

Best practices for portfolio optimization by quantum computing, experimented on real quantum devices

Giuseppe Buonaiuto

National Research Council

Francesco Gargiulo

National Research Council

Giuseppe De Pietro

National Research Council

Massimo Esposito

National Research Council

Marco Pota (✉ marco.pota@icar.cnr.it)

National Research Council

Article

Keywords:

Posted Date: February 24th, 2023

DOI: <https://doi.org/10.21203/rs.3.rs-2601946/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on November 8th, 2023.
See the published version at <https://doi.org/10.1038/s41598-023-45392-w>.

Best practices for portfolio optimization by quantum computing, experimented on real quantum devices

Giuseppe Buonaiuto¹, Francesco Gargiulo¹, Giuseppe De Pietro¹, Massimo Esposito¹, and Marco Pota^{1,*}

¹National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Naples, 80131, Italy

*marco.pota@icar.cnr.it

ABSTRACT

In finance, portfolio optimization aims at finding optimal investments maximizing a trade-off between return and risks, given some constraints. Classical formulations of this quadratic optimization problem have exact or heuristic solutions, but the complexity scales up as the market dimension increases. Recently, researchers are evaluating the possibility of facing the complexity scaling issue by employing quantum computing. In this paper, the problem is solved using the Variational Quantum Eigensolver (VQE), which in principle is very efficient. The main outcome of this work consists of the definition of the best hyperparameters to set, in order to perform Portfolio Optimization by VQE on real quantum computers. In particular, a quite general formulation of the constrained quadratic problem is considered, which is translated into Quadratic Unconstrained Binary Optimization by the binary encoding of variables and by including constraints in the objective function. This is converted into a set of quantum operators (Ising Hamiltonian), whose minimum eigenvalue is found by VQE and corresponds to the optimal solution. In this work, different hyperparameters of the procedure are analyzed, including different ansatzes and optimization methods by means of experiments on both simulators and real quantum computers. Experiments show that there is a strong dependence of solutions quality on the sufficiently sized quantum computer and correct hyperparameters, and with the best choices, the quantum algorithm run on real quantum devices reaches solutions very close to the exact one, with a strong convergence rate towards the classical solution, even without error-mitigation techniques. Moreover, results obtained on different real quantum devices, for a small-sized example, show the relation between the quality of the solution and the dimension of the quantum processor. Evidences allow concluding which are the best ways to solve real Portfolio Optimization problems by VQE on quantum devices, and confirm the possibility to solve them with higher efficiency, with respect to existing methods, as soon as the size of quantum hardware will be sufficiently high.

Introduction

Portfolio Optimization (PO) is a fundamental financial task, with interesting applications in different scenarios, such as investment funds, pension schemes, and so on. Given a budget and/or a set of assets, it aims at finding optimal trades, within a market that can comprise a very high number of assets.

As formulated by Markowitz¹, it can be expressed as a constrained quadratic optimization problem, where the objective function weighs different objectives including the maximal return and minimal risk (which constitute the quadratic term), subject to budget and/or other constraints. The optimization problem aims at finding optimal values of investments, which may be expressed as continuous variables (in terms of the fraction of budget to invest), but due to their discrete nature, are better represented as integers (in terms of the number of assets units to buy or sell) or binary variables (obtained by the binary encoding of integers). In practice, heuristic methods are currently employed, such as the branch-and-bound method^{2,3}, Particle Swarms, Genetic Algorithms, and Simulated Annealing⁴⁻⁷. These have some limitations but allow to obtain approximate solutions. In this work, the branch-and-bound method^{3,8} is used as a classical benchmark to test the quality of the proposed approach.

However, classical methods may present complexity issues. If the general integer/binary version is regarded, combinatorial optimization is needed, which should find the right one among several tentative solutions that grow exponentially with the market dimension.

To overcome these issues, this work considers that quadratic optimization problems are expected to be solved efficiently and with high accuracy on near-future quantum computers. A detailed discussion about the complexity of different approaches is reported in the Methods section to motivate this alternative approach. In short, the possibilities offered by quantum effects might, in principle, promote quantum computers as valid trade-off solvers of NP-complete problems, giving improved performances in terms of approximation quality and computational time. Indeed, rather than looking for a provable global optimum, which may

be unfeasible with hardware and performance limitations, quantum algorithms can find near-optimal solutions in acceptable computational time. This possibility is acquiring an interest in a growing number of fields, particularly relevant to optimization problems.

Optimization by Quantum Computing (QC) includes, on the one hand, quantum annealers⁹, designed to solve specific optimization problems by mapping them onto physical quantum Hamiltonian and have shown promising results when benchmarked with classically available algorithms. On the other hand, the real computational capabilities of gate-based quantum and hybrid algorithms are still to be explored. In particular, PO has been approached in a few works by quantum annealers^{10,11}. At the same time, the solution of a simplified version of the problem by Variational Quantum Eigensolver (VQE) is available on the IBM Qiskit platform¹².

Until nowadays, the size of quantum devices is limited, and the computation is not entirely fault-tolerant, i.e., an efficient quantum error correction is not available yet. But very recently, the availability of Noisy Intermediate Scale Quantum (NISQ) devices allows testing the performances of both quantum and hybrid algorithms to explore new computational paradigms that find various applications in fields such as chemistry, biology^{13–15}, and artificial intelligence¹⁶. However, a detailed study of the performances on PO of various NISQ devices is still missing.

In this work, PO is approached by QC, specifically making use of the VQE algorithm to find an optimal or sub-optimal solution to the problem. Although the VQE methodology has been extensively used in various scenarios, a complete experimental analysis of this technique, including the encoding strategy, on a significant subset of the actual state-of-the-art quantum machines is still missing. With respect to the state-of-the-art approaches for PO with QC, VQE is chosen instead of quantum annealers for its generality, and the existing decision problem formulation¹² is generalized into the optimization problem. Moreover, an experimental investigation is performed using real IBM quantum computers. In particular, this work presents solutions to the problem obtained with different hyperparameters settings to find the best practices to perform PO by VQE on real quantum devices.

More in detail, a sample of a limited size of real financial data is employed since, albeit the scale of the system considered does not match realistic requirements, it allows exploring the efficiency of the QC approach. These data are used to construct the objective as a trade-off between the expected return and variance, weighted by a risk aversion coefficient. Then, the general constrained integer quadratic formulation is transformed into a Quadratic Unconstrained Binary Optimization (QUBO) problem through the binary encoding of variables, designed to reduce the required number of qubits, and by a penalty coefficient, which weights constraints satisfaction with respect to objectives, to include constraints into the objective function¹¹. This is converted into a set of quantum operators (Ising Hamiltonian), whose minimum eigenvalue corresponds to the optimal solution. An approximation of this solution is found by VQE, a hybrid algorithm involving the choices of a parametric tentative quantum state (ansatz), and a classical optimization algorithm.

Therefore, this paper aims to find the best hyperparameters settings to perform PO by VQE, i.e., appropriate ansatz and optimizer are found so that the effect of noise is minimized. At the same time, the convergence rate is maximized, and a suitable penalty coefficient is found based on its effect on the convergence toward the correct solution. Finally, the optimal solutions are compared among those obtained on simulators and on real quantum computers of different sizes and architectures and with the benchmark solution.

The paper is structured as follows: firstly, the materials and methods are presented, including the dataset description, the formulation of the PO problem, its translation into a QUBO problem, then into a quantum Hamiltonian, the generalities of the VQE method, and details of the IBM NISQ devices. In the following section, results are shown and discussed. Finally, conclusions and future perspectives are outlined.

Materials and methods

Dataset

The data are collected from Yahoo!@finance¹⁷ using *yfinance*¹⁸, an open-source tool that uses Yahoo's publicly available APIs. This tool, according to its creator, is intended for research and educational purposes.

To explore the efficiency of the proposed approach, small-sized examples are considered by extracting at most $N = 4$ different assets: Apple, IBM, Netflix and Tesla. These are representative global assets with interesting dynamics influenced by financial and social events. For each asset i , with $1 \leq i \leq N$, the temporal range between 2011/12/23 and 2022/10/21 is considered. For each day t in this range ($0 \leq t \leq T$), the performance of an asset is well represented by its closing price p_i^t . A sub-interval of dates considered is shown in Table 1.

