Benchmarking Classical Methods against Quantum Methods to Solve the Vehicle Routing Problem

Ben Cleveland
University of Cape Town
Cape Town, South Africa

Tayla Rogers
University of Cape Town
Cape Town, South Africa

1 INTRODUCTION

The Vehicle Routing Problem (VRP) [10] is a well-known combinatorial optimisation problem. It focuses on finding the most efficient routes for a fleet of vehicles to deliver goods or services to a set of customers while minimising costs. The problem is mathematically modelled, as illustrated in Figure 1, with a focus on minimising the objective function outlined in Equation 1. Rich VRPs (RVRPs) expand the use of VRP to various real-world challenges by introducing constraints such as congestion and delays [8]. While traditionally applied to road transportation, VRP is used in diverse domains such as in-home health care and refuse collection [25]. Efficient resolution of VRP is crucial for businesses and individuals reliant on timely transport services, especially for small and medium-sized enterprises seeking cost-effective transportation solutions [8].

Despite its significance, the VRP faces challenges due to its classification as an $\mathcal{NP}$-hard problem [24]. The challenge is evident in the exponential growth in problem size with an increase in input size, making the VRP largely impractical to solve for large instances.

$$VRP(n,k) = min_{\{x_{ij}\}_{i->j} \in \{0,1\}} \sum_{i->j} w_{ij} x_{ij} [7] \tag{1}$$

Interest in using quantum principles, most notably the principles of entanglement and superposition, for solving combinatorial optimisation problems such as the VRP has been sparked by recent advances in quantum computing. Quantum algorithms, implemented on IBM's Noisy Intermediate-Scale Quantum (NISQ) devices, explore multiple potential routes concurrently across a large solution space using these principles. The VRP's representation as an energy function, depicted in Figure 2, facilitates the creation of a Quadratic Unconstrained Binary Optimisation (QUBO) formulation, as shown in Figure 3. This formulation is then mapped to an Ising Hamiltonian, depicted in Figure 4, enabling the ability to use quantum to solve the VRP. This Ising formulation of the problem, shown in Equation 2, describes the energy associated with a specific qubit configuration in a quantum circuit.

$$H_{\text{Ising}} = -\sum_{i} \sum_{j<i} I_{ij} s_i s_j + \sum_{i} h_i s_i + d [7] \tag{2}$$

Insight into the current capabilities of quantum algorithms, constrained by available qubits [4, 23], is provided by benchmarking quantum algorithms. The viability of quantum algorithms is gauged by comparisons against common classical benchmarks.

This study extends prior work on quantum VRP algorithms [5, 7], offering a comprehensive comparison with classical algorithms across various metrics, confirming the current potential of quantum approaches. The analysis includes solution quality, time complexity, and feasibility over a reduced meta-heuristic dataset.

2 RELATED WORK

One of the major pieces of literature on benchmarking quantum algorithms against classical algorithms was provided by Khumalo et al. [15], who conducted an in-depth analysis of VQE and QAOA against Branch and Bound (B&B) and Simulated Annealing (SA). The solution quality, feasibility, and computational time of these algorithms in addressing the Travelling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP) were compared in the paper. The experiments showed the significant out-performance of classical methods. The quantum methods were found to be limited in solving larger problems due to the restricted number of qubits available, impacting circuit depth and operation capacity. While informative, the paper's focused problems are simpler than the VRP, which is a generalisation of the TSP as noted by Azad et al. [7]. Generalised problems present a more complex context, underscoring the importance of extending this research to understand quantum viability.

Further research was undertaken by Alsaiyari and Felemban [5], who investigated VQE and QAOA compared to IBM's CPLEX classical optimiser. The CPLEX optimiser uses exact algorithms and provides high-quality solutions for comparison with quantum solutions. The analysis provided showed the abilities of the QAOA algorithm to be comparable to the classical optimiser in terms of solution quality. However, this paper only compared the algorithms on solution quality without considering other important metrics such as computational time and feasibility. Additionally, the comparison was limited to the CPLEX optimiser, excluding other classical techniques such as meta-heuristics, which could enhance the analysis of quantum algorithms.

Azad et al. [7] took a similar approach to benchmarking, with a more in-depth view of QAOA's performance. Similarly, the paper lacked a comprehensive analysis of multiple metrics and multiple algorithms, only comparing QAOA against the solution quality of the CPLEX classical optimiser.

3 PROBLEM STATEMENT, RESEARCH QUESTIONS AND AIMS

The overall purpose of our investigation is encapsulated in the following problem statement: How do classical and quantum methods compare, in terms of computational time and solution quality, in solving instances of the VRP?

The goal of our research is to compare classical and quantum methods in solving varied instances of the VRP. The specific aspects of this problem statement that will be investigated are represented by the following research aims and sub-questions:

(1) How do variational quantum methods perform concerning solution quality and computational time in comparison

with metaheuristic and exact classical methods on different problem instance sizes of the VRP?
(2) Is there a quantum advantage over classical methods for solving the VRP, and if so under what conditions?
(3) For what range of problem instance sizes can the latest publicly available IBM quantum NISQ devices produce solutions that are of acceptable quality, defined as solutions within 95% of the globally optimal solution?

