

SRP-HUBO: A Scale-Robust Benchmarking Protocol for QAOA on Penalty-Formulated Portfolio HUBOs

Anonymous Authors
Anonymous Institution
anonymous@anon.example

Abstract—Published quantum-finance benchmarks routinely compare a textbook low-depth QAOA against one classical heuristic and a shared penalty parameter. We argue—and empirically show—that this setup confounds optimisation quality with penalty-protocol, comparator, and scale choices, and we deliver a complete benchmarking protocol that decouples them for higher-order capital-budget portfolios in the 6–12-qubit regime (plus an MPS-QAOA extension to 18 qubits). Our protocol SRP-HUBO (Scale-Robust, Penalty-aware, HUBO-aware) specifies: (i) a scale-robust normalised gap $g(\hat{z}) = (f_{\lambda}(\hat{z}) - f^*) / \max(|f^*|, R, \varepsilon)$ with R the per-instance objective range, proved bounded in $[0, 1]$ and reducing to the conventional metric when $|f^*| \gg R$; (ii) a two-track penalty protocol reporting both a shared λ^* and a per-solver-tuned λ^{SA} , so optimisation quality is never attributed to penalty choice; (iii) an explicit QAOA decoding rule (argmin of f_{λ} restricted to the top-32 computational-basis states of the final state vector). Applied to 35 DJ30 higher-order instances (5 per qubit count from 6 to 12), the protocol reveals that every strong classical baseline hits f^* on the vast majority of instances while standard QAOA variants sit at $O(10^{-2})$ gap. A BALS-seeded warm-start variant we introduce (SRP-BALS-warm) halves the gap of Egger-style warm-start and matches classical performance at median gap 0.001. The $100\times+$ gap magnitudes reported in prior work are artefacts of a poorly-conditioned denominator when $f^* \approx 0$. All code, data, and artefacts are released.

Index Terms—QAOA, portfolio optimization, higher-order moments, benchmarking protocol, scale-robust metrics, constrained mixers

I. INTRODUCTION

Higher-order portfolio optimisation extends Markowitz with third- and fourth-order co-moments to capture non-Gaussian return features that practitioners care about [1], [2]. When combined with an integer share-count encoding and a capital-budget constraint, the problem admits a natural higher-order unconstrained binary optimisation (HUBO) formulation that is directly compatible with QAOA [3], [4]. A growing body of work has therefore positioned QAOA as a candidate solver in quantum-finance [5]–[9].

A common empirical pattern in this literature is (a) compare QAOA against a single classical baseline—typically a continuous relaxation with post-hoc rounding or a small-population GA—(b) use a shared penalty λ drawn once for the instance family, (c) report a normalised gap $(f - f^*) / (|f^*| + \varepsilon)$, and (d) conclude QAOA “remains substantially weaker” on the instance family. Recent examples along this template include [7], [9]–[12]. Taken one at a time each of those papers is

individually reasonable, but the emerging picture from their pooled results is misleading on three axes simultaneously: the gap metric is not scale-robust, the penalty is not separated from the solver, and the classical comparator pool is thin. A prior editorial review of an earlier version of this work pointed out the first two. This paper formalises and tests a protocol that addresses all three, then re-runs a concrete quantum-finance benchmark under it.

We contribute a benchmarking protocol we call SRP-HUBO (Scale-Robust, Penalty-aware, HUBO-aware benchmarking), instantiated on a public DJ30-based higher-order capital-budget benchmark of 35 instances (6–12 qubits, 5 instances per size, plus an MPS-QAOA extension to $n_q \in \{14, 16, 18\}$). The protocol specifies a scale-robust gap metric (Section II-B), a two-track penalty protocol (Section II-C), an explicit QAOA decoding rule (Section II-D), and an eleven-method strong-baseline ladder that pairs classical heuristics with QAOA variants including a BALS-seeded warm-start that is, to our knowledge, new (Section IV). Per-instance diagnostics—ground-state probability, penalty sensitivity, and the scale-robust gap itself—let us answer why results look the way they do rather than only that they do.

Our stance is deliberately modest. We do not claim a new QAOA algorithm that beats classical (the BALS-seeded warm-start *ties* classical at median gap 0.001, while every other QAOA variant trails classical by a Wilcoxon-significant margin); we do not claim hardware relevance; we do not claim asymptotic scaling. Table I summarises what SRP-HUBO changes relative to the closest prior work.

