Quantum quantitative finance

Master's Thesis submitted to the
Constructor Institute
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Software Engineering

presented by
Kseniya Sidorova

under the supervision of
Prof. Bertrand Meyer
co-supervised by
Prof. Peter Schupp

June 2023
I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Kseniya Sidorova
Schaffhausen, 15 June 2023
To my family
Discovering a new explanation is inherently an act of creativity.

David Deutsch
Abstract

The field of quantum computing has been developing rapidly since the beginning of the 21st century, with many large companies investing in research and development in this area. Finance is considered one of the top four industries likely to see the earliest economic impact from quantum computing. Nevertheless, the initial excitement among banks is fading as the leap needed for real-life applications is still significant. This paper analyzes potential applications of quantum computing to finance and provides cases where quantum computing is already used in production. Detailed implementations of portfolio optimization using quantum technologies are examined, offering essential tools for finance professionals without a background in physics. It is important to prepare for advancements in quantum computing ahead of their full realization to gain a competitive advantage, mitigate future risks, and foster innovation.
Contents

List of Figures xi
List of Tables xiii

1 Introduction 1

2 Quantum computing overview 3
  2.1 History 3
  2.2 Basics 4
  2.3 Readiness for commercial use 5
  2.4 Quantum errors 6
  2.5 Funding 8

3 Application of quantum computing to quantitative finance 9
  3.1 Optimization 9
  3.2 Simulation 10
  3.3 Machine Learning 11
  3.4 Literature Review 12

4 Portfolio optimization analysis 15
  4.1 Problem 15
  4.2 Data selection and preparation 15
  4.3 Returns projection 16
  4.4 Quadratic model 18
  4.5 Minimizing risk while achieving a predetermined minimum return 18
  4.6 Maximizing return within a predefined risk limit 19
  4.7 Binary portfolio optimization 20
  4.8 Quantum computing frameworks 23

5 Conclusions 25

A Portfolio optimization results 27

Bibliography 39
Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison between the simulator results (left) and the real device (right)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>for a circuit with one qubit and one measurement.</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison between the simulator results (left) and the real device (right)</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>for a circuit with one Hadamard gate.</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>Bell's state.</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Comparison between the simulator results (left) and the real device (right)</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>for a Bell state circuit.</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>Number of Quantum Organizations by date. Source: The Quantum Insider</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Intelligence Platform</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Set 1 historical prices and returns.</td>
<td>16</td>
</tr>
<tr>
<td>4.2</td>
<td>Set 2 historical prices and returns.</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>Set 1 projected monthly returns and correlation matrix.</td>
<td>17</td>
</tr>
<tr>
<td>4.4</td>
<td>Set 2 projected monthly returns and correlation matrix.</td>
<td>17</td>
</tr>
<tr>
<td>4.5</td>
<td>Set 1 - Return/risk graph for cases 1.</td>
<td>19</td>
</tr>
<tr>
<td>4.6</td>
<td>Set 2 - Return/risk graph for cases 4.</td>
<td>19</td>
</tr>
<tr>
<td>4.7</td>
<td>Sampling returned by a real quantum device.</td>
<td>22</td>
</tr>
<tr>
<td>4.8</td>
<td>The nodes in light blue show software platforms that can be installed on the</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>user’s personal computer. White nodes show simulators run locally, i.e. on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the user's local device. In this case, the number of qubits that can be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>simulated is dependent on the performance of the user’s local computer. The</td>
<td></td>
</tr>
<tr>
<td></td>
<td>quantum computing company resources are shown in clouds. The red triangles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and squares represent the requirements to get access to the corresponding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>resource. The quantum simulators and usable quantum computers provided are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shown in dark green.</td>
<td></td>
</tr>
</tbody>
</table>
Tables

A.1 Minimizing risk while achieving a predetermined minimum return . . . . . . . . 28
A.2 Maximizing return within a predefined risk limit . . . . . . . . . . . . . . . . . . . 34
Chapter 1

Introduction

Quantum computing (QC), as a new prominent technology, has already received a lot of attention from academia and industry. It is a rapidly evolving field that holds the promise of solving complex computational problems more efficiently than classical computers. The four industries likely to see the earliest economic impact from QC are automotive, chemicals, finance and life sciences [McKinsey 2023]. In this paper we explore the advantages QC can bring into the field of quantitative finance. This field is computationally demanding and extensively employs stochastic modeling, making it a good match for quantum algorithms and hardware. Quantum computers are expected to solve the problems the classical computers would not be able to tackle.

The paper is organized as follows. Chapter 2 provides an overview of QC history, basic concepts and challenges. Using Qiskit SDK and free access to quantum computers provided by IBM Quantum we demonstrate how quantum errors influence the results of the simplest quantum circuits. In chapter 3, we present how three methodologies - optimization, simulation and quantum machine learning - can be applied to various problems in finance. Chapter 4 contains implementation details and analysis of portfolio optimization using quantum computing, and its comparison to the classical approach. It ends with an overview of quantum computing frameworks and some of the errors we encountered.

We believe our work can help quickly grasp the basics of QC and provide quantitative analysts and software engineers with a direction they can follow to leverage quantum technologies in financial applications.
Chapter 2
Quantum computing overview

Quantum computing appeared as a separate field in the late 20th century, although the foundational concepts were developed much earlier. It's heavily interdisciplinary, combining elements of computer science, physics, mathematics, and engineering.

2.1 History

The concept of a quantum computer first emerged in 1980 when physicist Paul Benioff proposed a quantum mechanical variant of the Turing machine. Richard Feynman and Yuri Manin later suggested that a quantum computer had the potential to simulate things that a classical computer could not. In 1982 Feynman proposed that a quantum computer would be able to act as a simulator for quantum physics - a natural task for a quantum system. It's difficult to pinpoint a single moment as the birth of quantum computing, but that 1982 issue of the International Journal of Theoretical Physics, where Feynman's work was published, often serves as a marker for the concept of a quantum computer's crystallization. In 1985, David Deutsch took these ideas further by describing the universal quantum computer, thus demonstrating that it was theoretically possible to build a quantum computer that could solve any problem that a classical computer could, only faster.

The 1990s and early 2000s saw further refinement of the theoretical model of quantum computing. Peter Shor, a theoretical computer scientist, developed a quantum algorithm in 1994, known as Shor's Algorithm. This algorithm can efficiently factor large numbers, a task that cannot be solved efficiently by any classical computer. This capability suggests that RSA encryption, a widely used method of securing data transmission that relies on the difficulty of factoring large numbers, could potentially be broken by quantum computers. In 1995, Peter Shor, along with Andrew Steane, contributed to the development of quantum error correction algorithms. These advancements collectively made quantum computing practical and applicable to real-world problems. In 1996 Lov Grover developed the database search algorithm, known as Grover's algorithm, best known for searching a database for a specific element, solving the unstructured search problem.

Isaac Chuang, Neil Gershenfeld and Mark Kubinec created a 2-qubit quantum computer in 1998, which represented an early, important milestone in the physical realization of quantum computers. Larger physical realizations of quantum computers started appearing in the 21st century with companies like IBM, Google, and D-Wave developing quantum processors. In 2019, Google
said they had achieved 'quantum supremacy' because their quantum computer performed tasks in 200 seconds that they believed would take a supercomputer around 10,000 years. But IBM disagreed with Google, saying a supercomputer could do the same work in about 2.5 days by using certain techniques to speed up the process. The research is ongoing as there are many practical difficulties in building large scale, error-free quantum computers and suitable quantum algorithms. A very recent paper by IBM and Berkeley states that a 127-qubit quantum processor with error mitigation was able to outperform supercomputers in performing complex physical simulations. That could mean that quantum computers can become useful before the full realization of error correction.

