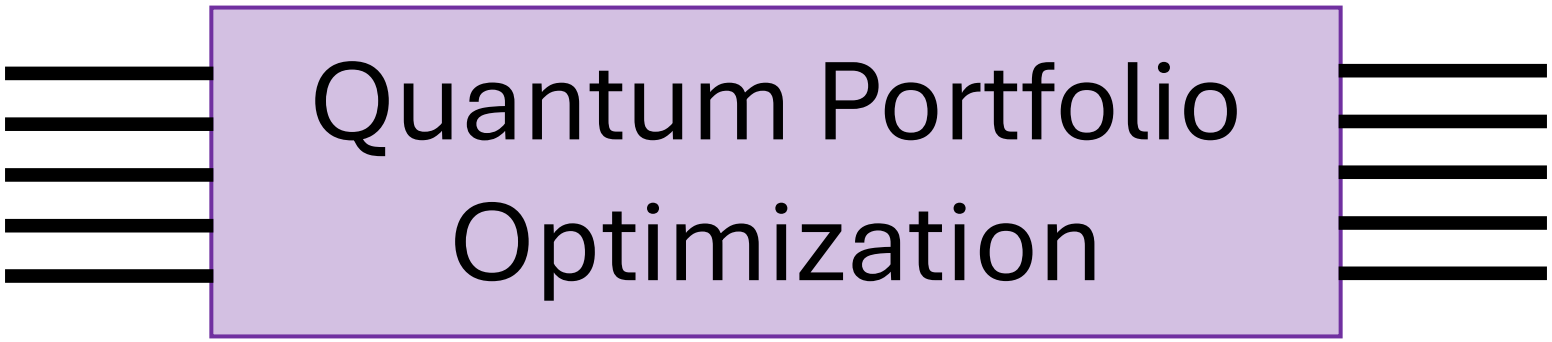
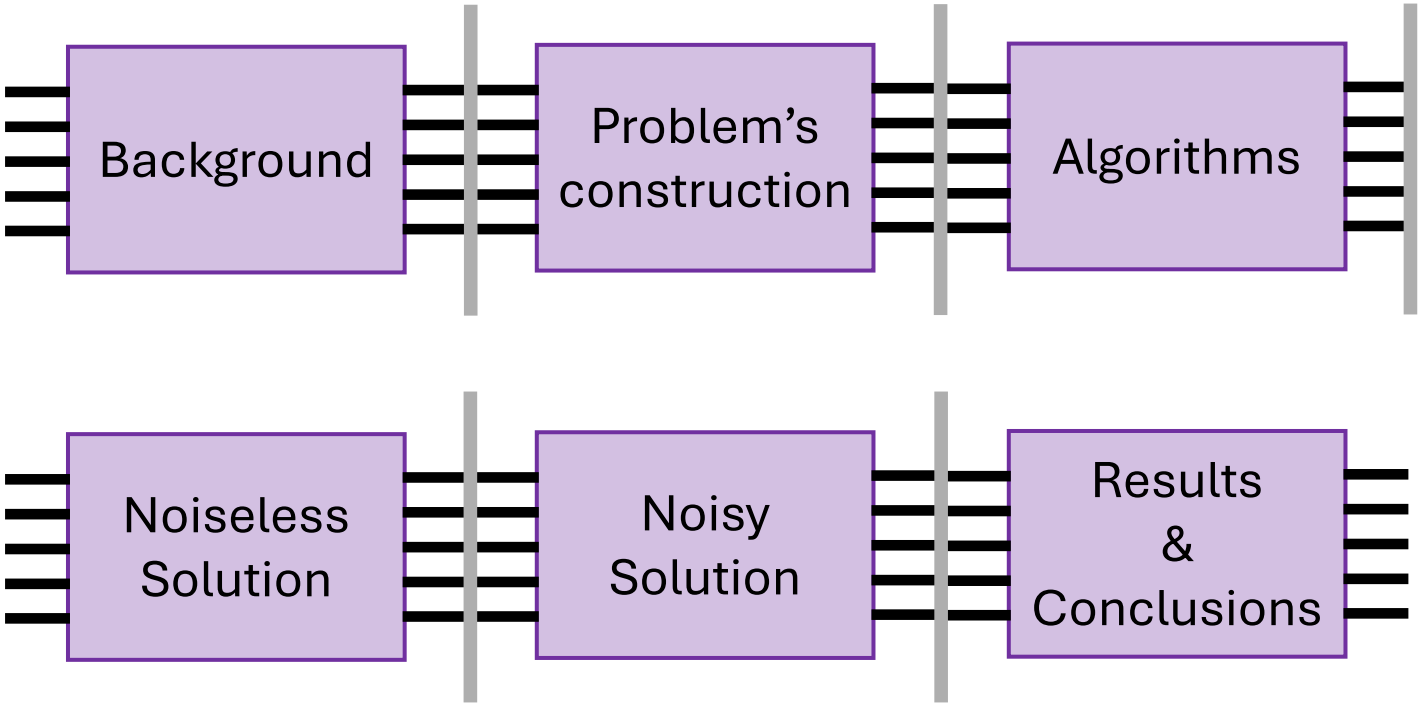

Quantum Portfolio
Optimization

Index

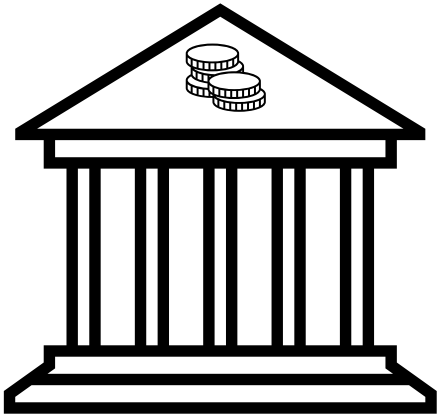

Background
Problem's construction
Algorithms
Noiseless Solution
Noisy Solution
Results & Conclusions

Background: Finance

Finance is a term that addresses matters regarding the management, creation, and study of money and investments.

