



# AQC Hack the Horizon Quantum Finance Challenge Portfolio Optimisation

Team 20

A K-aware, constraint-driven quantum portfolio optimiser



INTERNATIONAL YEAR OF  
Quantum Science  
and Technology

# Agenda

Team Introduction

Problem Statement

Our Approach

Mathematical Framework

QAOA Circuit

Results

Bonus Features

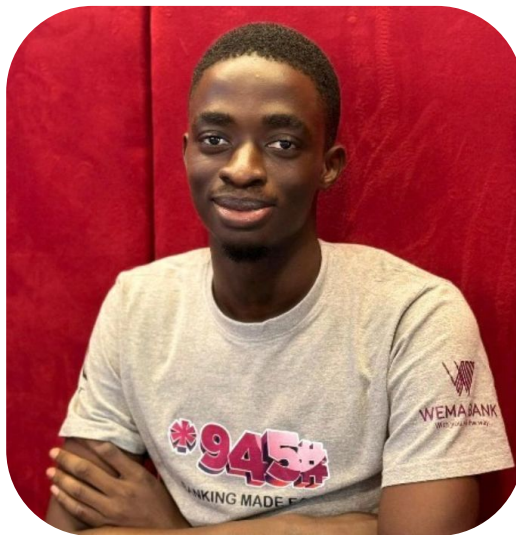
Limitations & Future Work





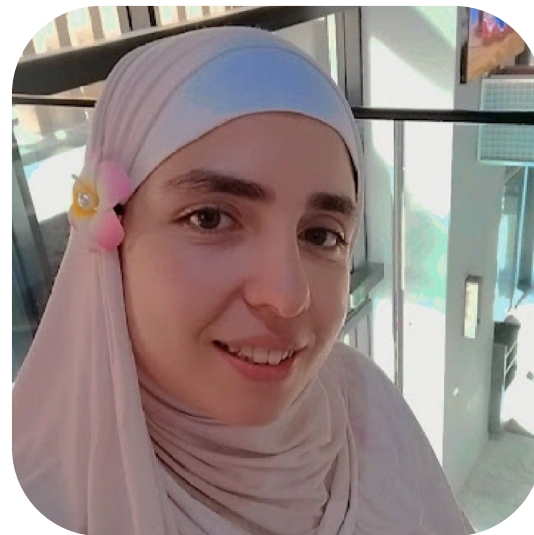
Musarandega Kudzai

Research Fellow  
The WISER



Omolaja Emmanuel

Software Developer  
Freelance/Contract



Khlifa Amal

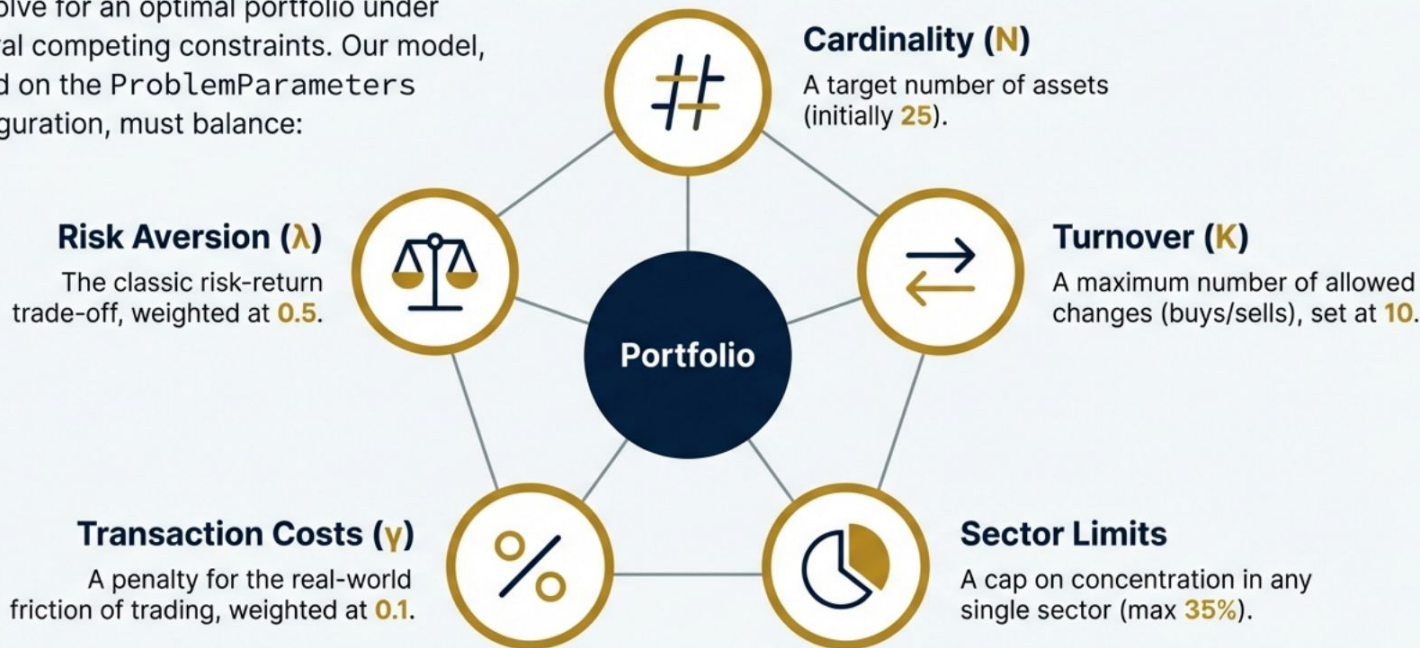
BSc Mechatronics Engineering  
National Engineering School  
Sousse



