PORTFOLIO OPTIMIZATION OF ASSETS USING THE VARIATIONAL QUANTUM EIGENSOLVER (VQE) VS THE CLASSICAL APPROACH (ELEN4022 GROUP PROJECT)

Group 2: Tristan Lilford (1843691) and Robin Jonker (1827572)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: The design and implementation of different portfolio optimization methods is reported. The computation methods will be explained, simulated and compared. A discussion will review and clarify the results to draw conclusions on the different methods.

Key words: Hamiltonian, Ansatz, Eigenvalues

1. INTRODUCTION

Portfolio optimization is the process of determining the best investment options, so that an investor can minimize risk whilst maximizing possible returns. This report will first explain the theory behind portfolio optimization. Then an investigation into the different computational methods, namely classical eigensolvers and VQE, will be conducted. The design and implementation of these portfolio optimization methods will be completed. The designs will be simulated to demonstrate the accuracy, performance and scalability of each method. A discussion will explain why the specific results were produced. Lastly, a conclusion will summarize the key findings of these different portfolio optimization methods.

2. BACKGROUND THEORY

2.1 Mean Variance Portfolio Optimization

Mean variance portfolio optimization is a tool which helps investors make the most suitable investment choices, based on their risk portfolio. It does this by weighing asset risk, expressed as variance, against asset returns [1]. In theory higher risk assets, such as equities, often yield higher returns. This relationship can be seen in Appendix A *Figure 1* . Mean variance portfolio theory however aims to reduce the variance whilst maintaining higher returns. It does through using diversification [2]. By investing diversely in uncorrelated assets the Markowitz Model [3] is found. This can be seen in Appendix A *Figure 2*. By determining asset returns and the relevant covariance matrices, portfolio optimization can be expressed and solved mathematically, seen in equations 1 and 2 below.

$$\min_{x\in\{0,1\}^n} qx^T\Sigma x - \mu^T x[4] \tag{1}$$

$$\text{subject to: } 1^T x = B \tag{2}$$

- $\min_{x\in\{0,1\}^n}$: binary decision variables
- $\mu \in R^n$: expected asset returns
- $\Sigma \in R^{n\times n}$: covariance between assets
- $\min_{x\in\{0,1\}^n}$: investor risk factor
- B: number of assets to be invested in

2.1.1 Period Return Mean():

μ or period return mean is the average return from an asset's appreciation over a period of time [5]. This refers to the income that can be made from investing into an asset.

2.1.2 Covariance Matrix:

covariance, with regards to the stock market, is a measure of how two random assets jointly vary [6]. It measures how these assets change with regards to one another. If under the same conditions both stock values increase and decrease, there is positive covariance. This would indicate that the assets correlate to one another and this would hamper diversification. Ideally to achieve the Markowitz Model in section 2.1 the assets should be uncorrelated to maximise on diversification. This would yield the highest chance for investment success. To gain a comprehensive view of all this information a covariance matrix, can be created. The formula for developing this matrix can be seen in equation 3 below.

$$M = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T \tag{3}$$

All the period return means and covariances of the assets would need to be calculated to optimise an investment portfolio. Considering a stock market such as the Nasdaq which had 3,626 companies listed in March 2022 [7], the amount of calculations and processing time required to analyse such data is exuberant. These calculations however can be automated and this is where quantum computing and VQE become extremely relevant, see section 2.3. With this data acquired equation 1 could be solved determining the best possible binary decision variable to invest in.

2.2 Classical Computing Model

The process of optimizing a portfolio as described above results in whats known as a minimum variance portfolio which is described using equation 4 [8]. With the above information an eigen portfolio can be developed. The covariance matrix's eigen values give the variance/risk of each factor and the eigen vectors give the strategy weight allocations. The minimum variance problem can be solved with an eigensolver. More industry relevant model focus on using machine learning and AI to determine an optimal portfolio [9].

$$\min_{\mu} \mu^T \Sigma M \tag{4}$$

2.2.1 Classical Minimum Eigensolver A minimum eigensolver works by analysing the feasible states and returning the eigenstate that has the smallest eigenvalue [10]. This means that every possible state would have to be analysed. Performance of this technique is seen in section 4.1.

2.3 Quantum Computing Model

Quantum computers thrive on problems which include complexity, uncertainty and large data pools. For portfolio optimization there are two current algorithms which are heavily studied, variational quantum eigensolver (VQE) and the quantum approximate optimization algorithm (QAOA).

2.3.1 Variational Quantum Eigensolver(VQE): The VQE algorithm is a classical/quantum hybrid. It is used to determine the eigen values of a large Hamiltonian matrix, H , to find the ground state energy [11]. The process of finding this lowest energy state is known as quantum annealing. For portfolio optimisation a Hamiltonian can be developed using portfolio data and the minimum ground energy state corresponds to the minimum variance portfolio, see section 3.3. This algorithm has 2 fundamental steps. Firstly a trial wave function known as an ansatz [12] must be designed with adjustable parameters, $|\Psi(\vec{\theta})\rangle$. This ansatz is a reasonable assumption about the form of the solution target wave function. This ansatz is then realised as a quantum circuit [13]. This ansatz is the used to measure the expectation value, $\langle \Psi(\vec{\theta}) \mid H \mid \Psi(\vec{\theta}) \rangle$ (quantum processing). The measured expectation value will always be greater than the smallest eigen value of the Matrix due to the variational principle. Then this process is iterated through an optimisation loop (classical processing), where the parameters of the ansatz are then variationally adjusted until the expectation value of the electronic Hamiltonian is minimized. This occurs when measured value converges to the minimum eigen value.

