Received August 10, 2020; revised September 28, 2020; accepted October 7, 2020; date of publication October 13, 2020;
date of current version November 5, 2020.

Digital Object Identifier 10.1109/TQE.2020.3030314

Quantum Computing for Finance: State-of-the-Art and Future Prospects

DANIEL J. EGGER[1], CLAUDIO GAMBELLA[2], JAKUB MARECEK[2], SCOTT McFADDIN[3,4], MARTIN MEVISSEN[2], RUDY RAYMOND[5], ANDREA SIMONETTO[2], STEFAN WOERNER[1], AND ELENA YNDURAIN[6]

[1] IBM Quantum, IBM Research - Zurich, 8803 Rüschlikon, Switzerland
[2] IBM Quantum, IBM Research - Ireland, Dublin 15, Ireland
[3] IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights NY 10598, USA
[4] MIT-IBM Watson Artificial Intelligence Laboratory, Cambridge, MA 02142, USA
[5] IBM Quantum, IBM Research - Tokyo, Tokyo, 103-8510, Japan
[6] IBM Quantum, IBM Services, Yorktown Heights, NY 10598, USA

Corresponding author: Andrea Simonetto (andrea.simonetto@ibm.com)

Jakub Marecek, who has performed the present work while at IBM, is currently with the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, 121 35, Prague, Czech Republic

ABSTRACT This article outlines our point of view regarding the applicability, state-of-the-art, and potential of quantum computing for problems in finance. We provide an introduction to quantum computing as well as a survey on problem classes in finance that are computationally challenging classically and for which quantum computing algorithms are promising. In the main part, we describe in detail quantum algorithms for specific applications arising in financial services, such as those involving simulation, optimization, and machine learning problems. In addition, we include demonstrations of quantum algorithms on IBM Quantum back-ends and discuss the potential benefits of quantum algorithms for problems in financial services. We conclude with a summary of technical challenges and future prospects.

INDEX TERMS Financial management, machine learning algorithms, optimization, quantum computing, simulation.

I. INTRODUCTION

In the financial services industry, there are many computationally challenging problems arising in applications across asset management, investment banking, and retail and corporate banking. Quantum computing holds the promise of revolutionizing how we solve such computationally challenging problems. With the first noisy quantum devices—leveraging the principles of quantum mechanics—available publicly today (see, e.g., [1] and [2]), the applicability of quantum computing for problems in finance and demonstrating Quantum Advantage in first applications are active topics of current research [3]–[12].

In this article, we provide an introduction to quantum computing and the necessary foundational concepts to understand this new technology and its implications to the financial services industry [13]. We extend previous summaries [3], [14], [15] in multiple directions, as follows. First, we review the main algorithms, the benefits they bring as well as the technical challenges they pose, and how to approach problems from a quantum perspective. We then also highlight the economic benefits that applying quantum computing may bring to financial institutions in improving operations, revenues, and quality. Algorithms are categorized based on the type of problems they solve and mapped to the financial solutions they can be applied to. We showcase real-life examples of using quantum computing algorithms, explaining how the problems are solved and the solutions obtained. Overall, we offer a holistic practical guide to quantum computing and its applicability to financial problems for financial institutions in banking, financial markets, and insurance.

All computing systems rely on a fundamental ability to store and manipulate information. Today's classical computers manipulate individual bits, which store information as binary 0 and 1 states. Millions of bits work together to process and display information with a *speed* that everyone is familiar with on smartphones, laptops, and the servers in the cloud. Quantum computers use the physical phenomena of nature to manipulate information via quantum mechanics. At this fundamental level, we have quantum bits, or qubits. Unlike a bit that has to be a 0 or a 1 (or, their probabilistic combination), a qubit can be in a complex-value-weighted combination of states called a *superposition*. Multiple qubits may be purposefully *entangled* into linear combinations of (complex-valued-weighted) states across the qubits, which "correlates" them, so their quantum state cannot be described independently, i.e., entangled states. And quantum *interference* allows us to bias the measurement of a qubit toward a desired state; thus, controlling the probability a system of qubits collapses into particular measurement states. Quantum computers are not a replacement for classical ones. They complement the traditional systems by possibly being able to solve some forms of intractable problems that *blow up*, or become extremely large or time-consuming during computation. Remarkable progress in the control and construction of quantum computing hardware in recent years has led to the development of systems with tens of physical qubits, which in the absence of quantum error correction (QEC) are dubbed noisy quantum processors [16]. This has sparked strong interest in the pursuit of Quantum Advantage—the exploration of computational tasks which a quantum computer solves faster than a classical one. An example of a task for which Quantum Advantage has been proven was introduced in [4] This is also a step on the way to the development of fault-tolerant universal quantum computers (FTQCs), which require error correction [17]–[19] and which will allow for arbitrary quantum algorithms backed by theory that proves quantum speedup compared to classical algorithms.

The quantum computing hardware pursued by IBM involves superconducting quantum circuits [20]. The fundamental building blocks of the hardware are Josephson-junction-based qubits called transmons. When cooled to ca. 10 mK, these transmons behave as artificial atoms, where the two lowest energy levels may be employed as the computational 0 and 1 states. Over the past two decades, progress with superconducting qubit technology has been driven by tremendous improvements in coherence, control, and fabrication capabilities. A metric that has been proposed and employed by IBM to measure progress in development in quantum computing is Quantum Volume [21]. Quantum Volume indicates the relative complexity of a problem that can be solved on a quantum computer, and it depends on a number of factors such as number of qubits, coherence time, measurement errors, device crosstalk, circuit compiler efficiency, and others. IBM currently has more than 20 superconducting processors available via the cloud with up to 65 qubits and put forward a roadmap for scaling quantum technology in the years ahead [22].

In the near-term future, quantum computers will continue to rely on noisy qubits with relatively high error rates and limited coherence times. In this era of noisy quantum devices [16], we will be confidently in the realm of Quantum Advantage once we are able to solve a good number of significant real-world problems more efficiently with the help of quantum devices than compared with solving them on a classical computers only. The problem classes where we expect to demonstrate advantage first include: 1) modeling physical processes of nature; 2) obtaining better solutions to optimization problems; and 3) finding better patterns within machine learning (ML) processes, since first promising proposals for near-term compatible quantum heuristics exist [23]–[25].

At a basic level, any solution built on a quantum computer will comprise three fundamental steps.

1) *The Loading Step:* The solution must load its data from a classical computer into the quantum computer in a small number of steps. This results in a superposition, which combines the fundamental states described above in a way that reflects the dataset being loaded. While loading data into a computer are often ignored as a triviality in classical computing, quantum data loading can be a substantial aspect of solution design. While there are loading techniques that are linear in the size of the dataset, this can often eclipse the coherence times of the physical superposition, and techniques are often employed to reduce the input data before loading or encode the data into the computational steps.
2) *The Compute Step:* Once data are loaded, the solution must rapidly perform a computation on the loaded data within the quantum computer. This involves manipulation of the qubits in a manner that changes the fundamental states in a way that reflects the outcome of a desired computation. There is an increasing body of research into quantum algorithms that solve problems that are considered computationally difficult or intractable classically. Quantum computations often compute their results as an approximation to an optimal value within a high-dimensional search space and often exploit the nature of quantum superpositions to simultaneously consider vast numbers of possibilities. The computed "output" superposition reflects a probabilistic distribution of possible outcomes, with the preferred outcomes associated with higher probabilities in the distribution.
3) *The Measurement Step:* The measurement step makes an observation of the computed "output" superposition and reports it back to a classical computer. However, the act of observation "snaps" the superposition back to a basis state, so the computation step must be designed in such a way that the solution is often encoded in a narrow decision space. Quantum algorithms are often repeated multiple times, with each repetition called a *shot*. Each shot reports an output among the distribution of possible outputs from the compute step. With multiple repetitions, a probabilistic picture emerges of which output has received the highest probability within the superposition, and an "answer" may be read. Typically, hundreds or thousands of shots are used. It should be noted that the higher the number of qubits, the higher requirements on precision, and hence, the higher the requirements on the number of measurements.

TABLE 1. Segments of the financial services industry

Banking	Financial Markets	Insurance
Monetary Authorities	Commodities	Health Insurance
Retail Banking	Stock Exchanges	Property & Casualty Insurance
Commercial Banking	Bond Markets	Life Insurance & Annuity
Investment Banking	Money Markets	Reinsurance
Non-Banking FIs	Derivatives	

II. PROBLEMS IN FINANCIAL SERVICES

Financial services are a forward-looking industry that has always been in the lookout to leverage new technologies to increase profits. Broadly, this industry covers three vertical sectors (cf. also Table 1 for segments).

1) *Banking:* Banking products are mainly bank accounts, investments, and loans for commercial and retail customers. Their main challenges are to balance cash with interest rates, while controlling threats related to liquidity, fraud, money laundry, or nonperforming loans (NPLs).
2) *Financial Markets:* They are focused mainly on future gains and the marketplace to sell and buy assets by dealers, exchanges, brokers, or clearing houses. Their main challenges are to manage geographic time zones, immediacy needs, and counterparty risk.
3) *Insurance:* Health insurance, automobile and property, life insurance and annuity, and reinsurance. The main challenge here is to maximize premiums and manage threats related to unplanned risks, such as catastrophes or market crashes.

The digital financial services revolution is full blast disrupting the industry and opening the door to new players threatening the current status quo [26]: FinTechs and InsurTechs with new digital-style offerings, RegTech with automatized regulation processes, and new competition from other industry corporations that can now offer digital financial services to their customers. In addition, clients are demanding customized offerings based on their behavioral data, for example, personalized insurance premiums based on their life events or targeted loan offers for smaller customer segments. The regulatory environment poses an operational challenge by requiring risk mitigation and strict compliance. All these trends have created new financial services experiences: smooth global supply chains for trade finance asset management, Artificial Intelligence (AI) augmentation with trusted decision making for investment banking, and banking platforms that become finance "as-a-service."

In this article, we consider the regulatory framework given by the Basel III rules, which is being implemented by regulators worldwide. Under Basel III, a key performance indicator for both the regulator and regulated entities is capital adequacy ratio, which is a ratio of the combined tier-1 and tier-2 capital and risk-weighted holdings. In particular, the minimum is 8%, while the required ratio is 10.5%.

Moreover, the liquidity coverage ratio (LCR) reform [27], with the aim of improving short-term resilience, promotes the holding of unencumbered high-quality liquid assets (HQLA), whose amounts are tested in the so-called 30 calendar day liquidity stress scenario. By January 1, 2019, the LCR, i.e., the proportion of the value of the stock of HQLA in stressed conditions to total net cash outflows in the same scenario, has risen to 100%. Within HQLA, "Level 1" assets include cash and certain state-backed securities, as well as select other safe assets. "Level 2" include further state-backed securities and bonds of nonstate and nonbank entities with long-term rating AA- or better. "Level 2" assets can only comprise up to 40% of the stock of HQLA. Furthermore, the same LCR reform introduced diversification requirements on the stock of HQLA. These extensions complicate both risk assessment and portfolio management problems considerably.

Indeed, many problems, such as risk management and portfolio management, rely on various risk measures that we now introduce for future reference. Volatility captures the risk in a portfolio of assets. It is the standard deviation of the returns, which can be calculated from the variance $\omega^\top \Sigma \omega$. Here, ω is a (column) vector made of the weights of each asset in the portfolio and Σ is the covariance matrix of the returns of the assets.

Risk management often uses value at risk (VaR) as a risk measure. VaR_α of a random variate X is the $1-\alpha$ quantile of the loss distribution $Y = -X$. VaR is, therefore, the minimal γ such that the probability that X exceeds γ is α, i.e.

$$\text{VaR}_\alpha(X) := -\inf\{\gamma \text{ such that } F_X(\gamma) > \alpha\} \tag{1}$$

where $F_X(x)$ is the cumulative distribution function (CDF) of X. Since VaR is a quantile, it has the shortcoming that it is not sensitive to extreme losses in the tail of X. The conditional value at risk (CVaR) is, therefore, often used as an additional risk metric. CVaR_α, sometimes also called expected shortfall, is the expectation value of all losses up to the VaR_α. In the financial services challenges discussed in the remainder of this article, we are assuming a market environment operating under the Basel III regulations.

Planned updates to the framework, upcoming under Basel IV, focus on the approach to calculate risk-weighted assets (RWAs) regardless of risk type. Banks will need to change their projection models toward forward-looking statements, which are statements that are not solely based on historical facts and, therefore, have more assumptions. This will require more scenario building to comply with the new requirements for capital lower limits (72.5% "output floor"), credit risk with common approach rather than internal, market risk sensitivity-driven analysis as a standard, and operational risk measured by "unadjusted business indicator" leveraging historical loss data. Overall Basel IV will reshape banks' trading activities and portfolio structures.

In this article, the focus is in three areas of financial services, where problems challenging for classical computers arise today:

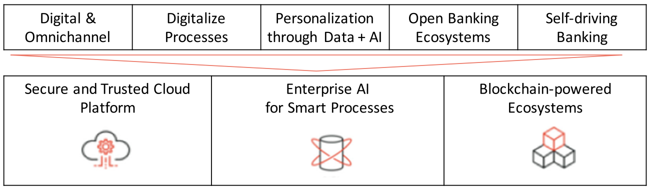

Digital & Omnichannel
Digitalize Processes
Personalization through Data + AI
Open Banking Ecosystems
Self-driving Banking
Secure and Trusted Cloud Platform
Enterprise AI for Smart Processes
Blockchain-powered Ecosystems

FIGURE 1. Transformative technologies in support of digital themes in financial services [28].

1) asset management;
2) investment banking;
3) retail and corporate banking.

A few examples of transformative technologies leveraged in these areas include the use of AI for Asset Management through bots as the new user interfaces, the adoption of algorithmic trading that changed the stock market with High Frequency Trading for Investment Banking, and the increased use of mobile devices for payments impacting financial transactions in Retail Banking (see also Fig. 1).

The adaption of new technologies and evolution of financial services can be traced in waves leading to a digital future. The first wave focused on mobile adoption to engage customers. This was followed by the exploration of new business models to leverage new data sources and the threat from technology providers. The third current wave is driven by a combination of challenges, including cost pressures, technology disruptions, and regulatory changes (see Fig. 2). This is causing a fundamental redefinition of the financial institution, its internal operations, and how it engages externally. Further background is provided in [28]. Within this context, the future benefits of quantum computing for businesses could be measured across a set of key business metrics.

1) Reduce regulatory penalty costs or avoidable human labor.
2) Improve customer satisfaction and brand perception.
3) Increase customer interaction and financial activity.
4) Reduce capital levels and improve cash flow.

In the following sections, we group specific problems arising in the financial services where classical computers face challenges or are insufficient, in three classes: *simulation*, *optimization*, and *ML*; we introduce the quantum algorithms applicable to them and discuss results obtained on IBM Quantum back-ends for some specific problems. Each quantum algorithm is applied on tasks and calculations that affect one or more phases of the customer life cycle, shown in Fig. 3, and we describe the specific business benefits it may bring.

III. SIMULATION

In this section, we discuss simulation problems, where quantum computing may be beneficial. There are simulation problems at each stage of the customer life cycle (see Fig. 3).

1) *Customer Identification:* Obtain new revenue sources for value-added services such as derivative pricing using sophisticated quantum computing algorithms. This offering can help compensate for the monetary losses of MiFIDII new transparency measures in trading, estimated to cost $240 million [29].
2) *Financial Products:* Better manage VaR and the economic capital requirement (ECR) providing more accurate estimates to improve liquidity management by actively managing the balance sheet, increase capital to maintain a 7% equity capital ratio to its riskier assets, and avoid Basel-III-related compliance fines [30] that are up to 10% of Bank's turnover (revenue).
3) *Monitor Transactions:* Allow for a more precise approach to incorporating market volatility into an institution's Tier 1 reporting [31], optimizing RWA results through a much more accurate/precise calculation process.
4) *Customer Retention:* Improve risk analysis for the new net stable funding ratio time-frame requirements that will impact the cost of doing business for Prime Brokers and hedge funds. The stable funding for securities Lending Transactions shifts from 0% to 10% for Level 1 collateral and to 15% for other collateral [32].

Overall, Basel regulation implementation and the new needs of risk management is of mandatory interest for financial institutions, with an estimated technology cost between EUR 45 million and EUR 70 Million [33].

Simulation focuses on creating scenarios of potential outcomes, such as the impact of volatility on risk, evaluating asset values for pricing, or monitoring economic system impacts in the market.