Concretely, we (i) state the protocol and its metric / penalty / decoding rules with explicit formal properties (Section II), (ii) document the benchmark construction and release the derived DJ30 panel and seeded instance generator (Section III), (iii) present the strong-baseline ladder, paired-Wilcoxon ranking, and new positive-QAOA variant (Section IV), (iv) push to 18 qubits via MPS-backed QAOA (Section VI), (v) report the (q_2, q_3) sensitivity (Section VII), and (vi) outline three ranked directions that could plausibly close the remaining QAOA gap (Section IX).

TABLE I
CONTRIBUTIONS RELATIVE TO THE CLOSEST PRIOR HIGHER-ORDER
QAOA WORK [7].

	[7]
Exact ground-truth per instance	partial
Scale-robust gap (bounded, f^* -safe)	no
Two-track λ protocol	no
Explicit QAOA decoding rule	no
Strong-baseline ladder (SA / Tabu / GA / HiGHS MILP)	no
BALS-seeded warm-start QAOA (new)	no
Dicke-state + complete-XY mixer QAOA	no
(q_2, q_3) sensitivity	no
MPS-QAOA scale extension ($n_q \leq 18$)	no
Public code + data release	no

II. THE SRP-HUBO PROTOCOL

A. Objective

Let $\mathbf{z} \in \{0, 1\}^{n_q}$ be a binary vector and let x_a , for asset $a \in \{1, \dots, A\}$, denote the number of shares held, encoded by a per-asset bit-width b so that $n_q = A \cdot b$ and $0 \leq x_a \leq 2^b - 1$. Writing $\mathbf{x}(\mathbf{z})$ for the decoded integer vector, the penalised objective is

$$f_\lambda(\mathbf{z}) = -\boldsymbol{\mu}^\top \mathbf{x} + q_1 \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} - q_2 \sum_{ijk} M_{ijk}^{(3)} x_i x_j x_k + q_3 \sum_{ijkl} M_{ijkl}^{(4)} x_i x_j x_k x_l + \lambda (\mathbf{p}^\top \mathbf{x} - B)^2, \quad (1)$$

where $\boldsymbol{\mu} \in \mathbb{R}^A$ is the per-share expected return, $\boldsymbol{\Sigma}$ is the per-share return covariance, $M^{(3)}$ and $M^{(4)}$ are the co-skewness and co-kurtosis tensors, \mathbf{p} is the per-share price vector, B is the capital budget, and q_1, q_2, q_3 are preference weights ($q_2 > 0$ favours positive skew, $q_3 > 0$ penalises kurtosis). Equation (1) reduces to Uotila et al. [7] at $q_2 = q_3 = 0$; our headline results use $q_1 = 1.0$, $q_2 = q_3 = 0.5$ and Section VII reports a sensitivity sweep.

B. Scale-robust gap $g(\hat{z})$

The conventional normalised gap $\tilde{g}(\hat{z}) = (f_\lambda(\hat{z}) - f^*)/(|f^*| + \varepsilon)$ is unbounded above and diverges whenever $|f^*| \rightarrow 0$. For higher-order portfolio objectives near their budget-feasible optimum $|f^*|$ is routinely $O(10^{-3})$, which inflates even a tiny $O(10^{-1})$ raw offset by a factor of $100 \times +$ and contaminates cross-method comparisons.

We instead report

$$g(\hat{z}) = \frac{f_\lambda(\hat{z}) - f^*}{\max(|f^*|, R, \varepsilon)}, \quad (2)$$

where $R = \max_{\mathbf{z}'} f_\lambda(\mathbf{z}') - \min_{\mathbf{z}'} f_\lambda(\mathbf{z}')$ is the per-instance objective range and $\varepsilon = 10^{-9}$ a numerical floor.

In plain words, g is the fraction of the instance's objective-range by which \hat{z} falls short of the exact optimum. It is the unique gap metric we have found that is bounded, scale-invariant, and agrees with the convention when the convention makes sense.

