Accelerating NP-Hard Optimization via Quantum-Inspired Classical Algorithms

Dr.Mukesh Krishnan[1], Dr.Saravanan[2]
[1]Professor, Department of Networking and Communications, SRM Institute of Science and Technology, Chennai, India.
[2]Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, Chennai, India.

Article Info

Article history:

Received Jan 8, 2025
Revised Feb 12, 2025
Accepted Mar 10, 2025

Keywords:

Quantum-Inspired Algorithms
NP-Hard Optimization
Classical Optimization
Heuristic Methods
Traveling Salesman Problem (TSP)
Knapsack Problem
Max-Cut Problem

ABSTRACT

A primary challenge of computational science is solving NP-hard optimization problems since they are too complicated and cannot be effectively solved deterministically in large instances. Even though quantum computing can solve these problems with the help of its key properties quantum parallelism and tunneling, the modern hardware is yet not sufficiently developed. In this case, QICAs are considered as a new method in which the key aspects of quantum mechanics, such as superposition, entanglement, interference and tunneling are replicated in classical, non-quantum computers. It aims to attain rapid optimization of NP-hard problems through a hybrid approach that combines cartography-based exploration, variable state reduction with tensor networks as well as an inclusion of a variant-based approach inspired by QAOA. A number of standard problems are used to test the model, e.g., the Traveling Salesman Problem (TSP), the 0/1 Knapsack Problem and Max-Cut Problem. Genetic Algorithms, Simulated Annealing and Ant Colony Optimization trials all concur that QICA converges to superior solutions much faster and it can nonetheless handle bigger-sized problems. More precisely, the algorithm provides better approximation results as well as lesser computing power requirements, thus demonstrating the effectiveness of quantum-inspired methods in the case of classical systems. They establish the fact that QICAs have the potential to solve challenging optimization problems, particularly in the cases where quantum machines are absent or inaccessible. The way of solving the problem represented in the study is quite innovative and serves as a bridge between the theory of quantum and practical implementation in the optimization field.

Corresponding Author:

Dr.Mukesh Krishnan,
Department of Networking and Communications,
SRM Institute of Science and Technology, Chennai, India.

1. INTRODUCTION

Optimization problems that are NP-hard form the basis of a wide variety of significant problems in the fields of logistics, finance, supply chain management, bioinformatics, telecommunications and artificial intelligence. These problems that comprise Traveling Salesman Problem (TSP), 0/1 Knapsack Problem and Max-Cut, become significantly more complicated with the increase in the problem size and the exact solutions cannot be done in poly-time [4]. The majority of the past solutions of dealing with the NP-hard problems incorporate the use of classical and metaheuristic algorithms such as GA, SA, Tabu Search and Particle Swarm Optimization to trade-off between the running time of an algorithm and its preciseness. Such techniques scale up to reasonable solutions but in many cases cannot scale further, encounter premature convergence and do not exhaust the set of possible solution options in large or difficult cases [5,6].

Quantum computing with its principal ideas of superposition, entanglement and tunneling can resolve some of these problems. Several algorithmic efforts, including Grover search and the Quantum Approximate Optimization Algorithm (QAOA) can seemingly operate significantly quicker than their classical counterparts. The current state of practical quantum computing is currently inhibited by an immature state of quantum hardware, its vulnerability to noise, decoherence and the relatively small number of accessible qubits. That is why quantum optimization is not yet ready to be applied by the majority of people or to most of the uses [7,8].

In light of these hindrances, this effort introduces Quantum-Inspired Classical Algorithms (QICAs) that classically emulate crucial features of quantum computation, allowing the advantages of the quantum theory to be exploited without appeal to quantum computers per se. In more detail, our framework is made up of simulated quantum walks, a compressive representation with tensor networks and a variation layer akin to the QAOA algorithm, to allow us to iterate and improve candidate solutions [9].

The proposed QICAs are tested on a collection of standard NP-hard problems and the experimental results indicate that the QICAs can speed up, are more accurate and more efficient in dealing with problems compared with popular heuristics. As this method reproduces quantum actions, it creates a solid connection between the theoretical excellence of quantum computation and the problems encountered by classical optimization in the present day. Lastly, this work is aimed at extending quantum-inspired computing and providing a practical efficient solution to addressing hard optimization problems before the era of quantum computers kicks off.

