

Higher-Order Portfolio Optimization with Quantum Approximate Optimization Algorithm

Valter Uotila

Aalto University & University of Helsinki
valter.uotila@aalto.fi

Julia Ripatti

Aalto University
julia.1.ripatti@aalto.fi

Bo Zhao

Aalto University
bo.zhao@aalto.fi

Abstract—Portfolio optimization is one of the most studied optimization problems at the intersection of quantum computing and finance. In this work, we develop the first quantum formulation for a portfolio optimization problem with higher-order moments, skewness and kurtosis. Including higher-order moments leads to more detailed modeling of portfolio return distributions. Portfolio optimization with higher-order moments has been studied in classical portfolio optimization approaches but with limited exploration within quantum formulations. In the context of quantum optimization, higher-order moments generate higher-order terms in the cost Hamiltonian. Thus, instead of obtaining a quadratic unconstrained binary optimization problem, we obtain a higher-order unconstrained binary optimization (HUBO) problem, which has a natural formulation as a parametrized circuit. Additionally, we employ realistic integer variable encoding and a capital-based budget constraint. We consider the classical continuous variable solution with integer programming-based discretization to be the computationally efficient classical baseline for the problem. Our extensive experimental evaluation of 100 portfolio optimization problems shows that the solutions to the HUBO formulation often correspond to better portfolio allocations than the classical baseline. This is a promising result for those who want to perform computationally challenging portfolio optimization on quantum hardware, as portfolio optimization with higher moments is classically complex. Moreover, the experimental evaluation studies QAOA’s performance with higher-order terms in this practically relevant problem.

Index Terms—portfolio optimization, higher-order binary optimization, QAOA, higher moments

I. INTRODUCTION

A key goal of quantum algorithm development is to identify and construct well-motivated problems that can be solved faster or more efficiently on quantum computers than on classical hardware in real-life applications. At the intersection of quantum computing and finance, portfolio optimization is one of the most studied and promising problems in this regard [1]–[17]. Some of the reasons for its popularity are its central role in finance and the fact that it has a straightforward quantum computational formulation as a quadratic unconstrained binary optimization (QUBO) problem. This article presents a more advanced higher-order portfolio optimization problem for quantum computers and benchmarks its performance with the standard *quantum approximate optimization algorithm* (QAOA) [18]. The problem formulation includes encoding of higher-order moments: skewness and kurtosis. Additionally, we consider realistic integer variables and a capital-based budget constraint.

Our problem formulation can be divided into two parts: the higher-order unconstrained binary optimization formulation and the classical continuous variable formulation. A solution to both problems is an allocation, which describes how many assets an investor should buy, staying close to the budget. Comparing the expected returns of the allocations and the leftover budgets provides precise metrics for evaluating these two methods. Higher-order unconstrained binary optimization problems are eigenvalue problems that can be solved with QAOA. However, QAOA has not been extensively experimentally evaluated in these types of real-world higher-order optimization use cases. The classical continuous variable solution is a standard classical algorithm to solve portfolio optimization problems [19]. It is efficient because it utilizes continuous variables, which are generally easier to optimize than integer variables [20], [21].

The contributions of this work are as follows.

- We develop the first quantum portfolio optimization problem that includes higher moments (skewness and kurtosis), making the optimization problem more realistic.
- Our evaluation shows that the solution quality of the higher-order binary optimization formulation is often better than the solution quality of the classical continuous variable formulation with integer programming-based discretization. This finding motivates further research of higher-order binary optimization with quantum methods.
- The experimental evaluation benchmarks QAOA on 100 instances of higher-order binary portfolio optimization problems and identifies several challenges associated with its application.

a) *Related work on quantum and finance*: Finance is one of the key areas where quantum computing is believed to produce value even in the near future [3], [4], [22]. So far, portfolio optimization has been the most studied problem in finance with quantum computing [1]–[4], [6]–[10], [23] and its connection to spin glass energy minimization problems was suggested as early as 1998 [5]. Interestingly, higher-order polynomials or higher-order problems have not been discussed in the literature, except in [7], where they are briefly mentioned in the context of developing more suitable penalty functions for the budget constraint.

Table I reveals that no previous work has benchmarked portfolio optimization with QAOA using the realistic capital-

TABLE I: Quantum portfolio optimization approaches. If some works have not evaluated problems or used only a small number of them, this indicates that the research focus has been theoretical.

Paper	Integer vars	Capital budget	#Problems evaluated	Method
[7]	Yes	No	-	Annealing
[11]	Yes	Yes	1000	Counterdiabatic
[15]	Yes	Yes	4	Annealing
[2]	No	No	20	QAOA
[17]	No	No	209	QAOA
[5]	Yes	Yes	3	Theory
[12]	No	No	1	QAOA
[6]	No	No	210	Annealing
[14]	Yes	No	6	Various
[1]	Yes	Yes	3	VQE
[16]	No	No	2	Walks
[8]	Yes	Yes	-	HHL
[24]	Yes	Yes	6	HHL++
[25]	No	No	-	QAOA
[9]	No	No	20	Annealing

based budget constraint and integer variables as we do in this work. None of the earlier works have employed higher-order moments. This is also the first time classical continuous-variable portfolio optimization with an integer programming-based discretization has been compared to quantum formulations.

b) Related work on higher-order binary optimization:

In this work, we apply a higher-order unconstrained binary optimization (HUBO) model [26]. The HUBO problems resemble QUBO problems except that higher-degree interactions between the binary variables are allowed. Formally, given a binary variable vector $x = (x_1, \dots, x_n)$ of length n and the index set $[n] = \{1, \dots, n\}$, the objective function is

$$f(x) = \sum_{S \subset V} \alpha_S \prod_{i \in S} x_i, \quad (1)$$

where $\alpha_S \in \mathbb{R}$ and S runs over all the subsets of $[n]$.

Without relying on quantum computing, HUBO problems can be addressed by using various classical methods, which often employ approximation methods, especially for larger problem sizes, due to the exponential nature of the problem. Integer programming methods can be utilized to approximate solutions to HUBO problems, but are limited by exponential complexity and do not guarantee exact feasibility [27]. Different learning-based methods, such as graph neural networks, can approximate HUBO solutions [28], but they might not generalize well to unseen problems and can require significant training data. Previously, simulated annealing has been used to solve HUBO problems [29]. While these classical approaches remain competitive in solving small- to medium-scale problem sizes, they struggle with scalability, which could be potentially addressed with quantum native approaches.

Solving HUBO problems on quantum computers is a small but growing area of interest in quantum optimization. Considering the previous work, [30] studied solving a higher-order graph coloring problem with QAOA. Bias-field digitized counterdiabatic quantum optimization [31], [32] has been developed as a method to improve solving higher-order problems and there are methods to optimize variational circuits

for higher-order binary problems [33]. Higher-order problems have also been used as a benchmark between QAOA and quantum annealing [34]–[37] and they have been used to understand the scaling of QAOA and its parameter concentration [38]. Some initial applications of HUBO problems have been join order selection in relational database optimization [39], formalizing search for practical matrix multiplication algorithms [40], and optimizing railway rescheduling [41].

This paper begins by reviewing the classical mean-variance portfolio optimization problem and its extension to higher moments. We briefly discuss discretization via integer programming before formulating the quantum optimization approach, focusing on binary encoding and capital-based budgets. We then present the experimental setup, results, and visualizations, followed by conclusions and future directions. The implementation is available on GitHub [42].

II. BACKGROUND

We use consistent notation for variables across the paper: $w_i \in [0, 1]$ are the continuous variables in the unit interval, $z_i \in \mathbb{Z}_{\geq 0}$ are non-negative integer variables, $x_i \in \{0, 1\}$ are binary variables and $s_i \in \{-1, 1\}$ are spin variables.

