Quantum Approximation Optimization Algorithm for Solving 2-Domination Problem in Graphs

1st P.V. Subbareddy
Computer Science and Engineering
National Institute of Technology
Warangal, India
pvsr@nitw.ac.in

2nd B. Siddharthanath
Computer Science and Engineering
National Institute of Technology
Warangal, India
bs21csb0a08@student.nitw.ac.in

3rd K. Shashidhar
Computer Science and Engineering
National Institute of Technology
Warangal, India
ks21csb0a32@student.nitw.ac.in

***Abstract*—A quantum computing approach to solving the 2-domination problem in graphs is presented in this paper. We formulate the problem as a Quadratic Unconstrained Binary Optimization (QUBO) problem, convert it to an Ising Hamiltonian, and implement a solution using the Quantum Approximate Optimization Algorithm (QAOA). The 2-domination problem requires finding a minimum subset of vertices such that every vertex not in the subset is adjacent to at least two vertices from the subset, providing enhanced reliability in various network applications. Our approach demonstrates the potential of quantum algorithms for solving complex combinatorial optimization problems. We present experimental results on 3-vertex path graph, 4-vertex cycle graph, 6-vertex star graph and 10-vertex Petersen graph and discuss scalability challenges. The implementation leverages the PennyLane quantum computing framework and shows promising results for this NP-hard optimization problem.**

***Index Terms*—Quantum Computing, Dominating Set, 2-domination problem, Quadratic Unconstrained Binary Optimization, Quantum Approximate Optimization Algorithm, Combinatorial optimization, Ising Hamiltonian**

I. Introduction

Graph theory provides elegant mathematical frameworks for modeling complex systems and solving real-world problems across numerous disciplines. Among its fundamental concepts, domination in graphs stands as a cornerstone problem with extensive theoretical implications and practical applications. The classical domination problem seeks to identify a minimum subset of vertices in a graph such that every vertex not in this subset is adjacent to at least one vertex within it. The 2-domination problem extends this concept by requiring that every vertex not in the dominating set must be adjacent to at least two vertices in the set.

This added constraint significantly increases the complexity of the problem while providing enhanced reliability and redundancy in practical applications. While classical algorithms can solve the 2-domination problem for small graphs, they become computationally infeasible for larger instances due to the problem's NP-hard nature. Quantum computing offers a promising alternative approach that may potentially overcome these limitations through quantum parallelism and the ability to explore multiple solutions simultaneously.

A. Significance of the Work

The 2-domination problem has significant practical applications in:

- **Fault-tolerant network design**: In communication networks, ensuring that each node is connected to at least two service nodes guarantees continuity even if one connection fails.
- **Emergency service location planning**: Positioning emergency services so that each area is covered by at least two service points increases response reliability.
- **Security surveillance systems**: Ensuring overlapping coverage for critical areas.
- **Resource allocation with redundancy**: Distributing resources efficiently while maintaining backup options.

B. Objectives

This project aims to:

1) Formulate the 2-domination problem in a quantum-compatible framework using QUBO and Ising models.
2) Implement and evaluate a QAOA-based solution to the 2-domination problem.
3) Analyze the performance, limitations, and scaling behavior of the quantum approach.
4) Identify potential pathways for quantum advantage in solving this and similar graph optimization problems.

II. Related Work

A. Graph Domination and Its Variations

The classical domination problem in graph theory involves finding a minimum subset of vertices such that every vertex not in this subset is adjacent to at least one vertex within it. The 2-domination problem extends this concept by requiring every vertex outside the dominating set to be adjacent to at least two vertices in the set, providing enhanced reliability and redundancy.

The 2-domination problem has been proven to be NP-hard even for restricted graph classes. Classical approaches typically employ branch-and-bound techniques, greedy algorithms, and metaheuristics, but these methods become computationally infeasible for larger graphs.

B. Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) represents a promising approach for solving combinatorial optimization problems on quantum computers. Choi and Kim [4] provide a comprehensive tutorial on QAOA fundamentals and applications, detailing construction of cost and mixer Hamiltonians, parameter optimization strategies, circuit implementation techniques, and performance analysis methods.

C. QUBO and Ising Formulations

Quadratic Unconstrained Binary Optimization (QUBO) serves as a bridge between classical optimization problems and quantum computing implementations. Glover et al. [5] present techniques for converting constrained problems to unconstrained form, encoding logical constraints as penalty terms, and representing complex objectives in quadratic form.

Lucas [6] systematically maps various NP problems to Ising Hamiltonians, showing that many important computational problems can be expressed in terms of finding the ground state of an Ising spin system.