4 PROCEDURES AND METHODS

4.1 Overview

To carry out this research, the dataset to be used will first be reduced. The algorithms will then be implemented in code, incorporating any sub-strategies and specialised parameters necessary. The performance of the algorithms will be verified, and they will then be run on our dataset on their respective devices. The results will then be analysed and compared to formulate the results of the research.

4.2 Experimental Settings

Classical and simulator experiments aim to be run on UCT's High-Performance Cluster (HPC), and otherwise our local devices. Development of VQE and QAOA algorithms for solving the VRP will be use Qiskit, an open-source Python-based SDK. Implementation of both classical algorithms will also be carried out using Python to ensure a fair comparison. Experiments on publicly available quantum devices will be conducted using IBM's Quantum Experience Cloud.

The data used in this research will be adapted from 240 instances of the capacitated vehicle routing problem (CVRP) specified in [18], with the original dataset residing at [17]. CVRP-specific information such as capacity and demand constraints will be removed to form general VRP problem instances. The number of vehicles and destinations will be reduced from the original dataset to create instances of an appropriate size for the Variational Quantum Algorithms (VQAs) and B&B under testing. The resulting dataset will be relatively small, ranging approximately from problems of 2 cities and 1 vehicle to 7 cities and 3 vehicles.

4.3 Classical Algorithm Implementations

The diverse benefits of classical algorithms will be shown by the selection of both an exact and a metaheuristic approach. These methods will provide a benchmark for evaluating and comparing solution quality and computational efficiency against quantum approaches.

An initial solution is required for both algorithms to begin calculations. The savings algorithm, proposed by Clarke and Wright [9], is a well-known heuristic commonly used for this purpose due to its speed in providing a suitable solution, although without guaranteeing optimality.

While well-defined in theory, these algorithms require a specific implementation to suit the VRP. The adjustment of the solution space definition, the sub-strategies used, and the parameter values are necessary to accommodate the VRP.

4.3.1 Branch and Bound. B&B, widely used in exact algorithms, has proven effective in solving combinatorial optimisation problems [24]. It explores the solution space using a tree-like structure, where branches represent potential solutions and sub-optimal branches are pruned using bounding techniques.

To implement B&B, a branching strategy divides the problem into sub-problems at each iteration. A wide branching strategy [21] has been chosen to prevent the creation of excessively large search trees [20] inherent to the VRP. Non-memory-based dominance rules [12] will be applied as a bounding strategy to avoid duplication of sub-problems. Selection rules, employing a Best-First Search (BFS) strategy [11], prioritise exploring solutions with the smallest lower bound first, reducing computational time.

The algorithm continues until all candidate sub-problems have been exhausted or it has run for over 8 hours, indicating that it cannot be solved in a reasonable amount of time. Algorithm 1, adapted from Morrison et al. [20], has outlined these strategies, providing a framework for addressing the VRP using the B&B approach. An explanation of the variables used can be found in Table 1.

4.3.2 Simulated Annealing. SA is a guided-random local search meta-heuristic algorithm based on the process of metallurgical annealing [13]. An initial solution is evolved towards optimality through a balance of the exploitation of good solutions and the exploration of the search space. A neighbouring solution is generated and accepted if it is better than the current solution. Worse solutions are accepted according to some probability derived from the Boltzmann distribution and the current value of the temperature parameter [6]. Entrapment in a local optimum is prevented by conditionally accepting worse solutions, and the probability of this happening is decreased over time[6]. The parameters and inputs required for SA consist of the initial solution, the initial temperature parameter, the cooling schedule value, a stopping criterion, an equilibrium condition and an objective function [6]. The initial temperature and cooling schedule must both be balanced to not get stuck in local optima as well as to not waste large amounts of computation and must be chosen through experimentation [14]. The objective function is responsible for minimising the total cumulative route lengths. The equilibrium condition is defined as a value that denotes the number of iterations to be executed at each level of the temperature parameter [14]. A stopping condition for this algorithm is defined as either a predefined number of cooling cycle iterations or the predefined final temperature being reached [14].

The pseudocode for SA, adapted from [6], is shown in Algorithm 2. An explanation of the variables used can be found in Table 2.

4.4 Quantum Algorithm Implementations

The current potential of hybrid-quantum algorithms will be demonstrated by the two selected VQAs. Hybrid algorithms are currently considered the best use of quantum computing, given the susceptibility of current NISQ devices to noise and errors, and the limited qubit availability.

Algorithm formulation and usage are enabled by IBM's Qiskit framework, but customising them specifically for the VRP is necessary [1]. The design and coding of these algorithms will also need to be aligned with theoretical best practices to make them applicable to solving this problem.

These algorithms will then be tested with various datasets and problem sizes using IBM's quantum simulator, which will be operated locally, as well as IBM's quantum devices.

