Quantum Optimization for the Vehicle Routing Problem (VRP)

Drestanto Muhammad Dyasputro (1450107)
ddyasputro@student.unimelb.edu.au

Research Proposal / Plan

1 Introduction

The Vehicle Routing Problem (VRP) is a fundamental combinatorial optimization challenge in logistics, aiming to determine the most efficient routes for a fleet of vehicles to service a set of customers. Over the years, VRP and its different versions have become a popular topic in research. However, since the problem can be defined in many versions (many ways with many different assumptions), researchers need to be aware that VRP itself has a broad range of variants.

Given its NP-hard[1] nature, classical algorithms often struggle with large-scale instances due to computational limitations. The City of Casey[2] provides detailed datasets on waste facilities, public litter bins, and collection areas, making it an ideal case study for applying Vehicle Routing Problem (VRP) optimization techniques. By integrating real-world constraints such as traffic conditions, weather, and road incidents, this study models a practical VRP instance. With the problem's NP-hard nature, classical approaches face computational challenges in large-scale optimization.

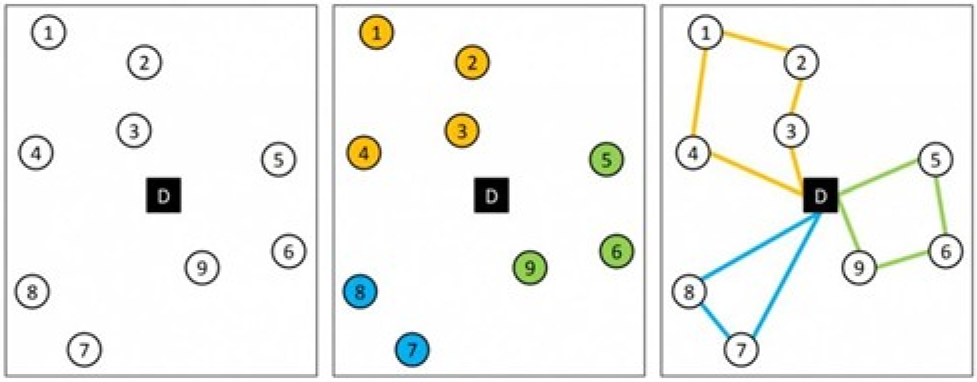
Quantum computing offers a paradigm for approaching and solving such problems. Algorithms like the Quantum Approximate Optimization Algorithm[3] (QAOA) and Quantum Walk[4]-based Optimization have shown promise in tackling NP-hard tasks. This research proposes to explore these quantum techniques to develop efficient approximation methods for the VRP.

2 Scope and Boundaries

2.1 Vehicle Routing Problem

VRP has numerous real-world applications, including logistics, supply chain management, and last-mile delivery services. Various VRP variants exist, such as the Capacitated VRP (CVRP), Time-Window VRP (VRPTW), and Stochastic VRP (SVRP), each introducing unique constraints that impact solution complexity. Classical methods struggle to find optimal solutions as problem size increases due to exponential computational growth.

This diagram is one method of finding solution to VRP[5]:



Unassigned Customers (Left Panel)

- Customers (represented by numbered circles) are scattered around a central depot (black square labeled "D").
- At this stage, no routes have been assigned.

Customer Clustering (Middle Panel)

- Customers are grouped into different sets based on their locations and other factors such as demand or distance (this is where heuristic comes into play).
- Each color represents a different vehicle that will serve a specific group (with certain heuristics) of customers.

Optimized Routes (Right Panel)

- Vehicles are assigned optimized routes that start and end at the depot ("D").
- Each route ensures that all customers are served efficiently while minimizing travel distance and cost.

2.2 Case Study: Waste Collection in the City of Casey

To analyze the Vehicle Routing Problem (VRP) in a practical setting, we consider a case study based on real-world waste collection services. The waste collection scenario is chosen due to its public, detailed, and structured dataset. The study focuses on the City of Casey, as this place offers granular data, an ideal dataset for modeling and validating VRP solutions.

The VRP model is constructed based on three key components: depots, customers, and routes:

- **Depots:** Waste collection facilities serve as depots, where vehicles begin and end their routes.

- **Customers:** Public litter bin locations represent customer nodes, where waste collection must be performed and combined with waste collection area data for comprehensive coverage, ensuring a comprehensive mapping of all waste pickup points across the city.
- **Routes:** Based on Victoria's road network, paths for vehicle travel between depots and collection points.

To enhance the model's accuracy and reflect real-world complexities, additional constraints can be introduced:

- Traffic: Higher traffic increases travel time.
- Weather (Rainfall): More rainfall slows travel.
- Road Crashes: Accidents cause delays.

The data of depots[2], customers[6, 7], routes[8], and constraints (traffic[9], weather[10], crashes[11]) can be imported from victoria open data.

2.3 Quantum Optimization

Quantum computing offers a novel approach to solving VRP by leveraging quantum superposition and entanglement to explore multiple solutions simultaneously. Quantum algorithms like QAOA and Quantum Walk-based methods have shown potential in improving combinatorial optimization efficiency. However, implementing these algorithms for practical VRP scenarios requires addressing efficiency, scalability, and quantum hardware limitations.

3 Background and Challenges

3.1 Classical Approaches to VRP

Traditional methods for solving the VRP include:

- Exact Algorithms: Techniques such as Branch-and-Bound for CVRP[12].
- Metaheuristic Approaches[13]: Algorithms like Genetic Algorithms[13] (GA) and Ant Colony Optimization[13] (ACO) offer approximate solutions more feasibly but may not guarantee optimality.