One key task in the finance industry where simulation is crucial is the pricing of financial instruments and estimating their risk. For example, the buyers and sellers of complex financial instruments gain a competitive advantage when they can price such instruments with a better accuracy than their competition. Regulations such as Basel III require banks to perform stress tests and to hold an amount of capital that depends on their RWAs. However, pricing and estimating the risk of many financial instruments is computationally intensive. Analytical models are often too simplistic to capture the complex dependencies between financial instruments or cannot take into account some of their features such as path dependence. Monte Carlo (MC) simulations are, therefore, used to estimate risk metrics, to price financial instruments, and to perform scenario analysis that can be used in stress tests. In an MC simulation, M samples are drawn from the model input distributions and are used to construct an estimation of a quantity of interest. The confidence interval of this estimation scales as $\mathcal{O}(1/\sqrt{M})$, making MC computationally intensive. For instance, decreasing the size of the confidence interval by an order of magnitude requires increasing the computational cost by a factor of 100. In this section, we briefly discuss MC for option pricing and risk calculations

	Engaging Customers	New Business Models	Re-define the Bank's internal operations and external role
Drivers	Mobile adoption, new digital touchpoints	New data sources, threat from tech providers	Internal: Need for OPEX reduction External: Banking ecosystems
Key Business Areas	Customer interactions	SMB Lending, Asset Management, Usage-based insurance	Risk, Compliance, KYC, AML, Financial ecosystems
Tech Solutions	AI	AI, IoT	AI, IoT, Blockchain, Cyber-security, API

FIGURE 2. **Finance industry transformation in waves driving to a digital future [28].**

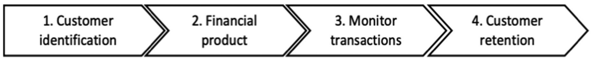


FIGURE 3. **Customer life cycle conceptual design. Here, a customer can be a corporation, fund, financial institution, government, or individual.**

and then discuss how such tasks can be performed on quantum computers using amplitude estimation (AE) (see Section III-A).

(1) *Option Pricing:* Options are financial derivative contracts that give the buyer the right, but not the obligation, to buy (call option) or sell (put option) an underlying asset at an agreed-upon price (strike) and time-frame (exercise window). In their simplest form, the strike price is a fixed value and the time frame is a single point in time, but exotic variants may be defined on more than one underlying asset; the strike price can be a function of several market parameters and could allow for multiple exercise dates [34]. Options provide investors with a vehicle to profit by taking a view on the market or exploit arbitrage opportunities and are core to various hedging strategies. As such, understanding their properties is a fundamental objective of financial engineering. Due to the stochastic nature of the parameters options are defined on, calculating their fair value can be an arduous task. Analytical models exist for the simplest types of options [35], but the simplifying assumptions on the market dynamics required for the models to provide closed-form solutions often limit their applicability [36]. Hence, more often than not, numerical MC simulations are employed for option pricing since they are flexible and can generically handle stochastic parameters [37], [38]. Option pricing with MC generally proceeds by simulating many paths of the time evolution undergone by the underlying assets to build a distribution of the option payoff at maturity. The option price is then obtained by discounting the expected value of this distribution. Classical MC methods require extensive computational resources to provide accurate option price estimates, particularly for complex options. Because of the widespread use of options in the finance industry, accelerating the methodology to price them can significantly impact the operations of a financial institution.

(2) *Risk Management:* Risk management plays a central role in the financial system. It allows companies, institutions, and individuals to avoid monetary losses and grow their business. Financial risk, which comes in many forms such as credit risk, liquidity risk, and market risk, is often estimated using models and simulations. The accuracy of these models has a direct impact on the operations of the entity using them. For instance, the capital requirements imposed on banks under the Basel accords depend on the accuracy of risk models [39]. Therefore, banks with more accurate models can make better use of their capital. VaR [40], a quantile of the loss distribution, is a widely used risk metric. For example, the Basel III regulations require banks to perform stress tests using VaR [41]. MC simulations are the method of choice to determine VaR and CVaR. They are done by building a model and computing the loss/profit distribution for M different realizations of the model input parameters. Many different runs are needed to achieve a representative distribution of the loss/profit distribution. Classical attempts to improve the performance are variance reduction or quasi-MC techniques [38], [42], [43]. The first aims at reducing the constants while not changing the asymptotic scaling, whereas the latter improves the asymptotic behavior, but only works well for low-dimensional problems.

In Section III-A, we discuss how quantum AE can provide a quadratic speedup over classical MC simulations and highlight the steps needed to calculate VaR in Section III-B. We then employ these methods in the context of credit risk, as already discussed in [5] and summarized here in Section III-C. For a detailed discussion on how quantum AE can provide an advantage for options pricing, we refer to [6], [7], and [44].

A. QUANTUM AE

AE is a quantum algorithm that can estimate a parameter a with a convergence rate $\mathcal{O}(1/M)$, where M is the number of quantum samples. This corresponds to a quadratic speedup compared to classical MC. AE is based on a unitary operator $\mathcal{A}$ acting on a register of $(n+1)$ qubits such that $\mathcal{A}\left|0\right\rangle_{n+1} = \sqrt{1-a}\left|\psi_0\right\rangle_n\left|0\right\rangle + \sqrt{a}\left|\psi_1\right\rangle_n\left|1\right\rangle$ for some normalized states $\left|\psi_0\right\rangle_n$ and $\left|\psi_1\right\rangle_n$, where $a \in [0, 1]$ is unknown. AE allows the efficient estimation of a, i.e., the probability of measuring $\left|1\right\rangle$ in the last qubit [45]. This is done using an operator Q defined as

$$Q = \mathcal{A}(\mathbb{I} - 2\left|0\right\rangle_{n+1}\left\langle 0\right|_{n+1})\mathcal{A}^{\dagger}(\mathbb{I} - 2\left|\psi_0\right\rangle_n\left|0\right\rangle\left\langle\psi_0\right|_n\left\langle 0\right|)$$

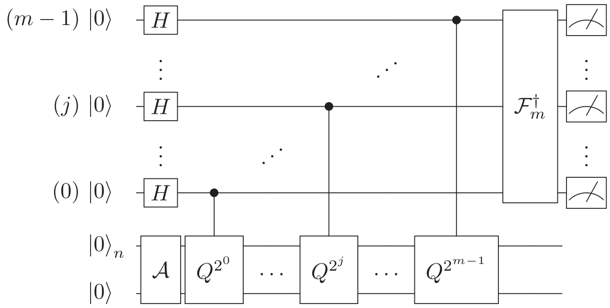


FIGURE 4. Quantum circuit for AE as introduced in [45]. ***H*** is the Hadamard gate and $\mathcal{F}_m^\dagger$ denotes the QFFT^{-1} on m qubits.

where $\mathbb{I}$ denotes the identity operator, and quantum phase estimation [46] to approximate certain eigenvalues of Q. AE requires m additional qubits and $M = 2^m$ (controlled) applications of Q. The m qubits are first put into equal superposition, i.e., $|0\rangle_m \rightarrow 1/\sqrt{M}\sum_{k=0}^{M-1}|k\rangle_m$, by applying Hadamard gates, which are single-qubit gates that perform $|0\rangle \rightarrow (|0\rangle + |1\rangle)/\sqrt{2}$. Then, the m qubits are used to control the applied power of Q, which leads to the state

$$1/\sqrt{M}\sum_{k=0}^{M-1}|k\rangle_m Q^k \mathcal{A}\,|0\rangle_{n+1} \tag{2}$$

where each Q^k imparts the phase $e^{-2ik\theta_a}$ and $e^{+2ik\theta_a}$, respectively, to the two eigenstates of $\mathcal{Q}$ that lie in the span of the states $|\psi_0\rangle_n\,|0\rangle$ and $|\psi_1\rangle_n\,|1\rangle$. Next, the inverse quantum Fourier transform (QFFT^{-1})

$$\mathcal{F}_M^\dagger : |k\rangle_m \rightarrow \frac{1}{\sqrt{M}}\sum_{y=0}^{M-1} e^{-2\pi i yk/M}\,|y\rangle_m \tag{3}$$

is applied to interfere these phases. Last, measuring the m qubits results in an integer $y \in \{0,\ldots,M-1\}$, which corresponds to the states for which the phases interfered constructively, i.e., $y \simeq M\theta_a/\pi$, see the circuit in Fig. 4. The integer y is then classically mapped to the estimator $\tilde{a} = \sin^2(y\pi/M) \in [0,1]$. The estimator $\tilde{a}$ satisfies

$$\begin{aligned}|a-\tilde{a}| &\leq \frac{2\sqrt{a(1-a)}\pi}{M} + \frac{\pi^2}{M^2}\\ &\leq \frac{\pi}{M} + \frac{\pi^2}{M^2} = O(M^{-1})\end{aligned} \tag{4}$$

with probability of at least $8/\pi^2$. This represents a quadratic speedup compared to the $O(M^{-1/2})$ convergence rate of classical MC methods [40].

Recently, several variants of AE have been proposed that simplify the required quantum circuits [8], [47], [48]. They leverage the same underlying structure, but allow us to remove the m additional qubits as well as the quantum Fourier transform. For a comparison of available AE variants, we refer to [48].

B. ESTIMATING VAR WITH AE

The discussion above shows that to efficiently estimate a parameter a, we need the corresponding operator $\mathcal{A}$. We make use of AE in finance by building the $\mathcal{A}$ operator for each quantity of interest, such as a risk measure, of a random variate X. We represent the distribution of X as an n-qubit quantum state

$$|\psi\rangle_n = \sum_{i=0}^{N-1}\sqrt{p_i}\,|i\rangle_n \tag{5}$$

by discretizing the outcomes of X and mapping them to $i \in \{0, \ldots, N-1\}$, where $N = 2^n$. Here, $p_i \in [0,1]$ is the probability of measuring the state $|i\rangle_n$, which is a binary representation of i. By adding an ancilla qubit to the n-qubit register and applying the operator

$$F : |i\rangle_n\,|0\rangle \rightarrow |i\rangle_n\left(\sqrt{1-f(i)}\,|0\rangle + \sqrt{f(i)}\,|1\rangle\right) \tag{6}$$

for some function $f(i)$, to the state $|\psi\rangle_n\,|0\rangle$ results in the quantum state

$$\sum_{i=0}^{N-1}\sqrt{p_i}\sqrt{1-f(i)}\,|i\rangle_n\,|0\rangle + \sum_{i=0}^{N-1}\sqrt{p_i}\sqrt{f(i)}\,|i\rangle_n\,|1\rangle\,. \tag{7}$$

By comparing this state to $\mathcal{A}\,|0\rangle_{n+1} = \sqrt{1-a}\,|\psi_0\rangle_n\,|0\rangle + \sqrt{a}\,|\psi_1\rangle_n\,|1\rangle$, we find that the probability of measuring $|1\rangle$ in the ancilla qubit is $a = \sum_{i=0}^{N-1} p_i f_i$. We can, therefore, obtain an estimate for VaR$_\alpha(X)$ by choosing

$$f(i) = \begin{cases} 1, & i \leq l\\ 0, & \text{otherwise}\end{cases} \tag{8}$$

for some level l. With this definition of $f(i)$, the probability of measuring $|1\rangle$ in the ancilla qubit is $\sum_{i=0}^{l} p_i = \mathbb{P}[X \leq l]$. With a binary search over l, we find the smallest l_α such that $\mathbb{P}[X \leq l_\alpha] \geq 1-\alpha$. The smallest l_α corresponds to the VaR. This estimation of VaR$_\alpha(X)$ has accuracy $\mathcal{O}(M^{-1})$, i.e., a quadratic speedup compared to classical MC methods (omitting the additional logarithmic complexity of the bisection search)—where $\mathcal{O}(\cdot)$ indicates the big-$\mathcal{O}$ notation. Here, we assumed a 1-D probability distribution. Probability distributions with more than one random variable can be loaded into a corresponding number of qubit registers [9]. Correlations between the random variables are then introduced by suitably entangling these qubit registers.

C. CREDIT RISK

The quantum method to calculate VaR, outlined in the previous sections, can be applied in the context of credit risk to determine the ECR associated with holding a portfolio of K loans [5]. The ECR is the amount of capital that needs to be held on the balance sheet to protect against unexpected losses. It is, therefore, defined as the VaR less the expected value of the loss distribution $\mathcal{L}$, i.e.

$$\text{ECR}_\alpha(\mathcal{L}) = \text{VaR}_\alpha(\mathcal{L}) - \mathbb{E}(\mathcal{L}) \tag{9}$$

where α is the confidence level. Estimating the VaR is often a computationally intensive task requiring classical MC simulation. However, quantum AE can achieve the same result with a quadratic speedup. We illustrate AE for a portfolio of K assets for which the multivariate random variable $(L_1, ..., L_K) \in \mathbb{R}^K_{\geq 0}$ denotes each possible loss associated with each asset. The expected value of the total loss $\mathcal{L} = \sum_{k=1}^K L_k$ is $\mathbb{E}[\mathcal{L}] = \sum_{k=1}^K \mathbb{E}[L_k]$. The VaR for a given confidence level $\alpha \in [0, 1]$ is defined as the smallest total loss that still has a probability greater than or equal to α, i.e.

$$\mathrm{VaR}_\alpha[\mathcal{L}] = \inf_{x \geq 0} \{x \mid \mathbb{P}[\mathcal{L} \leq x] \geq \alpha\}. \qquad (10)$$

Common values of α for ECR found in the finance industry are around 99.9%. We assign a Bernoulli random variable X_k to each asset to indicate if it is in a default state such that $L_k = \lambda_k X_k$, where $\lambda_k > 0$ is the loss given default (LGD). The probability that $X_k = 1$, i.e., a loss for asset k, is p_k. The expected loss of the portfolio $\mathbb{E}[\mathcal{L}] = \sum_{k=1}^K \lambda_k p_k$ is easier to evaluate than $\mathrm{VaR}_\alpha[\mathcal{L}]$, which usually requires an MC simulation. The defaults X_k are usually correlated which we model following a conditional independence scheme [49]. Given a realization z of a latent random variable $\mathcal{Z}$, the Bernoulli random variables $X_k \mid \mathcal{Z} = z$ are assumed independent, but their default probabilities p_k depend on z. We follow [49] and assume that $\mathcal{Z}$ has a standard normal distribution and that

$$p_k(z) = \Phi\left(\frac{\Phi^{-1}(p_k^0) - \sqrt{\rho_k} z}{\sqrt{1 - \rho_k}}\right) \qquad (11)$$

where p_k^0 denotes the default probability for $z = 0$, Φ is the CDF of the standard normal distribution, and $\rho_k \in [0, 1)$ determines the sensitivity of X_k to $\mathcal{Z}$. This scheme is similar to the one used for regulatory purposes in the Basel II (and following) internal rating-based approach to credit risk [41], [50] and is called the *Gaussian conditional independence model* [49].

To estimate VaR, we use AE to efficiently evaluate the CDF of the total loss, i.e., we will construct $\mathcal{A}$ such that $a = \mathbb{P}[\mathcal{L} \leq x]$ for a given $x \geq 0$, and apply a bisection search to find the smallest $x_\alpha \geq 0$ such that $\mathbb{P}[\mathcal{L} \leq x_\alpha] \geq \alpha$, which implies $x_\alpha = \mathrm{VaR}_\alpha[\mathcal{L}]$ [9].

Mapping the CDF of the total loss to a quantum operator $\mathcal{A}$ requires three steps. Each step corresponds to a quantum operator. First, $\mathcal{U}$ loads the uncertainty model. Second, $\mathcal{S}$ computes the total loss into a quantum register with n_S qubits. Finally, $\mathcal{C}$ flips a target qubit if the total loss is less than or equal to a given level x, which is used to search for VaR_α. Thus, we have $\mathcal{A} = \mathcal{CSU}$, and the corresponding circuit is illustrated in Fig. 5 on a high level.

We now discuss the operators $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}$ in more detail. The loading operator $\mathcal{U}$ loads the distribution of $\mathcal{Z}$ and prepares the X_k of each asset. To include correlations between the default events, we represent $\mathcal{Z}$ in a register with n_Z qubits. We use a truncated and discretized approximation with 2^{n_Z} values, where we consider an affine mapping $z_i = a_z i + b_z$ from $i \in \{0, ..., 2^{n_Z} - 1\}$ to the desired range of values of $\mathcal{Z}$.