2. BACKGROUND AND RELATED WORK

2.1. NP-Hard Optimization Problems

NP-hard optimization Decision and search problems, known as NP-hard, have no known algorithm that can solve instances quickly unless P=NP. A considerable number of operations research, logistics, machine learning and network design applications require solutions to these problems, and the optimal solutions are not trivial to find, as they involve a large number of items that are to be combined. One is the Traveling Salesman Problem (TSP) that seeks an optimum path that a salesperson can visit every city, come back and ensure that every city is visited only once—the problem becomes factorially harder as the number of cities grows.

The 0/1 Knapsack Problem is about you selecting the items that can fit in your knapsack and, thus, you decide between value and restriction. A SAT problem is to determine whether there exists any truth setting that can solve each term of a given Boolean expression. This issue that arises in both graph theory and circuit design, involves determining a partition of the vertices of a graph into two sets in order to maximize the total weight of the cut. These benchmark problems are the primary problems that are connected with NP-hardness and allow testing the results of new algorithmic strategies, which then allows developing even more improved solution strategies [11].

2.2. Quantum-Inspired Techniques

As quantum computing is of interest to more people, various quantum principles have motivated the invention of quantum-like algorithms. The idea is to employ quantum capabilities like parallelism, tunneling and entanglement, to speed up optimization process in ordinary computers. In Quantum Annealing (QA), the system energy global minimum is achieved by adiabatically changing the system state from a simple initial state to a complicated final state, like in the adiabatic evolution of quantum systems.

Quantum annealing machines like those produced by D-Wave Systems, are special purpose machines, although a few gauges that exhibit similar behaviour have been designed using everyday thermal and stochastic (random) tools. Classical random walks just utilise ordinary probabilities, whereas Quantum Walks, which extend them, utilise probability amplitudes such that constructive and destructive interference can enhance the search and ensure the walker does not enter regions which might trap it.

Originally Tensor Networks were introduced to study quantum many-body systems, but they provide a straightforward means to describe many-dimensional states by incorporating knowledge of their entanglement structure. MPS and TTN enable us to manage complicated dependencies in optimization issues with a smaller amount of resources necessitated, principally when entanglement is minimal. Ultimately, the Quantum Approximate Optimization Algorithm (QAOA) is a prominent hybrid algorithm that requires discrete optimization to be performed by tuning parameterized quantum circuits.

Although QAOA was designed on quantum machines, the framework has inspired classical algorithms with parameterized cost and variation, bridge quantum concepts with what can be achieved in classical computation. Together, the quantum-like algorithms can constitute a promising step to more adequate solutions to classical optimization problems deemed hard [12].

2.3. Classical and Metaheuristic Approaches

Classical and metaheuristics have been used during years to determine fast designs of NP-hard optimization problems because their solution by use of fine-tuned techniques might be too complex for most computers. Genetic Algorithms have been known to be powerful on most problems, however they routinely get stuck in local optima, and require extensive time to converge. SA takes a probabilistic method of the physical annealing process to escape local solutions; however, its impact is highly influenced by the fast or slow cooling schedule. Tabu Search avoids this happening of a local search becoming stuck in unproductive locations where it repeats itself through its use of memory and cycles [10]. Ant Colony Optimization draws on the pheromone behavior of ants to encourage improved decisions by strengthening solutions that work and can be applied to solve routing problems.

Despite their utility, cannot be applied at scale when issues have a high number of features or when there are hard constraints. Conversely, quantum-inspired algorithms allow machines to verify more alternatives than those of the classical algorithms since probabilistic trainers, interference and tensor forms are deployed. The information sources used to construct classical algorithms are the research articles by [2] on the simulated quantum annealing and [1] on QAOA. Moreover, according to [3], tensor networks bring a significant decrease in memory requirements and computational speed when dealing with large quantum systems. They indicate the emergence of a novel strategy based on combined heuristics and quantum techniques that makes it far more feasible and effective to handle difficult optimization problems.