2.2 Basics

Before discussing the possibility of using QC in business we would like to describe the basic concepts. Quantum computers use three quantum-mechanical phenomena to perform computation.

1. In classical computers, a bit is either in a state of 0 or 1. In quantum computing, a quantum bit or qubit, the basic unit of quantum information, can be in a state of 0, 1, or both at the same time. This principle is called superposition, and it allows quantum computers to process a large number of possibilities at once.

2. Entanglement is a unique quantum property where pairs of qubits can be linked to each other in such a way that the state of one qubit is directly related to the state of another, no matter the distance between them. This property is essential for quantum computing as it allows qubits that are in an entangled state to be used for multiple computations simultaneously.

3. Interference is used to manipulate the probabilities of qubits being in one state or another. Through constructive and destructive interference, the states with wrong answers can be canceled out, and the states with the correct answers can be amplified.

Orús et al.\[2019\] describe the quantum algorithm as a sequence of the following 5 steps:

1. Input data is encoded into the state of a set of qubits.

2. Quantum superposition is used to connect the qubits.

3. Algorithm is applied simultaneously to all the states (i.e. quantum entanglement amongst the qubits is used); at the end of this step one of the states holds the correct answer.

4. Quantum interference is used to amplify the probability of measuring the correct state.

5. Measurement.

Currently there are two prominent methods in quantum computation that utilize quantum phenomena to process information: gate-based quantum computing and quantum annealing. Gate-based (or universal) QC operates by manipulating qubits using various gates, similar to how classical bits are manipulated via logical gates in classical computing. Quantum gates act on one or more qubits, causing them to change their state. Computations are executed by systematically applying a sequence of these gates to form a quantum circuit. The model is
universal, implying that, theoretically, with sufficient qubits and appropriate gates, any computational task can be performed. Offering more flexibility than the second quantum annealing method, gate-based quantum computing can address a broader scope of problems.

Quantum annealing is a quantum computational method used to find the global minimum of a given objective function over a given set of candidate solutions. It’s primarily used for solving optimization problems and is best suited for problems where the goal is to find the lowest energy state of a system. Quantum annealers are more noise-tolerant than their gate-based counterparts. That can be partially credited to their hardware architecture, wherein the qubits are interconnected in clusters that are mutually entangled. There has been speculation regarding the authenticity of D-Wave’s computer as a quantum device, but it has been demonstrated that it possesses a speed advantage over a classical computer [Leprince-Ringuet, 2021]. For more details on quantum annealers see sections 2.3 and 3.1.

2.3 Readiness for commercial use

Quantum annealers are already used in commercial applications [Yarkoni et al., 2022]. The largest quantum annealer was produced by D-Wave Systems in 2022, has over 5000 qubits and 15-way qubit connectivity (meaning each qubit has 15 connections). Although this just results in 5000 bit values per sample, while in the gate-based model quantum states exhibit an exponential growth in their dimensionality with the addition of each qubit. This allows a quantum computer to store significantly more information compared to a classical computer with an equivalent number of bits. In November, 2022, IBM launched the most powerful universal quantum computer with 433 qubits. While it is 3 times larger than their previous quantum processor, its capacity is far from being suitable for production systems. According to Woerner and Egger [2019], quantum advantage for derivative pricing, for example, is achievable with 7500 logical (ideal) qubits. Wilkens and Moorhouse [2023] estimate that “a market risk application in a large financial institution typically encounters the following ‘size challenges’:

- Portfolios: 1,000 to 10,000
- Instruments/trades: 100,000 to 1,000,000
- Valuation parameters: 10,000 to 10,000,000

For the actual risk quantification, one would rely on:

- Scenarios (for a subset of the valuation parameters): 100 to 1,000,000
- Time horizons (not necessarily equally spaced): 1 to 1,000”.

The calculations need to be accurate, not exceeding 10% margin of error, and should not break certain time limits. These requirements also determine the storage capacity and processing speed that a risk system must meet. Braun et al. [2021] estimate that with 200 (error corrected) qubits, the quantum algorithm could solve problems of practical relevance, which are impractical to perform on classical hardware.

The operation of quantum computers is highly dependent on the quantity of available qubits and accuracy of their states. Maintaining the correct state of qubits and handling errors are vital for further use of the technology. Quantum computers are extremely sensitive to noise and errors caused by interactions with their environment. Even small disturbances can cause
them to lose their quantum properties, a phenomenon known as *decoherence*. Overcoming this issue may require new materials, new computational methodologies and detailed exploration of various quantum approaches. Moreover, the challenges in QC are not solely confined to hardware. Quantum algorithms are also much more complex than classical algorithms and require developers to approach computational problems in original ways to make near term quantum computers applicable to real world problems.

### 2.4 Quantum errors

To examine the issue of quantum errors, we conducted a series of experiments on an actual quantum computer, which is accessible for free through [IBM Quantum](https://www.ibm.com/quantum). We used the following two 5-qubit instances: ibmq_quito and ibmq_lima. Below we provide a comparison of the circuit’s execution on a simulator versus a real computer.

1. Executing a circuit with one qubit (instantiated with zero) and a single measurement operation 1000 times produces outcomes summarized in Figure 2.1. We can see that even without any gates (basic units of quantum processing) 6.8% of the information is incorrect.

2. After adding a Hadamard gate, which creates a superposition of the basis $|0\rangle$ and $|1\rangle$ states (states in quantum mechanics are written using bra-ket notation, composed of angle brackets and vertical bars), we expect to see a 50% probability of each outcome. The allowable error should not exceed the square root of the number of observations. In this case we expect that the difference between the first and second bars will be less than the square root of $\sqrt{1000} \approx 32$ [Persch 2021]. Figure 2.2 summarizes the results.

We can see that the results of the quantum computer are more skewed. While the difference between the frequencies of 0s and 1s is exactly 32 in the case of the classical computer, it is double that value for the quantum case.
3. Bell’s state is a quantum state of two qubits that represents the simplest (and most commonly used) example of quantum entanglement. It can be illustrated by the Figure 2.3.

First, a Hadamard gate is applied to the qubit q0, putting it into a superposition of $|0\rangle$ and $|1\rangle$ (with a 50% probability being each), then a CNOT gate is applied to the qubit q1 with the q0 being the control qubit. The CNOT gate negates the target input only if the control input is set to 1. Thus we expect that the outcome of running the circuit would be either $|00\rangle$ or $|11\rangle$. The results summarized in Figure 2.4 demonstrate a bias towards zero for the quantum computer, alongside a 7.3% occurrence of error states. From the examples above we can see that even the simplest circuits are error-prone on a real quantum device. The errors are caused by the aforementioned decoherence and gate infidelity. The latter refers to discrepancies between ideal logical gates and their physical counterparts provided by quantum hardware, which arise due to a variety of factors, including control errors, environmental noise, or interaction with other qubits.
The current state of QC is referred to as the noisy intermediate-scale quantum (NISQ) era, characterized by intermediate-sized quantum processors that are subject to noise and imperfections. It should be noted that newer quantum computers than the ones used in our experiments can potentially yield better results. Researchers are actively working on developing error correction techniques and improving the performance of NISQ systems to pave the way for more powerful quantum computers in the future.

