DEPARTMENT OF COMPUTER SCIENCE

CS Honours Project
Final Paper 2024

Title: Benchmarking Classical Methods against Variational Quantum Eigensolver to Solve the Vehicle Routing Problem: Simulated Annealing

Author: Ben Cleveland

Project Abbreviation: VRP

Supervisor(s): Krupa Prag

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

Benchmarking Classical Methods against Variational Quantum Eigensolver to Solve the Vehicle Routing Problem: Simulated Annealing

Ben Cleveland
University of Cape Town
Cape Town, South Africa

1 Abstract

The Vehicle Routing Problem (VRP), a well-established combinatorial optimisation problem, presents significant challenges due to its classification as an NP-hard problem. This study has benchmarked the performance of the classical algorithm Simulated Annealing (SA) against the Variational Quantum Eigensolver (VQE), a hybrid quantum-classical algorithm implemented on IBM's quantum simulators using their Qiskit framework. SA has been tested using varied parameters, while VQE has been evaluated with two distinct classical optimisers. Additionally, results of Branch and Bound (B&B) from a publication by [35] have been used as a reference, representing the optimal solution for each dataset instance. The experiments have been conducted on a dataset derived and adapted from a Capacitated Vehicle Routing Problem (CVRP) dataset, where the number of vehicles and customers has been reduced to create problem instances suitable for both classical and quantum methods. It has been found that while SA has efficiently solved all problem instances and has offered significantly faster computational times, the VQE algorithm has only been able to generate solutions for a limited part of the dataset. The increase in the complexity of the VRP instances, combined with the limitations of current quantum simulators and algorithms, contributed to the reduced performance of the VQE algorithm. This paper has contributed to the growing body of the literature in this field by providing a well-documented comparative analysis of VQE and SA, with an emphasis on solution quality, computational time and the feasibility of using quantum algorithms for optimisation problems. The contribution also lies in the code, data and experimentation pipelines written and made publicly available. The results underscore the need for further advancements in quantum computing to enhance the applicability of quantum algorithms for complex optimisation tasks like the VRP.

2 Introduction

2.1 Problem Formulation

The Vehicle Routing Problem (VRP), originally introduced by Dantzig and Ramser in 1959 [7], is a famous combinatorial optimisation problem that has significant relevance to the logistics industry and the research field of computational complexity alike. The problem is involved in determining optimal routes for a set of vehicles tasked with visiting a group of customers by minimising the cumulative distance travelled by the vehicle. The mathematical formulation is shown in Figure 9. Optimisation is achieved by minimising over the objective function, shown in equation 1, obtained from [6]. Adaptations and additions to this initial base problem are represented by Rich VRPs (RVRPs) [10]. The relevance of this problem has been extended by RVRPs to other specific scenario sets, for example by considering time as another parameter to be minimised [10]. Thus, the implications of this problem have been broadened from simple transportation logistics to other domains like scheduling, healthcare and refuse collection [42]. Advances that have been made in finding efficient solutions to various forms of this problem could have far-reaching effects on many individuals and industries across numerous circumstances, with the most obvious effects being to reduce costs [10]. The VRP is representative of a wider class of algorithms, defined as an $\mathcal{NP}$-hard problem [41]. These problems grow exponentially in size relative to the increase in input size, making a comprehensive search of the solution space via brute force and exact methods intractable at a certain point. The complexity of this problem and therefore the potential impact of finding improved solution methods, has given the VRP its significance.

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

Quantum computing has become an area of growing interest in solving combinatorial optimisation problems. With ongoing technological advancement in the field, quantum principles like entanglement and superposition have been explored as potential alternative solution approaches. Quantum algorithms rely on these principles to explore multiple potential routes concurrently across a large solution space. These algorithms can be executed on IBM's Noisy Intermediate-Scale Quantum (NISQ) devices or by using quantum simulators. The VRP can be formulated as an energy function, shown in Figure10. This has allowed for the development of a Quadratic Unconstrained Binary Optimisation (QUBO) model, as shown in Figure 11. This QUBO model has been subsequently transformed to an Ising Hamiltonian, shown in Figure 12. This final transformation allows the problem to be represented in a manner that can be used by a quantum algorithm to generate solutions. This Ising model, shown in equation 2, obtained from [6], calculates the energy level of a particular qubit arrangement in a quantum circuit.

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

Current quantum algorithms have been constrained by limited qubits used to represent information [2, 36]. By benchmarking these algorithms against one another, as well as against other classical benchmarks, insight into their current capabilities can be yielded. This study has built upon past work done on quantum algorithms

[3, 6], comparing a quantum algorithm to two well-established classical algorithms using a variety of metrics.

2.2 Problem Statement

This research has investigated the capabilities of quantum methods in solving the VRP and compared their performance with classical methods. Quantum algorithms, such as VQE, have been evaluated against classical algorithms like Simulated Annealing (SA) using IBM's Qiskit framework and simulators. A benchmark based on instances of the CVRP has been developed to evaluate solution quality, computational time and scalability. The limitations of quantum algorithms under current NISQ device constraints have also been explored to understand their potential advantages or use case over classical approaches.

2.3 Research questions

The research has been intended to address the following questions:

- How do variational quantum methods perform in terms of solution quality and computational time compared to metaheuristic and exact classical methods across different problem instance sizes of the VRP?
- Is there a quantum advantage over classical methods for solving the VRP, and if so, under what conditions does this advantage manifest?
- For what range of problem instance sizes can the latest publicly available IBM quantum Qiskit algorithms and simulators produce solutions that are within 95% and 99% of the globally optimal solution?

3 Related Work

Both exact methods and heuristics are referred to as being in the class of classical approaches, with heuristics being further divided into classical heuristics and metaheuristics. Exact approaches are comprised of comprehensive and exhaustive searches of the solution space using mathematical principles and only small-to-medium sized instances of optimisation problems are able to be solved to optimality with these methods, with larger instances quickly becoming intractable due to time or memory constraints [16]. Branch and bound (B&B), proposed in 1969 by Christofides and Eilon [13], is an initial version of such an approach. Modern heuristic solutions to the VRP started with the increased research into metaheuristic algorithms during the 1980's and 1990's [26], consisting of population-based, local search-based and learning mechanisms [15]. Over the past few decades, metaheuristics have been deemed as the preferred approach for solving the VRP, driven by advances in computational power that have enabled more sophisticated algorithms [17]. These algorithms have been observed to strike a balance between exploration and exploitation, allowing them to find high-quality solutions while navigating the search space efficiently, even permitting less optimal or infeasible solutions to enhance exploration[25]. Due to their ability to produce good solutions quickly and scale effectively [18], metaheuristics have become the most widely used techniques for solving the VRP, as confirmed by a survey [24]. SA, a guided-random local search heuristic based on actual annealing, was first developed in 1983 by [23]. In some notable applications, it has been shown to outperform or match other algorithms in the CVRP with 2D loading constraints, often improving the best-known solutions (BKS) [44]. A parallel SA approach applied to the VRP with simultaneous pickup-and-delivery and time windows has also outperformed a genetic algorithm (GA) across various instance sizes [12, 43]. Current quantum computing hardware is characterised by NISQ devices, which are limited by their small qubit size and high error rates, restricting their overall capabilities [8]. Variational quantum algorithms (VQAs) offer a promising way to leverage NISQ devices. These hybrid quantum-classical algorithms utilise shallow quantum circuits alongside classical optimisers to minimise a cost function, effectively managing the noise inherent in NISQ systems while optimising parameters with classical optimisers [30, 32]. Quantum computing has been applied far less frequently to combinatorial optimisation problems (COPs) compared to classical methods like metaheuristics, which have reliably been employed for decades. Quantum approaches are still being developed and have been more limited in scope. The first significant publication specifically addressing the VRP in a quantum context has appeared in 2019 [21]. The quantum approach investigated in this paper has been the Variational Quantum Eigensolver (VQE). Introduced in 2014 [34], it was initially developed for quantum chemistry to approximate the ground state of a Hamiltonian [1]. Khumalo et al. [22] has conducted a significant study benchmarking quantum algorithms, specifically VQE and QAOA, against classical methods like B&B and SA. Their analysis has focused on the Travelling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP) and has shown that classical algorithms significantly outperformed their quantum alternatives, particularly in solving medium to large problem instances. This large difference in performance has been attributed to the limited qubit availability in quantum hardware, which has constrained circuit depth and computational capacity. Although insightful, their study has been focused on simpler problems compared to the VRP, which is a more complex generalization of the TSP, as highlighted by Azad et al. [6]. The need to further explore the viability of applying quantum algorithms to more challenging problems has been underscored by this complexity. In subsequent research by Alsaiyari and Felemban [3] VQE and QAOA have been evaluated against IBM's CPLEX classical optimiser, which has been based on exact algorithms. However, their study has been limited to evaluating solution quality alone, where other important metrics like computational time and scalability have been neglected. Moreover, the comparison has been confined to the CPLEX optimiser and has omitted other classical methods such as metaheuristics that could provide a more comprehensive assessment of quantum algorithms. VQE and QAOA have been applied to various combinatorial optimisation problems (COPs), including frequency allocation, register allocation and flight gate allocation [31]. Notably, VQE has been tested on the Max-Cut problem, an $\mathcal{NP}$-Complete binary optimisation problem, using an ideal quantum simulator. In instances with up to four nodes, the optimal solution has been successfully found by VQE 95% of the time [30].

4 Methodology

4.1 Methods

4.1.1 Simulated Annealing SA is a metaheuristic algorithm inspired by the process of metallurgical annealing, as detailed by Gendreau and Potvin [19]. The algorithm works by evolving an initial solution towards optimality using a guided-random local search, balancing the exploration of the search space with the exploitation of promising known solutions. In each iteration, a neighbouring solution is created and evaluated. If this neighbouring solution provides an improvement, it is accepted unconditionally. However, if the solution is worse, it may still be accepted based on a probability derived from the Boltzmann distribution and modulated by the current temperature parameter [5]. This probabilistic acceptance of worse solutions allows the algorithm to escape local optima, with the likelihood of such acceptance being decreased as the temperature parameter cools over time [5]. The SA algorithm requires several key parameters and inputs: the initial solution, the initial temperature, the cooling schedule, a stopping criterion, an equilibrium condition and an objective function [5]. The initial temperature and the cooling schedule must be carefully calibrated to prevent early convergence to local optima while also avoiding unnecessary computational expenditure [20]. The objective function typically aims to minimise the total cumulative route lengths, serving as the metric by which solutions are evaluated. The equilibrium condition specifies the number of iterations performed at each temperature level, providing a balance between exploration and exploitation at different stages of the cooling process [20]. The algorithm is typically terminated either when a predefined number of cooling cycles have been completed or when a final temperature threshold is reached [20]. The pseudocode for SA, adapted from [5], is shown in Algorithm 1. An explanation of the variables used can be found in Table 4.

