Hindawi
Quantum Engineering
Volume 2023, Article ID 7810974, 10 pages
https://doi.org/10.1155/2023/7810974

*Research Article*

# BQ-Bank: A Quantum Software for Finance and Banking

**Hang Li**,[1] **Tonghao Xing**,[1] **Shijie Wei**,[1] **Zhiyuan Liu**,[1] **Jiang Zhang**,[1]
**and Gui-Lu Long**[1,2,3,4]

[1]*Beijing Academy of Quantum Information Sciences, Beijing 100193, China*
[2]*State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China*
[3]*Beijing National Research Center for Information Science and Technology and School of Information, Tsinghua University, Beijing 100084, China*
[4]*Frontier Science Center for Quantum Information, Beijing 100084, China*

Correspondence should be addressed to Jiang Zhang; zhangjiang@baqis.ac.cn and Gui-Lu Long; gllong@tsinghua.edu.cn

Received 21 October 2022; Revised 7 December 2022; Accepted 16 December 2022; Published 16 January 2023

Academic Editor: YuBo Sheng

The power of quantum computing may bring a revolution in finance and banking. Here, we present quantum software, BQ-Bank for option pricing, Value at Risk, portfolio optimization, and others. BQ-Bank can be run on a real quantum computer such as superconducting system or on an emulation system based on a classical computer with an interface. BQ-Bank, such as other quantum types of software, represents a new generation of the toolbox that likely brings disruptive innovations to the financial industry and banking market in the future. BQ-Bank also provides the classical Monte Carlo solution, so that users can compare their quantum results with classical ones directly. Our simulation results for a variety of examples show the superiority of quantum solutions.

## 1. Introduction

Based on quantum physical principles, such as quantum superposition and quantum entanglement, quantum computers [1, 2] can provide unprecedented powerful computational capabilities. Compared with classical algorithms, quantum algorithms possess tremendous speedup in solving certain problems, such as Shor's algorithm [3] for integer factorization, Grover's algorithm [4, 5] for unsorted database search, and Harrow–Hassidim–Lloyd (HHL) quantum algorithm [6] for linear systems of equations.

Quantum computers are expected to have a substantial impact on the financial industry and banking market [7, 8]. Financial institutions such as J. P. Morgan Chase, Barclays Bank, and Goldman Sachs Group have already established collaborations with quantum computing companies such as IBM and Amazon to explore the use of quantum computing in finance. Spanish bank CaixaBank has also joined hands with IBM to develop a financial asset risk analysis simulation project. Canada's BMO Financial Group and Scotiabank have partnered with Australian quantum computing startup Xanadu to develop quantum Monte Carlo algorithms to improve the efficiency of financial transactions and optimize real-time pricing. Recently, Everbright Technology, together with the Beijing Academy of Quantum Information Sciences and Boson Quantum Technology, has released a quantum portfolio product–Tiangong Jingshi (https://finworld.qboson.com) quantum computing quantitative strategy platform.

On the one hand, the concept of quantum computing can be mysterious to the general public, and performing quantum algorithms requires prior knowledge for people in the finance and banking sectors and is beyond their capabilities on their own; this limits the desire of the public to explore in the quantum computing world; on the other hand, experts in quantum computing are interested in how quantum computing can apply in the industry such as finance and banking but do not know how to realize them.

Here, we report a quantum software platform designed for the finance and banking industry, Biwon Quantum Bank (BQ-Bank) (http://biwonq.baqis.ac.cn/#/pages/finance). It includes four commonly used modules in finance and banking: option pricing, option revenues, Value at Risk (VaR), and portfolio optimization. With the graphical interface and preset data, one can access and use the previous four modules without any prior knowledge of quantum computing, helping the public especially financial workers to learn and perform applications of quantum computing in finance.

## 2. BQ-Bank

In this section, we present quantum algorithms that are required in finance and banking, and demonstrate how they are harnessed to solve problems in the modules. In each module, we first offer detailed descriptions of the problem and models, then introduce the solution with conventional classical methods of simulation, and a quantum solution. Finally, we compare the results of the conventional method with its quantum counterpart. More specially, in subsections 2.1 and 2.2, we introduce how quantum amplitude estimation algorithm can apply to calculating option pricing and VaR; and in subsection 2.3, we give a simple description of quantum approximate optimization algorithm (QAOA), a method that is believed to be adaptive to current noisy devices.

### *2.1. Option Pricing and Revenues*

*2.1.1. Black-Scholes Option Pricing Model.* An option is a right for the buyer in a contract, who can choose whether to exercise the right or not. Therefore, the option buyer needs to pay a certain price to the option seller for the option right. The price of option buying and selling is often determined by the option market exchange, which reflects the entire option market's assessment of the future price trend of the underlying asset. However, such a method is often not that efficient, and options market prices often fail to reflect their actual prices, especially in markets with smaller volumes. So, we need a price to guide options trading. On the other hand, a large number of speculators in the market need to evaluate the difference between the market price of the option and its actual value, to invest in option buying and selling.

The research on option pricing can be traced back to the year 1900, when French financial expert Laures Bachelier discussed option pricing in his doctoral dissertation [9]. Since then, various option pricing models have been proposed, such as the binomial option pricing model [10, 11], trinomial option pricing Model [12], and Black-Scholes option pricing model [13]. Among these, the Black-Scholes model is the most widely known, which was proposed by Fisher Black and Myron Scholes and furthermore developed by Robert Merton.

Generally, to use the Black-Scholes model to estimate the price of an option, say, European-style call and put option, some assumptions shall be made, such as the revenues of the underlying asset follows a normal distribution, the risk-free rate is constant, and investors can borrow unlimitedly at this rate, (more details can be found in "Section I: Option Pricing and Black-Scholes Model" of the supplementary material from section VIII of the main manuscript.)

