Quantum-Enhanced Portfolio Optimization Using Pauli Correlation Encoding for Efficient QUBO with Linear Constraints

Shiplu Sarker[1] and Shreeram Jawadekar[2]

[1]*Department of Electrical and Computer Engineering, Iowa State University*[*]

[2]*Department of Physics and Astronomy, Iowa State University*[†]

(Dated: August 11, 2025)

Abstract

This report presents a novel quantum computing approach to tackle large-scale portfolio optimization problems formulated as Quadratic Unconstrained Binary Optimization (QUBO) with linear inequality constraints. Leveraging Pauli Correlation Encoding (PCE), originally developed for MaxCut problems, we significantly reduce the quantum resource requirements by compressing logical variables into multi-qubit Pauli correlators. Our method maps each binary portfolio decision variable to a compressed spin representation via smooth relaxations using hyperbolic tangent functions of Pauli expectation values, enabling efficient hybrid quantum-classical optimization. To handle linear constraints inherent to realistic portfolio construction, we incorporate them as penalty terms within the QUBO formulation, resulting in a unified objective function as derived in Appendix C. This work pioneers the application of PCE to QUBO problems with linear inequalities, demonstrating a promising pathway for scalable quantum advantage in finance. We detail the theoretical framework, problem encoding, and hybrid variational algorithm design, supported by rigorous mathematical derivations and concrete Qiskit implementation.

I. INTRODUCTION

Portfolio optimization is a fundamental challenge in financial engineering, essential for constructing investment portfolios that balance return, risk, and regulatory requirements.Portfolio construction process exemplifies this complexity, involving numerous binary decision variables representing asset inclusions and a variety of linear inequality constraints capturing investment limits, risk factors, and business guardrails. Classical optimization approaches face scaling and accuracy challenges when addressing such high-dimensional constrained problems.

Quantum computing offers a promising alternative for these combinatorial optimization problems by harnessing quantum superposition and entanglement to explore vast solution spaces efficiently. In particular, expressing portfolio optimization as a Quadratic Unconstrained Binary Optimization (QUBO) problem enables direct implementation on current quantum devices.

* ssarker@iastate.edu

† shreeram@iastate.edu

This report presents an innovative application of Pauli Correlation Encoding (PCE)—originally developed for MaxCut problems—to effectively encode large-scale portfolio optimization QUBOs with linear inequality constraints. By exploiting PCE's qubit compression abilities, portfolio binary variables are succinctly mapped to multi-qubit Pauli correlators, significantly reducing quantum resource requirements without compromising problem fidelity. Linear constraints are incorporated as penalty terms within the QUBO objective function, following the explicit mathematical derivations provided in Appendices A through C.

To the best of our knowledge, this work is the first to apply PCE to QUBOs with linear inequality constraints in practical portfolio optimization settings. We detail the theoretical framework, including binary-to-spin transformations and penalty formulation, describe the PCE method with smooth hyperbolic tangent relaxations, and integrate these components within a hybrid quantum-classical variational optimization scheme. Building upon and generalizing the Quantum MaxCut PCE implementation, this effort establishes a robust foundation for scalable quantum portfolio optimization addressing real-world investment constraints and business objectives.

II. THEORETICAL FRAMEWORK: PORTFOLIO OPTIMIZATION AS CONSTRAINED QUBO

In this section, we formulate the portfolio construction problem as a Quadratic Unconstrained Binary Optimization model with additional linear inequality constraints encoded as penalty terms. The formulation follows the derivations in Appendix B and Appendix C, where Appendix B provides the unconstrained QUBO objective for portfolio index-tracking, and Appendix C incorporates the guardrails and investment limits as penalties to obtain a single, unconstrained QUBO form. Detailed term-by-term derivations are available in those appendices and are not repeated here.