4.4.1 Variational Quantum Eigensolver. The Variational quantum eigensolver (VQE) is categorised as a hybrid quantum-classical algorithm, with both quantum computing and classical computing technology being utilised to solve for the eigenvalue of a Hamiltonian matrix [19]. The solution for the minimum eigenvalue, corresponding to an eigenstate equivalent to the "ground state" or lowest energy state of the Hamiltonian, can be found, representing the optimal solution to an optimisation problem [3]. An ansatz or "best guess" is used by the algorithm as a parameterised trial solution [2]. The quantum computer evaluates the value of the energy of the ansatz [15], and an optimiser that is run on a classical machine is used to adjust the parameters towards optimality [3]. This quantum-classical iterative loop is managed by a classical machine and is continued until a termination criterion is met.

Optimisation of our chosen ansatz from Qiskit and our Hamiltonian is necessary to conform to the target machine's instruction set architecture (ISA) using code from a Qiskit library [2]. A cost function and a function to return updates of the optimisation process state (a callback function) are to be created using Qiskit library code. The cost function will be optimised by our chosen classical optimiser[2]. The callback function's output can be used to view our solution. The pseudocode for VQE, adapted from [15], is shown in Algorithm 3. An explanation of the variables used can be found in Table 3.

4.4.2 Quantum Approximate Optimisation Algorithm. QAOA, developed for combinatorial optimization problems, has been extensively researched as a method for approximating solutions [5]. At the core of this algorithm are the Mixer Hamiltonian, H_M, and the Cost Hamiltonian, H_C. The initial quantum state is prepared by the quantum operator represented by H_M, and the objective function is encoded by H_C. These Hamiltonians are precomputed before QAOA is invoked to navigate the solution space.

The chosen ansatz is modelled as a parameterised quantum circuit used to prepare the initial quantum state [7]. The circuit's parameters are iteratively fine-tuned by a classical optimiser, with the aim of minimising the expectation value of H_C.

Pseudocode for QAOA seen in Algorithm 4, adapted from Khumalo et al. [15], offers a practical approach for the utilisation of quantum computing in combinatorial optimisation problems, including the VRP. An explanation of the variables used can be found in Table 4.

4.5 Metrics

In the comparison of algorithms for solving the VRP, the minimisation of vehicle trip distances will be the primary metric to be considered, referred to in this research as solution quality. CPU time, an important indicator of the feasibility of using a particular algorithm, has been considered as another metric. Algorithms whose running times have reached a predefined timeout value will be stopped. Algorithms will be run 30 times per instance, noting average, best and worst case performance per problem instance. Solution consistency, represented by the standard deviation or the variance from the mean performance, will be assessed. The metric of scalability, indicative of the size ranges over which algorithms can effectively operate, will be considered. The feasibility percentage, represented as the percentage of solutions for a given problem instance that are feasible, will be assessed for all methods. Two success rate metrics will be assessed, namely the percentage of solutions produced by an algorithm that are within 95% and 99% of the global optimal solution for a problem instance.

5 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

The authors declare that they have no conflict of interest. The research will be conducted on an established dataset - there is no ethical clearance necessary and no specimens used. All code will be made publicly available for readers to view.

6 ANTICIPATED OUTCOMES

6.1 Research Outcomes

The main outcome is a comparative assessment of VQAs against two well-developed classical algorithms, aiming to offer a benchmark for fellow researchers. Additionally, this study seeks to enhance understanding of the abilities of current NISQ devices and quantum algorithms. By evaluating VQAs alongside classical counterparts, we aim to demonstrate the applicability of VQAs on existing NISQ devices for solving the VRP. Insights drawn from this benchmarking research are expected to extend to other combinatorial optimisation problems.

It is anticipated that the classical algorithms will outperform quantum methods, particularly at larger problem scales. It is expected that better solution quality and computational time will be achieved by SA and B&B when compared to the VQAs.

The main outcome will be a comparative assessment of VQAs against two well-developed classical algorithms, aiming to provide a benchmark for future research. Additionally, this study aims to enhance understanding of the capabilities of current NISQ devices and quantum algorithms. By evaluating VQAs alongside classical counterparts, we will aim to demonstrate the applicability of VQAs on existing NISQ devices for solving the VRP. Insights drawn from this benchmarking research are expected to extend to other combinatorial optimization problems.

It has been anticipated that the classical algorithms will outperform quantum methods, particularly at larger problem scales. Better solution quality and computational time are expected to be achieved by SA and B&B when compared to the VQAs.

6.2 Key Success Factors

6.2.1 Algorithm Validation. Results from the most recent established papers on a simple variation of the VRP will be used to compare the classical algorithms, B&B and SA, in order to validate their implementation [16, 22]. Due to advancements in algorithmic strategies and technology since the publication of those papers, direct comparison will not yield exact results. Furthermore, the specific sub-strategies used within the algorithms are not stated in the papers, impacting our ability to precisely replicate the results through our chosen strategies. Therefore, this comparison simply aims to ensure that the performance reported in prior research is matched or exceeded by our algorithms, serving as a reliable benchmark.

