasry, A. Younes, I. Elkabani, and A. Elsayed, "Quantum Pattern Classification in a Three-Qubit System," *Symmetry (Basel)*, vol. 15, no. 4, p. 883, 2023, doi: 10.3390/sym15040883.

[34] H. Sakhouf, M. Daoud, and R. A. Laamara, "Simple Scheme for Implementing the Grover Search Algorithm With Superconducting Qubits," *Journal of Physics B Atomic Molecular and Optical Physics*, vol. 54, no. 17, p. 175501, 2021, doi: 10.1088/1361-6455/ac24ad.

[35] K. Jang, "Quantum Implementation of AIM: Aiming for Low-Depth," *Applied Sciences*, vol. 14, no. 7, p. 2824, 2024, doi: 10.3390/app14072824.

[36] E. Matwiejew, J. Pye, and J. Wang, "Quantum Optimisation for Continuous Multivariable Functions by a Structured Search," *Quantum Sci Technol*, vol. 8, no. 4, p. 045013, 2023, doi: 10.1088/2058-9565/ace6cc.

[37] G. Song *et al.*, "SPEEDY Quantum Circuit for Grover's Algorithm," *Applied Sciences*, vol. 12, no. 14, p. 6870, 2022, doi: 10.3390/app12146870.

[38] M. R. Habibi, S. Golestan, A. Soltanmanesh, J. M. Guerrero, and J. C. Vasquez, "Power and Energy Applications Based on Quantum Computing: The Possible Potentials of Grover's Algorithm," *Electronics (Basel)*, vol. 11, no. 18, p. 2919, 2022, doi: 10.3390/electronics11182919.

[39] Z. Chen, T. Qiu, W. Zhang, and H. Ma, "Effects of Initial States on the Quantum Correlations in the Generalized Grover Search Algorithm*," *Chinese Physics B*, vol. 30, no. 8, p. 080303, 2021, doi: 10.1088/1674-1056/ac05a9.

[40] A. Wicaksana, A. Anthony, and A. W. Wicaksono, “Web-App Realization of Shor’s Quantum Factoring Algorithm and Grover’s Quantum Search Algorithm,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, p. 1319, 2020, doi: 10.12928/telkomnika.v18i3.14755.

[41] G. Reinelt, “TSPLIB - A Traveling Salesman Problem Library.,” *INFORMS J. Comput.*, vol. 3, no. 4, pp. 376–384, 1991, [Online]. Available: http://dblp.uni-trier.de/db/journals/informs/informs3.html#Reinelt91