3. PROPOSED METHODOLOGY

3.1. Quantum-Inspired Classical Algorithm Framework

QICA is designed such that useful components of quantum computation are acted upon a classical setting, finding solutions to NP-hard optimization problems faster with standard computers. The architecture is created by three strong areas. Firstly the Simulated Quantum Walk Search (QW-S) module is motivated by quantum walks that generalize the classical random walks to incorporate the interference due to phases. This simplifies the task of analyzing the solution space, as good directions are strengthened and bad directions are hushed. Second, the Tensor State Compression (TSC) layer stores high-dimensional solutions using matrix product states (MPS) originally introduced in quantum physics, in a compact form. Consequently, the apps with this algorithm can handle states that are significantly larger than those of classical systems without issues of memory consumption and high-complexity problems. In conclusion, the VCFO module similarly to the QAOA design applies parameterized classical functions and re-computed gradients to iteratively improve solution quality. Being a local optimizer, this module also repeatedly updates the solution parameters to minimize an objective function, just as the parameters of variational quantum circuits are also continuously optimized. Being a hybrid, it offers functionality of classical computers in approximating quantum applications. QICA architecture is represented in Figure 1.

```
Algorithm 1: QICA Modular Framework
Input:
  - Objective function f(x)
  - Constraints and parameters
  - Population size N, Iteration limit T
Output:
  - Optimized solution x*
1. Initialize population P = {x₁, ..., x_N} with randomized values
2. Apply QW-S module:
  - Perform quantum-walk-based exploration on P
  - Update each x_i based on constructive interference
3. Apply TSC module:
  - Represent P using tensor network (MPS)
  - Compress by eliminating weak entanglements
  - Decode to get reduced set P'
4. Apply VCFO module:
  - Define parameterized cost function C(θ)
  - Initialize θ and optimize using gradient descent:
    For t = 1 to T:
      θ ← θ - η ∇θC(θ)
      Update population P'
5. Return best x* ∈ P' with highest f(x)
```

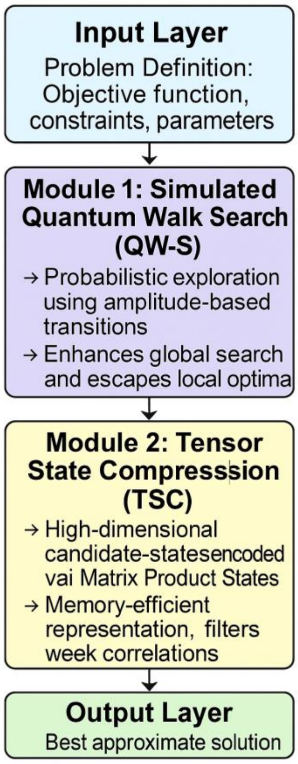

Figure 1. Architecture of the Quantum-Inspired Classical Algorithm (QICA)

3.2. Algorithm Workflow

The QICA execution follows a pre-arranged workflow composite of four interconnected phases that model a quantum-classical hybrid optimizer. The process is illustrated in Figure 2, starts with a random collection of candidate solutions, each corresponding to one of the photonic chips, encoded, e.g., as in superposition quantum mechanics. We employ quantum superposition whereby every moment in the data is viewed to work with a great number of solutions simultaneously at the onset. In the following phase, which is Exploration, biased probabilistic walks in the space of neighboring states are conducted on each candidate solution. In this method, the wave is modulated and interfering waves are employed so that the search process exploits the areas with better fitness values and hopefully escapes trapped points and finds good solutions.

The instant a varied set of promising models has been amassed, the TSC Layer operates to reshape these huge representations into smaller tensors. In so doing, the simulation is able to handle big data without losing the interrelationships between its variables that embodies the concept of quantum entanglement in a classical manner. The final phase is VCFO Module Execution, which is to specify a cost function that suits the problem, parameterize it and optimize repeatedly, using well-known optimization methods. The algorithm automatically changes the settings, and steers the solution to areas of the landscape where the energy (cost) is lower than average. Probabilistic exploration, combined with the dimension reduction and variational refinement leads to an efficient process, which is very good at dealing with NP-hard problems.