4.1.2 Variational Quantum Eigensolver VQE is a hybrid algorithm that leverages both quantum and classical computational resources to approximate the eigenvalue of a Hamiltonian matrix [29]. This hybrid approach is designed to find the minimum eigenvalue, which corresponds to the ground state or the lowest energy state, of the Hamiltonian — a solution that represents the optimal solution of the associated optimisation problem [1]. The algorithm begins with an ansatz, a parameterised trial wavefunction that is used as an initial guess for the solution [11]. The quantum component of the algorithm evaluates the energy of the ansatz [22], while a classical optimiser iteratively tunes the parameters to minimise this energy [1]. This quantum-classical iterative process, which is managed by a classical machine, is continued until a termination criterion is met, at which point the algorithm will converge to an approximate solution for the Hamiltonian's ground state. The pseudocode for VQE, adapted from [22], is shown in Algorithm 2. An explanation of the variables used can be found in Table 5.

4.2 Design and Implementation

In order to apply SA and VQE to VRP and to our chosen dataset, many design and implementation decisions have been undertaken. These decisions and parameters are detailed in the following section. The full dataset and code implementation can be found on the project's Github page.

4.3 SA for VRP: Implementation

An initial solution is required for the SA algorithm to begin the iterative process. A default approach involves the generation of random solutions; however, a more effective method is to employ the savings algorithm. The savings algorithm, proposed by Clarke and Wright [14], has been recognised as a well-known heuristic favoured for its efficiency in generating suitable solutions. Although optimality is not guaranteed, experimentation conducted in this study has demonstrated that it generally outperforms random generation in terms of solution quality. Several SA parameters have been selected based on established approaches in the literature. For instance, a cooling rate of 0.99 has been chosen, implying that the total temperature has been reduced by 1% at each step. This value has been selected because it is small enough to allow for gradual cooling of the system. While the optimal cooling rate is problem-dependent, a range between 0.85 and 0.99 has been suggested in the literature [4]. Preliminary experimentation has indicated that a cooling rate of 0.01 has been effective, enabling the implementation to navigate the complex solution landscape of the VRP instances gradually. The initial temperature and the number of iterations at each temperature level have been determined through a series of experiments with various parameter values. Initial temperature values tested included T=10, 20, 50, 100, 500 and 1000. The number of iterations at each temperature has been tested with values of 10, 20, 50 and 100. The combination that, for which, the highest solution quality has been consistently produced has involved setting both the initial temperature and the number of iterations at each temperature level to 100. In addition to the default stopping criterion of reaching a temperature of zero, an additional stagnation parameter has been introduced. This parameter terminates the search and returns the best solution if no improvement is observed in the current best solution after a predetermined number of iterations. Experiments using SA have been conducted with the stagnation limit set initially to 100 and subsequently to 1000. These two implementations of SA are hereafter referred to as SA100 and SA1000, respectively.

4.4 VQE for VRP: Implementation

VQE has been implemented using Qiskit's optimisation module [40]. The code provided by [39] has been adapted for this purpose. Qiskit's optimisation module offers the "MinimumEigenOptimizer" class, which facilitates the translation of optimisation problem instances into Ising Hamiltonians, the solution of these Hamiltonians (using SamplingVQE in this case), and the translation of the results back into an optimisationResult [38]. SamplingVQE focuses on sampling circuits rather than directly calculating expectation values [37]. Several linear constraints have been incorporated to ensure that valid solutions have been generated by the algorithm. These constraints have been necessary to guarantee that the same number of vehicles have departed from and returned to the depot and to ensure that all customers have been visited exactly once. The ansatz employed has been Real Amplitudes and two classical optimisers, COBYLA and SPSA, have been used in the experiments.

4.5 Experimental Settings

Classical algorithms have been implemented in Python, while the quantum algorithms have been implemented using IBM's Qiskit, a Python-based SDK. All experiments have been executed on a remote Linux server equipped with an AMD Ryzen 7 3800X 8-Core Processor with 16 threads, 32 GB of RAM and an NVIDIA GeForce RTX 2060 GPU. The quantum experiments utilised IBM's Qiskit quantum simulators. Two pipelines have been developed in Python to automate the execution and recording of the output for each data point in the dataset. Each configuration of each algorithm has been run six times on each dataset item and the output has been discarded if a memory limit has been reached or if the time limit of six hours has been exceeded on a given run. The data used in this research has been adapted from 240 instances of the CVRP as specified in [28], with the original dataset available at [27]. CVRP-specific constraints have been removed to create general asymmetric VRP instances. The size of the dataset items has been reduced to create instances of appropriate size for the quantum methods. The final dataset has included instances ranging from 2 customers and 1 vehicle to 15 customers and 8 vehicles.

4.6 Metrics

In this research, the evaluation of algorithms for addressing the VRP has been centred on the minimisation of vehicle trip distances, which has been referred to as solution quality. CPU time has also been considered as a crucial metric, reflecting the practicality of each algorithm. Algorithms have been terminated upon having reached a predefined timeout limit of 6 hours. The algorithms under investigation, specifically SA and VQE, have been executed 6 times per problem instance to comprehensively record the best, average and worst-case results. The consistency of the solutions has been evaluated by calculating the standard deviation and variance of the results. The scalability of the algorithms has been assessed to determine the range of problem sizes over which each algorithm has maintained effective performance. The feasibility percentage has been analysed by noting the proportion of solutions that meet all problem constraints for each instance. Global optimal solutions, generated using the B&B method as documented in [35], have served as benchmarks. The success rates of the algorithms have been determined by the calculation of the percentage of solutions that fall within 95% and 99% proximity to these benchmark solutions. The following metrics have been systematically recorded:

- Optimal Costs: Including the lowest and highest costs along with their respective frequencies across runs.
- Execution Times: Documenting the best, average and worst-case durations for algorithm completion.
- Standard Deviations: Measuring the variability in both optimal costs and execution times to assess the stability of the solutions.

This comprehensive evaluation framework has been designed to facilitate a thorough comparison of the effectiveness, efficiency and reliability of the algorithms in solving complex VRP instances.

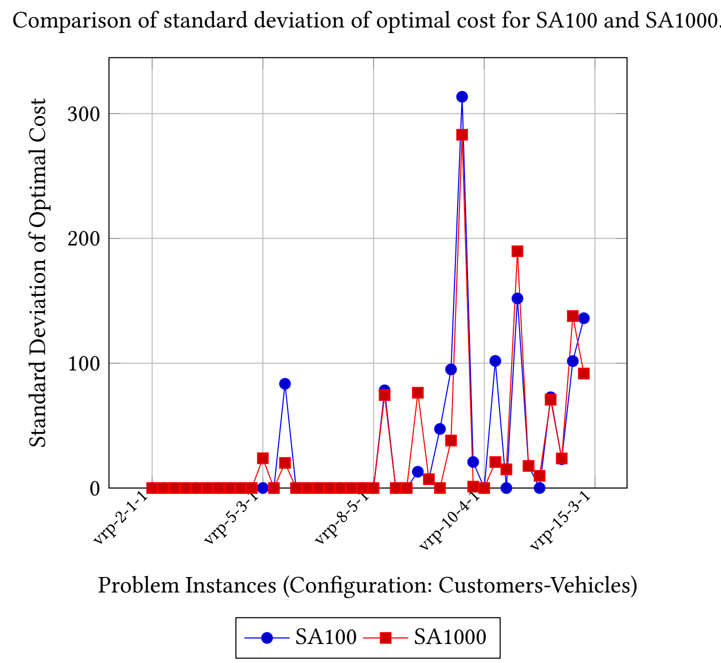

Figure 1: Comparison of standard deviation of optimal cost for SA100 and SA1000.

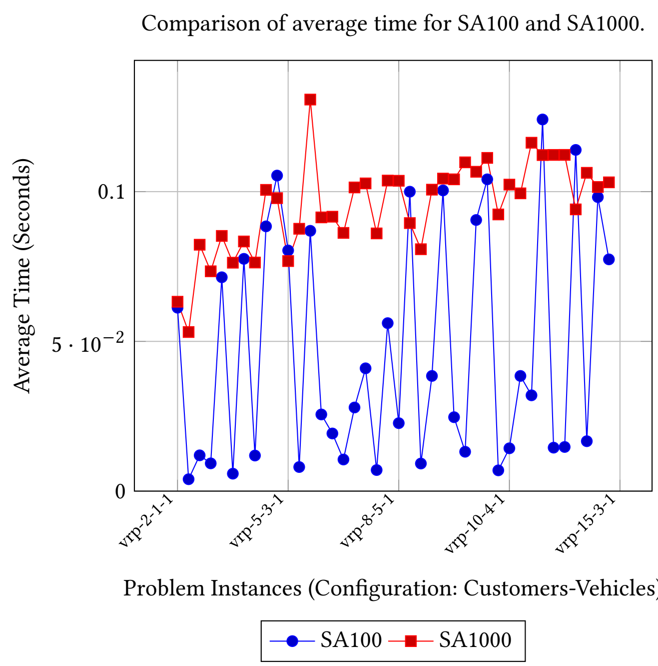

Figure 2: Comparison of average time for SA100 and SA1000.