Quantum algorithms will be evaluated against findings from previous studies on quantum VRP research using a local quantum simulator [5, 7]. The verification of performance on quantum devices will not be able to be done due to the long waiting queues for quantum devices. Therefore similarities in the performance of our simulator results will have to be sought in order to refine our algorithms to a representative performance before placing them in quantum queues.

Additional verification of the output of the algorithms will be done by taking a random sample of the outputted routes and subjecting them to manual checking to confirm that the algorithm produces valid results.

6.2.2 Research Success. For the research to succeed, the experiments must address our initial research questions and contribute to existing knowledge, providing readers with a better understanding of the applicability of quantum computers in solving complex problems such as the VRP. This entails filling the gap in the literature regarding the comprehensive comparison of quantum and classical algorithms in solving the VRP across various metrics and comparing them to different classes of classical algorithms.

7 PROJECT PLAN

7.1 Risks

The risks and mitigation strategies per risk appear in the following list as risk and risk mitigation strategy item pairs.

- Risk 1
  - Risk: It may be found that IBM's Qiskit queue is very long, and a wait of 2-3 months for quantum algorithm results may be required.
  - Mitigation strategy: The quantum algorithm code is to be placed in the queue well in advance (by mid-May), making it feasible to wait even 3 months for results. In the worst-case scenario of a never-ending wait in a queue, IBM's quantum simulator software may be utilised for experiments.
- Risk 2
  - Risk: Failing to adhere to the constrained timeline given that project tasks are of various types, with varying lengths of completion times, and at times lack strict order of completion.
  - Mitigation strategy: Good project planning is to be used, utilising mini-deadlines to maintain progress on track.
- Risk 3
  - Risk: Lack of uniformity in datasets, as the different algorithms being tested are usually tested independently under specific circumstances (such as data set sizes, and specific constrained versions of the VRP).
  - Mitigation strategy: A publicly available dataset is to be adapted to suit the constraints of the benchmarked algorithms, and this adapted dataset is to be made publicly available for the purpose of future research.
- Risk 4
  - Risk: Experimental results not being representative of the truth due to stochastic elements or random variation.
  - Mitigation strategy: The algorithms are to be run on a large set of instances, with each algorithm run 30 times on each instance, and a statistical analysis of the results is to be conducted.
- Risk 5
  - Risk: Ineffective parameter optimisation. If for example, the value of the temperature parameter in SA is not high enough, the algorithm will fail to find globally optimal solutions.
  - Mitigation strategy: An optimisation algorithm is to be used for tuning parameters.
- Risk 6
  - Risk: One of the two members of the research team drops out of the course.
  - Mitigation strategy: The research project consists of two distinct and independent pieces of work.

7.2 Timeline

Timeline detailed in Gantt chart seen in Figure 5.

7.3 Resources Required

The resources required for this research can be divided into three distinct categories:

7.3.1 Hardware. The hardware resources necessary include IBM's quantum computers available through Qiskit and UCT's HPC, or our personal devices.

7.3.2 Software. The software resources necessary include IBM's Qiskit software to code and run the quantum algorithms, the python programming language to code the classical algorithms, a python environment and an operating system in which to run the classical algorithms.

7.3.3 Personnel. The personnel resources necessary include the two computer science honors students and the project supervisor.

7.4 Deliverables

- Final project proposal
- Qiskit code formulations for the VQAs: QAOA and VQE.
- Python code formulations for the classical algorithms: SA and B&B.
- The final adapted dataset.
- Experimental results for the quantum algorithms run on instances of the VRP on IBM's NISQ devices.
- Experimental results for the quantum algorithms run on instances of the VRP on IBM's quantum simulator.
- Experimental results for the classical algorithms run on instances of the VRP.
- Progress report presentation.
- Comparative analysis of the results per algorithm per problem instance.

7.5 Milestones

All milestones can be seen in and are named in accordance with the Gantt chart (Figure 5) in the appendix.

- Final project proposal - 30th April 2024
- Code development completed - 24th July 2024
- Running of experiments completed - 3rd August 2024
- Experimental results analysis completed - 14th August 2024
- Writing of draft for final report completed - 23rd August 2024
- Writing of final report completed - 30th August 2024

7.6 Work Allocation

The formulation and coding of the B&B and QAOA, as well as the interpretation of results from these algorithms and the final project paper, have been assigned to Tayla Rogers. The formulation and coding of the SA algorithm and VQE algorithm, along with the interpretation of results from these algorithms and the final project paper, have been assigned to Ben Cleveland.