*(1) Call Option.* The equation for pricing a call option based on the Black-Scholes model has the following form:

$$C = S \cdot N(d_1) - Ke^{-rT} N(d_2), \tag{1}$$

where $C$ is the call option price, $S$ is the current price of the underlying asset, $N$ is the standard normal cumulative distribution function, $K$ is the exercise price, $r$ is the risk-free interest rate, $T$ is the time of option expiration, and $d_1$ and $d_2$ are given as follows:

$$\begin{cases} d_1 = \dfrac{\ln S/K + \left(r + (1/2)\sigma^2\right)T}{\sigma\sqrt{T}}, \\ \\ d_2 = \dfrac{\ln S/K + \left(r - (1/2)\sigma^2\right)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}, \end{cases} \tag{2}$$

where $\sigma$ is the standard deviation of the stock's returns.

*(2) Put Option.* The equation for pricing a put option based on the Black–Scholes model represented by the previous same parameters can be expressed as follows:

$$P = Ke^{-rT}\left(1 - N(d_2)\right) - S\left(1 - N(d_1)\right), \tag{3}$$

where $P$ is the put option price.

A classical simulation method for calculating the option pricing value is Monte Carlo simulation, which is widely used for dealing with uncertainty in many aspects of business operations. In option pricing, Monte Carlo simulation can be used for sampling estimation to calculate the value of an option with multiple sources of uncertainties and random features.

*2.1.2. Quantum Solution.* Now, we show a quantum option pricing algorithm, as a quantum version of Monte Carlo solution to the option pricing problem. Considering European-style options, the procedure of the typical quantum algorithm for option pricing is as follows:

(1) Generate $M$ expiration date prices $\{X_1, X_2, \cdots X_M\}$ for the underlying asset by a stochastic process

(2) Calculate $M$ revenues of the option $f(X_i)$ at each price $X_i$

(3) Obtain the averaged revenues on the expiration date in a risk-neutral world:

$$\widehat{\mathbb{E}}_P[f(X)] = \frac{1}{M}\sum_{i=1}^{M} f(X_i). \tag{4}$$

(4) Calculate its present value $e^{-rT}\widehat{\mathbb{E}}_P[f(X)]$ which is the option price, where $r$ is the risk-free interest rate and $T$ is the option expiration date.

The quantum option pricing algorithm starts with generating the quantum state of the option revenues and the underlying asset distribution on the expiration date. Then, the quantum amplitude estimation algorithm [14–17] (QAE, for more details, see "Section II: Quantum Amplitude Estimation" in the supplementary material from section VIII of the main manuscript.) is utilized to obtain the option price.

Now, we explain how the QAE algorithm is used for option pricing. By applying an operator $|0\rangle_{n+1}$ to an $(n+1)-$ qubit system initialized in $|0_{n+1}$, the final quantum state reads

$$\mathscr{A}|0\rangle_{n+1} = \sum_{i=0}^{2^n-1}\sqrt{1-f(S_i)}\sqrt{p_i}|S_i\rangle_n|0\rangle + \sum_{i=0}^{2^n-1}\sqrt{f(S_i)}\sqrt{p_i}|S_i\rangle_n|1\rangle, \tag{5}$$

where $S_i$ can be encoded as the option price, $f(S_i)$ represents the revenue of option contracts at price $S_i$, and $p_i$ stands for the probability distribution of price $S_i$. Let $a$ be the probability of the $(n+1)$-th qubit staying in state $|1\rangle$, the value of $a$ can be estimated by the QAE algorithm as follows:

$$a = \sum_{i=0}^{2^n-1} f(S_i)p_i = \mathbb{E}[f(S)], \tag{6}$$

which is the expected value of the option on the expiration date in (4).

The previous equation shows how to obtain option value with the QAE algorithm in a simplified way; a more detailed and accurate procedure lists are as follows:

Step 1. Under the assumption of random walk theory, the price distribution of the underlying asset at the option expiration date $T$ is

$$p(S_T) = \frac{1}{S_T\sigma\sqrt{2\pi T}}\exp\left(-\frac{(\ln S_T-\mu)^2}{2\sigma^2 T}\right). \tag{7}$$

After sampling and discretizing of the distribution into $p_i$, they are encoded into the amplitudes of a certain $n-$ qubit basis, and the corresponding state can be efficiently prepared by parameterized quantum circuits [18]:

$$|\psi(\theta)\rangle_n = \sum_{i=0}^{2^n-1}\sqrt{p_i(\theta)}|i\rangle_n, \tag{8}$$

where $i$ corresponds to the price $S_T$ in (7).

Step 2. For the revenue function of $f(S_T) = \max(0, S_T - K)$ in the case of call option, the revenue function can be divided into two parts, $S_T < K$ and $S_T \geq K$. This is achieved by adding an extraqubit (the $(n+1)$-th qubit) to the system, distinguishing the two parts with $|0\rangle$ and $|1\rangle$ of the $(n+1)$-th qubit, respectively.

$$|\psi\rangle_n|0\rangle \rightarrow \sum_{i<K}\sqrt{p_i}|i\rangle_n|0\rangle + \sum_{i\geq K}\sqrt{p_i}|i\rangle_n|1\rangle. \tag{9}$$

Step 3. By encoding the revenue function $f(i)$ into a special function which satisfies $\widetilde{f}(i) \in [-1,1]$

$$\widetilde{f}(i) = 2\frac{f(i)-f_{\min}}{f_{\max}-f_{\min}} - 1, \tag{10}$$

and adding an ancillary qubit initialized in state $|0\rangle_a$, the first $(n+1)$ qubits can be used as control qubits to rotate the ancillary qubit as an angle in a function of $\widetilde{f}(i)$ as follows:

$$|i\rangle_n|1\rangle|0\rangle_a \rightarrow |i\rangle_n|1\rangle\left(\cos\left(c\widetilde{f}(i)+\frac{\pi}{4}\right)|0\rangle_a + \sin\left(c\widetilde{f}(i)+\frac{\pi}{4}\right)|1\rangle_a\right). \tag{11}$$