3.2 Quantum Programming Approaches to VRP

Quantum Optimization Techniques can offer a better solution than a classical one. Several approaches have been proposed to explore the advantages of quantum algorithms over classical optimization methods, particularly for NP-hard problems like the Vehicle Routing Problem (VRP):

- Quantum Approximate Optimization Algorithm (QAOA): Designed to find approximate solutions to combinatorial problems[3] by leveraging quantum superposition and entanglement.
- Quantum Walk-based Optimization: Quantum walks, the quantum analog of classical random walks, have been applied to search and optimization problems[14], offering potential speedups over classical methods.
- Variational Quantum Eigensolver[15] (VQE): Studied for solving VRP, though currently limited to small instances.
- Quantum Annealing[16]: A novel QUBO formulation[16] has been proposed for CVRP, incorporating a time-table representation to model the time evolution of each vehicle.

4 Research Objectives

1. Develop quantum optimization approaches for VRP (in this case, waste collection) using Quantum Algorithms.
2. Compare quantum and classical algorithms by analyzing solution quality, computational efficiency, and scalability.
3. Investigate the effectiveness of quantum approximations in solving VRP and their ability to find near-optimal solutions.
4. Implement a quantum-based VRP solution with the potential to outperform classical methods in specific scenarios.

5 Research Questions

- RQ1: How can spatial datasets be transformed into a VRP formulation suitable for quantum optimization?
- RQ2: How can quantum optimization techniques such as QAOA, Quantum Walk, VQE, and/or Quantum Annealing provide a more effective solution for VRP compared to classical methods?
- RQ3: In terms of efficiency and scalability, how do quantum algorithms compare to state-of-the-art classical VRP solvers?
- RQ4: What are the key limitations of using quantum algorithms for VRP, and how can these challenges be addressed?

6 Literature Review

6.1 Quantum Approximate Optimization for VRP

QAOA is useful for VRP because it can efficiently find approximate solutions to complex optimization problems, including route planning. VRP involves selecting the best paths for multiple vehicles while considering constraints like distance, capacity, and time windows. Since VRP is an NP-hard problem, classical algorithms struggle to solve large instances efficiently. QAOA leverages quantum superposition and interference to explore multiple possible solutions simultaneously, potentially finding good routes faster than classical heuristics.

Mathematically, QAOA encodes the VRP as a cost function H_C, which represents the total travel distance or other constraints, and a mixing function H_M, which helps the algorithm explore different routes. The quantum circuit is defined as:

$$|\psi(\gamma, \beta)\rangle = U_M(\beta_p)U_C(\gamma_p) \cdots U_M(\beta_1)U_C(\gamma_1)|s\rangle$$

where $U_C(\gamma) = e^{-i\gamma H_C}$ encodes the VRP constraints, and $U_M(\beta) = e^{-i\beta H_M}$ allows transitions between different solutions. Fitzek et al.[17] applied this method to handle vehicle-specific constraints, improving efficiency. Leonidas et al.[18] optimized QAOA for small quantum devices by reducing qubit requirements. Farhi et al.[3] originally introduced QAOA, showing that deeper circuits can improve results, though real quantum hardware still has limitations.

6.2 Quantum Walk-based Approaches

Quantum walks, which serve as a quantum analog to classical random walks, have been explored for VRP optimization. Quantum walks are useful for VRP because they can explore large search spaces efficiently by leveraging quantum interference and superposition. Unlike classical random walks, where the probability of moving to a new state depends only on the previous step, quantum walks maintain phase coherence, allowing for faster propagation through the solution space. The evolution of a discrete-time quantum walk is governed by the unitary operator:

$$U = S \cdot (I \otimes C)$$

where S is the shift operator that moves the walker between states, and C is the coin operator that controls the direction of movement. Bennett et al.[14] applied this framework to vehicle routing, demonstrating that quantum walks can help find optimal paths by efficiently traversing possible routes. Marsh and Wang[4] introduced an approximation method based on quantum walks, showing its applicability to bounded NP-hard problems. In a later study, they improved the efficiency of combinatorial optimization by refining the quantum walk formulation to enhance solution quality[19]. These works suggest that quantum walks could provide speedups for VRP compared to classical heuristics, though practical implementation remains limited by current quantum hardware constraints.

6.3 Quantum Annealing for VRP

Quantum annealing has been another area of research in VRP optimization. Tambunan et al. (2023) investigated quantum annealing for VRP with weighted segments, focusing on integrating dynamic constraints within the optimization process. Similarly, Irie et al. (2019) formulated a Quadratic Unconstrained Binary Optimization (QUBO) model for VRP with time, state, and capacity constraints, showcasing quantum annealing's applicability in complex routing scenarios.

Quantum annealing (QA) is a heuristic algorithm used for solving optimization problems. It relies on the principle of quantum mechanics, specifically the quantum superposition and tunneling phenomena, to find the global minimum of a cost function. The general form of a Hamiltonian used in quantum annealing for VRP can be expressed as:

$$H(t) = A(t)H_P + B(t)H_D$$

where H_P is the problem Hamiltonian that encodes the VRP objective and constraints, and HD is the driver Hamiltonian that governs the evolution of the quantum state. The functions $A(t)$ and $B(t)$ control the mixing between these Hamiltonians during the annealing process. As the annealing progresses, the system ideally transitions into the lowest energy state, providing a solution to the VRP. Recent studies, such as those by Irie et al.[16] and Tambunan et al.[20], demonstrate the viability of quantum annealing in addressing complex VRP instances with dynamic and weighted constraints, highlighting its potential as a tool for optimization in real-world logistics applications.

6.4 Hybrid and Machine Learning Approaches

Beyond traditional quantum techniques, researchers have explored hybrid quantum-classical methods and machine learning integration for VRP. Hybrid methods, such as QSVMs and VQAs, offer promising approaches. For example, the decision function for QSVM is given by:

$$f(x) = \operatorname{sgn}\left(\sum \alpha_i y_i \langle \phi(x_i) | \phi(x) \rangle + b\right)$$

where $\langle \phi(x_i) | \phi(x) \rangle$ represents the inner product between quantum states, and α_i and b are parameters determined through training. This approach enhances the accuracy of route classification, improving overall VRP optimization. On the other hand, VQAs utilize parameterized quantum circuits in combination with classical optimization algorithms to iteratively adjust quantum states and minimize the objective function. The quantum circuit's parameters are adjusted using a classical optimizer to solve for the optimal routing decisions under given constraints. These hybrid methods take advantage of quantum computing's potential for high-dimensional state space exploration, while maintaining practical feasibility in hardware-limited environments, as demonstrated by Alsaiyari and Felemban[15] and Mohanty et al.[21].

