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 problem in logistics studied for finding optimum delivery routes for several vehicles in order to visit a series of customers. Research on VRP and its iterations has indicated increased interest across the years. There are many different versions of the VRP, since there are many ways in which this problem can be posed, with different assumptions so one must be aware that it has many variants.

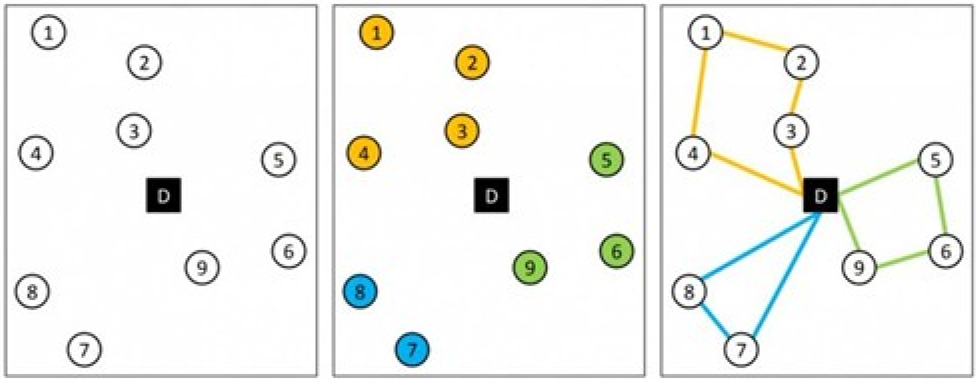
The problem is NP-hard [1] and classical solutions, due to computational limitations are unable to solve large-scale instances. The City of Casey Waste Facility[2] dataset is rich with the details of waste facilities, public litter bins and collection points which makes it an appropriate candidate for VRP optimization applications. The study incorporates a set of realistic VRP constraints, i.e., traffic information and weather and road incidents data, to construct an applicable VRP model. Due to the NP-hard nature of the problem, classical methods are so far incapable to perform large scale optimization.

Quantum computation offers potential techniques for solving problems of this type. QAOA (by Farhi et al. 2014) and Quantum walk technique of Marsh et al. 2019 provide useful solutions to NP-hard problems. The objective of the analysis is to develop quantum approaches for deriving some approximate response solutions for the VRP.

1.1 The Vehicle Routing Problem Explained

In the application, Virtual Routing Problem is further spread to three aspects: logistics operations, supply chain operations and delivery distribution system. The Vehicle Routing Problem has some variants as Capacitated VRP (CVRP) and Time-Window VRP (VRPTW) and Stochastic VRP (SVRP) that add specific constraints that make the solution more difficult. Classical methods are unable to find optimal solutions, and the running time of these methods increases exponentially when dealing with larger problems.

The presented visual picture is a method for VRP solving [3]:



Unassigned Customers (left panel)

- Customers (labelled by numbered circles) are distributed around a central depot (the black square marked "D").
- No routes have been specified at this point.

Customer Segmentation (Middle Panel)

- Customers are divided into different sets depending on their locations and other criteria (demand or distance, this is the heuristic part).
- Each hue is a vehicle that will serve a subset(options heuristic) of the customers (NOT: each color is a vehicle and every vehicle will serve a subset of the customers).

Optimal Paths (Right Panel)

- Optimized routes of vehicles, which begin and end at the depot (*"D"*). Each tour is such that all customers are served in an efficient manner, and the total travel time and costs are minimized.

2 Scope and Limitations

2.1 Scope: Waste Collection in City of Casey

For a pragmatic evaluation of the VRP we use a case study of actual waste collection services. We opt for the waste collection setting because it's a publicly available dataset with a well-structured representation. The research concentrates on the City of Casey due to its data granularity, a perfect dataset for modeling and validating VRP solutions. Three main elements depot, customer and route are utilized to describe the VRP model:

- **Depots:** Depots are waste collection operations points where vehicles start and end their routes.
- **Customers:** Locations of the public litter bins are the customer nodes for which they are required to receive collection and by combining with waste collection area data a full coverage of all the waste pickup nodes can be achieved over the city.

- **Routes:** Following Victoria's road network, vehicle travel paths from depots to collection sites.

Further constraints representing realistic operational complications can increase the model's accuracy:

- Relatively increased traffic level will increases travel time.
- Weather condition directly impacts travel time.
- Road Crashes can add delay to trips.

The data on these three constraints are publicly available: depot [2], customer [4, 5], and route [6]. Real-world constraints' data are also available in victoria open data: traffic [7], weather [8], and crashes [9].