A. Objective Function for Portfolio Optimization (See Appendix B for details)

The primary goal is to construct a portfolio whose aggregated characteristics across defined risk buckets L and features J closely match specified target values $K_{l,j}^{\text{target}}$ where

$l \in L, j \in J$. Let $y_c \in \{0,1\}$ be the binary decision variable indicating whether asset c is included in the portfolio ($y_c = 1$) or not ($y_c = 0$). Starting from the classical form, the optimization objective is given by:

$$\min \sum_{l \in L} \sum_{j \in J} \rho_j \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} x_c - K_{l,j}^{\text{target}} \right)^2 . \tag{1}$$

Here, x_c is the assigned quantity of asset c if it is included in the portfolio, $\beta_{c,j}$ represents the contribution of asset c to characteristic j, ρ_j refers to the weighting factor for characteristic j in the objective, and $\mathbb{K}_l$ denotes the set of securities within risk bucket l as defined by the model. The quantity x_c is fixed as:

$$x_c = a_c y_c, \quad a_c = \frac{m_c + \min\{M_c, i_c\}}{2\delta_c}, \tag{2}$$

where m_c is the minimum tradable amount, M_c is the maximum tradable amount, i_c represents the basket inventory for asset c, and δ_c refers to the minimum increment associated with asset c. Substituting $x_c = a_c y_c$ into Eq (B1) and expanding the squared terms separates constant, linear, and quadratic contributions in y. The resulting QUBO objective function can be written as:

$$\mathcal{L}^{\text{obj}}(y) = \sum_{\substack{c',c \\ c' \neq c}} Q_{c',c}^{\text{obj}} y_{c'} y_c + \sum_c h_c^{\text{obj}} y_c + d^{\text{obj}}. \tag{3}$$

The coefficients corresponding to the quadratic, linear, and constant contributions in the objective function are given by:

$$Q_{c'c}^{\text{obj}} = \begin{cases} \sum_{l \in L} \sum_{j \in J} \rho_j \mathbb{I}_{[c', c \in \mathbb{K}_l]} \beta_{c',j} a_{c'} \beta_{c,j} a_c & \text{if } c' \neq c \\ 0 & \text{if } c' = c \end{cases}, \tag{4}$$

$$h_c^{\text{obj}} = \sum_{l \in L} \sum_{j \in J} \rho_j \mathbb{I}_{[c \in \mathbb{K}_l]} \left[(\beta_{c,j} a_c)^2 - 2K_{l,j}^{\text{target}} \beta_{c,j} a_c \right], \quad d^{\text{obj}} = \sum_{l \in L} \sum_{j \in J} \rho_j \left(K_{l,j}^{\text{target}} \right)^2 . \tag{5}$$

In the expression above, $\mathbb{I}_{[.]}$ is the indicator function, which is defined as 1 if the condition in the argument is true, and 0 otherwise.

B. Linear Inequality Constraints as QUBO Penalty (See Appendix C for details)

The realistic portfolio must satisfy several linear guardrail constraints. The constraints are as follows:

$$\sum_{c \in C} y_c \leq N, \tag{6}$$

This constraint limits the total number of assets permitted in the portfolio, where N is the maximum number of assets.

$$\frac{\text{minRC}}{MV^b} \leq \sum_{c \in C} \frac{p_c \delta_c}{100 MV^b} x_c \leq \frac{\text{maxRC}}{MV^b}, \tag{7}$$

This constraint given above limits the residual cash flow of the portfolio, where p_c is the market price for asset c. the last set of constraints are the min/max characteristic value constraints. They are defined for each characteristic j in each risk group l and are expressed as:

$$b_{l,j}^{\text{low}} \leq \sum_{c \in \mathbb{K}_l} \frac{p_c \delta_c}{100 MV^b} \beta_{c,j} x_c \leq b_{l,j}^{\text{up}}, \quad \forall j \in J, l \in L. \tag{8}$$

All constraints can be written in the standard form:

$$Ay \geq b, \tag{9}$$

where each row of A correspond to a constraint. The column vector b contains the bounds. To embed these constraints into our unconstrained QUBO, a squared hinge loss penalty is used (see Appendix C for details):

$$\mathcal{L}^{\text{penalty}}(y) = \text{penalty} \times \sum_i \max(b_i - A_i y, 0)^2. \tag{10}$$

For a satisfied constraint ($b_i - A_i y_i \leq 0$), the penalty function is zero. However, for a violated constraint, the penalty increases quadratically with the magnitude of the violation. The penalty coefficient is fixed in our approach and is given by:

$$\text{penalty} = (\text{max\_obj} - \text{min\_obj}) \times 1.1 \tag{11}$$

where max_obj and min_obj are the sums of the positive and negative entries of Q^{obj}, respectively. This ensures that constraint violations are heavily penalized relative to gains from the primary objective. The final unconstrained QUBO for the constrained portfolio optimization is:

$$\mathcal{L}^{\text{total}}(y) = \mathcal{L}^{\text{obj}}(y) + \mathcal{L}^{\text{penalty}}(y) \tag{12}$$

This formulation allows the constrained portfolio optimization problem to be handled entirely within a QUBO framework, ensuring that feasible portfolios which satisfy all guardrails are favored in the search for an optimal solution.