6.5 Conclusion

The exploration of quantum algorithms for VRP has revealed promising directions, particularly with QAOA and quantum annealing. While current research demonstrates theoretical advantages over classical methods, practical implementations remain limited by hardware constraints. Future work should focus on large scale VRP application which needs improvement in qubit efficiency, hybrid algorithm designs, and benchmarking against state-of-the-art classical solvers to fully leverage quantum computing.

7 Detailed Methodology: QAOA for VRP

7.1 Vehicle Routing Problem and Graph Representation

The Vehicle Routing Problem (VRP) is a classic optimization challenge that involves determining the most efficient routes for one or more vehicles to serve a set of customers. For simplicity, we've focused on a scenario with a single depot and a single vehicle. Our input consists of customer and depot coordinates, which define their spatial locations.

To translate this spatial problem into a format suitable for computation, we treat each customer and the depot as a node in a graph. The routes connecting these nodes are represented as edges, with weights assigned to each edge to reflect the distance or travel time between the nodes. Classical algorithms, such as Dijkstra's or A*, can be used to construct this graph and calculate the edge weights.

For n customers and 1 depot, the graph can be represented as an NxN (square) matrix where N equals n+1, where each element in the matrix, row i and column j corresponds to the weight of the edge between nodes i and j. The matrix will be symmetric with zeros in the diagonal, meaning that the transpose will equal to the matrix itself

7.2 Quantum Approximate Optimization Algorithm (QAOA) Overview

QAOA is a quantum algorithm (or hybrid quantum-classical algorithm to be exact) designed to find approximate solutions to combinatorial optimization problems. Unlike exact quantum algorithms, QAOA aims to find good (but not necessarily best/optimal) solutions within a reasonable timeframe.

The QAOA process involves:

1. **Cost Function**: Defining a cost function that represents the problem's objective. In our VRP, this function measures the total distance or travel time of a route.
2. **Hamiltonian**: Encoding the cost function into a quantum Hamiltonian.
3. **Quantum Circuit**: Constructing a parameterized quantum circuit that alternates between applying the cost Hamiltonian and a mixing Hamiltonian.
4. **Optimization**: Iteratively adjusting the parameters (gamma and beta) of the quantum circuit to minimize the cost function, using a classical optimizer.
5. **Approximate Solution**: Extracting an approximate solution from the final quantum state.

Key features of QAOA include:

- **Approximate Solutions**: QAOA provides approximate (not necessarily optimal) solutions.
- **Parameterized Circuits**: The quantum circuit's parameters (gamma and beta) are optimized classically.
- **Depth (p)**: The depth of the QAOA circuit determines the number of iterations and affects the solution's quality.

7.3 VRP Representation in Quantum Qubits

To apply QAOA to the VRP, we need to represent the problem using quantum qubits. A naive approach is to represent all possible edges between nodes. However, this is not optimal.

- **Graph Representation in Matrix**: As mentioned earlier, we use an (n+1) x (n+1) matrix to represent the graph.
- **Qubit Allocation**: The key challenge is to efficiently encode the order of customer visits. Using $n+1$ qubits is insufficient. VRP requires representing not just binary choices, but the order of visits. We found that $2^k > n!$, where k is the number of qubits, is sufficient. This is because there are $n!$ possible permutations of order in n customers. The easiest way to find k is to use the matrix number of element. If $k = (n+1)^2$, $2k$ will grow much faster than $n!$ (We can proof this using mathematical ratio with techniques such as stirling's approximation and logarithm)

7.4 Qubit Allocation Improvements and Discussion

One of the key challenges in implementing the Quantum Approximate Optimization Algorithm (QAOA) for the Vehicle Routing Problem (VRP) is determining an efficient and scalable strategy for qubit allocation. The choice of encoding directly affects both the computational feasibility and the expressiveness of the quantum circuit. In this study, we evaluate and compare three distinct approaches to qubit allocation:

1. naive adjacency matrix encoding,
2. logarithmic permutation encoding, and
3. node-visit-time encoding.

Each method presents trade-offs in terms of qubit efficiency, implementation complexity, and suitability for enforcing constraints.

Naive Matrix-Based Encoding
The most straightforward approach assigns one qubit per possible directed edge between nodes, resulting in an $N \times N$ grid of qubits for a problem with $N = n + d$ locations (customers and depots). This approach provides a natural representation for the cost function, as distances are encoded directly via edge weights. However, the qubit count grows quadratically with N, which leads to prohibitively large state vectors in simulation and quantum hardware contexts.

Example

Let's assume $N = 6$, which requires 36 qubits and a 2^{36}-dimensional state space. Each qubit q_{ij} represents whether the route includes a directed edge from node i to node j. For example, in a system with four nodes, qubit $q_{1,2}$ encodes whether the vehicle travels from location 1 to location 2. The final route is interpreted as a path composed of edges where the corresponding qubits are in the $|1\rangle$ state.

Logarithmic Permutation Encoding

This approach encodes the entire route as a permutation of customers using the smallest number of qubits possible, i.e., $k = \lceil \log_2(n!) \rceil$. While this method is optimal in terms of qubit usage, it introduces substantial difficulty in mapping quantum bitstrings to meaningful cost evaluations. Constructing a quantum Hamiltonian that accurately reflects VRP constraints under this encoding is highly non-trivial. Furthermore, validity checks and constraint penalties must be implemented as post-processing steps, which diminishes the effectiveness of in-circuit cost optimization.