This process is heavily dependant on the initial ansatz design. A more accurate design will ensure that optimised result is found through the least amount of iterations. The number of qubits of the design represents the number of assets being analysed. The circuit is fully entangled with CZ Gates [14] and RY gates [15] see appendix B *Figure 3* and *Figure 4.* Linear entanglement can also be done, however this greatly reduces accuracy as the number of qubits increase. Another form of portfolio optimisation is using QAOA.

2.3.2 Quantum Approximate Optimization Algorithm (QAOA): QAOA is also a classical/quantum hybrid algorithm similar to that of the VQE. It is used to solve combinatorial problems such as portfolio optimisation. This algorithm defines a mixing operator that explores all 2^n combinatorial solutions and attempts to find the closest approximate solution. This section however will not be studied as it provides inaccurate results for NISQ machines with low depth. More can be read from portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz [16].

3. IMPLEMENTATION AND DESIGN OF PORTFOLIO OPTIMIZATION MODELS

3.1 Data acquisition

3.1.1 Data input: For the purpose of this report stock data was imported using yahoo finance. 'TSLA', 'ABNB', 'FB', 'GOOGL' tickers were chosen as the assets to demonstrated portfolio optimization. Adjusted closing price [17] stock data was collected between 2021-01-01 to 2022-01-01. This data is displayed in Appendix C *Figure 5* and the summary table can be seen in *Figure 6.*

With this data acquired the return mean, μ, can be determined. This data is display in Appendix C *Figure 7.* From this point the covariance matrix can be generated automatically using these returns, this determines how correlated the stocks are. With this information collected equation 1 can be solved. The best method used in order to automate solving this equation is to develop a quadratic unconstrained binary optimization (QUBO) problem using the input data.

3.1.2 Quadratic Program: Qiskit offers very useful libraries in order to convert an investors portfolio into a QUBO program, specifically the "qiskit_finance.applications.optimization" [18]. Firstly the investors risk factor and budget must be noted. Then the asset period return means and covariances must be stored. With this data collected a (QUBO) problem can be developed. The minimum solution to this binary problem will be the optimized binary decision variables [19]. In order to solve a QUBO problem one can use classical computing methods such as an eigensolver or quantum computing methods such as a variational quantum eigensolver (VQE).

3.2 Classical Eigensolver

For the purpose of processing the quadratic program created above the inbuilt numpy minimum eigensolver was implored. After the eigen solver iterates through all the possible best combinations it will determine which combination has the lowest variance. This result will be explored in section 4.2.

3.3 Variational Quantum Eigensolver(VQE)

The input to the VQE method is a quadratic program as mentioned in section 3.1.2. This program is converted using the qiskit "Ising operator into a Pauli

Sum Operator method". The Pauli Sum operator is then converted into a Hamiltonian that is required in the VQE process. The Hamiltonian is then transformed into a dictionary holding the Pauli basis multiples. This dictionary will be used to optimize the adjustable parameters of the variational ansatz.

The creation of a variational ansatz form is the most important part of the process. In the implementation of the VQE, a fully entangled ansatz circuit was designed with parameterised y-rotation gates. It was implemented to have 3 repetitions of this design, as can be seen in Appendix D *Figure 9*

The factor that alters these parameters through every iteration is dependant on the dictionary values based on the Hamiltonian as discussed above. This value is a summation of the product of the Hamiltonian coefficients and the expectation value of the supplied observable. Finding the supplied observable is dependant on the particular basis list. This list is a combination of Identity and PauliZ matrices for each parameter (angle for the y-rotation gates) within the circuit. .

The creation of the ansatz is initially made with a random numbers as the parameters of the y-rotation angles. This circuit is then processed within an optimization loop. After each iteration the parameters recursively update to try and get the circuit output closer to the minimum eigen value. The parameters will begin to converge to the final rotational angles.

In the implementation of the designed VQE, the Gradient Descent Optimizer was used. The goal is that at the end of the multiple iterations, the minimum expectation value is found.

Similarly to the classical Numpy Eigensolver method above, the result for the optimal selection of stocks is found. The results is the stock choices with the highest expected returns and minimal the financial risk. An example of this result is shown in Appendix D *Figure 17*

3.4 Custom ansatz circuit

AS stated in section 2.3.1 the design of the ansatz can make a great difference in processing time. A general function was made to generate an ansatz circuit for any number of possible portfolio assets. This circuit gets expanded and was designed to have 3 repetitions of y-rotation gates and entanglement gates. The variational parameter used within the y-rotation gates was implemented as theta, as shown in *Figure 9*.

3.5 Simulated Models

For the purpose of the report the following simulations were created: A basic numpy eigensolver , a general Qiskit VQE, A custom ansatz Run Qiskit VQE and a Custom designed VQE. The VQE methods operated using a statevector simulator. This is due to the fact that a varaitional quantum circuit needs to be iterated multiple times whilst being optimized through classical methods. As such the processing time of the VQE methods cannot be directly compared to that of a classical machine however they can be compared to one another. For the purpose of comparing classical machines to quantum machines the research paper, Portfolio Optimization of 40 Stocks Using DWaves Quantum Annealer will be briefly reviewed[https://arxiv.org/pdf/2007.01430.pdf].