The first information extracted from this data set consists in the list P of current prices P_i of the considered assets.

$$P_i = p_i^T. \quad (1)$$

	AAPL	IBM	NFLX	TSLA
Date				
2016-12-23	27.219765	119.262428	125.589996	14.222667
2016-12-27	27.392632	119.570061	128.350006	14.635333
2016-12-28	27.275822	118.890434	125.889999	14.649333
2016-12-29	27.268820	119.183701	125.330002	14.312000
2016-12-30	27.056236	118.747368	123.800003	14.246000
2017-01-03	27.133329	119.605820	127.489998	14.466000
2017-01-04	27.102961	121.086693	129.410004	15.132667
2017-01-05	27.240784	120.686058	131.809998	15.116667
2017-01-06	27.544472	121.279861	131.070007	15.267333
2017-01-09	27.796768	119.934898	130.949997	15.418667
2017-01-10	27.824797	118.411110	129.889999	15.324667
2017-01-11	27.974308	120.006439	130.500000	15.315333

Table 1. Closing prices of four assets, *Apple*, *IBM*, *Netflix* and *Tesla*, for a sub-interval of the whole time period, extracted from Yahoo!@finance using *yfinance* Python package, and considered for experiments in this work.

Moreover, for each asset, the return r_i^t between the days $t - 1$ and t can be calculated:

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

These returns, calculated for days when the initial and the end prices are known, cannot be used for inference. Instead, it is convenient to define the expected return of an asset as an educated guess of its future performance. Assuming a normal distribution of the returns, the average of their values at each time t on the set of historical observations is a good estimator of the expected return. Therefore, given the entire historical data set, the expected return of each asset μ_i is calculated by:

$$\mu_i = E[r_i] = \frac{1}{T} \sum_{t=1}^T r_i^t. \quad (3)$$

Following the same principle, the variance of each asset return and the covariance between returns of different assets over the historical series can be calculated as follows:

$$\begin{aligned} \sigma_i^2 &= E[(r_i - \mu_i)^2] = \frac{1}{T-1} \sum_{t=1}^T (r_i^t - \mu_i)^2, \\ \sigma_{ij} &= E[(r_i - \mu_i)(r_j - \mu_j)] = \frac{1}{T-1} \sum_{t=1}^T ((r_i^t - \mu_i)(r_j^t - \mu_j)). \end{aligned} \quad (4)$$

Portfolio Optimization

The traditional theory of PO was initially formulated by Markowitz¹. There are multiple possible formulations of PO, all embodying different degrees of approximation of the real-life problem. This work deals with Multi-Objective Portfolio optimization: this approach tries to simultaneously maximize the return and minimize the risk while investing the available budget. Even if other formulations include more objectives, the aim is still the solution of a constrained quadratic optimization problem; therefore, the formulation considered here is general enough to test the performances of the proposed approach.

A portfolio is defined as the set of investments x_i (measured as a fraction of the budget or number of asset units) allocated for each i th asset of the market. Therefore, the portfolio consists of a vector of real or integer numbers with dimensions equal to the number of assets considered. An optimal strategy for portfolio allocations aims to achieve the maximum portfolio return $\mu^T x$ while minimizing risk, defined as the portfolio variance $x^T \Sigma x$ (whose square root is the portfolio volatility), where μ is the vector of mean asset returns for each asset i calculated by (3), Σ is the covariance matrix calculated by (4), and x is the vector of investments measured as fractions of budget. Hence, the task of finding the optimal portfolio aims at finding the x vector that maximizes the following objective function:

$$\mathcal{L}(x) : \mu^T x - q x^T \Sigma x, \quad (5)$$

where the risk aversion parameter q expresses the propensity to risk of the investor (a trade-off weight between the risk and the return).

In a realistic scenario, the available budget B is fixed. Therefore, the constraint that the sum of x_i equals 1 must hold. Moreover, if only buying is allowed, each $x_i \geq 0$, this constraint does not hold if either buying or selling is possible. As a consequence, in the general case, the problem can be stated as follows:

$$\begin{aligned} \max_x \mathcal{L}(x) : \max_x (\mu^T x - q x^T \Sigma x), \\ \text{s.t.} \quad \sum_{i=1}^N x_i = 1 \end{aligned} \quad (6)$$

However, if x is a possible solution to the problem with continuous variables, each product $x_i B$ must be an integer multiple of the corresponding price P_i calculated by (1) since an integer number of units of each asset can be exchanged. Therefore, only a subset of the possible solutions corresponding to integer units is acceptable, and the problem is better stated as follows:

$$\begin{aligned} \max_n \mathcal{L}(n) : \max_n (\mu'^T n - q n^T \Sigma' n), \\ \text{s.t.} \quad P'^T n = 1 \end{aligned} \quad (7)$$

where n is the vector of n_i integer units of each asset, while $P' = P/B$, $\mu' = P' \circ \mu$ and $\Sigma' = (P' \circ \Sigma)^T \circ P'$ are appropriate transformations of μ and Σ . The latter formulation (7) is an integer constrained quadratic optimization problem.

Possible solutions to the problem (6) are those satisfying the constraint. Among them, some correspond to possible solutions to problem (7). The collection of possible solutions corresponding to portfolios with maximum return for any risk is called "Markowitz efficient frontier". The solution of the constrained quadratic optimization problem lies on the efficient frontier, and the distance from minimum risk depends on q .

Complexity

The general problem, if regarded in terms of continuous variables, can be solved exactly by Lagrange multipliers in case of equality constraints, or by Karush–Kuhn–Tucker conditions, which generalize the method of Lagrange multipliers to include inequality constraint¹⁹, as the covariance matrix is positive semi-definite²⁰. Optimizing a quadratic function subject to linear constraints leads to a linear system of equations, solvable by Cholesky decomposition²¹ of the symmetrical covariance matrix. The exact solution involves the computation of the inverse of an $N \times N$ matrix, where N is the number of assets, thus requiring about $O(N^3)$ floating-point operations²².

As long as integer or binary variables are considered, the problem turns into combinatorial optimization. The computational complexity is known to be high since the optimization problem is NP-hard^{23,24}, while the decision version is NP-complete²⁵. Indeed, a search approach should find the optimal one among possible solutions whose number increases exponentially with the number of assets (e.g., for b binary variables, 2^b possible solutions, while for N integer variables ranging from 0 to n_{max} , $(n_{max} + 1)^N$ possible solutions).

In practice, various methods are currently employed, either based on geometric assumptions, such as the branch-and-bound method^{2,3}, or rather heuristic algorithms⁴⁻⁷, such as Particle Swarms, Genetic Algorithms, and Simulated Annealing. These have some limitations but allow to obtain approximate solutions. However, in all cases, the exact or approximate solution is feasible only for a few tens of assets on current classical computers.

Using quantum mechanical effects, like interference and entanglement, quantum computers can perform computational operations within the Bounded-error Quantum Polynomial (BQP) class of complexity, outperforming classical computers virtually on any P problem on some NP problems. Problems existing in BQP could be classified as harder than NP-Complete problems, and many practical BQP problems are suspected to exist outside of P. This highlights the potential power of QC in comparison with classical computing²⁶.

Classical solution

The branch-and-bound method^{3,8} is used in this work as a classical benchmark to compare the results of the proposed approach. It is based on the Lagrangian dual relaxation and continuous relaxation for discrete multi-factor portfolio selection model, which leads to an integer quadratic programming problem. The separable structure of the model is investigated by using Lagrangian relaxation and dual search. This algorithm is capable of solving portfolio problems with up to 120 assets.

Specifically, the library CPLEX freely available on Python provides a robust implementation of the aforementioned classical solving scheme.

Quantum formulation

As formulated in eq. (7), the PO problem lies within the class of quadratic optimization problems. To be quantum-native, it has to be converted into a Quadratic Unbounded Binary Optimization (QUBO) problem, i.e., the target vector to be found has to be expressed as a vector of zeros and ones, and constraints have to be avoided.