2.5 Funding

Quantum computing is an actively developing field. It is represented not only by established companies like Google, IBM, Microsoft, Honeywell, but also by a considerable number of start-ups with the most well known ones being D-Wave, Rigetti, IonQ and Xanadu. Figure 2.5 presents the number of QC companies over the last 20 years. Worldwide investments in quantum technology start-ups reached their highest levels in 2022, at $2.35 billion [McKinsey 2023], but the rate of new-company creation has not kept pace with investments. Only 19 quantum technology start-ups were founded in 2022, compared with 41 in 2021, bringing the total number of start-ups in the quantum technology ecosystem to 350. This implies that a greater amount of investment is being directed towards established start-ups rather than new companies.

Even though quantum computing hasn’t reached mainstream adoption yet, a variety of entities including major corporations, governments, and universities are actively investing in its advancement, aiming for quantum supremacy or preparing themselves for its eventual introduction. At the same time it should be noted that some banks after several years of research are already losing interest in the technology, not seeing the advantage over the classical methods and given that ready-to-use technology is still distant [Clancy 2020, 2023]. The graph above also supports the view that the excitement surrounding QC has diminished over the last couple of years.

Figure 2.5. Number of Quantum Organizations by date. Source: The Quantum Insider Intelligence Platform.
Chapter 3

Application of quantum computing to quantitative finance

According to McKinsey [2023] financial services are among the four industries likely to see the earliest economic impact from quantum computing, with three others being automotive, chemicals, and life sciences. Quantitative finance uses mathematical models and computational algorithms to predict and analyze the behavior of markets and financial instruments. It involves the application of physics-style modeling techniques to the financial markets, such as use of diffusion concept, Brownian motion and mean reversion.

There are three major classes of financial problems, which can be addressed by QC:

- Optimization
- Machine Learning
- Simulation

In this chapter we provide details and potential applications for each of the methods.

3.1 Optimization

Optimization is at the core of many financial problems. One of the most popular applications in this category is portfolio optimization. In this context, investors aim to maximize returns while maintaining a predefined risk level. There are certain constraints, such as budget, asset classes, transaction costs which should be incorporated into the problem. Portfolio optimization is often an NP-hard problem, with efficient frontiers (the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return) being calculated by randomized algorithms.

Quantum optimization algorithms primarily utilize a technique known as adiabatic quantum computation. The first step involves mapping the optimization problem onto a physical one, specifically, finding the ground state of a Hamiltonian $H_p$, which encodes the cost function to be minimized. A well-known and simple Hamiltonian $H_0$ is gradually transformed into the target Hamiltonian $H_p$ over a long time $T$. The adiabatic theorem states that a system initiated in its ground state will remain close to its instantaneous ground state, provided its lowest energy
3.2 Simulation

Simulation is one of the essential tools daily employed by financial institutions. Analytical models rarely cover all complex relationships between financial instruments. Hence, Monte Carlo (MC) simulation, a method utilizing repeated random sampling to solve the problems numerically, is largely employed. MC methods are used to calculate risk metrics, price financial instruments, estimate economic capital, run scenario analysis for stress tests and solve other computational problems. The quantum version of MC methods, based on the Quantum Amplitude Estimation (QAE) algorithm, provides a quadratic speedup over its classical counterpart [Brassard et al., 2002]. The following two groups of problems are expected to benefit from the use of quantum simulation techniques.

1. **Derivatives pricing.** Financial derivatives are contracts with a payoff that depends on the future price of some asset. Preparing relevant probability distributions in quantum superposition, implementing payoff functions via quantum circuits, and extracting option prices through quantum measurements are some potential applications. With an efficient breakdown of the payoff function into elementary arithmetic operations within a quantum circuit, this approach could potentially be applied to any derivative [Rebentrost et al., 2018]. For a detailed discus-
sion on how QAE can provide an advantage for options pricing, we refer to Stamatopoulos et al. [2020].

2. Risk management. Established in response to the 2008 financial crisis, Basel III is an international regulatory framework. It sets requirements on capital, risk coverage, leverage, market discipline, liquidity, and supervisory monitoring [Bank for International Settlements (BIS)]. The banks are obliged to hold predefined levels of non-risk capital, and regularly report certain risk values to the Basel Committee. Part of these regulations involves the computation of risk metrics like Value at Risk (VaR) and Conditional Value at Risk (CVaR), economic capital requirement. QAE can be used to evaluate those values faster than the classical MC methods.

To assess financial risks banks perform sensitivity analysis, a modeling tool that helps determine how independent variables affect dependent ones. It helps identify which factors have the most significant impact on the model’s outcomes. Those sensitivities are called Greeks in quantitative finance, and are calculated as partial derivatives. The results help banks understand potential risks and uncertainties, and make better informed decisions. They also have to be reported to the central authorities (European Central Bank, for example).

Sensitivity analysis are performed through extensive scenario simulation and are usually done overnight due to their long duration. Given the random nature of the parameters involved, closed-form analytical solutions are often unavailable, requiring the use of numerical methods. Braun et al. [2021] present a study on sensitivity analysis utilizing quantum computation. CVA and XVA are two more important quantities calculated using MC methods and required by regulators [Gregory 2020].

Taking into consideration all of the aforementioned applications, along with their computational complexity, the field of risk management can certainly benefit from the speedup and extensive coverage of scenarios gained by the use of quantum Monte Carlo methods.

3.3 Machine Learning

Machine Learning essentially involves the creation and application of algorithms that can be trained to perform a multitude of tasks, including pattern recognition and data classification. Training refers to the optimization of the algorithm’s parameters to identify specific inputs or training data, which can subsequently be applied to evaluate new inputs. The field of classical machine learning has seen exponential growth, primarily owing to advancements in hardware and algorithms, such as those that facilitate the training of deep learning networks.

The intersection of machine learning and finance has opened up new ways to solve complex financial problems more effectively. Use of quantum computers would result in increased computational speed. The field of Quantum Machine Learning (QML) focuses on two main aspects: developing quantum versions of machine learning algorithms and using classical machine learning to understand quantum systems. It’s important to note that many potential breakthrough QML algorithms require the functionality of a universal quantum computer, which is technologically more challenging and advanced than quantum annealers. Consequently, while optimization problems can currently benefit from the first generation of experimental quantum annealers, certain QML algorithms won’t be feasible until the technology evolves further. Potential applications of QML include credit assessment through data classification techniques, optimization of supply-chain operations using regression models, and comprehensive analysis of interest rates via principal component analysis. In this paper we do not provide detailed quantum machine learning algorithms, leaving it for further work.
3.4 Literature Review

There has been extensive research and development of QC techniques and their applications in the field of finance. We would like to highlight a few surveys that offer valuable guidance for interested individuals.

To our knowledge, Herman et al. [2022] provide the most structured and comprehensive summary of financial problems, with particular emphasis on stochastic modeling, optimization and machine learning, and how they can be solved using quantum computers.

The paper by Fedorov et al. [2022] includes an “Economically impactful application” chapter, where one can find names of many industry leaders (for example “Airbus”, “BMW”, “Daimler-Benz” etc), who have already invested into the quantum implementation.