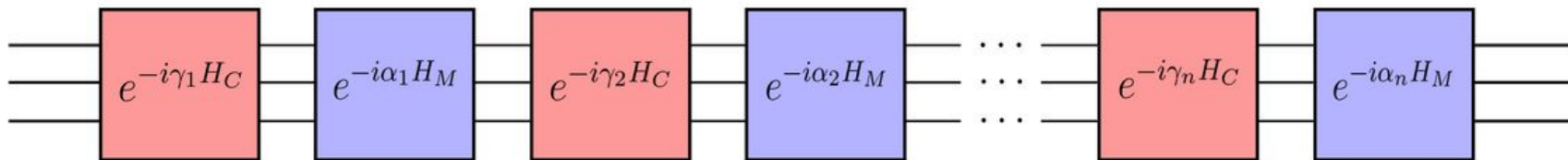
# Problem Statement

## Modern portfolio rebalancing is a multi-constraint balancing act.

We solve for an optimal portfolio under several competing constraints. Our model, based on the `ProblemParameters` configuration, must balance:

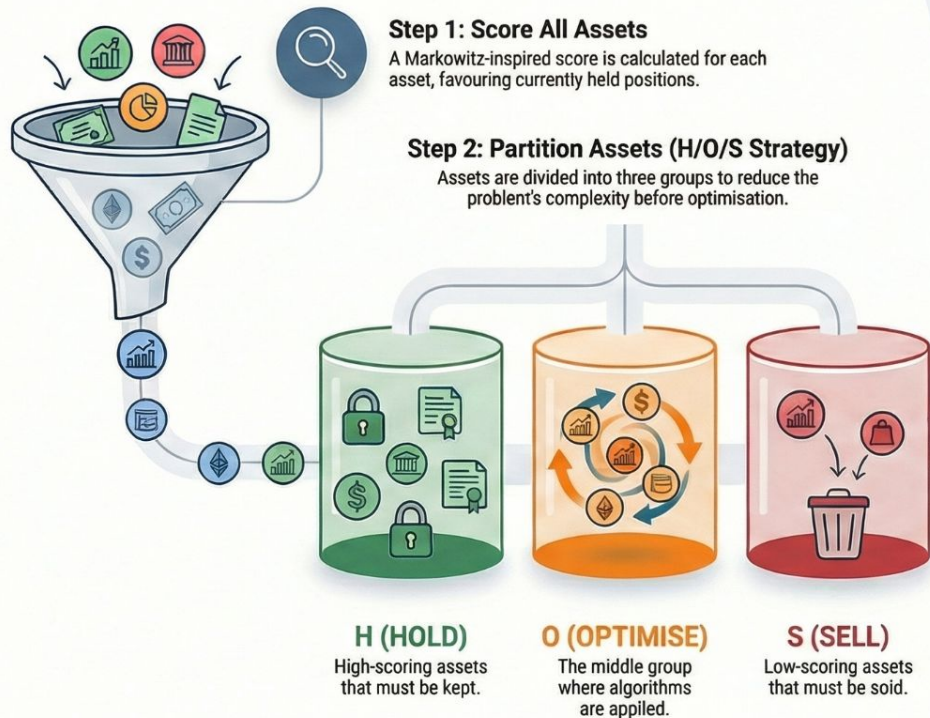


# Why Quantum? Why QAOA?



# A Modern Portfolio Optimisation Pipeline

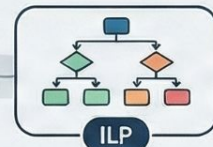
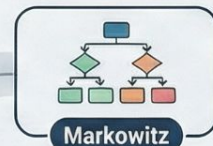
## STAGE 1: ASSET SCORING & PARTITIONING



## STAGE 2: OPTIMISATION & EVALUATION

### Step 3: Run Competing Optimisers on the 'O' Set

Classical (Markowitz, ILP) and Quantum (QAOA) methods find the optimal portfolio selection.



**KEY FINDING:**  
**QAOA Achieves the Highest Sharpe Ratio**  
The quantum algorithm finds a superior risk-adjusted return but requires significantly more time.

Optimisation Method Performance & Efficiency			
Method	Sharpe Ratio	Final Cardinality	Computation (s)
QAOA	2.58 ↑	30	⌚ 2893
Markowitz	2.46	26	0.53
ILP	2.46	26	0.03
Greedy	2.46	26	< 0.01

# Load the Data

Asset	Sector	Expected_Return	Previous_Position	Transaction_Cost	Market_Cap (billions)
ASSET_000	ENERGY	0.5895963173	1	0.0141923226	6.625272362
ASSET_001	CONS	-0.3505488782	1	0.02994283198	4.533083617
ASSET_002	HEALTH	0.7651140969	1	0.0254048048	5.646328397
ASSET_003	CONS	0.3865594574	0	0.04463280661	35.34552993