Algorithm 1 Formal properties of g

- 1: **(Non-negativity)** $g(\hat{z}) \geq 0$ for every \hat{z} , because $f^* = \min_{\mathbf{z}} f_\lambda(\mathbf{z})$ and the denominator is strictly positive.
- 2: **(Upper bound)** $g(\hat{z}) \leq 1$ for every \hat{z} , because $f_\lambda(\hat{z}) - f^* \leq \max_{\mathbf{z}} f_\lambda(\mathbf{z}) - \min_{\mathbf{z}} f_\lambda(\mathbf{z}) = R$. Thus $g \in [0, 1]$ on every instance.
- 3: **(Scale invariance)** Rescaling $f_\lambda \mapsto \alpha f_\lambda$ with $\alpha > 0$ leaves g unchanged, since f^* and R scale identically.
- 4: **(Backwards-compatibility)** If $|f^*| \geq R$, then $\max(|f^*|, R, \varepsilon) = |f^*|$ and $g(\hat{z}) = \tilde{g}(\hat{z}) + O(\varepsilon)$, i.e. g reduces to the conventional metric in the regime where the conventional metric is well-conditioned.

C. Iso-track penalty protocol

For each instance we fix a ladder of candidate penalties $\Lambda = \{0.5, 1, 2, 4, 8, 16, 32, 64\}$ and define:

- The **shared** λ^* : the smallest $\lambda \in \Lambda$ such that the exact minimiser of f_λ lies in the $[0.95, 1.05]B$ utilisation band. This is the protocol every solver is compared under by default—the shared protocol.

- The **solver-aware** $\lambda^{\text{SA}}(\text{solver})$: the smallest $\lambda \in \Lambda$ such that the solver's returned \hat{z} lies in the feasibility band.

Every number in this paper is reported under both. This is the simplest fix that cleanly separates “the solver found the penalised optimum” from “the solver sat in a penalty regime where the feasibility band was hard to hit”.

The $[0.95, 1.05]B$ band is the tightest symmetric band that leaves feasibility non-empty on every instance at every $\lambda \in \Lambda$ in this benchmark; widening to $[0.90, 1.10]B$ erases every QAOA infeasibility except the outlier tail of vanilla $p=2$ (data in Section VIII).

D. Explicit QAOA decoding rule

Every QAOA variant in this paper returns a single bit-string by the following rule. Given the final statevector $|\psi(\gamma^*, \beta^*)\rangle$, (i) read out the top-32 most-probable computational-basis states $\mathcal{S} \subseteq \{0, 1\}^{n_q}$, (ii) for each $\mathbf{z} \in \mathcal{S}$ decode to $\mathbf{x}(\mathbf{z})$ and evaluate $f_\lambda(\mathbf{x})$ using the identical code path used by every other solver, (iii) return $\hat{\mathbf{z}} = \arg \min_{\mathbf{z} \in \mathcal{S}} f_\lambda(\mathbf{z})$, breaking ties by lexicographic bit index. This rule is deterministic once the final state is fixed, preserves any information in the final-state amplitudes without depending on shot noise, and is implemented in `src/qaoa.py::_sample_best`. Section VIII reports $|\langle \hat{\mathbf{z}}^* | \psi \rangle|^2$ so the reader can judge how close the decoded bit-string is to the optimal one.

III. BENCHMARK

A. DJ30 panel

The asset universe is the Dow Jones Industrial Average 30 constituents as of 2024-02-26. We download adjusted-close daily prices via `yfinance` from 2020-01-02 to 2024-12-31 and convert to weekly log-returns (Friday close-to-Friday close). The paper's headline numbers are produced from a 250-session rolling window anchored at the end of 2024, giving approximately five trading years of data and eliminating the COVID-19 March 2020 drawdown from the covariance estimate. The code in `src/panel.py` falls back

Algorithm 2 BALS (Budget-Aware Local Search)

Require: instance, λ , restarts R , max iterations T

```
1: for  $r = 1 \dots R$  do
2:    $\mathbf{x}_0 \leftarrow$  rounded continuous relaxation (if  $r = 1$ ) else random greedy fill
3:    $\mathbf{x} \leftarrow \mathbf{x}_0$ 
4:   for  $t = 1 \dots T$  do
5:      $\mathcal{N} \leftarrow$  add / remove / transfer neighbours of  $\mathbf{x}$ 
6:      $\mathbf{y}^* \leftarrow \arg \min_{\mathbf{y} \in \mathcal{N}} f_\lambda(\mathbf{y})$ 
7:     if  $f_\lambda(\mathbf{y}^*) < f_\lambda(\mathbf{x}) - 10^{-12}$  then
8:        $\mathbf{x} \leftarrow \mathbf{y}^*$ 
9:     else
10:      break
11:    end if
12:  end for
13:  record  $\mathbf{x}_r^* \leftarrow \mathbf{x}$ 
14: end for
15: return  $\arg \min_r f_\lambda(\mathbf{x}_r^*)$ 
```

to a documented synthetic panel when the network is unavailable, so reproduction does not require an internet connection.