D-Wave Systems website contains a list of 250+ featured real business applications of the quantum computing technology. The examples include Volkswagen team’s “Traffic Flow Optimization Using a Quantum Annealer” and E.ON’s grid partition into clusters necessary for decentralized energy distribution.

Many articles provide an introduction into the quantum field in general. To get a full picture, a book by Nielsen and Chuang [2000] is considered a classic. While a book by Hidary [2019] combines the fundamental principles of quantum computing with a practical, hands-on coding approach. We would also like to mention several online resources on QC:

- A detailed course on Quantum Computation taught by Preskill from Caltech.
- Lectures by Vazirani, a professor from Berkeley, which are concise and do not require a physics background.
- A channel by Marchenkova, a research scientist from Bleximo, that can provide some direction and inspiration for those starting in the quantum computing field.

The book by Jacquier et al. [2022] explores the main quantum computing algorithms and highlights financial applications that can benefit from this new approach. Egger et al. [2020] highlight problem areas in finance that are difficult to solve using classical computers but hold promise for quantum algorithms. The paper includes demonstrations of quantum algorithms on IBM Quantum back-ends and states potential advantages they offer for financial services.

Albareti et al. [2022] provide analysis and classification of 13 peer-reviewed papers with concrete applications of QC to Finance dating to 2021. The use cases covered include: portfolio optimisation, transaction settlement, predicting financial crashes, estimating risk measure(s) and derivative pricing. The authors map the cases to the methodology applied (Optimisation or Monte Carlo) and then to the algorithms and hardware used.

During the last couple of years, as the quantum technologies were progressing and SDKs were providing an increasing number of tools to developers, there has been a notable number of new publications in the field. A recent paper by Wilkens and Moorhouse examines requirements and approaches for the application of QC to risk management in financial institutions. They come to a conclusion that quantum solutions are not very strong at this point as they do not provide sufficient accuracy, while the classical algorithms perform well and in reasonable time. Several papers by Mugel et al. [2021, 2022], Palmer et al. [2021], a group of researchers from a Spanish company “Multiverse Computing”, analyze various methods of portfolio optimization, coming to a conclusion that D-Wave’s hybrid solver outperforms other approaches. Similar results are achieved by Sakuler et al. [2023]. The authors demonstrate that the new hybrid solvers implemented by D-Wave provide the best results, but the shortcoming is that the
exact algorithm is not known. The article by Venturelli and Kondratyev [2019] presents an optimized reverse annealing approach to the mean-variance portfolio optimization problems, which is on average 100 times faster than the corresponding forward quantum annealing. Braun et al. [2021], a team of developers and researchers from a German-based company “JoS QUANTUM”, present a paper on the use of QC for sensitivity analysis of business risks. Their library of various quantum algorithms 'pyqrnd' can be found on github JoS QUANTUM. The documentation is clear and contains a lot of examples, providing the end users with an ability to apply QC techniques to their business cases (especially in such areas as Finance and Insurance).
3.4 Literature Review
Chapter 4

Portfolio optimization analysis

Quantum annealing is a heuristic quantum optimization algorithm that can be used to solve combinatorial optimization problems. In this chapter we present the results of portfolio optimization, a practical financial problem commonly encountered in the field of investment management, using D-Wave quantum annealer. We also perform binary optimization using the gate-based approach. We examine both approaches and compare the results between them and with those obtained using classical methods.

4.1 Problem

We choose a mean-variance approach to portfolio optimization, proposed by Markowitz in his Modern Portfolio Theory. It is a methodology that aims to balance the trade-off between risk and return in an investment portfolio. Its ultimate goal is to find the efficient frontier - a set of optimal portfolios that offer the highest expected return for a given level of risk or the lowest risk for a given level of expected return. The optimal portfolio is therefore not a one-size-fits-all concept, but instead is specific to each investor’s individual risk tolerance, return expectations, and the correlation between the different investments. Thus we evaluate two different approaches for choosing an optimal portfolio:

1. Minimizing risk while achieving a predetermined minimum return.

2. Maximizing return within a predefined risk limit.

4.2 Data selection and preparation

The data used for analysis consists of stocks from the S&P index, issued by 500 large-cap companies traded on American stock exchanges. The timeframe extends from December 2017 to May 2022. We intentionally chose historical data to enable testing and comparison of results against actual values. Prices were downloaded from Yahoo Finance using the ‘pandas_datareader’ wrapper. We then removed stocks with blank values and resampled the data to reflect monthly returns. We selected 30 random stocks from the set because a balance should be maintained between the number of observations and the number of variables for proper analysis.
4.3 Returns projection

After retrieving historical prices we calculate the returns and their covariance matrix. The goal of this step is to forecast future returns and pick the best performing stocks for the portfolio. To achieve this we look at different methodologies and compare the results to the actual stock returns.

1. Projection based on the historical average return.
2. Monte Carlo simulation, based on the assumption of returns’ normal distribution.
3. Prophet library, a forecasting framework based on machine learning.

According to Sakuler et al. [2023] it is common for the client’s portfolio to hold 9 to 11 assets. Thus we select 10 stocks with the highest projected returns using each methodology. We then compare the projected returns to the actual ones. The results vary from one run to another (as random stocks are selected each run). It is concluded that each approach can be used for forecasting returns. We decide to proceed with the results from the first methodology, taking them as input data for the next step.

After experimenting with random stocks, we fix two datasets for further analysis:


Different time periods are used for research. Figure 4.1 and Figure 4.2 display the prices and returns of the two sets being considered. Figure 4.3 and Figure 4.4 present predicted average returns and correlation matrices of the selected stocks sets.

![Monthly prices of the selected top 10 stocks](image1)

![Monthly returns of the selected top 10 stocks](image2)

Figure 4.1. Set 1 historical prices and returns.

We can see that the sets chosen are different in terms of returns and diversification, with the first one having much lower predicted average return and lower correlation. This can partly be attributed to the difference of the time periods under consideration.
4.3 Returns projection

Figure 4.2. Set 2 historical prices and returns.

Figure 4.3. Set 1 projected monthly returns and correlation matrix.

Figure 4.4. Set 2 projected monthly returns and correlation matrix.
4.4 Quadratic model

We are now ready to construct the optimization problem and solve it using different techniques. LeapHybridCQMSampler is a D-Wave quantum-classical hybrid solver that solves constrained quadratic model (CQM) problems. A CQM problem includes an objective (a quadratic function that the solver tries to minimize) and one or more constraints (conditions that candidate solutions must satisfy). The hybrid solver uses a combination of classical and quantum resources to solve these problems. The exact algorithms used by D-Wave’s hybrid solvers are proprietary and not fully disclosed.

As a classical counterpart we use the GEKKO library, an optimization suite for Python, also used by Mugel et al. [2021]. It utilizes dynamic optimization to find the optimal solution.

4.5 Minimizing risk while achieving a predetermined minimum return

The problem can be summarized as follows.

$$\min_x \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} p_i p_j x_i x_j$$

subject to: $n \sum_{i=1}^{n} x_i p_i \leq B$, and

$$\sum_{i=1}^{n} r_i x_i \geq r$$

where

- $x_i \in \{0, 1, 2, \ldots\}, i \in \{1, 2, \ldots, n\}$ denote the amount of i-th stock to be purchased,
- $p_i$ are stocks purchase prices,
- $r_i$ are projected returns,
- $\sigma_{ij}$ are elements of the covariance matrix,
- $r$ is the minimum return,
- and $B$ denotes the budget.