The finance area includes some **key concepts**:

- **Assets**, which is something of value such as cash.
- **Risk factor**, which is a characteristic that can influence the return profile of an asset class.
- **Budget**, which is a financial plan that estimates future revenues and expenses over a specific period (a month or a year)



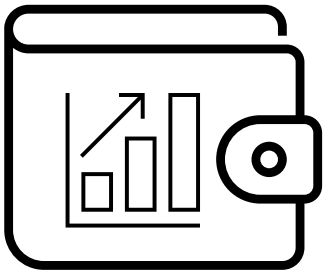
Background: The problem

Portfolio optimisation is a **primary financial activity**, that has many different applications.

Given a **budget and a set of assets**, is necessary to **identify optimal operations** inside the market.

A correct set of operations implies a **correct allocation of the assets**, which has a direct impact on the **profitability of the investments**.

The **classical algorithms available** to compute a correct allocation have **exponential complexity** $O(2^n)$.

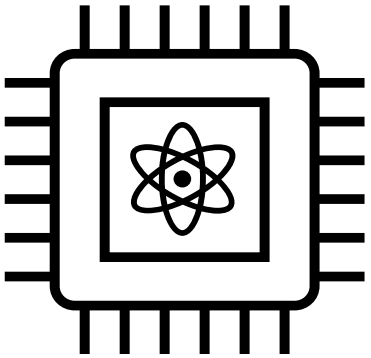


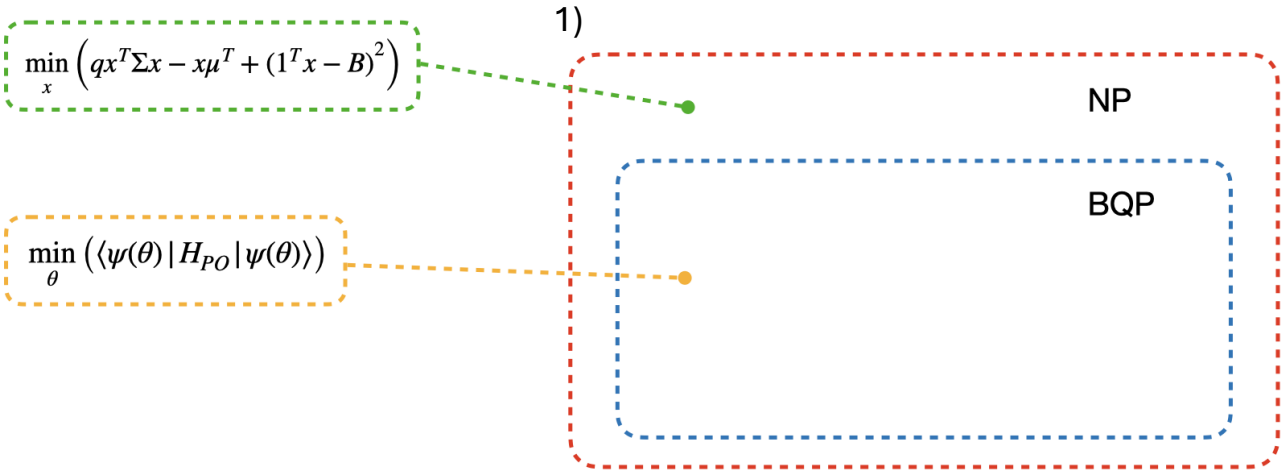
Background: Quantum computing

Problems related to **finance require too much computational power** for classical computing methodologies.

Quantum computing overcomes some of these limits, exploiting principles like **superposition and entanglement**.

Quantum algorithms can deal with NP-hard problems by harnessing the power of the **BQP complexity class** to solve some problems more efficiently. (1)





Problem's construction

The objective is to **choose a set of assets** that **maximise return while minimising risk**.

The **available budget must be respected**.

It can be **written in the form of an equation** (1), where:

- $x \in \{0,1\}^n$ represents the vector of binary decision variables (set of assets to be selected).
- $\mu \in \mathbb{R}^n$ represents the expected returns of assets.
- $\Sigma \in \mathbb{R}^{\{n \times n\}}$ represents the relationship of the change in asset prices (covariance).
- $q > 0$ represents risk aversion (higher numbers imply more risky allocations).
- B represents the available budget.

1)
$$\min_x \left(q x^T \Sigma x - x\mu^T + (1^T x - B)^2 \right)$$

2)
$$\min_x \left(\underbrace{q x^T \Sigma x}_{\text{Risk}} \overbrace{- x\mu^T}^{\text{Return}} + \underbrace{(1^T x - B)^2}_{\text{Budget penalty}} \right)$$

Algorithms: Overview

There are **different algorithms** to solve portfolio optimisation problems.

Three algorithms are divided into **2 groups**:

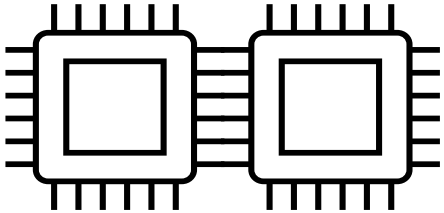
- Classical. (1)
- Hybrid. (2)

Hybrid solutions are pretty interesting because the computation is done by **combining quantum and classical approaches**.

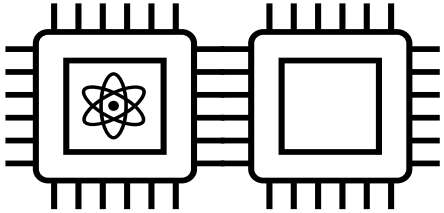
All algorithms are composed of 2 parts:

- Selection of the minimum eigen solver.
- Use a MinimumEigenOptimizer to search for an optimal solution.