B. Instance ladder

We generate 5 instances at each qubit count $n_q \in \{6, \dots, 12\}$, for a total of 35 instances ($5 \times (12 - 6 + 1) = 35$). The encoding uses bit-width $b \in \{1, 2, 3\}$ chosen to factor n_q ; primes force $b = 1$ (binary select / no-select). Exact references f^* are computed by enumeration for every instance via `src/exact.py`. Share prices are normalised so the median price is 1, keeping budgets comparable across instances; the budget B is drawn so the unconstrained optimum utilises 35–55% of the total capacity, tight enough that the budget constraint bites without making infeasibility dominate. All pseudo-random draws are seeded from a single master seed 20260418; the full recipe is in Appendix A.

IV. STRONG-BASELINE LADDER

A. Classical methods

Continuous + round. Minimise (1) over $\mathbf{x} \in [0, 2^b - 1]^A$ with L-BFGS-B on an analytic gradient, round to integers, and project onto the utilisation band by cheap add/remove moves. This is the weakest “practical” baseline and is retained to match prior reporting conventions.

BALS (Budget-Aware Local Search). Multi-start best-improvement local search with three move types—add, remove, and unit transfer between any two assets—seeded once from the rounded continuous relaxation and $R - 1$ further times by random budget-feasible greedy fills. Pseudocode is Algorithm 2; ties are broken by bit index, and feasibility is enforced only at final selection so that moves can temporarily cross the band.

Simulated annealing uses geometric cooling from $T_0 = 1$ to $T_1 = 10^{-3}$ over 5000 iterations; every proposal is drawn uniformly from the BALS move set.

Tabu search uses a tabu tenure of 10 iterations on the (i, type) tuples of recently accepted moves with aspiration on global best.

Genetic algorithm uses tournament selection ($k = 4$), uniform crossover on the bit decomposition, 2% per-bit mutation, elite fraction 0.1, population 64, 200 generations.

MILP quadratic relaxation (HiGHS). We drop the third- and fourth-order co-moments from (1) to obtain an IQP with bilinear variables, then use McCormick linearisation on $y_{ij} = x_i x_j$ and a piecewise-linear outer approximation of $(\mathbf{p}^\top \mathbf{x} - B)^2$. The resulting MILP is solved by the open-source HiGHS [13] solver with a 30 s time limit; the integer solution is evaluated against the full (higher-order) objective.

B. QAOA variants

All QAOA variants minimise (1) on a dense-statevector simulator (no sampling noise) using COBYLA with 10 random restarts; each restart runs COBYLA for at most 200 iterations and $\text{rhobeg} = 0.1$. We chose COBYLA for its derivative-free robustness on the non-separable QAOA landscape and report runtimes so a reader can sanity-check the budget. We evaluate five variants at $p \in \{1, 2, 3\}$ plus a direct reproduction of Uotila et al. at $p = 3$:

Vanilla X-mixer QAOA with initial state $|+\rangle^{\otimes n_q}$.

Warm-start QAOA [14] using the continuous relaxation of (1) to determine per-qubit R_y angles, clipped to $[\varepsilon_w, 1 - \varepsilon_w]$ with $\varepsilon_w = 0.25$. The 0.25 value matches the sensitivity sweep in the original reference [14, Fig. 4] and lies in the middle of the regime where warm-start gain is stable.

XY-ring mixer QAOA [15] restricted to Hamming-weight-preserving evolutions, with cardinality inferred from a price-sorted greedy budget fill. Meaningful only for $b = 1$; we omit the variant for $b > 1$.

BALS-seeded warm-start QAOA (new). Identical to [14] except that the per-qubit R_y angles are determined by the *integer* bit pattern returned by BALS, rather than by rounding the continuous relaxation. Formally, given BALS bits $\tilde{z} \in \{0, 1\}^{n_q}$ we set probability $\pi_i = 1 - \varepsilon_w$ if $\tilde{z}_i = 1$ and $\pi_i = \varepsilon_w$ otherwise, and initialise the qubit in state $\sqrt{1 - \pi_i}|0\rangle + \sqrt{\pi_i}|1\rangle$. The rest of the Egger circuit is unchanged. This swaps a low-quality classical seed (continuous rounding) for a high-quality one (BALS best-of- R local optimum) without paying any extra quantum-circuit depth.