Therefore, the binary conversion matrix C is constructed with a number of binarizing elements d_i for each asset i depending on the price P_i . Hence

$$n_i^{max} = \text{Int} \left(\frac{B}{P_i} \right), \quad (8)$$

and

$$d_i = \text{Int} (\log_2 n_i^{max}), \quad (9)$$

such that

$$n_i = \sum_{j=0}^{d_i} 2^j b_{i,j}. \quad (10)$$

In this way, the overall dimension of the binarized target vector, $b = [b_{1,0}, \dots, b_{1,d_1}, \dots, b_{N,0}, \dots, b_{N,d_N}]$, is $\dim(b) = \sum_{i=1}^N (d_i + 1)$, which is lower than that used in implementation available in Qiskit¹². Conveniently, the encoding matrix C is defined as follows:

$$C = \begin{pmatrix} 2^0 & \dots & 2^{d_1} & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 2^0 & \dots & 2^{d_2} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 2^0 & \dots & 2^{d_N} \end{pmatrix}, \quad (11)$$

and thus, the conversion can be written in short notation as $n = Cb$. It is possible to redefine the problem (7), in terms of the binary vector b , applying the encoding matrix by $\mu'' = C^T \mu'$, $\Sigma'' = C^T \Sigma' C$ and $P'' = C^T P'$:

$$\begin{aligned} \max_b \mathcal{L}(b) : & \max_b (\mu''^T b - q b^T \Sigma'' b), \\ \text{s.t.} \quad & P''^T b = 1 \\ & b_i \in \{0, 1\} \quad \forall i \in [1, \dots, \dim(b)]. \end{aligned} \quad (12)$$

The problem (12) falls into the wide set of binary quadratic optimization problems, with a constraint, given by the total budget. In this form, the problem cannot be cast directly into a suitable set of quantum operators that run on quantum hardware: the constraint, in particular, is troublesome, as it poses a hard limitation on the sector of Hilbert space that needs to be explored by the algorithm, to find a solution. It is thus necessary to convert the problem into a QUBO (Quadratic Unconstrained Binary Optimization) by transforming the constraint into a penalty term in the objective function. Each kind of constraint can be converted into a specific penalty term²⁷, and the one considered in (12), which is equality, linear in the target variable, maps into $\lambda (P''^T b - 1)^2$, such that (12) can be written in terms of the following QUBO problem:

$$\max_b \mathcal{L}(b) : \max_b (\mu''^T b - q b^T \Sigma'' b - \lambda (P''^T b - 1)^2). \quad (13)$$

The penalty coefficient λ is a key hyperparameter to state the problem as the QUBO of the objective function (13).

There is a strong connection, technically an isomorphism, between the QUBO and the Ising Hamiltonian²⁸: Ising Hamiltonian was originally constructed to understand the microscopic behavior of magnetic materials, particularly to grasp the condition that leads to a phase transition. However, its relative simplicity and natural mapping into QUBO have made the Ising model a fundamental benchmark well beyond the field of quantum physics. To convert (13) into an Ising, it is convenient to expand it in its components:

$$\mathcal{L}(b) : \sum_i \mu_i'' b_i - q \sum_{i,j} \Sigma_{i,j}'' b_i b_j - \lambda (\sum_i P_i'' b_i - 1)^2, \quad (14)$$

where $\mu_i'', \Sigma_{i,j}'', P_i''$, are the components of the transformed return, covariance, and price, respectively, and $i, j \in [1, \dim(b)]$. Since the Ising represents spin variables s_i , which have values $\{-1, 1\}$, the transformation $b_i \rightarrow \frac{1+s_i}{2}$ is applied and coefficients are re-arranged, to obtain the Ising objective function to minimize:

$$\begin{aligned} \min_s \mathcal{L}(s) : \min_s & \left(\sum_i h_i s_i + \sum_{i,j} J_{i,j} s_i s_j + \lambda (\sum_i \pi_i s_i - \beta)^2 \right), \\ \text{s.t. } & s_{i,j} \in \{-1, 1\} \quad \forall i, \end{aligned} \quad (15)$$

with $J_{i,j}$ being the coupling term between two spin variables. It is now straightforward to obtain the corresponding quantum Hamiltonian, whose eigenvector corresponding to the minimum eigenvalue corresponds to the solution: in fact, the eigenvalues of the Pauli operators Z are ± 1 . Thus they are suitable for describing the classical spin variables s_i . Furthermore, the two-body interaction term can be modeled with the tensor product between two Pauli operators, i.e., $Z_i \otimes Z_j$. The quantum Ising Hamiltonian reads:

$$H = \sum_i h_i Z_i + \sum_{i,j} J_{i,j} Z_i \otimes Z_j + \lambda (\sum_i \pi_i Z_i - \beta)^2. \quad (16)$$

With the procedure described above, the integer quadratic optimization problem of a portfolio allocation with budget constraints is expressed first as a binary problem via the binary encoding, then it is translated into a QUBO, transforming the constraints into a penalty term by the chosen penalty coefficient, and finally into a quantum Hamiltonian written in term of Pauli gates. Hence, the PO problem (7) is now formulated as the search of the ground state, i.e., the minimum energy eigenstate, of the Hamiltonian (16). Therefore, it is possible to use the VQE, employing real quantum hardware, and iteratively approximate such a state, as described in the following section, which corresponds to the optimal portfolio.

Variational Quantum Eigensolver

The VQE is a hybrid quantum-classical algorithm²⁹, which is based on the variational principle: it consists in the estimation of the upper bound of the lowest possible eigenvalue of a given observable with respect to a parameterized wave-function (ansatz). Specifically, given a Hamiltonian H representing the observable, and a parameterized wave-function $|\psi(\theta)\rangle$, the ground state E_0 is the minimum energy eigenstate associated with the Hamiltonian:

$$E_0 \leq \frac{\langle \psi(\theta) | H | \psi(\theta) \rangle}{\langle \psi(\theta) | \psi(\theta) \rangle}, \quad \forall \theta. \quad (17)$$

Hence, the task of the VQE is finding the optimal set of parameters, such that the energy associated with the state is nearly indistinguishable from its ground state, i.e., finding the set of parameters θ , corresponding to energy E_{min} , for which $|E_{min} - E_0| < \varepsilon$, being ε an arbitrarily small constant. This problem can be formulated on a quantum computer as a series of parameterized quantum gates, which are applied on the initial state to realize a structured ansatz for the Hamiltonian problem. Conventionally, the initial state is set to be the vacuum state, i.e., for Q qubit system $|0\rangle^{\otimes Q} = |\mathbf{0}\rangle$, where \otimes stands for the tensor product between each state describing the single qubit system. Thus, on a quantum device, the problem of maximizing the objective function (17) can be expressed as:

$$E_{min} = \min_{\theta} \langle \mathbf{0} | U^\dagger(\theta) H U(\theta) | \mathbf{0} \rangle. \quad (18)$$

where $U(\theta)$ is the parametrized unitary operator that gives the ansatz wave-function when applied on the initial state, E_{min} is the energy associated with the parametrized ansatz. The Hamiltonian H , defined for the specific problem, and in this case corresponding to (16), can be written in a specific operator basis that makes it naturally measurable on a quantum computer: this choice depends on the architecture considered. In this work, given the extensive use of the IBM quantum experience³⁰, it is convenient to map the Hamiltonian into spin operators' base. This base is formed by the tensor product of Pauli strings: $P_l \in \{I, X, Y, Z\}^{\otimes N}$. In this base the Hamiltonian can always be written in the general form, $H = \sum_l^D c_l P_l$, where D is the number of Pauli strings that define the Hamiltonian and c_l is a suitable set of weights. It follows that the VQE in eq. (18) can be written as:

$$E_{min} = \min_{\theta} \sum_l^D c_l \langle \mathbf{0} | U^\dagger(\theta) P_l U(\theta) | \mathbf{0} \rangle. \quad (19)$$

Each term in eq. (19) corresponds to the expectation value of the string P_l and is computed on quantum hardware (or a simulator). The summation and the optimization of the parameters are computed on a classical computer, choosing an ad-hoc