We set up ConstrainedQuadraticModel with the following constraints:

- The invested amount should not exceed the budget of $10.000 or $1.000;
- Most of the budget (99.7%) should be invested;
- The total return should exceed the predefined level.

The objective of the model is the minimization of the risk (variance). The solution of the problem is a vector containing quantities of stocks to be purchased. We run the optimization for several cases. The results are summarized in the table [A.1]. Let’s take a closer look at some of the cases. Cases 1-3.

Data: set 1, 2020-07-01 - 2022-04-30, budget - $1.000, minimum return - 3%.

Figure 4.5 displays feasible solutions for case 1. The resulting return is approximately 3% with a volatility slightly higher than 6%. Cases 4-7.
4.6 Maximizing return within a predefined risk limit

Data: set 2, 2020-07-01 - 2022-04-30, budget - $1.000, minimum return - 3%.

Figure 4.6 displays feasible solutions for case 4. The predicted returns for Set 2 are considerably higher than of the Set 1. The initial lower bound of 5% return is always satisfied, thus we increase it to 8% and proceed with optimization.

In both first and second cases the quantum algorithms were able to outperform the classical one. At the same time it should be noted that the classical algorithm returned the answer in 0.17 seconds vs minimum of 5 seconds in case of the quantum annealer.

The results for the second set are consistent with the ones for the first set. We can see that the quantum annealer is already solving optimization problems on a satisfactory level, in some cases beating the classical algorithm (with default settings). The less constrained variables are, the more we see possibilities for the quantum advantage. We also ran the optimization algorithm on a larger set of stocks. When the number reached 30 the classical GEKKO solver failed to generate a solution due to a string overflow error.

4.6 Maximizing return within a predefined risk limit

To proceed with analysis we approach the portfolio optimization problem using the maximization of returns. The problem can be formulated as follows:
4.7 Binary portfolio optimization

In order to compare different quantum hardware and algorithms provided by different SDK’s we construct a binary portfolio optimization problem, solvable on both D-Wave Quantum Annealer and IBM Quantum gate-based computers. The problem we are to solve is as follows:

\[
\min_{x \in \{0, 1\}^n} qx^T \Sigma x - \mu^T x \\
\text{subject to: } 1^T x = B
\]

where:
- \(x \in \{0, 1\}^n\) is a binary vector, which indicate which assets to pick (\(x[i] = 1\)) and which not to pick (\(x[i] = 0\)),
- \(\mu \in \mathbb{R}^n\) defines the expected returns for the assets,
- \(\Sigma \in \mathbb{R}^{n \times n}\) specifies the covariances between the assets,
- \(q > 0\) controls the risk tolerance of the investor,
- and \(B\) denotes the budget.

In that case the budget constraint is the total amount of stocks the investor can purchase, with the solution vector being a binary one. We ran the optimization problem on Set 2, using several methods, which produced the same result.

Full result: \([1 1 1 0 0 1 0 1 0 0]\), selection value: -0.2807, probability: 1.0000

Below we present a short description of each method.

1. **NumPyMinimumEigenSolver** is a classical algorithm provided by Qiskit. It’s used to find the...
minimum eigenvalue of an operator (like a Hamiltonian representing a quantum system) directly using classical computations. The input to the solver is an operator, which is usually a Hamiltonian representing a quantum system or problem. The solver first uses NumPy’s linear algebra to diagonalize the Hamiltonian. The eigenvalues of a Hermitian operator (the values you get when you diagonalize it) represent the possible outcomes when you measure the system, and the associated eigenvectors represent the states the system will be in when you get those outcomes. Then the algorithm simply finds the smallest eigenvalue and its corresponding eigenvector.

This solver is accurate because it computes the exact solution, but it has a high computational cost, as diagonalizing a matrix is a computationally intensive task. It becomes infeasible for large systems (large matrices) due to its exponential time complexity. It’s often used as a benchmark for other quantum algorithms like VQE or QAOA, which attempt to approximate the same minimum eigenvalue using fewer resources, but potentially with less accuracy. To solve the quadratic problem NumPyMinimumEigensolver is used in conjunction with the MinimumEigenOptimizer, a Qiskit tool for translating Quadratic Programs into a form suitable for quantum eigenvalue solvers. Without the optimizer one would need to manually perform this translation, converting the quadratic program to a QUBO form and then to a Pauli operator (or a list of Pauli terms), which can be handled by quantum algorithms.

2. Sampling Variational Quantum Eigensolver (SamplingVQE) is a Variational Quantum Eigensolver algorithm, optimized for diagonal Hamiltonians. Let’s explain the algorithm and its constituents by analyzing the code snippet.

```python
cobyla = COBYLA()
cobyla.set_options(maxiter=15)
ry = TwoLocal(n, "ry", "cx", reps=3, entanglement="full")
vqe_mes = SamplingVQE(sampler=Sampler(), ansatz=ry, optimizer=cobyla)
vqe = MinimumEigenOptimizer(vqe_mes)
result_samplingvqe = vqe.solve qp
```

COBYLA or Constrained Optimization By Linear Approximations is a numerical optimization method for constrained problems where the derivative of the objective function is not known. Experiments with a maxiter parameter showed that a larger value may lead to worse results. There could be several reasons for such behavior: the optimizer could be trying to refine a sub-optimal solution or there could be several local minima.

TwoLocal function constructs a variational quantum circuit which consists of rotation layers (with ry gates, which are y-axis rotations on the Bloch sphere) and entangling layers (with cx gates, which are controlled-X gates, also known as CNOT gates). This circuit is repeated (reps) 3 times. The entanglement is set to “full”, which means that each qubit will be entangled with all others.

The VQE (Variational Quantum Eigensolver) is a quantum/classical hybrid algorithm that can be used to find the ground state energy of a Hamiltonian. The SamplingVQE is a modification of VQE that uses a sampler to approximate the expectation value. It also requires an ansatz, a parameterized QuantumCircuit, to prepare the trial state, and a classical optimizer for varying the circuit parameters.

After having tuned the maxiter parameter and rotation and entangling gates of TwoLocal we were able to achieve the same result as other methods.
3. Quantum Approximate Optimization Algorithm (QAOA) is another variational algorithm and it uses an internal variational form that is created based on the problem. The use of QAOA is similar to SamplingVQE, as it also needs a sampler and an optimizer to solve the problem, but uses its own fine-tuned ansatz. QAOA is thus principally configured by the single integer parameter, reps, which dictates the depth of the ansatz, and thus affects the approximation quality.

4. To solve the problem on the D-Wave Annealer we first use ‘pyqubo’ library to create a QUBO model. Then it is passed to a D-Wave sampler for solving.

```python
sampler = DWaveSampler(solver=' Advantage_system6.2', token=token)
sampler_qa = EmbeddingComposite(sampler)
sampleset_qa = sampler_qa.sample_qubo(qubo,
    label='Forward Annealing', num_reads=2000)
```

It should be noted that the optimal solution was not found after 1000 iterations. By increasing the maximum number of iterations (num_reads parameter), we were able to achieve the same result as classical methods. Instead of increasing the number of output solutions, which can result in an excessive QPU access time, we can use the output of the forward annealing algorithm as an input for the reverse annealing approach. The reverse annealing approach was able to find the optimal solution in about 250 iterations.