Step 4. After the measurement, we obtain the probability for which the $(n+1)$-th qubit and the ancillary qubit both in state $|1\rangle$:

$$P = \sum_{i=0}^{2^n-1} p_i\sin^2\left(c\widetilde{f}(i)+\frac{\pi}{4}\right). \tag{12}$$

As we know, the function $\sin^2(x+\pi/4)$ can be approximated as $x + 1/2$ while $x$ approaches zero. Thus, when the value of $c$ is small enough, the following approximation is valid:

$$\sin^2\left(c\widetilde{f}(i)+\frac{\pi}{4}\right) \approx c\widetilde{f}(i)+\frac{1}{2}. \tag{13}$$

By substituting (13) into (12), $P$ is shown as

$$P = c\frac{2(\mathbb{E}[f(S)]-f_{\min})}{f_{\max}-f_{\min}} - c + \frac{1}{2}. \tag{14}$$

Now, it is clear that $\mathbb{E}[f(S)]$ can be calculated with the probability $P$, and it follows that the result value of current option $e^{-rT}\mathbb{E}[f(S)]$ is also on a silver platter.

#### 2.1.3. Examples and Results.

In the previous subsection, one see that quantum computing plays an important role in option pricing. However, for the general public, due to the mysteriousness of quantum computing itself, and it is neither easy nor practical to learn quantum computing and quantum programming. To overcome this problem, we have established a BQ-Bank web page, which provides software applications related to option pricing with a friendly user interface. We illustrate our modules for option pricing and option revenues with concrete examples.

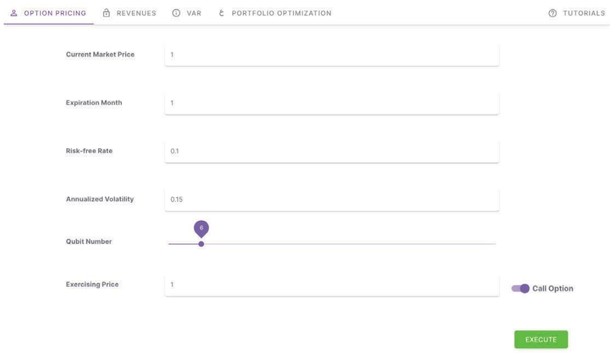
*(1) Option Pricing.* The user interface for the quantum option pricing module of European-style is shown in Figure 1. In this user interface, we provide six main inputs as follows:

(1) The current market price of the underlying asset

(2) The option contract's valid period

(3) The risk-free interest rate

(4) The annualized volatility



Current Market Price 1

Expiration Month 1

Risk-free Rate 0.1

Annualized Volatility 0.15

Qubit Number 6

Exercising Price 1 Call Option

Figure 1: The user interface for the option pricing problem.

(5) The option exercise price

(6) The choosing option of call or put options

In the backend, we provide three algorithms to calculate the option price as follows:

(1) Black–Scholes model

(2) Quantum algorithm

(3) Classic Monte Carlo algorithm

In conducting option pricing with the quantum algorithm, the users need to input the number of qubits (denoted with $n$) in the user interface. For the convenience of comparison, the sampling number of the classic Monte Carlo algorithm is set to $2^n$. After clicking the execution button, the calculation result is shown in Figure 2, with respect to the standard value shown by the Black–Scholes model, and the result of option pricing from the quantum algorithm is better than that from the classical Monte Carlo algorithm. Because of the randomness involved in both algorithms, the results will vary slightly from run to run.

| | |
|---|---|
| Black-Scholes Model | 0.0217 |
| Quantum Algorithm | 0.0209 |
| Monte Carlo Simulation | 0.0198 |

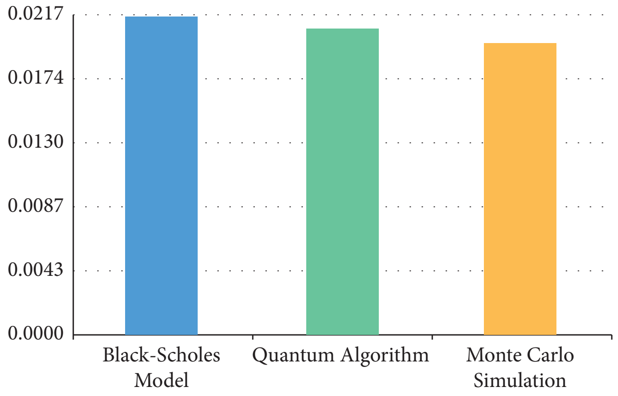
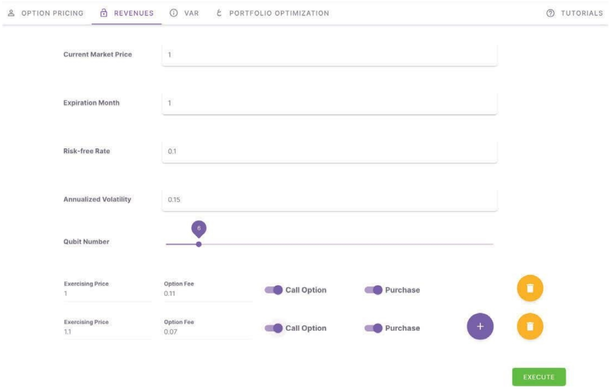


Figure 2: A result histogram of the option pricing with three algorithms. From this example, we see that the result obtained from the quantum algorithm is closer to that from the Black–Scholes model compared with that from the Monte Carlo simulation, indicating that the quantum algorithm is more accurate than the classical Monte Carlo algorithm.

*(2) Option Revenues.* Due to the discrepancy between the option price calculated by the theoretical model and the option price in the real market, speculators can arbitrage by buying and selling options. An application for calculating the expected option revenues strategy is also integrated into BQ-Bank, and the quantum algorithm in the backend is still based on the previous quantum option pricing algorithm. Specifically, this module calculates the expected option revenues portfolio by estimating the price of each option with a quantum option pricing algorithm and comparing it with the option market price.