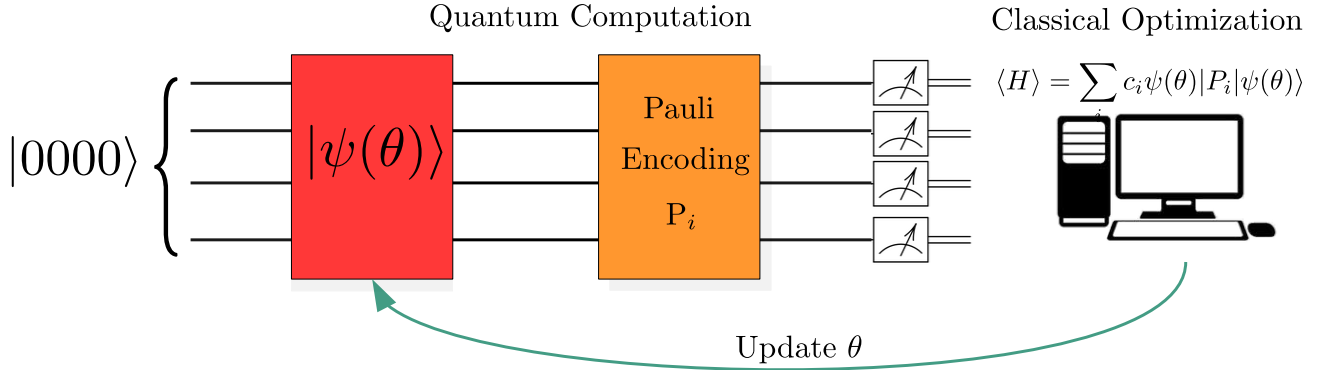


Figure 1. Schematic of the VQE algorithm. The ansatz wave-function $|\psi(\theta)\rangle$ is initialized with random parameters and encoded in a given set of quantum gates. The PO problem is translated into an Ising Hamiltonian and encoded into a set of Pauli gates. The collection of output measurement allows the reconstruction of the expectation value of the Hamiltonian H , which is the energy that needs to be minimized. A classical optimization algorithm provides an update rule for the parameters of the wave-function, which ideally moves iteratively towards the ground state of the problem, thus providing an estimation of the corresponding eigenstate. This corresponds to the solution of the original PO problem.

optimizer. The eigenvector corresponding to the ground state corresponds to the solution of the problem(13), thus to the optimal portfolio.

In light of what is stated above, the complete VQE estimation process can be decomposed in a series of steps, as depicted in Fig. 1. First, it is necessary to prepare a trial wave-function (ansatz) on which the expectation value needs to be evaluated and realized via a parameterized quantum circuit. Then, it is necessary to define the Hamiltonian (16), whose ground state is the solution to the problem to be addressed, and convert it into the Pauli basis so that the observable can be measured on the quantum computer. Finally, the parameters are trained using a classical optimizer. This hybrid system ideally converges to a form that produces a state compatible with the ground state of the Hamiltonian.

This procedure includes two hyperparameters that have to be settled, i.e., the type of ansatz and the optimizer. When defining the ansatz, two main features have to be taken into account: its expressibility, i.e., the set of states that can be spanned by the ansatz itself, and the trainability, i.e., the ability of the ansatz to be optimized efficiently with available techniques. It is worth pointing out the problem of the barren plateau³¹, related to the possibility of vanishing gradients when the cost function gradients converge to zero exponentially, as a function of the specific characteristic of the problem to be solved. The barren plateau depends on the number of qubits, the high expressivity of the ansatz wave-function, the degree of entanglement, and the quantum noise³². There are several methods to avoid or mitigate the effect of the barren plateau, especially in the context of VQE, most of which consist in finding a trade-off between the expressivity of the ansatz and its trainability and reducing the effective size of the Hilbert space of the problem formulation³³.

The following ansatzes are available in Qiskit and are analyzed in this work: *Two Local* ansatz, where qubits are coupled in pairs, the *Real Amplitude* ansatz, which assumes real-valued amplitude for each base element of the wave-function, and the *Pauli Two* ansatz, used mainly in quantum machine learning for the mitigation of barren plateau³⁴. Although other ansatzes are provided in Qiskit, they are generally unsuitable for a PO problem. For instance, the *Excitation preserving* ansatz preserves the ratio between basis vector components, hence does not allow, in principle, any weight imbalance in the output distribution while moving towards the solution of the problem.

For all the ansatzes considered, the convergence of four different possible assumptions on the entanglement structure of the wave-function is checked, namely the *full* entanglement, the *linear* entanglement, the *circular* and the *pairwise* entanglement. The former modifies the ansatz such that any qubit is entangled with all the others pairwise. In the linear case, the entanglement is built between consecutive pairs of qubits. The circular case is equivalent to the linear entanglement but with an additional entanglement layer connecting the first and the last qubit before the linear sector. Finally, in the pairwise entanglement construction, in one layer, the i th qubit is entangled with qubit $i + 1$ for all even i , and in a second layer, qubit i is entangled with qubit $i + 1$, for odd values of i .

Once the ansatz is defined, its parameters must be optimized classically until convergence is reached. The choice of the optimizer is crucial because it impacts the number of measurements that are necessary to complete the optimization cycle since, when properly chosen, it can mitigate the barren plateau problem and minimize the number of iterations required to reach

convergence. In this work, dealing with the PO problem, different optimizers are tested to select which one fulfills its task faster, among those available on Qiskit, i.e., *Cobyla*, *SPSA*, and *NFT*³⁵.

NISQ devices

The experimental results presented in this work are obtained on real quantum hardware, specifically using the platforms provided by IBM superconducting quantum computers. These quantum machines belong to the class of NISQ devices, which stands for Noisy Intermediate Scale Quantum devices, i.e., a class of hardware with a limited number of qubits and where noise is not suppressed. Noise, in quantum computers, comes from various sources: decoherence, gate fidelities, and measurement calibration. Decoherence is the process that most quantum mechanical systems undergo when interacting with an external environment³⁶. It causes the loss of virtually all the quantum properties of the qubits, which then collapse into classical bits. Gate fidelities measure the ability to implement the desired quantum gates physically: in the IBM superconducting qubits hardware, these are constructed via pulses, which are shaped and designed to control the superconductors. Given the limited ability to strictly control these pulses, a perfect gate implementation is highly non-trivial and subjected to imperfections. Last, measurement errors are caused by the limits of the measurement apparatus, improper calibration, and imperfect readout techniques. Hence, NISQ devices do not always provide reliable results due to the lack of fault tolerance. However, they provide a good benchmark for testing the possibilities of quantum computing. Furthermore, ongoing research is on the possibility of using NISQ in practical applications, such as machine learning and optimization problems.

In this work, both simulators and real quantum computers are used. Even though error mitigation techniques³⁷ can be applied, the main goal of this paper is to test the performances of the quantum computers on a QUBO problem, such as PO, without error mitigation, with the binary encoding strategies and the budget constraints as described in the previous sections. Therefore, in all computations, there is no error mitigation, aiming to build an indirect but comprehensive analysis of the hardware limitations and to improve the quality of the results offered by a proper selection of the hyperparameters. This will provide a solid benchmark for the following experimental stages, which will be enabled in the coming years by large and nearly fault-tolerant quantum computers.

Hence, the experiments run on simulators (without noise) are also executed by adding noise mimicking real hardware: this operation can be readily implemented on Qiskit by inserting a noise model containing the decoherence parameters and the gate error rate from real quantum hardware.

Moreover, experiments are run on IBM NISQ devices with up to 25 qubits. Specifically, a substantial subset of the available quantum computers in the IBM quantum experience was employed: IBM Guadalupe, Toronto, Geneva, Cairo, Auckland, Montreal, Mumbai, Kolkata, and Hanoi. These machines have either 16 or 27 qubits, but they have different quantum volumes (QV) and Circuit Layer Operations Per Second (CLOPS). QV and CLOPS are useful metrics to define the performances of a quantum computation pipeline³⁸. Generally, a bigger QV means that the hardware can sustain deeper circuits with a relatively small price on the performance. At the same time, the CLOPS quantifies the number of operations that can be handled by the hardware per unit of time. Hence, altogether, they qualify the quality and speed of quantum computation.

Results and discussion

In this section, PO results obtained with different hyperparameters and on different simulated and real quantum devices are presented and discussed in terms of the algorithm's convergence and the quality of the optimal solution found.

Experimental settings

The experiments involve all the assets described in the Dataset section. Prices are calculated from data by eq. (1), mean returns by (3), and covariance matrix by (4). The total expendable budget B is set to be commensurate with the number of assets considered, i.e., $B = 2000$ for all experiments. All the experiments are executed by considering the risk aversion parameter $q = 0.5$, representing a mid-way compromise between the risk and the return.