1)



2)



Algorithms: VQE

Variational Quantum Eigensolver is a **hybrid algorithm**.

Its principle is based on the variation principle and **aims to find a state where energy is minimal**.

To minimise the energy of the system the **algorithm is parameterised over θ**. (1)

The problem to optimise is described using a Hamiltonian that rapresents a cost function. (2)

And the parameterised wave function. (3)

$$E(\theta) = \frac{\langle \psi(\theta) | H | \psi(\theta) \rangle}{\langle \psi(\theta) | \psi(\theta) \rangle}$$

1) 2) 3)

Algorithms: VQE

Considering a ground state E_0 (1), **VQE search a set of parameters** θ to obtain an energy E_{min} (2) that is enought 'similar' to the ground state (3).

The problem about the related to the computation of E_{min} is rapresented with an ansaz (4).

The ansatz contains a **parametrized unitary operator** that generates the wave function. (5)

1)
$$E_0 \leq \frac{\langle \psi(\theta)|H|\psi(\theta)\rangle}{\langle \psi(\theta)|\psi(\theta)\rangle}$$

2) 5)
$$E_{\min} = \underbrace{\min_{\theta} \langle 0|U^{\dagger}(\theta) H \overbrace{U(\theta)}|0\rangle}_{4)}$$

3)
$$|E_{\min} - E_0| < \epsilon$$

Algorithms: QAOA

Quantum Approximate Optimization Algorithm is a **hybrid algorithm** projected to find an approximated solution to **combinatory optimisation problems**.

The algorithm works by **alternating a sequence of layers**, each composed of 2 main parts:

- **Cost operator**. (1)
- **Mixing operator**. (2)

The cost operator's Hamiltonian encodes the optimisation problem. (3)

The mixing operator's Hamiltonian allows the exploration of the solutions space. (4)

γ and β **variational parameters that are optimised during the execution** of the algorithm.

1) $\hat{U}_C(\gamma) = e^{-i\gamma \overbrace{\hat{H}_C}^{3)}}$

2) $\hat{U}_M(\beta) = e^{-i\beta \overbrace{\hat{H}_M}^{4)}}$

Algorithms: QAOA

The goal of the algorithm is to **maximise the expected value of the cost Hamiltonian**. (1)

Final states are obtained by a sequence of layers. (2)

Initial state is represented by a set of qubits in a superposition state. (3)

1)

$$F_p(\gamma, \beta) = \langle \psi_p(\gamma, \beta) | \hat{H}_C | \psi_p(\gamma, \beta) \rangle$$

3)

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

2)

$$|\psi_p(\gamma, \beta)\rangle = e^{-i\beta_p \hat{H}_M} e^{-i\gamma_p \hat{H}_C} \cdots e^{-i\beta_1 \hat{H}_M} e^{-i\gamma_1 \hat{H}_C} |s\rangle$$

Solution: Data generation

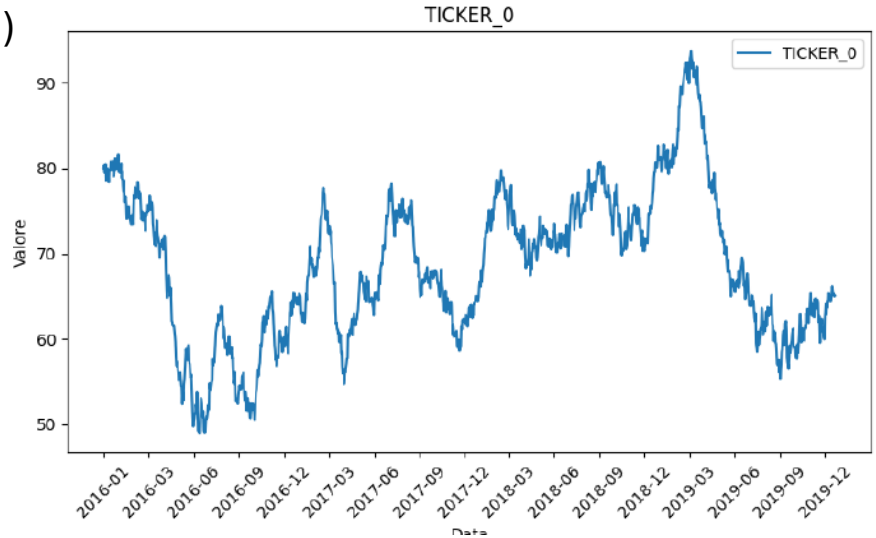
Data has been generated using qiskit «RandomDataProvider».

Data generated refers **to 4 assets**. (1)

The reference time interval starts on 2016-01-01 and ends on 2020-01-01.

On any given day, an activity may have a different value. (2)

1)



2)

Data	Ticker_0	Ticker_1	Ticker_2	Ticker_3
2016–01–01	80.0573	38.8648	69.2682	76.1653
2016–01–02	80.3390	38.7966	69.5354	76.6583
2016–01–03	79.4722	39.0100	70.7954	76.0500
...	...	...	...	...

Solution: Problem configuration

Average returns and the covariance matrix were calculated for each asset. (1)

Average returns, the covariance matrix, together with the risk factor, the budget and the penalty are used to configure the problem. (2)

The problem is converted into a quadratic program. (3)

2)
```
risk_factor = 0.2
budget = int(assets / 3 * 2)
penalty = budget

po = PortfolioOptimization (
    expected_returns = mu,
    covariances = sigma,
    risk_factor = risk_factor,
    budget = budget
)
```
3)
```
qp = po.to_quadratic_program()
```

1)
```
mu = data_provider.get_period_return_mean_vector()
sigma = data_provider.get_period_return_covariance_matrix()
```

Noiseless solution: Classical

The classical algorithm is based on the **computation of the minimal eigenvalues**.