Similar to the inputs of option pricing, one can find the user interface for calculating expected option revenues in Figure 3. In addition to four inputs (i.e., the current market price of the underlying asset, the option contract's valid period, the risk-free interest rate, and the annualized volatility of the asset), there are also other user input parameters required to construct an option portfolio. For example, to add another pattern of options by clicking the "+" sign icon at the end of its option input line, and to delete one pattern of options by clicking its respective "trash can" icon. Each pattern of options contains the option for call/put option, the option for buying/selling options, its exercising price and the option fee.



Current Market Price 1

Expiration Month 1

Risk-free Rate 0.1

Annualized Volatility 0.15

Qubit Number 6

Exercising Price 1 Option Fee 0.11 Call Option Purchase

Exercising Price 1.1 Option Fee 0.07 Call Option Purchase

Figure 3: The user interface for the expected option revenues problem.

After executing the calculation, the result is shown in Figure 4. From the figure, we can get the estimated value of the expected revenues obtained by the three algorithms for the option portfolio we set. With respect to the standard value shown by the Black-Scholes model, the result with the quantum algorithm is better than the classical Monte Carlo algorithm's. Also, due to the randomness involved in both algorithms, the results will vary slightly from run to run.

### 2.2. Value at Risk (VaR)

*2.2.1. Classical Solution.* VaR is a parameter used to assess investment losses [19], and is often used as a risk measure of risk management. A more specific definition gives that the value of the maximum expected loss of an asset portfolio due to

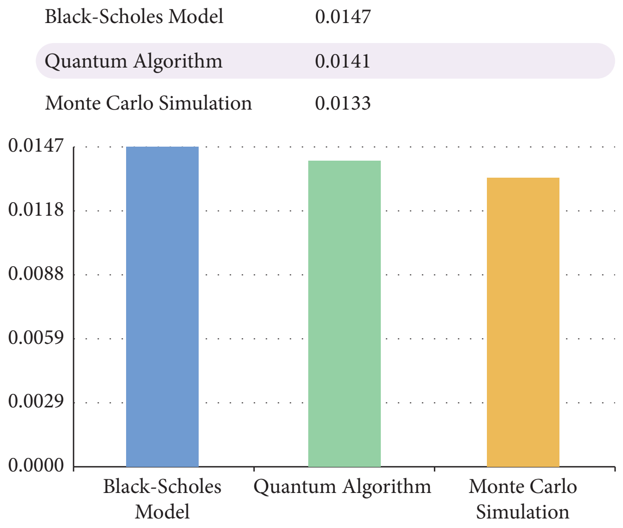


FIGURE 4: A result histogram of the expected option revenues with three methods, the Black–Scholes model, quantum algorithm, and Monte Carlo simulation, which shows the result of our quantum algorithm is closer to the Black–Scholes model's than the Monte Carlo simulation's, indicating the quantum algorithm method is more accurate than the classical Monte Carlo method.

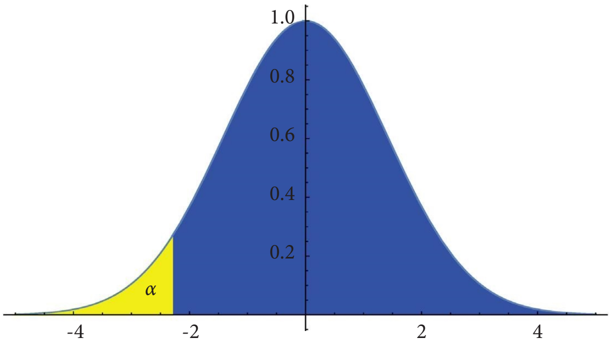


FIGURE 5: A loss probability distribution diagram of VaR.

market price changes within a given confidence interval during the holding period, as depicted in Figure 5, takes the form of

$$\mathrm{VaR}_\alpha(X) = -\inf\{x \in \mathbb{R}\colon F_X(x) > \alpha\}, \tag{15}$$

where $X$ is the profit and loss distribution (note the loss value is negative and the profit value is positive), $\alpha$ is the given probability, and $F_X$ is its cumulative distribution function.

There are three kinds of classical VaR algorithms, namely, Historical VaR (H-VaR), Parametric VaR (P-VaR), Monte Carlo VaR (MC-VaR) [20]. However, these algorithms rely on different assumptions, so that their final VaR values are often different.

*(1) H-VaR*. When we have a certain amount of historical data in our hands but do not know the probability distribution of asset losses, we can assume that the asset's future losses and gains are merely repetitions of past situations. Then, we use the VaR of the historical data to estimate the future VaR. The specific method is to sort the historical data by size, and determine the position of VaR in the sorting according to the product of the probability $\alpha$ and the size of the historical data.

*(2) P-VaR*. If we assume that the asset's loss distribution follows a normal distribution, we can use historical data to estimate the mean and variance of the normal distribution, then use the normal distribution function to calculate the VaR at the probability $\alpha$.

*(3) MC-VaR*. Here, we use the Monte Carlo method to simulate future asset market changes. A variety of risk factors need to be considered in the simulation process, which makes the simulation methods diverse and difficult. Among them, the geometric Brownian motion model is relatively simple. In the process of computer simulation, the geometric Brownian motion discrete model is

$$\Delta S_t = \mu \cdot \Delta t + \sigma \cdot S_t \cdot \epsilon \cdot \sqrt{\Delta t}, \tag{16}$$

where $S_t$ is the price of the asset at time $t$, $\Delta S_t$ is the first-order difference of the prices, $\mu$ is the mean value of historical returns, $\sigma$ is the standard deviation of historical returns, and $\Delta t$ is the unit time of returns. In the MC-VaR method, we randomly select $\epsilon$ according to the normal distribution $N(0,1)$ at each step of the time evolution of the asset holding period, so as to calculate sampling points on the expiration date.