The proposed approach is implemented on the Qiskit³⁹ software development kit. A set of 12 qubits was required to encode the spin variables of eq. (15), to encode the binarized number of investments. Results with varying hyperparameters are obtained by quantum simulators, with and without simulated noise. The best set of hyperparameters is used in the experiments on different real quantum computers.

Study of hyperparameters

The proposed approach involves the setting of methods and constants related to the optimization algorithm and the QUBO formalization of the problem. In this section, results obtained with different settings of hyperparameters are presented to find the best choices for PO.

Firstly, the hybrid VQE algorithm entails the choice of a type of ansatz to initialize the qubit wave-function, and the choice of a classical optimizer to tune the ansatz parameters toward the solution, as detailed in the VQE Section. In Fig. 2 and 3,

the convergence of experiments performed by using different ansatzes and optimizers are reported. Both figures show the convergence rate, during epochs, towards the minimum energy E_{min} , which approximates the ground energy of the Hamiltonian and corresponds to the quality of the solution of the PO problem.

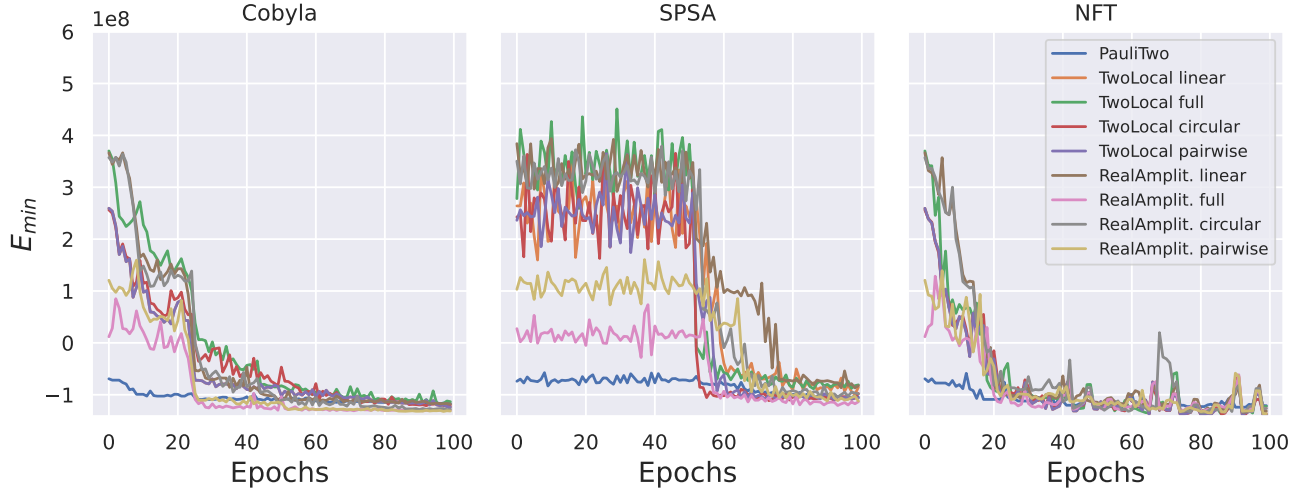


Figure 2. Noiseless experiments performed on IBM QASM simulator, supposing a fault-tolerant quantum machine, with no quantum noise influencing the quality of the results. Convergence of the solutions towards the optimal one during training epochs, evaluated with different optimizers and different ansatzes.



Figure 3. Noisy experiments, performed on IBM QASM simulator, by importing IBM Cairo quantum computer noise model. Convergence of the solutions towards the optimal one during training epochs, evaluated with different optimizers and ansatzes.

In particular, Fig. 2 reports results obtained on a noiseless quantum simulator, while Fig. 3 shows more noisy experiments on a simulated noisy quantum computer. Specifically, the first set of experiments is performed with the QASM quantum simulator provided by IBM⁴⁰, while the second set is done on the same simulator by importing the noise model from the specifications of the IBM Cairo quantum computer. In both cases, the experiments are performed over a set of nine ansatzes and three possible optimizers, all provided by Qiskit³⁵, as detailed in the Methods section.

Regarding the comparisons among classical optimizers, both figures allow some considerations. First, both the Cobyla and the NFT optimizers foster a rapid convergence towards low values of the energy for every ansatz, while SPSA presents a delayed behavior. On the other hand, the NFT⁴¹ optimizer, contrarily to the others, experiences relatively unstable behavior, with highly oscillating trajectories for each ansatz. Moreover, all optimizers are relatively robust against statistical noise,

but a more oscillating behavior is obtained in noisy simulations using SPSA or NFT optimizers. This preliminary analysis suggests that the Cobyla optimizer is the most stable and more adequate than others to reach the optimal solution in a reasonable computational time. This is particularly true in the noiseless simulation, in which case the good quality of the solutions reached by the Cobyla optimizer is not dependent on the ansatz. However, for quite all optimizers, and also for Cobyla optimizer in the presence of errors, the quality of the final solution depends on the ansatz.

As far as the ansatzes are compared, they present both different convergence rates and different final E_{min} reached. Moreover, some show delayed convergent behavior, thus suggesting that their inherent structure affects the training process. In particular, the PauliTwo ansatz soon reaches a good quality solution. After PauliTwo, the fastest ansatzes that reach the same solution quality, in the noiseless case, are RealAmplitude with either full or pairwise entanglement. The linear entanglement construction is less powerful as a parametrization, but it converges fast toward the optimal solution. Instead, circular entanglement on both TwoLocal and RealAmplitude structures is associated with a lower convergence rate. On the other hand, the full entanglement ansatzes converge in most cases to higher final energies, i.e., worse solutions, with noticeable effects in noisy simulations. With regard to the Cobyla optimizer, all the ansatzes converge to similar solutions in the noiseless case. In the presence of noise, good solutions are obtained soon with PauliTwo ansatz, and the best final solutions are obtained by both TwoLocal and RealAmplitude ansatzes with both linear and pairwise entanglement. Therefore, PauliTwo ansatz should be chosen to obtain a solution after very few epochs, while one of the latter ones could be preferred if a slightly longer computational time is acceptable.

Comparison of Figg. 2 and 3 clearly show the effect of noise and errors in the computation. Some ansatzes allow to approach the same minimum energy as in the noiseless case but require more epochs to converge. For other ansatzes, the solution converges to values appreciably different between the noisy and noiseless cases. Compared to the noiseless situation, the optimizers drift from the full convergence due to the effect of noise. However, for all ansatzes, target values are obtained after a few epochs by the Cobyla optimizer, even in the noisy case. Therefore, the convergence rate is influenced by errors. Still, stable solutions are found by the Cobyla optimizer immediately with the PauliTwo ansatz, designed to avoid plautau during training and after a few epochs with the others. This comparison thus reveals that the effect of quantum and measurement noise is quite relevant: noise hinders the convergence of the VQE in almost every situation by either slowing down the convergence rate or shifting up the value of the minimum energy, i.e., the quality of the solution found. However, the effect of the error is neglectable if the Cobyla optimizer and the appropriate ansatz are chosen.

Secondly, the effect of the parameter λ , used as penalty coefficient to weight the constraint satisfaction with respect to the objectives, to transform the constrained (12) into unconstrained quadratic problem (13), is investigated. Fig. 4 reports the results obtained.

In detail, Fig. 4 reports the expected return vs. volatility of different portfolios. Dots represent the random sampling of possible solutions satisfying the constraint of the continuous problem (6). The set of these points evidences the Markowitz efficient frontier, which is the set of solutions with maximal return for each volatility, where the optimal solution should lie. Among them, a few are possible solutions to the integer PO problem (7). The square corresponds to the optimal solution found by the classical branch-and-bound method. The other symbols correspond to the optimal solutions to the integer (7) or binary problem (12), with constraints embedded in the objective function (13) by setting values of the penalty coefficient λ of different orders of magnitude. The optimal solutions are obtained after 250 epochs on the simulated IBM Cairo machine using the Cobyla optimizer and the TwoLocal linear entanglement ansatz.