Example

With $n = 3$ customers, there are $3! = 6$ possible routes, and $k = \lceil \log_2(6) \rceil = 3$ qubits are sufficient. Each measured bitstring from the quantum circuit (e.g., 010 or 101) is interpreted as an integer index into a precomputed list of valid permutations. A bitstring like 011 might correspond to the customer sequence [1, 3, 2], yielding the route $[0, 1, 3, 2, 0]$ with the depot appended at start and end.

Node-Visit-Time Encoding (Proposed Hybrid Method)

To balance qubit efficiency and route validity, we introduce a hybrid approach wherein each customer is represented by a fixed-size binary register that denotes their position in the route (i.e., visit time). This method requires $n \cdot \lceil \log_2(N) \rceil$ qubits and allows for structured constraint enforcement. Each route is reconstructed by sorting customers based on their encoded visit times. Invalid configurations, such as duplicate positions or unassigned slots, are penalized through additional constraint terms in the Hamiltonian.

Example

Using the same example, with $n = 3$ customers and $N = 4$ total nodes, each customer is assigned a 2-qubit register to represent their visit time. If customer 1 is encoded by qubits q_0, q_1, and their values are 01, this indicates that customer 1 is visited second in the route. A full bitstring such as 01101011 might be interpreted as: customer 1 at position 1, customer 2 at position 2, customer 3 at position 3, leading to a route of $[0, 1, 2, 3, 0]$. Duplicate visit times can be easily detected and penalized by counting repeated integers in the decoded visit-time list.

Comparison

The following table summarizes the characteristics of the three qubit allocation strategies:

Encoding Method	**Qubit Count**	**Num of Qubit**	**Cost Function Simplicity**
Naive Matrix (NxN)	N^2	High	Simple
Logarithmic Permutation	$\lceil \log_2(n!) \rceil$	Low	Complex (impractical)
Node-Visit-Time	$n \cdot \lceil \log_2(N) \rceil$	Moderate	Moderate

In conclusion, while the naive approach is conceptually straightforward and suitable for initial experimentation, it becomes infeasible for larger instances. The logarithmic method offers minimal qubit usage but at the cost of interpretability and constraint encoding. The node-visit-time encoding offers a middle ground, enabling manageable circuit sizes while supporting effective constraint modeling within the QAOA framework.

7.5 Cost Function for Naive Matrix-Based Encoding

The cost function in our QAOA implementation of the VRP is designed to minimize the total travel distance while ensuring the solution represents a valid route. This involves two main components:

1. **Distance Cost:**: This term represents the total travel distance, which we aim to minimize.
2. **Constraint Penalties**: These terms penalize invalid routes that violate the VRP's constraints, such as visiting a customer more than once or not visiting a customer at all.

Let's consider representing the connections between locations using binary variables x_{ij}, where $x_{ij} = 1$ if we travel from location i to location j, and $x_{ij} = 0$ otherwise. Let d_{ij} be the distance between location i and location j. We can formulate the cost function as:

$$C(x) = Z_1 \sum (d_{ij} \cdot x_{ij}) + Z_2 \sum \left(\sum x_{ij} - 1 \right)^2 + Z_3 \sum \left(\sum x_{ji} - 1 \right)^2$$

where Z_1, Z_2, Z_3 are penalty parameters. In most cases, it makes sense to set $Z_2 = Z_3$.

7.6 Cost Function for Logarithmic Permutation Encoding

In the logarithmic permutation encoding approach, the solution space is limited to the factorial number of valid customer permutations. Each quantum state corresponds to an index in the set of valid permutations of customers. Therefore, cost evaluation must operate over the decoded bitstring, treating it as a lookup into a predefined set of valid routes.

Formally, let $|b\rangle$ be a basis state in the computational basis of the k-qubit system. This bitstring is interpreted as an integer index. Let $P = \{r_1, r_2, \ldots, r_{n!}\}$ be the ordered list of all valid customer visit sequences (without the depot). Each state $|b\rangle$ corresponds to a specific route r_i, where $i = \text{int}(b)$.

To compute the travel cost, the full route is formed by adding the depot at the beginning and end of the customer sequence: $[0, r_i[1], r_i[2], \ldots, r_i[n], 0]$, and the total distance is calculated based on this path.

The cost function is defined as:

$$C(|b\rangle) = \begin{cases} \text{TotalDistance}(r_i), & \text{if } i < n! \\ Z_4, & \text{otherwise (invalid index)} \end{cases}$$

Here, TotalDistance(r_i) is the sum of distances along the route that starts and ends at the depot, and Z_4 is a large penalty applied to any bitstring that refers to an invalid index (i.e., $i \geq n!$).

This cost function is difficult to implement directly as a Hamiltonian because it requires knowing the total travel cost for each possible bitstring in advance. As a result, it typically requires classical pre-processing and post-sampling evaluation. While this encoding uses very few qubits, it trades off ease of modeling and scalability. In the case of VRP, generating the full list of valid routes and their distances is already computationally expensive and similar to classical method problem-solving.

7.7 Cost Function for Node-Visit-Time Encoding

In the node-visit-time encoding, each customer is assigned a binary register that represents their position in the route, also called their "visit time." The complete route is reconstructed by sorting all customers based on their assigned visit times. The vehicle is assumed to start and end at the depot (location 0), so the final route becomes: depot $\to$ customer$_1$ $\to$ customer$_2$ $\to \ldots \to$ customer$_n$ $\to$ depot.

Let each customer i be assigned a visit time value v_i decoded from its corresponding group of qubits. Once all v_i values are obtained, we sort the customers by these values to determine the order in which they are visited.