4.7 Results and Outcomes

4.7.1 Comparison of SA100 and SA1000 The variability in time was lower for SA100 for the first 12 instances, as shown in Figure3. The comparison between the two became inconsistent over

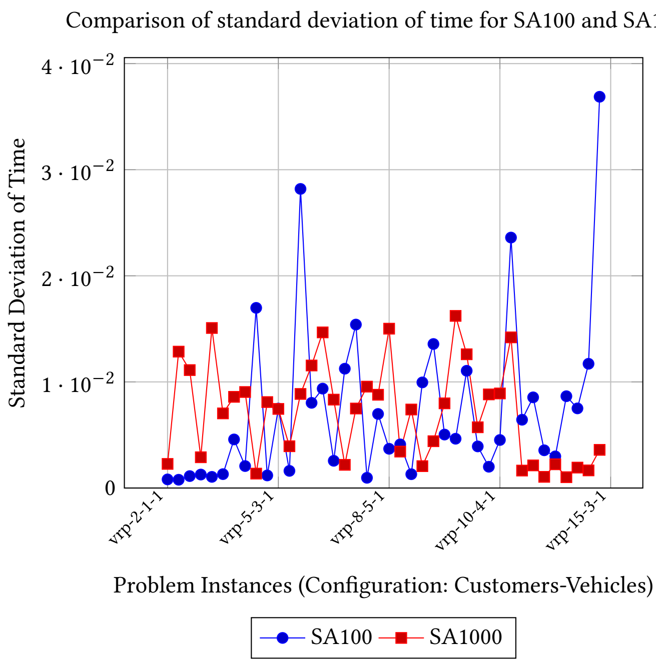

Figure 3: Comparison of standard deviation of time for SA100 and SA1000.

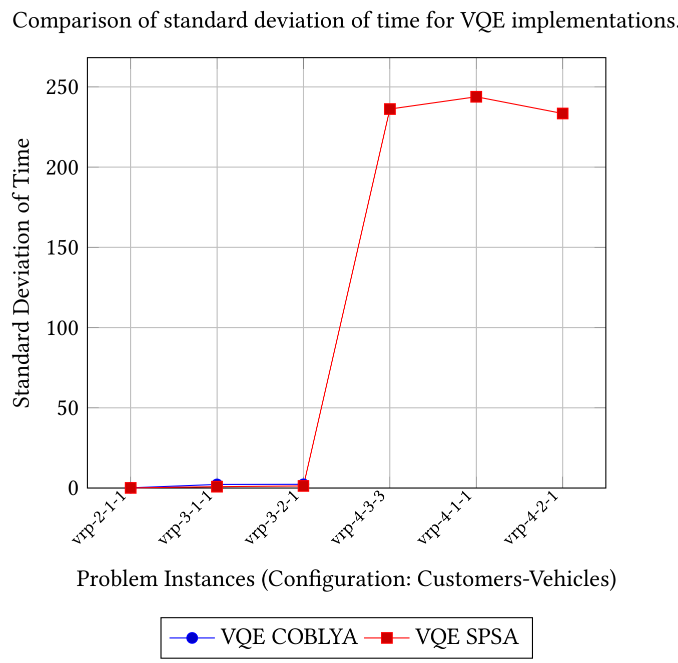

Figure 4: Comparison of standard deviation of time for VQE implementations.

the middle of the dataset, with values alternating between higher and lower for each algorithm, making it challenging to establish a clear trend. A significantly larger standard deviation in time was noted for SA100 over some instances including vrp-10-8-1, vrp-10-3-1 and vrp-15-3-1. A consistent reduction in the variability of SA1000's time was evident for the last 8 instances when compared with SA100. In standard deviation of lowest optimal cost, a similar pattern was displayed between the two algorithms, seen in Figure 1. Low and stable standard deviations were generally shown by both algorithms across many instances. However, several spikes in the standard deviation were noted for both algorithms, with particularly large deviations occurring at vrp-10-5-1, vrp-15-7-1 and vrp-15-2-1 1. These spikes were often seen in the same points, but their peaks also alternated unpredictably at times, showing the difference in response of the two algorithms over certain problem instances. Despite the high variability at times, 100% feasibility was maintained across all problem instances for both algorithmic implementations as shown in Table 1. The average time was consistently higher for SA1000 for nearly all problem instances. As demonstrated in Figure 2, the general trend and spiking patterns over specific instances were generally mirrored between the two algorithms, but with SA100's average time at a consistently lower level and SA1000's peaks and valleys more condensed. Notably, higher average time values were observed for SA100 for a few instances, despite it having a smaller stagnation tolerance. SA1000 demonstrated a higher success rate in reaching near-optimal solutions within 95% and 99% of the global optimum, as illustrated in Table 1, especially for complex instances. The complete tables of results for SA100 and SA1000 are presented in Table 7 and in Table 8 respectively.

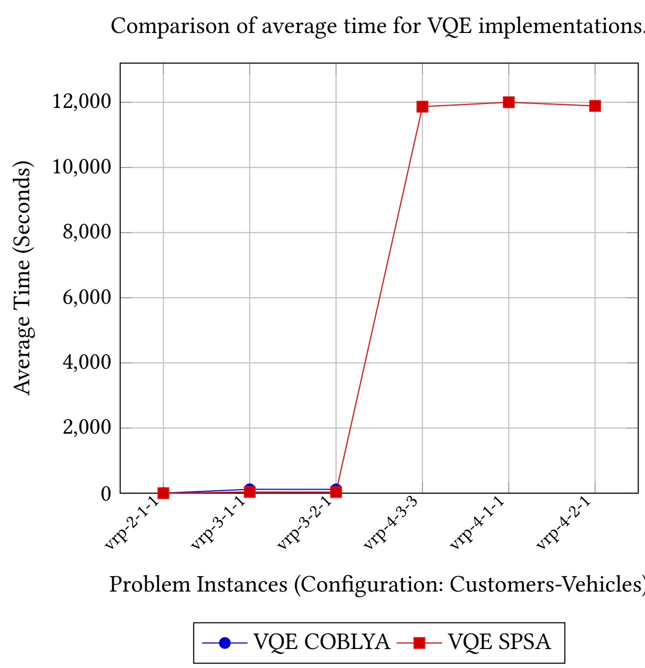

Figure 5: Comparison of average time for VQE implementations.

4.7.2 Comparison of VQE with COBYLA and VQE with SPSA Better performance was displayed by VQE with SPSA than VQE with

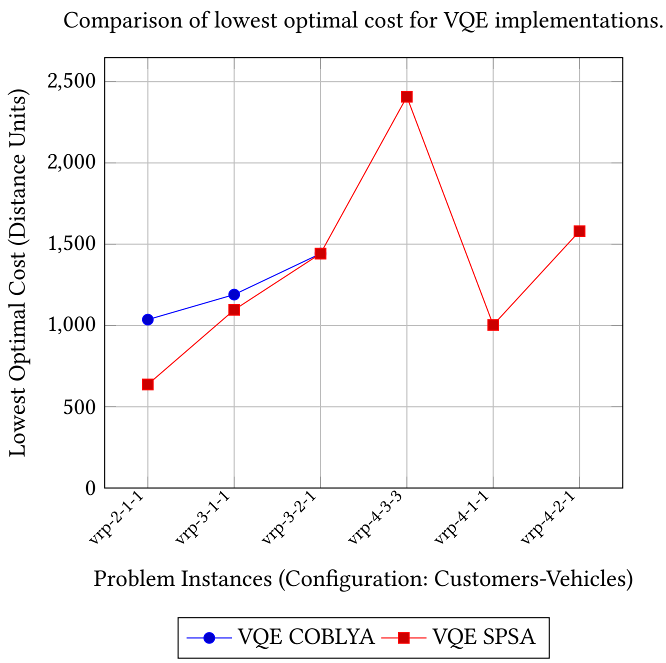

Figure 6: Comparison of lowest optimal cost for VQE implementations.

COBYLA in terms of scalability. Only smaller instances (vrp-2-1-1, vrp-3-1-1, vrp-3-2-1), were successfully solved by VQE with COBYLA, as shown in Table 3. An additional three instances, namely vrp-4-3-3, vrp-4-1-1 and vrp-4-2-1, were solved by VQE with SPSA. Feasible solutions could not be found for larger problem instances by either algorithmic implementation, often due to reaching qubit limits or running out of memory. The success rates of VQE with COBYLA, as presented in Table 1, were a mere 33% for reaching solutions within 95% and 99% of the optimal solution, significantly worse than those of VQE with the SPSA, by which scores of 83% for both metrics were achieved. The standard deviation of optimal cost for the instances that were successfully solved remained zero for both implementations. As demonstrated in Figure 4, the standard deviation of time was relatively stable for both algorithms for the first three instances. However, the average times for VQE with COBYLA for these instances were approximately three times larger than those for VQE with SPSA, as shown in Table 2. After the third instance, the average times for both algorithms increased dramatically, with VQE using COBYLA eventually reaching the timeout for later instances and VQE with SPSA showing a steep increase, seen in Figure 5. A sharp spike in the standard deviation of time from below one to between 200 and 250 was observed for VQE with SPSA at this point, as seen in Figure 4. VQE with SPSA achieved costs that were less than or equal to the optimal costs obtained by VQE with COBYLA, as shown in Figure6. The complete tables of results for VQE with COBYLA and VQE with SPSA are provided in Table 10 and Table 11 respectively.

4.7.3 *Overall comparison of SA and VQE*

Performance: The VQE implementations were outperformed by both SA implementations in terms of finding the lowest optimal costs and maintaining feasibility across all problem instances, as shown in Table1. The VQE implementations were limited by current quantum algorithm and simulator capabilities, with both instances struggling to handle larger instances, as indicated in Table3. VQE with SPSA significantly outperformed VQE with COBYLA, achieving a success rate at the 95% level that was only approximately 10-13% lower than that of SA, and only 4-7% lower at the 99% level for the instances that VQE with SPSA was able to solve, as illustrated in Figure 13.

Computation Time: The SA implementations were significantly faster, with all problem instances completing within seconds, as shown in Table 2. In contrast, the VQE algorithms exhibited much longer computation times, especially for larger instances where hardware limits were frequently reached, as indicated in Table 2. This was particularly evident when using the COBYLA optimiser. Exponential increases in computation time were displayed by VQE as the problem size grew, as seen in Figure 5, starkly contrasted against the stable computation time seen with SA 8

4.7.4 *SA vs B&B*

Solution Quality: The B&B results were closely matched by both SA implementations, with most instances showing no deviation from the global optimal solution, as demonstrated in Figure 7. A reliable reference was provided by B&B [35], but competitive solutions were offered by the SA implementations with significantly less computational overhead, as indicated in Figure 8. For some of the largest instances that B&B could handle, a slightly lower optimal cost was achieved when compared to the two SA algorithms, as shown in Figure 7. The complete table of results for B&B is available in Table 9, from [35].