A. Markowitz mean-variance portfolio optimization

In the original Markowitz mean-variance portfolio optimization problem [43], we are given a set of assets and a budget. The goal is to invest the optimal amount in these assets to maximize the expected return or minimize the financial risk. The key information in portfolio optimization is the price data of the assets from a selected time period. Assume that we have n assets indexed with $i \in [n] = \{1, \dots, n\}$ and let p_i^t denote the price of the asset i at time t . By using this price data, the return r_i^{t+1} is calculated for each asset between time t and $t + 1$ as

$$r_i^{t+1} = \frac{p_i^{t+1} - p_i^t}{p_i^t}. \quad (2)$$

In real-life portfolio optimization, we are interested in the expected return values of assets over a selected time period, smoothing daily variation. Multiple methods exist to estimate the portfolio moments based on the returns in Eq. (2), and making accurate estimations is an active area of research. Some standard ways have been implemented in [19], and we rely on these regarding the first two moments. For each asset i , the mean returns can be computed either as an arithmetic mean

$$\mu_{A,f}^i = \frac{f}{m} \sum_{t=1}^m r_i^t. \quad (3)$$

or as a geometric mean [19]

$$\mu_{G,f}^i = \left(\prod_{t=1}^m (1 + r_i^t) \right)^{\frac{f}{m}} - 1, \quad (4)$$

where f is a so-called frequency. Usually, in finance, the mean return values are annualized. Therefore, we set the frequency as the number of trading days in a year, $f = 252$. In this work,

we choose the geometric mean for calculating the mean return values since it is the default in [19].

The mean returns $\mu = (\mu_i)_{i=1}^n$ form the so-called first moment. The second moment is the covariance. The covariance between assets i and j can be defined as

$$c_{ij} = \frac{f}{m-1} \sum_{t=1}^m (r_i^t - \mu_{A,f=1}^i)(r_j^t - \mu_{A,f=1}^j), \quad (5)$$

where we have taken into account annualization by scaling the regular covariance with a frequency of $f = 252$. The arithmetic mean $\mu_{A,f=1}^i$ is the mean value given in Eq. 3 using frequency $f = 1$. The covariance matrix is $c = (c_{ij})_{i,j=1}^m$. We will define the third and fourth moments later. Before that, we briefly explain how the standard mean-variance portfolio optimization is formulated.

Let $w = (w_i)_{i \in [n]}$ be the vector of continuous variables. We assume that $w \geq 0$, which means that short selling is not allowed, which is a common assumption [2]. The very first mean-variance portfolio optimization formulation aims to balance expected returns and risks so that either the returns are at least at a fixed level or the risks are at most at a fixed level. In the first case, the mean target return is fixed to $\mu_{\text{fix}} \in \mathbb{R}$, and we solve the following quadratic programming problem with linear constraints:

$$\begin{aligned} & \text{minimize} && w^\top c w \\ & \text{subject to} && w^\top \mu \geq \mu_{\text{fix}}, \\ & && 1^\top w = 1. \end{aligned} \quad (6)$$

The constraint $1^\top w = 1$ is the shorthand notation for $\sum_i w_i = 1$, which means that 100% of the budget is invested. This program has a quantum formulation presented in [8]. In other words, the program aims to minimize risk, as modeled using covariances, while maintaining a fixed level of returns. Alternatively, we can formulate an optimization program that maximizes the mean returns while keeping the risk at an acceptable level:

$$\begin{aligned} & \text{maximize} && w^\top \mu \\ & \text{subject to} && w^\top c w \leq q_{\text{fix}}, \\ & && 1^\top w = 1. \end{aligned} \quad (7)$$

Here, the value $q_{\text{fix}} \in \mathbb{R}$ is the value that controls the investor's risk level. Next, we further develop these formulations and connect them to quantum computing.

B. Discretization

The original portfolio optimization formulation is defined in terms of continuous weights $w_i \in [0, 1]$. The solution to the continuous variable problem indicates that an investor should invest $w_i \cdot 100\%$ of the budget in asset i . This creates a challenge, as we need to find a way to interpret the continuous weights as a discrete number of assets. This subsection reviews an integer programming-based method to discretize these continuous weights for a given budget [19].

In discretization, we need information about the current prices of the stocks. Let $p^\tau = (p_i^\tau)_{i=1}^n$ be the vector of closing prices from the last time moment τ to calculate the

optimal allocation of stocks. In discretization, the goal is to convert the continuous variable weight vector $w \in [0, 1]^n$ into a vector of integer variables $z \in \mathbb{Z}_{\geq 0}^n$. Here, the integer vector z represents the quantities of each stock to be purchased to achieve the optimal portfolio. Thus, the integer vector z is the variable to be optimized in the discretization process. Let C be the total capital and $C_{\text{extra}} = C - (p^\tau)^\top z$ be the remaining unallocated capital. Then, the optimization problem is given by

$$\begin{aligned} & \text{minimize} && C_{\text{extra}} + |Cw - z^\top p^\tau| \\ & \text{subject to} && C_{\text{extra}} + z^\top p^\tau = C, \\ & && z \in \mathbb{Z}_{\geq 0}. \end{aligned} \quad (8)$$

The solution to this optimization problem is an integer vector z describing the discrete allocation of stocks.

C. Introducing discrete variables

Asset prices and quantities are discrete in real life. Modeling them with continuous variables is computationally more efficient, but leads to certain problems and inaccuracies. Continuous variables may produce impractical solutions using real-life constraints, including minimum purchase sizes and indivisible assets. Continuous solutions might allocate small fractions of capital to many assets, which is infeasible in real markets due to transaction costs and indivisibility. The previous integer programming-based allocation does not necessarily fix these issues in practice because the discretization problem is separate from the optimization, which optimizes the weights.

In the following, we formulate the problem using integer variables, making the problem NP-hard. Instead of defining a weight vector $w = (w_i)_{i \in [n]}$, we define an integer variable vector $z = (z_i)_{i \in [n]}$ such that $z_i \in \mathbb{Z}_{\geq 0}$. The interpretation of this vector is as follows: after optimizing the portfolio, we should have z_i many assets of asset type i . This formulation eliminates the need for discretization, as we automatically obtain a discrete solution. The following subsections will express the classical portfolio optimization using integer variables. Later, we will explain how the integer variables can be mapped to binary variables, which are used in the quantum formulation.

D. Unconstrained mean-variance portfolio optimization

Compared to the previous linear and quadratic programs in Eq. (6) and Eq. (7), we can also optimize the mean and variance simultaneously without limiting their values in the constraints. The weights should be optimized to maximize the mean return from the assets while minimizing the risk. The parameter $q_0 > 0$ controls the investor's willingness to take risks and the balance between risks and mean returns. In this case, the unconstrained portfolio optimization problem becomes

$$\min_{z \in \mathbb{Z}_{\geq 0}^n} q_0 z^\top c z - z^\top \mu. \quad (9)$$

This unconstrained formulation assumes an unlimited, unrealistic budget, leading to an unbounded problem without a feasible solution. Therefore, we must include a budget constraint to obtain a viable solution.

E. Budget constraint

The budget constraint limits the total amount of capital that an investor can invest, and thus, it limits the number of assets that an investor can buy. This work assumes that the investor prefers to use the whole budget.

Considering the continuous variable formulation, the problem with the budget constraint becomes

$$\begin{aligned} & \text{minimize} && q_0 w^\top c w - w^\top \mu \\ & \text{subject to} && 1^\top w = 1. \end{aligned}$$

As we can see, this formulation does not depend on the capital but assumes that we invest 100% by constraining $1^\top w = 1$. The discretization program in Eq. 8 is solved to obtain a capital-based discrete allocation.