*2.2.2. Quantum Solution.* In this subsection, we show how the QAE algorithm applies in calculating VaR [21, 22]. We discretize a random variable $X$ and then map it into a qubit system $\{0,1,2,\ldots,M-1\}, M = 2^m$, where $m$ is the number of sampled qubits, so any random variable $X$ can be represented by a quantum state with probability $p_i$ in state $|i\rangle$:

$$\begin{gathered} |\psi\rangle = \sum_{i=0}^{M-1} \sqrt{p_i}|i\rangle, \\ \sum_{i=0}^{M-1} p_i = 1, p_i \in [0,1]. \end{gathered} \tag{17}$$

Consider a quantum operation $F$ acting on sampling qubits and an ancillary qubit

$$F|i\rangle|0\rangle = |i\rangle\left(\sqrt{1-f(i)}\ |0\rangle + \sqrt{f(i)}\ |1\rangle\right), \tag{18}$$

where function $f(i) \in [0,1]$. Similar to (5), we can get the following equation:

$$|\psi'\rangle = F|\psi\rangle|0\rangle = \sum_{i=0}^{M-1} \sqrt{1-f(i)}\sqrt{p_i}|i\rangle|0\rangle + \sum_{i=0}^{M-1} \sqrt{f(i)}\sqrt{p_i}|i\rangle|1\rangle. \tag{19}$$

Employing the QAE algorithm, we can obtain the quantum state probability $\sum_{i=k+1}^{M-1} p_i$ with the ancillary qubit in state $|1\rangle$.

If we define function $f(i)$ as

$$f(i) = \begin{cases} 0, & i > k, \\ 1, & i \leq k. \end{cases} \tag{20}$$

Equation (19) can be rewritten as

$$|\psi'\rangle = \sum_{i=k+1}^{M-1} \sqrt{p_i}\,|i\rangle|0\rangle + \sum_{i=0}^{k} \sqrt{p_i}\,|i\rangle|1\rangle. \tag{21}$$

Thus, we can get the probability of the ancillary qubit in state $|1\rangle$, $\sum_{i=0}^{k} p_i = F_X(x \leq k)$. When $x \leq k$, there exists $F_X(x \leq k_\alpha) \geq \alpha$, which means $k_\alpha$ is the VaR with a confidence of $\alpha$ and a precision of $O(M^{-1})$, showing the quantum algorithm has a quadratic speedup compared to the classical MC-VaR method $O(M^{-1/2})$.

*2.2.3. Examples and Results.* Now, we show how three different classical VaR algorithms, H-VaR, P-VaR, and MC-VaR, and our quantum algorithm perform for the same instance. As we can see, in the case studied in Figure 6, we choose the normal distribution as our model, 0.95 confidence and 30 holding days of assets which are combined of two assets, China Bao An and Ping An Bank with 10 of each are purchased.

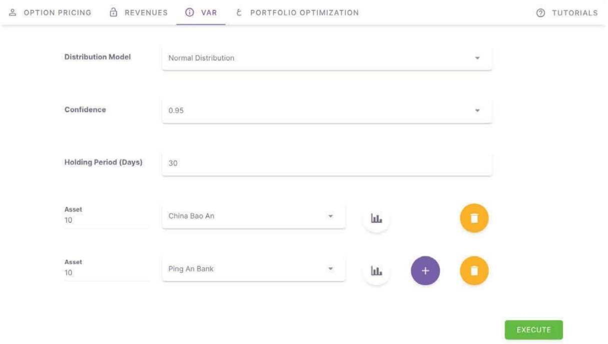
The results for both classical and quantum methods in different sampling points are shown in Figure 7, where we see that H-VaR and P-VaR are independent of the sampling points, which is quite intuitively obvious since both are calculated mainly based on historical data of assets without any sampling. As to MC-VaR, its VaR approaches to P-VaR with sampling points increasing, and even at 512 sampling points, there are still a small difference between P-VaR and MC-VaR. While the VaR with quantum algorithm approaches P-VaR monotonically with sampling points increasing and at 32 or larger sampling points, the VaR with the quantum algorithm is exactly equal to P-VaR. As the result shown in this instance, the VaR with the quantum algorithm is closer to P-VaR than that with the Monte Carlo method at the same sampling point, and we argue that the quantum algorithm performs better than the Monte Carlo method.

### 2.3. Portfolio Optimization

*2.3.1. Model.* Portfolio optimization [23] is an important problem that many investors will encounter. It requires investors to invest in candidate financial assets in order to minimize the risk under the condition of a certain return or maximize the investment return under the condition of a certain risk. If the number of candidate assets is very large, the classical algorithms used to solve such problems usually have high time complexity and are difficult to implement. Thanks to the powerful computing performance of quantum computing, the quantum algorithm can give the portfolio of assets with the optimal budget number among the candidate assets [24].

We use the mean-variance combinatorial optimization method, and the model for this problem can be expressed as

$$\begin{aligned} &\min_{x\in\{0,1\}^n} qx^T Cx - \mu^T x, \\ &\text{subject to: } 1^T x = b, \end{aligned} \tag{22}$$



Distribution Model: Normal Distribution

Confidence: 0.95

Holding Period (Days): 30

Asset 10: China Bao An

Asset 10: Ping An Bank

Figure 6: The user interface of a VaR instance. One can select the input parameters for the VaR model, such as the distribution model, preferred confidence, and holding period. The objective assets can be selected from a preset asset list and support adding or deleting an interest asset.

where

(i) $x \in \{0,1\}^n$ in quantum algorithms represents the vector of binary decision variables. If the asset $i$ is purchased, $x_i = 1$ and $x_i = 0$ when not. $x_i \in [0,1]$ in classical algorithms is the weight of purchasing the corresponding asset, and $\sum_i x_i = 1$.