III. PAULI CORRELATION ENCODING FOR QUBO WITH CONSTRAINTS

In this work, we extend the application of Pauli Correlation Encoding (PCE), originally introduced for the MaxCut problem in ref. [1], to solve the more general and practically relevant case of QUBO problems with linear inequality constraints typical in portfolio optimization. The central motivation for employing PCE lies in its demonstrated capability to substantially reduce quantum resource requirements by compressing classical variables into multi-qubit Pauli correlators, thereby enabling scalable quantum optimization on near-term hardware.

While PCE has been successfully applied to MaxCut instances in ref. [1]—a special subclass of unconstrained quadratic binary problems—it has, to the best of our knowledge, not been previously employed for constrained QUBOs. Our approach bridges this gap by integrating linear constraints as penalty terms into the QUBO formulation (as detailed in Appendices B and C), and leveraging PCE to encode the resulting binary variables efficiently.

For completeness, Appendix A provides a formal binary-to-spin variable mapping, foundational for variational quantum algorithms that operate naturally on spin variables $s_c \in \{-1, +1\}$. However, in our implementation, we directly optimize the total unconstrained penalized objective given in Eq: 12. Each binary variable y_c is mapped to its spin representation as

$$y_c = \frac{1 - s_c}{2}, \quad s_c = \tanh\left[\alpha \langle \psi(\theta) | \Pi_c | \psi(\theta) \rangle\right] \tag{13}$$

where $\langle \Pi_c \rangle$ denotes the expectation value of the Pauli correlator associated to variable c, and α is a scaling parameter controlling the smoothness and discreteness of the relaxation. In this work, we have set $\alpha = 15$. For the parametrized quantum state $|\psi(\theta)\rangle$, we have used the hardware-efficient ansatz.

Following the PCE construction in ref. [1], we group logical variables into sets encoded via multi-qubit Pauli strings of type XX, YY, and ZZ, allowing a compressed representation that dramatically lowers the number of physical qubits required compared to naive encodings. Specifically, for our quadratic QUBO formulation, each logical variable is mapped to a unique two-body Pauli correlator Π_c constructed as follows: The variables are partitioned into three groups (e.g., node sets for X, Y, and Z bases), and for each group, Π_c is formed by selecting a unique pair of physical qubits via combinations and placing the corresponding Pauli operator (X, Y, or Z) on those qubits, with identity I elsewhere. This two-body encoding ($k = 2$) aligns with the quadratic nature of our portfolio QUBO, ensuring efficient measurement of correlators in just three bases while compressing up to $\mathcal{O}(n^2)$ variables into n qubits. Unlike the MaxCut case in ref. [1], we do not include explicit regularization terms ($\mathcal{L}^{\text{reg}}$) in ref. [1] since, empirically and as supported by our preliminary investigations, the penalized total objective with tanh-relaxed spins suffices to guide the optimization to meaningful discrete solutions without such terms.

In our portfolio example with 31 bonds/assets, we use 6 qubit quantum circuit. We explicitly partition the set of logical variables into three groups to match the PCE mapping: the first 10 assets are encoded as two-body XX Pauli correlators, the next 10 as YY correlators, and the final 11 as ZZ correlators. This assignment is systematically performed: for each group, unique pairs of physical qubits are chosen, and the corresponding Pauli operator (X, Y, or Z) is placed on those qubits, while identity operators are used elsewhere. Such a construction ensures that all 31 binary variables are efficiently compressed into the available quantum register with minimal measurement overhead—only three measurement bases (X, Y, Z) are required for full state tomography.