2.2 Limitation

In this study, we will only deal with a general VRP without calculating the capacity of vehicle (CVRP solution: Capacitated VR) or time limit (Time-Window VRP: VRPTW). Contraints also limited to three type mentioned in the previos section (Scope).

Quantum computing is a new approach to solve VRP by investigating and evaluating the various solutions. Quantum computing techniques such as QAOA can be applied to the optimization of the VRP. This study limits the practical application of quantum algorithm for VRP in terms of scale (number of nodes) due to hardware limitations.

3 Motivation and Challenges

3.1 Traditional VRP Methods

Traditional methods for solving the VRP include:

- Exact algorithms: Methods like the branch-and-bound for CVRP[10]. Heuristic Techniques [11]such as Genetic Algorithm[11] and Ant Colony Optimization[11] that offer usable sub-optimal solutions.

3.1.1 Quantum Programming Approaches for VRP

Quantum Optimization Techniques Quantum optimization techniques generate better solutions compared to the classical optimization techniques. TDS studies on quantum algorithm advantage against the classical optimization algorithms solving NP-hard problems especially VRP:

- Quantum Approximate Optimization Algorithm(QAOA) is an approach for finding the approximate solutions of combinatorial problems using the power of quantum superposition and entanglement dynamics[12].

- Quantum walk-based optimization to use analogs of classical random walks to solve search and optimization problems[13].
- The Variational Quantum Eigensolver (VQE)[14] is one of such methods for the VRP but is limited to small-scale instances.
- Authors in [15] modeled a time-table with a QUBO formalism called Quantum Annealing.

4 Research Objectives and Research Questions

4.1 Objectives

1. Design Quantum Optimization Techniques for VRP (in this scenario, for waste-collection problem) using Quantum Algorithms [RQ1, RQ2].
2. Explains why VRP solution based on quantum approaches may perform better than classical approaches [RQ3].

4.2 Research Questions

- RQ1: How can spatial data of VRP can be converted and formulated into a graph?
- RQ2: How can we allocate qubits to solve VRP graph in using QAOA?
- RQ3: To what extent are quantum algorithms efficient and scalable compared to best performing classical VRP solvers?

5 Literature Review

5.1 Quantum Approximate Optimization for VRP

QAOA is a good candidate for VRP because it can efficiently provide approximate solutions for hard optimization problems such as vehicle routing (VRP). VRP is the problem of selecting the optimal routes for a number of vehicles considering constraints such as distance, capacity, and time windows. Because VRP is NP-hard, the state-of-the-art algorithms are less suitable to solve those large instances. QAOA utilizes quantum superposition and interference to consider a superposition of multiple candidate solutions at the same time that could allow it to find good routes faster than classical heuristics. Mathematically, QAOA expresses the VRP as a cost function H_C that corresponds to the total travelled distance, or other constraints, and the mixing function H_M that samples the algorithm through different routes. The quantum circuit is given by:

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

where $U_C(\gamma) = e^{-i\gamma H_C}$ enforces the VRP constraints, and $U_M(\beta) = e^{-i\beta H_M}$ permits transitions between different solutions.

Fitzek et al. [16] adapted the strategy for vehicle constraints to increase efficiency. Leonidas et al. [17] designed QAOA with decreased qusal bit needs for small quantum machines. Farhi et al. [12] first proposed QAOA and demonstrated that the results can be improved by using larger circuits, even though real quantum computer has noise profiles.

5.2 Other Quantum Algorithm to Consider

5.2.1 Quantum Walk-Based Methods

Quantum walks, the quantum equivalent of classical random walks, are also investigated for optimization of the VRP. Quantum walks are particularly attractive in the context of VRP because they can traverse a large search space more efficiently through quantum interference and superposition. In contrast to the conventional random walk where the transition probabilities depend exclusively on the preceding step, quantum walks have maintained phase coherence, and can travel across solution space at a greater speed. The unitary operator that describes the evolution of the discrete-time quantum walk is:

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

where S (C) is the shift (coin) operator. Bennett et al. [13] used this approach for vehicle routing, and showed that quantum walks could find good paths by efficiently searching for good solutions between all the alternative routes. Approximations algorithms based on quantum walks were proposed by Marsh and Wang[18] displaying its usefulness in the case of bounded NP–hard problems. A subsequent study refined the quantum walk definition to improve the quality of solution and efficiency of the combinatorial optimization[19]. These results hint at potential quantum advantage for VRP over classical heuristics through quantum walks, however limited by the current quantum hardware architectures.

5.2.2 Quantum Annealing

Researches also recommend quantum annealing for optimization in VRP. Tambunan et al. (2023) analyzed quantum annealing for VRP with weighted segments and attempted to incorporate dynamic constraints into the optimization. Similarly, Irie et al. (2019) developed a QUBO model of VRP with time, state, and capacity constraints, demonstrating quantum annealing's suitability for intricate routing problems.