Minimal eigenvalues are computed using an **exact minimum eigen solver** introduced by Numpy. (1)

The solver is wrapped up with Qiski's minimum eigen optimiser. (2)

This solution provides only one result, which is the optimal one.

1)
```
exact_mes = NumPyMinimumEigensolver()
```
2)
```
exact_eigensolver = MinimumEigenOptimizer(exact_mes)
result = exact_eigensolver.solve(qp)
```

Noiseless solution : VQE

VQE is a **hybrid algorithm**.

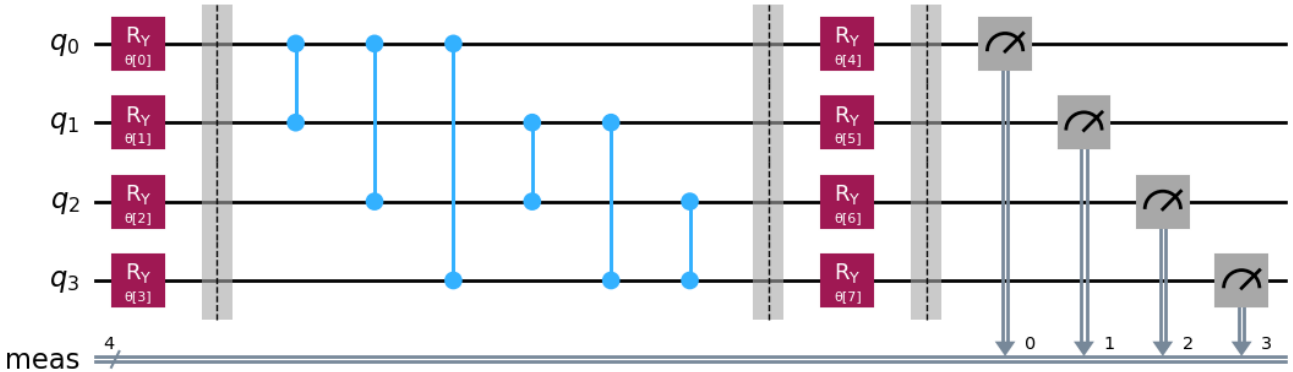
Combines the computational power of quantum circuits and the efficiency of classical optimisers.

The ansatz is created using «TwoLocal». (1)

The ansatz has been included in the SamplingVQE. (2)

1)
```
ry = TwoLocal(assets, "ry", "cz",
              reps = 1,
              entanglement = "full",
              insert_barriers = True )
```

2)
```
svqe_mes = SamplingVQE(sampler = Sampler() ,
                       ansatz =ry ,
                       optimizer = cobyla)
svqe = MinimumEigenOptimizer(svqe_mes, penalty)
result_svqe = svqe.solve(qp)
```



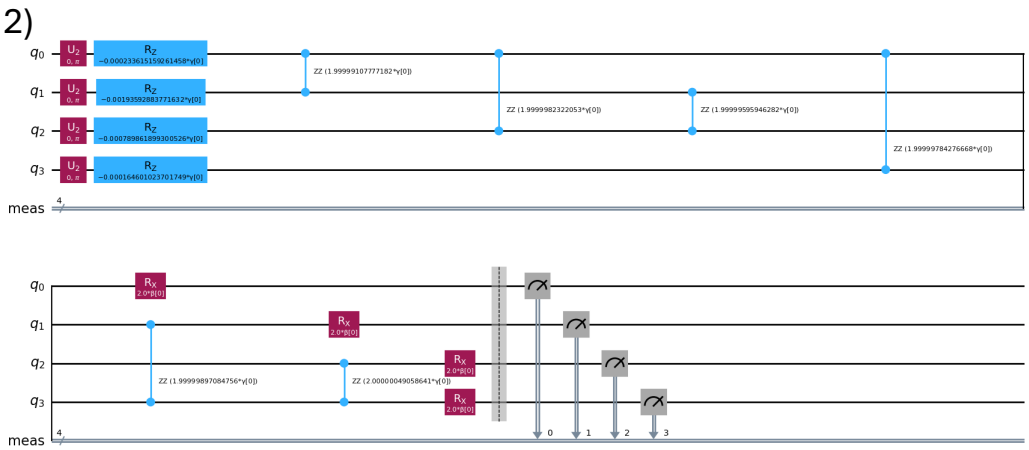
Noiseless solution: QAOA

QAOA is a **hybrid algorithm**.

Combines the computational power of quantum circuits and the efficiency of classical optimisers.

The **ansatz is automatically built over t problem**.

2)



1)

```
qaoa_mes = QAOA(sampler = Sampler(), optimizer = cobyla, reps = 1)
qaoa = MinimumEigenOptimizer(qaoa_mes, penalty)
result_qaoa = qaoa.solve(qp)
```

Noisy solution: Configuration

A **backend has been configured** to:

- Adapt to the number of qubits. (1)
- Include realistic details of noise sources. (2)
- Ensure greater fidelity in the simulation of device behaviour of the device. (3)