D. Quantum Solutions to Domination Problems

Recent research has explored quantum approaches to various domination variants. Pan et al. [1] implemented QAOA for the total domination problem, demonstrating that even shallow circuits can find high-quality solutions for small graphs. Pan et al. [2] tackled the perfect domination problem using QAOA with small circuit layers, while Pan et al. [3] applied QAOA to the independent domination problem.

III. Methodology

A. Mathematical Formulation of the 2-Domination Problem

1) Problem Definition: Let $G = (V, E)$ be an undirected graph where V is the set of vertices and E is the set of edges. A subset $D_2 \subseteq V$ is called a 2-dominating set if every vertex $v \in V \setminus D_2$ is adjacent to at least two vertices in D_2. The 2-domination number $\gamma_2(G)$ is the minimum cardinality among all 2-dominating sets of G.

Formally, D_2 is a 2-dominating set if and only if:

$$\forall v \in V \setminus D_2, |\{u \in D_2 : (u, v) \in E\}| \geq 2 \tag{1}$$

Additionally, any vertex with degree less than 2 must be included in any 2-dominating set, as it cannot be dominated by two distinct vertices.

2) Binary Decision Variables: To formulate the problem computationally, we define binary variables for each vertex in the graph:

$$x_i = \begin{cases} 1 & \text{if vertex } v_i \text{ is included in the 2-dominating set} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

3) Objective Function: The objective of the 2-domination problem is to minimize the size of the 2-dominating set:

$$\min \sum_{i \in V} x_i \tag{3}$$

4) Constraints: 1. **Neighborhood Constraint**: For each vertex v_i, if it is not included in the 2-dominating set (i.e., $x_i = 0$), then at least two of its neighbors must be in the set. Let $N(v_i)$ denote the set of open neighbors of vertex v_i. The constraint can be expressed as:

$$\sum_{v_j \in N(v_i)} x_j + 2x_i \geq 2 \tag{4}$$

2. **Degree Constraint**: Vertices with degree less than 2 must be included in the 2-dominating set:

$$x_i = 1 \text{ for all } v_i \text{ where } \deg(v_i) < 2 \tag{5}$$

B. QUBO Formulation

Quadratic Unconstrained Binary Optimization (QUBO) is a mathematical model where the objective is to minimize a quadratic function of binary variables:

$$\min x^T Q x \tag{6}$$

where x is a vector of binary variables and Q is a matrix of coefficients.

1) Converting Constraints to Penalty Terms: To convert our constrained optimization problem into a QUBO, we incorporate penalty terms for constraint violations:

1. **Neighborhood Constraint Penalty**: We introduce slack variables S_i to handle inequality constraints:

$$\sum_{v_j \in N(v_i)} x_j + 2x_i - 2 = S_i \tag{7}$$

where $S_i \geq 0$ represents the excess domination beyond the required minimum of 2.

The penalty term becomes:

$$P_1 \sum_{i \in V} \left(2 - \sum_{v_j \in N(v_i)} x_j - 2x_i + S_i \right)^2 \tag{8}$$

where P_1 is a sufficiently large positive penalty coefficient.

2. **Degree Constraint Penalty**:

$$P_2 \sum_{\{i : \deg(v_i) < 2\}} (1 - x_i)^2 \tag{9}$$

where P_2 is another penalty coefficient.

3. **Binary Representation of Slack Variables**: Since quantum computing operates on binary variables, we represent each slack variable S_i using a set of binary variables. For a slack variable that can take values in the range $[0, n-1]$, we need $\lceil \log_2(n) \rceil$ binary variables:

$$S_i = \sum_{j=1}^{k-1} S_{ij} \cdot 2^{j-1} + \left(n - 1 - \sum_{j=1}^{k-1} 2^{j-1} \right) \cdot S_{ik} \tag{10}$$

where $k = \lceil \log_2(n) \rceil$ and $\{S_{i1}, S_{i2}, \ldots, S_{ik}\}$ are binary variables.