Quantum annealing (QA) is known as an heuristic algorithm for optimization problems. It uses the principle of quantum mechanics, which is quantum superposition and tunneling, for the global minimum search of a cost function. A Hamiltonian for quantum annealing for VRP in its common form is defined as:

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

where H_P is the problem Hamiltonian representing the VRP objective and constraints, and HD the driver Hamiltonian describing the evolution of the quantum state. The mixing between these Hamiltonians is driven by the functions $A(t)$ and $B(t)$ during annealing. The system, at the end of the annealing process, hopefully, will reach the lowest energy state and constitutes a solution to the VRP. For example, recent work, including Irie et al. [15] and Tambunan et al. [20] show the potential of quantum annealing in solving challenging VRP with time variety and consideration as well as weighted constraints, and therefore using it in optimization of logistics problem in a real world.

5.2.3 Hybrid and Machine Learning Based Approaches

In addition to conventional quantum approaches, researchers have investigated mixed quantum-classical approaches and implemented machine learning for VRP. Hybrid techniques such as QSVMs (Quantum Support Vector Machines) and VQAs (Variational Quantum Algorithms) are promising. For instance, the decision function for QSVM is:

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

where, and α_i and b are parameters that are trained. The accuracy of route classification is improved, and the optimization of VRP in general is also promoted. In contrast, VQAs employ parameterized quantum circuits and classical optimizers to update quantum states iteratively to minimize the objective function. The parameters in the quantum circuit are tuned by a classical optimizer to search for the best routing decisions under certain constraints. These combined approaches leverage the high-dimensional state space that quantum computing is capable of exploring but share a practical ambit for hardware-constrained environments that is demonstrated in Alsaiyari and Felemban[14] and by Mohanty et al. [21].

5.3 Conclusion

Studying quantum VRP algorithms has pointed to some of the most promising directions, in particular, based on QAOA and quantum annealing. Although some recent theoretical researches show promising quantum advantages over classical counterparts, actual quantum implementations are still bottlenecked by hardwares. Future research will target realistic large-scale VRP instances, to be improved in qubit efficiency and hybrid algorithms and perform comparisons with the current state-of-the-art classical solvers, aiming to extract the full potential of quantum computing using VRP.

6 Methodology: QAOA for VRP

6.1 Vehicle Routing Problem (VRP) and Graph Modeling

The VRP (Ve-hicle Routing Problem) is a well-known optimization problem, which aims to find the optimal routes for one or many vehicles in order to serve a given set of customers. For simplicity, we consider a scenario with one depot and one vehicle. In our case, we are given customer and depot co-ordinate where they are located in space. To transform this spatial problem into a computational problem, we model each depot and each customer as a node of a graph. The connections between the nodes are shown as edges, and each edge is associated with a weight indicating the distance (or travel time) between the nodes. We can use classical algorithms like Dijkstra's or A* to build this graph and to calculate the edge weights. Given n customers and 1 depot we can define an nxn matrix (where n in this case is n+1) for an adjacency matrix the cell (i,j) represents the weight of edge between node i and node j*this matrix will have zeros on its diagonal and be symmetric (indicating that the transpose is equal to the matrix)

6.2 Quantum Approximate Optimization Algorithm (QAOA) Overview

QAOA is a quantum algorithm (or more accurately a hybrid quantum-classical algorithm) that aims to approximate solutions to combinatorial optimization problems. By contrast, exact quantum algorithms will give us the optimal solution (which is usually extremely hard or even impossible to find completely). QAOA is not after exact optimal solutions, but rather good (although not necessary optimally good) ones, obtained in a reasonable time. The QAOA process involves:

1. **Cost Function**: Specifying a cost function which models the desires of the problem. In our VRP, this function accounts for the overall distance or travel time of a route.
2. **Hamiltonian**: Casting the cost function as a quantum Hamiltonian.
3. **Quantum Circuit**: Generation of a parameterized quantum circuit, in which it follows switching between applying the cost Hamiltonian and the mixing Hamiltonian. Optimization:Updating the parameters (gamma and beta) of the quantum circuit iteratively to minimize the cost function, using a classical optimizer.
4. **Approximate solution**: obtain an approximate solution from the final quantum state.

Key features of QAOA include:

- **Sub-Optimality**: The QAOA only give an approximate (not always optimal) solution.
- **Parameterized Circuits**: The parameters (gamma and beta) of the quantum circuit are learned from data classically.
- **Depth (p)**: The depth of the QAOA circuit p can set the number of iterations and quality of the solution.