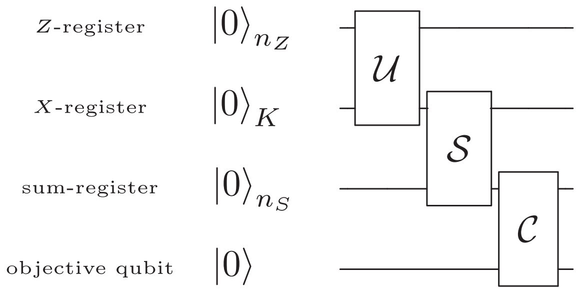


FIGURE 5. High-level circuit of the operator $\mathcal{A}$ used to evaluate the CDF of the total loss: the first qubit register with n_Z qubits represents $\mathcal{Z}$, the second qubit register with K qubits represents the X_k, the third qubit register with n_S qubits represents the sum of the losses, i.e., the total loss, and the last qubit is flipped to $|1\rangle$ if the total loss is less than or equal to a given x. The operators $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}$ represent the loading of uncertainty, the summation of losses, and the comparison to a given x, respectively.

Since $\mathcal{Z}$ follows a standard normal distribution, we can efficiently load it to a quantum register with controlled rotations [51]. We encode the X_k of each asset in the state of a corresponding qubit by applying to qubit k a Y-rotation $R_Y(\theta_p^k)$, controlled by the qubit register representing $\mathcal{Z}$, with angle $\theta_p^k(z) = 2\arcsin(\sqrt{p_k(z)})$. For simplicity, we use a first-order approximation[1] of $\theta_p^k(z)$ and include the affine mapping from z (a value of the normal distribution) to i (an integer represented by n_Z qubits), i.e., $\theta_p^k(z_i) \approx a_k i + b_k$. This prepares qubit k in the state $\sqrt{1 - p_k}\,|0\rangle + \sqrt{p_k}\,|1\rangle$ for which the probability to measure $|1\rangle$ is p_k. The $|1\rangle$ state of qubit k, thus, corresponds to a loss for asset k.

Next, we need to compute the resulting total loss for every realization of the X_k. Therefore, we use a weighted sum operator

$$\begin{aligned}\mathcal{S} : &|x_1, \ldots, x_K\rangle_K |0\rangle_{n_S} \\ &\mapsto |x_1, \ldots, x_K\rangle_K |\lambda_1 x_1 + \cdots + \lambda_K x_K\rangle_{n_S}\end{aligned} \qquad (12)$$

where $x_k \in \{0, 1\}$ denote the possible realizations of X_k. We set the size of the sum register to $n_S = \lfloor \log_2(\lambda_1 + \cdots + \lambda_K) \rfloor + 1$ qubits to represent all possible values of the sum of the losses given default λ_k, assumed to be integers. To implement $\mathcal{S}$, we apply a divide and conquer approach and first sum up pairs of assets, then pairs of the resulting sums, and so on until we computed the total sum. This implies that we start with a weighted-sum operator, discussed in detail in [7], and then continue with adder circuits [52] to iteratively combine the intermediate results.

Finally, we need an operator that compares a particular loss realization to a given x and then flips a target qubit from $|0\rangle$ to $|1\rangle$ if the loss is less than or equal to x. This operator is

[1] Higher order approximations of $\theta_p^k(z)$ can be implemented using multi-controlled rotations. Furthermore, by using quantum arithmetic, one could also compute $\theta_p^k(z)$ directly [9].

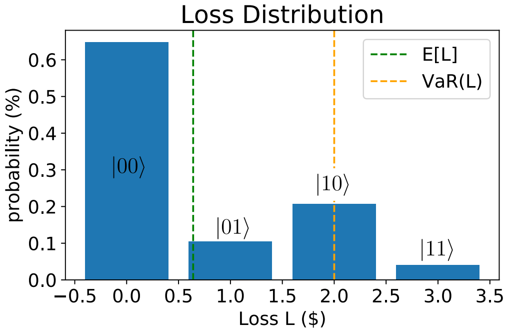


FIGURE 6. Loss distribution of the two-asset portfolio. The green- and orange-dashed lines show the expected value and the 95% VaR of the distribution, respectively.

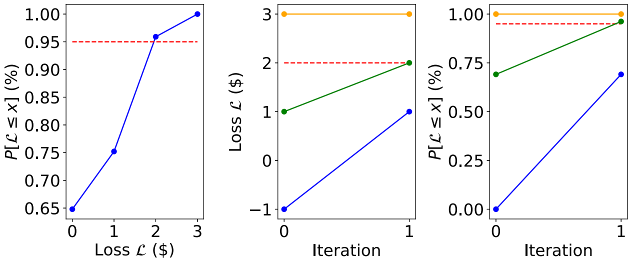


FIGURE 7. CDF (left) of the total loss $\mathcal{L}$ (blue), shown in Fig. 6, for a two-asset portfolio. The red-dashed line is the target VaR level $\alpha = 95\%$. Bisection search to compute VaR (middle and right) using $m = 4$ sampling qubits, i.e., 16 quantum samples: orange and blue represent the lower and upper bounds of the search interval, respectively, and green is the resulting midpoint. The red-dashed line shows the exact value.

defined by

$$\mathcal{C} : |i\rangle_{n_S} |0\rangle \mapsto \begin{cases} |i\rangle_{n_S} |1\rangle\,, & \text{if } i \leq x \\ |i\rangle_{n_S} |0\rangle\,, & \text{otherwise} \end{cases}. \qquad (13)$$

A fixed-value comparator, i.e., a comparator that takes a fixed value to compare to as a classical input, can be based on adder circuits [53]. Here, a logarithmic scaling can be achieved by adding a linear number of ancilla qubits.

We now show the performance of the quantum algorithm for an illustrative example with $K = 2$ assets. Each asset is defined by the triplet $(\lambda_k, p_k^0, \rho_k)$, i.e., the LGD, the default probability for $z = 0$, and the sensitivity of X_k to $\mathcal{Z}$, respectively. We chose $(1,\ 0.15,\ 0.1)$ and $(2,\ 0.25,\ 0.05)$ for asset 1 and 2, respectively, which results in the loss distribution shown in Fig. 6. This distribution is naturally encoded into the four states of two qubits, indicated by the state labels in Fig. 6, which illustrates how a qubit register can encode the exponential number of portfolio states as a quantum superposition. From the chosen λ_ks, it follows that the sum register requires $n_S = 2$ qubits to represent all possible losses. We represent $\mathcal{Z}$ with $n_Z = 2$ qubits. Thus, $\mathcal{A}$ is operating on seven qubits that represent this problem on a quantum computer, including the objective qubit.

To simulate our algorithm, we input the circuit for $\mathcal{A}$ to the AE subroutine implemented in *Qiskit* [54] and perform the bisection search using the result to find x_α. We use $m = 4$ evaluation qubits giving us 16 quantum samples. Our implementation requires one additional ancilla qubit to create $\mathcal{Q}$. Therefore, this experiment requires a total of 12 qubits that we simulate on classical computers using the `statevector_simulator` back-end provided by *Qiskit Aer*. Since $n_S = 2$, the bisection search requires at most two steps, as shown in Fig. 7. To ensure that the entire probability distribution is captured by the initial lower and upper bounds of the bisection search, we set them at losses of −1$ and 3$, respectively. The simulations, shown in Fig. 7, properly identify the 95% VaR, located at a loss of 2$, on the first iteration of the bisection search. We expect that the number of iterations needed in the bisection search will scale linearly with the number of assets in the portfolio.

We have investigated in [5] the quantum resources required for $K = 2^{20}$ assets, i.e., a portfolio of approximately 1 million assets. With $n_Z = 10$, $n_S = 30$, and $m = 10$, an FTQC would require approximately 37 million T/Toffoli gates. If we assume that error-corrected T/Toffoli gates can be executed in 10^{-4} s [55] and that the quantum phase estimation can be removed [8], which reduces the circuit depth by a factor of 2, we estimate a runtime of 30 min to estimate the VaR for a 1-million-asset portfolio [5]. This estimate may change as quantum computing technology advances.

D. DISTRIBUTION LOADING

Replacing an MC simulation with AE requires efficiently loading the distributions of the random variables in the model to the quantum computer to avoid diminishing the potential quantum advantage. This is feasible, e.g., for efficiently integrable probability distributions such as log-concave distributions for which the loading operator can be built from controlled rotations [51]. The loading of arbitrary states into quantum systems, however, requires exponentially many gates [56], making it inefficient to model arbitrary distributions as quantum gates. Since the distributions of interest are often of a special form, the limitation may be overcome by using quantum generative adversarial networks. These networks allow us to load a distribution using a polynomial number of gates [57].

E. SUMMARY

We have shown how AE can be used to estimate the ECR for a portfolio of loans. This results in a quadratic speedup (omitting the logarithmic cost of the bisection search in VaR) over classical MC simulations. The example also shows that the quantum circuit needed to implement $\mathcal{A}$ depends on the task at hand. Therefore, extending this work to other financial simulation tasks requires task-specific quantum circuits to implement $\mathcal{A}$.

IV. OPTIMIZATION

In this section, we discuss optimization problems, where quantum computing may be beneficial. As in the case of simulation, there are optimization problems at each stage of the customer life cycle (see Fig. 3).

1) *Customer Identification (and Assessment):* Improve Financial Supply Chain Efficiency [58] in procurement and payment focusing on customers and suppliers to increase to reduce working capital levels, enhance liquidity, minimize risk, and avoid late payments (47% of suppliers are paid late [59]).
2) *Financial Products:* Accelerate trade settlement capacity [10], [60] (i.e., from 45% transactions to 90%) to reduce associated capital requirements, systemic risk, and operational costs.
3) *Monitor Transactions:* Keep investment portfolios relevant by rebalancing aligned to market changes [61], while handling all the associated fees (taxes, commissions, etc.). This can reduce transaction costs by 50% and lead to $600k savings in trading costs for an example of four-asset $ 1billion portfolio [62].
4) *Customer Retention:* Improve the process of matching companies to potential buyers to avoid current customer churn toward automated investment banking. This can reduce current work losses; in 2015, 26% of the $1B+ merger and acquisitions were done without involvement of financial advisors [63].

Overall, investment banks are increasingly applying technology to automate the trading pipeline, hiring technologists, which are 20–40% job openings on the main investment banks [64]. Also, JPMorgan spends the most a year on technology with $10.8 billion [65].

Many financial services firms may want to take actions that result in the best possible outcome for a given goal. In the language of mathematical optimization, finding the best decision or action with respect to maximizing or minimizing given goals or objectives is cast as maximizing or minimizing an objective function in a decision variable, subject to constraints, often given again by functions of a decision variable. This has extensive applications, e.g., finding the best supply-chain route for delivery, determining the best investment strategy for a portfolio of assets, or increasing productivity with a number of fixed resources in operations.

Optimization problems in finance may consider a single period, where all information are available at time 0 and one takes a one-off decision, or their generalizations. In multistage problems, information become available at multiple points in time, and also, the decisions can be made at multiple points in time (stages). In optimal control problems, one optimizes over policies, which drive the repeated decisions. Throughout, one can work with either discrete decisions (e.g., yes/no, number of round lots) or real-valued decisions (e.g., price). Throughout, one can enforce the constraints given by the Basel III regulatory framework directly, or produce the decision that would be optimal without the constraints, and test whether these constraints are satisfied using simulation tools, as introduced above.

Correspondingly, there is a breadth of approaches, which model active and passive investment management. Within active investment management, one often tries to find the optimal investment strategy striking a balance between the expected profit and some measure of risk involved. Within passive investment management, one can imagine index-tracking funds and their “calibration problems,” which are based on portfolio diversification, and aim at representing a portfolio with a large number of assets by a smaller number of representative assets. Within auction mechanisms, the clearing of the so-called combinatorial auctions, where bids on a subset of items are accepted, is an example of a difficult discrete-valued optimization problem. In this section, we introduce idealized versions of these problems, where only the decision in the next period is considered, transaction costs are ignored, and Basel III constraints are not enforced, and the corresponding quantum algorithms.

As we will discuss shortly, different quantum algorithms have been advocated for different optimization problem classes, e.g., the algorithms for convex problems are different from the ones for discrete problems. In discrete problems, e.g., we will discuss variational approaches, such as variational quantum eigensolver (VQE) and quantum approximate optimization algorithm (QAOA). While, for other applications, quantum computing offers more clear-cut benefits with respect to classical computing (e.g., Shor’s algorithm), in the application of optimization algorithms, these benefits are still an active research area; see, for instance, [11], [12], [16], and [66]–[74]. What it is fair to say is that with the advance of technology, quantum computing will play a major role in optimization, and in some problem classes, one will see tangible benefits, either in terms of solution quality, computational time, or both.

A. PROBLEM CLASSES: CONVEX PROBLEMS

First, we consider convex optimization, which encompasses linear programming (LP), quadratic programming (QP), and semidefinite programming (SDP). Convex optimization [80] is a subclass of continuous optimization problems, where the decision variables are continuous, and it has been advocated for a large variety of applications, among which finance problems [81].

Not surprisingly, much of the recent interest in quantum algorithms for continuous optimization has focused on approaches to solving convex optimization problems and, in particular, SDP. An SDP can be mathematically modeled as

$$\inf \langle C, X\rangle \text{ s.t. } \mathcal{A}X = b, X \succeq_{\mathcal{K}} 0 \tag{14}$$

where cone $\mathcal{K}$ is the cone of positive-semidefinite symmetric $n \times n$ matrices $\mathcal{S}_+^n$, i.e., $\{X = X^\intercal \in \mathbb{R}^{n\times n} |\ X \text{ is positive semidefinite}\}$, and $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ is a linear operator between $\mathcal{S}_+^n$ and $\mathbb{R}^m$:

$$X \mapsto \begin{pmatrix} \langle A_1, X\rangle \\ \dots \\ \langle A_m, X\rangle \end{pmatrix}.$$

TABLE 2. **Overview of recently proposed quantum algorithms for convex optimization**

Ref.	Quantum run-time	Comment
[75]	$\tilde{O}(\sqrt{mn}s^2\mathsf{P}^{32}/\delta^{18})$	to error $1 \pm \delta$, where P is linear in n in general
[76]	$\tilde{O}(s^2\text{poly}(\mathsf{PD}/\epsilon)(\sqrt{m}/\epsilon^{10} + \sqrt{n}/\epsilon^{12})$	
[77]	$\tilde{O}(\sqrt{mn}s^2(\mathsf{PD}/\epsilon)^8)$	where P is linear in n in general
[78]	$\tilde{O}((\sqrt{mn}s(\mathsf{PD}/\epsilon)^4)$	where P is linear in n in general, and quantum-read/classical-write RAM can be accessed in constant time
[79]	$\tilde{O}(n^{2.5}\mu\kappa^3 \log 1/\epsilon)$	where κ goes to infinity for all instances

See Section IV-A for the notation, but note that $\tilde{O}$ hides polylogarithmic terms in the upper bound.

This is a proper generalization of LP, second-order cone programming, and convex cases of quadratically constrained QP and hence of considerable practical interest.

Within classical algorithms, there exist polynomial-time algorithms that can solve SDP. In particular, there are classical upper bounds on the runtime as

$$\mathcal{O}(m(m^2 + n\omega + mnz)\text{polylog}(m, n, \mathsf{P}, 1/\epsilon))$$

where, as said, $\mathcal{O}(\cdot)$ indicates the big-$\mathcal{O}$ notation, n is the dimension of the problem, $\omega \in [2, 2.373)$ is the exponent for matrix multiplication, m is the number of constraints, and nz is the maximal number of nonzero entries per row of the input matrices. P is an upper bound on the trace of an optimal primal solution of an SDP, which could be seen as bound on a diameter of a ball outscribed to feasible solutions, in a suitable norm. This bound shows that SDPs can be approximated to any ϵ in polynomial time classically.

As for quantum algorithms, the early papers [75], [77], and [78] quantized the so-called multiplicative-weight-update (MWU) algorithm of Arora and Kale and variants by Hazan. As has been shown in [77, Appendix E], in the MWU algorithm, $\frac{\mathsf{PD}}{\epsilon}$ should be seen as an important parameter (D being the dual counterpart of P), as one can trade off dependence on one of the three individual parameters for the dependence on the others.

Subsequently, the work in [79] attempted a translation of primal-dual interior-point methods to quantum computers but, due to their reliance on solving linear systems, ended up with a bound dependent on the condition number κ of the Karush–Kuhn–Tucker system, which goes to infinity for all instances, by the design of the method, which may be not ideal in practice.

Finally, in [82], the authors study the relationship of several oracles useful in subgradient algorithms, but do not claim a runtime of a particular algorithm for SDPs.