The cost function consists of three main components:

1. **Travel Distance (Objective)**: This is the total distance traveled through the full route. If the sorted customer order is $[c_1, c_2, \ldots, c_n]$, the route is $[0, c_1, c_2, \ldots, c_n, 0]$. The total travel cost is:

$$C_{\text{dist}} = d_{0,c_1} + \sum_{i=1}^{n-1} d_{c_i,c_{i+1}} + d_{c_n,0}$$

where $d_{i,j}$ is the distance between location i and location j.

2. **Uniqueness Constraint (No Duplicate Positions)**: We must ensure that each customer is assigned a unique visit time. If two customers are assigned the same position (e.g., both want to be second in the route), the solution is invalid. We penalize such cases by:

$$C_{\text{dup}} = Z_5 \cdot \sum_{\text{time}=1}^{n} (\text{count}(\text{time}) - 1)^2$$

where count(time) is the number of customers assigned to that visit time. If all visit times are unique, this penalty is zero.

3. **Out-of-Range Constraint (Invalid Bitstrings)**: Since each visit time is represented by a binary number, some combinations may produce values outside the valid range $[1, n]$. For example, if 2 bits are used per customer (allowing values 0 to 3),

then with 3 customers, a value of 0 or 4 is invalid. We penalize any customer with an out-of-range visit time:

$$C_{\text{out}} = Z_6 \cdot \sum_{i=1}^{n} \text{Invalid}(v_i)$$

where $\text{Invalid}(v_i)$ is 1 if v_i is less than 1 or greater than n, and 0 otherwise.

The final cost function is the combination of the three components:

$$C = C_{\text{dist}} + C_{\text{dup}} + C_{\text{out}}$$

This formulation makes it possible to balance between minimizing the route length and enforcing valid and unique visit assignments. Some parts of the cost function can be encoded directly into the quantum Hamiltonian, while others (such as the duplicate and out-of-range penalties) may be more efficiently applied in a classical post-processing step.

7.8 Qubit Allocation for Depots and Vehicles

The choice of encoding depots and vehicles to qubits directly affects both the computational feasibility and the performance of the quantum circuit. In this study, we evaluate and compare three distinct approaches to qubit allocation:

1. One-Hot Encoding,
2. Separator Combinatorial Encoding, and
3. Hybrid/Optimized Encoding.

Each method presents trade-offs in terms of qubit efficiency, implementation complexity, and suitability for enforcing constraints.

One-Hot Encoding
One-hot encoding is a widely used approach in classical optimization problems and involves assigning a separate qubit for each possible state or assignment. In the context of VRP, each node (depot, vehicle, and customer) is represented using a binary qubit string where only one of the bits is set to 1, corresponding to the active assignment, while all others are set to 0. This approach guarantees a straightforward representation of the problem but requires a large number of qubits as the number of locations (depots, vehicles, and customers) grows.

<u>Qubit Calculation</u>
Consider a scenario with d depots, v vehicles, and n customers. - The qubits for depot encoding would require:

$$Q_{\text{depot}} = d$$

- The qubits for vehicle encoding would require:

$$Q_{\text{vehicle}} = v$$

- The qubits for depot-to-vehicle assignment would require:

$$Q_{\text{depot-vehicle}} = v \cdot d$$

- The qubits for vehicle-to-customer assignment would require:

$$Q_{\text{vehicle-customer}} = v \cdot n$$

Thus, the total qubit count for One-Hot Encoding is:

$$Q_{\text{total}} = Q_{\text{depot}} + Q_{\text{vehicle}} + Q_{\text{depot-vehicle}} + Q_{\text{vehicle-customer}}$$

$$Q_{\text{total}} = d + v + v \cdot d + v \cdot n$$

Separator Combinatorial Encoding
The separator combinatorial method is an alternative encoding strategy where separators (such as '-1') are used to demarcate distinct groups or transitions in the routing process. This approach reduces the number of qubits by leveraging combinatorial structures rather than representing each possible state directly. It is particularly useful when working with problems that have natural divisions, such as a depot and its associated vehicles or customers.

Example
For the same 3 depots and 5 vehicles setup, a separator combinatorial approach could encode the depot-to-vehicle assignment as a binary index mask, where each bit represents whether a vehicle is assigned to a particular depot. For example, the depot-to-vehicle encoding could use '00011' to indicate that Vehicles 1 and 2 are assigned to Depot 1, '00110' for Depot 2, and '01000' for Depot 3. This reduces the total number of qubits needed by representing groupings and transitions more compactly.

Qubit Calculation
For Separator Combinatorial Encoding, the qubit count is as follows: - Depot-to-vehicle encoding requires:

$$Q_{\text{depot-vehicle}} = \lceil \log_2 [(v + (d-1))!] \rceil$$

- Vehicle-to-customer encoding requires:

$$Q_{\text{vehicle-customer}} = \lceil \log_2 [(n + (v-1))!] \rceil$$

Thus, the total qubit count for Separator Combinatorial Encoding is:

$$Q_{\text{total}} = Q_{\text{depot-vehicle}} + Q_{\text{vehicle-customer}}$$

$$Q_{\text{total}} = \lceil \log_2 [(v + (d-1))!] \rceil \; + \lceil \log_2 [(n + (v-1))!] \rceil$$

Hybrid/Optimized Encoding
The hybrid/optimized encoding method aims to combine the benefits of both one-hot encoding and combinatorial encoding. Instead of directly encoding all possible states for each component (depot, vehicle, customer), this method uses a more efficient representation by

assigning a binary register to each customer or vehicle that reflects their position or assignment in the route. This allows for a more compact encoding of the routing problem, reducing the qubit count while still enabling effective constraint enforcement.

Example

Consider a scenario with 3 depots, 5 vehicles, and 4 customers. The hybrid encoding would allocate:

- 2 qubits for the depot encoding (for 3 depots),
- 3 qubits for the vehicle encoding (for 5 vehicles),
- 4 qubits for the depot-to-vehicle assignment (15 possible arrangements for 3 depots and 5 vehicles),
- 5 qubits for each vehicle-to-customer assignment (20 possible arrangements for 5 vehicles and 4 customers).