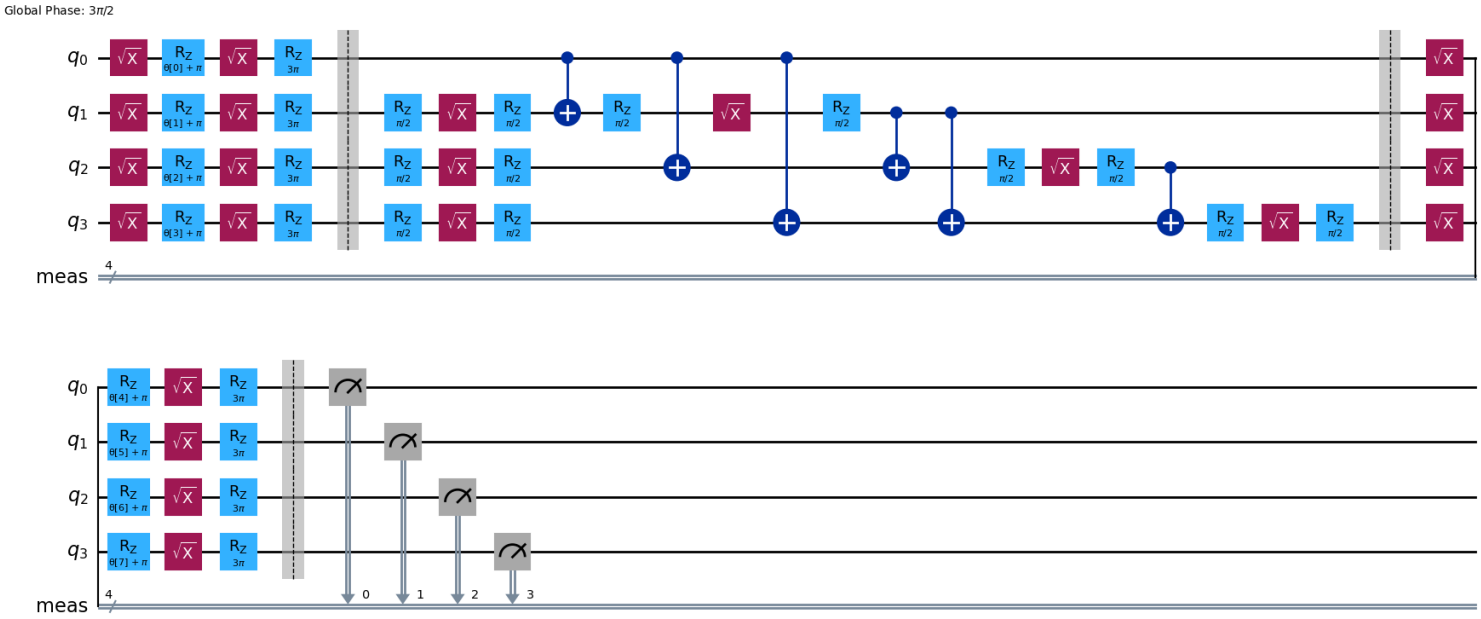
```
backend = GenericBackendV2 (
1)  num_qubits = assets,
    noise_info = True, 2)
    seed = seed,
3)  calibrate_instructions = True
)
# noise setting
noise_model = NoiseModel.from_backend(backend)
simulator = AerSimulator(noise_model = noise_model)
```

Noisy solution: VQE

Ansatz generated using TwoLocal (1) has been **transpiled to adapt to the new backend** (2).

1)
```
ry = TwoLocal(assets, "ry", "cz",
              reps = 1,
              entanglement = "full",
              insert_barriers = True )
```

2)
```
decomposed_ansatz = ansatz.decompose()
transpiled_ansatz = transpile(decomposed_ansatz,
backend = backend)
```



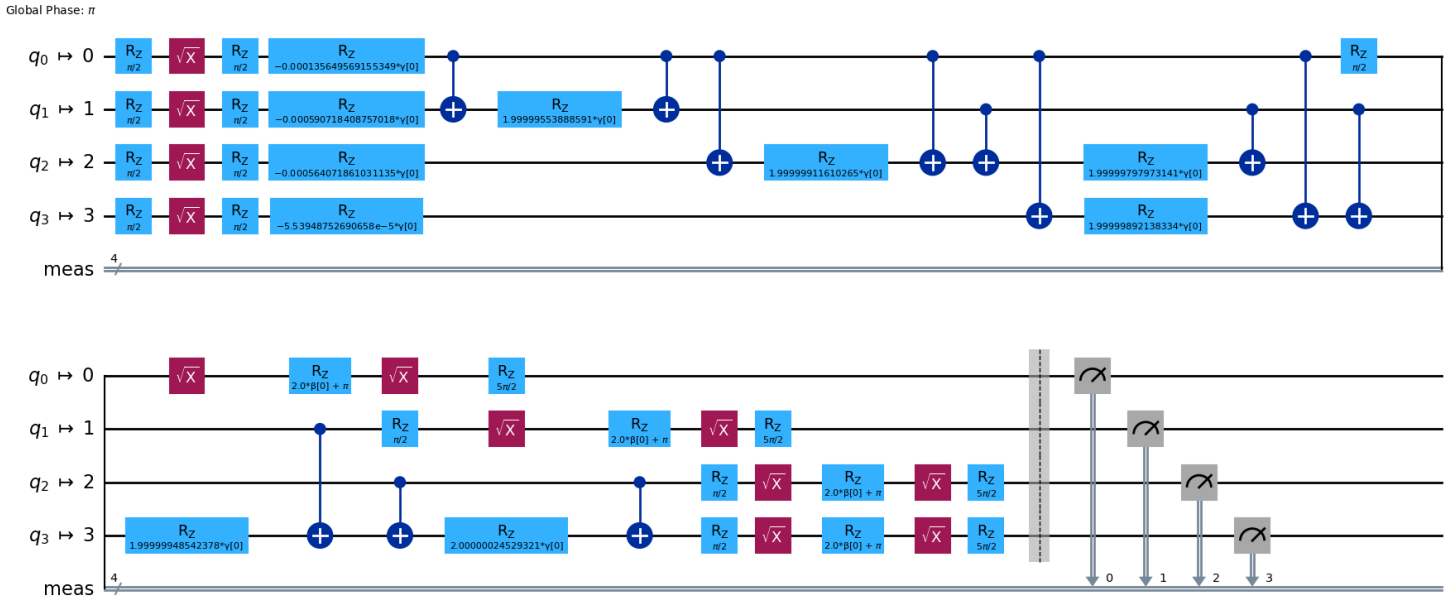
Noisy solution: QAOA

QAOA ansatz is generated after the solve.