There are two slightly different ways to encode the budget for the discrete optimization cases: the budget can be expressed in terms of assets or in terms of capital. Let us first consider the budget, which is expressed in terms of assets. In other words, this means that the sum $1^\top z = \sum_i z_i$ for $z = (z_1, \dots, z_n)$ should be limited. Let $B \in \mathbb{Z}_{>0}$ denote the budget in terms of assets, i.e., the number of assets that the investor can buy. After including the budget constraint, the previous unconstrained portfolio optimization in Eq. 9 becomes constrained:

$$\begin{aligned} & \text{minimize} && q_0 z^\top c z - z^\top \mu \\ & \text{subject to} && 1^\top z = B, \\ & && z \in \mathbb{Z}_{\geq 0}^n. \end{aligned}$$

This formulation selects a combination of assets such that their total value is B .

The problem with the previous discrete formulation is that it assumes that every asset has the same price, which is a very unrealistic simplification. The budget should be given as an investor's capital, which is a real number that describes the amount of capital the investor can invest in the market.

Next, we describe how to encode a realistic capital-based budget. Let $C \in \mathbb{R}_{>0}$ be the investor's capital. Let $p^\tau = (p_i^\tau)_{i=1}^n$ again denote the vector of closing prices from the last time moment τ , i.e. p^τ are the current prices that the investor uses to buy the stocks. Now the relation between discrete quantities of assets z , their recent closing prices p^τ and the capital C (monetary budget) can be written as follows

$$\sum_{i=1}^n p_i^\tau z_i = z^\top p^\tau = C. \quad (10)$$

Using the capital-based budget, the discrete optimization problem obtains a slightly modified formulation:

$$\begin{aligned} & \text{minimize} && q_0 z^\top c z - z^\top \mu \\ & \text{subject to} && z^\top p^\tau = C, \\ & && z \in \mathbb{Z}_{\geq 0}^n. \end{aligned}$$

In real-life cases, this leads to the problem that there might not be an integer variable solution that satisfies the constraint $z^\top p^\tau = C$ exactly. We will discuss this problem in more detail later and concretely show how it affects the evaluation of different portfolios.

F. Higher moments

Mean-variance portfolio optimization assumes normally distributed returns, which often do not reflect reality. To better estimate risks, researchers have incorporated higher moments, skewness and kurtosis, into the model [44]–[46]. Positive skewness indicates a long right tail and it is desirable as it suggests a higher chance of significant positive returns. In contrast, high kurtosis reflects fat tails and sharp peaks, signaling a greater risk of extreme outcomes, which investors typically avoid. Therefore, we aim to maximize skewness and minimize kurtosis.

Expected returns form a one-dimensional vector, and covariances form a two-dimensional matrix. We define the coskewness of returns as a three-dimensional tensor $S = \{S_{ijk}\}_{i,j,k}$ for $i, j, k = 1, \dots, n$ as

$$S_{ijk} = \frac{\mathbb{E}[r_i - \mathbb{E}[r_i]]\mathbb{E}[r_j - \mathbb{E}[r_j]]\mathbb{E}[r_k - \mathbb{E}[r_k]]}{\sigma_i \sigma_j \sigma_k}, \quad (11)$$

and co-kurtosis of returns as a four-dimensional tensor $K = \{K_{ijkl}\}_{i,j,k,l}$ for $i, j, k, l = 1, \dots, n$ as follows:

$$K_{ijkl} = \frac{\mathbb{E}[r_i - \mathbb{E}[r_i]]\mathbb{E}[r_j - \mathbb{E}[r_j]]\mathbb{E}[r_k - \mathbb{E}[r_k]]\mathbb{E}[r_l - \mathbb{E}[r_l]]}{\sigma_i \sigma_j \sigma_k \sigma_l}. \quad (12)$$

To simplify the notation, we denote

$$\begin{aligned} S(z) &:= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n S_{ijk} z_i z_j z_k \text{ and} \\ K(z) &:= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n K_{ijkl} z_i z_j z_k z_l. \end{aligned}$$

Then, the skewness and kurtosis can be included in the optimization problem:

$$\begin{aligned} & \text{minimize} && q_2 K(z) - q_1 S(z) + q_0 z^\top c z - z^\top \mu \\ & \text{subject to} && z^\top p^\tau = C \\ & && z \in \mathbb{Z}_{\geq 0}^n. \end{aligned} \quad (13)$$

This approach yields a higher-order problem for more realistic portfolio optimization with integer variables and a budget constraint that reflects the actual capital. Moreover, this formulation is relatively close to the higher-order unconstrained binary optimization format in Eq. (1). We must still rewrite the constrained optimization problem as the equivalent unconstrained problem and translate the integer variables into binary variables. We describe this process in the following subsection, connecting this problem and its quantum formulation. Note that the same problem can be expressed in terms of continuous variables as

$$\begin{aligned} & \text{minimize} && q_2 K(w) - q_1 S(w) + q_0 w^\top c w - w^\top \mu \\ & \text{subject to} && 1^\top w = 1. \end{aligned} \quad (14)$$

Finally, we are left to choose the scaling values q_0 , q_1 , and q_2 . Finding suitable values is generally a problem in finance. In this work, we choose

$$q_0 = \frac{\text{risk aversion}}{2}, \quad q_1 = \frac{\text{risk aversion}}{6}, \quad q_2 = \frac{\text{risk aversion}}{24},$$

which are motivated by the so-called Edgeworth expansion for distributions with higher moments [47]. We choose risk aversion = 3.

Example II.1. Next, we show an example to demonstrate how the continuous variable algorithm works in practice. The same example data is used throughout the article. We fix two companies: the Walt Disney Company (DIS) and the Travelers Companies, Inc. (TRV). The stock market data covers 10 years, from January 1, 2015, to January 1, 2025. The budget is randomly fixed at 723. First, we compute the expected returns vector with Eq. (4), covariance matrix with Eq. (5), coskewness tensor with Eq. (11), and cokurtosis tensors with Eq. (12). With this data, we construct the optimization problem defined in program (14). Solving the problem employs the Sequential Least Squares Programming, which is the default method in Scipy for these types of problems. The solution to the problem is a weight vector for the assets. In this case, the solution is DIS: 63% and TRV: 37%. These continuous weights are discretized with the integer program defined in Eq. (8). The final allocation is DIS: 4 and TRV: 1 with a left-over budget of 37.4.

III. QUANTUM FORMULATION

This section reformulates the classical higher-order portfolio optimization problem as a higher-order unconstrained binary optimization problem. Then, we discuss how to map the portfolio optimization problems to spin system energy minimization problems. Finally, we present the QAOA circuit, which is the last step before applying the standard QAOA quantum-classical optimization loop.

A. Encoding budget constraint

In quantum formulations for optimization problems, an unconstrained cost function is necessary. The constrained portfolio optimization problem can be written as an unconstrained problem by heavily penalizing cases that do not respect the constraint. In the case of portfolio optimization, this means introducing a term that is dependent on the remaining budget. The idea is that if the budget constraint is violated, it increases the value of the cost function, penalizing undesirable solutions. Hence, we include the term $\lambda(z^\top p^\tau - C)^2$ to the cost function in Eq. (13) and obtain the final cost function in terms of integer variables for the higher-order portfolio optimization, which can be written as

$$\min_{z \in \mathbb{Z}_{\geq 0}^n} q_2 K(z) - q_1 S(z) + q_0 z^\top c z - \mu^\top z + \lambda(z^\top p^\tau - C)^2, \quad (15)$$

where $\lambda > 0$ is a constant, determining the weight of the budget constraint. The constraint for the remaining budget is always non-negative. The penalizing constant λ should be sufficiently large so that we favor solutions that invest the entire budget. In this work, we explored $\lambda \in \{0.001, 0.01, 0.1, 0.9, 1.0, 10, 100, 1000\}$.