REFERENCES

[1] [n. d.]. IBM Quantum. https://www.ibm.com/quantum/qiskit
[2] [n. d.]. Variational quantum eigensolver | IBM Quantum Learning. https://learning.quantum.ibm.com/tutorial/variational-quantum-eigensolver
[3] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini, Bryce Fuller, Julien Gacon, Constantin Gonciulea, Sander Gribling, Swati Gupta, Stuart Hadfield, Raoul Heese, Gerhard Kircher, Thomas Kleinert, Thorsten Koch, Georgios Korpas, Steve Lenk, Jakub Marecek, Vanio Markov, Guglielmo Mazzola, Stefano Mensa, Naeimeh Mohseni, Giacomo Nannicini, Corey O'Meara, Elena Peña Tapia, Sebastian Pokutta, Manuel Proissl, Patrick Rebentrost, Emre Sahin, Benjamin C. B. Symons, Sabine Tornow, Victor Valls, Stefan Woerner, Mira L. Wolf-Bauwens, Jon Yard, Sheir Yarkoni, Dirk Zechiel, Sergiy Zhuk, and Christa Zoufal. 2023. Quantum Optimization: Potential, Challenges, and the Path Forward. https://doi.org/10.48550/arXiv.2312.02279
[4] Akshay Ajagekar and Fengqi You. 2019. Quantum computing for energy systems optimization: Challenges and opportunities. *Energy* 179 (07 2019), 76–89. https://doi.org/10.1016/j.energy.2019.04.186
[5] Muhammad Alsaiyari and Muhamad Felemban. 2023. Variational Quantum Algorithms for Solving Vehicle Routing Problem. , 4 pages. https://doi.org/10.1109/ICSCA57840.2023.10087522
[6] Ali Asghar, Fataneh Alavipour, Shahaboddin Shamshirbnd, and Valentina E Balas. 2016. A Novel Meta-Heuristic Combinatory Method for Solving Capacitated Vehicle Location-Routing Problem with Hard Time Windows. *Advances in intelligent systems and computing* (11 2016), 707–728. https://doi.org/10.1007/978-3-319-38789-5_77
[7] Utkarsh Azad, Bikash K. Behera, Emad A. Ahmed, Prasanta K. Panigrahi, and Ahmed Farouk. 2022. Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm. *IEEE Transactions on Intelligent Transportation Systems* 24 (2022), 1–10. https://doi.org/10.1109/TITS.2022.3172241
[8] Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A. Juan. 2014. Rich Vehicle Routing Problem. *Comput. Surveys* 47 (12 2014), 1–28. https://doi.org/10.1145/2666003
[9] G. Clarke and J. W. Wright. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* 12 (08 1964), 568–581. https://doi.org/10.1287/opre.12.4.568
[10] G. B. Dantzig and J. H. Ramser. 1959. The Truck Dispatching Problem. *Management Science* 6 (10 1959), 80–91. https://doi.org/10.1287/mnsc.6.1.80
[11] Rina Dechter and Judea Pearl. 1985. Generalized best-first search strategies and the optimality of A*. *J. ACM* 32 (07 1985), 505–536. https://doi.org/10.1145/3828.3830
[12] Matteo Fischetti and Domenico Salvagnin. 2010. Pruning Moves. *INFORMS journal on computing* 22 (02 2010), 108–119. https://doi.org/10.1287/ijoc.1090.0329
[13] Michel Gendreau, Jean-Yves Potvin, Olli Bräumlaysy, Geir Hasle, and Arne Løkketangen. 2008. Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. *Operations research, computer science. Interface series* (07 2008), 143–169. https://doi.org/10.1007/978-0-387-77778-8_7
[14] Darrall Henderson, Sheldon H. Jacobson, and Alan W. Johnson. 2020. The Theory and Practice of Simulated Annealing. *Handbook of Metaheuristics* (2020), 287–319. https://doi.org/10.1007/0-306-48056-5_10
[15] Maxine T. Khumalo, Hazel A. Chieza, Krupa Prag, and Matthew Woolway. 2022. An investigation of IBM quantum computing device performance on combinatorial optimisation problems. *Neural Computing and Applications* (06 2022). https://doi.org/10.1007/s00521-022-07438-4
[16] G. Laporte and Y. Nobert. 1983. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum* 5 (06 1983), 77–85. https://doi.org/10.1007/bf01720015
[17] Adam Letchford and Juan-Jose Salazar-Gonzalez. 2018. CVRP instances. *research-data.ull.es* 1 (06 2018). https://doi.org/10.17632/kcc52cw4zs.1
[18] Adam N. Letchford and Juan-José Salazar-González. 2019. The Capacitated Vehicle Routing Problem: Stronger bounds in pseudo-polynomial time. *European Journal of Operational Research* 272 (01 2019), 24–31. https://doi.org/10.1016/j.ejor.2018.06.002
[19] Nishikanta Mohanty, Bikash K Behera, and Christopher Ferrie. 2023. Analysis of the Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in the Presence of Noisy Channels. *IEEE transactions on quantum engineering* 4 (01 2023), 1–14. https://doi.org/10.1109/tqe.2023.3303989
[20] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19 (02 2016), 79–102. https://doi.org/10.1016/j.disopt.2016.01.005
[21] David R Morrison, Jason J Sauppe, Edward C Sewell, and Sheldon H Jacobson. 2014. A Wide Branching Strategy for the Graph Coloring Problem. *INFORMS journal on computing* 26 (11 2014), 704–717. https://doi.org/10.1287/ijoc.2014.0593
[22] Ibrahim Hassan Osman. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41 (12 1993), 421–451. https://doi.org/10.1007/bf02023004
[23] Karthik K Srinivasan, Saipriya Satyajit, Bikash K Behera, and Prasanta K Panigrahi. 2018. Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience. *arXiv (Cornell University)* (05 2018). https://doi.org/10.48550/arxiv.1805.10928
[24] Paolo Toth and Daniele Vigo (Eds.). 2002. *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718515
[25] Paolo Toth and Daniele Vigo. 2015. *Vehicle routing : problems, methods, and applications.* Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