A **first execution is necessary** to obtain the ansaz. (1)

Obtained ansatz has been **transpiled to adapt to the new backend**. (2)

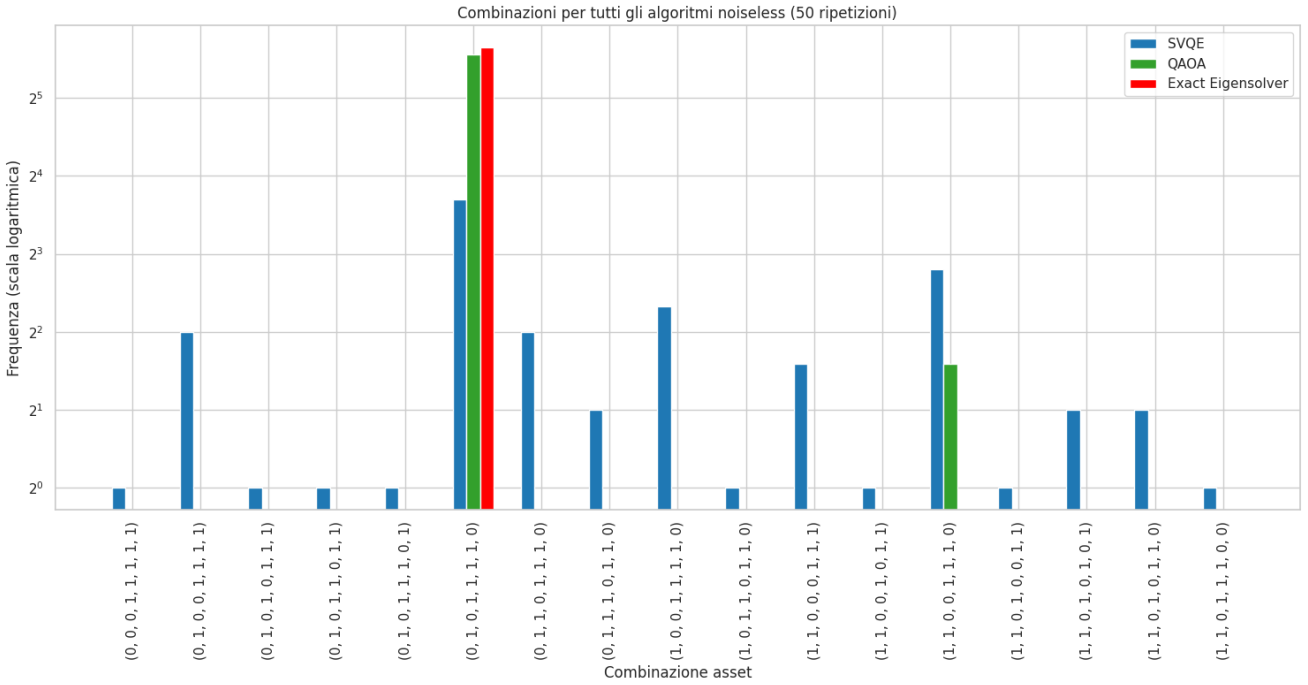
```
qaoa_mes_noisy = QAOA(sampler=Sampler(), optimizer=cobyla, reps=1)
qaoa_noisy = MinimumEigenOptimizer(qaoa_mes_noisy, penalty)
1)  _ = qaoa_noisy.solve(qp)

decomposed_circuit_noisy = qaoa_mes_noisy.ansatz.decompose()
2)  transpiled_circuit_noisy = transpile(decomposed_circuit_noisy,
                                         backend=simulator)

qaoa_mes_noisy.ansatz = transpiled_circuit_noisy
result_decomposed = qaoa_noisy.solve(qp)
```



Results: Noiseless

The classical solution is taken as a reference.

QAOA as an algorithm for solving (approximately) **combinatory problems** is **more accurate than VQE**.