(ii) $\mu \in \mathbb{R}^n$ represents the expected return of each asset.

(iii) $C \in \mathbb{R}^{n\times n}$ represents the covariance between the expected returns of each asset.

(iv) $q > 0$ is the risk factor, which indicates the investor's investment risk preference.

(v) 1 represents a $n$-dimensional vector whose elements are all 1 s.

(vi) $b$ represents the investment budget, which is the number of assets purchased from $n$ assets in quantum algorithms case (assuming the price of each asset is 1). Usually, $b = 1$ in classical algorithms case.

*2.3.2. Quantum Solution.* According to the equation of the problem model in (22), the loss function is given as follows:

$$L_x = q\sum_i \sum_j C_{ji} x_i x_j - \sum_i x_i \mu_i + A\left(b - \sum_i x_i\right)^2, \tag{23}$$

in which the first term represents the risk of the investment, the second term is the return of the investment, and the last term is a penalty function with $A$ as the penalty parameter and is usually set large.

Our portfolio optimization scheme uses a combination strategy of classical and quantum methods. First, it uses quantum algorithms to select the optimal portfolio of assets with a budgeted number of $n$ candidate assets. Then, it uses classical algorithms to calculate the assets selected by quantum algorithms (i.e., QAOA). Finally, it predicts the optimal strategy of purchasing assets and achieves maximum returns. With the aid of quantum algorithm, we can filter out the assets that do not need to be purchased, and the

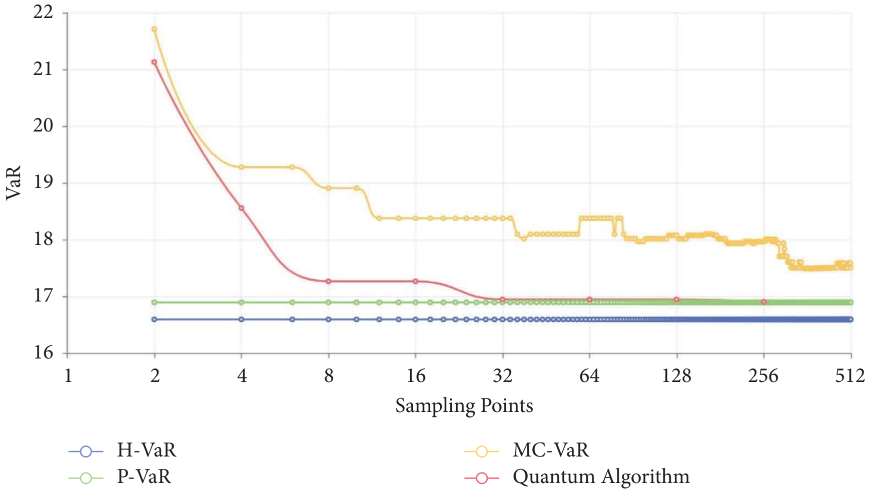


Figure 7: The result curves of 4 different methods for calculating VaR. The blue, green, yellow, and red curves show how historical VaR (H-VaR), parametric VaR (P-VaR), Monte Carlo VaR (MC-VaR), and VaR with quantum algorithm performing, respectively, at different sampling points.

difficulty in implementing the classical algorithm is greatly reduced.

QAOA [25, 26], which can be classified as a variational quantum algorithm (VQA) [27, 28], is a quantum algorithm with a wide range of applications in the recent Noisy Intermediate-Scale Quantum (NISQ) [29] era. QAOA was proposed by Edward Farhi et al. in 2014 to approximately solve the combinatorial optimization problems. For a combinatorial optimization problem, which can be defined by a binary string of $n$ bits $z = z_1 \ldots z_n$, the aim is to determine the string that minimizes or maximizes the classical objective function $C(z)$: $\{+1, -1\}^n \mapsto \mathbb{R}_{\geqslant 0}$.

QAOA can transform a combinatorial problem into an optimization problem. By mapping the binary variables into quantum spin operators $\sigma_i^z$, the classical objective function can be changed into a quantum Hamiltonian:

$$H_C = C(\sigma_1^z, \sigma_2^z, \ldots, \sigma_n^z), \tag{24}$$

whose ground state corresponds to the optimal solution of the problem.

In the case of QAOA with $p$ layers, we usually initialize the state of the quantum system as $| > +^{\otimes n}$ and then alternately act the problem Hamiltonian $H_C$ and the mixed Hamiltonian on the quantum system as follows:

$$H_B = \sum_{i=1}^{n} \sigma_x^i, \tag{25}$$

which results in a wave function with $2p$ variational parameters as follows:

$$|\varphi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \cdots e^{-i\beta_1 H_B} e^{-i\gamma_1 H_C} |+\rangle^{\otimes n}. \tag{26}$$

The expected value of $H_C$ under this variational wave function can be obtained by measuring the quantum system:

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \varphi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | H_C | \varphi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle, \tag{27}$$

then, solve the optimal parameter $(\gamma^*, \beta^*)$ on a classical computer, achieving

$$F_p(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) = \min_{\boldsymbol{\gamma},\boldsymbol{\beta}} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}). \tag{28}$$

In the ideal case, the quantum state $|\varphi(\gamma^*, \beta^*)\rangle$ should be the optimal solution, but it should be noted that the depth $p$ of a parameterized quantum circuit is not infinite, so it is likely that

$$F_p(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) > \min_{\boldsymbol{\gamma},\boldsymbol{\beta}} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}). \tag{29}$$

So, $|\varphi(\gamma^*, \beta^*)\rangle$ contains only the approximate optimal solution. Suppose $|\varphi(\gamma^*, \beta^*)\rangle$ is a superposition of $l$ computational basis states, i.e.,

$$|\varphi(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)\rangle = c_1|e_1\rangle + c_2|e_2\rangle + \cdots + c_l|e_l\rangle. \tag{30}$$

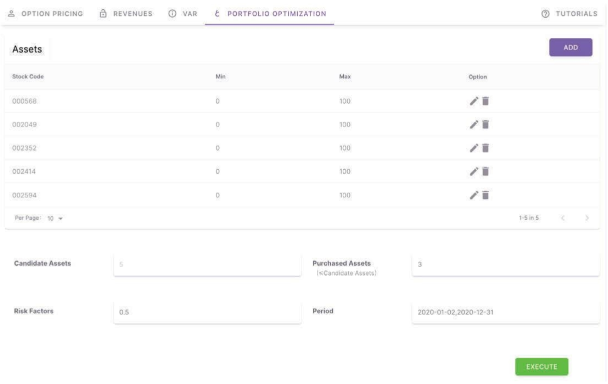
Usually, the greater of the probability that the ground state $|c_i|^2$ is measured in the eigenstate of $H_c$, the greater of the probability that its corresponding bit string $e_i$ is the optimal solution.