5. It is also possible to solve the problem using the LeapHybridSampler with the solver ‘hybrid_binary_quadratic_model_version2’. It converts QUBO to a Binary Quadratic Model (BQM) and performs sampling. Again the returned result is inline with other approaches.

6. In order to compare the simulator results to that of the available 5-qubit quantum computer we modify the portfolio optimization problem to choose 2 stocks out of 5. The five stocks under consideration are from Set 2: [‘DVN’, ‘MRO’, ‘MOS’, ‘OXY’, ‘MRNA’]. Figure 4.7 shows the sampling distribution by the quantum instance. According to the algorithm, after sampling

![Figure 4.7. Sampling returned by a real quantum device.](image-url)
the MinimumEigenOptimizer is applied. The solution coincides with the results received using the aforementioned approaches, which were run on the same subset. But it should be noted that the probability of the optimal sample is lower than for some other solutions. But due to the application of the classical MinimumEigenOptimizer the solution with the smallest value is chosen. Optimal: selection \[ [1. 0. 1. 0. 0.] \], value \(-0.1584\), probability < 0.1
This concludes our practical experiments with the portfolio optimization problem.

### 4.8 Quantum computing frameworks

During our research we use two frameworks for implementation - IBM Qiskit SDK and D-Wave Ocean SDK. Both of them are well documented and provide interesting examples. Additionally both IBM and D-Wave Systems provide free access to quantum hardware. Figure 4.8 by Albareti et al. shows available quantum platforms as of the end of January 2021.

While we didn’t have any major problems with D-Wave’s quantum annealer, there were several programming issues that we encountered while programming quantum circuits. We believe mentioning some of them and their solutions may save someone else’s time.

1. Running circuits on a real quantum computer device requires some extra code, which you will rarely find in the examples, mostly written for running the code on a simulator. Although it
may seem that the only difference you would make is changing the name of the instance (from a simulator to a quantum server) you want to use. The code of Qiskit SDK does not provide a good abstraction level, so the developer is actually expected to change the code based on the realization of the backend instance used. That is one of the examples of poor abstraction.

```python
from qiskit_ibm_provider import IBMProvider

provider = IBMProvider(instance='ibm-q/open/main')

SIM = True
backend_name = ''
if (SIM):
    backend_name = "simulator_statevector"
else:
    backend_name = "ibmq_lima"
backend = provider.get_backend(backend_name)
shots = 1000
qc = QuantumCircuit(1)
qc.reset(0)
qc.h(0)
qc.measure_all()

backend.run(qc, shots=shots)
```

If the variable SIM is set to True, the code is successfully executed on a simulator. But if it is False and the code is run on a real quantum device, an error will be thrown.

'Instruction h is not supported', 'code': 7001

It should be mentioned that the error will be seen only after the task gets through the execution queue (minimum waiting time is usually 15 minutes). The error results from the fact that a transpilation (conversion of non-basis gates to the basis ones) needs to be called before running the circuit on the real device. That could have been done by the framework. 2. Apart from demonstrating Qiskit’s poor abstraction level the example above also shows a mediocre error handling mechanism. The message clearly can be improved while the precondition check can be performed without using the actual quantum hardware, following the fail-fast principle. 3. System may crash without any explanation. While solving the optimization problem the classical numpy optimizer returned an understandable error: “MemoryError: Unable to allocate 256. TiB for an array with shape (35184372088833,) and data type uint64”. Running the same problem on a quantum simulator resulted in a kernel crash.

4. Qiskit is an open-source SDK. It has many submodules interacting with each other. Unfortunately the downside is that there are many inconsistencies in the API for end users. Not all of the examples are up to date, the SDKs are constantly changing, and the documentation is behind.

The version of Qiskit SDK is currently at 0.43.1 and it’s under active development, so we can hope that many of the existing issues will be resolved soon.
Chapter 5

Conclusions

Quantum computing is a rapidly developing technology. It has already seen interest from large companies, universities and governments. Nevertheless, investments in quantum technologies are similar to a bet: the uncertainty is high, and timing is unclear. The most popular gate-based models are prone to decoherence, noise, and gate infidelity. In our work we show how simple circuits are affected by such errors. In contrast, quantum annealers, which are less susceptible to decoherence and thermal noise, have already found their use in production systems.

Quantitative finance, a field requiring extensive complex computation, can benefit from QC in three major application areas: optimization, simulation and machine learning. We present many possible examples of financial applications and several references that would help interested individuals to map their problem to a QC methodology and proceed with implementation.

Some large banks are already losing interest in QC, as classical methods provide satisfying solutions within acceptable timing. Nevertheless, they have already performed some research and thus built a knowledge base within the organization. If there is a quantum breakthrough they will be prepared at least partially.

We present how the problem of portfolio optimization can be solved on a quantum computer. Our experiments show that the quantum annealer already yields better results than one of the classical approaches. Unfortunately, this improvement comes with a longer duration, which can be crucial for financial applications. We expect that in the short term quantum annealers will be useful for optimization problems, while in the long term gate-based quantum computers will even outperform them. To advocate for the use of quantum technologies it is necessary to look for applications where the gain from QC would be substantial enough for financial institutions to switch to the new approach.

To our mind, stress testing and sensitivity analysis requiring extensive simulation and usually performed overnight due to their long duration, could gain the most of new quantum technologies. It is definitely a field of further research. Quantum machine learning is another area of future work. The recent development of new error mitigation methods may bring the era of quantum advantage sooner than expected. At the same time it should be noted that the financial industry is highly regulated and quantum computation would need to go through formal compliance procedures to be applicable within financial organizations.
Appendix A

Portfolio optimization results

Two tables below present summarized results of portfolio optimization. 'Vol' denotes 'Volatility'.
Table A.1. Minimizing risk while achieving a predetermined minimum return

<table>
<thead>
<tr>
<th>Case</th>
<th>Set</th>
<th>Start date</th>
<th>End date</th>
<th>Budget</th>
<th>Min return</th>
<th>Methodology</th>
<th>Solution</th>
<th>Actual return</th>
<th>Risk</th>
<th>Vol, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>30 (3%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['MPC': 2, 'HPQ': 4, 'STX': 0, 'ANET': 0, 'NTAP': 2, 'ODFL': 1, 'CDW': 1, 'APTV': 1, 'INTU': 0, 'CRM': 0]</td>
<td>30.63</td>
<td>3,685</td>
<td>6.08</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>30 (3%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['MPC': 3, 'HPQ': 2, 'STX': 0, 'ANET': 0, 'NTAP': 1, 'ODFL': 1, 'CDW': 2, 'APTV': 0, 'INTU': 0, 'CRM': 0]</td>
<td>30.74</td>
<td>3,611</td>
<td>6.03</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>30 (3%)</td>
<td>Classical Gekko solver</td>
<td>['MPC': 3, 'HPQ': 2, 'STX': 0, 'ANET': 0, 'NTAP': 1, 'ODFL': 1, 'CDW': 2, 'APTV': 0, 'INTU': 0, 'CRM': 0]</td>
<td>30.74</td>
<td>3,611</td>
<td>6.03</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['MPC': 25, 'HPQ': 5, 'STX': 14, 'ANET': 0, 'NTAP': 27, 'ODFL': 5, 'CDW': 19, 'APTV': 0, 'INTU': 0, 'CRM': 1]</td>
<td>303.76</td>
<td>343,319</td>
<td>5.9</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>Classical Gekko solver</td>
<td>['MPC': 25, 'HPQ': 7, 'STX': 10, 'ANET': 0, 'NTAP': 25, 'ODFL': 7, 'CDW': 18, 'APTV': 0, 'INTU': 0, 'CRM': 1.0]</td>
<td>301.39</td>
<td>339,517</td>
<td>5.86</td>
</tr>
</tbody>
</table>
Table A.1 – continued from previous page