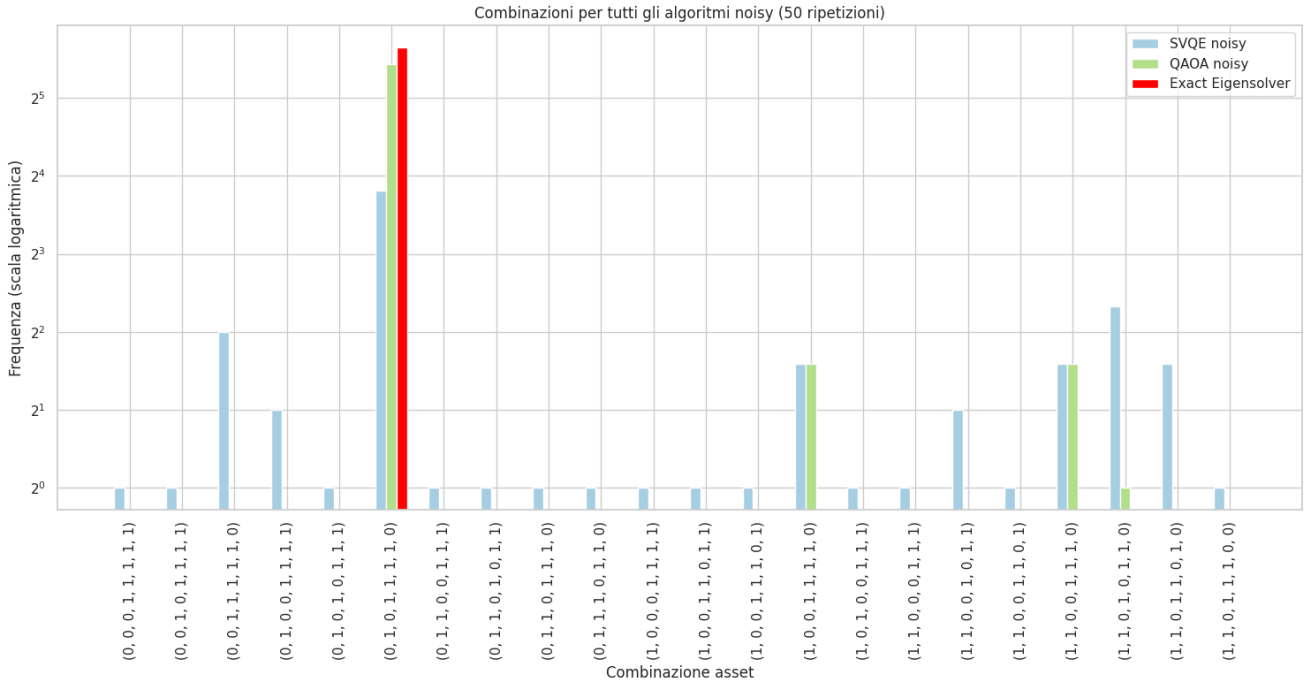


Results: Noisy

The classical solution is taken as a reference.

Even with noise, **QAOA performs better than VQE**.

Introducing noise increases the number of possible combinations.

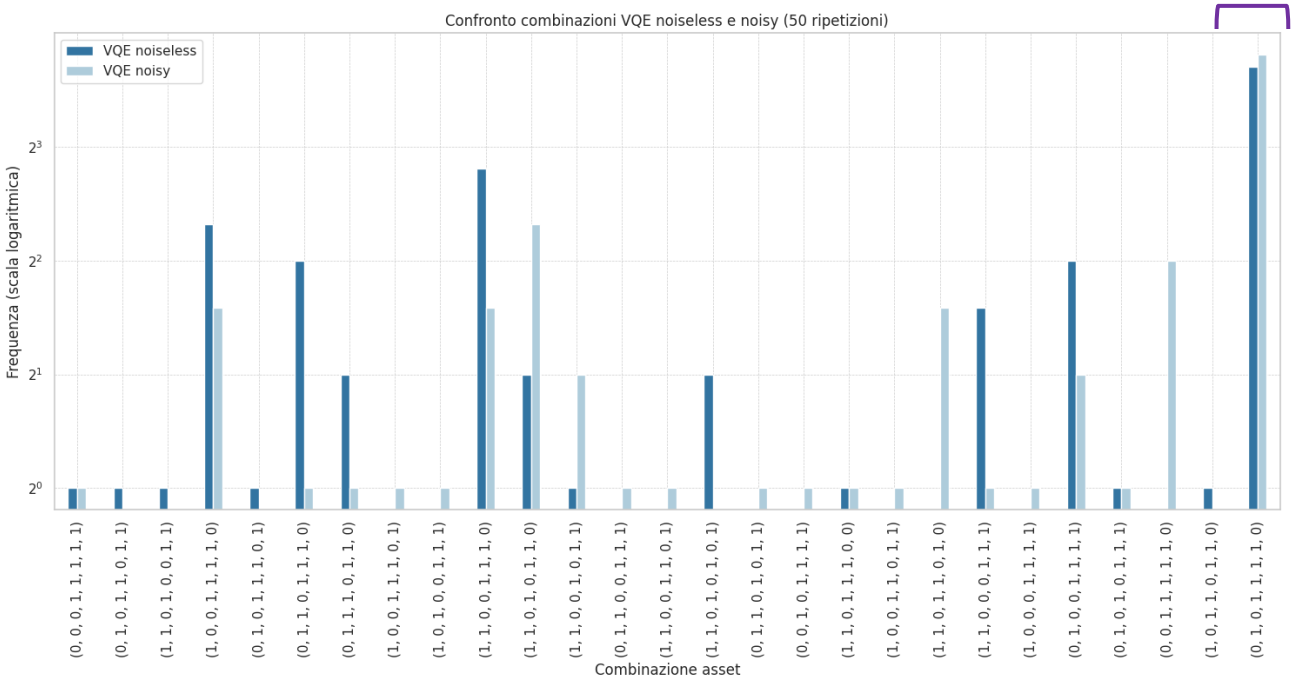


Results: Noiseless vs Noisy

The solutions proposed by the VQE **maintain a 'preference' regarding the main combination**. (1)

The noise version has more (sub-optimal) choices than the noiseless version.