Scalability: B&B struggled with the largest instances, often returning out-of-memory errors, while SA remained robust across all problem sizes, as shown in Figure 7 where a few missing data points for B&B are seen. B&B's difficulties with larger instances highlight the computational burden of exact methods. SA's rapid computation times, even for the largest instances, make it a more practical option for real-world applications where time efficiency is critical, as shown in Figure 8.

4.7.5 VQE vs B&B VQE algorithms have fallen short compared to B&B, particularly in terms of solution quality and feasibility for larger instances, as shown in Figure 13. Even the best performance from VQE (using SPSA) has consistently failed to match B&B in terms of solution quality but did achieve success rates of 83% at both the 95% and 99% levels. The gap in solution quality between VQE and B&B was more pronounced in complex instances, with feasible solutions frequently having failed to have been produced by VQE within the time limit, as seen in Table 3. The feasibility percentages for VQE were low, indicating a scaling issue, as shown in Table 1. Twice as many feasible solutions were produced by VQE

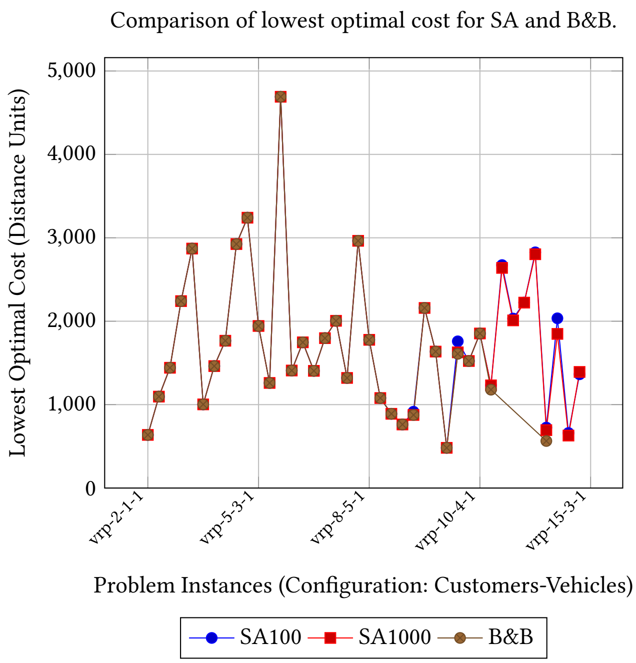

Figure 7: Comparison of lowest optimal cost for SA and B&B.

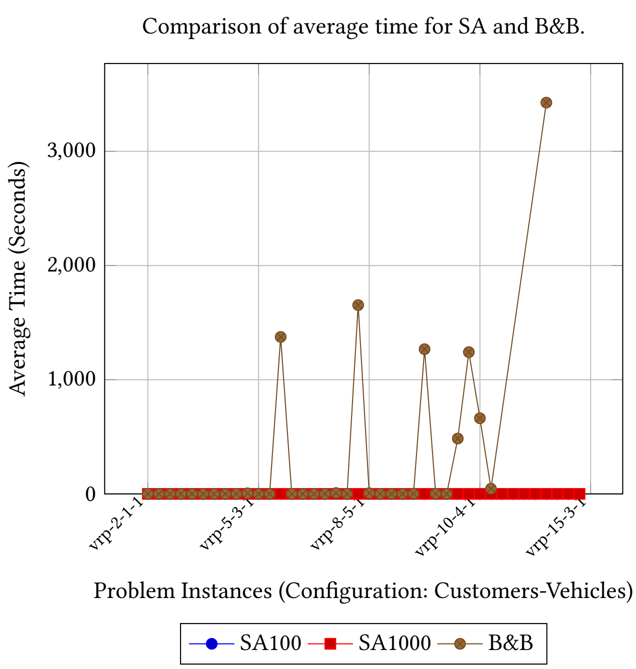

Figure 8: Comparison of average time for SA and B&B.

with SPSA when compared to VQE with COBYLA, as seen in Table 3.

4.8 Discussion

4.8.1 On SA SA100 has displayed very similar solution quality to SA1000, but with less computational overhead, which has made it the best choice in practical applications. A more thorough exploration of the solution space has been facilitated by the extended stagnation criterion of SA1000, exhibited in slightly lower variability in optimal cost and time for SA1000 over numerous instances. Spikes in the standard deviation of optimal cost have been seen in both SA implementations, indicating the complexity of certain VRP instances or variability in the initialization process when using the savings algorithm in a complex solution space. SA1000 has proven to be the most consistent method for solving the VRP across all instances, seen in lower standard deviations and higher success rates. This result has been expected, due to the greater opportunity for exploration of the solution space. The more condensed patterns observed in the average times SA1000 reinforce findings from the standard deviation of time chart, where the larger stagnation counter has contributed to more consistent exploration results. SA100 has occasionally recorded longer times than SA1000, despite its lower stagnation tolerance. This is likely due to the stochastic nature of the algorithm and initialisation method, where SA100 may continue exploring if it finds steady improvements, resulting in longer runs in some cases. The algorithm's execution time may have been prolonged by a gradually decreasing gradient in the solution landscape.

4.8.2 On VQE VQE remains limited by (simulated and real) hardware constraints and has yet to offer a practical advantage over classical methods for solving the VRP. For problem sizes with six or more customers, a qubit limit has been reached, indicating that the simulators are unable to handle the number of qubits required by the algorithm for problems of this complexity. In instances with five customers, 'out of memory' (OOM) errors have caused the termination of both VQE implementations, highlighting the significant computational demands of these simulations. Even with 32GB of RAM, these instances have proven too large to be solved by VQE. VQE with COBYLA has exhibited additional limitations. For all instances with four customers, it has failed to reach a solution within the six-hour timeout. Random parameter alterations and observations in output changes are utilised by SPSA to estimate the objective function gradient's direction, making it more effective in stochastic and noisy environments [9]. In contrast, COBYLA uses linear approximations to estimate the behaviour of the objective function and to simplify the complex solution space, which may account for its underperformance in noisy environments [9]. The superior performance of SPSA over COBYLA in this study is likely due to their differing abilities to handle noise. This aligns with the findings of [33], where, in a comparison of optimisers for a VQA on a quantum simulator, SPSA outperformed COBYLA and other algorithms in noisy conditions, while COBYLA performed well only in noise-free environments but was significantly affected by slight levels of noise.

4.8.3 On SA vs B&B The scalability of SA to different problem sizes without significant degradation in solution quality has contrasted with the limitations faced by B&B in large instances. For practical purposes, the metaheuristic principle of finding "good

Table 1: Algorithm Summary Statistics.

Algorithm Name	Scalability	Success Rate 95%	Success Rate 99%	Feasibility Percentage
VQE COBLYA	(vrp-2-1-1; vrp-3-2-1)	0.3333333333333333	0.3333333333333333	0.09090909090909091
VQE SPSA	(vrp-2-1-1; vrp-4-2-1)	0.8333333333333334	0.8333333333333334	0.18181818181818182
SA100	(vrp-2-1-1; vrp-15-3-1)	0.9393939393939394	0.8787878787878788	1.0
SA1000	(vrp-2-1-1; vrp-15-3-1)	0.9696969696969697	0.9090909090909091	1.0

Table 2: Average Execution Time per Algorithm for Each Problem Instance.

Problem Instance	B&B	SA100	SA1000	VQE COBYLA	VQE SPSA
vrp-2-1-1	3.2e-05	0.061245	0.063238	3.361962	1.356431
vrp-3-1-1	6e-05	0.003971	0.053122	119.490761	37.477671
vrp-3-2-1	0.000121	0.011948	0.082272	117.40928	35.350031
vrp-4-3-3	0.000665	0.009268	0.073405	T	11867.531866
vrp-5-4-1	0.00593	0.071408	0.08521	OOM	OOM
vrp-4-1-1	0.000172	0.005827	0.076273	T	12003.012198
vrp-6-5-1	0.057502	0.077558	0.083363	QL	QL
vrp-4-2-1	0.000498	0.011879	0.076331	T	11889.223621
vrp-7-6-1	0.524886	0.088409	0.100585	QL	QL
vrp-8-7-1	5.30178	0.105369	0.097823	QL	QL
vrp-5-3-1	0.005993	0.080363	0.076814	OOM	OOM
vrp-5-1-1	0.000876	0.008043	0.0876	OOM	OOM
vrp-10-8-1	1374.614076	0.08692	0.130749	QL	QL
vrp-5-2-1	0.002538	0.025599	0.0914	OOM	OOM
vrp-6-4-1	0.051211	0.019246	0.091629	QL	QL
vrp-6-1-1	0.005942	0.010556	0.086217	QL	QL
vrp-7-5-1	0.566002	0.027923	0.101403	QL	QL
vrp-8-6-1	7.359442	0.041005	0.102752	QL	QL
vrp-6-2-1	0.013876	0.007039	0.086052	QL	QL
vrp-10-7-1	1653.38202	0.05609	0.103701	QL	QL
vrp-8-5-1	6.394035	0.022693	0.103628	QL	QL
vrp-7-4-1	0.364664	0.099981	0.0895	QL	QL
vrp-6-3-1	0.0312	0.009211	0.080772	QL	QL
vrp-8-1-1	0.023606	0.03846	0.100662	QL	QL
vrp-8-4-1	2.592381	0.100406	0.104363	QL	QL
vrp-10-6-1	1267.361367	0.024684	0.10411	QL	QL
vrp-8-3-1	2.024722	0.013145	0.109783	QL	QL
vrp-8-2-1	0.11997	0.090529	0.106636	QL	QL
vrp-10-5-1	484.846432	0.104109	0.111267	QL	QL
vrp-10-1-1	1241.610846	0.006936	0.092386	QL	QL
vrp-10-4-1	662.675332	0.014299	0.102346	QL	QL
vrp-10-3-1	45.289391	0.038459	0.099431	QL	QL
vrp-15-8-1	OOM	0.031988	0.116334	QL	QL
vrp-15-7-1	OOM	0.124098	0.112213	QL	QL
vrp-15-6-1	OOM	0.014515	0.11226	QL	QL
vrp-15-5-1	OOM	0.014737	0.112272	QL	QL
vrp-15-1-1	3426.624613	0.113931	0.094072	QL	QL
vrp-15-4-1	OOM	0.016666	0.106286	QL	QL
vrp-15-2-1	T	0.098181	0.101552	QL	QL
vrp-15-3-1	T	0.077384	0.103104	QL	QL