<table>
<thead>
<tr>
<th>Case</th>
<th>Set</th>
<th>Start date</th>
<th>End date</th>
<th>Budget</th>
<th>Min return</th>
<th>Methodology</th>
<th>Solution</th>
<th>Actual return</th>
<th>Risk</th>
<th>Vol, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['MPC': 23, 'HPQ': 8, 'STX': 9, 'ANET': 0, 'NTAP': 26, 'ODFL': 8, 'CDW': 18, 'APTV': 0, 'INTU': 0, 'CRM': 0]</td>
<td>300.34</td>
<td>337,196</td>
<td>5.86</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['ANET': 62, 'ODFL': 9, 'INTU': 0, 'CDW': 0, 'MPC': 1, 'STX': 1, 'HPQ': 4, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>300.02</td>
<td>894,269</td>
<td>9.48</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>Classical Gekko solver</td>
<td>['ANET': 59, 'ODFL': 11, 'INTU': 0, 'CDW': 0, 'MPC': 0, 'STX': 0, 'HPQ': 3, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>300.57</td>
<td>852,102</td>
<td>9.24</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>300 (3%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['ANET': 59, 'ODFL': 11, 'INTU': 0, 'CDW': 0, 'MPC': 1, 'STX': 0, 'HPQ': 0, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>300.31</td>
<td>853,486</td>
<td>9.27</td>
</tr>
</tbody>
</table>
Table A.1 – continued from previous page

<table>
<thead>
<tr>
<th>Case</th>
<th>Set</th>
<th>Start date</th>
<th>End date</th>
<th>Budget</th>
<th>Min return</th>
<th>Methodology</th>
<th>Solution</th>
<th>Actual return</th>
<th>Risk</th>
<th>Vol, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>500 (5%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum) ['DVN': 3, 'MRO': 0, 'MOS': 4, 'OXY': 0, 'MRNA': 0, 'TRGP': 4, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>762.99</td>
<td>875,660</td>
<td>9.36</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>500 (5%)</td>
<td>Classical Gekko solver ['DVN': 0, 'MRO': 0, 'MOS': 37, 'OXY': 0, 'MRNA': 0, 'TRGP': 57, 'TSLA': 1, 'NUE': 22, 'FANG': 0, 'ENPH': 0.0]</td>
<td>773.13</td>
<td>828,667</td>
<td>9.10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>500 (5%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s ['DVN': 1, 'MRO': 0, 'MOS': 37, 'OXY': 0, 'MRNA': 0, 'TRGP': 56, 'TSLA': 1, 'NUE': 22, 'FANG': 0, 'ENPH': 0]</td>
<td>773</td>
<td>826,116</td>
<td>9.09</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>500 (5%)</td>
<td>D-Wave CQM hybrid solver with time_limit=30s ['DVN': 0, 'MRO': 0, 'MOS': 33, 'OXY': 0, 'MRNA': 0, 'TRGP': 58, 'TSLA': 1, 'NUE': 23, 'FANG': 0, 'ENPH': 0]</td>
<td>769.32</td>
<td>825,748</td>
<td>9.09</td>
<td></td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Min return</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>800 (8%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['DVN': 19, 'MRO': 0, 'MOS': 47, 'OXY': 0, 'MRNA': 0, 'TRGP': 43, 'TSLA': 0, 'NUE': 19, 'FANG': 1, 'ENPH': 0]</td>
<td>802.08</td>
<td>854,184</td>
<td>9.24</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>800 (8%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['DVN': 20, 'ENPH': 1, 'FANG': 0, 'MOS': 44, 'MRNA': 0, 'MRO': 0, 'NUE': 20, 'OXY': 0, 'TRGP': 42, 'TSLA': 0.0]</td>
<td>800.6</td>
<td>847,432</td>
<td>9.21</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>800 (8%)</td>
<td>Classical Gekko solver</td>
<td>['DVN': 21, 'MRO': 0, 'MOS': 43, 'OXY': 0, 'MRNA': 0, 'TRGP': 42, 'TSLA': 0, 'NUE': 20, 'FANG': 0, 'ENPH': 1.0]</td>
<td>800.83</td>
<td>846,986</td>
<td>9.20</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>800 (8%)</td>
<td>D-Wave CQM hybrid solver with time_limit=35</td>
<td>['DVN': 20, 'MRO': 0, 'MOS': 44, 'OXY': 0, 'MRNA': 0, 'TRGP': 44, 'TSLA': 0, 'NUE': 19, 'FANG': 0, 'ENPH': 1]</td>
<td>800.37</td>
<td>846,935</td>
<td>9.20</td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Min return</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
<td>--------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['DVN': 4, 'MRO': 1, 'MOS': 5, 'OXY': 0, 'MRNA': 0, 'TRGP': 2, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>82.86</td>
<td>9,052</td>
<td>9.58</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['DVN': 3, 'MRO': 0, 'MOS': 4, 'OXY': 0, 'MRNA': 0, 'TRGP': 4, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>81.17</td>
<td>8,628</td>
<td>9.32</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>Classical Gekko solver</td>
<td>['DVN': 3, 'MRO': 0, 'MOS': 4, 'OXY': 0, 'MRNA': 0, 'TRGP': 4, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>81.17</td>
<td>8,628</td>
<td>9.32</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['DVN': 4, 'MRO': 1, 'MOS': 5, 'OXY': 0, 'MRNA': 0, 'TRGP': 2, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>82.86</td>
<td>9,052</td>
<td>9.58</td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Min return</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>-----------</td>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['DVN': 3, 'MRO': 0, 'MOS': 4, 'OXY': 0, 'MRNA': 0, 'TRGP': 4, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>81.17</td>
<td>8,628</td>
<td>9.32</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>80 (8%)</td>
<td>Classical Gekko solver</td>
<td>['DVN': 3, 'MRO': 0, 'MOS': 4, 'OXY': 0, 'MRNA': 0, 'TRGP': 4, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>81.17</td>
<td>8,628</td>
<td>9.32</td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
<td>time, s</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>[&quot;MPC&quot;: 7, &quot;HPQ&quot;: 6, &quot;STX&quot;: 1, &quot;ANET&quot;: 1, &quot;NTAP&quot;: 0, &quot;ODFL&quot;: 0, &quot;CDW&quot;: 0, &quot;APTV&quot;: 0, &quot;INTU&quot;: 0, &quot;CRM&quot;: 0]</td>
<td>45.17</td>
<td>7,295</td>
<td>8.55</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>[&quot;MPC&quot;: 7, &quot;HPQ&quot;: 6, &quot;STX&quot;: 1, &quot;ANET&quot;: 1, &quot;NTAP&quot;: 0, &quot;ODFL&quot;: 0, &quot;CDW&quot;: 0, &quot;APTV&quot;: 0, &quot;INTU&quot;: 0, &quot;CRM&quot;: 0]</td>
<td>45.17</td>
<td>7,295</td>
<td>8.55</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>1,000</td>
<td>Classical Gekko solver</td>
<td>[&quot;MPC&quot;: 7, &quot;HPQ&quot;: 6, &quot;STX&quot;: 1, &quot;ANET&quot;: 1, &quot;NTAP&quot;: 0, &quot;ODFL&quot;: 0, &quot;CDW&quot;: 0, &quot;APTV&quot;: 0, &quot;INTU&quot;: 0, &quot;CRM&quot;: 0]</td>
<td>45.17</td>
<td>7,295</td>
<td>8.55</td>
<td>1.58</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>[&quot;MPC&quot;: 85, &quot;HPQ&quot;: 77, &quot;STX&quot;: 1, &quot;ANET&quot;: 0, &quot;NTAP&quot;: 0, &quot;ODFL&quot;: 0, &quot;CDW&quot;: 0, &quot;APTV&quot;: 0, &quot;INTU&quot;: 0, &quot;CRM&quot;: 0]</td>
<td>474.62</td>
<td>880,496</td>
<td>9.39</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>[&quot;MPC&quot;: 97, &quot;HPQ&quot;: 45, &quot;STX&quot;: 1, &quot;ANET&quot;: 1, &quot;NTAP&quot;: 0, &quot;ODFL&quot;: 0, &quot;CDW&quot;: 0, &quot;APTV&quot;: 0, &quot;INTU&quot;: 0, &quot;CRM&quot;: 0]</td>
<td>481.15</td>
<td>999,116</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>
Table A.2 – continued from previous page