To illustrate the Pauli operators explicitly, consider a system with $n = 6$ physical qubits (sufficient for up to ${}^6C_2 = 15$ correlators per group). The two-body correlators for the X-group (first 10 assets) are constructed as follows (in Qiskit little-endian notation, with qubit 0 as the least significant): for First 10 assets: Asset 1: $IIIIXX$ Asset 2: $IIIXIX$ Asset 3: $IIXIIX$ Asset 4: $IXIIIX$ Asset 5: $IIIXXI$.... Analogous strings are generated for the Y-group (replacing X with Y) and Z-group (replacing with Z), with the Z-group accommodating 11 assets by extending to additional combinations if needed (e.g., via a larger n). Each Π_c is thus a unique tensor product of two Paulis and identities, enabling the compressed encoding of our 31-variable portfolio QUBO. For assets 11-20 assets we used

$IIIIYY$ and so on.. For 21-31 assets we used $IIIZZ, \cdots$. Our approach thus constitutes a novel and resource-efficient extension of PCE for constrained QUBO problems, directly applicable to real-world portfolio optimization with linear constraints (see Appendices B and C). The detailed penalty formulation encapsulates investment guardrails as quadratic penalties, which seamlessly integrate with the PCE framework to enable hybrid quantum-classical variational algorithms capable of handling both objective fidelity and constraint satisfaction.

In summary, this section formalizes how PCE can be naturally and effectively adapted beyond MaxCut to the constrained portfolio QUBO optimization, maintaining rigorous mathematical fidelity while exploiting quantum encoding compression to facilitate near-term scalable implementations.

IV. CONCLUSION

In this report, we presented a novel quantum-enhanced framework for portfolio optimization, reformulating it as a constrained QUBO and pioneering the use of Pauli Correlation Encoding (PCE) to handle linear inequalities via penalties. By compressing 31 binary variables into 6 qubits, our approach drastically reduces resource requirements while maintaining constraint satisfaction and objective fidelity through tanh-relaxed spin mappings.

Our contributions include the first application of PCE to constrained QUBOs in finance, with simulations confirming efficient optimization and feasible solutions. This work establishes a scalable pathway for quantum portfolio construction, with future directions including real hardware experiments and advanced ansatz designs to further enhance performance.

ACKNOWLEDGEMENTS

Appendix A: Binary-to-Spin Variable Mapping for Quadratic Binary Forms

This appendix presents a general mathematical derivation to transform a quadratic form defined over binary variables into an equivalent representation using spin variables, commonly referred to as the Ising model formulation. Such a transformation is fundamental in the context of quantum optimization, allowing problems originally expressed in binary form to be naturally handled by quantum algorithms or Ising-based classical heuristics. By establishing this equivalence in a rigorous manner, the derivation provides a foundational tool applicable to a wide range of Quadratic Binary Optimization problems, both constrained and unconstrained, thereby furnishing a unified framework for their quantum-ready reformulations.

Consider the general quadratic form over binary variables:[1]

$$
\begin{aligned}
\mathcal{L}(y) &= \sum_{\substack{c',c \\ c'\neq c}} Q_{c',c} y_{c'} y_c + \sum_{c} h_c y_c + d \\
&= y^T Q y + h^T y + d \\
&= 2 \sum_{\substack{c',c \\ c'<c}} Q_{c',c} y_{c'} y_c + \sum_{c} h_c y_c + d,
\end{aligned} \tag{A1}
$$

where $y = (y_1, \ldots, y_n)^T$, with $y_c \in \{0, 1\}$ for all $c \in \{1, \cdots, n\}$. Q is a real, symmetric $n \times n$ matrix (i.e., $Q_{c'c} = Q_{cc'} \quad \forall c', c \in \{1, \cdots, n\}$) whose diagonal entries are zero (i.e, $Q_{c,c} = 0 \quad \forall c \in \{1, \cdots, n\}$). The term $h = (h_1, \ldots, h_n)^T$ is a column vector with real entries, and d is a real number.

To convert Eq. (A1) to the equivalent spin representation, one needs to introduce spin variables $s_c \in \{-1, 1\}$ corresponding to the binary variables y_c, defined by:

$$
y_c = \frac{1 - s_c}{2}, \quad \forall c \in \{1, \cdots, n\}. \tag{A2}
$$

This transformation maps the binary value $y_c = 0$ to the spin value $s_c = 1$, and $y_c = 1$ to $s_c = -1$. Using this change of variables, the product $y_{c'} y_c$ can be expressed as:

$$
y_{c'} y_c = \left(\frac{1 - s_{c'}}{2}\right)\left(\frac{1 - s_c}{2}\right) = \frac{s_{c'} s_c - s_{c'} - s_c + 1}{4}, \quad \forall c', c \in \{1, \cdots, n\}. \tag{A3}
$$