4. ANALYSIS AND DISCUSSION OF RESULTS

4.1 Performance

The performance of the respective methods is shown in Table 1. Appendix D *Figure 16* shows the execution times of multiple iterations, this data was then averaged to produce the information found in Table 1. As seen in this table, the classical model is extremely fast this is because only four assets were investigated within this portfolio optimisation problem. The built-in VQE method is the longest method due to the recursion that occurs in finding the ideal y rotation gate angle for each repetition on each qubit. The custom-ansatz method is relatively fast as there is no need for the VQE function to build the required circuit as there is a customly designed circuit inputted into the function. The full VQE is the fastest method when there is no recursion occurring in finding its y rotation angle parameters however when applying an appropriate amount of recursion, the execution time become extreme slow as seen in Appendix E Table 5.

Table 1: Execution times of methods for 4 assets.

Method	Time (s)
Classical eigen solver	0.0164
Built-in VQE	5.0478
Custom-ansatz VQE	0.6419
Full VQE	0.5015

4.2 Accuracy

The accuracy of the respective methods is shown in Table 2. The information of Table 2 comes from *Figure 11* to *Figure 14* in Appendix D. A common amount of assets is used as to have a general outcome for each of them. This table shows that the classical approach is flawless whereas the custom-ansatz VQE method is slightly more accurate than the built-in VQE method. The full VQE method is extremely erroneous, this is likely due to errors within finding the parameters needed for the y-rotation angles within the gates used in circuit. This error rate refers to the probability that the quantum hardware will result in the right solution. Each method always finds the optimal solution, as long as the error is not 100%. This is shown in the example of results that is printed in the console in Appendix D *Figure 17*. For the example where 'TSLA', 'ABNB', 'FB', 'GOOGL'is inputted into the program, every method gave the same optimal result, being that you should invest in 'TSLA' and 'GOOGL' (0011), at different possibilities, shown in Table 2.

Table 2: Error percentage of methods for 4 assets

Method	**Error (%)**
Classical eigen solver	0
Built-in VQE	68
Custom-ansatz VQE	67
Full VQE	96

4.3 Scalability

The scalability of the different methods was shown and studied in Tables 3 and 4. Table 3 shows the error percentage comparison between the built-in VQE method and our custom-ansatz circuit VQE method over a differing amounts of assets. For assets less than 4, both methods are fairly accurate however as the amount of assets increase the custom-ansatz circuit starts giving random error percentages until it eventually breaks and gives 100% error rates. This is likely due to the circuit being customly built with regard to low amount of asset numbers. The built-in VQE method had a relatively small linear error rate change for the increasing number of assets. Table 5 shows the execution time of the different methods depending on the amount of assets. The only notable change is within the full VQE method where the rate of change of performance is far bigger compared to that of the other methods.

Table 3: Error percentage of methods over different asset numbers

Assets	**Built-in (%)**	**Custom-ansatz (%)**
2	0	20
3	0	0
4	68	67
5	71	0
6	79	100
7	70	100

Table 4: Execution time (s) of the methods over different amount of assets

Assets Number	**4**	**5**	**6**	**7**
Classical	0.0164	0.0199	0.0249	0.0329
Built-in	5.0478	5.9513	7.6802	8.9727
Custom-ansatz	0.6419	0.7178	0.7924	0.8147
Full VQE	0.5015	0.8197	1.1990	1.8061

4.4 Classical vs Quantum Approaches

DWaves 2,041 qubit quantum annealer was used to analyse a portfolio consisting of 40 stocks. This number of stocks would create a possible solution space size of 2^{40} or 1.1 trillion possible portfolio combinations. This project was separated into different solution space sizes and various methods such as the classical Monte Carlo Method and the quantum genetic algorithm. The results are seen in Appendix E. DWave outperforms classical random sampling on average, for all portfolios n 3 to 25, 27, 28, 33, The report went on later to state, "At lower asset levels there are efficient classical algorithms that do better"[20].

4.5 Summary

The performance, accuracy, and scalability of the different methods were analysed. When comparing Table 1 and 2 it can be seen that the classical approach for portfolio optimization at the moment is fast and more accurate solution to use. When comparing the different VQE methods, the full VQE method was the fastest but also the most prone to errors. Under the current design, the full VQE method is not a viable option to use as the error percentage is too high. The built-in VQE method however works relatively accurately and it is assumed that for a higher number of assets it would outperform the classical methods if run on quantum hardware.

5. CONCLUSION

Portfolio optimization is key in ensuring individuals get the highest returns for their money while minimizing risk. Finding programs that can accurately optimize a portfolio is of great interest within the financial realm. Quantum computing's ability to handle more data in parallel than classical computing is a key attribute that makes its future use vital. Quantum computing has the potential to be able to manage a portfolio with numerous amounts of assets with interconnecting dependencies. However, the technology is not at a high enough or reliable enough level today. When managing someone's money, erroneous recommendations can have dire effects. As seen with all the VQE methods discussed above, as the number of assets increase, the error percentage also increases. Even with a high percentage of error, an optimal selection can be made. At the current stage, using the quantum method is best in combination with a classical approach for optimization. Quantum portfolio optimisation still needs a great deal of research and development before it will ever become commercially useful, however it is a very promising idea as shown by Dwave.

6. References

[1] SmartAsset. 2022. How Mean-Variance Optimization Works in Investing - SmartAsset. [ONLINE] Available at: https://smartasset.com/financial-advisor/mean-variance-optimization: :text=A%20mean-variance%20analysis%20is,reward%20based%20on%20its%20risk. [Accessed 09 May 2022].