From Fig. 4, it can be noticed that all optimal solutions lie close to the Markovitz frontier. Moreover, while the solution corresponding to $\lambda = 0$ does not necessarily respect the budget constraints, as long as moderated λ values are used, the optimal solutions overlap with the classical result. Instead, over this optimal region, as λ is still increased by powers of 10, sub-optimal solutions are obtained since the penalty term in (16) becomes dominant respect to the coupling terms and thus hinders the mapping onto the original problem. The quadratic term proportional to lambda, if not properly balanced, modifies the spectral properties of the Hamiltonian, shifting the energies and the eigenstate of the unperturbed problem (i.e., the Hamiltonian of the unconstrained problem). For these experiments, the interval $1 \leq \lambda \leq 10$ guarantees the convergence of the solution to the classical findings. In general, these results allow individuating the optimal value of λ within the same order of magnitude of the fraction between the objectives and the constraint satisfaction quadratic term.

These results confirm that the correct choice of the penalty coefficient λ is very important, and for the PO problem, the workable values are found in these experiments. In general, constraints can be divided into *hard* and *soft*²⁷. A hard constraint must be satisfied, then λ must be large enough to preclude violations. Instead, in this case, a soft constraint can be used, according to PO practical applications, so slight violations can be tolerated, and a moderate penalty value is sufficient. Experiments show that a too-large penalty value can negatively influence the solution process since the penalty terms overwhelm the original objective function information, which introduces difficulties in distinguishing the quality of different solutions. On the other hand, a too-small penalty value offers solutions not adequately in accord with the budget. The *Goldilocks region*²⁷, the interval of values that work for the PO problem, is found above.

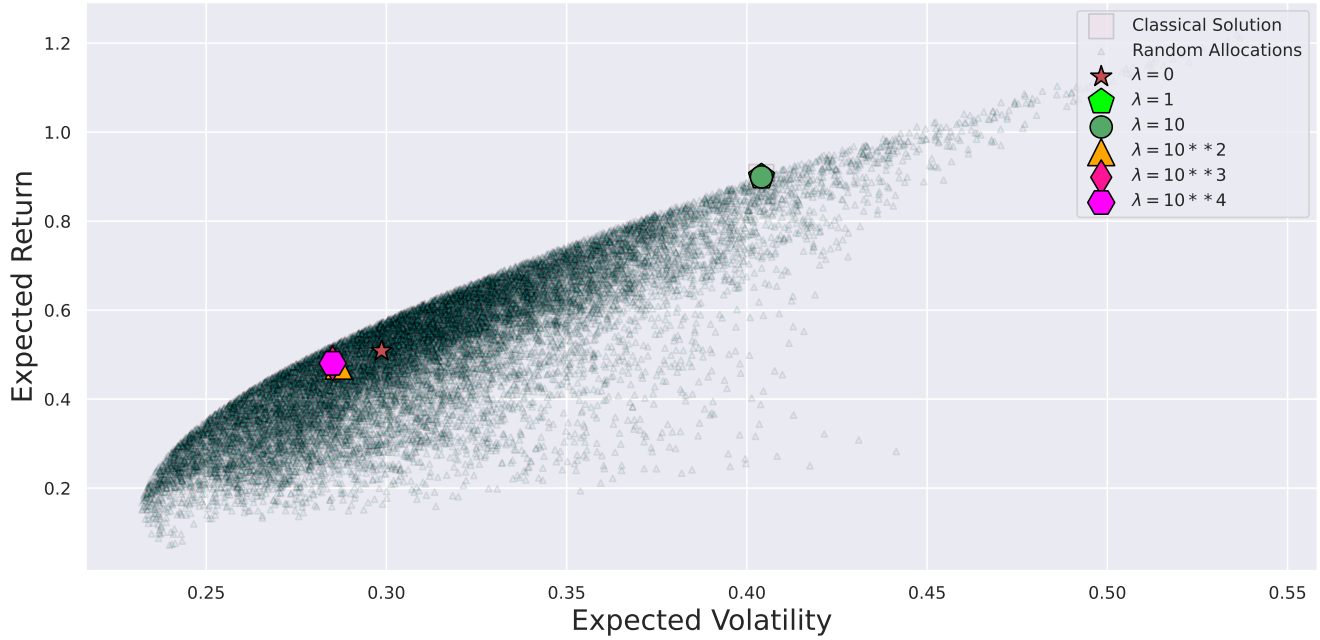


Figure 4. Effect of variable penalty coefficient, used to transform the constrained into an unconstrained problem. Dots represent the random sampling of possible solutions satisfying the constraint of the continuous problem. Among them, a few are also possible solutions to the integer PO problem. The square corresponds to the optimal solution found by the classical branch-and-bound method. The other symbols correspond to the optimal solutions found to the QUBO problem, with constraints embedded in the objective function by setting different values of the penalty coefficient.

Experiments on real quantum computers

In this section, experiments are run on real NISQ devices, detailed in the Methods Section. A fixed number of 200 epochs was chosen. The hyperparameters present the best behavior in the simulated runs, i.e., the QUBO model is obtained by $\lambda = 10$, and the VQE algorithm employs the Coyula classical optimizer and the TwoLocal linear entanglement ansatz.

The experimental results are shown in Fig. (5). In particular, the figure represents the return and volatility of the solution. The dots represent the random sampling of possible solutions to the continuous problem. Among them, a few are also possible solutions to the integer PO problem. The classical solution of the integer problem, which lies on the Markowitz efficient frontier, is indicated by a square. The other symbols indicate optimal solutions found by employing different IBM quantum computers.

From Fig. 5, it can be noticed that the solution found by some real quantum computers is perfectly matching with the classical solution. In particular, among those detailed in the Methods section and tested here, the following are those with an optimal solution: Toronto, Kolkata, and Auckland.

A more detailed discussion can be done on the basis of Fig. 6, which presents the fraction between the minimum energy found by the classical method and by VQE run on real devices. In particular, optimal results should approach 1, and the figure shows the statistics of the results over repetitions of the simulations on each real device, performed with different ansatzes. In particular, the devices are shown in order of growing quantum volume.

From Fig. 6, it can be noticed that there is an increasing trend both in the mean and in the median, as the QV of the quantum computer grows.

These results show that both the mapping of the ansatz structure on the hardware topology and the quantum volume is of pivotal importance for reaching the desired convergence. The topology of a quantum computer refers to the physical arrangement of qubits: while ansatzes connecting only the nearest qubits can be mapped efficiently, those entailing long-range connections require an overhead of gates that ultimately increases the depth of the circuit and hence foster an increase of the overall error rate during computation. On the contrary, densely structured ansatzes, like the TwoLocal full entanglement, provide a robust and potentially more expressive benchmark to explore the parameter space, and thus to find the global minimum of the objective function. In this sense, a balance needs to be found between the expressivity of the ansatz and the mapping on the hardware topology. Ultimately, a higher QV allows to perform computation on a deeper circuit without an exponential increase of the error rate: hence, as 6 suggests, higher quantum volumes, as for *ibm kolkata*, allows to run efficiently largely parametrized ansatzes, which converge better to the global optimum of the problem.