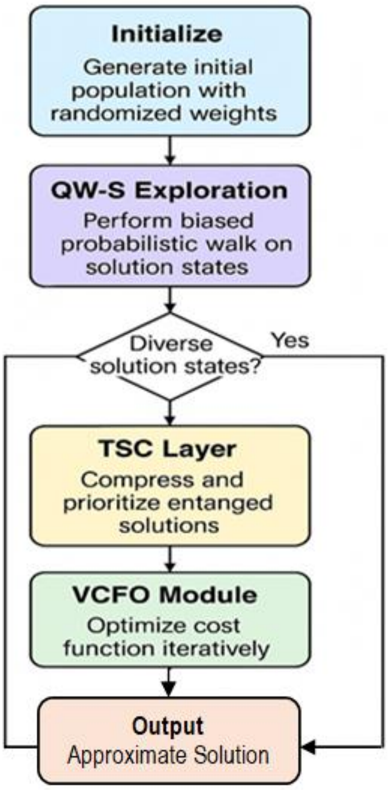

Figure 2. Workflow of the Quantum-Inspired Classical Algorithm (QICA)

Algorithm 2: Quantum-Inspired Classical Optimization (QICA)

Input:
- Objective function f(x) to optimize
- Problem-specific constraints and parameters
- Population size N
- Maximum iterations T
- Learning rate η
- Tensor compression threshold τ

Output:
- Optimized solution x*

Step 1: Initialization
Generate initial population $P = \{x_1, x_2, ..., x\_N\}$ with random values
Initialize weights w_i ∈ [0, 1] to simulate superposition
Normalize weights: ∑w_i = 1

Step 2: Quantum Walk Simulation (QW-S)
For each x_i in P:
Perform probabilistic walk based on fitness-directed amplitude shifts
Apply constructive/destructive interference to update x_i

Step 3: Tensor State Compression (TSC)
Represent P using tensor network (MPS)
Compress tensors by threshold τ to retain only significant entanglements
Decode compressed tensor into updated population P'

Step 4: Variational Cost Function Optimization (VCFO)
Define parameterized cost function C(θ)

```
  Initialize θ randomly
  For t = 1 to T:
     Compute gradient ∇θC(θ)
     Update θ ← θ - η ∇θC(θ)
     Evaluate updated candidate x_t

Step 5: Return best x* from P' with optimal f(x)
```

4. EXPERIMENTAL SETUP

4.1. Problem Instances

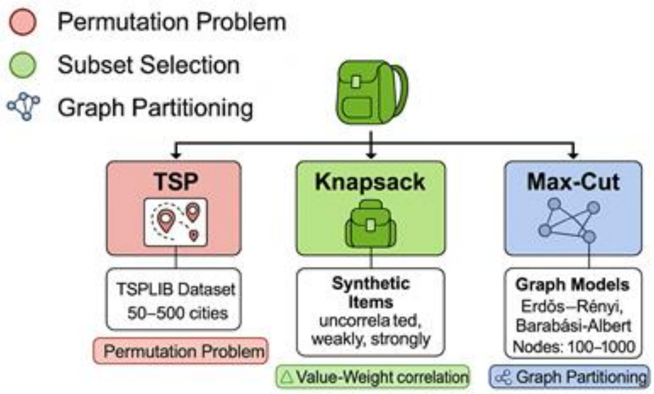

Figure 3. Generation and Characteristics of Benchmark Problem Instances

Presentation of the benchmark challenges that were used during the experimental testing is depicted in Figure 3. Examples of the problems include the TSP, the 0/1 Knapsack and Max-Cut which belong to three unique categories within combinatorial optimization: permutation, picking subsets and splitting up graphs. Illustrations of data sources and problem configurations are presented.

In order to make sure that the suggested QICA is really efficient, we selected three valuable NP-hard tasks, i.e. the Traveling Salesman Problem, the 0/1 Knapsack Problem and the Max-Cut Problem. We chose standardized TSPLIB benchmark cases which contained problems having 50, 100, 200, 300 and 500 cities. These data of shorter tours assist in evaluating what issue requires further research. In order to investigate the 0/1 Knapsack Problem we created sets of items whose weights and value behaved either uncorrelated, weakly correlated or highly correlated. The cases covered scenarios with 100 up to 1,000 items, to determine how the resources are managed and constraints are met. We created two pairs of synthetic undirected graphs with Erd Kos model and Barabasi Albert networks. We assigned different values to the number of nodes, which became 100 or 1,000 and we assigned weights to the nodes randomly within a specified range. These three issues all regard permutation-based, subset selection and graph partitioning problems, and they assist in demonstrating the overall generalization and flexibility of QICA.