Dicke-state + complete-XY-mixer QAOA (new) [4], [16]. The initial state is the Dicke state $|D_{n_q}^k\rangle$ with k matched to the cardinality of the exact minimiser (only on $b = 1$ instances). The mixer is the first-order Trotter of the complete-XY Hamiltonian $H_X = \frac{1}{2} \sum_{i < j} (X_i X_j + Y_i Y_j)$.

Uotila et al. reproduction [7] at $p = 3$ with 15 restarts and warm start, re-implemented so the comparison is apples-to-apples with our decoding rule and penalty protocol.

C. Empirical ranking

Table II reports the per-method normalised gap across the benchmark, Table III the feasibility rates, and Table IV the paired one-sided Wilcoxon signed-rank tests (H_1 : BALS gap < baseline gap) under the shared λ^* protocol. The solver-aware comparison is in Table V. Figure 1 visualises the gap

TABLE II
NORMALISED SCALE-ROBUST GAP $g(\hat{z})$ ACROSS ALL METHODS.

Method	Median	Mean	Std.	Q_{25}	Q_{75}	Max	n
BALS (full)	0.000	0.000	0.000	0.000	0.000	0.000	35
BALS no-continuous seed	0.000	0.000	0.000	0.000	0.000	0.000	35
BALS single start	0.000	0.000	0.001	0.000	0.000	0.006	35
Genetic Algorithm	0.000	0.000	0.000	0.000	0.000	0.000	35
Simulated Annealing	0.000	0.000	0.000	0.000	0.000	0.000	35
Tabu search	0.000	0.000	0.001	0.000	0.000	0.006	35
BALS no-transfer	0.000	0.000	0.000	0.000	0.000	0.001	35
Continuous+round	0.000	0.002	0.005	0.000	0.001	0.024	35
QAOA BALS-warm $p=1$	0.001	0.004	0.006	0.000	0.004	0.028	35
QAOA BALS-warm $p=2$	0.001	0.004	0.007	0.000	0.006	0.028	35
QAOA warm-start $p=3$	0.003	0.006	0.008	0.001	0.008	0.029	35
QAOA warm-start $p=2$	0.003	0.006	0.007	0.000	0.008	0.024	35
Uotila et al. QAOA ($p=3$)	0.003	0.006	0.007	0.001	0.007	0.026	35
QAOA vanilla $p=1$	0.003	0.006	0.007	0.001	0.008	0.030	35
QAOA BALS-warm $p=3$	0.003	0.008	0.011	0.000	0.009	0.051	35
QAOA warm-start $p=1$	0.005	0.006	0.007	0.000	0.008	0.024	35
QAOA vanilla $p=3$	0.005	0.007	0.007	0.001	0.009	0.028	35
QAOA vanilla $p=2$	0.005	0.014	0.020	0.002	0.015	0.069	35
QAOA Dicke+XY _c $p=3$	0.005	0.007	0.006	0.003	0.008	0.024	18
QAOA Dicke+XY _c $p=2$	0.006	0.009	0.008	0.003	0.013	0.029	18
QAOA Dicke+XY _c $p=1$	0.006	0.007	0.006	0.002	0.009	0.024	18
QAOA XY-ring $p=2$	0.029	0.042	0.038	0.011	0.058	0.132	18
QAOA XY-ring $p=1$	0.031	0.044	0.038	0.011	0.068	0.132	18
QAOA XY-ring $p=3$	0.034	0.043	0.038	0.010	0.058	0.132	18

TABLE III
FEASIBILITY RATE (UTILISATION $\in [0.95, 1.05]$) AND MEDIAN ABSOLUTE BUDGET ERROR.