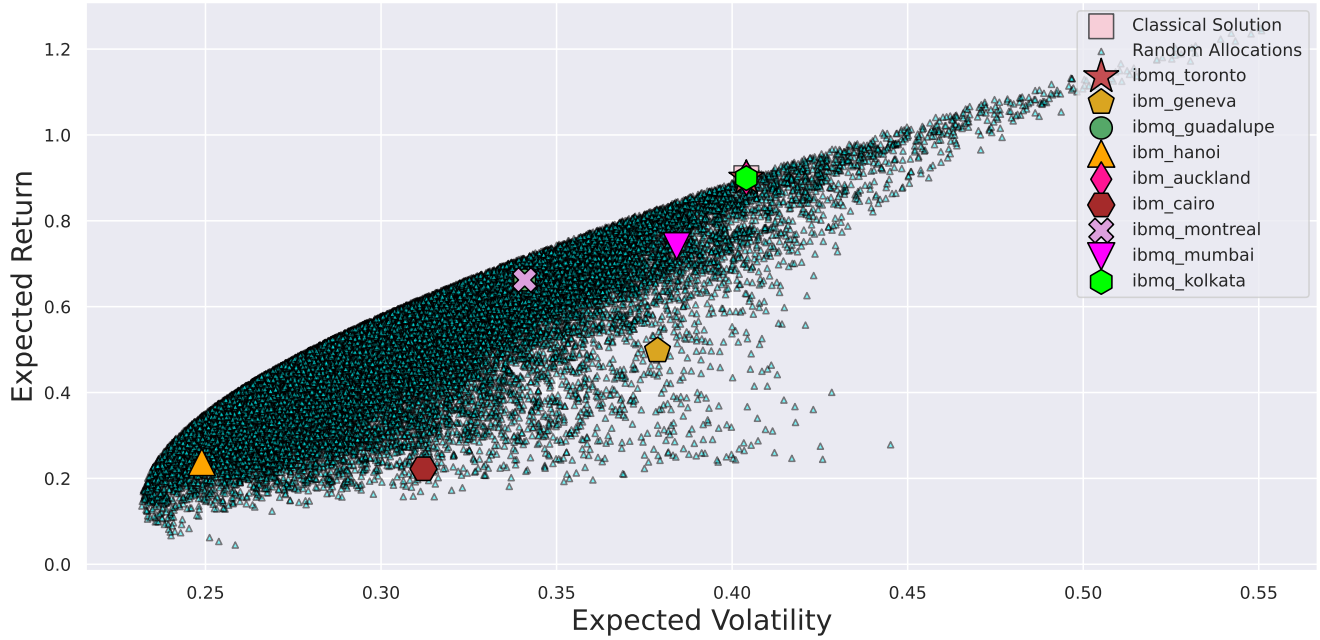


Figure 5. Results of experiments run on different real quantum devices. Dots represent the random sampling of possible solutions satisfying the constraint of the continuous problem. The square corresponds to the optimal solution found by the classical branch-and-bound method. The other symbols correspond to the optimal solutions found to the QUBO problem, by means of different IBM quantum computers.

Conclusions and future perspectives

In this paper, the Portfolio Optimization problem was approached by Quantum Computing, in particular by translating the quite general quadratic problem formulation into a Quadratic Unbounded Binary Optimization, mapped to a Hamiltonian, whose minimum eigenvalue is approximated by the Variational Quantum Eigensolver, and corresponds to the optimal portfolio.

In particular, different hyperparameters of this approach are analyzed, i.e., the penalty coefficient that enables the transformation of the problem from constrained to unconstrained, the type of parametric wave-function (ansatz), and the optimizer employed in VQE. Moreover, experiments were run on both simulators and on different real quantum computers.

The importance of selecting a proper ansatz and optimizer for the VQE and a proper penalty coefficient was revealed. Moreover, the best choices were individuated, in order to solve the most efficient PO by VQE, even in presence of quantum hardware noise. Furthermore, the relation between the quality of the solutions found by VQE and the characteristics of the quantum computers was found to be dependent on the intrinsic properties of quantum processors. Even though this is well known in the literature, here it has been proven and validated experimentally.

Finally, solutions found were bench-marked with the classical solution. Albeit the scale of the system considered is not matching with realistic requirements, the solutions of the VQE on NISQ devices reveal promising features, both in terms of complexity and the solution quality.

Future perspectives consist in solving real-life portfolio optimization problems, with higher market size, as soon as quantum devices with appropriate characteristics will be available. Moreover, future work includes the formalization of the problem in the most advanced ways available. Finally, the effect of the topology of the hardware on quantum variational algorithms will be matter of a follow-up investigation.

Data availability

All data analysed during this study are included in this published article and its supplementary information files.

References

1. Markowitz, H. Portfolio selection. *The J. Finance* **7**, 77–91 (1952).

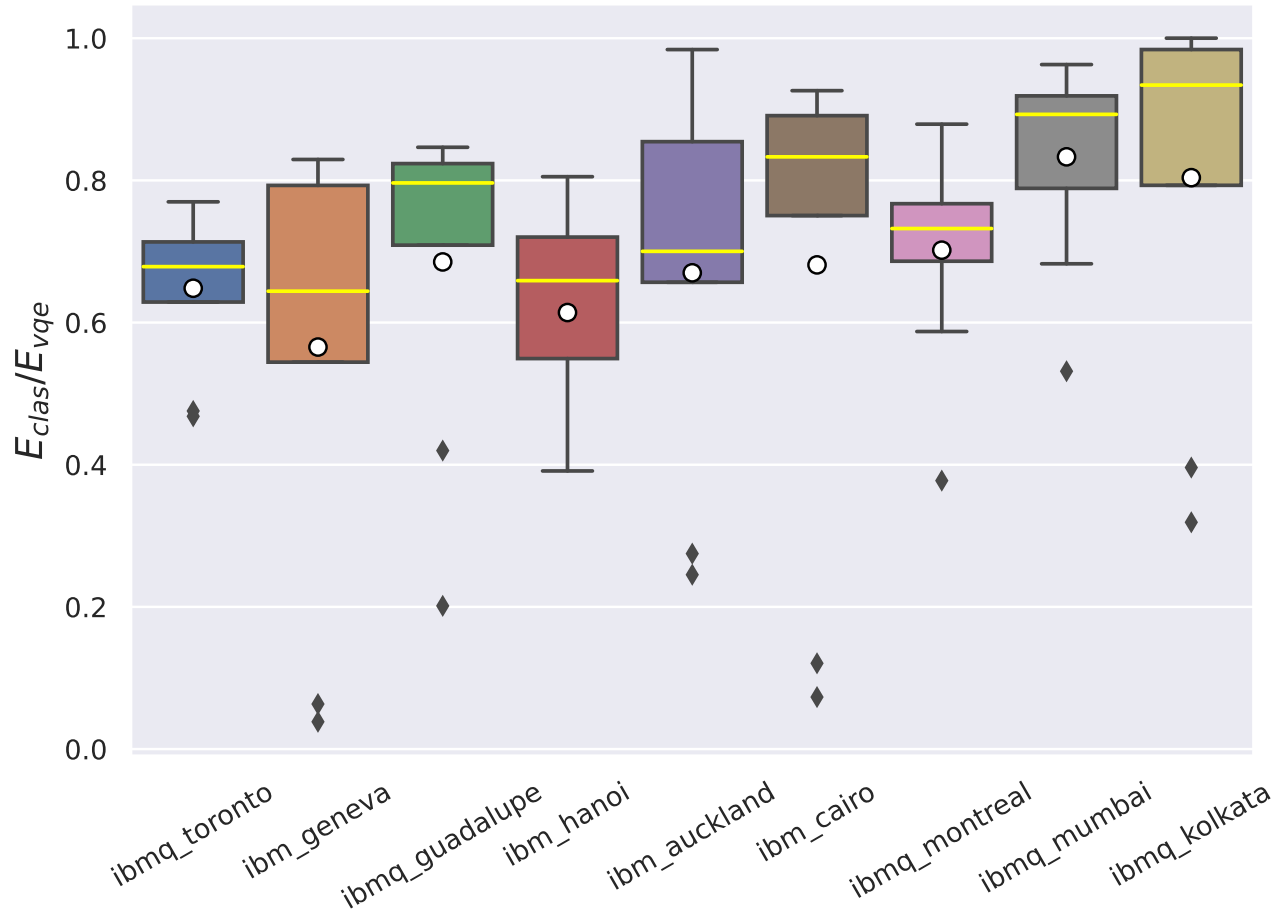


Figure 6. Results of experiments run with various ansatz, on different real quantum devices, ordered according to growing quantum volumes. The box extends from the quartile Q_1 to Q_3 of the data, with a yellow line at the median (Q_2). The black lines extend from the edges of the box to show the range of the data. Here a standard approach is followed, as they extend to a maximum of $1.5 * (Q_3 - Q_1)$ from the edges of the box, ending at the farthest data point in the interval. Data outliers are plotted as black squares, while the white dots represent the mean of the result distribution for each hardware.

- Marinescu, R. & Dechter, R. And/or branch-and-bound search for pure 0/1 integer linear programming problems. In Beck, J. C. & Smith, B. M. (eds.) *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 152–166 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006).
- Niu, S.-f., Wang, G.-x. & Sun, X.-l. A branch-and-bound algorithm for discrete multi-factor portfolio optimization model. *J. Shanghai Univ.* **12**, 26—30, DOI: [10.1007/s11741-008-0105-3](https://doi.org/10.1007/s11741-008-0105-3) (2008).
- Pinelis, M. & Ruppert, D. Machine learning portfolio allocation. *The J. Finance Data Sci.* **8**, 35–54, DOI: <https://doi.org/10.1016/j.jfds.2021.12.001> (2022).
- Zhang, Q. & Gao, Y. An asset allocation model and its solving method. *J. Syst. Sci. Inf.* **5**, 163–175, DOI: [doi: 10.21078/JSSI-2017-163-13](https://doi.org/10.21078/JSSI-2017-163-13) (2017).
- Gunjan, A. & Bhattacharyya, S. A brief review of portfolio optimization techniques. *Artif. Intell. Rev.* DOI: [10.1007/s10462-022-10273-7](https://doi.org/10.1007/s10462-022-10273-7) (2022).
- Zanjirdar, M. Overview of portfolio optimization models. *Adv. Math. Finance Appl.* **5**, 419–435, DOI: [10.22034/amfa.2020.674941](https://doi.org/10.22034/amfa.2020.674941) (2020). https://amfa.arak.iau.ir/article_674941_8e7bd0548442d117c51895837c1c7205.pdf.
- Cplex, I. I. V12. 1: User's manual for cplex. *Int. Bus. Mach. Corp.* **46**, 157 (2009).