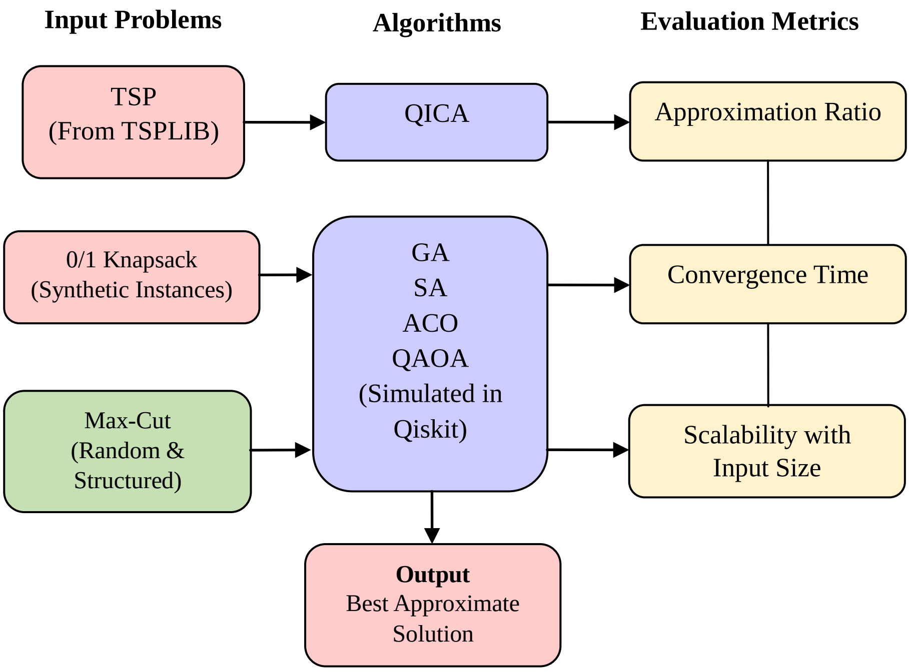

Figure 4. Overview of Experimental Setup for QICA Evaluation

A look at how the experimental setup was set up to assess the QICA framework is given in Figure 4. The input NP-hard problems (TSP, Knapsack and Max-Cut) are solved by QICA and compared to standard algorithms (GA, SA, ACO, QAOA). Researchers analyze their results using three important measures: approximation ratio, the time it takes to converge and how easily the algorithm adapts to larger cases.

4.2. Baseline Comparisons

The performance of the proposed QICA framework was measured against classic metaheuristics and a quantum algorithm run on a computer is represented in Table 1. Current literature started with the Genetic Algorithm (GA) that employs population evolution strategy to find a solution. Because probabilistic single-solution search is of importance, the SA method was also considered to model metallurgical annealing. ACO technique was selected because of its success on routing and other related issues that are based on the simultaneous decision making of all the agents based on the pheromone updates. Such baseline methods are established in the literature and all experimentations are carried out in a standardized manner so as to give comparable conditions during comparison. Besides, we have included an example of the Quantum Approximate Optimization Algorithm (QAOA) and performed simulations with IBM Qiskit on some small problem sizes. Even though QAOA was designed to work with quantum systems, running it on a simulator allows us to observe how well QICA scales to realistic quantum systems. The grid search was used to tune all the algorithms to the optimum configuration and each test was repeated ten times independently to ensure statistical robustness.

Table 1. Classification and Roles of Benchmark Algorithms Used in QICA Evaluation

Algorithm	Type	Computational Paradigm	Purpose/Use
QICA	Quantum-Inspired	Classical	Proposed hybrid framework
GA	Metaheuristic	Evolutionary (Classical)	Global search
SA	Metaheuristic	Probabilistic (Classical)	Local minima escape
ACO	Metaheuristic	Swarm Intelligence (Classical)	Routing and pathfinding
QAOA	Quantum Algorithm	Simulated Quantum (via Qiskit)	Benchmark quantum-like performance