Unfortunately, this formulation does not distinguish between allocations that invest over the budget and allocations that

invest under the budget. They are both treated similarly because the term $(z^\top p^\tau - C)^2$ is symmetric with respect to such solutions. It is possible that investing over the budget might not be possible, and thus, we want to avoid such solutions fully. To address this problem, we refine the formulation further. We adopt the constraint encoding idea from [48]. Instead of considering a budget that is a real number, we approximate it by the closest integer. Let $M = \lfloor \log_2(C) \rfloor$, where C is the budget capital. Then, we can rewrite the budget as

$$\sum_{n=0}^{M-1} 2^n y_n + (C + 1 - 2^M) y_M, \quad (16)$$

where y_1, \dots, y_M are binary variables. Then, the penalty term that theoretically prevents us from investing more than we have in the budget is

$$\lambda(z^\top p^\tau - \sum_{n=0}^{M-1} 2^n y_n - (C + 1 - 2^M) y_M)^2.$$

The idea is to activate and deactivate the binary variables y_1, \dots, y_M depending on how much budget is used. However, the optimal point will never include allocations where we would invest more than the budget. Although this encoding adds only a logarithmic overhead to the encoding, it easily creates portfolio optimization problem instances with a relatively large number of qubits. Hence, we continue with the first encoding in Eq. (15) so that both cases are penalized equally. We will discuss the effects of this choice later in the discussion section.

B. Variable encoding in quantum formulation

Since we are developing the portfolio optimization on quantum computers, we must use binary variables. To use integer variables that are encoded with binary variables, we employ a similar method as in Eq. (16), which is the standard variable rewriting method from [48]. The same encoding was also used in [1]. The method translates integer variables into binary variables with a logarithmic overhead. Let z be an integer variable that obtains values at range $0, \dots, N$. First, choose $M = \lfloor \log(N) \rfloor$ with base 2 which implies $2^M \leq N \leq 2^{M+1}$. Then, instead of using the naive encoding with y_1, \dots, y_M binary variables, we can employ the logarithmic encoding as

$$z = \sum_{n=1}^N n y_n \rightarrow (N + 1 - 2^M) y_M + \sum_{n=0}^{M-1} 2^n y_n. \quad (17)$$

We must know the range $0, \dots, N$ for the integer variables to rewrite the integer variables as binary variables. In portfolio optimization problems, this range can be computed from the latest prices and the budget. For each integer variable z_i that is assigned to an asset i , we have the latest price p_i^τ . Then, the maximum amount of assets we can buy is $\lfloor p_i^\tau / B \rfloor$, i.e., investing the whole budget to a single company, providing us with the range. We obtain the final higher-order unconstrained binary optimization problem when this variable transformation is applied to the problem formulation in Eq. (15).

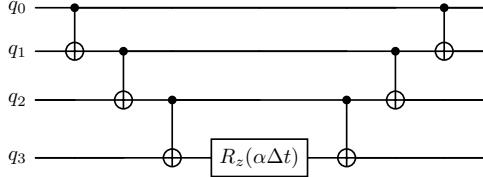


Fig. 1: Circuit block encoding higher-order term $\alpha Z_0 \otimes Z_1 \otimes Z_2 \otimes Z_3$

C. From optimization to spin system energy minimization

Following the binary formulation, the previously described portfolio optimization cost function in Eq. (15) is mapped to the Hamiltonian minimum eigenvalue problem. The goal is to find a configuration of spin variables that minimizes the energy of the whole system. In the portfolio optimization problem, the Hamiltonian can be written using Pauli-Z operators that have eigenvalues ± 1 . In binary optimization problems, using the mapping $x_i = \frac{1}{2}(1 - s_i)$, we map the binary variables $x_i \in \{0, 1\}$ to the spin variables $s_i \in \{-1, 1\}$. Therefore, the portfolio optimization problem can be written as finding the minimum eigenvalue of the Hamiltonian corresponding to the lowest cost function value.

D. QAOA circuit

In this subsection, we describe how to construct the parametrized QAOA circuit from the Hamiltonian that describes the higher-order unconstrained binary optimization problem in Eq. (15). Let I be the indexing set, i.e., in this case, the set containing the qubits. Currently, the Hamiltonian has the following form

$$H = \sum_{S \subset I} \alpha_S \prod_{i \in S} Z_i. \quad (18)$$

This is necessarily the same formulation that we expressed for abstract HUBO problems in Eq. (1). In this higher-order portfolio optimization case, the size of the sets satisfies $|S| \leq 4$, which follows from the dimension of the cokurtosis tensor. The other typical case is the QUBO problems where $|S| \leq 2$. As described in [49], the quantum circuit implementing $e^{-iH\Delta t}$ is relatively easy to construct, and the idea generalizes the method for QUBO problems. For example, consider that the Hamiltonian H contains a term $\alpha Z_0 \otimes Z_1 \otimes Z_2 \otimes Z_3$. This becomes the circuit block that is visualized in Fig. 1.

Additionally, the QAOA circuit consists of the mixer layer, and we employ the standard x -mixer. Finally, the expectation value of the cost Hamiltonian is measured. Modern quantum computing frameworks, such as Qiskit and Pennylane, perform the QAOA circuit construction and parametrization automatically, even for higher-order problems.

Example III.1. We continue Example II.1 by creating the corresponding higher-order binary optimization problem for the portfolio optimization problem, which includes Disney and Travelers Company. The closing prices for the companies are DIS: 111 and TRV: 240. The budget is 723. If the entire capital

is allocated to DIS, it is possible to buy at most $\lfloor 723/111 \rfloor = 6$ assets of Disney and, similarly, $\lfloor 723/240 \rfloor = 3$ assets of Travelers. At least 3 qubits are needed to encode the integers in the interval $[0, 6]$ for Disney and 2 qubits for integers in interval $[0, 3]$ for Travelers. Thus, the problem is encoded with 5 qubits.

Again, the expected returns, covariances, coskewness, and cokurtosis are computed. Based on this data, the corresponding unconstrained integer variable optimization problem described in Eq. (15) is constructed. Then, the integer variables are translated into binary variables by employing the substation in Eq. (17). For example, for Disney, the following substitution is performed at every position in the formulation

$$z \rightarrow x_0 + 2x_1 + 4x_2,$$

where z is the integer variable in interval $[0, 6]$ and x_0, x_1 and x_2 are binary variables. Three binary variables, i.e., three qubits, encode a single integer variable z . Again, these binary variables x_0, x_1 and x_2 representing the integer z are mapped into the spin glass ground state energy problem by using $x_i \leftrightarrow (1 - Z_i)/2$ as explained earlier. This is the final form for the Hamiltonian that encodes the higher-order portfolio optimization problem.

As in the standard QAOA, the higher-order problem becomes a parametrized quantum circuit. For this example, the circuit is partially visualized in Fig. 2. The figure includes an example of all terms with varying degrees from the first to fourth degree. The corresponding Pauli matrices from the final Hamiltonian are visualized on top of each term. The values in R_z -gates are obtained from the stock data and the initial parameter values for the algorithm.

For this example, the solution for the corresponding higher-order binary optimization problem is DIS: 0 and TRV: 3, and the left-over budget is 2.9. The problem is solved by finding the eigenvalues and eigenvectors of its Hamiltonian. In this particular case, QAOA could also find this optimal point.

IV. EXPERIMENTAL SETTING

This section describes the experimental setup that demonstrates the utility of the proposed algorithm compared to the classical baselines and mean-variance portfolio optimization. We describe the stock market data used to create the optimization problems, followed by the classical baseline implementations and the employed optimizers.

1) Data sets: We used the Dow Jones Industrial Average (DJIA), a stock market index that tracks 30 large, publicly traded U.S. companies. These companies represent a diverse range of industries, including technology, finance, healthcare, and consumer goods. The tickers (i.e., the shorthand stock market names) for the companies were downloaded from the Python library `pytickersymbols` [50], and the stock market data was downloaded from `yfinance` [51]. We downloaded data from January 1, 2015, to January 1, 2025, spanning ten years.