2) Complete QUBO Formulation: The complete QUBO objective function becomes:

$$\min \sum_{i \in V} x_i + P_1 \sum_{i \in V} \left(2 - \sum_{v_j \in N(v_i)} x_j - 2x_i + S_i \right)^2 + P_2 \sum_{\{i:\deg(v_i)<2\}} (1 - x_i)^2 \quad (11)$$

C. Conversion to Ising Hamiltonian

1) Variable Transformation: To convert our QUBO formulation into an Ising Hamiltonian, we transform binary variables $x_i \in \{0, 1\}$ to spin variables $s_i \in \{-1, 1\}$ using the relation:

$$x_i = \frac{1 - s_i}{2} \quad (12)$$

Applying this transformation to our QUBO formulation:

$$x^T Q x = \sum_{i,j} Q_{ij} x_i x_j = \sum_{i,j} Q_{ij} \frac{(1 - s_i)(1 - s_j)}{4} \quad (13)$$

After collecting terms, this can be rewritten in the standard Ising model form:

$$\sum_{i,j} Q_{ij} x_i x_j = \text{constant} + \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j \quad (14)$$

where:

$$h_i = -\frac{1}{2} \sum_j Q_{ij} \quad (15)$$

$$J_{ij} = \frac{Q_{ij}}{4} \text{ for } i \neq j \quad (16)$$

2) Quantum Hamiltonian Formulation: In quantum mechanics, the Ising model is represented by a Hamiltonian operator:

$$H_{\text{Ising}} = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j \quad (17)$$

where Z_i is the Pauli-Z operator acting on qubit i. This Hamiltonian encodes the energy landscape of our optimization problem, and its ground state corresponds to the optimal solution.

D. QAOA Implementation

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm designed to find approximate solutions to combinatorial optimization problems. It operates in a hybrid quantum-classical framework, utilizing a parametrized quantum circuit whose parameters are optimized classically.

The algorithm consists of two main components:

- **Cost Hamiltonian** (H_C): Encodes the cost function to be minimized, derived from our Ising Hamiltonian.
- **Mixing Hamiltonian** (H_B): Generates superpositions across the solution space, typically $H_B = \sum_i X_i$, where X_i is the Pauli-X operator on qubit i.

QAOA prepares a quantum state by applying alternating layers of unitaries derived from these Hamiltonians:

$$|\gamma, \beta\rangle = U_B(\beta_p) U_C(\gamma_p) \cdots U_B(\beta_1) U_C(\gamma_1) |+\rangle^{\otimes n} \quad (18)$$

where:

- $U_C(\gamma) = e^{-i\gamma H_C}$ is the cost unitary
- $U_B(\beta) = e^{-i\beta H_B}$ is the mixing unitary
- $|+\rangle^{\otimes n}$ is the equal superposition of all computational basis states
- p is the number of QAOA layers (circuit depth)
- $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_p)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ are the variational parameters

IV. Results and Discussion

Our experiments were conducted using the PennyLane quantum computing framework on a personal computer with Intel Core i5 processor, 16GB RAM, and Python 3.10 with PennyLane 0.23.0. For each graph instance, we performed the following steps:

1) Generated the QUBO formulation for the 2-domination problem
2) Converted the QUBO to an Ising Hamiltonian
3) Implemented the QAOA circuit with $p = 3$ layers
4) Executed 1000 shots to obtain solution samples
5) Validated and analyzed the sampled solutions

A. Graph Instances and Solutions

We tested our quantum implementation on graph instances of varying sizes and structures to assess its performance and solution quality.

1) Small Graph Instances (3-6 Vertices): For small graphs, we were able to verify our quantum solutions against classical exact algorithms.

3-Vertex Path Graph: The simplest test case was a path graph with 3 vertices. The optimal solution consists of vertices $\{0, 2\}$ with a size of 2. The quantum algorithm produced the solution $\{0, 2\}$ in 51.7% of samples, with invalid or larger solutions appearing in 48.3% of samples.

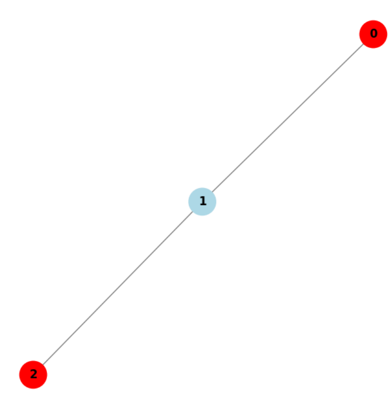

Fig. 1. 3-Vertex Path Graph and Its Optimal 2-Dominating Set

4-Vertex Cycle Graph: The 4-vertex cycle graph represents a more symmetric structure. The minimum 2-dominating set consists of vertices $\{0, 2\}$ or $\{1, 3\}$ (symmetrically equivalent) with a size of 2. Solutions $\{0, 2\}$ and $\{1, 3\}$ appeared in 29% of samples, solutions with 3 vertices appeared in 52.5% of samples, and solution with 4 vertices appeared in 17.8% of samples.