# Score assets

Score =     Marginal Return

–  $\lambda \times$  Marginal Risk

–  $\gamma \times$  Transaction Cost Penalty  
    only if not currently held

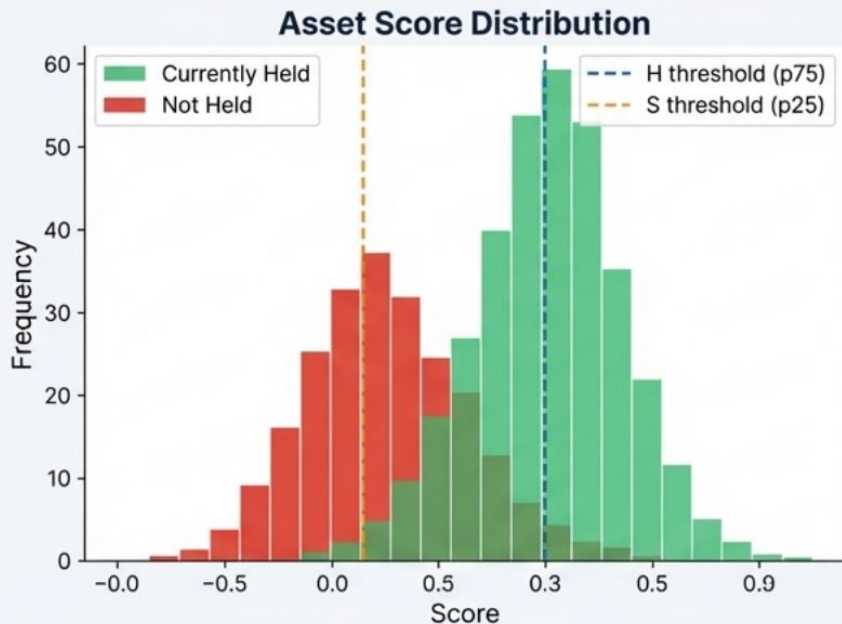
+  $\beta \times$  Holding Bonus  key lever for K-minimisation

–  $\eta \times$  Sector Concentration Penalty

```
# Marginal risk (correlation-aware)
portfolio_cov = data.cov_matrix[i, :] @ x_reference
marginal_risk = 2 * portfolio_cov + data.cov_matrix[i, i]

# STRONG holding bonus (encourages keeping current positions)
holding_bonus = params.beta_holding if is_currently_held else 0
```

# Scoring in Action and Distribution Cutoffs



**Key Insight:** The ``beta_holding`` bonus clearly shifts the distribution for currently held assets to the right, resulting in higher average scores. This ensures that the partitioning logic naturally prioritises retaining valuable existing positions.

# K-Aware Logic

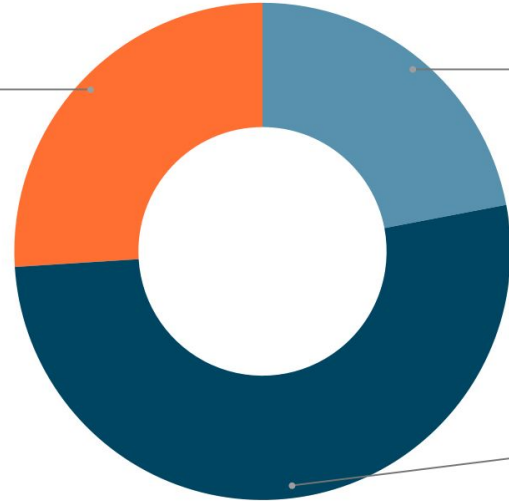
1. Feasibility Check
2. H Construction Prioritises K
3. K Budget for O

Asset Partiton

S (Sell)  
26.0%

H (Must Hold)  
22.0%

O (Optimise)  
52.0%



# Optimising $O$ with Classical Solvers

The core optimisation is performed **only on the 'O' subset** of assets, with a target of selecting ``d_target`` assets. We use a true **Mixed-Integer Quadratic Programme** (MIQP) solver for the Markowitz objective.

## CVXPY Formulation (Markowitz MIQP)

### Objective

```
objective = cp.Maximize(
    portfolio_return
    - params.lambda_risk * portfolio_variance
    - params.gamma_tc * transaction_cost
)
```

### Key Constraints

```
# Cardinality on O
constraints = [cp.sum(x_0) == d]

# K constraint
K_from_O = cp.sum(cp.abs(x_0 - x_prev_0))
constraints.append(
    K_from_H + K_from_O + K_from_S <= params.K_max_changes
)
```

**Note:** The K-constraint is not just a post-check; it is explicitly encoded into the mathematical programme, ensuring the solver produces a feasible solution.