[2] What Is Portfolio Diversification? - Fidelity. 2022. What Is Portfolio Diversification? - Fidelity. [ONLINE] Available at: https://www.fidelity.com/learning-center/investment-products/mutual-funds/diversification. [Accessed 09 May 2022].

[3] QuantPedia. 2022. Markowitz Model - QuantPedia. [ONLINE] Available at: https://quantpedia.com/markowitz- model. [Accessed 09 May 2022].

[4] Portfolio Optimization — Qiskit Finance 0.3.1 documentation. 2022. Portfolio Optimization — Qiskit Finance 0.3.1 documentation. [ONLINE] Available at: https://qiskit.org/documentation/finance/tutorials/01_portfolio_optimization.html. [Accessed 09 May 2022].

[5] study.com. 2022. No page title. [ONLINE] Available at: https://study.com/academy/lesson/holding-period-return-hpr-formula-examples.html: :text=The%20holding0period0return0is,))0F0original0value)0*0100 [Accessed 09 May 2022].

[6] Understanding the Covariance Matrix — DataScience+. 2022. Understanding the Covariance Matrix — DataScience+. [ONLINE] Available at: https://datascienceplus.com/understanding-the-covariance- matrix/. [Accessed 09 May 2022].

[7]Matthew Frankel, CFP. 2022. What is the Nasdaq Composite Index? — The Motley Fool . [ONLINE] Available at: https://www.fool.com/investing/stock-market/indexes/nasdaq/: :text=How%20many%20companies%20are%20in,in%20the%20Nasdaq%20Composite%20index.[Accessed 09 May 2022].

[8] Scott Rome. 2022. Eigen-vesting II. Optimize Your Portfolio With Optimization – Scott Rome – Math Ph.D. who works in Machine Learning. [ONLINE] Available at: https://srome.github.io/Eigenvesting-II-Optimize-Your-Portfolio-With-Optimization/. [Accessed 09 May 2022].

[9] Ban, G.-Y., El Karoui, N. and Lim, A.E.B. (2018). Machine Learning and Portfolio Optimization. Management Science, [online] 64(3), pp.1136–1154. doi:10.1287/mnsc.2016.2644.

[10] NumPyMinimumEigensolver — Qiskit 0.36.1 documentation. 2022. NumPyMinimumEigensolver — Qiskit 0.36.1 documentation. [ONLINE] Available at: https://qiskit.org/documentation/stubs/qiskit.algorithms.NumPyMinimumEigensolver.html: :text = The%20minimum%20eigensolver%20is%20only,consider%20this%20value%20or%2 [Accessed 09 May 2022].

[11] Variational-Quantum-Eigensolver (VQE) — Grove 1.7.0 documentation. 2022. Variational-Quantum-Eigensolver (VQE) — Grove 1.7.0 documentation. [ONLINE] Available at: https://grove-docs.readthedocs.io/en/latest/vqe.html. [Accessed 09 May 2022].

[12] Merriam-webster.com. (2015). Merriam-Webster Dictionary. [online] Available at: https://www.merriam-webster.com/dictionary/ansatz [Accessed 9 May 2022].

[13] Fedorov, D.A., Peng, B., Govind, N. and Alexeev, Y. (2022). VQE method: a short survey and recent developments. Materials Theory, 6(1). doi:10.1186/s41313-021-00032-6.

[14] Qiskit.org. (2021). CZGate — Qiskit 0.36.1 documentation. [online] Available at: https://qiskit.org/documentation/stubs/qiskit.circuit.library.CZGate.html [Accessed 9 May 2022].

[15] Qiskit.org. (2021). RYGate — Qiskit 0.36.1 documentation. [online] Available at: https://qiskit.org/documentation/stubs/qiskit.circuit.library.RYGate.html [Accessed 9 May 2022].

[16] Hodson, M., Ruck, B., Chuan, H., Ong, Garvin, D., Dulman, S. and Computing, R. (n.d.). Portfolio rebalancing experiments using the Quantum Alternating Operator Ansatz. [online] Available at: https://arxiv.org/pdf/1911.05296.pdf [Accessed 9 May 2022].

[17] Investopedia. (2022). How to Calculate a Stock's Adjusted Closing Price. [online] Available at: https://www.investopedia.com/ask/answers/06/adjustedclosingprice.asp [Accessed 9 May 2022].

[18] Qiskit.org. (2018). qiskit_finance.applications.optimization.portfolio_optimization — Qiskit Finance 0.3.1 documentation. [online] Available at: https://qiskit.org/documentation/finance/_modules/qiskit_finance/applications/optimization/portfolio_optimization.html [Accessed 9 May 2022].

[19] Qiskit.org. (2014). Minimum Eigen Optimizer — Qiskit Optimization 0.3.2 documentation. [online] Available at: https://qiskit.org/documentation/optimization/tutorials/03_minimum_eigen_optimizer.html [Accessed 9 May 2022].

[20] Cohen, J., Khan, A., Alexander, C. (2020). Portfolio Optimization of 40 Stocks Using DWaves Quantum Annealer. [online] Available at: https://arxiv.org/pdf/2007.01430.pdf [Accessed 9 May 2022].