The key results are summarized in Table 2. As can be observed, some of the quantum algorithms listed report scaling with $O(\sqrt{mn})$ [75] or even $O(\sqrt{m}\,\text{poly}(\log(m), \log n))$ [76]. However, these upper bounds hide the diameters of balls outscribed to the primal and dual solutions. That is, these upper bounds assume that parameters P and D are constants independent of dimension, which could, however, be hard to satisfy in practice.

In fact, if one assumes that P and D are dependent on the dimension of the problem, then lower bounds on the runtime of quantum algorithms can be derived, e.g., for continuous Markowitz portfolio optimization problems in Section IV-B.

Before moving on to an actual finance application, it is useful to briefly outline the quantum part of a quantum SDP algorithm. We focus here on the one of [75], since it seems to be the one that has spurred much of the following research. As said, the authors in [75] quantize the classical MWU algorithm for solving SDPs. In particular, they replace two steps of the classical algorithm by quantum subroutines that are more efficient than classical ones. Consequently, the quantum SDP algorithm is not purely quantum, which is also the case for VQE in combinatorial optimization. The steps that are replaced are as follows. First, it turns out that one can use a "Gibbs sampler" to prepare the new primal candidate as a $\log(n)$-qubit quantum state in much less time than needed to compute it as an $n \times n$ matrix. Second, one can efficiently implement an oracle that is needed in the algorithm based on a number of copies of the quantum state, and using those copies to estimate trace operations. The resulting oracle is weaker than what is used classically, in the sense that it outputs a sample rather than the whole vector (as typical in quantum computing). However, such sampling still suffices to make the algorithm work. The interested reader is referred to [73] and [77] for more technical details.

B. MODERN PORTFOLIO MANAGEMENT–ACTIVE INVESTMENT MANAGEMENT: CONTINUOUS CASE

Let us now consider modern portfolio management and develop lower bounds on the runtime of any quantum algorithm in the quantum query model of Beals *et al.* [83], where quantum computation with T queries is a sequence interleaving T unitary and T query (oracle) transformations, with a measurement at the end. These lower bounds on the runtime are generally based on lower bounds on parameters P, $\mathsf{D} = \Theta(\min\{n, m\}^2)$, as discussed in Section IV-A, and suggest that the quantum speedup of MWU algorithms [75], [77], [78] may be limited in practice.

In modern portfolio theory, one often assumes that there are n possible assets and a number m of forecasts of their returns $b_i \in \mathbb{R}^n$, $1 \leq i \leq m$, based on some historical returns $c \in \mathbb{R}^n$, with a known covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ of the returns. Minimization of the risk subject to lower bounds μ_i on the forecast returns leads to

$$\min_{w\in\mathbb{R}^n_+} \; w^{\mathsf{T}}\Sigma w \quad \text{s.t. } b_i^{\mathsf{T}} w \geq \mu_i \quad \forall\, 1 \leq i \leq m \tag{15}$$

possibly with normalization such as $\sum_{j=1}^{n} w_i = 1$ or similar. In the spirit of Markowitz, one may consider a linear combination of the returns and the risk

$$\max_{w \in \mathbb{R}^n_+} c^\mathsf{T} w - q w^\mathsf{T} \Sigma w \quad \text{s.t. } b_i^\mathsf{T} w \geq \mu_i \quad \forall\, 1 \leq i \leq m. \tag{16}$$

One can also formulate a Lagrangian of the Markowitz model above and maximize $c^\mathsf{T} w - q w^\mathsf{T} \Sigma w$, where the higher $q \geq 0$, the more risk-averse the portfolio will be.

In any case, due to the general result that there exist problem instances for which complexity of every quantum LP solver (and hence also SDP solver) is the same as classical [77], there are instances of Markowitz portfolio management (16) with n assets and m forecasts of the returns, for which a quantum algorithm has the same complexity of a classical one. This has to be taken as an understanding that quantum algorithms may help in some instances of Markowitz portfolio management but not in others, depending on the actual input data.

C. PROBLEM CLASSES: COMBINATORIAL PROBLEMS

We move now to overview combinatorial problems and quantum algorithms that have been advocated for them. Combinatorial optimization problems are the ones for which the decision variables can also be discrete. Combinatorial problems are, in general, nonconvex and not solvable with polynomial-time algorithms, classically. In the quantum domain, variational algorithms for general mixed-binary constrained optimization problems have been studied, and we will overview them as well as apply them for financial problems [84]. First, we will look at quadratic unconstrained binary optimization (QUBO), where VQE/QAOA heuristic approaches have been advocated on noisy quantum devices [12], [24]. Then, we will explore how the classical alternating direction method of multipliers (ADMM) can help solving certain classes of mixed-binary constrained optimization problems [11]. We note that, currently, there is no theoretical guarantee that variational algorithms on quantum devices can achieve significant speedups for QUBOs, and the algorithms reported in this section are used as heuristics. On the other hand, variational algorithms do have nontrivial provable guarantees, and they are not efficiently simulatable by classical computers. They are thus appealing algorithms to explore on near-term quantum machines [70].

D. VARIATIONAL APPROACHES FOR QUBO

The attempts to solve mathematical optimization problems on early generation of universal quantum computers have mainly focused on variational approaches [85]–[87]. In broad terms, a variational approach works by choosing a parameterization of the space of quantum states that depends on a relatively small set of parameters, and then by using classical optimization routines to determine values of the parameters corresponding to a quantum state that maximizes or minimizes a given utility function. Typically, the utility function is given by a Hamiltonian encoding the total energy of the system, to be minimized. The variational theorem then ensures that the expectation value of the Hamiltonian is greater than or equal to the minimum eigenvalue of the Hamiltonian. A large problem class tackled by such variational approaches is that of QUBO problems

Algorithm 1: Outline of VQE.

```
Require: Hamiltonian $H$. Set $\theta = \theta_0$
1: while Error tolerance is unmet: do
2:   Quantum part:
       • Form variational state $|\psi(\theta)\rangle = U(\theta)\,|0\rangle$
       • Compute information about $\lambda = \langle\psi(\theta)|H|\psi(\theta)\rangle$
3:   Classical part:
       • Update $\theta$ via a classical optimization algorithm
         (e.g., COBYLA, SPSA, etc.)
       • Compute the error metric
4: end while
5: return $\lambda, \theta$
```

$$\begin{aligned} \min_{x} \quad & c^\mathsf{T} x + x^\mathsf{T} Q x \\ \text{s.t.:} \quad & x \in \{0,1\}^n, \qquad \text{with } c \in \mathbb{R}^n, Q \in \mathbb{R}^{n \times n}. \end{aligned}$$

Each QUBO can be transformed into an Ising model with Hamiltonian constituted as a summation of weighted tensor products of Z Pauli operators, i.e., $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, by mapping the binary variables x to spin variables $y \in \{-1, 1\}$, i.e., $x = \frac{y+1}{2}$. In case equality constraints $Ax = b$ are present in the mathematical programming formulation, a QUBO can still be devised by adding a quadratic penalization $\alpha\|Ax - b\|^2$ to the objective function, as a soft constraint in an augmented Lagrangian fashion [12], [88], [89].

A typical variational approach on quantum devices, such as VQE [23], would involve the following two key steps in solving a QUBO, given its Ising Hamiltonian $H \in \mathbb{C}^{n \times n}$. First, one would parameterize the quantum state via a small set of rotation parameters θ: each state can then be expressed as $|\psi(\theta)\rangle = U(\theta)|0\rangle$, where $U(\theta)$ is the parameterized quantum circuit applied to the initial state $|0\rangle$. The variational approach would then aim at solving $\min_\theta \langle\psi(\theta)|H|\psi(\theta)\rangle$. Such optimization can be performed in a setting that uses a classical computer running an iterative algorithm to select θ and a quantum computer to compute information about $\langle\psi(\theta)|H|\psi(\theta)\rangle$ for given θ (e.g., its gradient). The algorithm outline of VQE is reported in Algorithm 1.

Another variational approach on quantum devices is QAOA [70], which can be seen as a generalization of VQE. First, one would define rotation parameters $\theta = [\theta_1, \ldots, \theta_d]$ and $\beta = [\beta_1, \ldots, \beta_d]$ together with the Ising Hamiltonian $H \in \mathbb{C}^{n \times n}$, and a mixing Hamiltonian $H_X \in \mathbb{C}^{n \times n}$ defined

```
Algorithm 2: Outline of QAOA.
  Require: Hamiltonian $H$, mixer Hamiltonian $H_X$. Set
    $\theta = \theta_0$, $\beta = \beta_0$
  while Error tolerance is unmet: do
      Quantum part:
        • Form variational state $|\psi(\theta, \beta)\rangle = U(\theta, \beta)|0\rangle$
        • Compute information about
            $\lambda = \langle\psi(\theta, \beta)|H|\psi(\theta, \beta)\rangle$
      Classical part:
        • Update $\theta, \beta$ via a classical optimization algorithm
            (e.g., COBYLA, SPSA, etc.)
        • Compute the error metric
  end while
  return $\lambda, \theta, \beta$
```

as a summations of X Pauli operators. Then, one would construct the quantum state as

$$|\psi(\theta, \beta)\rangle = \exp(-i\beta_d H_X)\exp(-i\theta_d H)\cdots \exp(-i\beta_1 H_X)\exp(-i\theta_1 H)|0\rangle = U(\theta, \beta)|0\rangle . \tag{17}$$

The variational approach would then aim at solving $\min_{\theta,\beta}\langle\psi(\theta, \beta)|H|\psi(\theta, \beta)\rangle$. Such optimization can be performed in a setting that uses a classical computer running an iterative algorithm to select θ, β, and a quantum computer to compute information about $\langle\psi(\theta, \beta)|H|\psi(\theta, \beta)\rangle$ for given θ and β (e.g., its gradient). The algorithm outline of QAOA is reported in Algorithm 2.

E. COMBINATORIAL APPLICATION 1: ACTIVE INVESTMENT MANAGEMENT AND PORTFOLIO OPTIMIZATION

To illustrate the VQE and the QAOA in the context of portfolio optimization, we solve a combinatorial optimization problem, in which we seek to allocate capital to a subset of $B = 3$ assets selected from a larger investment universe with size $n = 6$ [87]. In particular, we will solve the combinatorial problem

$$\min_{x\in\{0,1\}^n} qx^\mathsf{T}\Sigma x - \mu^\mathsf{T} x, \qquad \text{subject to: } 1^\mathsf{T} x = B \tag{18}$$

where we use the following notation.

1) $x \in \{0, 1\}^n$ denotes the vector of binary decision variables, which indicate which assets to pick ($x_i = 1$) and which not to pick ($x_i = 0$).
2) $\mu \in \mathbb{R}^n$ defines the expected returns for the assets.
3) $\Sigma \in \mathbb{R}^{n\times n}$ specifies the covariances between the assets.
4) $q > 0$ controls the risk appetite of the decision maker.
5) and B denotes the budget, i.e. the number of assets to be selected out of n.

We also assume the following simplifications: 1) all assets have the same price (normalized to 1); and 2) the full budget B has to be spent, i.e., one has to select exactly B assets. With these assumptions, the portfolio optimization problem corresponds to building a portfolio by selecting a subset of B assets from the available n assets and equally allocating capital to the B assets. To map the problem (18) to a QUBO, the equality constraint $1^\mathsf{T} x = B$ is mapped to a penalty term $(1^\mathsf{T} x - B)^2$, which is scaled by a parameter and subtracted from the objective function. The resulting problem can be mapped to a Hamiltonian whose ground state corresponds to the optimal solution.

TABLE 3. Comparison of the VQE solution, obtained with an R_y variational form of depth 3, a depth $p = 4$ QAOA solution, and a diagonalization of the hamiltonian of the portfolio optimization problem

Assets selected	Energy	Probability
Diagonalization		
[1 0 1 0 1 0]	-0.0036	1.0000
VQE		
[0 1 0 0 1 1]	-0.0021	0.4431
[0 1 0 1 1 0]	-0.0021	0.3660
[1 1 0 0 1 0]	-0.0036	0.1154
QAOA		
[1 1 0 0 1 0]	-0.0036	0.0487
[1 0 1 0 1 0]	-0.0036	0.0486
[1 0 0 1 1 0]	-0.0031	0.0485

Assets selected shows the candidate solution where a 1 in position i indicates that asset i was selected. The energy is the energy of the selected state, and the probability shows the likelihood of sampling the selected assets from the quantum state created by the quantum algorithm.

We use the state vector simulator in Qiskit Aer [54] as back-end and compare the results to a diagonalization of the Ising Hamiltonian, which encodes the portfolio optimization problem. The VQE and the QAOA both produce variational states, which, when sampled from, result with high probability in an asset selection that respects the budget constraint as shown by the fact that the three most probable states all select three assets (see Table 3). Furthermore, the selected states are either optimal or near optimal as seen by comparing them with the state resulting from the diagonalization in Table 3. The probability to sample near-optimal states with QAOA are lower than for VQE (see Table 3), which may indicate that deeper QAOA variational forms are needed [90] or that the COBYLA optimizer we used was trapped in a local minimum [68], [91]. For such a small problem size, the diagonalization runs in less time than simulations of the VQE and the QAOA. Performing VQE and QAOA on quantum hardware would require even more time. However, such a classical brute force approach scales exponentially in the number of assets, and even for a few tens of assets ($n \sim 30$–40), we expect it not to be a practically viable approach.

In a second example, we optimize the portfolio for different values of the risk–return tradeoff parameter q without the budget constraint. We compare solutions obtained with VQE and solutions obtained from a classical exhaustive search. The most probable asset selections obtained from the state of the VQE closely follow the efficient frontier, therefore maximizing return and minimizing risk (see Fig. 8).

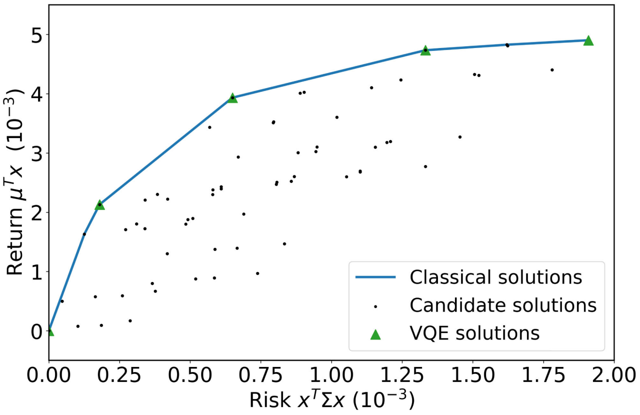


FIGURE 8. **Reconstruction of the efficient frontier using an exhaustive search with exponential scaling (blue line) and the VQE (green triangles). Both methods find the optimal portfolios, i.e., which maximize return and minimize risk, out of the 64 possible asset combinations (black dots). Both quantum and classical algorithms are run several times with different risk–return tradeoff values q to find the efficient frontier.**

F. COMBINATORIAL APPLICATION 2: PASSIVE INVESTMENT MANAGEMENT AND PORTFOLIO DIVERSIFICATION

In passive investment management, one of the main challenges is to build a diverse portfolio with a relatively small number of assets that track the dynamics of a portfolio with a much larger number of assets. This *portfolio diversification* makes it possible to mimic the performance of an index (or a similarly large set of assets) with a limited budget, at limited transaction costs. The purchase of all assets in the index may be impractical for a number of reasons: the total of even a single round lot per asset may amount to more than the assets under management, the large scale of the index-tracking problem with integrality constraints may render the optimization problem difficult, and the transaction costs of the frequent rebalancing to adjust the positions to the weights in the index may render the approach expensive. Thus, a popular approach is to select a portfolio of q assets that represent the market with n assets, where q is significantly smaller than n, but where the portfolio replicates the behavior of the underlying market. To determine how to group assets into q clusters and how to determine which q assets should represent, the q clusters amounts to solving a large-scale optimization problem.

As discussed in [81], we describe a mathematical model that clusters assets into groups of similar ones and selects one representative asset from each group to be included in the index fund portfolio. The model is based on the following data, which we will discuss in more detail later:

$$\rho_{ij} = \text{similarity between stock } i \text{ and stock } j.$$

For example, $\rho_{ii} = 1$ and $\rho_{ij} \leq 1$ for $i \neq j$ and ρ_{ij} is larger for more similar stocks. An example of this is the correlation between the returns of stocks i and j. It allows for similarity measures between time series beyond the covariance matrix. Consider, for instance, a company listed both in London and New York. Although both listings should be very similar, only parts of the time series of the prices of the two listings will overlap, because of the partial overlap of the times the markets open. Instead of covariance, one can consider, for example, dynamic time warping of [92] as a measure of similarity between two time series, which allows for the fact that for some time periods, the data are captured by only one of the time series, while for others, both time series exhibit the similarity due to the parallel evolution of the stock price.