6.3 VRP Statement in Quantum Qubits

In order to use QAOA for VRP, we should map the VRP into quantum bits. One naïve option is to materialize all potential couples between the nodes. However, this is not optimal.

- **Graph Representation in Matrix:** as mentioned previously we represent the graph using an (n+1) x (n+1) matrix.
- **Qubit Assignment**: A core difficulty is to store the order in which the customers are visited. We don't have enough with $n + 1$ qubits. For the VRP it isn't enough

to draw only binary decision, but also the order of visit. We found that $2^k > n!$ is enough, where k is the number of qubits. This is because there are $n!$ permutations of order among n customers. To find k in an easy way is to find the matrix number of item. If $k = (n+1)^2$, then $2k$ is going to be a way way bigger than $n!$. We can prove this through mathematical ratio along with tools such as stirling's approximation and logarithm)

6.4 Qubit Mapping Improvements and Discussion

One of the deciders for an efficient approach of the QAOA for the VRP is a in terms of qubit allocation strategy and a scaleable approach. The coding scheme we employ has a direct impact on the computational power and the expressiveness of the quantum circuit. In this paper, we assess and compare three different qubit placement methods:

1. naive representation based on the adjacency matrix,
2. permutation encoding with logarithmic number, and
3. node-visit-time representation.

There are trade-offs between qubit efficiency, circuit depth, and constraint enforcement that drive the decision on which method to use.

Generic Matrix Encoding
At the simplest level, one could use a single qubit to represent each possible directed edge between nodes, leading to an $N \times N$ grid of qubits for a problem with $N = n + d$ points (customers and depots). This method allows a natural formulation of the cost function - the distances are directly encoded by the edge weights. but the qubit number grows quadratically in N, and simulating this in terms of state vectors is infeasible when N is large, both in the simulation and real hardware setting. Example
Say $N = 6$ (this is 36 qubits) in an N-qubit space of 2^N dimensions. Every qubit q_{ij} corresponds to a directed edge from node i to node j being present in a path. For instance, in a 4-node system the qubit $q_{1,2}$ indicates whether the vehicle moves from location 1 to location 2. The output path is translated as a path that consists of edges whose corresponding qubits are in the state $|1\rangle$. **Logarithmic Permutation Encoding**
This method represents the whole route as permutation of customers using the minimal number of qubits required, that is, $k = \lceil \log_2(n!) \rceil$. Although this approach is theoretically most efficient in terms of number of qubits, it adds significant complexity for encoding the problem in terms of quantum bitstrings and it is less suitable for generating 'meaningful' cost evaluations. Note that designing a quantum Hamiltonian that correctly represents constraints of the VRP under this encoding is very difficult. In addition the validiy checks and constraint penalties can only be performed as a postprocess which degrades in-circuit cost-optimisation.

Example
For $n = 3$ customers, there are $3! = 6$ paths, and $k = \lceil \log_2(6) \rceil = 3$ qubits are all needed. Every measured bitstring of the quantum circuit (eg 010 or 101) is treated as an integer index into a list of pre calculated valid permutations. A bitstring looking like 011 could be

mapped to the customer sequence [1,3,2], resulting in the route $[0, 1, 3, 2, 0]$ with the depot added to front and end. **Node-Visit-Time Encoding (Hybrid Method)**
To keep a balance between the qubit efficiency and the route validity, we present a hybrid method such that each customer is encoded with a fixed-bits binary register which represents the location of their position in the route (i.e., the visit time). This approach uses $n \cdot \lceil \log_2(N) \rceil$ qubits and admits structured constraint enforcement. Each route is restored by reordering the customers by ascending order of the visit times similarity. The configurations that have any invalid position locations (ie. double positions, positions not filled) define penalty terms in which these configurations are penalized with additional constraints in the Hamiltonian.
Example
In the same example, if we have $n = 3$ customers and $N = 4$ nodes in total, then each customer will have a 2 qubit register representing the time of visit. If customer 1 is assigned by qubits q_0, q_1, and the value of them turns out 01, it means that customer 1 is the second visited one in the route. For example, a full bitstring like 01101011 could be interpreted as: customer 1 at position 1, customer 2 at position 2, customer 3 at position 3 and so on to obtain a route of $[0, 1, 2, 3, 0]$. Repeated visit times can be conveniently prevented and punished by calculating repeated integers in the decoded visit-time rec list.