Substituting this expression into the quadratic term of $\mathcal{L}(y)$ given in Eq (A1), one can write:

$$
\sum_{\substack{c',c \\ c'\neq c}} Q_{c',c} y_{c'} y_c = \sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4} s_{c'} s_c - \sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4} s_{c'} - \sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4} s_c + \sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4}. \tag{A4}
$$

The second term of the expression above can be rewritten as:

$$
\sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4} s_{c'} = \sum_{\substack{c,c' \\ c\neq c'}} \frac{Q_{c,c'}}{4} s_c = \sum_{\substack{c',c \\ c'\neq c}} \frac{Q_{c',c}}{4} s_c, \tag{A5}
$$

[1] This reformulation leverages the properties of Q defined in the text; the summation is restricted to $c' \neq c$ because the diagonal entries are zero, while the factor of 2 arises from the symmetry of Q.

where in the middle expression, the indices c' and c have been interchanged as they are dummy summation indices ranging from 1 to n. In the last equality, the symmetry of the matrix Q has been exploited. Thus, the second and third terms in Eq. (A4) are identical, and hence, Eq. (A4) simplifies to:

$$\sum_{\substack{c',c\\c'\neq c}} Q_{c',c} y_{c'} y_c = \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{4} s_{c'} s_c - \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{2} s_c + \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{4} \tag{A6}$$

On the other hand, the linear term in the expression for $\mathcal{L}(y)$ given in Eq. (A1) becomes:

$$\sum_c h_c y_c = \sum_c h_c \left(\frac{1 - s_c}{2} \right) = - \sum_c \frac{h_c}{2} s_c + \sum_c \frac{h_c}{2} \tag{A7}$$

Substituting Eqs. (A6) and (A7) into Eq. (A1) and collecting quadratic, linear, and constant terms separately, $\mathcal{L}(y)$ can be expressed in terms of spin variables as:

$$\mathcal{L}(y) = \mathcal{L}^{\text{spin}}(s) = \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{4} s_{c'} s_c - \sum_c \left(\sum_{\substack{c'\\c'\neq c}} \frac{Q_{c',c}}{2} + \frac{h_c}{2} \right) s_c + \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{4} + \sum_c \frac{h_c}{2} + d, \tag{A8}$$

where $s = (s_1, \ldots, s_n)^T$. Therefore, this quadratic form over spin variables can be written as:

$$\begin{aligned}
\mathcal{L}^{\text{spin}}(s) &= \sum_{\substack{c',c\\c'\neq c}} Q^{\text{spin}}_{c',c} s_{c'} s_c + \sum_c h^{\text{spin}}_c s_c + d^{\text{spin}} \\
&= s^T Q^{\text{spin}} s + (h^{\text{spin}})^T s + d^{\text{spin}} \\
&= 2 \sum_{\substack{c',c\\c'< c}} Q^{\text{spin}}_{c',c} s_{c'} s_c + \sum_c h^{\text{spin}}_c s_c + d^{\text{spin}},
\end{aligned} \tag{A9}$$

with the components identified as:

$$Q^{\text{spin}}_{c'c} = \begin{cases} \frac{Q_{c'c}}{4} & \text{if } c' \neq c \\ 0 & \text{if } c' = c \end{cases} = \frac{Q_{c'c}}{4} \tag{A10}$$

$$h^{\text{spin}}_c = - \sum_{\substack{c'\\c'\neq c}} \frac{Q_{c',c}}{2} - \frac{h_c}{2} = - \sum_{c'} \frac{Q_{c',c}}{2} - \frac{h_c}{2} \tag{A11}$$

$$d^{\text{spin}} = \sum_{\substack{c',c\\c'\neq c}} \frac{Q_{c',c}}{4} + \sum_c \frac{h_c}{2} + d = \sum_{c',c} \frac{Q_{c',c}}{4} + \sum_c \frac{h_c}{2} + d. \tag{A12}$$

In each case, the second form results from the fact that the diagonal entries of Q are zero, so summing without excluding $c' = c$ makes no difference.