Qubit Calculation

Thus, the total qubit count for Hybrid/Optimized Encoding is:

$$Q_{\text{total}} = \lceil \log_2(d) \rceil + \lceil \log_2(v) \rceil + \lceil \log_2(d \cdot v) \rceil + \lceil \log_2(v \cdot n) \rceil$$

Comparison

The following table summarizes the characteristics of the three qubit allocation strategies:

Encoding Method	**Num of Qubits**	**Cost Function Simplicity**
One-hot encoding	High	Simplest
Separator Combinatorial Encoding	Low	Complex
Hybrid/Optimized Encoding	Moderate	Moderate

In conclusion, while one-hot encoding is conceptually simple and easy to implement, it becomes inefficient as the problem size increases, leading to high qubit counts. The separator combinatorial method reduces the qubit count by encoding the problem in a more compact manner but may introduce complexity in handling constraints. The hybrid/optimized encoding strikes a balance between qubit efficiency and constraint handling, making it a promising candidate for solving larger VRP instances in a quantum setting.

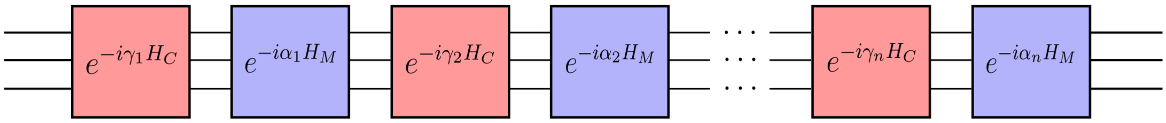
7.9 Quantum Circuit and Hamiltonian

The cost function $C(x)$ is translated into a quantum Hamiltonian H_C by replacing the binary variables x_{ij} with Pauli-Z operators. The quantum circuit alternates between the cost Hamiltonian and a mixing Hamiltonian, usually using a rotation gate. The depth p determines the number of layers. The unitaries used are:

$$U_C(\gamma) = e^{-i\gamma H_C}, \quad U_M(\beta) = e^{-i\beta H_M}$$

The γ (gamma) parameter as well as β (beta) in the QAOA circuit control the evolution under the cost and mixing Hamiltonians, respectively. We can start with random initial values for these parameters and then optimize them using a classical optimizer.

In an n-depth Hamiltonian, the circuit will looks something like this:



7.10 Memory Problem

We encountered a significant memory problem when attempting to simulate the QAOA circuit for even a small VRP instance (5 customers, 1 depot). The PennyLane simulator attempts to store the entire quantum state vector, requiring 2^k complex amplitudes. For $n = 5$, $k = 36$, which leads to:

$$2^{36} \cdot 16 \text{ bytes} = 2^{40} \text{ bytes} = 1 \text{ Terabyte}$$

Therefore, using $(n + 1)^2$ qubits would have resulted in a memory requirement of exactly 1 Terabyte. This is of course impractical. So we need to find another k where $2k > n!$. Which is why we can use another way of encoding such as logarithmic permutation on separator combinatorial encoding.

7.11 Addressing Constraints

This study addresses real-world constraints in the Vehicle Routing Problem (VRP), such as traffic congestion, weather conditions, and road accidents, through classical preprocessing techniques. These constraints are not directly encoded into the quantum model. Instead, they are used to modify the graph structure that is later passed to the quantum optimization algorithm. For example, areas with heavy traffic or frequent accidents are assigned higher edge weights to reflect longer travel times or added risks.

This hybrid approach reduces the complexity of the quantum circuit while still incorporating practical considerations. Given current limitations in quantum hardware, such as limited qubit count and short coherence time, it is more practical to handle dynamic, real-world factors using classical methods before quantum encoding. Once the graph has been adjusted to include these constraint-aware edge costs, it is converted into a QUBO or Hamiltonian format suitable for quantum algorithms like QAOA.

Although the primary focus is on quantum optimization, several stages (including data preparation, constraint integration, and solution verification) remain within the classical domain. This results in a hybrid quantum-classical workflow, where the classical layer prepares a realistic problem representation, and the quantum layer concentrates on exploring optimal or near-optimal solutions.

7.12 Comparison of Operation Number (Classical vs. Quantum)

- **Classical**: Classical algorithms for the VRP, such as exhaustive search, require exploring all possible permutations, resulting in approximately $(n+1)!$ operations for n customers.
- **Quantum**: QAOA aims to find approximate solutions with potentially fewer operations, but the exact reduction depends on the problem instance and QAOA parameters. The higher the depth p, the higher the complexity. The circuit qubit requirements and circuit complexity will also increase. The total quantum gate operation is estimated as $s \cdot p \cdot (n+1)^2$ operations (which is less than $(n+1)!$), with controlled iterations s and circuit depth p.

7.13 Other Notes: Using v Vehicles and d Depots

Extending to multiple depots and vehicles, we get:

$$T(n, d, v)_{Classical} \approx (n+d)! \cdot v^n$$

$$T(n, d, v, p, s)_{Quantum} \approx s \cdot p \cdot v \cdot (n+d)^2$$

- **Classical**: With v vehicles and d depots, the classical problem's complexity increases significantly, as we need to consider all possible assignments of customers to vehicles and depots.
- **QAOA**: QAOA can potentially handle this increased complexity, but the qubit requirements and circuit complexity will also increase. We would need to encode the vehicle assignments and depot selections into the quantum state.

8 Other Important Research on Quantum Optimization

8.1 Efficient Quantum Allocation

Balu *et al.* [22] proposed an innovative approach to the Traveling Salesman Problem (TSP) using a single qubit by encoding a cost Hamiltonian into amplitude parameters. This minimalist method opens possibilities for quantum optimization on constrained devices and highlights the potential for compact representations of complex NP problems.