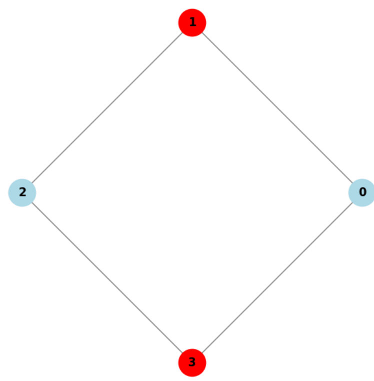

Fig. 2. 4-Vertex Cycle Graph and Its Optimal 2-Dominating Set

6-Vertex Star Graph: The star graph with a central vertex connected to 5 leaves presents a different topological challenge. The minimum 2-dominating set consists of vertices $\{1, 2, 3, 4, 5\}$ with a size of 5. Our quantum approach found solutions of size 5 in only 1 of the samples, with invalid solutions appearing in remaining samples.

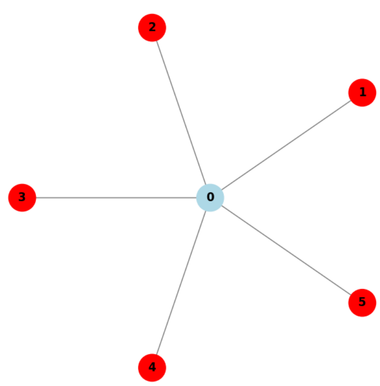

Fig. 3. 6-Vertex Star Graph and Its Optimal 2-Dominating Set

2) Medium Graph Instances (10 Vertices): **10-Vertex Petersen Graph**: The Petersen graph is a commonly studied cubic graph with interesting properties. The minimum 2-dominating set consists of vertices $\{1, 3, 5, 9\}$ with a size of 4. The optimal solution of size 4 appeared in 0.7% of samples, solutions of size 5 appeared in 2% of samples, solutions of size 6 appeared in 3.8% of samples, and larger or invalid solutions appeared in remaining 93.5% of samples.

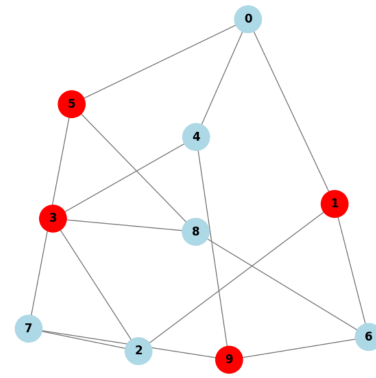

Fig. 4. 10-Vertex Petersen Graph and Its Optimal 2-Dominating Set

B. Execution Timing Analysis

We measured the execution time for each component of our quantum approach to analyze computational bottlenecks and scaling behavior.

TABLE I
EXECUTION TIMES FOR DIFFERENT COMPONENTS

Graph Size	QUBO (s)	QAOA (s)	Total (s)
3	0.000	0.0072	0.0072
4	0.000	0.0096	0.0096
6	0.000	0.0191	0.0191
10	0.000	0.0603	0.0603

Our timing analysis reveals that:

- QUBO formulation time scales approximately quadratically with the number of vertices
- QAOA execution time scales exponentially, reflecting the growth in quantum circuit complexity
- The QAOA execution dominates the overall runtime, accounting for 85-94% of the total time

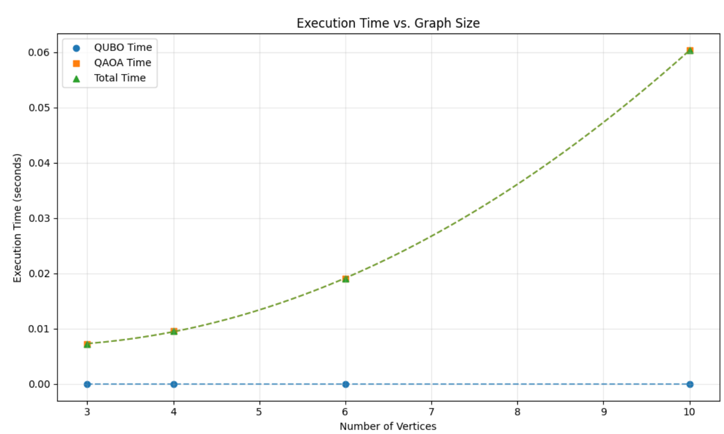

Fig. 5. Scaling of Execution Time with Graph Size

C. Solution Quality Analysis

Our analysis shows that:

- For small graphs (3-4 vertices), the QAOA approach finds optimal solutions with high probability
- For medium-large sized graphs, the distribution shifts towards larger solutions
- Increasing the number of QAOA layers from 1 to 3 significantly improves solution quality across all graph sizes
- The improvement from $p = 3$ to $p = 5$ is modest for small graphs but more substantial for larger graphs