9. Abel, S., Blance, A. & Spannowsky, M. Quantum optimization of complex systems with a quantum annealer. *Phys. Rev. A* **106**, 042607, DOI: [10.1103/PhysRevA.106.042607](https://doi.org/10.1103/PhysRevA.106.042607) (2022).
10. Lang, J., Zielinski, S. & Feld, S. Strategic portfolio optimization using simulated, digital, and quantum annealing. *Appl. Sci.* **12** (2022).
11. Phillipson, F. & Bhatia, H. Portfolio optimisation using the d-wave quantum annealer. In Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V., Dongarra, J. & Sloot, P. (eds.) *Computational Science – ICCS 2021*, vol. 12747 (Springer, Cham, 2021).
12. IBM. Portfolio optimization. https://qiskit.org/documentation/stable/0.26/tutorials/finance/01_portfolio_optimization.html.
13. Cheng, H.-P., Deumens, E., Freericks, J. K., Li, C. & Sanders, B. A. Application of quantum computing to biochemical systems: A look to the future. *Front. Chem.* **8**, DOI: [10.3389/fchem.2020.587143](https://doi.org/10.3389/fchem.2020.587143) (2020).
14. Andersson, M. P., Jones, M. N., Mikkelsen, K. V., You, F. & Mansouri, S. S. Quantum computing for chemical and biomolecular product design. *Curr. Opin. Chem. Eng.* **36**, 100754, DOI: <https://doi.org/10.1016/j.coche.2021.100754> (2022).
15. Cao, Y. *et al.* Quantum chemistry in the age of quantum computing. *Chem. Rev.* **119**, 10856–10915, DOI: [10.1021/acs.chemrev.8b00803](https://doi.org/10.1021/acs.chemrev.8b00803) (2019).
16. Peral García, D., Cruz-Benito, J. & José García-Peñalvo, F. Systematic Literature Review: Quantum Machine Learning and its applications. *arXiv e-prints* arXiv:2201.04093 (2022). [2201.04093](https://arxiv.org/abs/2201.04093).
17. Yahoo! Yahoo! finance. <https://finance.yahoo.com/>.
18. Yahoo! yfinance 0.2.10. <https://pypi.org/project/yfinance>.
19. Karush, W. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Thesis, Dept. Math. Univ. Chicago, Chicago, Ill.* (1939).
20. Gould, N. I. M., Hribar, M. E. & Nocedal, J. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM J. on Sci. Comput.* **23**, 1376–1395, DOI: [10.1137/S1064827598345667](https://doi.org/10.1137/S1064827598345667) (2001).
21. Note sur une methode de resolution des equations normales provenant de l’application de la methode des moindres carres a un systeme d’equations lineaires en nombre inferieur a celui des inconnues (procede du commandant cholesky). *Bull. Geod.* **2**, 66–67, DOI: [10.1007/BF03031308](https://doi.org/10.1007/BF03031308) (1924).
22. Intel® oneapi math kernel library developer reference. <https://www.intel.com/content/www/us/en/develop/documentation/onemkl-developer-reference-c/top/lapack-routines/lapack-linear-equation-routines/lapack-linear-equation-computational-routines/matrix-factorization-lapack-computational-routines/potrf.html#potrf>.
23. Cesarone, F., Scozzari, A. & Tardella, F. Linear vs. quadratic portfolio selection models with hard real-world constraints. *Comput. Manag. Sci.* **12**, 345–370, DOI: [10.1007/s10287-014-0210-1](https://doi.org/10.1007/s10287-014-0210-1) (2015).
24. Phillipson, F. & Bhatia, H. S. Portfolio optimisation using the d-wave quantum annealer. In Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V. V., Dongarra, J. J. & Sloot, P. M. A. (eds.) *Computational Science – ICCS 2021*, 45–59 (Springer International Publishing, Cham, 2021).
25. Del Pia, A., Dey, S. S. & Molinaro, M. Mixed-integer Quadratic Programming is in NP. *arXiv e-prints* arXiv:1407.4798 (2014). [1407.4798](https://arxiv.org/abs/1407.4798).
26. Aaronson, S. Bqp and the polynomial hierarchy. In *Proceedings of ACM STOC 2010* (2010).
27. Glover, F., Kochenberger, G. & Du, Y. Quantum bridge analytics i: a tutorial on formulating and using qubo models. *4OR* **17**, 335–371, DOI: [10.1007/s10288-019-00424-y](https://doi.org/10.1007/s10288-019-00424-y) (2019).
28. Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, DOI: [10.3389/fphy.2014.00005](https://doi.org/10.3389/fphy.2014.00005) (2014).
29. Tilly, J. *et al.* The variational quantum eigensolver: A review of methods and best practices. *Phys. Reports* **986**, 1–128, DOI: <https://doi.org/10.1016/j.physrep.2022.08.003> (2022). The Variational Quantum Eigensolver: a review of methods and best practices.
30. IBM. <https://quantum-computing.ibm.com/>.
31. Liu, X., Liu, G., Huang, J., Zhang, H.-K. & Wang, X. Mitigating barren plateaus of variational quantum eigensolvers. *arXiv e-prints* arXiv:2205.13539 (2022). [2205.13539](https://arxiv.org/abs/2205.13539).
32. Wang, S. *et al.* Noise-induced barren plateaus in variational quantum algorithms. *Nat. Commun.* **12**, 6961, DOI: [10.1038/s41467-021-27045-6](https://doi.org/10.1038/s41467-021-27045-6) (2021).

33. Uvarov, A. V. & Biamonte, J. D. On barren plateaus and cost function locality in variational quantum algorithms. *J. Phys. A: Math. Theor.* **54**, 245301, DOI: [10.1088/1751-8121/abfac7](https://doi.org/10.1088/1751-8121/abfac7) (2021).
34. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812, DOI: [10.1038/s41467-018-07090-4](https://doi.org/10.1038/s41467-018-07090-4) (2018).
35. IBM. qiskit.algorithms.optimizers. <https://qiskit.org/documentation/stubs/qiskit.algorithms.optimizers.html>.
36. Brandt, H. E. Qubit devices and the issue of quantum decoherence. *Prog. Quantum Electron.* **22**, 257–370, DOI: [https://doi.org/10.1016/S0079-6727\(99\)00003-8](https://doi.org/10.1016/S0079-6727(99)00003-8) (1999).
37. Kandala, A. *et al.* Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **567**, 491–495, DOI: [10.1038/s41586-019-1040-7](https://doi.org/10.1038/s41586-019-1040-7) (2019).
38. Wack, A. *et al.* Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers. *arXiv e-prints* arXiv:2110.14108 (2021). [2110.14108](https://arxiv.org/abs/2110.14108).
39. IBM. Qiskit. <https://qiskit.org/>.
40. IBM. Simulators overview. <https://quantum-computing.ibm.com/lab/docs/iql/manage/simulator/#qasm>.
41. Nakanishi, K. M., Fujii, K. & Todo, S. Sequential minimal optimization for quantum-classical hybrid algorithms. *Phys. Rev. Res.* **2**, 043158, DOI: [10.1103/PhysRevResearch.2.043158](https://doi.org/10.1103/PhysRevResearch.2.043158) (2020).

Author information

Contributions

M.P., M.E. and G.D.P. conceived the experiments, F.G. and G.B. conducted the experiments, G.B., M.E., M.P. and G.D.P. analysed the results. M.P. and G.B. wrote the original draft, and all authors reviewed the manuscript.

Ethics declarations

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [datafinance.csv](#)