Efficient qubit allocation is essential for practical quantum programs. Baek *et al.* [23] introduce type-based allocation for first-order quantum programming languages. Shi *et al.* [24] address distributed quantum computing scenarios, while Li *et al.* [25] propose an exact method tailored for NISQ architectures. These allocation strategies complement quantum optimization by improving resource usage.

8.2 Quantum Algorithms on TSP and VRP

Several works leverage quantum annealing to solve combinatorial problems. Wang *et al.* [26] combine QUBO formulations and Graph Neural Networks to tackle TSP. Kochenberger *et al.* [27] provide a comprehensive Ising formulation of NP problems, useful for annealing frameworks. Earlier work by D-Wave researchers such as Yu *et al.* [28] explore hybrid methods to solve capacitated VRP using annealers.

Several quantum algorithms target TSP directly. The recent work by Surya *et al.* [29] presents an efficient quantum algorithm with promising performance bounds. Another contribution from Younes [22] extends the application to a simplified setting, showing how TSP can be embedded into reduced quantum systems for resource-efficient solving.

8.3 Quantum Circuit Optimization

Reducing quantum circuit complexity is crucial for scalable optimization. Liu *et al.* [30] propose an optimization-driven reduction technique to minimize gate depth and number of operations, thereby improving success probability on Noisy Intermediate-Scale Quantum (NISQ) devices.

9 Proposed Methodology

9.1 Quantum Computing Approaches

- Formulation: Represent the VRP in a graph format which suitable for QAOA, Quantum Walk algorithms, and VQE.
- Theoretical Calculations: Analyze mathematical models and complexity to understand how quantum algorithms can outperform classical methods in VRP domain. This includes identifying cases where quantum optimization provides faster solutions or better approximations.
- Implementation: Develop and simulate quantum circuits using platforms like PennyLane or Qiskit (or other compatible library)

9.2 Benchmarking Strategy

- Classical Comparison: Utilize established solution using classical methods to obtain baseline solutions.
- Performance Metrics: Compare quantum and classical approaches based on solution quality, computational time, and scalability.

9.3 Datasets

- Data Acquisition: Employ real-world datasets from public sources[2, 6, 7, 8, 9, 10, 11].

- Data Processing: Transform raw data into graph representations that will be compatible with quantum algorithms.

10 Expected Outcomes and Contributions

- Development of quantum-based approximation methods: Providing new strategies for solving VRP using QAOA.
- Comparative Analysis: Offering insights into the practical advantages and limitations of quantum (in this case, QAOA) versus classical approaches for VRP.

11 Challenges and Limitations

Quantum Hardware Constraints: Real quantum cloud is expensive. Use a simpler graph to simulate in real quantum computer and use a simulator (or a pseudo-theoritical calculations) to simulate bigger graph.

12 Approximate Timeline

First Semester (50 Points):

- Week 1: Topic Formulation
  - Finalize research topic and objectives.
  - Output: Topic finalized by end of week 1.
- Week 2–4: Literature review.
  - Explore quantum algorithms and circuits (especially QAOA and Quantum Annealing)
  - Review existing research on quantum optimization for VRP
  - Output: Gain more hands-on experience with quantum algorithms.
- Week 5–6: Research Methodology development.
  - Develop research methodology to approach VRP using quantum optimization.
- Week 7–8: Research Proposal / Plan Writing.
  - Write research proposal with objectives, methodology, and case study specifics.
  - Output: Research proposal submitted by end of week 8.
- Week 9–11: Data acquisition.
  - Collect data from real-world sources (City of Casey waste collection, road networks, etc.).

- Output: Structured dataset for VRP.

- Week 11: Oral Presentation
  - Prepare and deliver a 10-15 minute oral presentation on research progress.
- Week 12: Continuing quantum algorithms formulation.

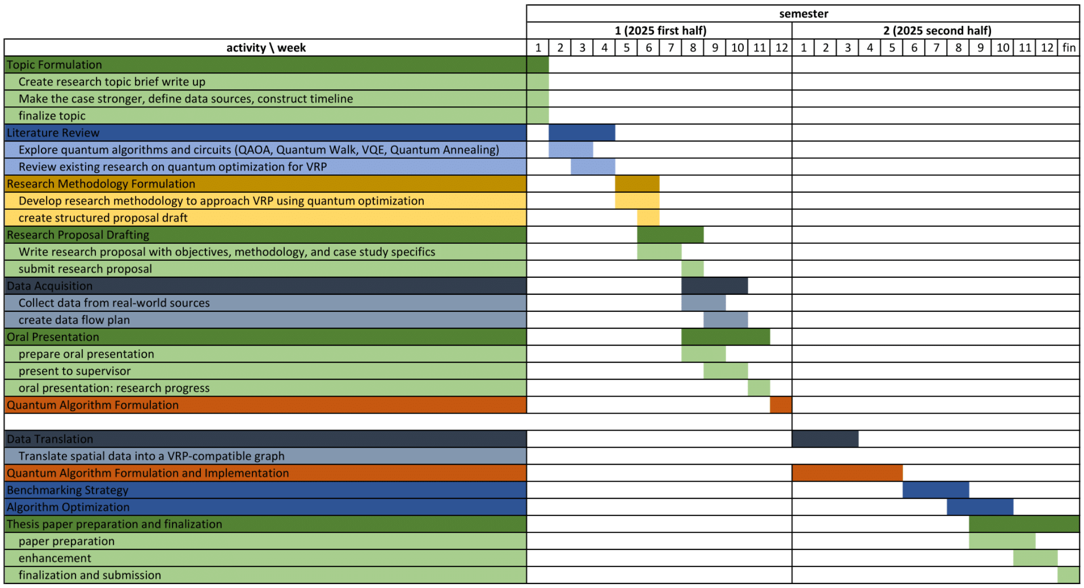
Second Semester (50 Points):

- Week 1–3: Data Translation (Spatial to VRP Graph)
  - Translate spatial data into a VRP-compatible graph.
  - Output: Transformed VRP data representation.
- Week 1–5: Quantum Algorithm Formulation
  - Develop quantum optimization algorithms (QAOA, Quantum Walk) for VRP.
  - Start with basic formulations and progress to complex VRP instances.
  - Output: Working quantum algorithms.
- Week 6–8: Benchmarking Strategy
  - Design a strategy to compare quantum optimization results with classical algorithms.
  - Collect performance metrics (solution quality, computational time, scalability).
- Week 8–10: Algorithm Optimization
  - Refine quantum algorithms to improve performance and scalability.
- Week 11–12: Thesis Defense
  - Final Week: Thesis Submission.
  - Output: Complete and submit a written thesis (25,000–30,000 words).