Comparison
The features of the three qubit layout strategies are shown in table 1:

Encoding Approach	**Qubits**	**Qubit Number**	**Simplicity of Cost Function**
Simple Naive Matrix (NxN)	N^2	Naive	High
Logarithmic Permutation	$\lceil \log_2(n!) \rceil$	Low	Complex (not feasible)
Node-Visit-Time	$n \cdot \lceil \log_2(N) \rceil$	Moderate	Moderate

To conclude, a naive approach is conceptaully simple and serves as a good starting point for experimentation, but doesn't scale well to bigger instances. The advent of the logarithmic method, which minimizes the number of qubits required in expression (4) at the expense of expressiveness and encoding, was another result. The node-visit-time encoding opts for a compromise, to secure a manageable circuit size while ensuring QAOA is able to express an effective constraint model.

6.5 Cost function for naive Matrix Based Encoding

In our QAOA VRP cost function, we minimize the total travel distance with the requirement that the solution corresponds to a valid route. It contains two essential parts as follows:

1. **Distance Cost:** This objective is used for minimizing the travel distance.
2. **Constraint penalties**: These are terms for (and thus are penalizing) rules that are imposed in the VRP, such as the fact that you have to visit a customer, and you can't visit a customer more than once in a route.

Suppose we want to express the structure of these locations using binary variables x_{ij}, with the interpretation that $x_{ij} = 1$ if we travel from location i to location j, and $x_{ij} = 0$ otherwise.

Denote by d_{ij} the distance of location i to location j. We may write the cost function as:

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

for some penalty parameters Z_1, Z_2, Z_3. Generally, one may even take $Z_2 = Z_3$.

6.6 Cost Function for Logarithmic Permutation Enoding

For the logarithmic permutation encoding method, the solution space consists only of the factorial number of possible customer permutations. Each quantum state is associated with an index which is in the range of valid permutations of customers. It follows that cost evaluation will have to act on the decoded bitstring and consider it as an index in the set of valid paths.

In other words, we assume, for now, that there exists a basis state $|b\rangle$ in the computational basis of the k-qubit system. This bitstring is read as an integer index. Denote by $P = \{r_1, r_2, \ldots, r_{n!}\}$ be the increasing sequence of all valid customer visit sequences (without the depot). Each state $|b\rangle$ corresponds to a particular route r_i with $i = \text{int}(b)$. The complete route includes both the depot at the beginning and at the end of the customer sequence: $[0, r_i[1], r_i[2], \ldots, r_i[n], 0]$ and the distance in total is computed according to this continuation. The cost function is expressed as:

$$C(|b\rangle) = \begin{cases} \text{TotalDistance}(r_i), & \text{if } i < n! \\ Z_4 & \text{else (not a valid index).} \end{cases}$$

Here, $\text{TotalDistance}(r_i)$ is the sum of distances along the route that starts and ends at the depot, and Z_4 is a big penalty for such bitstring referring to a non-existent index (i.e., $i \geq n!$).

This cost function is not easy to use directly in the form of a Hamiltonian, since we would need to know the total cost for all possible bitstring in the beginning. As such it will often need classical pre-processing and post-sampling analysis. Although this encoding requires very few qubits, it sacrifices ease of modeling and scalability. For VRP, the creation of the entire list of legal routes and their distances is already a computationally expensive (classic method like problem-solving).

6.7 Loss Function for Node-Visit-Time Encoding

With the node-visit-time encoding, each customer is associated with a binary register which tracks its position in the route (its "visit time"). By reordering all customers according to their scheduled arrival times, the entire route is reconstructed. The base case is that the vehicle will go from the depot (location 0) and visit all the customers back to the depot, i.e., the route is: depot $\rightarrow$ customer$_1$ $\rightarrow$ customer$_2$ $\rightarrow \ldots \rightarrow$ customer$_n$ $\rightarrow$ depot. Let each customer i have a visit time v_i associated with it, decoded from a certain group of qubits. After all values of v_i are calculated, we reorder the customers according to the value of these customers.

The objective function is composed of three major parts:

1. **Distance Travelled (Objective)**: The total distance travelled on the whole route. If the sorted customer set is $[c_1, c_2, \ldots, c_n]$, then 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 position i and position j.

2. **Depositionality Constraint (No Double Positions)**: We need to guarantee that each customer visit time be unique. If two customers are given the same order (i.e., both want to be the second in the route), the solution is infeasible. We penalize such cases by:

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

   where count(time) is the number of customers scheduled at that visit time. This penalty is zero if all visit times are unique.

3. **Rage of constraint (Invalid Bitstrings):** Values $\geq n+1$ would be invalid since each visit time is a binary number. For instance 2 bits per customer (values 0, 1, 2 or 3) then with 3 customers a value 0 and 4 are illegal. We penalize any customer at an out-of-range visit time with:

$$C_{\text{out}} = Z_6 \cdot C_{\text{bust}} \text{ where } 'C'_{\text{bust}} = \sum_{i=1}^{n} \text{Invalid}(v_i)$$

   where $\text{Invalid}(v_i)$ equals 1 when $v_i > n$ or $v_i < 1$, and 0 otherwise.