Alternatively, one may define the spin transformation as:

$$y_c = \frac{1 + s_c}{2}, \quad \forall c \in \{1, \cdots, n\}, \quad \text{(alternative formulation)} \tag{A13}$$

which maps the binary value $y_c = 0$ to the spin value $s_c = -1$, and $y_c = 1$ to $s_c = 1$. In this case, the expressions for Q^{spin}, h^{spin}, and d^{spin} remain as given above, while the sign of h^{spin} is reversed, namely:

$$h_c^{\text{spin}} = \sum_{c'} \frac{Q_{c',c}}{2} + \frac{h_c}{2}, \qquad \text{(alternative formulation)} \tag{A14}$$

Appendix B: Derivation of the QUBO Objective Function for Portfolio Optimization

This appendix provides a detailed mathematical derivation for converting the portfolio optimization objective function into the Quadratic Unconstrained Binary Optimization (QUBO) form. The initial objective function is formulated to minimize the squared deviation of portfolio characteristics from their specified targets, given by:

$$\min \sum_{l \in L} \sum_{j \in J} \rho_j \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} x_c - K_{l,j}^{\text{target}} \right)^2 . \tag{B1}$$

Here, the variables and parameters are defined as per the problem specification. The initial model includes continuous variables x_c, representing the quantity of each asset c. To create a binary objective function, binary variables $y_c \in \{0, 1\}$ are introduced, where $y_c = 1$ if asset c is included in the portfolio and $y_c = 0$ otherwise. The continuous variable x_c is then expressed in terms of y_c as follows:

$$x_c = a_c y_c, \quad a_c = \frac{m_c + \min\{M_c, i_c\}}{2\delta_c}, \tag{B2}$$

where the coefficient a_c is a pre-determined constant representing the effective quantity of asset c when selected. Substituting Eq. (B2) into Eq. (B1), the squared term within the summation can be expanded as:

$$\begin{aligned} \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} x_c - K_{l,j}^{\text{target}} \right)^2 = \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} a_c y_c - K_{l,j}^{\text{target}} \right)^2 = \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} a_c y_c \right)^2 \\ - 2K_{l,j}^{\text{target}} \left(\sum_{c \in \mathbb{K}_l} \beta_{c,j} a_c y_c \right) + \left(K_{l,j}^{\text{target}} \right)^2 \end{aligned} \tag{B3}$$

The quadratic term $\left(\sum_{c\in\mathbb{K}_l}\beta_{c,j}a_c y_c\right)^2$ in Eq. (B3) can be further expanded by separating its diagonal ($c'=c$) and off-diagonal ($c'\neq c$) components:

$$\begin{aligned}\left(\sum_{c\in\mathbb{K}_l}\beta_{c,j}a_c y_c\right)^2 &= \left(\sum_{c'\in\mathbb{K}_l}\beta_{c',j}a_{c'}y_{c'}\right)\left(\sum_{c\in\mathbb{K}_l}\beta_{c,j}a_c y_c\right) = \sum_{c',c\in\mathbb{K}_l}\beta_{c',j}a_{c'}\beta_{c,j}a_c y_{c'}y_c \\ &= \sum_{\substack{c',c\in\mathbb{K}_l\\ c'\neq c}}\beta_{c',j}a_{c'}\beta_{c,j}a_c y_{c'}y_c + \sum_{c\in\mathbb{K}_l}(\beta_{c,j}a_c)^2 y_c^2 \\ &= \sum_{\substack{c',c\in\mathbb{K}_l\\ c'\neq c}}\beta_{c',j}a_{c'}\beta_{c,j}a_c y_{c'}y_c + \sum_{c\in\mathbb{K}_l}(\beta_{c,j}a_c)^2 y_c.\end{aligned} \tag{B4}$$

In the final term of the preceding expression, the identity $y_c^2 = y_c$ is used, which holds because y_c is a binary variable. With the expression above, Eq. (B3) can be rewritten as:

$$\begin{aligned}\left(\sum_{c\in\mathbb{K}_l}\beta_{c,j}x_c - K_{l,j}^{\text{target}}\right)^2 = &\sum_{\substack{c',c\in\mathbb{K}_l\\ c'\neq c}}\beta_{c',j}a_{c'}\beta_{c,j}a_c y_{c'}y_c \\ &+ \sum_{c\in\mathbb{K}_l}\left[(\beta_{c,j}a_c)^2 - 2K_{l,j}^{\text{target}}\beta_{c,j}a_c\right]y_c + \left(K_{l,j}^{\text{target}}\right)^2\end{aligned} \tag{B5}$$