# Mathematical Formulation

## Challenge Objective Function

We maximize the portfolio objective  $J(x)$ :

$$J(x) = \mu^T x - \lambda \cdot x^T \Sigma x - \gamma \cdot TC(x, x^{\text{prev}})$$

Where:

$x \in \{0,1\}^{50}$  : Binary decision vector (1=hold, 0=sell)

$\mu \in \mathbb{R}^{50}$  : Expected returns vector

$\Sigma \in \mathbb{R}^{50 \times 50}$  : Covariance matrix (risk)

$\lambda = 0.5$  : Risk aversion parameter

$\gamma = 0.1$  : Transaction cost weight

$x^{\text{prev}} \in \{0,1\}^{50}$  : Previous portfolio positions

# Mathematical Formulation

## Hard Constraints

1. Cardinality:  $\sum x_i = N$  (target portfolio size = 25)
2. K constraint:  $\sum |x_i - x_i^{\text{prev}}| \leq K$  (max 10 position changes)
3. Sector limits:  $\sum x_i \leq 0.35N$  for each sector
4. Binary:  $x_i \in \{0,1\}$

## Evaluation Metric (Primary)

$$\text{Sharpe Ratio} = (R - r_e) / \sigma$$

Where:

$R = \mu^T x$  : Portfolio return

$\sigma = \sqrt{x^T \Sigma x}$  : Portfolio risk

$r_e = 0.02$  : Risk-free rate

# Optimisation Arena

Five distinct methods were tasked with selecting the optimal portfolio from the 'O' (Optimise) set, each representing a different approach to solving the constrained problem.

## Markowitz (MIQP)

The industry gold standard, a Mixed-Integer Quadratic Program.

## ILP

A linearised Integer Linear Program, offering a faster approximation.

## Greedy

A simple heuristic that selects assets based on the pre-computed score.

## Random

The essential baseline for performance comparison.

## QAOA

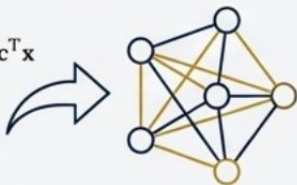
The quantum-inspired challenger, using a hybrid quantum-classical approach.

# The QAOA Method

The Quantum Approximate Optimisation Algorithm follows a three-step hybrid process:

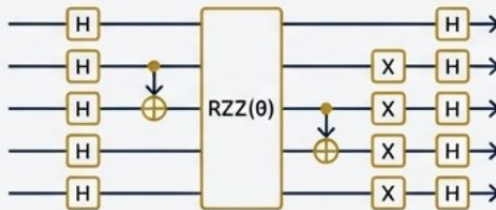
## 1. QUBO Formulation

$$\min_{\mathbf{x}} f(\mathbf{x}) = \underbrace{\mathbf{x}^T \mathbf{Q} \mathbf{x}}_{\text{return}} + \underbrace{\mathbf{c}^T \mathbf{x}}_{\text{risk}}$$



The objective function for the 'O' set, incorporating return, risk, transaction costs, and penalties for cardinality and turnover, is mapped to a Quadratic Unconstrained Binary Optimisation (QUBO) problem.

## 2. Quantum Simulation



The QAOA circuit explores the solution space to find a low-energy state, which corresponds to a high-quality portfolio candidate. The circuit is run on a classical simulator ('AerSimulator').

## 3. Classical Repair



The raw result from the quantum simulation is a strong candidate solution. A deterministic classical post-processing step refines this output to strictly enforce the final cardinality and turnover (K) constraints.

# QUBO Formulation for Set O

```
Q = -μ0·I           # Maximize return (diagonal)
    + λ·Σ00           # Minimize risk (quadratic)
    + γ·TC_penalty(x0, x_prev0) # Transaction costs
    + κ·K_penalty(x0, x_prev0)  # Discourage excessive changes (κ=0.6)
    + P·(Σx0 - d)2           # Cardinality constraint (P=60)
```

Where:

d = 14 : Target selections from 0  
P = 60 : Large penalty for cardinality violation

# Ising Hamiltonian Conversion

QUBO:  $E(x) = x^T Q x$  where  $x \in \{0, 1\}$

Transform:  $x_i = (1 - z_i)/2$  where  $z \in \{-1, +1\}$

Ising:  $H = \sum h_i z_i + \sum \sum J_{ij} z_i z_j$

Where:

$$h_i = Q_{ii}/2 + \sum_j Q_{ij}/4$$

$$J_{ij} = Q_{ij}/4$$

# QAOA Circuit Structure

$|\psi\rangle = |+\rangle^{\otimes 6}$  (Initial superposition)

For layer  $l = 1, \dots, p$ :

1. Cost Hamiltonian:

$$U_C(\gamma_l) = \exp(-i \cdot \gamma_l \cdot H)$$

Implemented as:

- Single-qubit:  $R_z(2\gamma_{li})$  on each qubit
- Two-qubit:  $R_{zz}(2\gamma_{lij})$  on each pair

2. Mixer Hamiltonian:

$$U_M(\beta_l) = \exp(-i \cdot \beta_l \cdot \sum_i X_i)$$

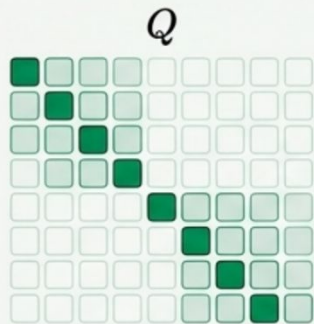
Implemented as:

- $R_x(2\beta_l)$  on each qubit

Parameters:  $\{\gamma_1, \dots, \gamma_p, \beta_1, \dots, \beta_p\}$

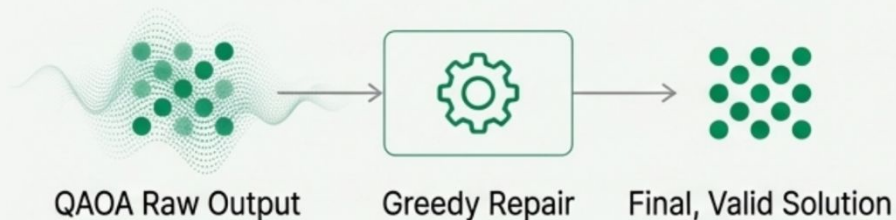
Optimized with: COBYLA (maxiter=100)