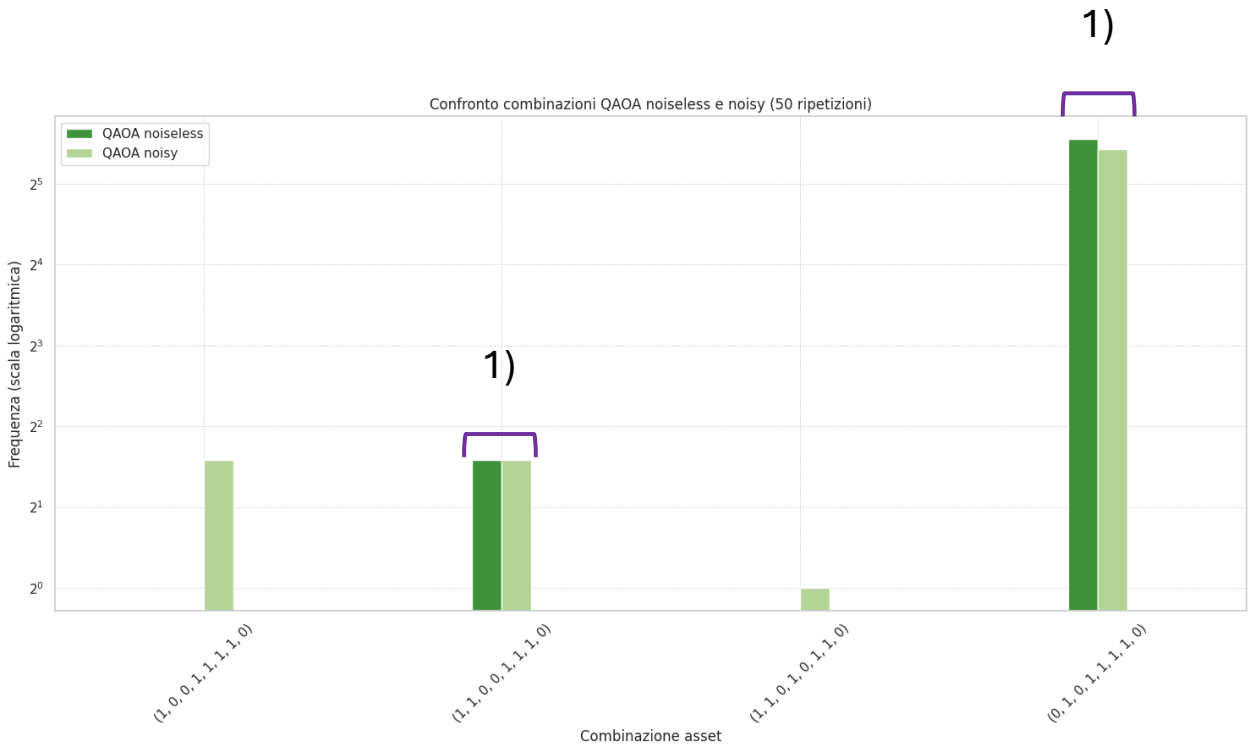
1)



Results: Noiseless vs Noisy

The solutions proposed by the QAOA **maintain a 'preference' regarding the main combinations**. (1)

The version with noise **has a few** (sub-optimal) **choices** compared to the version without noise.

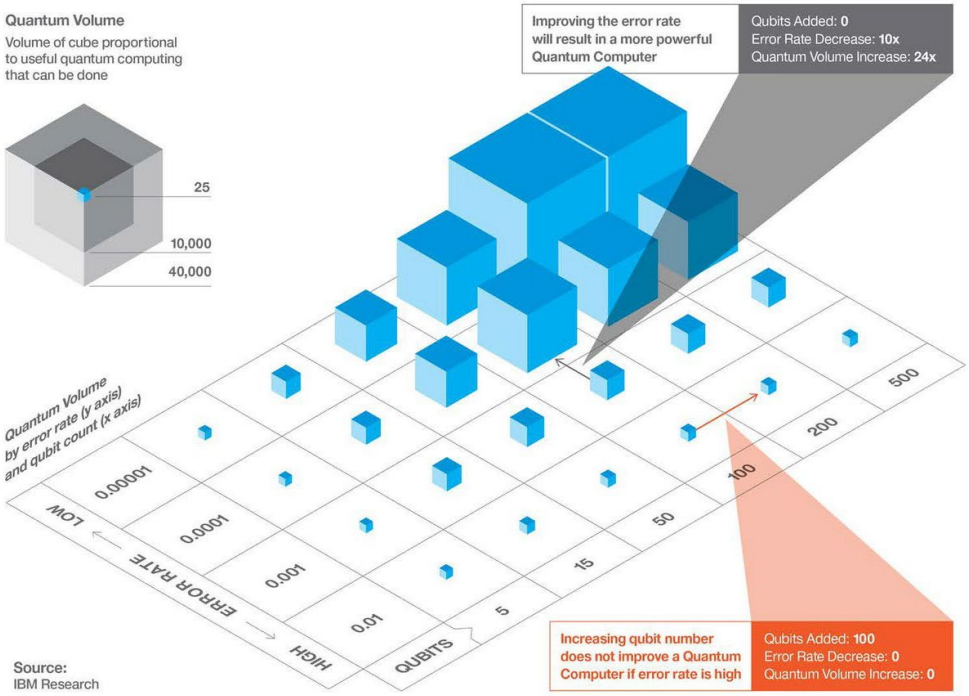


Conclusions

The **quantum approach offers several interesting solutions to solving complex problems**.

Unfortunately, **the technology currently available does not allow these approaches to be fully exploited**.

The presence of **noise compromises the quality of the results**, and even if managed at present, there is no physical possibility of handling a large number of qubits with a small error rate.



References

- https://www.investopedia.com/terms/f/finance.asp
- Buonaiuto, Giuseppe et al., (2023). “Best practices for portfolio optimization by quantum computing, experimented on real quantum devices”, Scientific Reports, Vol. 13 No. 1, p. 19434.
- Blekos, Kostas et al., (2024). “A review on quantum approximate optimization algorithm and its variants”, Physics Reports, Vol. 1068, pp. 1–66.
- Land, Ailsa H and Doig, Alison G (2010). An automatic method for solving discrete programming problems, Springer.
- Qiskit (2024). Portfolio Optimization using Qiskit Finance, https://qiskit-community.github. io/qiskit-finance/tutorials/01_portfolio_optimization.html.
- https://quantumcomputing.stackexchange.com/questions/8566/is-vqe-a-class-of-algorithms-or-a-specific-algorithm/8569#8569


Thanks