OOM: Out of memory, QL: Qubit limit, T: Timeout

Table 3: Lowest Optimal Cost per Algorithm for Each Problem Instance

Problem Instance	B&B	SA100	SA1000	VQE COBYLA	VQE SPSA
vrp-2-1-1	637.0	637.0	637.0	1036.0	637.0
vrp-3-1-1	1096.0	1096.0	1096.0	1190.0	1096.0
vrp-3-2-1	1442.0	1442.0	1442.0	1442.0	1442.0
vrp-4-3-3	2241.0	2241.0	2241.0	T	2407.0
vrp-5-4-1	2871.0	2871.0	2871.0	OOM	OOM
vrp-4-1-1	1003.0	1003.0	1003.0	T	1003.0
vrp-6-5-1	1462.0	1462.0	1462.0	QL	QL
vrp-4-2-1	1766.0	1766.0	1766.0	T	1580.0
vrp-7-6-1	2925.0	2925.0	2925.0	QL	QL
vrp-8-7-1	3241.0	3241.0	3241.0	QL	QL
vrp-5-3-1	1943.0	1943.0	1943.0	OOM	OOM
vrp-5-1-1	1260.0	1260.0	1260.0	OOM	OOM
vrp-10-8-1	4691.0	4691.0	4691.0	QL	QL
vrp-5-2-1	1409.0	1409.0	1409.0	OOM	OOM
vrp-6-4-1	1747.0	1747.0	1747.0	QL	QL
vrp-6-1-1	1405.0	1405.0	1405.0	QL	QL
vrp-7-5-1	1797.0	1797.0	1797.0	QL	QL
vrp-8-6-1	2005.0	2005.0	2005.0	QL	QL
vrp-6-2-1	1320.0	1320.0	1320.0	QL	QL
vrp-10-7-1	2964.0	2964.0	2964.0	QL	QL
vrp-8-5-1	1777.0	1777.0	1777.0	QL	QL
vrp-7-4-1	1079.0	1079.0	1079.0	QL	QL
vrp-6-3-1	890.0	890.0	890.0	QL	QL
vrp-8-1-1	762.0	762.0	762.0	QL	QL
vrp-8-4-1	876.0	915.0	876.0	QL	QL
vrp-10-6-1	2159.0	2159.0	2159.0	QL	QL
vrp-8-3-1	1636.0	1636.0	1636.0	QL	QL
vrp-8-2-1	482.0	482.0	482.0	QL	QL
vrp-10-5-1	1609.0	1759.0	1627.0	QL	QL
vrp-10-1-1	1523.0	1523.0	1523.0	QL	QL
vrp-10-4-1	1853.0	1853.0	1853.0	QL	QL
vrp-10-3-1	1179.0	1231.0	1231.0	QL	QL
vrp-15-8-1	OOM	2673.0	2639.0	QL	QL
vrp-15-7-1	OOM	2032.0	2010.0	QL	QL
vrp-15-6-1	OOM	2224.0	2223.0	QL	QL
vrp-15-5-1	OOM	2826.0	2803.0	QL	QL
vrp-15-1-1	564.0	725.0	694.0	QL	QL
vrp-15-4-1	OOM	2035.0	1848.0	QL	QL
vrp-15-2-1	T	660.0	629.0	QL	QL
vrp-15-3-1	T	1364.0	1392.0	QL	QL

enough" solutions in reasonable time is embodied in this analysis. The algorithmic designs of B&B and SA are such that near-optimal solutions have been found rapidly by SA, which then exits, whereas B&B can only exit after exploring all solutions. The relatively minor differences, on average, between the solution qualities of SA and B&B have been overshadowed by the substantial increases in solution time as B&B attempts larger instances. It has been suggested by B&B's results for the larger instances that instances even larger than those used in this study would likely be entirely intractable for exact methods, making metaheuristics the only viable approach.

4.8.4 On VQE vs B&B The limited success of VQE in matching B&B's optimal solutions, even in smaller instances, has suggested significant limitations in the current state of quantum algorithms. The low feasibility of VQE solutions, particularly in larger instances, indicates that current quantum methods are not yet competitive. Despite these challenges, the potential for quantum algorithms to explore solution spaces differently from classical methods remains a relevant research focus that should continue to be investigated as hardware and algorithmic capability increases.

4.8.5 On SA vs VQE The performance of VQE on VRP instances, in comparison to SA, has been hindered by poor scalability, worse solution quality and significantly more computational overhead. SA has consistently maintained feasibility and solution quality across all instances, while VQE has struggled, particularly in larger problem sets, due to qubit limits and memory constraints. The variability in VQE's computation time, even with SPSA, further underscores the maturity gap between quantum and classical methods. The scalability issues that are currently faced by quantum algorithms are exhibited by the exponential increase in computation time for VQE as the problem size grows. The ongoing challenges in making quantum computing a viable alternative for large-scale optimisation problems are highlighted by the contrast in computation times between SA and VQE.

5 Conclusions

This study has benchmarked SA against VQE using IBM's Qiskit quantum simulators in order to assess their performance with reference to solution quality, computation time and scalability. It has been demonstrated that VQE does not scale well compared to classical methods. The first six instances have been solved by VQE with SPSA, while only the first three instances have been solved by VQE with COBYLA within the specified timeout. For the instances that have been solved, VQE with COBYLA has performed poorly in terms of solution quality, having produced results within 95% to 99% of the optimal solution in only 33% of cases. In contrast, success rates of 83% have been achieved by VQE with SPSA. Near-optimal solutions have been consistently produced by SA, with success rates of 87% to 97% being achieved, typically within 0.2 seconds. In comparison, the computational times of VQE have ranged from 1 to 3 seconds for smaller instances, and up to 11,889 seconds for larger instances using SPSA, while the six-hour timeout has been reached by COBYLA on larger instances. It is indicated by these results that VQE has performed significantly worse than SA in terms of computational efficiency and scalability. A quantum advantage over classical methods has not been observed in this work. The inability of VQE to solve instances beyond the sixth, coupled with the exponential growth in computation time, has highlighted the lack of scalability of the algorithm compared to the consistently performant SA. Furthermore, B&B or other exact methods have been identified as the most accurate, though substantial time costs are incurred by these methods as problem size is increased [35]. As quantum simulators, hardware and algorithms are evolved, VQE may become more competitive, but for now, SA remains the more practical and reliable approach for solving the VRP. Considerable advancements are required in qubit count, error mitigation, noise tolerance and quantum algorithms to improve the viability of quantum computing for complex optimisation tasks. Metaheuristic approaches like SA have remained the most flexible and time-efficient solution for large-scale instances.

6 Limitations

The lack of access to real NISQ hardware has limited the research, as a more accurate assessment of quantum algorithms' practical performance would have been possible. Additionally, insufficient access to computers with more than 32GB of RAM has restricted the ability to handle larger problem instances, further constraining the experiments. Lastly, the available running time has not allowed for the originally intended 30 runs per instance on a larger dataset, which would have produced a more comprehensive benchmark.

7 Future Work

Future research should focus on optimising quantum algorithms to better handle larger instances and improve solution feasibility. Continued benchmarking against classical methods will be essential for measuring progress in the applicability of quantum computing to real-world problems like the VRP. Investigating hybrid approaches that combine classical heuristics with quantum algorithms may offer a strategy to leverage the strengths of both paradigms. The consistent and versatile performance of metaheuristics may suggest that there could be potential in combining quantum and metaheuristic approaches, as has been done for quantum and exact approaches in these VQAs. As quantum hardware evolves, the performance of algorithms like VQE will need to be re-evaluated to understand their potential in optimisation tasks. Furthermore, the development of more advanced quantum error correction and noise mitigation techniques could significantly enhance the feasibility and performance of quantum approaches in optimisation.

References

[1] 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 , 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. *arXiv.org* (12 2023). https://doi.org/10.48550/arXiv.2312.02279

[2] 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

[3] Muhammad Alsaiyari and Muhamad Felemban. 2023. Variational quantum algorithms for solving vehicle routing problem. *IEEE Xplore* (02 2023), 1–4. https://doi.org/10.1109/ICSCA57840.2023.10087522

[4] Maribel G Arenas, Juan , Pedro A Castillo, Pablo G Saanchez, Antonio M Mora, Alberto Prieto, and Juan J Merelo. 2010. Statistical analysis of the parameters of the simulated annealing algorithm. *IEEE Congress on Evolutionary Computation* (07 2010). https://doi.org/10.1109/cec.2010.5586160

[5] 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

[6] 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* (2022), 1–10. https://doi.org/10.1109/TITS.2022.3172241

[7] Dantzig G B and Ramser J H. 1959. The truck dispatching problem. *Management Science* 6 (10 1959), 80–91. https://doi.org/10.1287/mnsc.6.1.80

[8] Christopher Bentley, Samuel E Marsh, Andre Carvalho, Philip Kilby, and Michael J Biercuk. 2022. Quantum computing for transport optimization. (06 2022). https://doi.org/10.48550/arxiv.2206.07313

[9] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E O'Brien. 2023. Performance comparison of optimization methods on variational quantum algorithms. *Physical review. A/Physical review, A* 107 (03 2023). https://doi.org/10.1103/physreva.107.032407

[10] 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

[11] M Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J Coles. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3 (09 2021), 625–644. https://doi.org/10.1038/s42254-021-00348-9

[12] Wen-Chyuan Chiang and Robert A Russell. 1996. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research* 63 (02 1996), 3–27. https://doi.org/10.1007/bf02601637

[13] N Christofides and S Eilon. 1969. An algorithm for the vehicle-dispatching problem. *OR* 20 (09 1969), 309. https://doi.org/10.2307/3008733

[14] G Clarke and Wright J W. 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

[15] Jean-François Cordeau, Michel Gendreau, Alain Hertz, Gilbert Laporte, and Jean-Sylvain Sormany. 2005. New heuristics for the vehicle routing problem. *Logistics Systems: Design and Optimization* (12 2005), 279–297. https://doi.org/10.1007/0-387-24977-X_9

[16] J-F Cordeau, M Gendreau, G Laporte, J-Y Potvin, and F Semet. 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53 (05 2002), 512–522. https://doi.org/10.1057/palgrave.jors.2601319

[17] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. 2009. The vehicle routing problem: A taxonomic review. *Computers Industrial Engineering* 57 (11 2009), 1472–1483. https://doi.org/10.1016/j.cie.2009.05.009