# The Circuit and Critical Repair



## QUBO Formulation Highlights

- The QUBO matrix  $Q$  directly encodes the objective function  $J(x)$  for the 'O' subset.
- **Return:**  $-\mu_O$  terms on the diagonal.
- **Risk:**  $+\lambda * \Sigma_{OO}$  terms for covariance.
- **Sharpened Penalties:** To guide the quantum search, we used strong penalty weights for both cardinality (penalty = 60.0) and K-constraint violations (k\_penalty\_weight = 0.6).



## Hybrid Approach: The Repair Step

QAOA's raw output is a probabilistic bitstring. A deterministic, greedy post-processing step is used to *guarantee* that the final solution strictly adheres to the cardinality ( $d_{\text{target}}$ ) and K-constraints.

# Performance Evaluation

To compare the outputs of each optimisation method, we use a comprehensive evaluation function that calculates key financial metrics and explicitly checks for constraint violations.

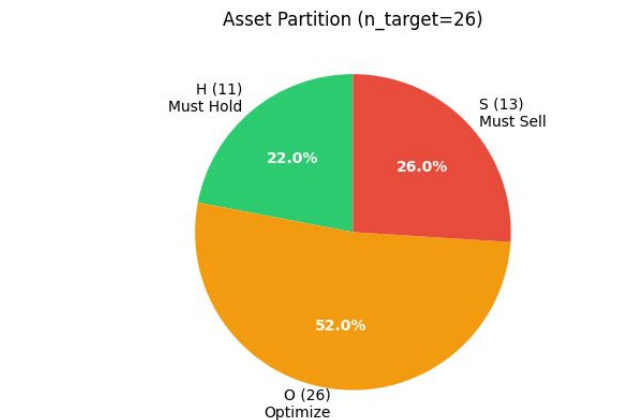
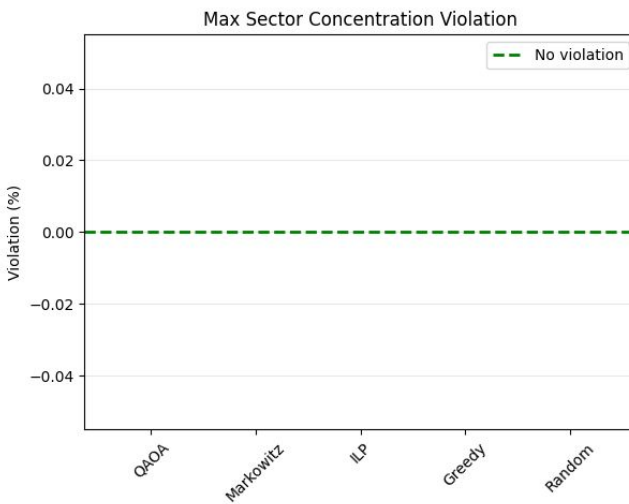
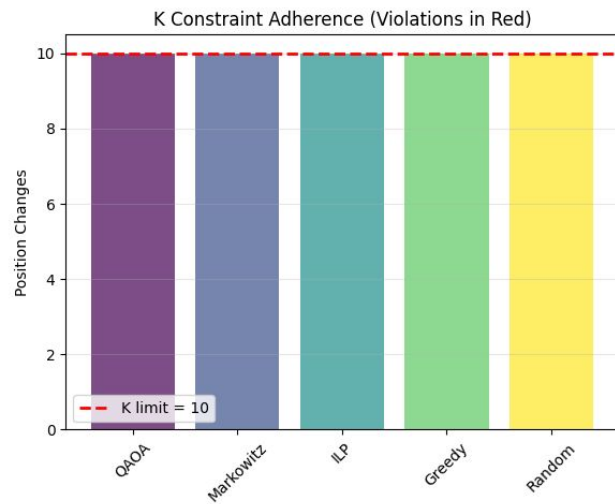
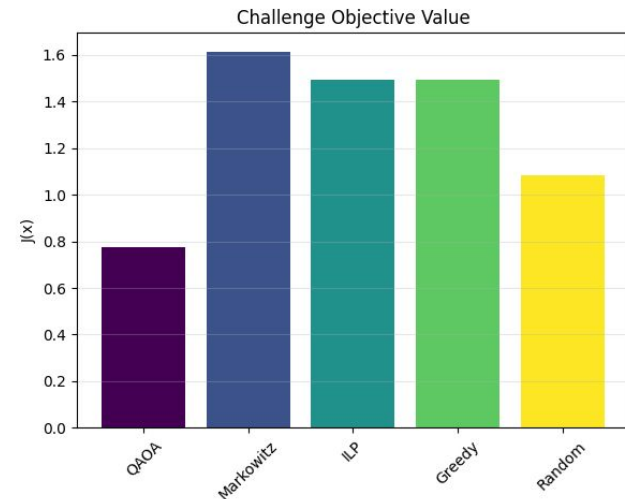
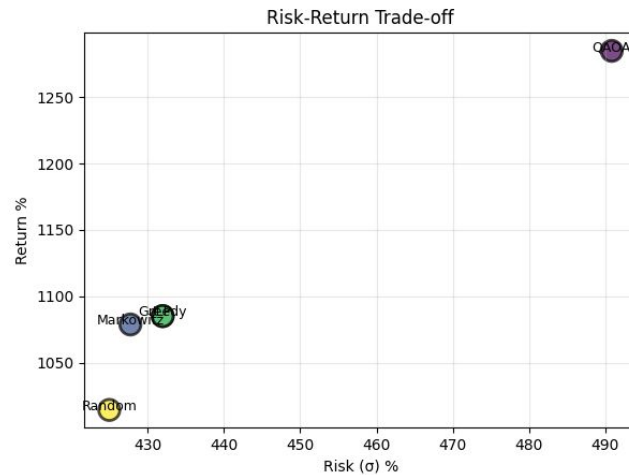
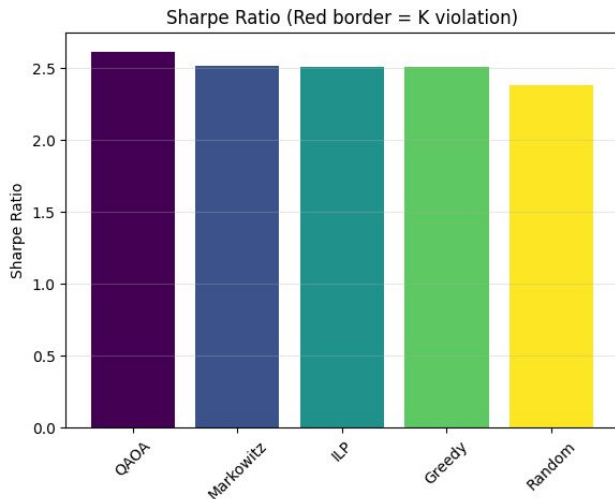
Key Metrics Captured ( <code>PerformanceMetrics</code> class)	
Core Performance	Constraint Satisfaction
<ul style="list-style-type: none"><li>• Sharpe Ratio</li><li>• Portfolio Return &amp; Risk (<math>\sigma</math>)</li><li>• Transaction Cost</li><li>• Challenge Objective <math>J(x)</math></li></ul>	<ul style="list-style-type: none"><li>• Final Cardinality</li><li>• K Used &amp; K Violation</li><li>• Max Sector Violation (%)</li><li>• Sector Minimums Violated (Bonus)</li></ul>