Appendices

A Appendix

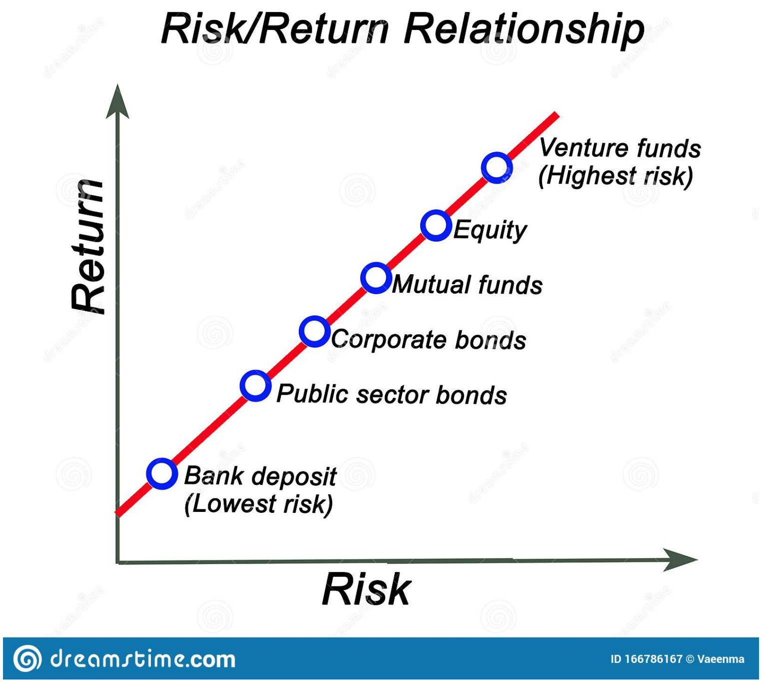


Figure 1: Return vs Risk[https://www.dreamstime.com/relationship-risk-return-investitions-image166786167]

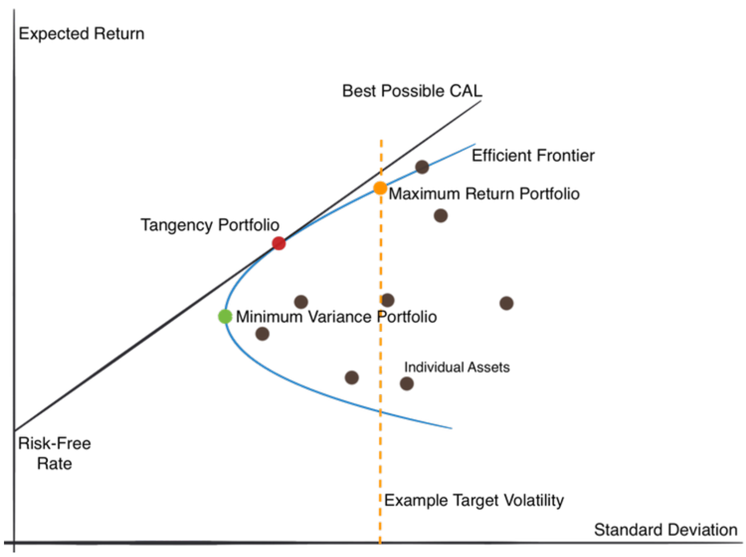


Figure 2: Markowitz Model[3]

B Appendix

Circuit symbol:

```
q_0: ─■─
      │
q_1: ─■─
```

Matrix representation:

$$CZ\ q_1, q_0 = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Figure 3: CZ Gate

Circuit symbol:

```
     ┌───────┐
q_0: ┤ Ry(θ) ├
     └───────┘
```

Matrix Representation:

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

Figure 4: RY gate

C Appendix

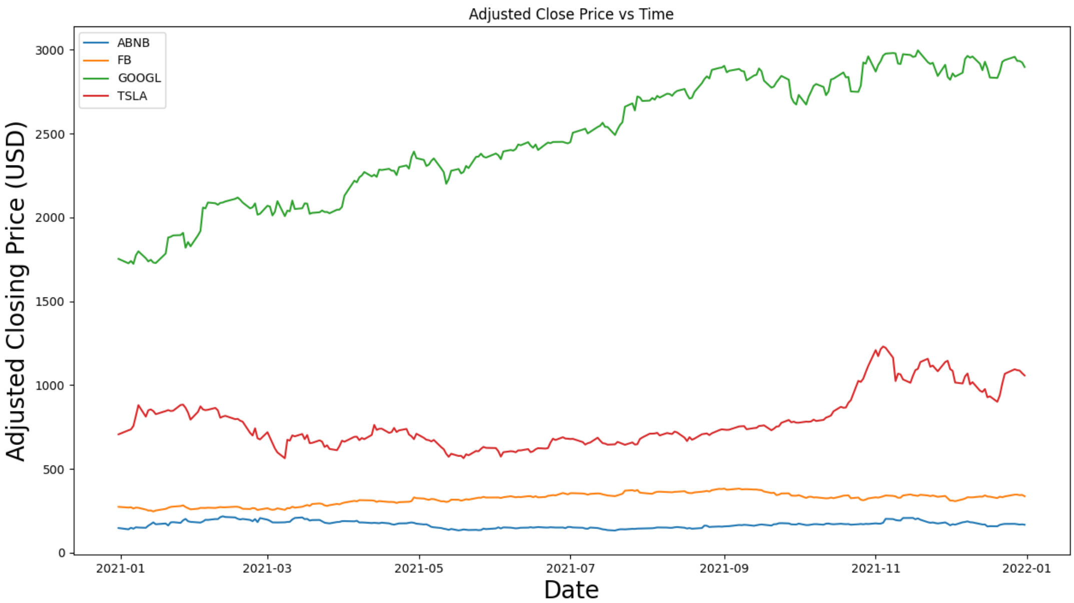


Figure 5: Adjusted closing price of the different assets

	ABNB	FB	GOOGL	TSLA
Date				
2020-12-31	146.800003	273.160004	1752.640015	705.669983
2021-01-04	139.149994	268.940002	1726.130005	729.770020
2021-01-05	148.300003	270.970001	1740.050049	735.109985
2021-01-06	142.770004	263.309998	1722.880005	755.979980
2021-01-07	151.270004	268.739990	1774.339966	816.039978
...	...	...	...	...
2021-12-27	171.679993	346.179993	2958.129883	1093.939941
2021-12-28	169.710007	346.220001	2933.739990	1088.469971
2021-12-29	167.440002	342.940002	2933.100098	1086.189941
2021-12-30	168.779999	344.359985	2924.010010	1070.339966
2021-12-31	166.490005	336.350006	2897.040039	1056.780029

Figure 6: Stock data of the different assets

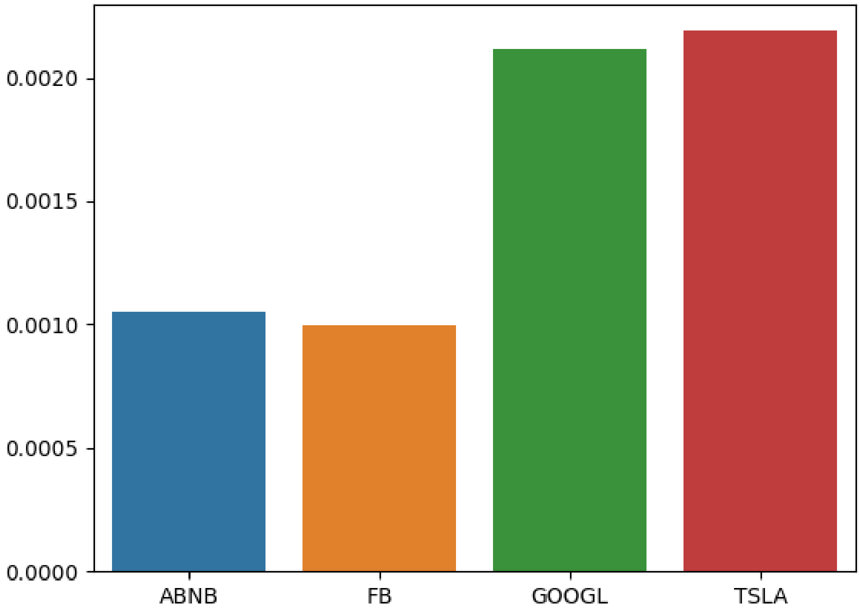


Figure 7: Mean Returns of the different assets

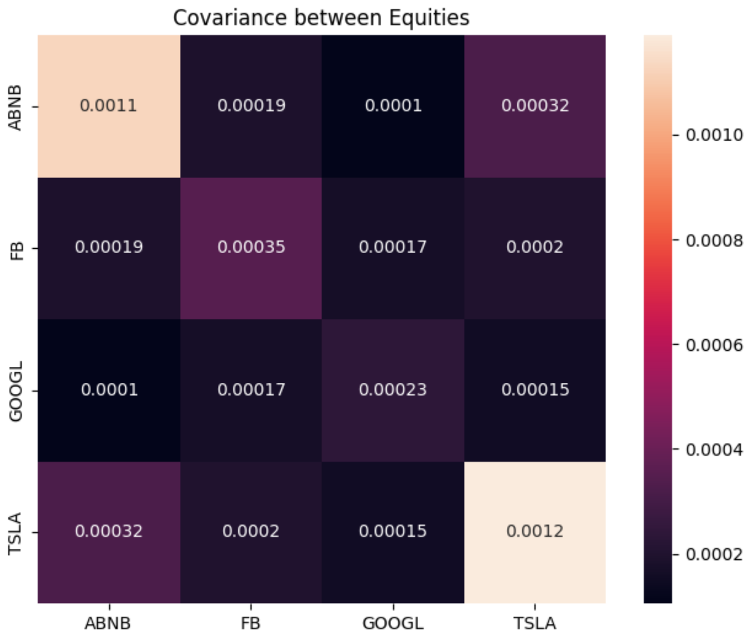


Figure 8: Covariance between equities

D APPENDIX

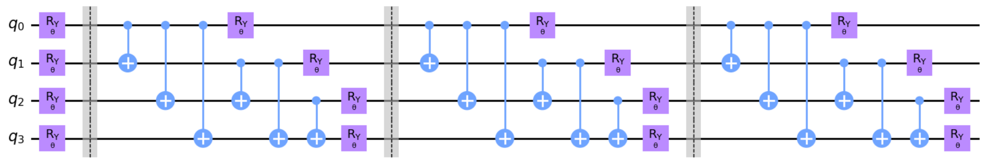


Figure 9: Ansatz circuit with theta parameters for y-rotational angles

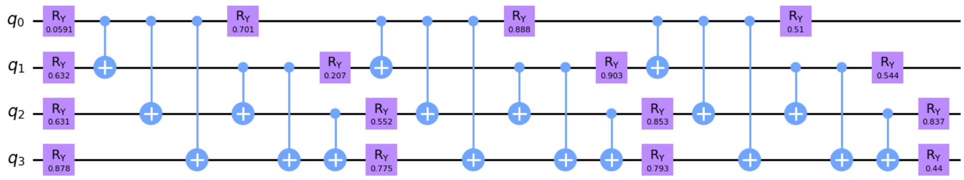


Figure 10: Ansatz circuit with example of parameters for y-rotational angles from full VQE recursions