[18] Raafat Elshaer and Hadeer Awad. 2020. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers Industrial Engineering* 140 (02 2020), 106242. https://doi.org/10.1016/j.cie.2019.106242

[19] 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

[20] 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

[21] Hirotaka Irie, Goragot Wongpaisarnsin, Masayoshi Terabe, Akira Miki, and Shinichirou Taguchi. 2019. Quantum annealing of vehicle routing problem with time, state and capacity. *Lecture Notes in Computer Science* (01 2019), 145–156. https://doi.org/10.1007/978-3-030-14082-3_13

[22] 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

[23] S Kirkpatrick, Gelatt C D, and Vecchi M P. 1983. Optimization by simulated annealing. *Science* 220 (05 1983), 671–680. https://doi.org/10.1126/science.220.4598.671

[24] Grigorios D Konstantakopoulos, Sotiris P Gayialis, and Evripidis P Kechagias. 2020. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research* (09 2020). https://doi.org/10.1007/s12351-020-00600-7

[25] Gilbert Laporte. 2007. What you should know about the vehicle routing problem. *Naval Research Logistics* 54 (2007), 811–819. https://doi.org/10.1002/nav.20261

[26] Gilbert Laporte, Paolo Toth, and Daniele Vigo. 2013. Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics* 2 (03 2013), 1–4. https://doi.org/10.1007/s13676-013-0020-6

[27] Adam Letchford and Juan-Jose Salazar-Gonzalez. 2018. CVRP instances. *research-data.ull.es* 1 (06 2018). https://doi.org/10.17632/kcc52cw4zs.1

[28] 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

[29] 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

[30] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme. 2018. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology* 3 (06 2018), 030503. https://doi.org/10.1088/2058-9565/aab822

[31] Young-Hyun Oh, Patrick Dreher, Anand Singh, Ibm Yu, and Andy Rindos. 2019. Solving multi-coloring combinatorial optimization problems using hybrid quantum algorithms. https://arxiv.org/pdf/1911.00595.pdf

[32] Lilly Palackal, Benedikt Poggel, Matthias Wulff, Hans Ehm, Jeanette Miriam Lorenz, and Christian B Mendl. 2023. Quantum-assisted solution paths for the capacitated vehicle routing problem. *arXiv.org* (05 2023). https://doi.org/10.48550/arXiv.2304.09629

[33] Aidan Pellow-Jarman, Ilya Sinayskiy, Anban Pillay, and Francesco Petruccione. 2021. A comparison of various classical optimizers for a variational quantum linear solver. *Quantum Information Processing* 20 (06 2021). https://doi.org/10.1007/s11128-021-03140-x

[34] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5 (07 2014). https://doi.org/10.1038/ncomms5213

[35] Tayla Rodgers. 2024. *Benchmarking Classical Methods against Quantum Approximate Optimisation Algorithm to Solve the Vehicle Routing Problem: Branch and Bound*. Honor's project.

[36] 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

[37] Qiskit Algorithms Development Team. 2017. SamplingVQE - Qiskit Algorithms 0.3.0. https://qiskit-community.github.io/qiskit-algorithms/stubs/qiskit_algorithms.SamplingVQE.html

[38] Qiskit Optimization Development Team. 2014. Minimum Eigen Optimizer - Qiskit Optimization 0.6.1. https://qiskit-community.github.io/qiskit-optimization/tutorials/03_minimum_eigen_optimizer.html

[39] Qiskit Optimization Development Team. 2015. Vehicle Routing - Qiskit Optimization 0.6.1. https://qiskit-community.github.io/qiskit-optimization/tutorials/07_examples_vehicle_routing.html

[40] Qiskit Optimization Development Team. 2018. Qiskit Optimization 0.6.1. https://qiskit-community.github.io/qiskit-optimization/index.html

[41] Paolo Toth and Daniele Vigo (Eds.). 2002. *The vehicle routing problem*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718515

[42] 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).

[43] Chao Wang, Dong Mu, Fu Zhao, and John W Sutherland. 2015. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers Industrial Engineering* 83 (05 2015), 111–122. https://doi.org/10.1016/j.cie.2015.02.005

[44] Lijun Wei, Zhenzhen Zhang, Defu Zhang, and Stephen C.H Leung. 2018. A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research* 265 (03 2018), 843–859. https://doi.org/10.1016/j.ejor.2017.08.035

Appendix

$$
\begin{aligned}
\min \quad & \sum_{i \to j} w_{ij} x_{ij} && && (3)\\
\text{s.t.} \quad & \sum_{i \in source[i]} x_{ij} = 1 && \forall i \in \{1, ..., n-1\} && (4)\\
& \sum_{j \in target[i]} x_{ji} = 1 && \forall i \in \{1, ..., n-1\} && (5)\\
& \sum_{j \in source[0]} x_{0j} = k && && (6)\\
& \sum_{j \in target[0]} x_{j0} = k && && (7)
\end{aligned}
$$

Figure 9: Mathematical model of the Vehicle Routing Problem

[6]

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

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

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

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

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

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

Figure 10: Energy Function of the Vehicle Routing Problem

[6]

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

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

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

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

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

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

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

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

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

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

Figure 11: QUBO Formulation of the Vehicle Routing Problem

[6]

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

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

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

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

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

Figure 12: Ising Formulation of the Vehicle Routing Problem

[6]

Algorithm 1 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 4: Table of symbols and descriptions for Simulated Annealing

Algorithm 2 Variational Quantum Eigensolver

Input:
$inputmatrix(ces)$, $circ$, $initialpoint$, $maxiter$
Output:
π^*, $cost^*$
Initialisation;
$qubitOpdocplex \leftarrow$ BuildModel(inputmatrix(ces))
$num \leftarrow$ number of qubits of qubitOpdocplex
$spsa \leftarrow$ SPSA(maxiter)
if $circ$ = RA **then**
    $ry \leftarrow$ RealAmplitudes(num, entanglement=linear)
else if $circ$ = TL **then**
    $ry \leftarrow$ TwoLocal(num, entanglement=linear)
end if
$vqe \leftarrow$ VQE($qubitOpdocplex$, ry, $spsa$, $initialpoint$)
$quantuminstance \leftarrow$ Backend(1024 shots)
$result \leftarrow$ Run(vqe,$quantuminstance$)
π^*, $cost^* \leftarrow$ 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 parameterised 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 5: Table of symbols and descriptions for Variational Quantum Eigensolver

7.1 Additional graphs and tables:

Table 6: List of abbreviations and corresponding full versions used in the result tables.

Full Version	Abbreviation
File	F
Lowest Optimal Cost	L.O.C
Number of Runs with the Lowest Optimal Cost	N.R.L.C
Highest Optimal Cost	H.C.O
Number of Runs with the Highest Optimal Cost	N.R.H.C
Best Time (seconds)	B.T
Average Time (seconds)	A.T
Worst Time (seconds)	W.T
Standard Deviation of Optimal Cost	STD.OC
Standard Deviation of Time	STD.T
Average Optimal Cost	A.O.C
Median Optimal Cost	M.O.C
Median Time	M.T

Table 7: SA100: Complete Results.

F	L.O.C	N.R.L.C	H.C.O	N.R.H.C	B.T	A.T	W.T	STD.OC	STD.T	A.O.C	M.O.C	M.T
vrp-2-1-1	637.0	6	637.0	1	0.059857	0.061245	0.062406	0.0	0.000812	637.0	637.0	0.061243
vrp-3-1-1	1096.0	6	1096.0	1	0.00294	0.003971	0.00508	0.0	0.000779	1096.0	1096.0	0.004134
vrp-3-2-1	1442.0	6	1442.0	1	0.010379	0.011948	0.014176	0.0	0.001124	1442.0	1442.0	0.011822
vrp-4-3-3	2241.0	6	2241.0	1	0.007498	0.009268	0.011212	0.0	0.001269	2241.0	2241.0	0.009281
vrp-5-4-1	2871.0	6	2871.0	1	0.069218	0.071408	0.072306	0.0	0.001051	2871.0	2871.0	0.071791
vrp-4-1-1	1003.0	6	1003.0	1	0.004169	0.005827	0.007562	0.0	0.001308	1003.0	1003.0	0.005739
vrp-6-5-1	1462.0	6	1462.0	1	0.074044	0.077558	0.086182	0.0	0.004579	1462.0	1462.0	0.074818
vrp-4-2-1	1766.0	6	1766.0	1	0.009173	0.011879	0.014548	0.0	0.002069	1766.0	1766.0	0.012305
vrp-7-6-1	2925.0	6	2925.0	1	0.078147	0.088409	0.125722	0.0	0.016978	2925.0	2925.0	0.080449
vrp-8-7-1	3241.0	6	3241.0	1	0.103555	0.105369	0.10704	0.0	0.001186	3241.0	3241.0	0.105708
vrp-5-3-1	1943.0	6	1943.0	1	0.068976	0.080363	0.090101	0.0	0.007489	1943.0	1943.0	0.082679
vrp-5-1-1	1260.0	6	1260.0	1	0.005505	0.008043	0.010893	0.0	0.001616	1260.0	1260.0	0.008034
vrp-10-8-1	4691.0	3	4924.0	1	0.038373	0.08692	0.118302	83.513971	0.028189	4754.5	4718.0	0.097102
vrp-5-2-1	1409.0	6	1409.0	1	0.015955	0.025599	0.040925	0.0	0.008028	1409.0	1409.0	0.025269
vrp-6-4-1	1747.0	6	1747.0	1	0.013016	0.019246	0.040079	0.0	0.009366	1747.0	1747.0	0.015475
vrp-6-1-1	1405.0	6	1405.0	1	0.00835	0.010556	0.015723	0.0	0.002565	1405.0	1405.0	0.009454
vrp-7-5-1	1797.0	6	1797.0	1	0.020302	0.027923	0.052806	0.0	0.011245	1797.0	1797.0	0.023763
vrp-8-6-1	2005.0	6	2005.0	1	0.02705	0.041005	0.072443	0.0	0.015404	2005.0	2005.0	0.034044
vrp-6-2-1	1320.0	6	1320.0	1	0.005901	0.007039	0.008533	0.0	0.000963	1320.0	1320.0	0.006917
vrp-10-7-1	2964.0	6	2964.0	1	0.045794	0.05609	0.065651	0.0	0.00698	2964.0	2964.0	0.055548
vrp-8-5-1	1777.0	6	1777.0	1	0.017726	0.022693	0.029173	0.0	0.003697	1777.0	1777.0	0.021972
vrp-7-4-1	1079.0	1	1289.0	2	0.092396	0.099981	0.103609	78.262379	0.004112	1254.0	1289.0	0.101748
vrp-6-3-1	890.0	6	890.0	1	0.007499	0.009211	0.011797	0.0	0.001302	890.0	890.0	0.008994
vrp-8-1-1	762.0	6	762.0	1	0.028366	0.03846	0.054597	0.0	0.009954	762.0	762.0	0.033927
vrp-8-4-1	915.0	5	950.0	1	0.083654	0.100406	0.123564	13.04373	0.013573	920.833333	915.0	0.094406
vrp-10-6-1	2159.0	1	2178.0	4	0.019844	0.024684	0.034653	7.080882	0.005038	2174.833333	2178.0	0.0229
vrp-8-3-1	1636.0	5	1763.0	1	0.007937	0.013145	0.022668	47.330106	0.004644	1657.166667	1636.0	0.011804
vrp-8-2-1	482.0	3	678.0	2	0.078071	0.090529	0.106755	95.04458	0.011049	576.833333	570.5	0.08757
vrp-10-5-1	1759.0	1	2619.0	1	0.099478	0.104109	0.109819	313.49681	0.003922	2053.5	1896.5	0.103903
vrp-10-1-1	1523.0	1	1580.0	2	0.004711	0.006936	0.010688	20.851992	0.002004	1569.166667	1580.0	0.00669
vrp-10-4-1	1853.0	6	1853.0	1	0.009893	0.014299	0.023284	0.0	0.004526	1853.0	1853.0	0.013311
vrp-10-3-1	1231.0	1	1522.0	1	0.016282	0.038459	0.080092	101.858971	0.023604	1351.5	1355.0	0.029672
vrp-15-8-1	2673.0	6	2673.0	1	0.026705	0.031988	0.045893	0.0	0.006434	2673.0	2673.0	0.029325
vrp-15-7-1	2032.0	1	2545.0	1	0.108579	0.124098	0.133208	151.907867	0.008546	2276.0	2275.5	0.126427
vrp-15-6-1	2224.0	1	2272.0	5	0.010793	0.014515	0.021242	17.888544	0.003556	2264.0	2272.0	0.014241
vrp-15-5-1	2826.0	6	2826.0	1	0.009809	0.014737	0.018576	0.0	0.002983	2826.0	2826.0	0.015797
vrp-15-1-1	725.0	1	915.0	1	0.095395	0.113931	0.12155	72.72723	0.008652	815.5	799.0	0.115936
vrp-15-4-1	2035.0	1	2102.0	2	0.009636	0.016666	0.032212	23.056091	0.007506	2080.5	2087.0	0.015428
vrp-15-2-1	660.0	1	978.0	1	0.072503	0.098181	0.107768	101.707096	0.011709	826.0	821.5	0.102362
vrp-15-3-1	1364.0	1	1690.0	1	0.03619	0.077384	0.124709	136.053503	0.036874	1561.333333	1610.0	0.076887