The problem that we are interested in solving is

$$(M) \quad f = \max_{x_{ij}, y_j} \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_{ij}$$

subject to the clustering constraint

$$\sum_{j=1}^{n} y_j = q$$

to consistency constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i = 1, \dots, n$$
$$x_{ij} \leq y_j \qquad \forall\, i = 1, \dots, n;\ j = 1, \dots, n$$
$$x_{jj} = y_j \qquad \forall\, j = 1, \dots, n$$

and integral constraints

$$x_{ij}, y_j \in \{0, 1\} \qquad \forall\, i = 1, \dots, n;\ j = 1, \dots, n.$$

The variables y_j describe which stocks j are in the index fund ($y_j = 1$ if j is selected in the fund, 0 otherwise). For each stock $i = 1, \dots, n$, the variable x_{ij} indicates which stock j in the index fund is most similar to i ($x_{ij} = 1$ if j is the most similar stock in the index fund, 0 otherwise).

The first constraint selects q stocks in the fund. The second constraint imposes that each stock i has exactly one representative stock j in the fund. The third and fourth constraints guarantee that stock i can be represented by stock j only if j is in the fund. The objective of the model maximizes the similarity between the n stocks and their representatives in the fund. Different cost functions can also be considered.

From (M), one can construct a binary polynomial optimization with equality constraints only, by substituting the $x_{ij} \leq y_j$ inequality constraints with the equivalent equality constraints $x_{ij}(1 - y_j) = 0$. Then, the problem becomes

$$\max_{x_{ij}, y_j} \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_{ij} \tag{19a}$$

$$\text{s. t.:} \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i = 1, \dots, n \tag{19b}$$

$$x_{ij}(1 - y_j) = 0 \qquad \forall\, i = 1, \dots, n \quad \forall\, j = 1, \dots, n \tag{19c}$$

$$x_{jj} = y_j \qquad \forall\, j = 1, \dots, n. \tag{19d}$$

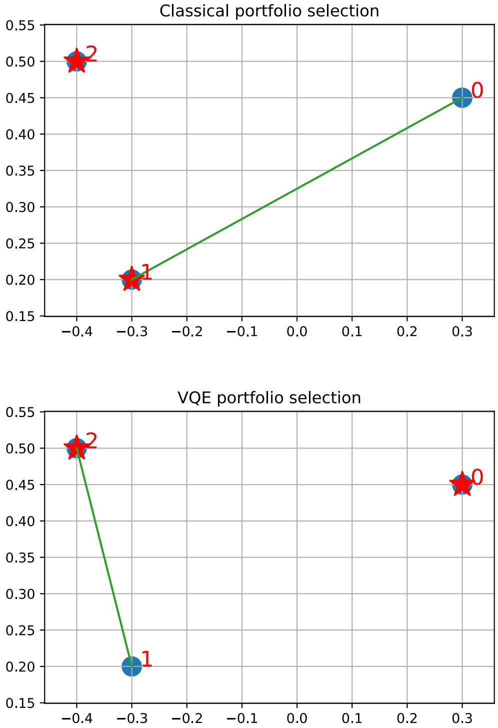


FIGURE 9. **Portfolio selection for classical and quantum algorithm. The stocks are represented in an arbitrary 2-D plane for visualization purposes. The selected stocks are indicated with red stars, while the green link indicates the association of the unselected stock to a selected one. Classical and quantum solution are different but fare very similarly in terms of benefit.**

We can now construct the Ising Hamiltonian (QUBO) by penalty methods (introducing a penalty coefficient A for each equality constraint) as

$$H = \sum_{i=1}^{n}\sum_{j=1}^{n} \rho_{ij}x_{ij} + A\left(\sum_{j=1}^{n} y_j - q\right)^2 + \sum_{i=1}^{n} A\left(\sum_{j=1}^{n} x_{ij} - 1\right)^2 + \sum_{j=1}^{n} A(x_{jj} - y_j)^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} A\left(x_{ij}(1-y_j)\right). \quad (20)$$

For the simulation in Qiskit, we use three assets ($n = 3$) and two clusters ($q = 2$); this leads to a 12-qubit Hamiltonian. We solve the problem classical with CPLEX and on the quantum computer with VQE (with depth 7 and full entanglement).

In Fig. 9, we report the results that we obtain conveniently (and arbitrarily) displayed in a 2-D graph for visualization purposes. Solution shows the selected stocks via the stars and in green the links (via similarities) with other stocks that are represented in the fund by the linked stock. As we see, both for classical and quantum, we can find a feasible solution for our diversification (clustering and selecting two stocks, while associating the third to a selected one), although the classical algorithm here finds a slightly better solution (the classical benefit fares at 2.001, while the quantum one at 2.000). This is reasonable for such small problem instance, since the classical solver can be run to find the exact solution, while VQE is a heuristic and may find less optimal solutions, but this might not be the case when the classical solver will not be able to run at optimality for larger size problems.

Algorithm 3: 3-ADMM-H Mixed-Binary Heuristic.

```
Require: Initial choice of $x_0, \bar{x}_0, y_0, \lambda_0$. Choice of $\varrho, \beta, c > 0$, tolerance $\epsilon > 0$, and maximum number of iterations $K_{\max}$.
while $k < K_{\max}$ and $\|A_0 x_k - A_1 \bar{x}_k - y_k\| < \epsilon$, do
    First block update (QUBO) on the quantum device:
        $x_k = \arg\min_{x\in\{0,1\}^n} q(x) + \frac{c}{2}\|Gx - b\|_2^2 + \lambda_{k-1}^{\mathsf{T}} A_0 x + \frac{\varrho}{2}\|A_0 x + A_1 \bar{x}_{k-1} - y_{k-1}\|^2$   (21)
    Second block update (Convex) on the classical device:
        $\bar{x}_k = \arg\min_{\bar{x}\in\mathbb{R}^m} f_1(\bar{x}) + \lambda_{k-1}^{\mathsf{T}} A_1 \bar{x} + \frac{\varrho}{2}\|A_0 x_k + A_1 \bar{x} - y_{k-1}\|^2$   (22)
    Third block update (Convex+quadratic) on the classical device:
        $y_k = \arg\min_{y\in\mathbb{R}^n} \frac{\beta}{2}\|y\|_2^2 - \lambda_{k-1}^{\mathsf{T}} y + \frac{\varrho}{2}\|A_0 x_k + A_1 \bar{x}_k - y\|^2$
    Dual variable update on the classical device:
        $\lambda_k = \lambda_{k-1} + \varrho(A_0 x_k + A_1 \bar{x}_k - y_k)$
    Compute merit value:
        $\eta_k = q(x_k) + \phi(\bar{x}_k) + \mu(\max(g(x_k), 0) + \max(l(x_k, \bar{x}_k), 0))$   (23)
end while
return $x_{k^*}, \bar{x}_{k^*}, y_{k^*}$, with $k^* = \min_k \eta_k$.
```

G. MULTIBLOCK ADMM HEURISTIC FOR MIXED-BINARY OPTIMIZATION

We move on to mixed-binary optimization (MBO) formulations. In a general MBO problem, the decision maker faces binary and continuous decisions, subject to equality and inequality constraints. MBO formulations enable to tackle finance problems, such as the combinatorial auction problem, which is the scope of Section IV-H.

In order to introduce solvers for MBO, we consider the following reference MBO problem (P):

$$\min_{x\in\mathcal{X}, u\in\mathcal{U}\subseteq\mathbb{R}^l} \quad q(x) + \varphi(u) \quad (24a)$$

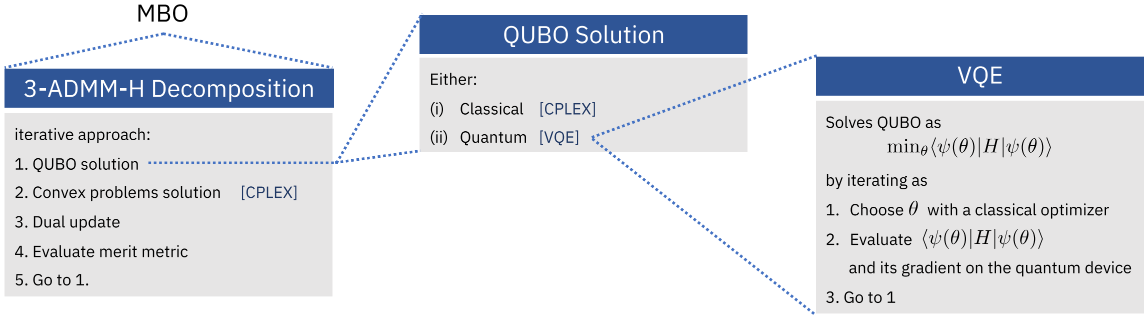

MBO
3-ADMM-H Decomposition
iterative approach:
1. QUBO solution
2. Convex problems solution [CPLEX]
3. Dual update
4. Evaluate merit metric
5. Go to 1.
QUBO Solution
Either:
(i) Classical [CPLEX]
(ii) Quantum [VQE]
VQE
Solves QUBO as
$\min_\theta \langle\psi(\theta)|H|\psi(\theta)\rangle$
by iterating as
1. Choose θ with a classical optimizer
2. Evaluate $\langle\psi(\theta)|H|\psi(\theta)\rangle$
and its gradient on the quantum device
3. Go to 1

FIGURE 10. **Illustration diagram of the 3-ADMM-Happroach. There are two nested loops for the selected implementation, specifically the outer ADMM loop and the inner VQE loop (Note that VQE can be substituted via QAOA seamlessly).**

$$\text{s.t. :} \quad Gx = b, \quad g(x) \leq 0 \tag{24b}$$

$$\ell(x, u) \leq 0 \tag{24c}$$

with the corresponding functional assumptions.

Assumption 1: The following assumptions hold.

1) Function $q : \mathbb{R}^n \to \mathbb{R}$ is quadratic, i.e., $q(x) = x^\intercal Qx + a^\intercal x$ for a given symmetric squared matrix $Q \in \mathbb{R}^n \times \mathbb{R}^n$, $Q = Q^\intercal$, and vector $a \in \mathbb{R}^n$.
2) The set $\mathcal{X} = \{0, 1\}^n = \{x_{(i)}(1 - x_{(i)}) = 0, \forall i\}$ enforces the binary constraints.
3) Matrix $G \in \mathbb{R}^n \times \mathbb{R}^{n'}$, vector $b \in \mathbb{R}^{n'}$, and function $g : \mathbb{R}^n \to \mathbb{R}$ is convex.
4) Function $\varphi : \mathbb{R}^l \to \mathbb{R}$ is convex and $\mathcal{U}$ is a convex set.
5) Function $\ell : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ is *jointly* convex in x and u.

In order to solve MBO problems, Gambella and Simonetto [11] proposed heuristics for (P) based on the ADMM [93]. ADMM is an operator-splitting algorithm with a long history in convex optimization, and it is known to have residual, objective, and dual variable convergence properties, provided that convexity assumptions hold [93].

The method of [11] (referred to as 3-ADMM-H, and displayed in Fig. 10) leverages the ADMM operator-splitting procedure to devise a decomposition for certain classes of MBOs into:

1) a QUBO subproblem to be solved by on the quantum device via variational algorithms, such as VQE or QAOA, described in Section IV-D;
2) a continuous convex constrained subproblem, which can be efficiently solved with classical optimization solvers [80].

Algorithm 3 reports the 3-ADMM-H algorithm, along with stopping criteria and evaluation metrics. A comprehensive discussion on the conditions for convergence, feasibility, and optimality of 3-ADMM-H is out of the scope of this article and can be found in [11]. Combinatorial auction (A) belongs to the class of MBOs represented by (P) and can be solved by 3-ADMM-H. Simulations on representative instances are conducted in Section IV-H.

H. COMBINATORIAL APPLICATION 3: AUCTIONS

Both governments and private issuers finance its activities, in part, by the sale of marketable securities. The issuer often uses an auction process to sell such marketable securities and determine their parameters (such as yield). For example, the United States treasury issued over \$10T (ten trillion U.S. dollars) in securities in 2018. Many further auction mechanisms abound in electrical energy markets, pollution management, and within airport operations (airport landing slots). Some of these auctions may be combinatorial, in the sense that the value that a bidder has for a set of items may not be the sum of the values that he has for individual items. It may be more or it may be less.

Combinatorial auctions allow the bidders to submit bids on subsets (combinations) of items. Specifically, let $M = \{1, 2, \ldots, m\}$ be the set of items that the auctioneer has to sell. A bid is a pair $B_j = (S_j, p_j)$ where $S_j \subseteq M$ is a nonempty set of items and p_j is the price offer for this set. Suppose that the auctioneer has received n bids $B_1, B_2, \ldots, B_n$. How should the auctioneer determine the winners in order to maximize his revenue? This can be formulated as an integer program. To render the problem a bit more interesting, we consider the case in which (as in some auctions) there are multiple indistinguishable units of each item for sale. A bid in this setting is defined as $B_j = (\lambda_1^j, \lambda_2^j, \ldots, \lambda_m^j; p_j)$, where λ_i^j is the desired number of units of item i and p_j is the price offer. Let x_j be a binary variable that takes the value 1 if bid B_j wins, and 0 if it loses. The auctioneer maximizes his revenue by solving the integer program (A)

$$\max_{x_j} \quad \sum_{j=1}^{n} p_j x_j \tag{25}$$

$$\text{s.t.:} \sum_{j:i \in S_j} \lambda_i^j x_j \leq u_i, \quad \text{for } i = 1, \ldots, m \tag{26}$$

$$x_j \in \{0, 1\}, \quad \text{for } j = 1, \ldots, n \tag{27}$$

where u_i is the number of units of item i for sale [81].

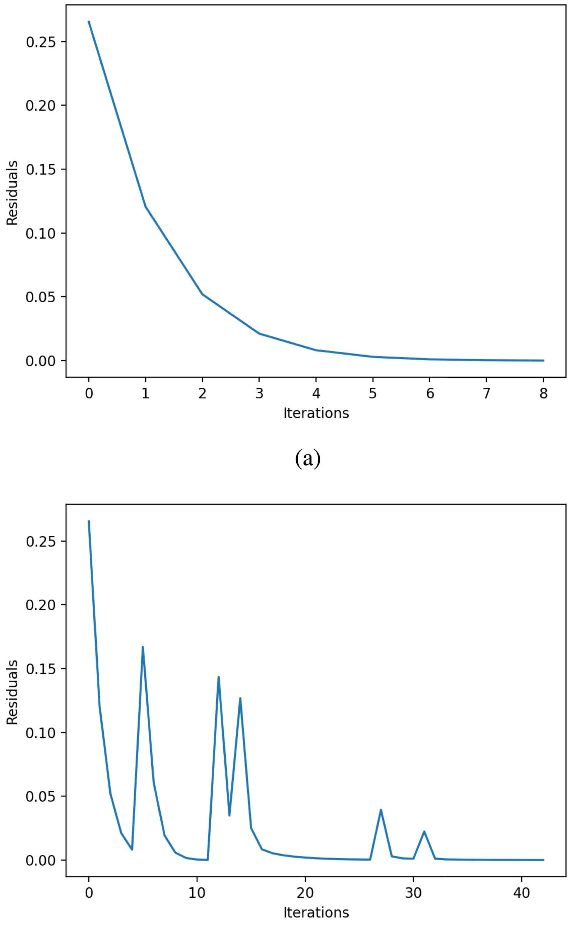
The presence of inequality constraints (26) makes a reformulation of (A) into a QUBO not possible; hence, the

VQE algorithm described in Section IV-D is not directly applicable. We here report results obtained by solving (A) with the 3-ADMM-H heuristic described in Section IV-G.

For simulation purposes, an instance with $m = 3$ items and $n = 16$ bids with randomly generated profits has been created. For each item, six units are available. The number of units of the items in each bid has been randomly sampled from the interval $[1, 6]$. This means that not all bids are necessarily feasible if more than one object is in the bid. Because the decision of accepting a bid j is a binary decision x_j, the number of bids is the number of qubits the algorithm 3-ADMM-H necessitates. The optimal solution, found by solving Problem (A) via the classical optimization solver IBM ILOG CPLEX, consists of accepting bids $B_0 = \{0\}, B_1 = \{1\}, B_4 = \{1, 2\}$ with a profit of 24. The 3-ADMM-H algorithm has been tested on the instance by choosing VQE as quantum solver in Qiskit Aqua and Constrained Optimization By Linear Approximation (COBYLA) [94] as classical VQE optimizer with 20 maximum iterations. The `qasm_simulator` has been used as Qiskit Aer backend for the simulations on quantum devices. The ADMM parameters ρ and β have been set to 12 and 11, respectively: this is to leverage the convergence properties described in [11] for $\rho > \beta$. When run on classical devices, the first block update is performed with the CPLEX solver.