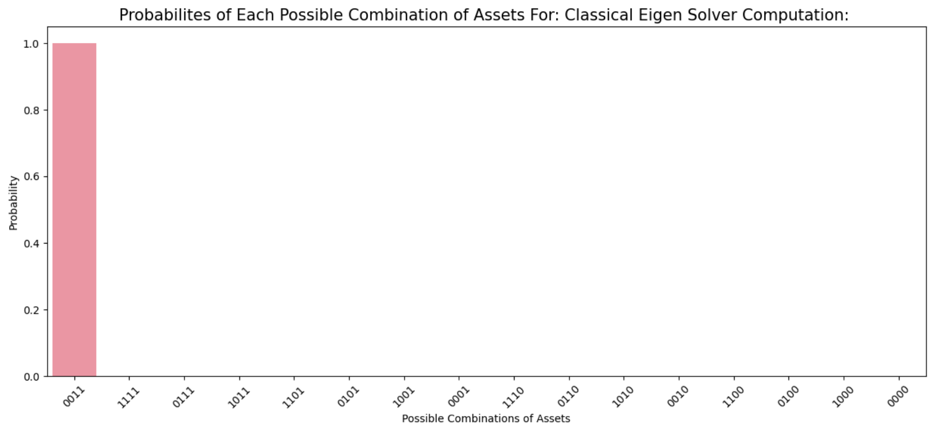


Figure 11: Probabilities of each outcome using the classical approach

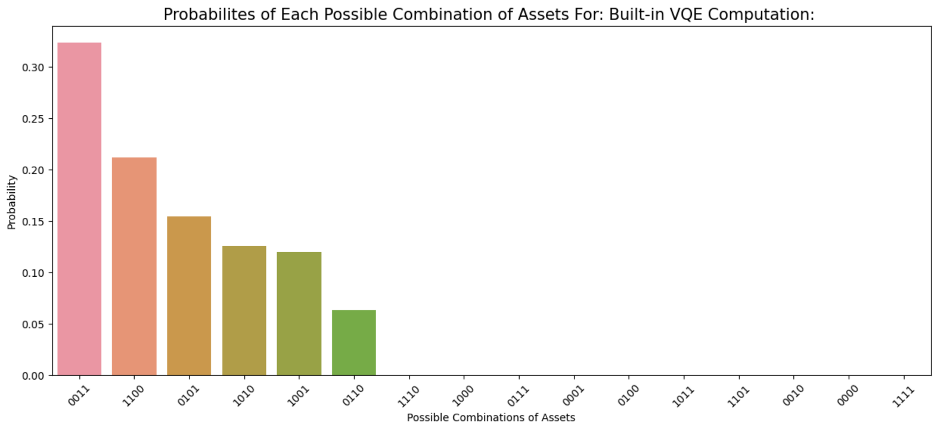


Figure 12: Probabilities of each outcome using the qiskit built-in VQE approach

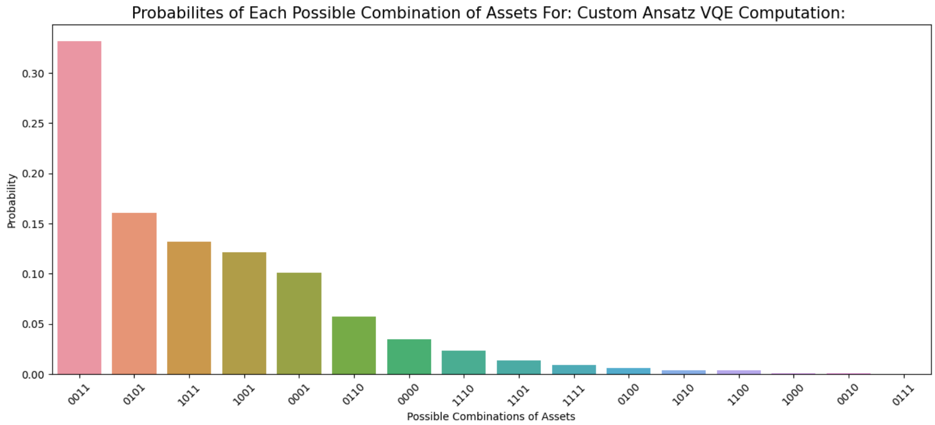


Figure 13: Probabilities of each outcome using the custom ansatz circuit designed and implemented with the qiskit VQE approach