# Baseline Results

# Benchmark Methods (Fair Comparison)

Method	Optimisation on $O$	Solver	Time
QAOA	QUBO $\rightarrow$ Ising $\rightarrow$ Quantum circuit	Qiskit + COBYLA	$\sim 45$ min
Markowitz	Quadratic programming (full $\Sigma$ )	SCIP/ECOS_BB	0.3s
ILP	Integer linear (diagonal risk)	GLPK_MI	0.02s
Greedy	Top-d by score	Sorting	$< 0.001$ s
Random	Random d selections	Random	$< 0.001$ s

## Portfolio Optimisation Comparison



Method	Sharpe Ratio	Return	Risk ( $\sigma$ )	Trans. Cost	Challenge J(x)	Cardinality	K Used	K Violation	Max Sector Violation	Time (s)
QAOA	2.614	12.85	4.908	0.285	0.776	30	10	0	0	2733
Markowitz	2.517	10.79	4.277	0.266	1.612	26	10	0	0	0.300
ILP	2.507	10.85	4.319	0.274	1.493	26	10	0	0	0.024
Greedy	2.507	10.85	4.319	0.274	1.493	26	10	0	0	0.000
Random	2.382	10.14	4.250	0.279	1.085	26	10	0	0	0.000

**Bonus**

# Bonus Features Implemented

## 1. Liquidity Constraints (Bonus Challenge 3)

python

Filter:  $TC > 0.06 \rightarrow$  Force to S (must avoid)

Result: 8 illiquid assets excluded from optimization

## 2. Minimum Sector Diversification (Bonus Challenge 2)

python

Requirement:  $\geq 1$  asset per sector

Tracking: sector\_min\_violations metric

Result: All portfolios satisfy minimum (0 violations)