The 3-ADMM-H solution with CPLEX as QUBO solver is $B_1 = \{1\}, B_3 = \{0, 2\}, B_4 = \{1, 2\}$, with a profit of 27, and a violation of constraints (26) of 2. Setting VQE as QUBO solver makes 3-ADMM-H converge to the same solution in 43 iterations. The residuals $r_k = A_0 x_k - A_1 \bar{x}_k - y_k$ are reported for the classical and quantum simulations, in Fig. 11(a) and (b). Residuals are not guaranteed to decrease in each 3-ADMM-H iteration. The convergence guarantees of 3-ADMM-H are not valid for inexact QUBO solvers, such as the currently available quantum algorithms. However, the convergence curves show that 3-ADMM-H terminates in a finite number of iterations, even when the QUBO solver is inexact. Hence, 3-ADMM-H exhibits a certain degree of tolerance to inexact computations. This corroborates the empirical findings of [11] on packing problems.

The 3-ADMM-H algorithm proposes a decomposition of an MBO problem, in which the most computationally demanding part is solving the QUBO subproblem (21). The advantage of using 3-ADMM-H algorithm over classical optimization solvers, such as CPLEX, lies in leveraging quantum algorithm to tackle QUBO subproblems.



(a)

(b)

FIGURE 11. Convergence of the residuals for 3-ADMM-H algorithm, in the classical (a) and quantum (b) simulations. The simulation with quantum QUBO solver shows that, even with an inexact QUBO solver, 3-ADMM-H can reach convergence in a finite, and relatively low, number of iterations.

V. MACHINE LEARNING

Finally, in this section, we discuss ML problems where quantum algorithms may demonstrate an advantage.

ML focuses on finding relations in data and building assumptions around them for the following: 1) prediction by anticipating future events from historic data; or to 2) classify data by dividing an end result into different categories; or 3) to find patterns by the discovery of regularities or anomalies in data. In finance, such approaches are important in many financial problems that deal with uncertainty in the future evolution of asset prices and risk. For example, the investment management strategies and optimization in the previous section make use of the estimation of future risks and asset prices that can be obtained from the output of ML algorithms. Banks can estimate the risk level of their customers' loans by credit scoring, which can be formulated as classification [95] and/or regression problem based on the rich features of customers, such as age, salary, historical payment, micro- and macroeconomic indicators, and so on. Financial institutions can also detect frauds by finding patterns that deviates greatly from normal behavior by classification and/or anomaly detection [96], [97]. Such ML tasks are known to face *the curse of dimensionality* as there are much more features available to model customers. Principal component analysis and variational autoencoder [98] are some of the popular methods for dimensionality reduction when dealing with high-dimensional features.

To summarize ML problems at each stage of the customer life cycle (see Fig. 3):

1) *Customer Identification (and Scoring):* Refine customer rating and segmentation to improve know your customer (KYC) and avoid noncompliance annual penalties, which have grown to \$8.3 Million annually by 2018 [99].
2) *Financial Products:* Increase credit scoring realism to improve customer targeting and align product offering.
   a) Avoid NPL costs that are still increasing in many European economies [100].
   b) Increase customer interaction with the bank resulting in more profits.
   c) Improve recommendation techniques to target customers prone to accept them.
3) *Monitor Transactions:* Improve suspicious transaction signals to decrease false alerts, that today cost \$4 trillion due to 75–90% false positives in anti money laundering (AML) and credit card fraud alerts.
4) *Customer Retention:* Maximize engagement to avoid customer churn rates to new entrants, 25% of small-medium business (SMB) are turning to FinTech companies for ease and speed of completing loan applications [101].

Overall, AI and ML are of deep interest for financial institutions, with a current global spending in the banking industry worth of \$3.3 billion in 2018 [102] with the hopes to build better classification models that will improve customer service in external facing and internal activities. The International Data Corporation reported that of global spending on AI worth \$50.1 Billion in 2020, which is expected to double in four years, banking is one of the largest industries spending the most in AI/ML solutions for fraud analysis, investigation, program advisors, and recommendation systems [103].

In the following, we discuss how quantum-enhanced feature space can be used in a simple task of binary classification that can be applied to financial applications, such as fraud detection (for transaction monitoring) and credit risk scoring (for customer identification). There are many other tasks addressable by quantum ML techniques (see, e.g., [3] and [104]) for more tasks and applicable quantum techniques. We focus on supervised learning using support vector machine (SVM): We have access to labeled training data S to classify test data T and labels of unseen future data (with assumption that all data come from the same underlying distribution).

Assume that we are given the training data $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{m_S}, y_{m_S})\}$, where each $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. The goal of learning a binary classifier from S is to construct a function $f(x)$ so that $f(x)y_i > 0$. The simplest form of such function is a linear classifier $f(x) = w^\intercal x + b$, where $(w, b) \in \mathbb{R}^{d+1}$. S is called linearly separable if there is a $(w, b) \in \mathbb{R}^{d+1}$ satisfying $f(x)y_i > 0$. Such function, if exists, can be found by solving an optimization problem known as Hard-SVM.

In general, the dataset may not be linearly separable. In such a case, we can still find a classifier that predicts the training dataset with some error margin. The formulation is known as Soft-SVM, as shown below, and can be solved efficiently by techniques such as stochastic gradient descent (SGD)

$$\begin{aligned}(w_0, b_0) = &\ \underset{(w,b)}{\arg\min}\ \|w\|^2 + C\sum_{i=1}^{m_S}\epsilon_i\\ \text{subject to}\ :&\ \ y_j\,(w^\intercal x_j + b) \geq 1 - \epsilon_j \text{ for } j \in [m_S]\\ &\ \ \epsilon_j \geq 0 \text{ for } j \in [m_S].\end{aligned}$$

The slack variables $\{\epsilon_i\}$ determine the quality of the classifier: the closer they are to zero, the better the classifier. For this purpose, we can embed the data $\{x_j\}$ into a larger space by preprocessing the data, namely, by finding a map $x : \Phi(x) \in \mathbb{R}^n$ for $n > d$. The classifier $f(x)$ is now defined as $f(x) = w^\intercal\Phi(x) + b$. When $\Phi(x)$ is an embedding of data nonlinearly to quantum state $|\Phi(x)\rangle$, then we can use quantum-enhanced feature space for the classifier. There are two techniques to construct such a quantum-enhanced feature space that may lead to a quantum advantage: variational quantum classification (VQC) and quantum kernel estimation (QKE).

The VQC is similar to SGD for finding the best hyperplane (w, b) that linearly separates the embedded data. At VQC, the data $x \in \mathbb{R}^d$ are mapped to (pure) quantum state by the feature map circuit $U_{\Phi(x)}$ that realizes $\Phi(x)$. This means that conditioned on the data x, we apply the circuit $U_{\Phi(x)}$ to the n-qubit all-zero state $|0_n\rangle$ to obtain the quantum state $|\Phi(x)\rangle$. A short-depth quantum circuit $W(\theta)$ is then applied to the quantum state, where θ is the hyperparameter set of the quantum circuit that can be learned from the training data. Finding the circuit $W(\theta)$ is akin to finding the separating hyperplane (w, b) in the Hard-SVM and Soft-SVM, with the path to quantum advantage stemming from the fact that there is no efficient classical procedure to realize the feature map $\Phi(x)$. While the size of the hyperparameter set θ is polynomial in the number of qubits and can be tuned with variational methods similar to Algorithms 1 and 2, it controls an exponentially large space of the feature map. The binary decision is obtained by measuring the quantum state in the computational basis to obtain $z \in \{0, 1\}^n$ and linearly combining the measurement results, say with $g = \sum_{z\in\{0,1\}^n} g(z)|z\rangle\langle z|$, where $g(\cdot) \in \{-1, 1\}$.

A quantum circuit that realizes the quantum feature map, as well as the variational classifier is shown in Fig. 12. We can see that the probability of observing z is given as

$$|\langle z|\,W(\theta)\,|\Phi(x)\rangle|^2 = \langle\Phi(x)|W^\dagger(\theta)\,|z\rangle\langle z|\,W(\theta)|\Phi(x)\rangle.$$

By linear combination of the measurement results z with g, we can obtain the function $f(x)$ as follows, which resembles the linear classifier $f(x) = w^\intercal\Phi(x) + b$

$$f(x) = \langle\Phi(x)|W^\dagger(\theta)gW(\theta)|\Phi(x)\rangle + b.$$

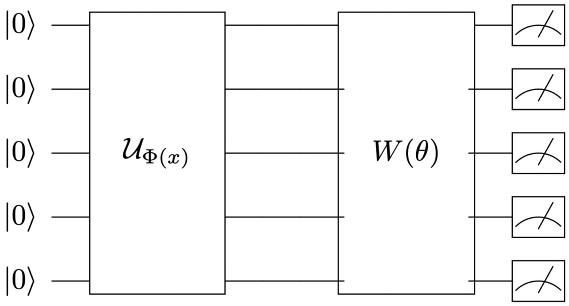


FIGURE 12. Quantum circuit for VQC that consists of fixed quantum feature mapping $\mathcal{U}_{\Phi(x)}$ and the separator $W(\theta)$ trained with variational methods.

The predicted label of $f(x)$ is then simply its sign. The hyperplane (w, b) is now parameterized by θ. The ith element of $w(\theta)$ is $w_i(\theta) = \mathrm{tr}(W^\dagger(\theta) g W(\theta) P_i)$, where P_i is a diagonal matrix whose elements are all zeros except at the ith row and column which is 1, and the ith element of $\Phi(x)$ is $\Phi_i(x) = \langle \Phi(x)|P_i|\Phi(x)\rangle$.

Learning the best θ can be obtained by minimizing the empirical risk $R(\theta)$ with regard to the training data $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{m_S}, y_{m_S})\}$. Namely, the empirical risk (or, cost function) to be minimized is

$$R(\theta) = \frac{1}{|S|} \sum_{i \in [m_S]} |f(x_i) - y_i| \,. \tag{28}$$

The above empirical risk can then be approximated with a continuous function using *sigmoid* function, as detailed in [25]. This enables applying variational methods as in Algorithms 1 and 2 with SGD algorithms (such as COBYLA or SPSA) for tuning θ to minimize the cost function.

The binary classification with VQC now follows from first training the classifier to learn the best θ^*, that minimizes the empirical risk $R(\theta)$, to obtain $(w(\theta^*), b^*)$. This can be done with Algorithm 1 with the Hamiltonian replaced by the empirical risk. The classification against unseen data x is then performed according to the classifier function $f(x)$ with $(w(\theta^*), b^*)$. Both training and classification need to be repeated for multiple times (or shots) due to the probabilistic nature of quantum computation. The former may need significant number of shots proportional to the size of S, but it can be performed in batch offline. On the other hand, the latter needs much less number of shots and may be performed online (or, near real time) as long as the quantum feature map for nonlinear embedding can be computed efficiently.

In the conventional SVM, there are many known methods of nonlinear embedding of data $x : \Phi(x) \in \mathbb{R}^n$ for $n > d$, such as Polynomial-SVMs. For example, in a Polynomial-SVM, the 2-D data (x_1, x_2) can be embedded into a 3-D (z_1, z_2, z_3) such that $z_1 = x_1^2$, $z_2 = \sqrt{2}x_1x_2$, and $z_3 = x_2^2$. On the other hand, in the quantum-enhanced SVM, the embedding of data to n-qubit feature space can be performed by applying the unitary $\mathcal{U}_{\Phi(x)} = U_{\Phi(x)} H^{\otimes n} U_{\Phi(x)} H^{\otimes n}$, where H is the Hadamard gate, and $U_{\Phi(x)}$ denotes a diagonal gate in the Pauli-Z basis as follows:

$$U_{\Phi(x)} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(x) \prod_{k \in S} Z_k \right) \tag{29}$$

where the coefficients $\phi_S(x) \in \mathbb{R}$ are fixed to encode the data x. For example, for $n = d = 2$ qubits, $\phi_i(x) = x_i$ and $\phi_{1,2}(x) = (\pi - x_1)(\pi - x_2)$ were used in [25]. In general, the $U_{\Phi(x)}$ can be any diagonal unitary that can be implemented efficiently with short-depth quantum circuits. In total, one needs at least $n \geq d$ qubits to construct such a quantum-enhanced feature map. There are other proposed methods promising quantum advantage for nonlinear embedding of data into quantum feature space, such as *squeezing* in continuous quantum systems [105] that guarantees linear separability or *amplitude encoding* [106] that can exploit tensorial feature map or *density-operator encoding* [107]. A recent paper studies the embedding in the context of *metric learning* [108].

Classification models in real-world datasets often also depend on binary features, such as gender and yes–no answers to questions, in addition to (discrete) categorical and qualitative features, such as zip code, age, and color. Such discrete features have to be encoded into continuous features before they can be used effectively in ML models that rely on continuity of their inputs, such as VQCs. There have been many proposed encodings, with *one-hot* encoding as one of the most populars, for such purposes [109]. It is known that the encodings can heavily impact the performance of the learning models. Efficient mapping of such discrete features into quantum-enhanced feature space is very important in finance models with structured data. A recent study [110] reports the possibility of using *quantum random access coding* (QRAC) [111] to map discrete features into the quantum-enhanced feature space resulting in faster training and better classification accuracy due to using less number of qubits and hence less hyperparameters in the VQC models. The idea is to split the encoding of x into that for the discrete and continuous parts, each represented as $x^{(b)}$ and $x^{(r)}$. The discrete parts $x^{(b)}$ are obtained from the encoding of categorical features into binary strings using determined techniques such as one-hot encoding or into integer numbers for ordinal features. Fig. 13 depicts a VQC with QRAC for encoding discrete features.

In particular, let us consider the case of classifying credit card transactions into fraudulent or not from a synthesized dataset from [112], which was generated with state machines in simulated world to be representative for the U.S. customers. For our purpose, the synthesized credit card transaction data were prepared to contain 100 records of purchase transactions. The ith transaction x_i contains the transaction time, the transaction amount, the transaction method, the transaction location (in ZIP code), and the Merchant Category Code (MCC). The first two are in real numbers, and the rest are categorical; there are three types of transaction methods, ten different locations, and ten different MCCs.

TABLE 4. Average and standard deviation of accuracy of classifiers on fivefold cross validations of the synthetic credit-card transaction dataset

	LR	SVC1	SVC2	VQC	VQCwQRAC
train	0.79 ± 0.02	0.80 ± 0.03	0.87 ± 0.02	0.74 ± 0.05	0.85 ± 0.03
test	0.77 ± 0.06	0.81 ± 0.07	0.82 ± 0.07	0.64 ± 0.09	0.83 ± 0.06

LR, SVC1, and SVC2, are, respectively, the logistic regression, the SVC with linear, and RBF kernel. VQC and VQCwQRAC are quantum-enhanced SVMs with the latter using QRAC for encoding the transaction method.

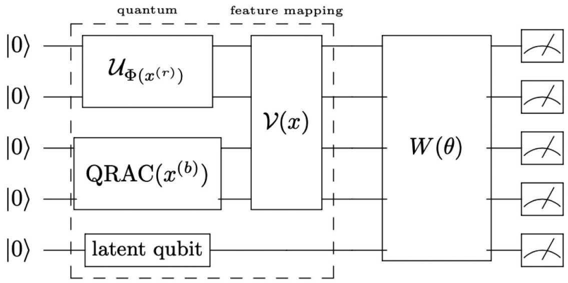


FIGURE 13. Quantum circuits for variational quantum classifier with QRAC for encoding discrete features. Latent qubits may be included to add the dimension of the embedding in the Hilbert space.

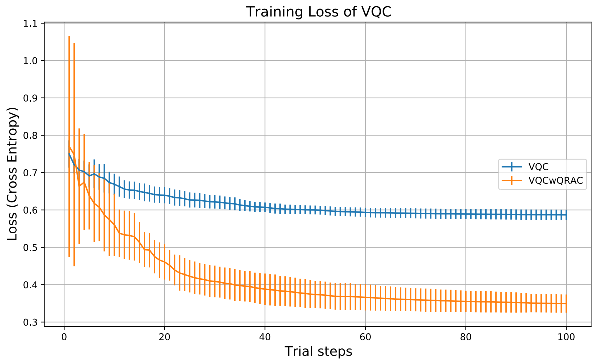


FIGURE 14. Comparison of the training loss of VQC against that with QRAC (VQCwQRAC) for encoding discrete features on a synthetic credit-card transaction dataset.