Substituting Eq. (B5) into the expression in Eq. (B1) , the objective function to be minimized can be written as:

$$\begin{aligned}\mathcal{L}^{\text{obj}}(y) = &\sum_{l\in L}\sum_{j\in J}\rho_j\left(\sum_{c\in\mathbb{K}_l}\beta_{c,j}x_c - K_{l,j}^{\text{target}}\right)^2 = \sum_{l\in L}\sum_{j\in J}\rho_j\sum_{\substack{c',c\in\mathbb{K}_l\\ c'\neq c}}\beta_{c',j}a_{c'}\beta_{c,j}a_c y_{c'}y_c \\ &+ \sum_{l\in L}\sum_{j\in J}\rho_j\sum_{c\in\mathbb{K}_l}\left[(\beta_{c,j}a_c)^2 - 2K_{l,j}^{\text{target}}\beta_{c,j}a_c\right]y_c + \sum_{l\in L}\sum_{j\in J}\rho_j\left(K_{l,j}^{\text{target}}\right)^2 \\ &= \sum_{\substack{c',c\\ c'\neq c}}\left(\sum_{l\in L}\sum_{j\in J}\rho_j\mathbb{I}_{[c',c\in\mathbb{K}_l]}\beta_{c',j}a_{c'}\beta_{c,j}a_c\right)y_{c'}y_c \\ &+ \sum_{c}\left(\sum_{l\in L}\sum_{j\in J}\rho_j\mathbb{I}_{[c\in\mathbb{K}_l]}\left[(\beta_{c,j}a_c)^2 - 2K_{l,j}^{\text{target}}\beta_{c,j}a_c\right]\right)y_c + \sum_{l\in L}\sum_{j\in J}\rho_j\left(K_{l,j}^{\text{target}}\right)^2\end{aligned} \tag{B6}$$

In the expression above, $\mathbb{I}_{[.]}$ is the indicator function, which is defined as 1 if the condition in the argument is true, and 0 otherwise. The variable y represents a column vector whose entries are the binary variables y_c. Thus, the objective function can be expressed in QUBO form as follows:

$$
\begin{aligned}
\mathcal{L}^{\mathrm{obj}}(y) &= \sum_{\substack{c',c \\ c' \neq c}} Q^{\mathrm{obj}}_{c',c} y_{c'} y_c + \sum_c h^{\mathrm{obj}}_c y_c + d^{\mathrm{obj}} \\
&= y^T Q^{\mathrm{obj}} y + (h^{\mathrm{obj}})^T y + d^{\mathrm{obj}} \\
&= 2 \sum_{\substack{c',c \\ c' < c}} Q^{\mathrm{obj}}_{c',c} y_{c'} y_c + \sum_c h^{\mathrm{obj}}_c y_c + d^{\mathrm{obj}}
\end{aligned} \tag{B7}
$$

Here, h^{obj} is a column vector of the same dimension as y whose entries are the linear coefficients h^{obj}_c. The final expression of Eq. (B7) is derived by utilizing the symmetry of the matrix Q^{obj} (i.e., $Q^{\mathrm{obj}}_{c'c} = Q^{\mathrm{obj}}_{c,c'}$), which allows the quadratic term to be written as a sum over its upper triangle. The components of this formulation are identified as:

$$
Q^{\mathrm{obj}}_{c'c} = \begin{cases} \sum_{l \in L} \sum_{j \in J} \rho_j \mathbb{I}_{[c',c \in \mathbb{K}_l]} \beta_{c',j} a_{c'} \beta_{c,j} a_c & \text{if } c' \neq c \\ 0 & \text{if } c' = c \end{cases}, \tag{B8}
$$

$$
h^{\mathrm{obj}}_c = \sum_{l \in L} \sum_{j \in J} \rho_j \mathbb{I}_{[c \in \mathbb{K}_l]} \left[(\beta_{c,j} a_c)^2 - 2 K^{\mathrm{target}}_{l,j} \beta_{c,j} a_c \right], \quad d^{\mathrm{obj}} = \sum_{l \in L} \sum_{j \in J} \rho_j \left(K^{\mathrm{target}}_{l,j} \right)^2. \tag{B9}
$$

Appendix C: Derivation of the QUBO Penalty Function for Portfolio Optimization

This appendix presents the mathematical derivation for converting linear inequality constraints into an equivalent penalty term in the QUBO objective function for the portfolio optimization problem. The objective function for this problem was formulated in Appendix B (see Eqs. B7, B8, and B9). The complete model, formulated as a quadratic optimization problem with linear inequality constraints, incorporates constraints such as limits on the number of assets in the portfolio, residual cash flow requirements, and bounds on various risk characteristics. The following sections provide the specific QUBO penalty term used to represent these constraints, enabling them to be incorporated into the main objective function to yield a single unified objective for the portfolio optimization problem.