Next, we describe how we have constructed the portfolio optimization problems. We randomly sampled between 2 and 10 companies from the 30 companies in the index. Then, we

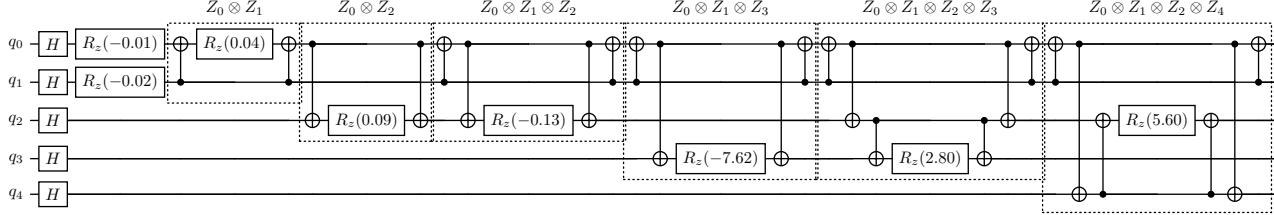


Fig. 2: Part of the QAOA circuit

randomly assigned a budget, which we limited to 6000. If the budget exceeded 6000, it was unlikely that the corresponding higher-order QAOA circuit for the portfolio optimization problem would have been simulable with the available resources. We calculated the number of qubits required for this portfolio optimization case. If the number of qubits was over 15, we excluded the optimization case from the dataset. Otherwise, it was included until every case from 6 to 15 qubits had ten random problems. This way, we obtained 100 random portfolio optimization problems that were ensured to fit a 15-qubit simulator.

Next, we argue that the stock market data included in the experimental dataset is not normally distributed, meaning it exhibits skewness and kurtosis. This motivates us to include the higher moments in the optimization problem. Fig. 3 demonstrates returns distributions for two companies, 3M and Travelers. Compared to the normal distribution, both distributions show clear positive kurtosis, i.e., the high peak. The distribution for Travelers also exhibits negative skewness, as the distribution has shifted to the right. Later in the discussion section, we will also explore how including higher-order moments produces more diverse portfolios, which can be considered a positive argument for including those moments.

2) *Classical algorithms and optimizers:* We employ classical optimizers in three different ways: classical optimizers solve the classical continuous variable baseline, another classical optimizer is used in the classical subroutine of QAOA, and a standard eigenvalue solver computes the exact spectrum for the HUBO problems when we have less than 14 qubits. When the problem consists of 14 or 15 qubits, we used the sparse eigenvalue solver, which does not return the whole spectrum but only approximates the smallest eigenvalues.

First, the classical continuous variable baseline utilizes SLSQP (Sequential Least Squares Programming), which is the default optimizer for SciPy if the optimization problem has constraints and bounds. Then, an integer program turns the continuous variables into discrete allocations. The integer program is solved with CVXPY [52], [53].

Second, we benchmarked six optimizers (Powell, SLSQP, COBYLA, CMA-ES, Nelder-Mead, and L-BFGS-B) in the QAOA's subroutine. A brief comparison of the optimizers for solving higher-order portfolio optimization with QAOA is presented in Figure 4. The lowest average objective function values can often be achieved with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [54]–[56], but this

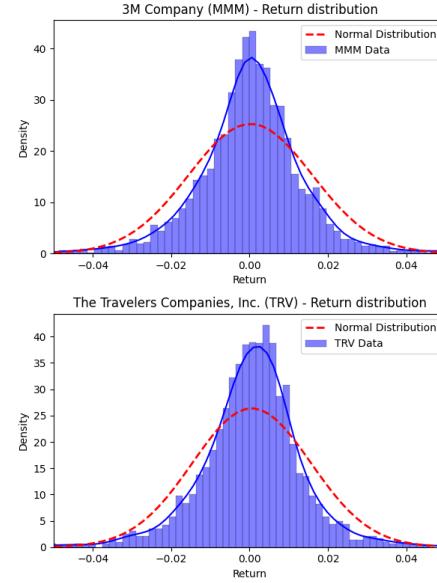


Fig. 3: Return distributions for two companies with one of the highest skewness and kurtosis terms. The actual distribution of companies' return values is presented in purple, and the normal distribution is denoted as a red dotted line. From the figures, it is clear that the assumption of a normal distribution of returns is often inaccurate; therefore, higher-order portfolio optimization may be a better approach to modeling the problem.

comparison lacks, for instance, noise and hyperparameter search. Nevertheless, we chose to investigate the performance of the QAOA algorithm with a CMA-ES optimizer. The optimizer is a stochastic gradient-free numerical optimization algorithm for complex (non-convex, ill-conditioned, multimodal, rugged, and noisy) optimization problems in continuous search spaces. We have not previously seen it employed as an optimizer for QAOA. We chose $\sigma = 0.1$ for the CMA-ES algorithm and limited the number of iterations, but the other hyperparameters were left to their default values.

Finally, we briefly describe the classical baselines. The classical baseline is to use the continuous variable higher-order program in Eq. 14. As noted earlier, this formulation is solved so that it does not invest over the budget, i.e., the budget constraint is strictly satisfied. On the other hand,

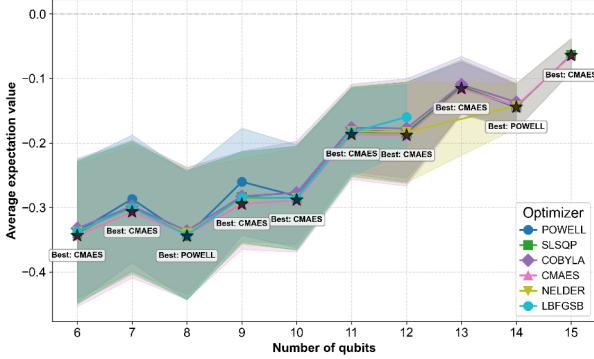


Fig. 4: Comparison between the different classical optimizers in optimizing the QAOA circuits solving the HUBO problem. Shaded areas represent 95% confidence intervals around the mean. The results are only presented for optimizers that can complete at least 8 of 10 problems for each qubit count.

the quantum formulation does not strictly satisfy the budget constraint because the constraint is part of the objective function. Thus, we have included the unconstrained continuous variable formulation

$$\min_{w \in [0,1]^n} q_2 K(w) - q_1 S(w) + q_0 w^\top \Sigma w - \mu^\top w + (1^\top w - 1)^2, \quad (19)$$

which is closer to the HUBO formulation. Both continuous variable solutions are discretized with the program in Eq. (8).

3) *Evaluation metric:* The integer optimization problem was described in Eq. (13). It consists of two parts: the objective function

$$f(z) = -q_2 K(z) + q_1 S(z) - q_0 z^\top c z + \mu^\top z, \quad (20)$$

and the budget constraint. The objective $f(z)$ provides a value that can be used to compare different solutions. On the other hand, in real-life portfolio optimization, we rarely find a portfolio allocation with the same value as the budget. Thus, the realistic final result always breaks the budget constraint, and this difference in budget should penalize the solution. Hence, the final evaluation for a solution z should be based on the value obtained from the objective function $f(z)$ and the difference in budget denoted by d_i .

Each method produces a discrete portfolio allocation z . The values from the cost function $f(z)$ without the budget constraint vary greatly between different portfolio optimization cases. Thus, we employ min-max normalization, which is commonly used in machine learning. This normalizes the value of $f(z)$ to the interval $[0, 1]$ and is defined

$$f_{\text{norm}}(z) = \frac{f(z) - \min \{f(z_i) \mid i \in I\}}{\max \{f(z_i) \mid i \in I\} - \min \{f(z_i) \mid i \in I\}}, \quad (21)$$

where I is the set containing the indices for different methods, i.e., classical methods, QAOA, and exact eigensolver, and z_i are the optimized allocations for each method $i \in I$.