Each ith transaction is labeled as either fraudulent ($y_i = -1$) or normal ($y_i = 1$). A similar study on the same dataset has also been carried out using *variational quantum Boltzmann machines*, an alternative approach to VQC or QKE, and we refer to [113] for more details.

We applied the VQC as in Fig. 12 by regarding all features as real values to use the feature mapping in (29) with second-order expansion. On the other hand, we applied the VQC as in Fig. 13 by the QRAC of the one-hot encoding of the transaction method, and the rest similar to the VQC. The latter is denoted as VQCwQRAC. Both models used 5 qubits and were trained with variational circuits $W(\theta)$ defining the separating hyperplanes that consist of the RXRY variational gates and one layer of fully connected entangler as implemented in Qiskit [54]. Both classification models were run on qiskit simulators and tested with fivefold cross validation of the dataset. The average training losses, where (28) is approximated with the *cross entropy*, are shown in Fig. 14. We can see that using QRAC for encoding binary features can result in better training losses. The accuracy of VQCwQRAC is better than the VQC using real-valued quantum feature mapping, as shown in Table 4, and is comparable to support vector classifier with radial basis function (RBF) kernel (SVC2 in the table).

Finally, we note that the possible quantum advantage for ML task is somewhat speculative; there is no known theoretical proof that the quantum feature map, which is hard to compute classically, can result in better accuracy than any classical classifiers. Also, the underlying variational methods, as in Algorithms 1 and 2, are heuristics that may only find local optima instead of the global one and thus can lower the accuracy of the resulting quantum-enhanced SVM.

VI. TECHNICAL CHALLENGES IN QUANTUM COMPUTING

In the following section, we outline some of the technical challenges to address when solving computationally challenging problems on a quantum computer.

A. LOADING DATA

To understand constraints on quantum computing both near term and long term, it may be useful to contrast quantum computers against classical computers. Classical computing utilizes the well-known von Neumann model: there is a central processing unit (CPU), which performs nonreversible computation, including branching, and this is connected by a system bus to volatile memory (RAM) and nonvolatile memory (such as a hard drive). Loading data from the nonvolatile memory to RAM and accessing the data in RAM from the CPU is taken for granted. In contrast, there are no quantum (memory) hard drives at the current level of hardware technology; most blueprints do not involve any RAM, and all of the computations are reversible without branching (excepting postselection).

The key difference lies in the time complexity of “loading data.” A quantum state can be seen as a volatile memory of substantial capacity, but with nontrivial issues in addressing it. With k qubits, we work with $2^k \times 2^k$ density matrices,[2] but working with these matrices is limited to a certain set of one- and two-qubit gates (unitary matrices applied to the quantum state). The quantum circuit complexity of state preparation,

[2] A density matrix is a representation of a quantum state. A density matrix can represent both pure states (i.e., states represented by a state vector $|\psi\rangle$) and mixed quantum states (i.e., a statistical ensemble of pure states).

i.e., minimum number of gates required in order to "load" a given arbitrary quantum state U using *any* sequence of one- and two-qubit gates, is greater or equal than 4^k for almost all U. Note that this is not a worst-case result: this holds generically for all states and it applies to the best possible sequence of one- and two-qubit gates. Consider a dimension-counting argument. There are also explicit constructions showing that this is tight.

While this means theoretically that a 4^k-dimensional state can be prepared into a quantum machine having k qubits via a quantum loader having $O(4^k)$ circuit complexity—which is linear in the data dimension—it is also exponential in the number of qubits k. Whereas in classical computing, we usually assume that we can load the data in time linear in the number of bits and then worry about the runtime (circuit complexity of processing) on the loaded data, in quantum computing, preparation of a quantum state may require a quantum circuit complexity exponential in the number of qubits. The complexity of generic state preparation can impede a potential quantum advantage, as the loading time may eclipse coherence times for the physical quantum state and also for some algorithms, loading the data can become computationally as expensive as using a classical algorithm to solve the problem [114].

There are multiple ways of circumventing this issue. One is to allow j-qubit gates, where j may grow with k, which poses a major challenge in quantum optimal control. One is to "split" the state preparation into an independent system, such as circuitry of some potential qRAM [115], and utilize quantum optimal control there, perhaps across all of the k qubits. Given the (quantum circuit complexity) equivalence [116] of state preparation and an arbitrary circuit, it seems unlikely that it would be possible to implement one way of circumventing the quantum circuit complexity without being able to implement the other, and without being able to utilize the same quantum optimal control in the execution of the quantum circuit. Indeed, it is believed that the physical realization of qRAM model may be even much more difficult than the fault-tolerant quantum computers [104], [114].

In some cases, the problem can also be circumvented because the data to be loaded have structure or properties that can be exploited for efficient loading, e.g., if the data can be described by a log-concave probability distribution [51]. Alternatively, and depending on the application, we may drop the goal of loading the data exactly and try to prepare a quantum state that, at least, is close to our original data. This enables approximate data loading schemes, which have some potential to work around this problem [57]. It is also possible to exploit periodic properties in certain datasets, for example, time-series data, which exhibit periodic properties. By extracting periods in the data via classical techniques such as fast Fourier transform, we may then load only the dominant periods via a small number of steps onto the quantum machine and then recover an approximation of the original data in the quantum machine via a QFFT^{-1} algorithm.

B. QUANTUM ERROR CORRECTION

As has been suggested in Section I, a key watershed between noisy quantum computers and universal fault-tolerant quantum computers is the availability of QEC.

The key technical challenge within QEC is the tradeoff between the overhead of the QEC and the so-called error threshold. The overhead is, essentially, the number of physical qubits required to protect a certain number of logical qubits against errors. The error threshold comes from the famous (quantum) threshold theorem [17], [18], which shows that if the errors on individual qubits are not correlated and the error of the physical qubits falls below a certain threshold, QEC schemes can correct the remainder of the error, at a cost of the overhead. Actually, the dissertation of Gottesman [17] shows that there is a simple construction, starting with classical error correcting codes, which makes it possible to estimate the threshold. For one of the best-known classes of QEC, it is sometimes assumed [19] that a 0.1% probability of a depolarizing error would require more than 1000 physical qubits per protected qubit—although the details of the calculation are also often disputed. There is a substantial interest in further classes of QEC (e.g., hyperbolic surface codes), which could perform substantially better.

For the same QEC mentioned above [19], one should notice that there need not be a substantial increase in the depth of the circuit: for gates within the Clifford algebra, which includes the Hadamard gate (H), controlled not (CNOT), and $S = \mathrm{diag}(1, e^{i\pi/2})$, we can apply the gate to all the physical qubits in order to apply the same gate to the protected qubit. The increase in depth of the circuit is hence only due to gates outside of the Clifford algebra.

In Section III, we showed that AE can replace MC-based simulations. The resulting quantum circuits are too deep for quantum computers without error correction due to the controlled $\mathcal{Q}^{2^j}$ operators (see Fig. 4). We, therefore, anticipate that AE-based applications will require fault-tolerant quantum computers. By contrast, the optimization and ML applications discussed in Sections IV and V that are based on variational quantum circuits as in the VQE and the QAOA could be executed on near-term noisy quantum computers. However, these heuristic algorithms do not provide a theoretical guarantee as does AE. Further research is thus needed to fully understand under what conditions they will outperform their classical counterparts.

C. PRECISION AND SAMPLE COMPLEXITY

Generally, the higher probability of outputting the correct answer is required, the more "shots," or repetitions of the execution of quantum circuit followed by measurement, are needed. In some cases (e.g., HHL), because the solution is encoded in the probability amplitudes of the quantum states, one may need to perform quantum state tomography to obtain the complete solution. The quantum state tomography requires an exponential number of shots in the number of

TABLE 5. Algorithms can improve computational efficiency, accuracy, and addressability for defined use case

	Quantum Algorithm	Description	Impact	Needs	Simulation	Optimization	ML
VQE	Variational Quantum Eigensolver	Use energy states to calculate the function of the variables to optimize	Procedure to assign compute-intensive functions to quantum and those of controls to classical	Qubit number increases significantly with problem size		x	
QAOA	Quantum Approximate Optimization	Optimize combinatorial style problems to find solutions with complex constraints	Simplify analysis clauses for constraints and provide robust optimization in complex scenarios	Uncertain ability to expand to more optimization classes		x	
AE	Quantum Amplitude Estimator	Create simulation scenarios by estimating an unknown property, Monte Carlo style	Handle random distributions directly, instead of only sampling, to solve dynamic problems quadratically speeding up simulations	High Quantum Volume required for good efficiency	x	x	x
QSVM	Quantum Support Vector Machines	Supervised machine learning for high dimensional problem sets	Map data to quantum-enhanced feature space to enable separation and better separate data points to achieve more accuracy	Runtime can be slowed by kernel computation and data structure			x
HHL	Harrow, Hassidim, and Lloyd	Estimate the resulting measurement of large linear systems	Solve high dimensional problems speeding up exponentially calculations	Hard to satisfy prerequisites and high measurement costs to recover solutions		x	x
QSDP	Quantum Semidefinite Programming	Optimize a linear objective over a set of positive semidefinite matrices	Estimate quantum system states with less measurements to exponentially speedup in terms of dimension and constraints	High Quantum Volume required for good efficiency		x	

qubits involved and, hence, can diminish the exponential advantages of the subroutines. In many algorithms, the error also depends on the number of qubits used in the output register.

For example, for the phase estimation mentioned in Section III-A, the probability of not determining phase angle to an accuracy of s bits, i.e., up to error 2^{-s}, using $s + p$ qubits for the output is

$$\epsilon(s, p) = 1 - \frac{1}{2^{2(p+s)-2}} \sum_{l=1}^{2^{p-1}} \frac{1}{1 - \cos \frac{\pi(2l-1)}{2^{p+s}}}. \tag{30}$$

While the expression may be difficult to read, it is essentially positive, in that it suggests that the error rate decays exponentially with the extra p qubits. Especially, when many instances of phase estimation are used sequentially, the error propagation may still be a cause for concern, though, and it may get progressively more difficult to analyze the error. Still, estimates of forward error of more complex algorithms [117] are available.

VII. CONCLUSION

There are a number of computationally challenging problems in financial services that are demanding in terms of required precision or runtime. For these, we have outlined three problem classes.

1) One class are *optimization* problems that scale exponentially limiting their resolution in a given time frame. The holistic problem-solving approach to optimization problems of quantum computers raises the potential to find better solutions in a smaller number of steps.
2) A second class are *ML* problems, where one faces complex data structures that hinder classification or prediction accuracy. The multidimensional data modeling capacity of quantum computers may allow us to find better patterns, with increasing accuracy.
3) A third class are *simulation* problems, where there are time limits to perform sufficient scenario tests to find the best potential solution. Efficient sampling methods leveraging quantum computers may require less samples to reach a more accurate solution faster.

TABLE 6. Financial services focus areas and algorithms

Financial Services	Example Problems	Solution Approach	Quantum Algorithm
Asset Management	Option Pricing Portfolio risk	Simulation	AE
Investment	Portfolio Optimization	Optimization	Combinatorial: VQE, QAOA
Banking	Portfolio Diversification Issuance: Auctions		Continuous: QSDP AE
Retail & Corporate Banking	Financial Forecasting Credit Scoring (e.g. SME Banking) Financial Crimes: Fraud + AML	Machine Learning	QSVM HHL AE

For each of them, we have introduced quantum algorithms, which can be applied to specific problems in there. Table 5 summarizes the quantum algorithms introduced, their applicability for the three problem classes, and their advantages and challenges. In addition, for the three initial focus areas in financial services, asset management, investment banking, retail and corporate banking, example problems, and applicable quantum algorithms are summarized in Table 6.

Quantum computers and the algorithms that leverage them may help to solve hurdles and challenges arising in the financial industry given increasing demand for more sophisticated risk analysis, dynamic client management, constant updates to market volatility, and faster transaction speeds.

Finally, we have also demonstrated the performance of quantum algorithms on IBM Quantum back-ends for three specific applications. In general, simulation, optimization, and ML are among the areas where we may demonstrate an advantage of quantum computing over classical computing for certain applications first.

ACKNOWLEDGMENT

Rudy Raymond would like to acknowledge Erik Altman of IBM Research for providing the credit-card transaction dataset and Hiroshi Yano of Keio University for his help in analyzing the dataset. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at https://www.ibm.com/legal/copytrade

REFERENCES

[1] R. LaRose, “Overview and comparison of gate level quantum software platforms,” *Quantum*, vol. 3, 2019, Art. no. 130, doi: 10.22331/q-2019-03-25-130.

[2] Wikipedia, Cloud-based quantum computing, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Cloud-based-quantum-computing

[3] R. Orús, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” *Rev. Phys.*, vol. 4, 2019, Art. no. 100028, doi:10.1016/j.revip.2019.100028.

[4] S. Bravyi, D. Gosset, and R. Koenig, “Quantum advantage with shallow circuits,” *Science*, vol. 362, no. 6412, pp. 308–311, 2018, doi: 10.1126/science.aar3106.

[5] D. J. Egger, R. G. Gutiérrez, J. C. Mestre, and S. Woerner, “Credit risk analysis using quantum computers,” 2019, *arXiv:1907.03044*.

[6] P. Rebentrost, B. Gupt, and T. R. Bromley, “Quantum computational finance: Monte Carlo pricing of financial derivatives,” *Phys. Rev. A*, vol. 98, no. 2, Aug. 2018, Art. no. 022321, doi: 10.1103/physreva.98.022321.

[7] N. Stamatopoulos *et al.*, “Option pricing using quantum computers,” *Quantum*, vol. 4, 2020, Art. no. 291, doi: 10.22331/q-2020-07-06-291.

[8] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Amplitude estimation without phase estimation,” *Quantum Inf. Process.*, vol. 19 no. 2, Jan. 2020, Art. no. 75, doi: 10.1007/s11128-019-2565-2.

[9] S. Woerner and D. J. Egger, “Quantum risk analysis,” *npj Quantum Inf.*, vol. 5, no. 152019, Art. no. 15, doi: 10.1038/s41534-019-0130-6.

[10] L. Braine, D. J. Egger, J. Glick, and S. Woerner, “Quantum algorithms for mixed binary optimization applied to transaction settlement,” 2019, *arXiv:1910.05788*.

[11] C. Gambella and A. Simonetto, “Multi-block ADMM heuristics for mixed-binary optimization on classical and quantum computers,” *IEEE Trans. Quantum Eng.*, vol. 1, 2020, Art. no. 3102022, doi: 10.1109/TQE.2020.3033139.

[12] G. Nannicini, “Performance of hybrid quantum-classical variational heuristics for combinatorial optimization,” *Phys. Rev. E*, vol. 99, no. 1, 2019, Art. no. 013304, doi: 10.1103/PhysRevE.99.013304.

[13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[14] IBM, “Getting your financial institution ready for the quantum computing revolution,” 2018. [Online]. Available: https://www.ibm.com/thought-leadership/institute-business-value/report/quantumfinancial#

[15] IBM, “Exploring quantum computing use cases for financial services,” 2018. [Online]. Available: https://www.ibm.com/thought-leadership/institute-business-value/report/exploring-quantum-financial#

[16] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, 2018, Art. no. 79, doi: 10.22331/q-2018-08-06-79.

[17] D. E. Gottesman, “Stabilizer codes and quantum error correction,” Ph.D. dissertation, Dept. Phys., Math. Astron., California Inst. Technol., Pasadena, CA, USA, 1997.

[18] D. Aharonov and M. Ben-Or, “Fault-tolerant quantum computation with constant error rate,” *SIAM J. Comput.*, vol. 38 no. 4, pp. 1207–1282, 2008, doi: 10.1137/S0097539799359385.

[19] E. T. Campbell, B. M. Terhal, and C. Vuillot, “Roads towards fault-tolerant universal quantum computation,” *Nature*, vol. 549, no. 7671, pp. 172–179, 2017, doi: 10.1038/nature23460.

[20] IBM Quantum, *IBM Quantum systems overview*, 2020. [Online]. Available: https://quantum-computing.ibm.com/docs/manage/backends/

[21] A. Cross, L. Bishop, S. Sheldon, P. Nation, and J. Gambetta, “Validating quantum computers using randomized model circuits,” *Phys. Rev. A*, vol. 100, 2019, Art. no. 032328, doi: 10.1103/PhysRevA.100.032328.

[22] J. Gambetta, *IBM’s Roadmap for Scaling Quantum Technology*, 2020. [Online]. Available: https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/

[23] A. Peruzzo *et al.*, “A variational eigenvalue solver on a photonic quantum processor,” *Nat. Commun.*, vol. 5, 2014, Art. no. 4213, doi:10.1038/ncomms5213.