# Bonus Features Implemented

## 3. Real-Time Rebalancing Trigger (Bonus Challenge 1)

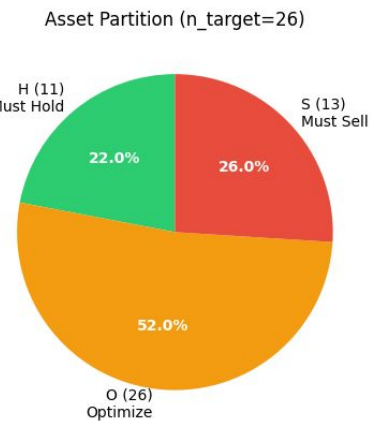
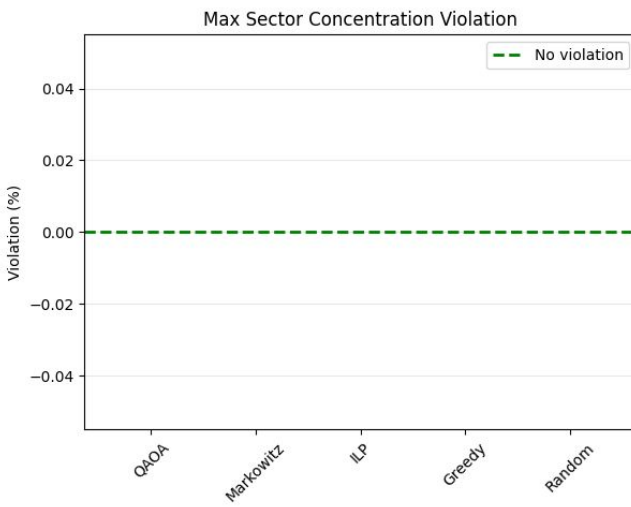
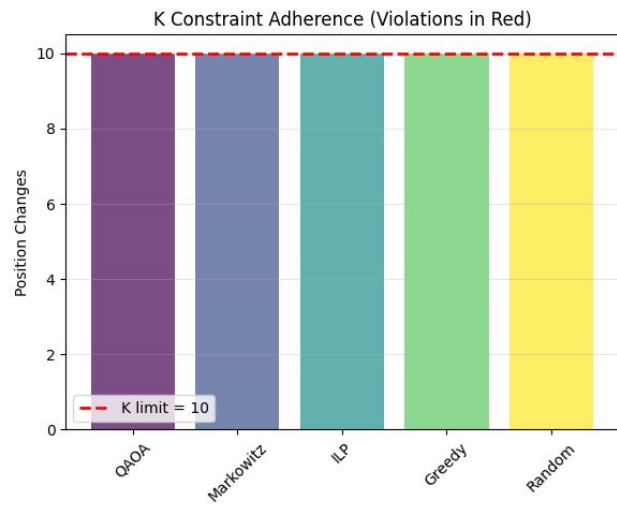
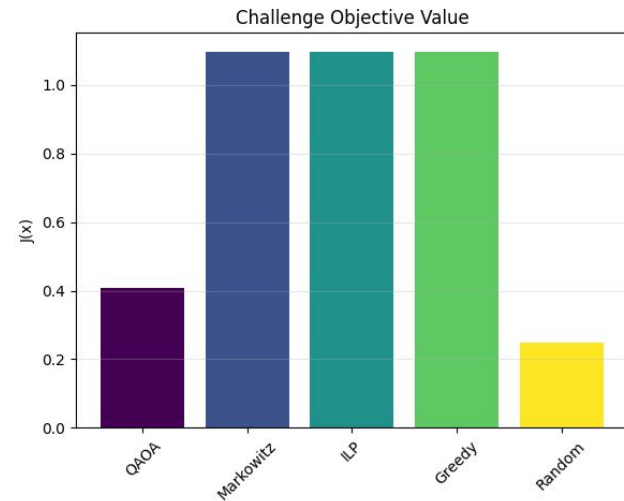
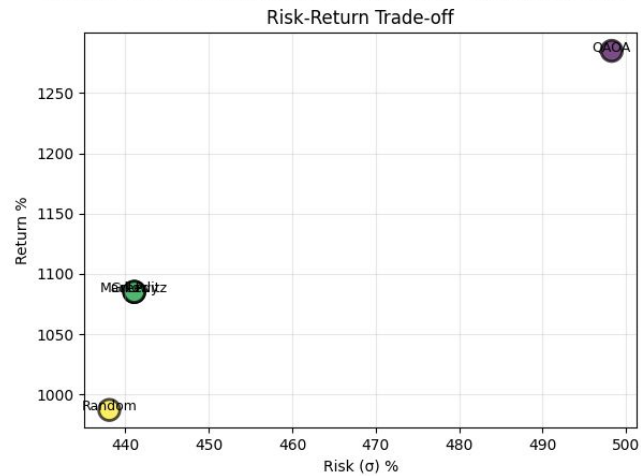
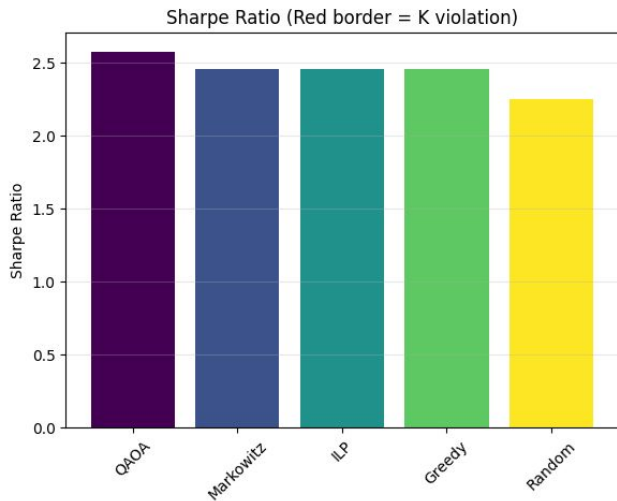
python

```
Triggers = {  
    'sharpe_drop':    Sharpe < 2.0  
    'k_violation':    K_used > K_max  
    'sector_breach':  Sector_conc > 0.35  
    'risk_spike':      $\sigma$  > 0.05  
    'return_drop':    R < 0.08  
}
```