The ultimate cost function is the sum of the three terms:

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

This formulation enables the trade-off between route length minimization and valid/unique visit assignments. Certain terms of the cost function may be encodable directly in the quantum Hamiltonian and others (the duplicate and out-ofrange penalties e.g.) may be more efficiently relegated to a classical post-processing.

6.8 Qubit Allocation on Depots and Vehicles

The choice of the encoding of depots and vehicles into the qubits itself significantly influences the computational feasibility and the efficiency of the quantum circuit. In this paper, we test and contrast three different methods of allocating qubits:

1. One-Hot Encoding,

2. Separator Combinatorial Encoding, and

3. The Hybrid (Optimized) Encoding.

Each of these approaches has trade-offs between qubit efficiency, implementation complexity, and the extent to which it can be used to enforce constraints. **One-Hot Encoding**
It is a well-established method for classical optimization problems, and utilizes one qubit for each possible state or assignment. In the context of the VRP, each node (depot, vehicle, and customer) is encoded in a binary qubit-string in which only one bit representing the active assignment is 1, and all the others are set to 0. This ensures a simple problem representation, but requires many qubits proportional to the problem size given by the number of locations (depots, vehicles, customers). Qubit Computer
Let us consider a problem with d depots, each capable of handling v vehicles and n customers.
- Depot encoding would need qubits:

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

The qubits for encoding of the vehicle would need:

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

- The depot-to-vehicle assignment qubits would need:

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

- Qubits for vehicle-to-customer assignment would need:

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

So, the overall no. of qubits in One-Hot Encoding is:

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

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

Combinatorial Encoding using Separator
The separator combinatorial method is another choice as the mechanism to encode separators ("-1" in this case) is used to break up the groups or event transitions in the routing. In this way the number of qubits is scaled down,by exploiting combinatorial structures as opposed to describing all possible states directly. It becomes even more practical, especially when the problems possess a natural separation point, such as a depot and its associated vehicles or customers.

Example
For this same 3 depots and 5 vehicles setting, a separator combinational method could encode the depot vehicle assignment as a binary index mask, in which each bit denotes the assignment of a vehicle to a depot. To illustrate, the depot-to-vehicle encoding could employ '00011' to denote Depots 1 and 2 are allocated to Vehicle 1; '00110' to Depot 2; and '01000' to Depot 3. This way, the total amount of qubits is lower, since data can be encoded more compact, codifying groupings and state transitions.

Calculation of Qubit
For SCQE, here is the qubit count: - Depot to vehicle encoding:

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

- Vehicle-to-customer coding needs:

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

So, the overall number of qubits 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 intends to have the advantages of both the one-hot encoding and combinatorial encoding. Instead of representing explicitly all the states for each component (depot, vehicle, customer), a more efficient encoding is applied using a binary register for each customer or vehicle encoding the position or the assignment of a customer or a vehicle in a route. This results in a more compact encoding of the routing problem, facilitating decrease in the number of qubits without sacrificing capability of constraint implementation. Example
A case with 3 depots, 5 vehicles and 4 customers is shown in Figure 2. The hybrid indexing would dedicate:

- 2 qubits for the depot encoding (for 3 depots), 3 qubits for the vehicle encoding (5 vehicles),
- 4 qubits for the depot-vehicle assignment (15 feasible configurations for 3 depots and 5 vehicles), 5 qubits per vehicle-to-customer assignment (20 possibilities for 5 vehicles and 4 customers).

Qudit Computation
Therefore, the overall qubit number of 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 features of the three (qubit) allocation schemes are presented in table below:

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

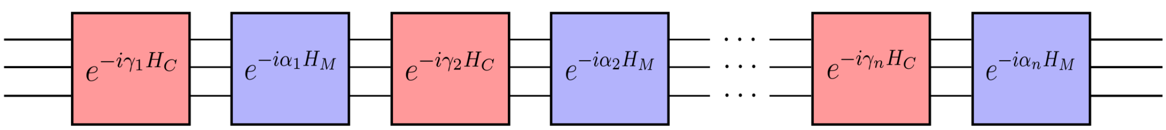
In summary, although one-hot encoding is simple to understand and carry out, its performance is poor scaling with problem size, thus requiring a large number of qubits. The combinatorial method for exploiting separators reduces the number of used qubits by stating the problem in a more compressed form, and the price to pay for that choice is a less simple way to take into account the constraint. The hybrid/optimised encoding combines elements of two encoding strategies, being further studied due to its good trade-off between qubit efficiency and constraint handling, and is a potential candidate for larger VRP problems in the quantum domain.