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

[25] V. Havlicek *et al.*, Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, 2019, doi: 10.1038/s41586-019-0980-2.

[26] J. Courbe, M. von Engel, C. O’Hara, J. Garvey, and J. Lyons, “Financial Services Technology 2020 and Beyond: Embracing disruption, 2020. [Online]. Available: https://www.pwc.com/gx/en/financial-services/assets/pdf/technology2020-and-beyond.pdf

[27] The Basel Committee on Banking Supervision (BCBS), “Basel III: The Liquidity Coverage Ratio and Liquidity Risk Monitoring Tools,” 2013. [Online]. Available: https://www.bis.org/publ/bcbs238.htm

[28] K. Bhaskaran, R. Chang, P. Dey, and J. Sanz, “Financial services industry challenges and innovation opportunities,” in *Proc. IEEE World Congr. Services*, vol. 1, pp. 224–229, 2019, doi: 10.1109/SERVICES.2019.00062.

[29] Reuters, “Some banks risk usd 240 million research loss as MiFID gives low-cost players a leg-up,” 2017. [Online]. Available: https://www.reuters.com/article/uk-banks-research/some-banks-risk-240-million-research-loss-as-mifid-gives-low-cost-players-a-leg-up-idUKKBN19H0C0

[30] Reuters, “EU banks face fines if they fall short on capital,” 2011. [Online]. Available: https://www.reuters.com/article/us-eu-banks/eu-banks-face-fines-if-they-fall-short-on-capital-idUSTRE76J21I20110720

[31] VOB, *The Consequences of Basel IV—A Quantitative Impact Study*, May 25, 2018.

[32] Nomura, The net stable funding ratio (NSFR)—What does it mean for you (2014), Accessed: Oct. 23, 2020. [Online]. Available: https://www.nomura.com/primefinance/assets/documents/nsfr.pdf

[33] McKinsey, Basel III and European banking: Its impact, how banks might respond, and the challenges of implementation, 2010. [Online]. Available: https://www.mckinsey.com//media/mckinsey/industries/financial%20services/our%20insights/basel%20iii%20now%20the%20hard%20part%20for%20european%20banks/26_basel_iii_and_european_banking.ashx

[34] J. Hull, *Options, Futures, and Other Derivatives*, 6th ed., Englewood Cliffs, NJ, USA: Prentice-Hall, 2006.

[35] F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *J. Political Econ.*, vol. 81, no. 3. pp. 637–654, 1973, doi: 10.1086/260062.

[36] B. Dupire, “Pricing with a smile,” *Risk Magazine*, pp. 18–20, 1994. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.320.5063

[37] P. P. Boyle, “Options: A Monte Carlo approach,” *J. Financial Econ.*, vol. 4, no. 3, pp. 323–338, 1977, doi: 10.1016/0304-405X(77)90005-8.

[38] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. New York, NY, USA: Springer-Verlag, 2003.

[39] Basel Committee on Banking Supervision, Minimum capital requirements for market risk, 2019. [Online]. Available: https://www.bis.org/bcbs/publ/d457.htm

[40] P. Glasserman, P. Heidelberger, and P. Shahabuddin, “Efficient monte carlo methods for value-at-risk,” vol. 2, 2000. [Online]. Available: https://www0.gsb.columbia.edu/faculty/pglasserman/Other/masteringrisk.pdf

[41] Basel Committee on Banking Supervision, “Revisions to the Basel II market risk framework,” 2009. [Online]. Available: https://www.bis.org/publ/bcbs158.htm

[42] C. Joy, P. P. Boyle, and K. S. Tan, “Quasi-Monte Carlo methods in numerical finance,” *Manag. Sci.*, vol. 42, no. 6, pp. 926–938, 1996, doi: 10.1287/mnsc.42.6.926.

[43] P. Glasserman and J. Li, “Importance sampling for portfolio credit risk,” *Manag. Sci.*, vol. 51, no. 11, pp. 1593–1732, 2005, doi: 10.1287/mnsc.1050.0415.

[44] A. Carrera Vazquez and S. Woerner, “Efficient state preparation for quantum amplitude estimation,” 2020, *arXiv:2005.07711*.

[45] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” *AMS Contemporary Math.*, vol. 305, pp. 53–74, 2002, doi: 10.1090/conm/305.

[46] A. Y. Kitaev, “Quantum measurements and the Abelian stabilizer problem,” 1995, *arXiv:9511026*.

[47] S. Aaronson and P. Rall, “Quantum approximate counting, simplified,” *Symp. Simplicity Algorithms*, pp. 24–32, 2019, doi: 10.1137/1.9781611976014.5.

[48] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, “Iterative quantum amplitude estimation,” 2019, *arXiv:1912.05559*.

[49] M. Rutkowski and S. Tarca, "Regulatory capital modelling for credit risk," *Int. J. Theor. Appl. Finance*, vol. 18, no. 5, 2015, Art. no. 1550034, doi: 10.1142/S021902491550034X.

[50] Basel Committee on Banking Supervision, "International convergence of capital measurement and capital standards," 2006. [Online]. Available: https://www.bis.org/publ/bcbs128.htm

[51] L. Grover and T. Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," 2002, *arXiv:0208112*.

[52] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," 2004, *arXiv:0410184*.

[53] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *Quantum Inf. Comput.*, vol. 6, pp. 351–369, 2006. [Online]. Available: https://dl.acm.org/doi/10.5555/2012086.2012090

[54] H. Abraham *et al.*, "Qiskit: An open-source framework for quantum computing," 2019, doi: 10.5281/zenodo.2562111.

[55] A. G. Fowler and C. Gidney, "Low overhead quantum computation using lattice surgery," 2018, *arXiv:1808.06709*.

[56] M. Plesch and C. Brukner, "Quantum-state preparation with universal gate decompositions," *Phys. Rev. A*, vol. 83, 2011, Art. no. 032302, doi: 10.1103/PhysRevA.83.032302.

[57] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Inf.*, vol. 5, 2019, Art. no. 103, doi: 10.1038/s41534-019-0223-2.

[58] Treasury, "An integrated approach to financial supply chain efficiency," 2013. [Online]. Available: https://www.treasury-management.com/article/4/270/2269/an-integrated-approach-to-financial-supply-chain-efficiency.html

[59] KPMG, Financial supply chain management, 2016. [Online]. Available: https://assets.kpmg/content/dam/kpmg/pdf/2016/04/transforming-financial-supply-chain.pdf

[60] DTCC, A different approach to accelerating settlement and reducing risk, 2018. [Online]. Available: http://www.dtcc.com/dtcc-connection/articles/2018/july/31/a-different-approach-to-accelerating-settlement-and-reducing-risk

[61] CNBC, 5 mistakes to avoid when rebalancing a portfolio, 2017. [Online]. Available: https://www.cnbc.com/2017/02/14/5-mistakes-to-avoid-when-rebalancing-a-portfolio.html

[62] IPE, Portfolio construction: A roadmap for portfolio rebalancing, Accessed: Oct. 23, 2020. [Online]. Available: https://www.ipe.com/portfolio-construction-a-roadmap-for-portfolio-rebalancing/41189.article

[63] CBInsights, Killing the i-bank: The disruption of investment banking, Accessed: Oct. 23, 2020. [Online]. Available: https://www.cbinsights.com/research/report/disrupting-investment-banking

[64] Efinancialcareers, This is who goldman sachs will be mostly hiring in 2020. [Online]. Available: https://news.efinancialcareers.com/us-en/3002977/goldman-sachs-technology-hiring

[65] OutsideInsight, JPMorgan, Goldman Sachs and more double down on tech innovation, Accessed: Oct. 23, 2020. [Online]. Available: https://outsideinsight.com/insights/jpmorgan-goldman-sachs-and-more-double-down-on-tech-innovation/

[66] F. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," 2018, *arXiv:1812.04170*.

[67] C. Huang, M. Szegedy, F. Zhang, X. Gao, J. Chen, and Y. Shi, "Alibaba cloud quantum development platform: Applications to quantum algorithm design," 2019, *arXiv:1909.02559*.

[68] R. Shaydulin and Y. Alexeev, "Evaluating quantum approximate optimization algorithm: A case study," in *Proc. 10th Int. Green Sustain. Comput. Conf.*, 2019, pp. 1–6, doi: 10.1109/IGSC48788.2019.8957201.

[69] G. G. Guerreschi and A. Y. Matsuura, "QAOA for max-cut requires hundreds of qubits for quantum speed-up," *Sci. Rep.*, vol. 9, 2019, Art. no. 6903, doi: 10.1038/s41598-019-43176-9.

[70] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," 2018, *arXiv:1812.01041*.

[71] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," 2018, *arXiv:1811.08419*.

[72] A. Gilliam, S. Woerner, and C. Gonciulea, "Grover adaptive search for constrained polynomial binary optimization," 2019, *arXiv:1912.04088*.

[73] F. Brandao, R. Kueng, and D. Stilck França, "Faster quantum and classical SDP approximations for quadratic binary optimization," 2019, *arXiv:1909.04613*.

[74] S. Aaronson, *Quantum Computing since Democritus*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[75] F. G. S. L. Brandao and K. M. Svore, "Quantum speed-ups for solving semidefinite programs," in *Proc. IEEE 58th Annu. Symp. Found. Comput. Sci.*, 2017, pp. 415–426. doi: 10.1109/FOCS.2017.45.

[76] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, "Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning," 2017, *arXiv:1710.02581*.

[77] J. Van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, "Quantum SDP-solvers: Better upper and lower bounds," *Quantum* vol. 4, 2020, Art. no. 230, doi: 10.22331/q-2020-02-14-230.

[78] J. van Apeldoorn and A. Gilyén, "Improvements in quantum SDP-solving with applications," in *Proc. 46th Int. Colloq. Autom., Lang., Program.*, 2019, pp. 99:1-99:15, doi: 10.4230/LIPIcs.ICALP.2019.99.

[79] I. Kerenidis and A. Prakash, "A quantum interior point method for LPs and SDPs," *ACM Trans. Quantum Comput.*, vol. 1, no. 1, Oct. 2020, doi: 10.1145/3406306.

[80] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[81] G. Cornuejols and R. Tutuncu, *Optimization Methods in Finance*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[82] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, "Convex optimization using quantum oracles," Quantum, vol. 4, 2020, Art. no. 220, doi: 10.22331/q-2020-01-13-220.

[83] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, "Quantum lower bounds by polynomials," *J. ACM*, vol. 48, no. 4, 2001, Art. no. 778–797, doi: 10.1145/502090.502097.

[84] N. Moll *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Sci. Technol.*, vol. 3, no. 3, 2018, Art. no. 030503, doi: 10.1088/2058-9565/aab822.

[85] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, 2016, Art. no. 023023, doi: 10.1088/1367-2630/18/2/023023.

[86] E. Farhi, J. Goldstone, S. Gutmann, and H. Neven, "Quantum algorithms for fixed qubit architectures," 2017, *arXiv:1703.06199*.

[87] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving variational quantum optimization using CVaR," *Quantum*, vol. 4, 2020, Art. no. 256, doi: 10.22331/q-2020-04-20-256.

[88] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, vol. 4, Philadelphia, PA, USA: SIAM, 1990.

[89] C.-Y. Wang and D. Li, "Unified theory of augmented Lagrangian methods for constrained global optimization," *J. Global Optim.*, vol. 44, no. 3, 2009, Art. no. 433, doi: 10.1007/s10898-008-9347-1.

[90] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte, "Reachability deficits in quantum approximate optimization," *Phys. Rev. Lett.*, vol. 124, 2020, Art. no. 090504, doi: 10.1103/PhysRevLett.124.090504.

[91] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2019, pp. 1–8, doi: 10.1109/HPEC.2019.8916288.

[92] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, 1994, pp. 359–370, [Online]. Available: https://dl.acm.org/doi/10.5555/3000850.3000887

[93] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011, doi: 10.1561/2200000016.

[94] S. Gomez and J. P. Hennart, *Advances in Optimization and Numerical Analysis*, vol. 275. New York, NY, USA: Springer, 2013, doi: 10.1007/978-94-015-8330-5.

[95] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *Eur. J. Oper. Res.*, vol. 247, no. 1, pp. 124–136, 2015, doi: 10.1016/j.ejor.2015.05.030.

[96] S.-H. Li, D. C. Yen, W.-H. Lu, and C. Wang, "Identifying the signs of fraudulent accounts using data mining techniques," *Comput. Human Behav.*, vol. 28, no. 3, 2012, Art. no. 10021013, doi: 10.1016/j.chb.2012.01.002.

[97] P. Ram and A. G. Gray, “Fraud detection with density estimation trees,” in *Proc. Workshop Anomaly Detection Finance*, 2018, pp. 85–94. [Online]. Available: http://proceedings.mlr.press/v71/ram18a.html

[98] C. Doersch, “Tutorial on variational autoencoders,” 2016, *arXiv:1606.05908*.

[99] Realwire, KYC compliance costing banks Euro 50 million a year as financial sanctions soar, 2019. [Online]. Available: https://www.realwire.com/releases/KYC-compliance-costing-banks-50-million-a-year-as-financial-sanctions-soar

[100] WorldBank, “Non-performing loans are increasing in many European and Sub-Saharan African economies,” 2019. [Online]. Available: https://data.worldbank.org

[101] Pymnts, “Fintechs give banks a wake-up call with loan disbursements,” 2017. [Online]. Available: https://www.pymnts.com/disbursements/2017/loan-disbursements-for-smbs-alt-lending-vs-banks/

[102] Emerj, “AI for Customer Service in Banking—Current Applications,” 2019. [Online]. Available: https://emerj.com/ai-sector-overviews/ai-for-customer-service-in-banking-current-applications/

[103] IDC, “Worldwide Spending on Artificial Intelligence Is Expected to Double in Four Years, Reaching $ 110 Billion in 2024, According to New IDC Spending Guide,” 2020. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS46794720

[104] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” *Nature*, vol. 549, no. 7671, 2017, Art. no. 195202, doi: 10.1038/nature23474.

[105] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Phys. Rev. Lett.*, vol. 122, no. 4, 2019, Art. no. 040504. doi: 10.1103/physrevlett.122.040504.

[106] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, “Circuit-centric quantum classifiers,” *Phys. Rev. A*, vol. 101, no. 3, 2020, Art. no. 032308, doi: 10.1103/PhysRevA.101.032308.

[107] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” *Phys. Rev. A*, vol. 98, no. 3, Art. no. 032309, doi: 10.1103/physreva.98.032309.

[108] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, “Quantum embeddings for machine learning,” 2020, *arXiv:2001.03622*.

[109] J. T. Hancock and T. M. Khoshgoftaar, “Survey on categorical data for neural networks,” *J. Big Data*, vol. 7, no. 1, 2020, Art. no. 28, doi: 10.1186/s40537-020-00305-w.

[110] H. Yano, Y. Suzuki, R. Raymond, and N. Yamamoto, “Efficient discrete feature encoding for variational quantum classifier,” 2020, *arXiv:2005.14382*.

[111] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, “Dense quantum coding and a lower bound for 1-way quantum automata,” in *Proc. 31st Annu. ACM Symp. Theory Comput.*, 1999, pp. 376–383, doi: 10.1145/301250.301347.

[112] E. R. Altman, “Synthesizing credit card transactions,” 2019, *arXiv:1910.03033*.

[113] C. Zoufal, A. Lucchi, and S. Woerner, “Variational quantum Boltzmann machines,” 2020, *arXiv:2006.06004*.

[114] C. Ciliberto *et al.*, “Quantum machine learning: a classical perspective,” *Proc. Roy. Soc. A: Math.*, vol. 474, no. 2209, 2018, Art. no. 20170551, doi: 10.1098/rspa.2017.0551.

[115] N. Jiang, Y. F. Pu, W. Chang, C. Li, S. Zhang, and L. M. Duan, “Experimental realization of 105-qubit random access quantum memory,” *npj Quantum Inf.*, vol. 5, no. 1, 2019, Art. no. 28, doi: 10.1038/s41534-019-0144-0.

[116] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, “Optimal control, geometry, and quantum computing,” *Phys. Rev. A*, vol. 73, no. 6, 2006, Art. no. 062323, doi: 10.1103/PhysRevA.73.062323.

[117] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, “Quantum algorithms revisited,” *Proc. Roy. Soc. London. Ser. A: Math.*, vol. 454, no. 1969, pp. 339–354, 1998, doi: 10.1098/rspa.1998.0164.