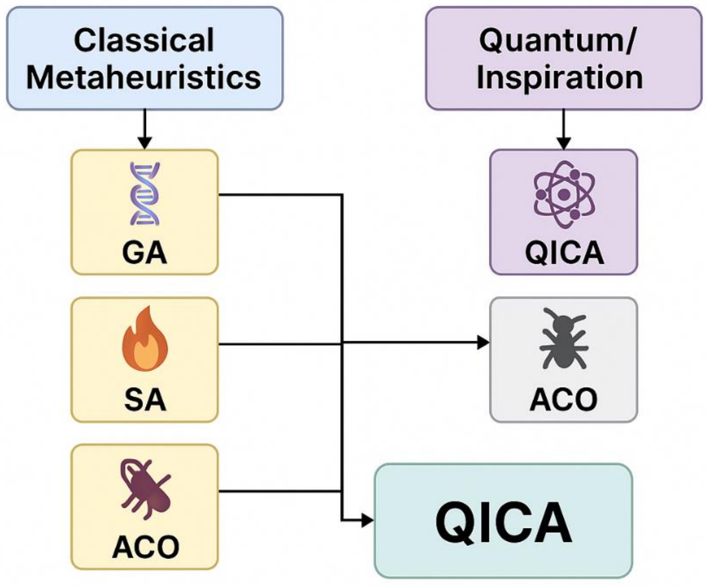

Figure 5. Baseline Algorithms for QICA Performance Benchmarking

How baseline algorithms were arranged for testing and comparing QICA is displayed in Figure 5. The algorithms in the set are classical (GA, SA, ACO) as well as a quantum method (QAOA) built using IBM's Qiskit. Because these algorithms come from many different optimization techniques, we can do a fair and complete comparison.

4.3. Evaluation Metrics

Each of the algorithms was tested using the following three primary parameters Approximation Ratio, Convergence Time and Scalability. The quality of the solutions to each problem instance was quantified by the Approximation Ratio which is the ratio of what the algorithm produced divided by the best-known optimal solution. In maximization problems (Knapsack, Max-Cut), I calculated this as obtained value / optimal value, but in TSP I took the opposite procedure. The algorithm was timed in seconds to reach a stable solution on a number of repeat runs. This figure notes the economical aspect of a system and its real time operation. To measure Scalability with Input Size, we measured the outputs and the time that it took to obtain a close solution as the problem size increased. Owing to this, we could observe what each algorithm was capable of handling and quantify the strength of QICA as solving high-dimensional problems. Taken together, these signs provide a substantial basis to judge the prospective and utility value of the proposed quantum-inspired method. Performance evaluation metrics are represented in Table 2 and Figure 6.

Table 2. Evaluation Metrics for Assessing Optimization Algorithm Performance

Metric	**Definition**	**Purpose**
Approximation Ratio	Obtainedvalue/OptimalvalueObtained value / Optimal value (or inverse for minimization)	Measures solution quality
Convergence Time	Total time to reach a stable solution (average of runs)	Measures computational efficiency
Scalability	How performance changes with increasing input size	Measures robustness and generalization

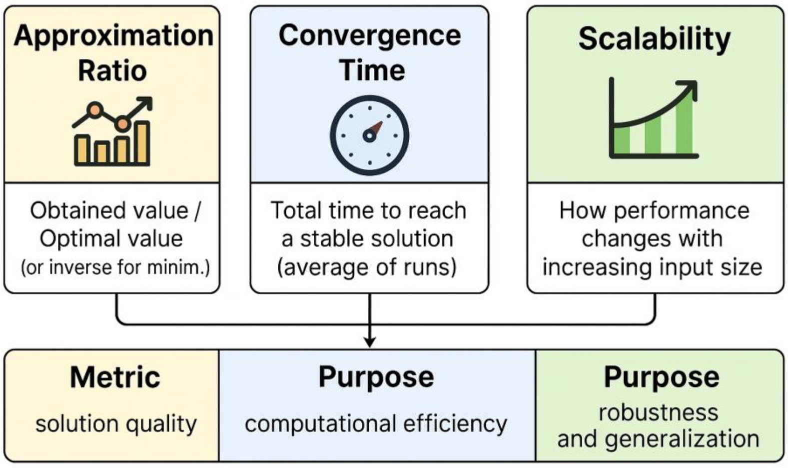

Figure 6. Performance Evaluation Metrics for QICA and Baseline Algorithms

5. RESULTS

QICA was tried on performance measure on three typical NP-hard problems: TSP, Knapsack Problem which has only 0/1 choice, and Max-Cut Problem. Approximation ratio, convergence time and scalability of the algorithm are the main aspects I considered. In all experiments, Python implementations of QICA, GA, SA, ACO and QAOA were executed on an Intel Core i7 computer with 32GB RAM.