*2.3.3. Examples and Results.* In conducting portfolio optimization with quantum algorithms, we made two assumptions about the problem to simplify the model:

(1) All assets should have the same price (assumed to be 1)

(2) The given budget should be an integral multiple of the price of an asset, and it needs to be all used for investment, which means that $b$ assets need to be purchased

In this case, the portfolio optimization problem under these two assumptions corresponds to choosing a subset of $b$ assets from $n$ assets. We need to convert the loss function shown in (23) into a Hamiltonian to complete the encoding



Assets

| Stock Code | Min | Max | Option |
|---|---|---|---|
| 000568 | 0 | 100 | ✎ 🗑 |
| 002049 | 0 | 100 | ✎ 🗑 |
| 002352 | 0 | 100 | ✎ 🗑 |
| 002414 | 0 | 100 | ✎ 🗑 |
| 002594 | 0 | 100 | ✎ 🗑 |

Per Page: 10 ▾ 1-5 in 5 ‹ ›

Candidate Assets: 5

Purchased Assets (≤Candidate Assets): 3

Risk Factors: 0.5

Period: 2020-01-02,2020-12-31

Figure 8: The user interface of portfolio optimization at the preparation stage.

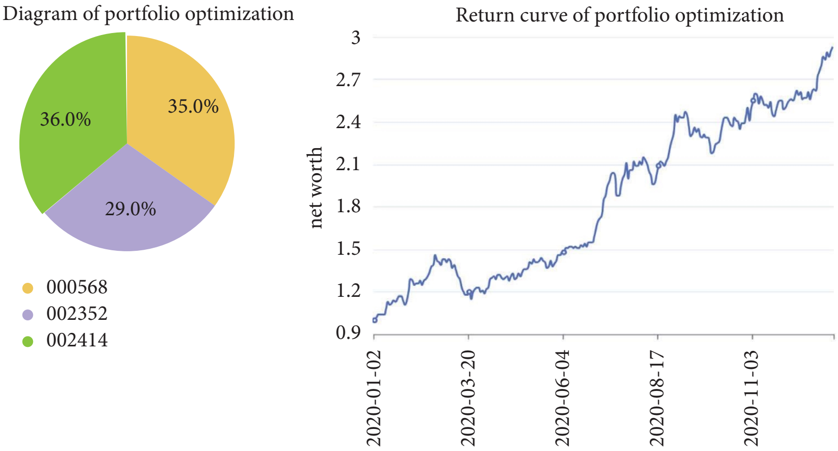


Figure 9: Result diagram of an optimal portfolio and its corresponding history return curve of an example as shown in Figure 8.

of the portfolio optimization problem. Because the value of the binary variable $x_i$ (0 or 1) indicates whether the asset needs to be purchased, we need to construct a Hamiltonian whose eigenvalues corresponding to them. Since the eigenvalue of the Pauli operator $\sigma^z$ is $\pm 1$, we can construct the Hamiltonian as $(\mathbb{I} - \sigma^z)/2$, which corresponds to two eigenvalues 0 and 1. So, one can use the mapping $x_i \mapsto \mathbb{I} - \sigma_i^z/2$ to convert the loss function $L_x$ into a Hamiltonian whose ground state corresponds to the optimal solution of the portfolio optimization problem.

As mentioned previously, we use a variational quantum algorithm, especially QAOA, to approximately solve the ground state of a Hamiltonian. Here, we list the procedure how we solve the portfolio optimization problem with QAOA and other algorithms:

*(1) Preparation*. For each candidate asset x, given its preferred maximum share $\max_x$ and minimum share $\min_x$ (where $\max_x, \min_x \in [0, 1]$) of the combinatorial assets, purchased assets budget $b\,(b \leqslant n)$, risk factor $q$, and historical period of all candidate assets $[t_i, t_f]$, as shown in Figure 8.

*(2) Assignment*. When the minimum share of an asset satisfies $\min_x > 0$, it means the asset will definitely be purchased and will be assigned into the assets set $X_1$, and the total number of the assets set $X_1$ is defined as $n_1$. Else when $\min_x = 0$, the asset needs to be determined by QAOA whether it needs to be purchased or not, and which asset will assigned into another assets set $X_2$.

*(3) Optimal Selection with QAOA*. Since there are already $n_1$ assets that have been determined to be purchased for sure, only $(b - n_1)$ assets need to be selected from the set $X_2$ using QAOA. According to the mapping $x_i \mapsto I - \sigma_i^z/2$, the loss function $L_x$ can be converted into a systematic Hamiltonian $H_C$ with qubit number $(n - n_1)$, and the ground state of this Hamiltonian is the optimal solution, where the mixing Hamiltonian is $H_B = \sum_{j=1}^{n-n_1} \sigma_j^x$.

The following procedure is divided into two parts:

(i) *Quantum Part*. Prepare a variational state $|\varphi(\theta, \beta) = U(\theta, \beta)| +^{\otimes n}$ on a quantum computer, where $U(\theta, \beta) = \prod_{l=1}^{p} e^{-i\beta_l H_B} e^{-i\gamma_l H_C}$, and then obtain the expectation of Hamiltonian $|\varphi(\theta, \beta)\rangle = U(\theta, \beta) + \rangle^{\otimes n}$.