Table 8: SA1000: Complete Results.

F	L.O.C	N.R.L.C	H.C.O	N.R.H.C	B.T	A.T	W.T	STD.OC	STD.T	A.O.C	M.O.C	M.T
vrp-2-1-1	637.0	6	637.0	1	0.061438	0.063238	0.068118	0.0	0.002272	637.0	637.0	0.062404
vrp-3-1-1	1096.0	6	1096.0	1	0.040535	0.053122	0.079178	0.0	0.012854	1096.0	1096.0	0.049335
vrp-3-2-1	1442.0	6	1442.0	1	0.066848	0.082272	0.102212	0.0	0.011114	1442.0	1442.0	0.082743
vrp-4-3-3	2241.0	6	2241.0	1	0.069733	0.073405	0.077741	0.0	0.002899	2241.0	2241.0	0.073314
vrp-5-4-1	2871.0	6	2871.0	1	0.070146	0.08521	0.115294	0.0	0.015081	2871.0	2871.0	0.078921
vrp-4-1-1	1003.0	6	1003.0	1	0.066914	0.076273	0.085027	0.0	0.007034	1003.0	1003.0	0.078203
vrp-6-5-1	1462.0	6	1462.0	1	0.073214	0.083363	0.094969	0.0	0.008593	1462.0	1462.0	0.083492
vrp-4-2-1	1766.0	6	1766.0	1	0.067275	0.076331	0.086815	0.0	0.009049	1766.0	1766.0	0.075342
vrp-7-6-1	2925.0	6	2925.0	1	0.09826	0.100585	0.102232	0.0	0.001358	2925.0	2925.0	0.100554
vrp-8-7-1	3241.0	6	3241.0	1	0.085661	0.097823	0.107142	0.0	0.0081	3241.0	3241.0	0.101097
vrp-5-3-1	1943.0	5	2007.0	1	0.068682	0.076814	0.088072	23.851392	0.007455	1953.666667	1943.0	0.075935
vrp-5-1-1	1260.0	6	1260.0	1	0.08232	0.0876	0.092638	0.0	0.003938	1260.0	1260.0	0.087934
vrp-10-8-1	4691.0	5	4745.0	1	0.116765	0.130749	0.141723	20.124612	0.008867	4700.0	4691.0	0.132137
vrp-5-2-1	1409.0	6	1409.0	1	0.078591	0.0914	0.113651	0.0	0.011548	1409.0	1409.0	0.091574
vrp-6-4-1	1747.0	6	1747.0	1	0.076451	0.091629	0.120168	0.0	0.014667	1747.0	1747.0	0.089933
vrp-6-1-1	1405.0	6	1405.0	1	0.0717	0.086217	0.095267	0.0	0.008331	1405.0	1405.0	0.087497
vrp-7-5-1	1797.0	6	1797.0	1	0.097865	0.101403	0.105049	0.0	0.002185	1797.0	1797.0	0.10165
vrp-8-6-1	2005.0	6	2005.0	1	0.086332	0.102752	0.108304	0.0	0.007492	2005.0	2005.0	0.105506
vrp-6-2-1	1320.0	6	1320.0	1	0.075767	0.086052	0.099226	0.0	0.009555	1320.0	1320.0	0.082019
vrp-10-7-1	2964.0	6	2964.0	1	0.096089	0.103701	0.116675	0.0	0.00879	2964.0	2964.0	0.098762
vrp-8-5-1	1777.0	6	1777.0	1	0.088634	0.103628	0.13347	0.0	0.015017	1777.0	1777.0	0.101615
vrp-7-4-1	1079.0	2	1289.0	1	0.084036	0.0895	0.09356	74.498322	0.003433	1172.0	1189.5	0.089874
vrp-6-3-1	890.0	6	890.0	1	0.0728	0.080772	0.089179	0.0	0.007399	890.0	890.0	0.080053
vrp-8-1-1	762.0	6	762.0	1	0.097943	0.100662	0.103385	0.0	0.002061	762.0	762.0	0.10058
vrp-8-4-1	876.0	3	1090.0	1	0.094913	0.104363	0.107484	76.345159	0.004411	930.5	895.5	0.106364
vrp-10-6-1	2159.0	5	2178.0	1	0.095921	0.10411	0.114863	7.080882	0.007981	2162.166667	2159.0	0.101967
vrp-8-3-1	1636.0	6	1636.0	1	0.090004	0.109783	0.143469	0.0	0.016222	1636.0	1636.0	0.106081
vrp-8-2-1	482.0	2	588.0	1	0.098338	0.106636	0.133955	38.052595	0.012605	504.0	486.0	0.100524
vrp-10-5-1	1627.0	1	2484.0	1	0.103142	0.111267	0.117249	283.066934	0.005725	1980.333333	1942.0	0.114235
vrp-10-1-1	1523.0	5	1526.0	1	0.083082	0.092386	0.104847	1.118034	0.008823	1523.5	1523.0	0.09035
vrp-10-4-1	1853.0	6	1853.0	1	0.091354	0.102346	0.1141	0.0	0.008913	1853.0	1853.0	0.101787
vrp-10-3-1	1231.0	1	1290.0	1	0.091139	0.099431	0.131092	20.83	0.014198	1245.333333	1235.5	0.093754
vrp-15-8-1	2639.0	2	2673.0	2	0.114852	0.116334	0.119464	14.988885	0.001645	2660.0	2668.0	0.115512
vrp-15-7-1	2010.0	1	2539.0	1	0.108263	0.112213	0.114675	189.718578	0.002132	2263.166667	2198.0	0.112558
vrp-15-6-1	2223.0	2	2272.0	1	0.110422	0.11226	0.113715	17.669811	0.001053	2238.333333	2234.0	0.112227
vrp-15-5-1	2803.0	2	2826.0	2	0.108539	0.112272	0.115143	9.788031	0.002236	2813.833333	2812.5	0.112102
vrp-15-1-1	694.0	1	874.0	1	0.092015	0.094072	0.095256	70.74465	0.00102	773.833333	752.0	0.094237
vrp-15-4-1	1848.0	1	1917.0	1	0.10295	0.106286	0.108732	23.704782	0.001925	1876.5	1878.0	0.106345
vrp-15-2-1	629.0	1	1015.0	1	0.099294	0.101552	0.104831	137.803806	0.001661	843.666667	854.0	0.101253
vrp-15-3-1	1392.0	1	1660.0	1	0.098051	0.103104	0.10923	91.721953	0.003596	1545.5	1544.5	0.102771

Table 9: B&B: Complete Results from [35]