APPENDIX

$$\min \quad \sum_{i\to j} w_{ij}x_{ij} \tag{3}$$

$$\text{s.t.} \quad \sum_{i\in source[i]} x_{ij} = 1 \qquad \forall i \in \{1,...,n-1\} \tag{4}$$

$$\sum_{j\in target[i]} x_{ji} = 1 \qquad \forall i \in \{1,...,n-1\} \tag{5}$$

$$\sum_{j\in source[0]} x_{0j} = k \tag{6}$$

$$\sum_{j\in target[0]} x_{j0} = k \tag{7}$$

Figure 1: Mathematical model of the Vehicle Routing Problem

[7]

$$H_{VRP} = H_A + H_B + H_C + H_D + H_E \tag{8}$$

$$H_A = \sum_{i\to j} w_{ij}x_{ij} \tag{9}$$

$$H_B = A\sum_{i\in 1,\dots,n-1}\left(1 - \sum_{j\in source[i]} x_{ij}\right)^2 \tag{10}$$

$$H_C = A\sum_{i\in 1,\dots,n-1}\left(1 - \sum_{j\in target[i]} x_{ji}\right)^2 \tag{11}$$

$$H_D = A\left(k - \sum_{j\in source[0]} x_{0j}\right)^2 \tag{12}$$

$$H_E = A\left(k - \sum_{j\in target[0]} x_{j0}\right)^2 \tag{13}$$

Figure 2: Energy Function of the Vehicle Routing Problem

[7]

$$\min \quad \vec{x}^T Q\vec{x} + \vec{g}^T\vec{x} + c \tag{14}$$

$$\vec{x} = [x_{(0,1)}, x_{(0,2)}, \dots x_{(1,0)}, x_{(1,2)}, \dots x_{(n-1,n-2)}]^T \tag{15}$$

$$\sum_{j\in source[i]} x_{ij} = \vec{z}^T_{S[i]}\vec{x} \tag{16}$$

$$\sum_{j\in target[i]} x_{ji} = \vec{z}^T_{T[i]}\vec{x} \tag{17}$$

$$Q = A\Big([\vec{z}_{T[1]}, \vec{z}_{T[2]}, \dots, \vec{z}_{T[n-1]}, \vec{z}_{T[0]}, \vec{z}_{T[2]}, \dots, \vec{z}_{T[n-2]}]^T, + [[\vec{z}_{S[0]}]^{\times(n-1)}[\vec{z}_{S[1]}]^{\times(n-1)} \dots [\vec{z}_{S[n-1]}]^{\times(n-1)}]\Big) \tag{18}$$

$$\vec{g} = \vec{w} - 2A(\vec{J} + \vec{K}) - 2Ak(\vec{z}_{S[0]} + \vec{z}_{T[0]}) \tag{19}$$

$$c = 2A(n-1) + 2Ak^2 \tag{20}$$

$$\vec{J} = n\times n \text{ matrix with first } n-1 \text{ elements } = 0 \text{ and next } (n-1)^2 \text{ elements } = 1 \tag{21}$$

$$\vec{K} = \vec{x} \text{ with } x_{ij} = 1 \text{ if } j \neq 0, \forall i \in \{0,\dots,n-1\}, \text{ else } 0 \tag{22}$$

$$\vec{w} = \text{ weight vector} \tag{23}$$

Figure 3: QUBO Formulation of the Vehicle Routing Problem

[7]

$$H_{\text{Ising}} = -\sum_i\sum_{j<i} I_{ij}s_is_j + \sum_i h_is_i + d \tag{24}$$

$$x_{ij} = \frac{s_{ij}+1}{2} \qquad s_{ij} \in \{-1, 1\} \tag{25}$$

$$I_{ij} = -\frac{Q_{ij}}{4} \qquad \forall i < j, \quad I_{ii} = 0 \quad \forall i \tag{26}$$

$$h_i = \frac{g_i}{2} + \sum_j \frac{Q_{ij}}{4} + \sum_j \frac{Q_{ji}}{4} \tag{27}$$

$$d = c + \sum_i \frac{g_i}{2} + \sum_i \frac{Q_{ii}}{4} + \sum_i\sum_j \frac{Q_{ij}}{4} \tag{28}$$