(ii) *Classical Part*: Update $\theta, \beta$ with the classical optimization algorithm, such as COBYLA, and go back to the quantum part until $\lambda$ approximately reaches a minimum. The optimal combination of assets to be purchased, as defined $X_2'$, is obtained from the variational state at this point, and the optimal combination of assets to be purchased is $X_1 + X_2'$.

*(4) Proportion Optimization*. Finally, a trust-constr constraint algorithm is used to give the optimal proportion of the assets in the optimal portfolio $X_1 + X_2'$ to the constraints of every asset as assigned in the preparation step. Figure 9 shows the optimal result of an input example corresponding to Figure 8, including the optimal portfolio and its corresponding history return curve.

## 3. Summary

We have developed a quantum software BQ-Bank, which comprises 4 modules: option pricing, option revenues, VaR, and portfolio optimization. In each module, we start with the underlying problems and models, then we discuss the conventional algorithms and their quantum solution, and at the end we take an example and compare the results of the conventional algorithms with their quantum solution. The results show that the quantum solution usually outperforms the conventional algorithms. With the succinct design and the preset data set of BQ-Bank, people with or without quantum background can access and complete their interest tasks on a silver platter without trained skills.

## Data Availability

The data sets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank Dr. Zai-sheng Lin for helpful discussions. This work was supported by the National Natural Science Foundation of China (Grant nos. 12004206 and 12005015).

## Supplementary Materials

For details of the Black–Scholes model and the Quantum Amplitude Estimation algorithm, see the supplementary material. (*Supplementary Materials*)

## References

[1] B. Paul, "The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines," *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, 1980.

[2] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6/7, 1982.

[3] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th annual symposium on foundations of computer science*, pp. 124–134, IEEE, Santa Fe, NM, USA, November 1994.

[4] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, pp. 325–328, 1997.

[5] G. L. Long, "Grover algorithm with zero theoretical failure rate," *Physical Review A*, vol. 64, no. 2, Article ID 022307, 2001.

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, Article ID 150502, 2009.

[7] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: overview and prospects," *Reviews in Physics*, vol. 4, Article ID 100028, 2019.

[8] H. Tang, A. Pal, T.-Yu Wang, Lu-F. Qiao, J. Gao, and X.-M. Jin, "Quantum computation for pricing the collateralized debt obligations," *Quantum Engineering*, vol. 3, no. 4, p. e84, 2021.

[9] L. Bachelier, "Théorie de la spéculation," *Annales Scientifiques de l'Ecole Normale Superieure*, vol. 17, pp. 21–86, 1900.

[10] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option pricing: a simplified approach," *Journal of Financial Economics*, vol. 7, no. 3, pp. 229–263, 1979.

[11] R. J. Rendleman and B. J. Bartter, "Two-state option pricing," *The Journal of Finance*, vol. 34, no. 5, pp. 1093–1110, 1979.

[12] P. P. Boyle, "A lattice framework for option pricing with two state variables," *Journal of Financial and Quantitative Analysis*, vol. 23, no. 1, pp. 1–12, 1988.

[13] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.

[14] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Contemporary Mathematics*, vol. 305, pp. 53–74, 2002.

[15] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, "Amplitude estimation without phase estimation," *Quantum Information Processing*, vol. 19, no. 2, pp. 75–17, 2020.

[16] A. Scott and P. Rall, "Quantum approximate counting, simplified," in *Symposium on Simplicity in Algorithms*,vol. 24, SIAM,32 pages, SIAM, 2020.

[17] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, "Iterative quantum amplitude estimation," *Npj Quantum Information*, vol. 7, no. 1, pp. 52–56, 2021.

[18] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *Npj Quantum Information*, vol. 5, no. 1, pp. 103–109, 2019.

[19] Y. YamaiT. Yoshiba et al., "On the validity of value-at-risk: comparative analyses with expected shortfall," *Monetary and Economic Studies*, vol. 20, no. 1, pp. 57–85, 2002.

[20] G. Paul, P. Heidelberger, and P. Shahabuddin, "Efficient Monte Carlo methods for value-at-risk," *Mastering Risk*, vol. 2, pp. 5–18, 2000.

[21] D. J. Egger, R. Garcia Gutierrez, J. C. Mestre, and S. Woerner, "Credit risk analysis using quantum computers," *IEEE Transactions on Computers*, vol. 70, no. 12, pp. 2136–2145, 2021.

[22] S. Woerner and D. J. Egger, “Quantum risk analysis,” *Npj Quantum Information*, vol. 5, no. 1, pp. 15–18, 2019.

[23] D. Durand, “Portfolio selection: efficient diversification of investments,” *The American Economic Review*, vol. 50, no. 1, pp. 234–236, 1960.

[24] G. Nannicini, “Performance of hybrid quantum-classical variational heuristics for combinatorial optimization,” *Physical Review A*, vol. 99, no. 1, Article ID 013304, 2019.

[25] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” 2014, https://arxiv.org/abs/1411.4028.

[26] L. Zhou, S. T. Wang, S. Choi, H. Pichl2.1er, and M. D. Lukin, “Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices,” *Physical Review X*, vol. 10, no. 2, Article ID 021067, 2020.

[27] A. Peruzzo, J. McClean, P. Shadbolt et al., “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, pp. 4213–4217, 2014.

[28] N. Moll, P. Barkoutsos, L. S. Bishop et al., “Quantum optimization using variational algorithms on near-term quantum devices,” *Quantum Science and Technology*, vol. 3, no. 3, Article ID 030503, 2018.

[29] M. Brooks, “Beyond quantum supremacy: the hunt for useful quantum computers,” *Nature*, vol. 574, no. 7776, pp. 19–21, 2019.