V. RESULTS

In this section, we discuss our findings and visualize the problem characteristics that explain QAOA's performance in comparison with the exact solution and classical methods in this higher-order portfolio optimization problem. Due to space limitations, we have included the results from the smallest qubit counts, ranging from 6 qubits to 8 qubits, and the largest qubit counts, ranging from 12 to 15 qubits. The complete set of detailed results can be viewed on GitHub [42].

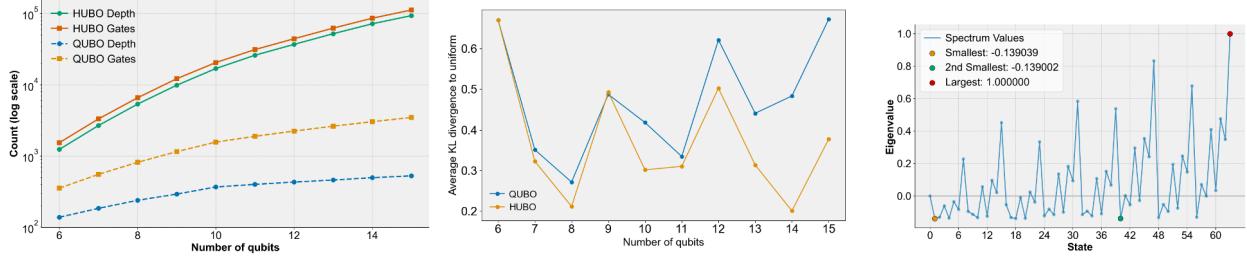
The selection of λ -values, which balances between the objective function in Eq. 20 and the budget constraint, affects the performance of QAOA. We performed the same optimization setup with varying λ values (0.001, 0.01, 0.1, 1.0, 0.9, 10, 100, and 1000). In the results, we present the QAOA performance with the most suitable λ . Due to space limitations, we do not visualize the optimal λ for each optimization problem. However, generally, values higher than 1 performed the best.

We also briefly discuss how much the QAOA algorithm improves the probability of measuring the state that we consider as a solution compared to the randomly sampled state. For each qubit we compute a so-called QAOA enhancement factor which is the observed highest probability after applying QAOA divided by the uniformly random probability of $1/2^n$, where n is the number of qubits. The average enhancement across all cases is 14.43, which means that on average it is 14.43x more probable to measure the minimizing state compared to a random state. The worst enhancement factor we observed was 3.79, and the largest was 108.94.

A. Results compared to the classical baseline

The results of solving selected portfolio optimization problems are presented in Fig. 6, where budget utilization is shown on the y-axis, and the x-axis shows the min-max normalized value of the cost function $f(z)$ without the budget constraint. The best solutions are in the upper right corner and along the 100% line. One of the findings is that our higher-order formulation for the portfolio optimization problem is often theoretically capable of encoding portfolio allocations whose quality outperforms that of solutions from classical baselines, as seen in Fig. 6 and in Table II. This result motivates further study of methods to solve HUBO problems on quantum hardware. The solutions to the HUBO problems have aimed to invest the whole budget, and this goal has sometimes come at the cost of smaller expected returns. Unfortunately, solving higher-order portfolio optimization problems with QAOA proved challenging, and only a small subset of problems matched the quality of the solutions obtained with exact methods or classical algorithms. This suggests that QAOA may require modifications to the optimization pipeline before it can effectively address more complex HUBO optimization problems.

We suspect that the QAOA's performance may be limited by the complex optimization landscape of portfolio problems, which often have many local minima. The current results demonstrate that the optimization becomes more challenging as the number of qubits increases. Fig. 5c shows the eigenvalue



(a) Gate count and depth comparison for QAOA. (b) Average KL divergence of QUBO and HUBO allocations compared to uniform. (c) Eigenvalues for a higher-order portfolio optimization problem with 6 qubits.

Fig. 5: Comparison of circuit metrics and allocation divergences for QUBO and HUBO settings.

TABLE II: Results of comparing the optimization approaches

Metric	Constrained classical	Unconstrained classical	QAOA	HUBO exact
95% \leq budget util. \leq 105%	74	70	19	100
min-max objective \geq 0.95	70	68	23	53
Intersection of both	50	46	3	53

spectrum for a 6-qubit instance, with several values near the optimum, suggesting a rugged landscape that makes these problems challenging for optimization. It will be a future research topic to improve the methods so that QAOA can tackle HUBO problems more efficiently.

B. Comparison to mean-variance portfolio optimization

Our implementation [42] enables users to solve the same portfolio optimization problems using either the mean-variance formulation, which is a QUBO problem, or the higher-order formulation, which is an HUBO problem. This subsection compares the HUBO problem to the corresponding QUBO problem. This clarifies the effect of including the higher-order terms and the complexity of the HUBO problem.

The first comparison is between the required gate counts and depths for each problem, which is presented in Fig. 5a. We did not perform optimization routines for the circuits, and the gates used are the standard for QAOA: Hadamard, CNOT, R_z , and R_x rotations. Both problems demonstrate scalability, which remains a challenge for current real quantum computing hardware.

Next, we compare the spectra of each problem by computing the exact spectra for HUBO and QUBO problems in the cases between 6 and 13 qubits and calculating the variance of the differences. This showed that for small qubit counts, the difference in the spectra between HUBO and QUBO problems is very small at level $10^{-9} - 10^{-11}$. This reveals that with the current weights for coskewness and cokurtosis, their effect on the values of the spectrum is small. The difference in the spectra becomes more dominant with increasing qubit counts.

Although the spectra for these optimization problems resemble each other, the allocations start differing substantially when the number of qubits and complexity increase. We visualize the difference in the allocations by computing how much they differ as distributions and how far they are from the

allocation, which would invest an equal amount in every asset. The results are in Fig. 5b. KL divergence, i.e., relative entropy, is a standard metric for comparing two distributions. The lower KL divergence value in the figure indicates that the distribution is closer to the uniform distribution. The closer the two plots are, the more similar allocations they provide on average.

The results in Fig. 5b show that when the number of qubits increases, the difference between QUBO and HUBO allocations becomes more significant. We can also read that the HUBO solutions stay closer to the uniform allocation, which means that this formulation produces more diverse portfolios compared to the mean-variance problem. We consider portfolio diversification to be a positive feature.

VI. CONCLUSION

Despite being a complex problem with a strong connection to real-life data, portfolio optimization has a central role in quantum optimization. In this article, we introduced a higher-order portfolio optimization problem and showed how it can be formulated for quantum computers and solved with QAOA. We demonstrated its performance against the classical baselines and showed that the HUBO formulation can encode better portfolios than the classical solutions. This work also concretely demonstrated differences between mean-variance and higher-order portfolio optimization and briefly compared the most common classical optimizers. While the resource requirements for solving HUBO problems with QAOA are likely too demanding for the current hardware, HUBO problems show characteristics that make them a promising class of problems to be solved on quantum computers, especially because the classical solvers do not natively support them.

Future research will include improving QAOA, optimization strategies, and classical subroutines to work more efficiently for HUBO problems. We are especially interested in exploring tensor networks and Bayesian optimizers.

VII. ACKNOWLEDGEMENT

This work is funded by Business Finland (grant number 169/31/2024), Research Council of Finland (grant number 362729) and the Quantum Doctoral Programme.