F	L.O.C	N.R.L.C	H.C.O	N.R.H.C	B.T	A.T	W.T	STD.OC	STD.T	A.O.C	M.O.C	M.T
vrp-2-1-1	637.0	6	637.0	1	2.8e-05	3.2e-05	3.7e-05	0.0	2e-06	637.0	637.0	3.2e-05
vrp-3-1-1	1096.0	6	1096.0	1	5e-05	6e-05	6.7e-05	0.0	7e-06	1096.0	1096.0	6.4e-05
vrp-3-2-1	1442.0	6	1442.0	1	0.000108	0.000121	0.000145	0.0	1.6e-05	1442.0	1442.0	0.00011
vrp-4-3-3	2241.0	6	2241.0	1	0.000646	0.000665	0.000699	0.0	1.7e-05	2241.0	2241.0	0.00066
vrp-5-4-1	2871.0	6	2871.0	1	0.005133	0.00593	0.007068	0.0	0.00081	2871.0	2871.0	0.005703
vrp-4-1-1	1003.0	6	1003.0	1	0.000129	0.000172	0.000261	0.0	4.4e-05	1003.0	1003.0	0.000164
vrp-6-5-1	1462.0	6	1462.0	1	0.043174	0.057502	0.079944	0.0	0.011368	1462.0	1462.0	0.054857
vrp-4-2-1	1766.0	6	1766.0	1	0.000416	0.000498	0.000581	0.0	8e-05	1766.0	1766.0	0.000496
vrp-7-6-1	2925.0	6	2925.0	1	0.504105	0.524886	0.56284	0.0	0.019458	2925.0	2925.0	0.521142
vrp-8-7-1	3241.0	6	3241.0	1	4.777039	5.30178	5.777327	0.0	0.405692	3241.0	3241.0	5.360137
vrp-5-3-1	1943.0	6	1943.0	1	0.005927	0.005993	0.006049	0.0	3.7e-05	1943.0	1943.0	0.005998
vrp-5-1-1	1260.0	6	1260.0	1	0.000865	0.000876	0.00089	0.0	8e-06	1260.0	1260.0	0.000875
vrp-10-8-1	4691.0	6	4691.0	1	1340.280738	1374.614076	1394.311367	0.0	17.794824	4691.0	4691.0	1377.039436
vrp-5-2-1	1409.0	6	1409.0	1	0.002525	0.002538	0.002557	0.0	1.3e-05	1409.0	1409.0	0.002532
vrp-6-4-1	1747.0	6	1747.0	1	0.05055	0.051211	0.052069	0.0	0.000608	1747.0	1747.0	0.05118
vrp-6-1-1	1405.0	6	1405.0	1	0.005714	0.005942	0.006129	0.0	0.000134	1405.0	1405.0	0.005931
vrp-7-5-1	1797.0	6	1797.0	1	0.559252	0.566002	0.574091	0.0	0.005649	1797.0	1797.0	0.565279
vrp-8-6-1	2005.0	6	2005.0	1	7.192865	7.359442	7.498744	0.0	0.103419	2005.0	2005.0	7.373557
vrp-6-2-1	1320.0	6	1320.0	1	0.012326	0.013876	0.014244	0.0	0.000695	1320.0	1320.0	0.014174
vrp-10-7-1	2964.0	6	2964.0	1	1606.999897	1653.38202	1678.100694	0.0	22.819985	2964.0	2964.0	1656.99855
vrp-8-5-1	1777.0	6	1777.0	1	6.326951	6.394035	6.478947	0.0	0.046344	1777.0	1777.0	6.385606
vrp-7-4-1	1079.0	6	1079.0	1	0.358979	0.364664	0.369137	0.0	0.003524	1079.0	1079.0	0.365581
vrp-6-3-1	890.0	6	890.0	1	0.030886	0.0312	0.031564	0.0	0.00022	890.0	890.0	0.031176
vrp-8-1-1	762.0	6	762.0	1	0.022665	0.023606	0.024601	0.0	0.00059	762.0	762.0	0.023617
vrp-8-4-1	876.0	6	876.0	1	2.553252	2.592381	2.654794	0.0	0.035348	876.0	876.0	2.578397
vrp-10-6-1	2159.0	6	2159.0	1	1247.94656	1267.361367	1291.401167	0.0	15.237008	2159.0	2159.0	1269.323882
vrp-8-3-1	1636.0	6	1636.0	1	2.005327	2.024722	2.033233	0.0	0.009757	1636.0	1636.0	2.028357
vrp-8-2-1	482.0	6	482.0	1	0.117696	0.11997	0.120933	0.0	0.001099	482.0	482.0	0.120464
vrp-10-5-1	1609.0	6	1609.0	1	476.584875	484.846432	493.766686	0.0	7.21826	1609.0	1609.0	484.48651
vrp-10-1-1	1523.0	6	1523.0	1	1229.297308	1241.610846	1248.376827	0.0	6.520491	1523.0	1523.0	1244.450076
vrp-10-4-1	1853.0	6	1853.0	1	653.847146	662.675332	671.461679	0.0	5.537529	1853.0	1853.0	662.980953
vrp-10-3-1	1179.0	6	1179.0	1	44.323531	45.289391	46.034607	0.0	0.585469	1179.0	1179.0	45.326722
vrp-15-8-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-15-7-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-15-6-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-15-5-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-15-1-1	564.0	6	564.0	1	3392.724981	3426.624613	3480.424556	0.0	29.922919	564.0	564.0	3420.691974
vrp-15-4-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-15-2-1	T	T	T	T	T	T	T	T	T	T	T	T
vrp-15-3-1	T	T	T	T	T	T	T	T	T	T	T	T

Table 10: VQE COBLYA: Complete Results.

F	L.O.C	N.R.L.C	H.C.O	N.R.H.C	B.T	A.T	W.T	STD.OC	STD.T	A.O.C	M.O.C	M.T
vrp-2-1-1	1036.0	6	1036.0	1	3.115037	3.361962	3.512469	0.0	0.124162	1036.0	1036.0	3.386071
vrp-3-1-1	1190.0	6	1190.0	1	115.962761	119.490761	121.807448	0.0	2.235359	1190.0	1190.0	120.098696
vrp-3-2-1	1442.0	6	1442.0	1	113.819546	117.40928	120.466456	0.0	2.310664	1442.0	1442.0	117.156381
vrp-4-3-3	T	T	T	T	T	T	T	T	T	T	T	T
vrp-5-4-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-4-1-1	T	T	T	T	T	T	T	T	T	T	T	T
vrp-6-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-4-2-1	T	T	T	T	T	T	T	T	T	T	T	T
vrp-7-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-5-3-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-5-1-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-10-8-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-5-2-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-6-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-7-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-7-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-8-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL

Table 11: VQE SPSA: Complete Results.

F	L.O.C	N.R.L.C	H.C.O	N.R.H.C	B.T	A.T	W.T	STD.OC	STD.T	A.O.C	M.O.C	M.T
vrp-2-1-1	637.0	6	637.0	1	1.239295	1.356431	1.433478	0.0	0.072004	637.0	637.0	1.367474
vrp-3-1-1	1096.0	6	1096.0	1	36.612248	37.477671	38.622316	0.0	0.845221	1096.0	1096.0	37.378751
vrp-3-2-1	1442.0	6	1442.0	1	33.205623	35.350031	36.731278	0.0	1.297137	1442.0	1442.0	35.52447
vrp-4-3-3	2407.0	6	2407.0	1	11483.24374	11867.531866	12106.143711	0.0	236.187902	2407.0	2407.0	11952.754974
vrp-5-4-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-4-1-1	1003.0	6	1003.0	1	11631.92455	12003.012198	12231.61737	0.0	243.843503	1003.0	1003.0	12113.985547
vrp-6-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-4-2-1	1580.0	6	1580.0	1	11456.304434	11889.223621	12084.278247	0.0	233.419808	1580.0	1580.0	12014.180126
vrp-7-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-5-3-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-5-1-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-10-8-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-5-2-1	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
vrp-6-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-7-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-7-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-6-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-8-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-10-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-8-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-7-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-6-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-5-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-1-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-4-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-2-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL
vrp-15-3-1	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL	QL

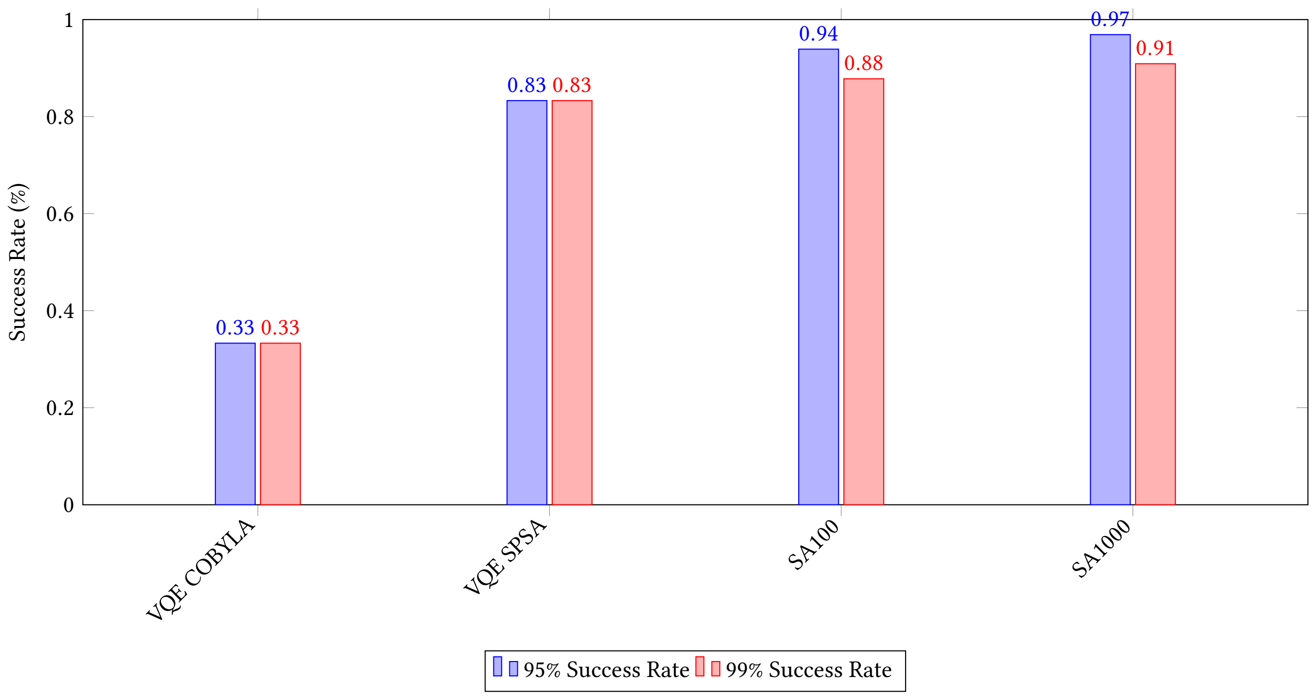

Figure 13: Success Rates at 95% and 99% levels for each algorithm.