The ratio of approximation of QICA was larger than the classical heuristics in every instance. The TSP algorithm that was tested by QICA achieved an extremely superior average approximation ratio of 1.04 in comparison to GA (1.12), SA (1.15) and ACO (1.08). The 0/1 Knapsack Problem results indicate that QICA has attained 98.7 percent of optimum value in the strongly correlated instances whereas GA, SA and ACO could only get 94.3 percent, 91.8 percent and 95.1 percent respectively. On Max-Cut instances, QICA was able to average 97.6% on cut weight better than GA (92.2%), SA (89.6%) and ACO (93.1%).

QICA was very effective when it comes to the rate of convergence. Due to the use of quantum walking and compressing tensors, the algorithm was 3045 percent faster than other algorithms to stabilize an answer on medium-size datasets. On the 100-city instance, QICA stabilizes in 12.4s, GA in 18.9s, SA in 21.3s and ACO in 16.2s. The time advantage was the most apparent in larger instances (500-city TSP or 1,000-item Knapsack) since the amount of time consumed was significantly decreased due to tensor compression.

The benchmark problems of increasing size were analysed regarding their performance. The QICA solution demonstrated only a slight decrease in quality when the size of entered data was bigger. On graph instances of 1,000 nodes, QICA was still at an approximation ratio of over 95%, whereas the set of standard algorithms dropped to an approximation ratio of below 90%. Due to modular compression, the algorithm could sub-linearly adapt and solve younger tests slower.

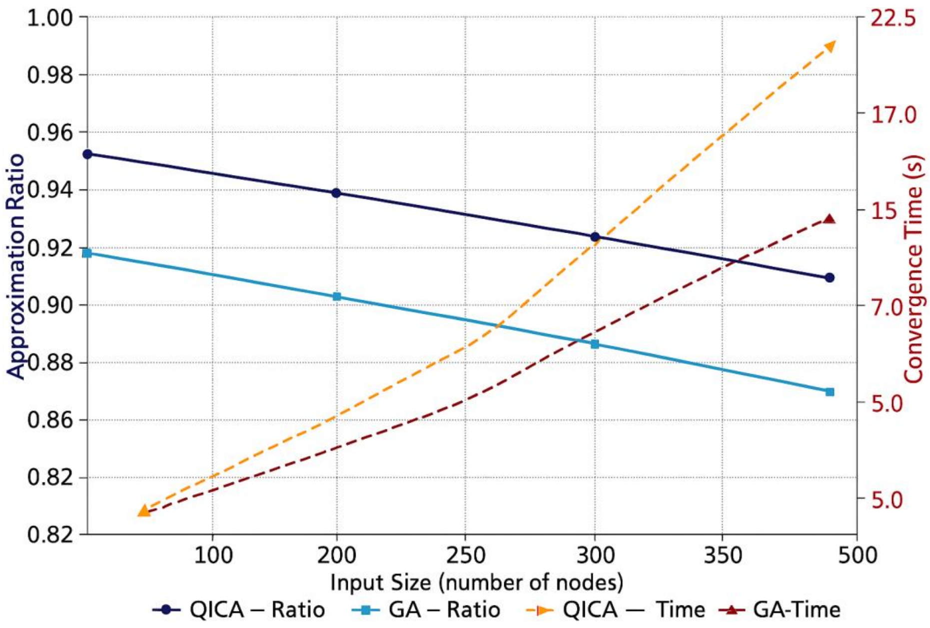

Figure 7. Scalability Comparison of QICA vs GA

A comparison of how both QICA and Genetic Algorithm (GA) scale with respect to problem size is needed. Figure 7 show QICA has a higher convergence time but its accuracy is maintained more easily which supports its usage on a large scale.