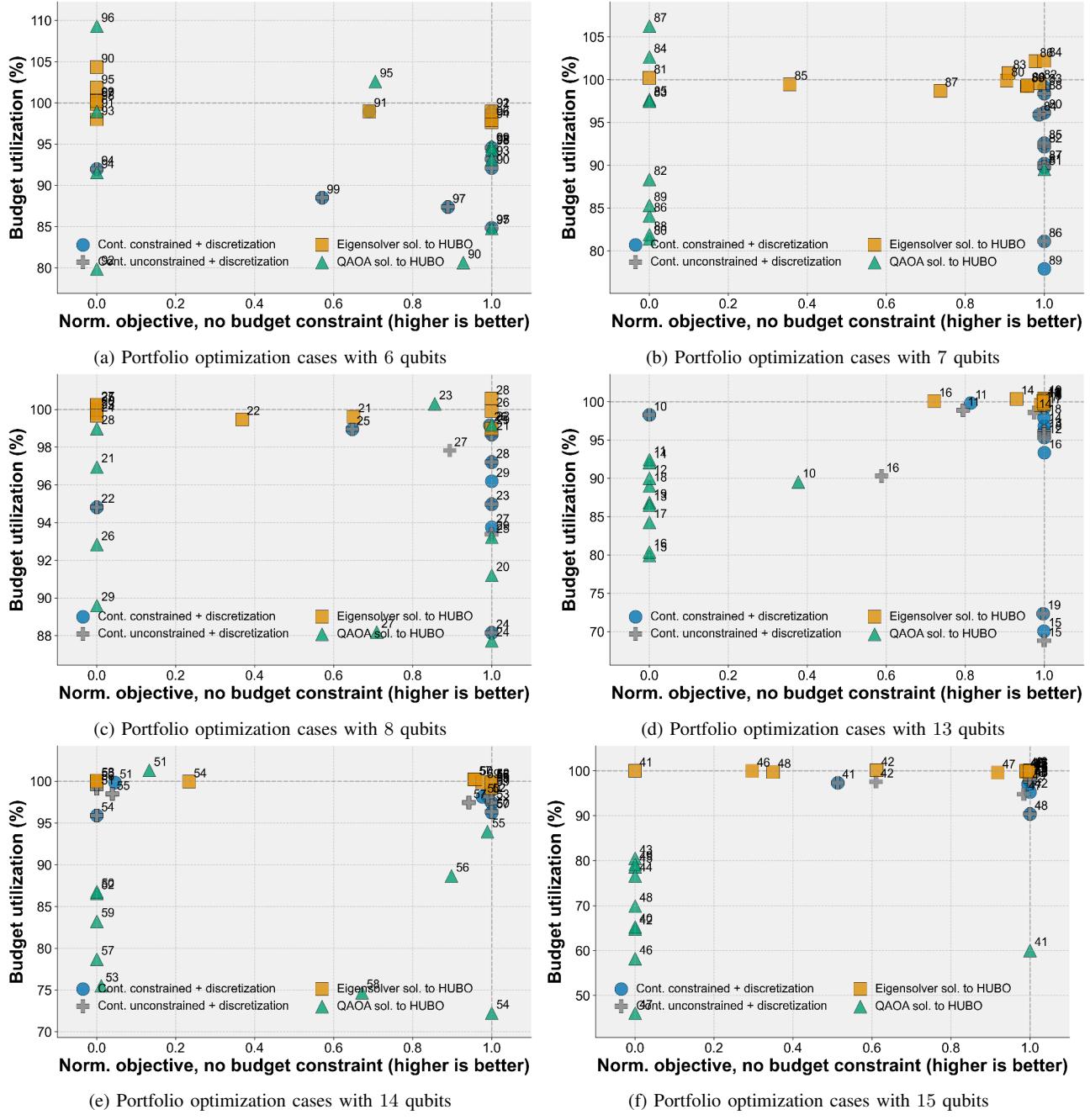


Fig. 6: Portfolio optimization cases with various qubit counts

REFERENCES

- [1] G. Buonaiuto, F. Gargiulo, G. De Pietro, M. Esposito, and M. Pota, "Best practices for portfolio optimization by quantum computing, experimented on real quantum devices," *Scientific Reports*, vol. 13, no. 1, p. 19434, Nov. 2023.
- [2] S. Brandhofer, D. Braun, V. Dehn, G. Hellstern, M. Hüls, Y. Ji, I. Polian, A. S. Bhatia, and T. Wellens, "Benchmarking the performance of portfolio optimization with qaoa," *Quantum Information Processing*, vol. 22, no. 1, p. 25, Dec. 2022, arXiv:2207.10555 [quant-ph].
- [3] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, "A survey of quantum computing for finance," *arXiv preprint arXiv:2201.02773*, jun 2022. [Online]. Available: <https://arxiv.org/abs/2201.02773>
- [4] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," *Reviews in Physics*, vol. 4, p. 100028, Nov. 2019.
- [5] S. Galluccio, J.-P. Bouchaud, and M. Potters, "Rational decisions, random matrices and spin glasses," *Physica A: Statistical Mechanics and its Applications*, vol. 259, no. 3, pp. 449–456, 1998. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S03784371980032X>
- [6] D. Venturelli and A. Kondratyev, "Reverse quantum annealing approach to portfolio optimization problems," *Quantum Machine Intelligence*, vol. 1, no. 1–2, p. 17–30, May 2019, arXiv:1810.08584 [quant-ph].
- [7] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. d. Prado, "Solving the optimal trading trajectory problem using a quantum annealer," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 6, p. 1053–1060, Sep. 2016, arXiv:1508.06182 [q-fin].
- [8] P. Rebentrost and S. Lloyd, "Quantum computational finance: Quantum algorithm for portfolio optimization," *KI - Künstliche Intelligenz*, vol. 38, no. 4, p. 327–338, Dec. 2024.
- [9] F. Phillipson and H. S. Bhatia, "Portfolio optimisation using the d-wave quantum annealer," in *Computational Science – ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2021, p. 45–59.
- [10] I. Kerenidis, A. Prakash, and D. Szilágyi, "Quantum algorithms for portfolio optimization," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, ser. AFT '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 147–155. [Online]. Available: <https://doi.org/10.1145/3318041.3355465>
- [11] N. N. Hegade, P. Chandarana, K. Paul, X. Chen, F. Albarrán-Arriagada, and E. Solano, "Portfolio optimization with digitized counterdiabatic quantum algorithms," *Physical Review Research*, vol. 4, no. 4, p. 043204, Dec. 2022.
- [12] M. Hodson, B. Ruck, H. Ong, D. Garvin, and S. Dulman, "Portfolio rebalancing experiments using the quantum alternating operator ansatz," *arXiv preprint arXiv:1911.05296*, nov 2019. [Online]. Available: <https://arxiv.org/abs/1911.05296>
- [13] M. Marzec, "Portfolio optimization: Applications in quantum computing," *SSRN Electronic Journal*, 2013. [Online]. Available: <http://www.ssrn.com/abstract=2278729>
- [14] S. Mugel, C. Kuchkovsky, E. Sánchez, S. Fernández-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orús, "Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks," *Physical Review Research*, vol. 4, no. 1, p. 013006, Jan. 2022.
- [15] E. Grant, T. S. Humble, and B. Stump, "Benchmarking quantum annealing controls with portfolio optimization," *Physical Review Applied*, vol. 15, no. 1, p. 014012, Jan. 2021.
- [16] N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang, "Quantum walk-based portfolio optimisation," *Quantum*, vol. 5, p. 513, Jul. 2021.
- [17] J. S. Baker and S. K. Radha, "Wasserstein solution quality and the quantum approximate optimization algorithm: A portfolio optimization case study," *arXiv preprint arXiv:2202.06782*, feb 2022. [Online]. Available: <https://arxiv.org/abs/2202.06782>
- [18] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
- [19] R. A. Martin, "Pyportfolioopt: portfolio optimization in python," *Journal of Open Source Software*, vol. 6, no. 61, p. 3066, 2021. [Online]. Available: <https://doi.org/10.21105/joss.03066>
- [20] A. Thomaser, J. De Nobel, D. Vermetten, F. Ye, T. Bäck, and A. Kononova, "When to be discrete: Analyzing algorithm performance on discretized continuous problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '23. New York, NY, USA: Association for Computing Machinery, Jul. 2023, p. 856–863. [Online]. Available: <https://dl.acm.org/doi/10.1145/3583131.3590410>
- [21] A. Basu, "Complexity of optimizing over the integers," *Math. Program.*, vol. 200, no. 2, p. 739–780, Jul. 2022. [Online]. Available: <https://doi.org/10.1007/s10107-022-01862-z>
- [22] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Transactions on Quantum Engineering*, vol. 1, p. 1–24, 2020.
- [23] J. Cohen, A. Khan, and C. Alexander, "Portfolio optimization of 60 stocks using classical and quantum algorithms," *arXiv preprint arXiv:2008.08669*, aug 2020. [Online]. Available: <https://arxiv.org/abs/2008.08669>
- [24] R. Yalovetzky, P. Minnen, D. Herman, and M. Pistoia, "Solving linear systems on quantum hardware with hybrid hhl+," *Scientific Reports*, vol. 14, no. 1, p. 20610, 2024.
- [25] Q. C. Team, "Portfolio optimization," https://qiskit-community.github.io/qiskit-finance/tutorials/01_portfolio_optimization.html, 2023, accessed: 2025-04-08.
- [26] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 155–225, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0166218X01003419>
- [27] G. Munoz, *Integer programming techniques for polynomial optimization*. Columbia University, 2017.
- [28] M. J. Schuetz, J. K. Brubaker, and H. G. Katzgraber, "Combinatorial optimization with physics-inspired graph neural networks," *Nature Machine Intelligence*, vol. 4, no. 4, pp. 367–377, 2022.