<table>
<thead>
<tr>
<th>Case</th>
<th>Set</th>
<th>Start date</th>
<th>End date</th>
<th>Budget</th>
<th>Methodology</th>
<th>Solution</th>
<th>Actual return</th>
<th>Risk</th>
<th>Vol, %</th>
<th>time, s</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>1</td>
<td>2016-11-08</td>
<td>2022-05-01</td>
<td>10,000</td>
<td>Classical Gekko solver</td>
<td>[‘MPC’: 96, ‘HPQ’: 53, ‘STX’: 0, ‘ANET’: 0, ‘NTAP’: 0, ‘ODFL’: 0, ‘CDW’: 0, ‘APTV’: 0, ‘INTU’: 0, ‘CRM’: 0.0]</td>
<td>482.09</td>
<td>992,648</td>
<td>9.96</td>
<td>8.5</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>[‘ANET’: 5, ‘ODFL’: 1, ‘INTU’: 0, ‘CDW’: 0, ‘MPC’: 0, ‘STX’: 0, ‘HPQ’: 4, ‘APTV’: 0, ‘CRM’: 0, ‘NTAP’: 0.0]</td>
<td>29</td>
<td>6857.58</td>
<td>8.29</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>[‘ANET’: 5, ‘ODFL’: 1, ‘INTU’: 0, ‘CDW’: 0, ‘MPC’: 0, ‘STX’: 0, ‘HPQ’: 4, ‘APTV’: 0, ‘CRM’: 0, ‘NTAP’: 0.0]</td>
<td>29</td>
<td>6857.58</td>
<td>8.29</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>Classical Gekko solver</td>
<td>[‘ANET’: 5, ‘ODFL’: 1, ‘INTU’: 0, ‘CDW’: 0, ‘MPC’: 0, ‘STX’: 0, ‘HPQ’: 4, ‘APTV’: 0, ‘CRM’: 0, ‘NTAP’: 0.0]</td>
<td>29</td>
<td>6857.58</td>
<td>8.29</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Table A.2 – continued from previous page

<table>
<thead>
<tr>
<th>Case</th>
<th>Set</th>
<th>Start date</th>
<th>End date</th>
<th>Budget</th>
<th>Methodology</th>
<th>Solution</th>
<th>Actual return</th>
<th>Risk</th>
<th>Vol, %</th>
<th>time, s</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['ANET': 66, 'ODFL': 6, 'INTU': 0, 'CDW': 3, 'MPC': 1, 'STX': 1, 'HPQ': 1, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>299.61</td>
<td>953,682</td>
<td>9.78</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['ANET': 68, 'ODFL': 6, 'INTU': 1, 'CDW': 0, 'MPC': 0, 'STX': 0, 'HPQ': 1, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>303</td>
<td>996,490</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>Classical Gekko solver</td>
<td>['ANET': 67, 'ODFL': 8, 'INTU': 0, 'CDW': 0, 'MPC': 0, 'STX': 0, 'HPQ': 0, 'APTV': 0, 'CRM': 0, 'NTAP': 0]</td>
<td>304.34</td>
<td>983,199</td>
<td>9.94</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['DVN': 5, 'MRO': 2, 'MOS': 5, 'OXY': 0, 'MRNA': 0, 'TRGP': 1, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>84.91</td>
<td>9,663</td>
<td>9.84</td>
<td>4.7</td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
<td>time, s</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['DVN': 5, 'MRO': 2, 'MOS': 5, 'OXY': 0, 'MRNA': 0, 'TRGP': 1, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>84.91</td>
<td>9,663</td>
<td>9.84</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>1,000</td>
<td>Classical Gekko solver</td>
<td>['DVN': 5, 'MRO': 2, 'MOS': 5, 'OXY': 0, 'MRNA': 0, 'TRGP': 1, 'TSLA': 0, 'NUE': 2, 'FANG': 0, 'ENPH': 0]</td>
<td>84.91</td>
<td>9,663</td>
<td>9.84</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with default time_limit (minimum)</td>
<td>['DVN': 51, 'MRO': 27, 'MOS': 33, 'OXY': 0, 'MRNA': 0, 'TRGP': 15, 'TSLA': 2, 'NUE': 17, 'FANG': 0, 'ENPH': 2]</td>
<td>842.36</td>
<td>9,663</td>
<td>9.95</td>
<td>4.7</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with time_limit=20s</td>
<td>['DVN': 70, 'MRO': 9, 'MOS': 43, 'OXY': 0, 'MRNA': 0, 'TRGP': 9, 'TSLA': 0, 'NUE': 18, 'FANG': 0, 'ENPH': 0]</td>
<td>868.58</td>
<td>999,206</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Case</td>
<td>Set</td>
<td>Start date</td>
<td>End date</td>
<td>Budget</td>
<td>Methodology</td>
<td>Solution</td>
<td>Actual return</td>
<td>Risk</td>
<td>Vol, %</td>
<td>time, s</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>Classical Gekko solver</td>
<td>['DVN': 75, 'MRO': 0, 'MOS': 49, 'OXY': 0, 'MRNA': 0, 'TRGP': 3, 'TSLA': 0, 'NUE': 17, 'FANG': 0, 'ENPH': 1.0]</td>
<td>873.45</td>
<td>997,240</td>
<td>9.99</td>
<td>0.17</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>2020-07-01</td>
<td>2022-04-30</td>
<td>10,000</td>
<td>D-Wave CQM hybrid solver with time_limit=30s</td>
<td>['DVN': 77, 'MRO': 1, 'MOS': 44, 'OXY': 0, 'MRNA': 0, 'TRGP': 3, 'TSLA': 0, 'NUE': 17, 'FANG': 0, 'ENPH': 2]</td>
<td>871.77</td>
<td>996,587</td>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>
Bibliography