6.9 Quantum Circuits and Hamiltonians

One way to do this is to map this cost function, $C(x)$, to a quantum Hamiltonian, H_C by replacing the binary variables x_{ij} by Pauli-Z operators. The quantum circuit repeatedly applies the cost Hamiltonian and a mixing Hamiltonian, typically implemented as a rotation gate. The number of layers are determined by the depth p. The unitaries used are:

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

The parameters γ (gamma) and β (beta) in the QAOA circuit describe evolution under the cost and mixing Hamiltonians, respectively. We could begin to initialize the values of these parameters to a random state, before optimizing them using a classical optimizer. One will have something of this form, in an n-depth Hamiltonian and here's how the circuit will look.



6.10 Memory Problem

We ran into a severe memory issue when trying to simulate the QAOA circuit even for a small VRP instance (5 customers, 1 depot). The PennyLane simulator, as it stands, tries to save these 2^k complex amplitudes of the full quantum state vector. For n = 5, k = 36:

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

Thus, with $(n+1)^2$ qubits, an exact 1 Terabyte of memory would have been used. This is obviously absurd. Therefore, we have to look for a different k such that $2k > n!$. So we can use an other encoding method like logperm on separator comb encoding.

6.11 Handling Constraints

In this work we handle real-world constraints of Vehicle Routing Problem (VRP) (e.g., due to traffic congestion, weather conditions and road accidents, etc.) using the classic preprocessing approaches. Such restrictions are not explicitly imposed in the quantum model. Rather they are employed to adapt the graph structure that is fed to quantum optimization algorithm. For instance, regions of high traffic or high accident rates are assigned greater edge weights to indicate generally longer travel times or additional hazards. This hybrid scheme offers a complexity reduction of the quantum circuit and maintains the consideration for a realistic system. Due to the constraints of existing quantum hardware, for example the small number of qubits and short coherence time, it is more efficient to deal with dynamic real-world variables through classical computations before encoding them into quantum states. After construction of the graph along with these constraint-aware edge costs, the graph is translated into a representation in QUBO or Hamiltonian types compatible with quantum applications such as QAOA. Although the direct focus is placed on quantum optimization, various steps (such as data preparation, constraint embedding, and solution validation) still reside in classical domain. As a consequence, a hybrid quantum-classical approach is adopted, with the classical part J..z providing a realistic representation of the problem and the quantum part focussing on optimal or near-optimal solutions.

6.12 Operation Number Comparison (Classical Vs. Quantum)

- **Classical** Classical algorithms for the VRP (e.g., the exhaustive search algorithm) must search through all the possible permutations, yielding an order of $(n+1)!$ operations for n clients.
- **Quantum**: QAOA approximately solves the problem with possibly fewer operations, however, the level of reduction is problem-specific and is due to a specific problem instance and QAOA setting. Complexity is increased with an increase in depth p. The circuit complexity and number of qubits required will also be higher. The total number of quantum gates is approximated by $s \cdot p \cdot (n+1)^2$ operations (much less than $(n+1)!$), with controlled iterations s and circuit depth p.

6.13 Other Remarks: The v Vehicles and d Depots Case

For depots and vehicles the above is generalized as:

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

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

- **Classical**: For v vehicles and d depots, the complexity of the classical problem is much higher, as we should also take all the possible assignments of customers to vehicles and depots into account
- **QAOA**: QAOA can in principle be capable of dealing with this additional complexity, but the number of qubits required and the complexity of the circuit will also increase. Quantum state would have to be used to encode the assignments of vehicles, and depot choices.

7 Other Relevant Research on Quantum Optimization

7.1 Quantum Allocation with Efficiency

Balu *et al.* [22] has presented a new way to solve the TSP with a single qubit by encoding a cost Hamiltonian in terms of the amplitude parameters. This minimal approach opens the door for quantum optimization on otherwise constrained hardware and indicates TAC's capability to represent compact encoding of complex NP problems. Good qubit placement is crucial while writing a real quantum program. Baek *et al.* [23] propose type-based allocation for first order QPL (quantum programming language). Shi *et al.* [24] deal with distributed quantum computing and Li *et al.* [25] suggest an exact algorithm specifically designed for NISQ computers. These allocation methods are used to supplement quantum optimization by utilizing resources more efficiently.