Figure 4: Ising Formulation of the Vehicle Routing Problem

[7]

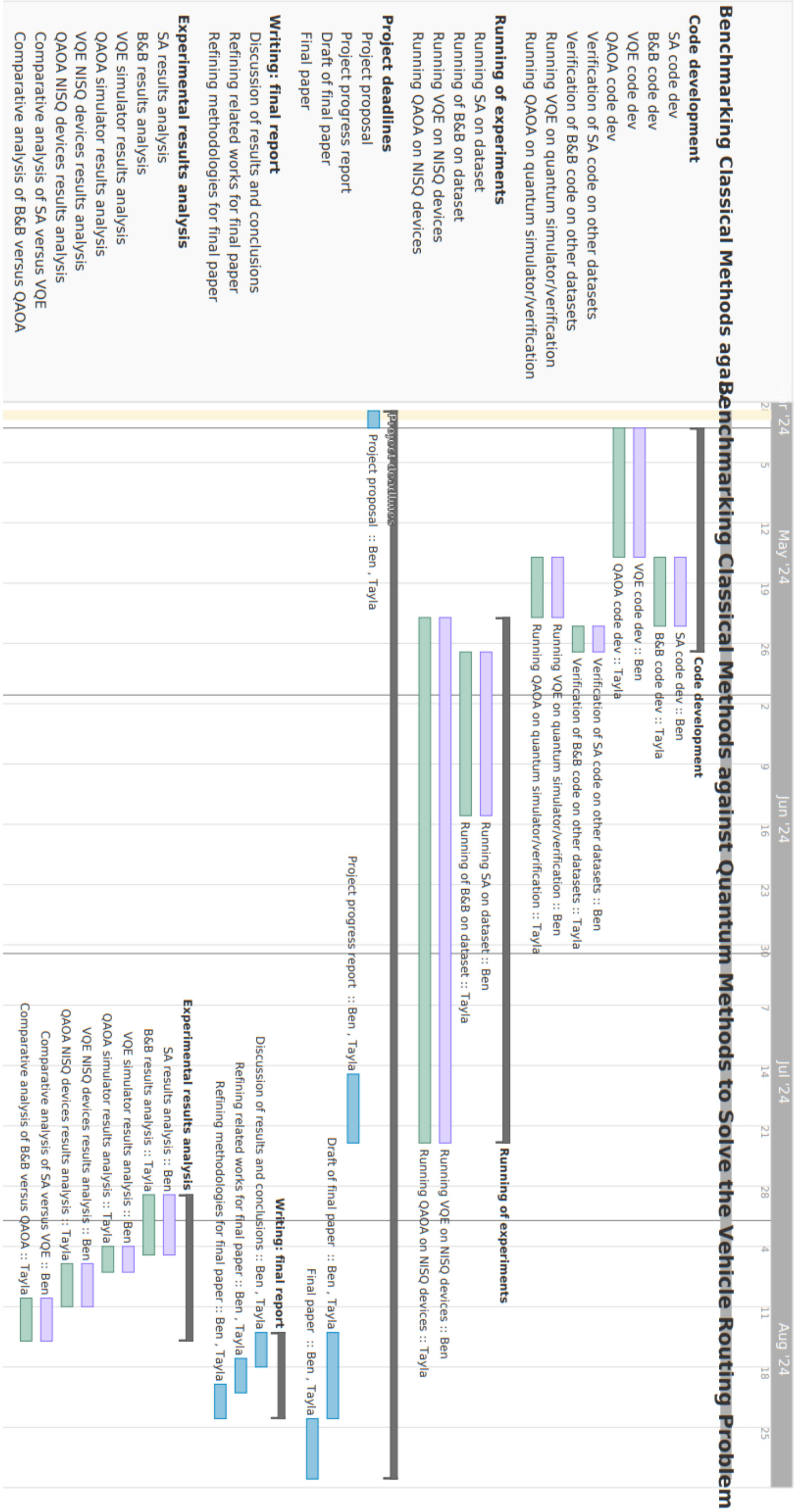

Figure 5: Gantt chart

Algorithm 1 Branch and Bound

$activeSet \leftarrow$ Initial Node
$bestVal \leftarrow$ Initial Value
$currentBest \leftarrow$ Initial Solution
$T0 \leftarrow Tmax$
while $activeSet$ not empty **do**
    Choose branching node, node $k \in activeSet$
    Remove node k from $activeSet$
    Generate children of node k, child i, where $i \leftarrow 1, ..., nk$
    and their respective lower bounds, lb_i
    for $i = 1$ to nk **do**
        if child i dominated by another **then**
            reject child i as possible solution
        else if child i is a complete solution **then**
            $bestVal \leftarrow lb_i$
            $currentBest \leftarrow$ child i
        else
            add child i to $activeSet$
        end if
    end for
end while
return currentBest