RESEARCH PROJECT TIMELINE

Quantum Optimisation for the Vehicle Routing Problem

Drestanto Muhammad Dyasputro - 1450107



References

[1] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.

[2] City of Casey. Waste facility locations. `https://data.casey.vic.gov.au/explore/dataset/waste-facility-locations`, n.d.

[3] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[4] S. Marsh and J. B. Wang. A quantum walk-assisted approximate algorithm for bounded np optimisation problems. *Quantum Information Processing*, 18(3):61, 2019.

[5] M. Yousefikhoshbakht and E. Khorram. Solving the vehicle routing problem by a hybrid meta-heuristic algorithm. *Journal of Industrial Engineering International*, 8:1–9, 2012.

[6] City of Casey. Public litter bins. `https://data.casey.vic.gov.au/explore/dataset/public-litter-bins`, n.d.

[7] City of Casey. Waste collection areas. `https://data.casey.vic.gov.au/explore/dataset/waste-collection-area`, n.d.

[8] VicRoads. Victoria road network. `https://vicdata.vicroads.vic.gov.au/portal/home/group.html?id=82b544768d5b4c3cbd4a79a4df322984`, n.d.

[9] VicRoads Open Data. Traffic volume. `https://vicroadsopendata-vicroadsmaps.opendata.arcgis.com/datasets/vicroadsmaps::traffic-volume/about`, n.d.

[10] City of Casey. Rainfall data. `https://data.casey.vic.gov.au/explore/dataset/rainfall-data`, n.d.

[11] Data VIC. Victoria road crash data. `https://discover.data.vic.gov.au/dataset/victoria-road-crash-data`, n.d.

[12] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, Philadelphia, PA, USA, 2014.

[13] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[14] T. Bennett, E. Matwiejew, S. Marsh, and J. B. Wang. Quantum walk-based vehicle routing optimisation. *Frontiers in Physics*, 9:730856, 2021.

[15] M. Alsaiyari and M. Felemban. Variational quantum algorithms for solving vehicle routing problem. In *Proc. 2023 Int. Conf. Smart Computing Applications (ICSCA)*, pages 1–4, 2023.

[16] H. Irie, G. Wongpaisarnsin, M. Terabe, A. Miki, and S. Taguchi. Quantum annealing of vehicle routing problem with time, state and capacity. In *Quantum Technology and Optimization Problems: First International Workshop, QTOP 2019*, pages 145–156. Springer, 2019.

[17] D. Fitzek, T. Ghandriz, L. Laine, M. Granath, and A. F. Kockum. Applying quantum approximate optimization to the heterogeneous vehicle routing problem. *Scientific Reports*, 14(1):25415, 2024.

[18] I. D. Leonidas, A. Dukakis, B. Tan, and D. G. Angelakis. Qubit efficient quantum algorithms for the vehicle routing problem on nisq processors. *arXiv preprint arXiv:2306.08507*, 2023.

[19] S. Marsh and J. B. Wang. Combinatorial optimization via highly efficient quantum walks. *Physical Review Research*, 2(2):023302, 2020.

[20] Toufan Diansyah Tambunan, Andriyan Bayu Suksmono, Ian Joseph Matheus Edward, and Rahmat Mulyawan. Quantum annealing for vehicle routing problem with weighted segment. In *AIP Conference Proceedings*, volume 2906. AIP Publishing, November 2023.

[21] N. Mohanty, B. K. Behera, and C. Ferrie. Solving the vehicle routing problem via quantum support vector machines. *Quantum Machine Intelligence*, 6(1):34, 2024.

[22] Radhakrishnan Balu and Ahmed Younes. Solving the travelling salesman problem using a single qubit. *arXiv preprint arXiv:2407.17207*, 2024.

[23] Sungjin Baek, Zhihao Qian, Yihan Zhang, and Frederic T Chong. Type-based qubit allocation for a first-order quantum programming language. *arXiv preprint arXiv:2306.01856*, 2023.

[24] Yingkai Shi, Yudong Cao, Hanrui Tang, Huanrui Fan, Ke Pei, Tianshi Wu, and Jason Cong. Qubit allocation for distributed quantum computing. *2023 IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 695–709, 2023.

[25] Guoming Li, Yu Ding, Yuan Xie, and Mingyu Zhou. An exact qubit allocation approach for nisq architectures. *Quantum Information Processing*, 19(2):1–19, 2020.

[26] Bowen Wang, Ziqi Li, Ping Chen, Wenjun Hu, and Yuanyuan Ma. Quantum annealing and graph neural networks for solving tsp with qubo. *arXiv preprint arXiv:2402.14036*, 2024.

[27] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Matthew Lewis, Haibo Wang, Yang Wang, and Taylor Wuster. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

[28] Zhen Yu, Shuai Sun, Hongjun Xu, and Jianjun Wu. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *arXiv preprint arXiv:1811.07403*, 2018.

[29] Gautham Surya, Moin Khan, and Andre Chailloux. An efficient quantum algorithm for the traveling salesman problem. *Cryptology ePrint Archive*, 2024:626, 2024.

[30] Sitao Liu and Yongshan Lu. Optimization driven quantum circuit reduction. *arXiv preprint arXiv:2502.14715*, 2024.