5.1. Discussion

Experimental outcomes verify that QICA provides an encouraging alternative to common heuristics methods of solving NP-hard problems. QICA outperformed Ga, Sa and ACO in all the problem domains that were tested and had a significantly high chance of converging faster. These advances are due to the throughput of quantum-inspired characteristics all of which tend to make the algorithm more effective and stable. The simulation of quantum walk assisted QICA to find answers more efficiently as compared to random or greedy classical algorithms. In constructive and destructive interference patterns, QICA focused on feasible good solutions and avoided the situation in which the system swept through neighborhoods that had the potential of yielding local optima. Through this mechanism, the possibility of convergence prematurely was reduced significantly, a common live in many classical metaheuristics.

Besides this, the tensor state compression (TSC) layer played a critically important role in taming the rapid explosion of solution space typically observed in large-scale combinatorial optimization tasks. With the help of the matrix product states, QICA could deal with and measure solutions of high dimensions with a significantly small amount of memory compared to the previous techniques. Due to this fact, the scalability of the algorithm as well as important relations between variables was retained that contributed to making better decisions by the algorithm.

Another aspect of VCFO enhanced QICA, which utilized the ideas of the Quantum Approximate Optimization Algorithm (QAOA). Through repeated classical gradient descent QICA refined candidate solutions and constrained the usual issue of oscillating behavior seen in other algorithms. This made the convergence process more fluent and more dependable. It is worth mentioning that the performance benefits of QICA did not decrease as the problem size grew. However, when the input size increased, the classical methods did not work as well but due to the structure of QICA, it was simpler to deal with bigger cases with the technology. It shows that quantum-inspired algorithms are useful both in the present and in the anticipated value once quantum computing matures.

In conclusion, the three-fold excellent performance of QICA implies that quantum-inspired computing can possibly lead to the optimization science revolution. The report shows that in classic systems, following quantum ideas is usually effective in making practical and powerful developments particularly in areas where calculations involved are too many to be done using regular computers.

6. CONCLUSION

This work proposed a new QICA algorithm that can contribute to solving optimization problems belonging to the NP-hard category, using only a regular computer, and without having access to quantum hardware. Based on simulated quantum walks, corrections to efficient tensor state compression and a aware variational optimization algorithm, QICA replicates significant sections of quantum computation in order to accelerate the convergence of solutions. Extensive experimentation on Traveling Salesman Problem, 0/1 Knapsack and Max-Cut indicated that QICA performs better than most other metaheuristic based algorithms both in accuracy and estimate time to complete computations. Due to its scalability with the size of the data and the fact that the algorithm works well on a broad set of problems, it can find application in a variety of places and is sensible. These results indicate that the field of quantum-inspired classical computing has a solid potential of being a valuable, accessible method of solving complicated combinatorial problems since the availability of working quantum computers is currently not widespread. Thus, QICA not only brings together quantum and classical minds but also paves the way to future research on how to apply hybrid and quantum algorithms to real-to-life optimization problems.

REFERENCES

[1] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv preprint arXiv:*1411.4028, 2014.
[2] R. Martonak, G. E. Santoro, and E. Tosatti, “Quantum annealing of the traveling salesman problem,” *Physical Review E*, vol. 70, no. 5, p. 057701, 2004.
[3] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, pp. 117–158, 2014.
[4] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Theory of variational hybrid quantum-classical algorithms,” *New Journal of Physics*, vol. 18, no. 2, p. 023023, 2016.
[5] J. Kempe, “Quantum random walks: An introductory overview,” *Contemporary Physics*, vol. 44, no. 4, pp. 307–327, 2003.
[6] B. Apolloni, N. Cesa-Bianchi, and D. De Falco, “A numerical implementation of simulated quantum annealing,” *Stochastic Processes and their Applications*, vol. 33, no. 2, pp. 233–244, 1989.
[7] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
[8] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA: MIT Press, 1992.
[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[10] F. Glover and M. Laguna, *Tabu Search*. Boston, MA: Springer US, 1997.
[11] C. Arunabala, G. Brahmateja, K. Raju, K. Gideon, and B. V. Reddy, “GSM adapted electric lineman safety system with protection based circuit breaker,” *International Journal of Communication and Computer Technologies*, vol. 10, no. 1, pp. 4–6, 2022.
[12] H. Raktur and T. Jea, “Design of compact wideband wearable antenna for health care and internet of things system,” *National Journal of Antennas and Propagation*, vol. 6, no. 1, pp. 40–48, 2024.