- [29] B.-Y. Wang, X. Cui, Q. Zeng, Y. Zhan, M.-H. Yung, and Y. Shi, "Speedup of high-order unconstrained binary optimization using quantum z 2 lattice gauge theory," *Communications Physics*, vol. 8, no. 1, p. 150, 2025.
- [30] C. Campbell and E. Dahl, "Qaoa of the highest order," in *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2022, pp. 141–146.
- [31] S. V. Romero, A.-M. Visuri, A. G. Cadavid, E. Solano, and N. N. Hegade, "Bias-field digitized counterdiabatic quantum algorithm for higher-order binary optimization," *arXiv preprint arXiv:2409.04477*, sep 2024. [Online]. Available: <https://arxiv.org/abs/2409.04477>
- [32] A. G. Cadavid, A. Dalal, A. Simen, E. Solano, and N. N. Hegade, "Bias-field digitized counterdiabatic quantum optimization," *Phys. Rev. Res.*, vol. 7, p. L022010, Apr 2025. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevResearch.7.L022010>
- [33] Z. Verchere, S. Elloumi, and A. Simonetto, "Optimizing Variational Circuits for Higher-Order Binary Optimization," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Los Alamitos, CA, USA: IEEE Computer Society, Sep. 2023, pp. 19–25. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/QCE57702.2023.00011>
- [34] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Quantum annealing vs. qaoa: 127 qubit higher-order ising problems on nisq computers," in *High Performance Computing: 38th International Conference, ISC High Performance 2023, Hamburg, Germany, May 21–25, 2023, Proceedings*. Berlin, Heidelberg: Springer-Verlag, May 2023, p. 240–258. [Online]. Available: https://doi.org/10.1007/978-3-031-32041-5_13
- [35] N. Sachdeva, G. S. Hartnett, S. Maity, S. Marsh, Y. Wang, A. Winick, R. Dougherty, D. Canuto, Y. Q. Chong, M. Hush, P. S. Mundada, C. D. B. Bentley, M. J. Biercuk, and Y. Baum, "Quantum optimization using a 127-qubit gate-model ibm quantum computer can outperform quantum annealers for nontrivial binary optimization problems," *arXiv preprint arXiv:2406.01743*, oct 2024. [Online]. Available: <https://arxiv.org/abs/2406.01743>
- [36] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Short-depth qaoa circuits and quantum annealing on higher-order ising models," *npj Quantum Information*, vol. 10, no. 1, p. 1–19, Mar. 2024.
- [37] V. Gilbert, J. Rodriguez, S. Louise, and R. Sirdey, "Solving higher order binary optimization problems on nisq devices: Experiments and limitations," in *Lecture Notes in Computer Science*, ser. Computational Science – ICCS 2023, J. Mikyška, C. d. Matalier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., vol. LNCS-10477. Prague, Czech Republic: Springer Nature Switzerland, Jul. 2023, p. 224–232. [Online]. Available: <https://hal.science/hal-04394545>

- [38] E. Pelofske, A. Bärtschi, L. Cincio, J. Golden, and S. Eidenbenz, “Scaling whole-chip qaoa for higher-order ising spin glass models on heavy-hex graphs,” *npj Quantum Information*, vol. 10, no. 1, p. 1–18, Nov. 2024.
- [39] V. Uotila, “Left-deep join order selection with higher-order unconstrained binary optimization on quantum computers,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.00362>
- [40] ———, “Tensor decompositions and adiabatic quantum computing for discovering practical matrix multiplication algorithms,” in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 01, 2024, pp. 390–401.
- [41] K. Domino, A. Kundu, Ö. Salehi, and K. Krawiec, “Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing,” *Quantum Information Processing*, vol. 21, no. 9, p. 337, Sep. 2022.
- [42] V. Uotila, “Quantum portfolio optimization,” <https://github.com/valterUo/quantum-portfolio>, 2025, accessed: 2025-03-28.
- [43] H. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, p. 77–91, 1952.
- [44] R. Bhandari and S. R. Das, “Options on portfolios with higher-order moments,” *Finance Research Letters*, vol. 6, no. 3, p. 122–129, Sep. 2009.
- [45] E. Lezmi, H. Malongo, T. Roncalli, and R. Sobotka, “Portfolio allocation with skewness risk: A practical guide,” *SSRN Electronic Journal*, 2018. [Online]. Available: <https://www.ssrn.com/abstract=3201319>
- [46] C. R. Harvey, J. Liechty, M. W. Liechty, and P. Mueller, “Portfolio selection with higher moments,” *SSRN Electronic Journal*, 2004. [Online]. Available: <http://www.ssrn.com/abstract=634141>
- [47] R. G. Balieiro Filho and R. Rosenfeld, “Testing option pricing with the edgeworth expansion,” *Physica A: Statistical Mechanics and its Applications*, vol. 344, no. 3, p. 484–490, Dec. 2004.
- [48] A. Lucas, “Ising formulations of many np problems,” *Frontiers in Physics*, vol. 2, 2014. [Online]. Available: <https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2014.00005>
- [49] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th ed. Cambridge; New York: Cambridge University Press, 2010.
- [50] Portfolio+, “pytickersymbols: A python library for ticker symbol data,” 2020, accessed: 2025-03-11. [Online]. Available: <https://github.com/portfolioplus/pytickersymbols>
- [51] R. Aroussi, “yfinance: Yahoo! finance market data downloader,” 2019, accessed: 2025-03-11. [Online]. Available: <https://yfinance-python.org/index.html>
- [52] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [53] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, “A rewriting system for convex optimization problems,” *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.
- [54] N. Hansen, “The cma evolution strategy: A tutorial,” *arXiv preprint arXiv:1604.00772*, mar 2023. [Online]. Available: <https://arxiv.org/abs/1604.00772>
- [55] N. Hansen, Y. Akimoto, and P. Baudis, “CMA-ES/pycma on Github,” Zenodo, DOI:10.5281/zenodo.2559634, Feb. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.2559634>
- [56] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.