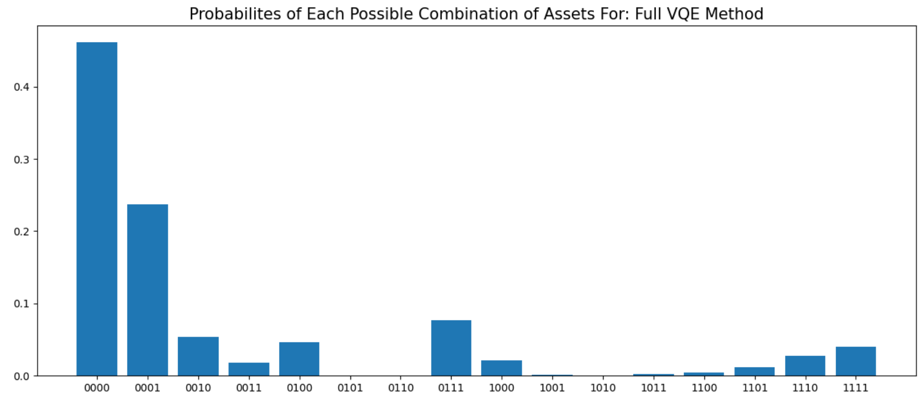


Figure 14: Probabilities of each outcome using the full VQE using 1 step of recursions approach

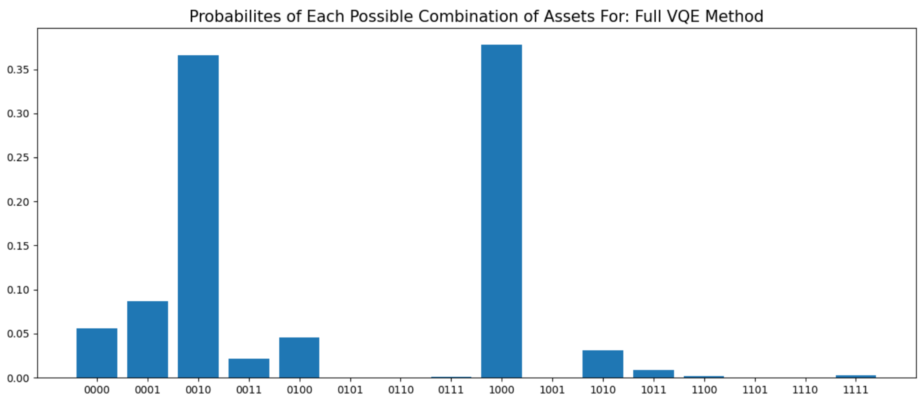


Figure 15: Probabilities of each outcome using the full VQE using 100 steps of recursions approach

1:

Computation Method	Execution Time (s)
Classical Eigen Solver	0.0159564
Built-in VQE	5.00699
Custom ansatz VQE	0.643002
Full VQE	0.491774

2:

Computation Method	Execution Time (s)
Classical Eigen Solver	0.0159576
Built-in VQE	4.99405
Custom ansatz VQE	0.640018
Full VQE	0.488872

3:

Computation Method	Execution Time (s)
Classical Eigen Solver	0.0159576
Built-in VQE	5.15026
Custom ansatz VQE	0.634169
Full VQE	0.539557

4:

Computation Method	Execution Time (s)
Classical Eigen Solver	0.0169547
Built-in VQE	5.07017
Custom ansatz VQE	0.648306
Full VQE	0.48778

5:

Computation Method	Execution Time (s)
Classical Eigen Solver	0.0169539
Built-in VQE	5.00948
Custom ansatz VQE	0.644228
Full VQE	0.499663

Figure 16: Execution times of different programs ran 5 times to get an accurate average

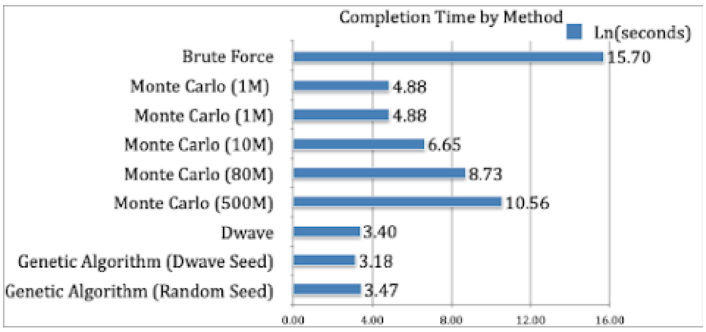
```
Custom Ansatz VQE Computation:
Optimal: selection [0. 0. 1. 1.], value -0.0031
Invest in the following stocks: ['GOOGL', 'TSLA']

----------------- Full result ---------------------
selection    value       probability
---------------------------------------------------
 [0 0 1 1]   -0.0031     0.3317
 [0 1 0 1]   -0.0018     0.1605
 [1 0 1 1]   1.0072      0.1322
 [1 0 0 1]   -0.0012     0.1216
 [0 0 0 1]   1.0086      0.1011
 [0 1 1 0]   -0.0025     0.0572
 [0 0 0 0]   4.0398      0.0346
 [1 1 1 0]   1.0076      0.0232
 [1 1 0 1]   1.0086      0.0133
 [1 1 1 1]   4.0371      0.0094
 [0 1 0 0]   1.0092      0.0061
 [1 0 1 0]   -0.0021     0.0038
 [1 1 0 0]   -0.0008     0.0038
 [1 0 0 0]   1.0097      0.0007
 [0 0 1 0]   1.0080      0.0006
 [0 1 1 1]   1.0066      0.0002
```

Figure 17: Example of results printed to console showing the optimal selection and the relative probability that this result will be found when run on quantum hardware

E APPENDIX

Table 5: Full VQE method over different rotational angle parameter recursions

Steps	**Time (s)**	**Error (%)**
1	0.492	98
10	4.968	97
50	23.596	96
100	47.319	96
500	235.881	96



(b) Completion Time by Method

Figure 18: Dwave Portfolio Optimisation Results