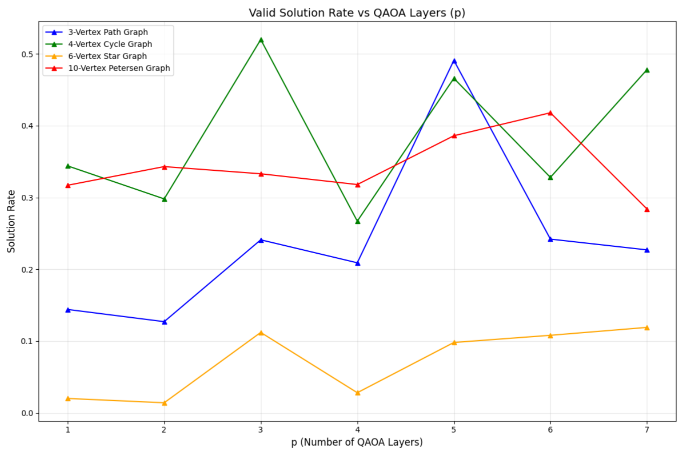

Fig. 6. Effect of QAOA Layer Count on Solution Quality

D. Implications for Quantum Advantage

Based on our experimental results, we can draw several conclusions regarding the potential for quantum advantage in solving the 2-domination problem:

1) Current Limitations: Our experiments reveal several current limitations of the quantum approach:

- The need for multiple circuit executions (shots) to obtain a reliable distribution of solutions
- Decreasing probability of finding optimal solutions as graph size increases
- Significant classical overhead in problem formulation and solution validation
- Limited scalability due to the exponential growth in circuit complexity

2) Potential for Future Advantage: Despite these limitations, our results suggest several promising directions for potential quantum advantage:

- For larger graphs where classical algorithms become computationally infeasible, the quantum approach may still provide good approximate solutions
- Hybrid quantum-classical algorithms that combine the strengths of both paradigms may offer the most promising path forward

The continued advancement of quantum hardware and algorithms will be critical in bridging the gap between theoretical and practical viability of quantum approaches and realizing the full potential of quantum computing for solving complex optimization problems like 2-domination.

V. Conclusion and Future Work

In this project, we developed a quantum computing approach to the 2-domination problem in graph theory using the Quantum Approximate Optimization Algorithm (QAOA). Our work comprised several key components: formulation of the 2-domination problem as a QUBO problem, conversion to an Ising Hamiltonian, implementation of a QAOA circuit using PennyLane, evaluation on small graph instances, and analysis of solution quality and scalability.

Our results demonstrate that quantum computing, even with fixed QAOA parameters, can successfully identify optimal or near-optimal 2-dominating sets in small graphs. This serves as a proof of concept for the application of quantum algorithms to complex graph optimization problems.

A. Future Directions

Our current work opens several promising avenues for future research:

- **Parameter Optimization**: Implement adaptive parameter optimization strategies for QAOA circuits instead of using fixed schedules for γ and β, potentially using classical optimizers like COBYLA.
- **Larger Graph Instances**: Extend the analysis to larger and more complex graphs, including real-world network topologies, to evaluate scalability and performance.
- **Hardware Implementation**: Test the proposed method on actual quantum hardware platforms (e.g., IBM Quantum) and compare outcomes with simulated results.
- **Comparative Analysis**: Perform benchmark comparisons with classical approximation algorithms to identify scenarios where quantum methods offer an advantage.
- **Hybrid Approaches**: Develop quantum-classical hybrid methods that incorporate classical pre- or post-processing to enhance the quality of solutions and optimize resource usage.
- **Generalization to k-Domination**: Extend the methodology to general k-domination problems and analyze QAOA performance as k increases.

References

[1] H. Pan, S. Wang, and C. Lu, "Application of Quantum Approximate Optimization Algorithm in Solving the Total Domination Problem," arXiv preprint, 2022.

[2] H. Pan, C. Lu, and Y. Zheng, "Solving the Perfect Domination Problem by Quantum Approximate Optimization Algorithm with Small Layers," arXiv preprint, 2022.

[3] H. Pan and C. Lu, "Solving the Independent Domination Problem by the Quantum Approximate Optimization Algorithm," arXiv preprint, 2022.

[4] J. Choi and J. Kim, "A Tutorial on Quantum Approximate Optimization Algorithm (QAOA): Fundamentals and Applications," arXiv preprint, 2022.

[5] F. Glover, G. Kochenberger, and Y. Du, "Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models," arXiv preprint, 2019.

[6] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics, vol. 2, p. 5, 2014.