If `any(Triggers)` → Rebalance recommended

# Bonus Results

## Portfolio Optimisation Comparison



Method	Sharpe Ratio	Return	Risk ( $\sigma$ )	Trans. Cost	Challenge J(x)	Cardinality	K Used	Max, Min Sector Violation	Rebalance	Time (s)
QAOA	2.575	12.85	4.983	0.285	0.408	30	10	0, 0	Yes	2893
Markowitz	2.456	10.85	4.411	0.274	1.096	26	10	0, 0	Yes	0.526
ILP	2.456	10.85	4.411	0.274	1.096	26	10	0, 0	Yes	0.031
Greedy	2.456	10.85	4.411	0.274	1.096	26	10	0, 0	Yes	0.000
Random	2.248	9.87	4.381	0.287	0.247	26	10	0, 0	Yes	0.000

# Effect of Bonus Implementations

1. Performance
2. Constraint Behaviour
3. Rebalancing Trigger Behaviour
4. Computational Cost

## QAOA

Metric	Without Bonus	With Bonus	Effect
Sharpe	<b>2.6140</b>	<b>2.5750</b>	Slight decrease
Return	12.85	12.85	No change
Risk	4.91	4.98	Slight increase
$J(x)$	0.776	0.408	Lower (due to new penalties)

## Classical Methods (Markowitz, ILP, Greedy)

Metric	Without Bonus	With Bonus	Effect
Sharpe	~2.51	~2.45	Slight decrease
Return	~10.8	~10.85	Essentially unchanged
Risk	~4.28–4.32	~4.41	Slight increase
$J(x)$	1.49–1.61	1.096	Lower (due to new penalties)

# Technical Highlights

## Algorithm Complexity

Traditional QAOA:  $O(2^{50})$  search space

Our H/O/S QAOA:  $O(2^{26})$  search space (99.999% reduction)

Circuit depth:  $p = 1$  &  $2$  layers

Gate count:  $\sim 700$  gates (26 qubits, pairwise interactions)

Classical optim: COBYLA with 100 iterations

# Innovation Summary

## 1. K-Aware Partitioning

→ Novel pre-filtering that respects position change limits

## 2. Correlation-Aware Scoring

→ Uses full covariance matrix (not just diagonal)

## 3. Hybrid Quantum-Classical

→ QAOA handles complex optimisation (O)  
→ Classical handles obvious decisions (H, S)

## 4. Constraint Repair Pipeline

→ Post-processing ensures all constraints satisfied

## 5. Fair Benchmarking

→ All methods optimise same problem (frozen H/S)

**Our work shows a robust pipeline for modern portfolio challenges, where quantum-inspired methods demonstrate significant promise.**

**1**

A Robust Framework

**2**

Efficient Classical Baselines

**3**

A Differentiated Quantum Approach

# Future Work Directions

1. Quantum Hardware Implementation and Noise Resilience
2. Dynamic Time-Series Optimisation with Adaptive H/O/S
3. Multi-Objective Optimisation Beyond Sharpe Ratio
4. Scalability to Large-Scale Portfolios via Hierarchical Decomposition
5. Integration with Quantum Machine Learning for Predictive Alpha

**Thank You**

# Equations Summary Sheet: Core Formulas

Portfolio Return:	$R = \sum_i \mu_i x_i$
Portfolio Risk:	$\sigma = \sqrt{\sum_{ij} x_i \sigma_{ij} x_j}$
Transaction Cost:	$TC = \sum_i c_i  x_i - x_i^{\text{prev}} $
Challenge Objective:	$J = R - \lambda \sigma^2 - \gamma \cdot TC$
Sharpe Ratio:	$S = (R - r_e) / \sigma$

Constraint: K	$\sum_i  x_i - x_i^{\text{prev}}  \leq 10$
Constraint: Cardinality	$\sum_i x_i = 25$
Constraint: Sector	$\sum_{i \in \text{sector}} x_i \leq 0.35 \cdot 25 = 8.75$

# Equations Summary Sheet: Scoring Function

$$\begin{aligned} \text{score}_i = & \mu_i && \# \text{ Expected return} \\ & - \lambda \cdot (2 \cdot \sum_{ij} x_j^{\text{prev}} + \sigma_{ii}) && \# \text{ Marginal risk} \\ & - \gamma \cdot \text{TC}_i \cdot (1 - x_i^{\text{prev}}) && \# \text{ Transaction cost penalty} \\ & + \beta \cdot x_i^{\text{prev}} && \# \text{ Holding bonus } (\beta=0.5) \\ & - n \cdot \max(0, \text{sector\_conc}_i - 0.20) && \# \text{ Sector penalty } (n=0.2) \end{aligned}$$

# Equations Summary Sheet: QAOA-Specific

Ising Energy:	$E = \sum_i h_i z_i + \sum_{ij} J_{ij} z_i z_j$
Cost Unitary:	$U_C(\gamma) = \exp(-i\gamma H)$
Mixer Unitary:	$U_M(\beta) = \exp(-i\beta \sum X_i)$
QAOA State:	$ \psi(\gamma, \beta)\rangle = U_M(\beta_p) U_C(\gamma_p) \dots U_M(\beta_1) U_C(\gamma_1)  +\rangle$
Expectation:	$\langle H \rangle = \langle \psi   H   \psi \rangle$