Symbol	Description
$activeSet$	Set of nodes currently being considered
$bestVal$	Current best objective function value
$currentBest$	Current best allocation of routes to vehicles
$T0$	Maximum time allowed to run
i	Current sub-problem being analysed
nk	Number of sub-problems of current node
lb	Lower-bound of current sub-problem

Table 1: Table of symbols and descriptions for Branch and Bound

Algorithm 2 Simulated Annealing

$S \leftarrow$ Generate initial solution()
$CoolingFactor \leftarrow$ some low number between 0 and 1
$k \leftarrow$ Boltzmann constant (often set to 1)
$T0 \leftarrow Tmax$
$Tfinal \leftarrow$ Predefined minimum temperature
Temp $\leftarrow T0$
while Temp $< Tfinal$ **do**
    while equilibrium condition not satisfied **do**
        $S' \leftarrow$ Generate new solution in neighborhood of S
        if $f(S') < f(S)$ **then**
            $S \leftarrow S'$
        else
            $\Delta \leftarrow f(S') - f(S)$
            $r \leftarrow$ generate random number between 0 and 1()
            if $r < \exp(-\Delta/(k/Temp))$ **then**
                $S \leftarrow S'$
            end if
        end if
        end while
    $T \leftarrow Temp \times CoolingFactor$
end while
return best found solution

Symbol	Description
S	Initial solution provided by a construction heuristic
$CoolingFactor$	Some low number between 0 and 1, controls temperature decrease
k	Boltzmann constant (often set to 1)
$T0$	Initial temperature
$Temp$	Current value of temperature parameter, $Temp \leftarrow T0$
S'	New solution in neighborhood of S
$f(S)$	Objective function value of solution S
ΔE	$f(S') - f(S)$, change in objective function value
r	Random number between 0 and 1 (uniform distribution)
$Equilibriumcondition$	Predefined number of iterations to be performed at each temperature

Table 2: Table of symbols and descriptions for Simulated Annealing

Algorithm 3 Variational Quantum Eigensolver

Input:
inputmatrix(*ces*), *circ*, *initialpoint*, *maxiter*
Output:
π^*, $cost^*$
Initialisation;
qubitOpdocplex ← BuildModel(inputmatrix(ces))
num ← number of qubits of qubitOpdocplex
spsa ← SPSA(maxiter)
if *circ* = RA **then**
    ry ← RealAmplitudes(num, entanglement=linear)
else if *circ* = TL **then**
    ry ← TwoLocal(num, entanglement=linear)
end if
vqe ← VQE(*qubitOpdocplex*, *ry*, *spsa*, *initialpoint*)
quantuminstance ← Backend(1024 shots)
result ← Run(*vqe*,*quantuminstance*)
π^*, $cost^*$ ← FeasibleOutput(result[eigenstate])

Symbol	Description
inputmatrix(*ces*)	Cost matrix of the VRP
circ	Quantum circuit
initialpoint	Initial set of parameters for the quantum circuit
maxiter	Maximum number of iterations for the optimisation process
π^*	Solution's allocation of vehicles to routes, $Temp \leftarrow T0$
$cost*$	Cost associated with solution
qubitOpdocplex	Quantum operator
num	The number of qubits required for the quantum operator that represents the VRP.
spsa	Variable assigned to instance of optimiser
ry	Variable representing the parameterized quantum circuit used in the VQE
vqe	The variable that will store the instance of the VQE algorithm
quantuminstance	A quantum environment where the VQE algorithm will be executed
result	The output from executing the VQE instance on the quantum instance

Table 3: Table of symbols and descriptions for Variational Quantum Eigensolver

Algorithm 4 Quantum Approximate Optimisation Algorithm

Input:
inputmatrix(*ces*), *initialpoint*, *maxiter*
Output:
π^*, $cost^*$
initialisation
qubitOpdocplex ← BuildModel(*inputmatrix*(*ces*))
num ← number of qubits of *qubitOpdocplex*
spsa ← SPSA(*maxiter*)
qaoa ← QAOA(*qubitOpdocplex*, *spsa*, *initialpoint*)
quantuminstance ← Backend(1024 shots)
result ← Run(*qaoa*, *quantuminstance*)
π^*, $cost^*$ ← FeasibleOutput(*result*[*eigenstate*])

Symbol	Description
inputmatrix(*ces*)	Cost matrix of the VRP
initialpoint	Initial parameters for the quantum circuit that will be optimised
maxiter	Maximum number of iterations for optimisation process
π^*	Solution's allocation of routes to vehicles
$cost^*$	Cost associated with solution
qubitOpdocplex	Quantum operator
num	Number of qubits required for quantum operator to represent VRP
spsa	Variable assigned to instance of optimiser
qaoa	Instance of QAOA algorithm
quantuminstance	Quantum environment where QAOA algorithm will be executed
result	Output from executing the QAOA instance on quantum instance

Table 4: Table of symbols and descriptions for Quantum Approximate Optimisation Algorithm