The first constraint limits the total number of bonds selected in the portfolio to a maximum value, N. This constraint,

$$
\sum_{c \in C} y_c \leq N,
$$

can be rewritten as:

$$\sum_{c \in C} -y_c \geq -N. \tag{C1}$$

The second set of constraints involves the residual cash flow. This two-sided constraint,

$$\frac{\text{minRC}}{MV^b} \leq \sum_{c \in C} \frac{p_c \delta_c}{100 MV^b} x_c \leq \frac{\text{maxRC}}{MV^b},$$

is converted to binary variables by substituting $x_c = a_c y_c$. This results in two lower-bound inequality constraints:

$$\sum_{c \in C} \frac{p_c \delta_c}{100 MV^b} a_c y_c \geq \frac{\text{minRC}}{MV^b}, \tag{C2}$$

and

$$\sum_{c \in C} -\frac{p_c \delta_c}{100 MV^b} a_c y_c \geq -\frac{\text{maxRC}}{MV^b}. \tag{C3}$$

The min/max characteristic value constraints for each characteristic j in each group l are given by:

$$b_{l,j}^{\text{low}} \leq \sum_{c \in \mathbb{K}_l} \frac{p_c \delta_c}{100 MV^b} \beta_{c,j} x_c \leq b_{l,j}^{\text{up}}, \quad \forall j \in J, l \in L.$$

By substituting $x_c = a_c y_c$, these are also expressed as two lower-bound inequality constraints:

$$\sum_{c \in \mathbb{K}_l} \frac{p_c \delta_c}{100 MV^b} \beta_{c,j} a_c y_c \geq b_{l,j}^{\text{low}}, \quad \forall j \in J, l \in L, \tag{C4}$$

and

$$\sum_{c \in \mathbb{K}_l} -\frac{p_c \delta_c}{100 MV^b} \beta_{c,j} a_c y_c \geq -b_{l,j}^{\text{up}}, \quad \forall j \in J, l \in L. \tag{C5}$$

All of these constraints are of the general form:

$$\sum_{c} A_{i,c} y_c \geq b_i, \quad A_{i,c} \in \mathcal{R}, \quad b_i \in \mathcal{R}, \tag{C6}$$

which allows them to be cast into a matrix form:

$$Ay \geq b, \text{or } Ay - b \geq 0. \tag{C7}$$

Here, y is a binary vector, each row of matrix A represents a constraint, and the column vector b contains the bounds.

A standard approach to enforce linear inequality constraints like $Ay \geq b$ is to introduce a squared hinge loss penalty term into the objective function, which converts the problem into an unconstrained one. The penalty is defined as:

$$\mathcal{L}^{\text{penalty}}(y) = \text{penalty} \times \sum_i \max(b_i - A_i y, 0)^2, \tag{C8}$$

where A_i, the i-th row of the matrix A, corresponds to the i-th constraint. This formulation ensures that only violated constraints—where $b_i - A_i y_i > 0$—contribute to the penalty, with the squaring of the term magnifying the cost of more severe violations. Constraints that are satisfied, where $b_i - A_i y_i \leq 0$, result in a zero penalty. The penalty scalar is a hyperparameter that governs the stiffness of the constraints, allowing a trade-off between satisfying the constraints and optimizing the primary objective function.

In this formulation, the penalty parameter is determined by first calculating the sum of all positive elements (max_obj) and negative elements (min_obj) of the QUBO matrix Q^{obj}. The penalty coefficient is then set to 1.1 times the difference between these sums:

$$\text{penalty} = (\text{max\_obj} - \text{min\_obj}) \times 1.1 \tag{C9}$$

By setting the penalty in this manner, it is ensured that violations of the constraints are discouraged relative to the overall objective scale, providing consistent enforcement of constraints within the unconstrained QUBO framework. The final objective function is the sum of the main objective and the penalty function:

$$\mathcal{L}^{\text{total}}(y) = \mathcal{L}^{\text{obj}}(y) + \mathcal{L}^{\text{penalty}}(y) \tag{C10}$$

[1] M. Sciorilli, L. Borges, T. L. Patti, D. García-Martín, G. Camilo, A. Anandkumar, and L. Aolita, Nature Commun. **16**, 476 (2025), arXiv:2401.09421 [quant-ph].