7.2 Quantum Algorithms for TSP and VRP

Quantum annealing is utilized by a number of works for combinatorial problems. Wang *et al.* [26] employ QUBO formulations and Graph Neural Networks to solve the TSP. Kochenberger et al. (2010)30 and further work by the same authors (2011)31 provides a comprehensive review on MIP methods for the TSP. [27] define an extensive Ising representation of NP problems for annealing architectures. Previous research by D-Wave scientists including Yu et al. [28] also propose the hybrid VN model to solve capacitated VRP through annealers. Some quantum algorithms are designed to solve TSP directly. The recent results of Surya *et al.* proposed an efficient quantum algorithm with good performance bounds. Another work from Younes[22] generalizes the setup to a simplified setting, where reduced quantum system methods to solve TSP are demonstrated.

7.3 Optimizing Quantum Circuits

Optimising the computational complexity of quantum circuits is central for scalability. Liu *et al.* [29] introduce an optimization-based reduction method that is quantized to smaller gate number and shorter circuit depth for increasing success probability on Noisy Intermediate-Scale Quantum (NISQ) devices.

8 Methodology

8.1 Algorithms for Quantum Computing

- Formulate: (We will further elaborate on this) Represent the VRP in a graph representation that is compatible for the QAOA, Quantum Walk algorithms, and VQE. Theoretical Calculations: Study the VRP domain mathematics andize complexity to understand how quantum algorithms could perform better than classical ones on VRPT. This involves recognition of where quantum optimization is faster or where it provides better approximations.

- Implementation: Construct quantum circuits and simulate them (e.g. with PennyLane or Qiskit, or any other compatible library)

8.2 Method of benchmarking

- Solution with Classical Comparison: Apply introduced solution with classical estimation to provide a baseline of solutions.
- Performance criteria: Quantitative comparison based on solution quality, computational efforts and scalability between quantum and classical methods.

8.3 Datasets

- Data Acquisition: Use real life datasets from public datasets[2, 4, 5, 6, 7, 8, 9].
- Data Processing: Organize raw data into a graph representation which can be used by quantum algorithms.

9 Expected Results and Contribution

- Construction of quantum-inspired approximation algorithms: Adding new angles to address VRP by leveraging QAOA.
- Comparative Analysis: Providing contrast to current practical limitations and advantages with respect to quantum (in this instance, QAOA) and classical approaches to VRP.

10 Limitations and Challenges

Quantum Hardware Limitations: Quantum cloud is cost-prohibitive. Use a small graph to run on real quantum computer and use a simulator (or a pseudo-theoritical calculations) to simulate a larger graph.

11 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.

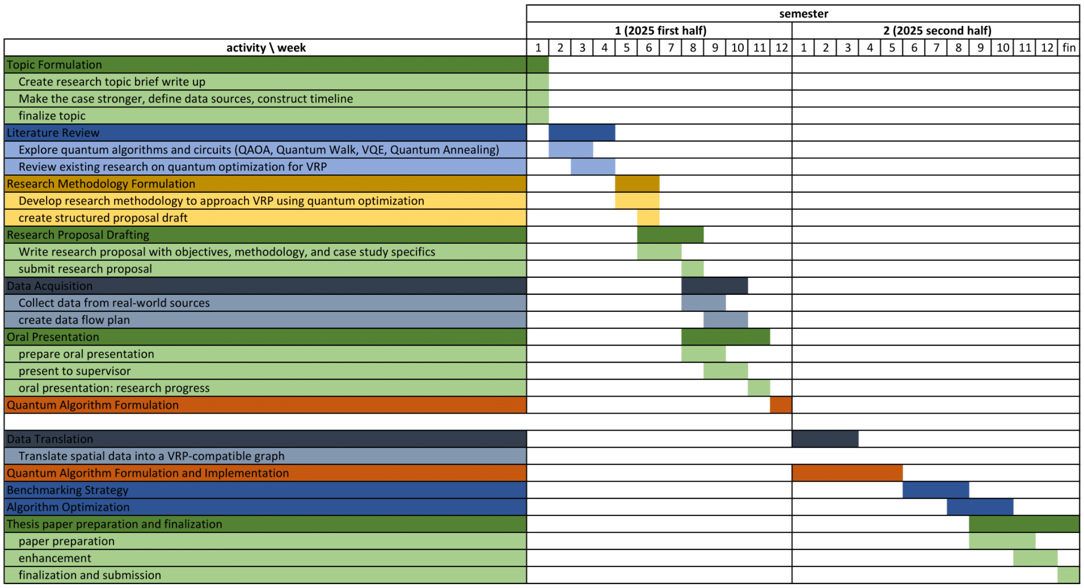
Second Semester (50 Points):

- 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] 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.

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

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

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

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

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

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

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

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

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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