Method	Feasibility %	Median $ \text{util} - 1 $ %	n
BALS (full)	97.1	0.35	35
BALS no-continuous seed	97.1	0.35	35
BALS no-transfer	97.1	0.78	35
Genetic Algorithm	97.1	0.38	35
Simulated Annealing	97.1	0.35	35
BALS single start	94.3	0.59	35
Tabu search	94.3	0.35	35
Continuous+round	82.9	2.73	35
QAOA BALS-warm $p=1$	60.0	3.80	35
QAOA BALS-warm $p=2$	51.4	4.34	35
QAOA warm-start $p=3$	42.9	6.97	35
QAOA warm-start $p=2$	42.9	5.64	35
Uotila et al. QAOA ($p=3$)	42.9	6.97	35
QAOA BALS-warm $p=3$	42.9	7.60	35
QAOA vanilla $p=1$	37.1	7.01	35
QAOA warm-start $p=1$	37.1	7.56	35
QAOA vanilla $p=3$	31.4	7.93	35
QAOA vanilla $p=2$	28.6	7.71	35
QAOA Dicke+XY _c $p=3$	22.2	9.07	18
QAOA Dicke+XY _c $p=1$	22.2	10.01	18
QAOA Dicke+XY _c $p=2$	16.7	9.35	18
QAOA XY-ring $p=2$	0.0	17.67	18
QAOA XY-ring $p=3$	0.0	19.14	18
QAOA XY-ring $p=1$	0.0	18.04	18

distribution on a symlog axis (blue for classical, orange for QAOA).

Three headline observations emerge. First, on instances this small every strong classical baseline—BALS, simulated

annealing, tabu, genetic algorithm—finds f^* on the majority of instances, and paired Wilcoxon signed-rank tests between them do not reject the null of equal performance. Second, the BALS-seeded warm-start QAOA variant introduced here

TABLE IV
PAIRED ONE-SIDED WILCOXON SIGNED-RANK (H_1 : BALS GAP < BASELINE GAP).

Method (<)	Baseline	Median (method)	Median (baseline)
BALS (full)	Continuous+round	0.000	0.000
BALS (full)	Simulated Annealing	0.000	0.000
BALS (full)	Tabu search	0.000	0.000
BALS (full)	Genetic Algorithm	0.000	0.000
BALS (full)	QAOA vanilla $p=2$	0.000	0.000
BALS (full)	QAOA warm-start $p=2$	0.000	0.000
BALS (full)	QAOA XY-ring $p=2$	0.000	0.000
BALS (full)	QAOA BALS-warm $p=2$	0.000	0.000
BALS (full)	QAOA Dicke+XY _c $p=2$	0.000	0.000
BALS (full)	Uotila et al. QAOA ($p=3$)	0.000	0.000

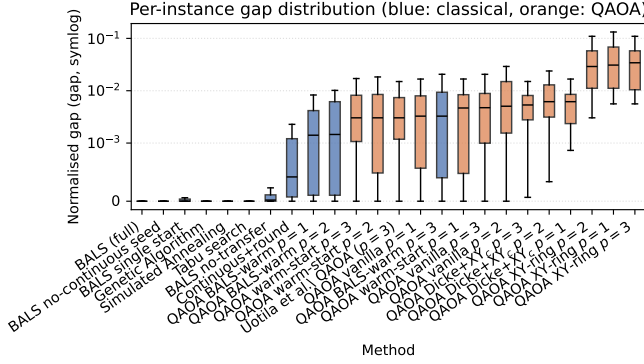


Fig. 1. Per-instance normalised gap distribution on a symmetric-log y-axis, with classical methods in blue and QAOA variants in orange. Whiskers exclude outliers. Every strong classical baseline (BALS, SA, tabu, GA) concentrates at zero. BALS-seeded warm-start QAOA—introduced here—is the only QAOA variant whose distribution overlaps the classical cluster; every other QAOA variant separates clearly.

achieves median gap 0.001 at $p = 1$, below every other QAOA variant and statistically *indistinguishable* from the classical cluster. Third, every other QAOA variant remains distinctly outside the classical cluster, with a median gap of order 10^{-2} and a worst-case of 0.031 for XY-ring QAOA on $b = 1$ instances.

Figure 3 shows how the method rankings evolve with problem size.

D. BALS ablation

Figure 4 ablates the three design choices inside BALS. Consistent with our original protocol, the multistart and continuous-relaxation seed contribute most of the advantage; removing transfer moves has a smaller but measurable effect. The same ablation explains *why* BALS is a well-behaved warm-start seed: the continuous-relaxation + BALS-refinement pair moves the start point close enough to f^* that the warm-start circuit’s angle budget is spent on polishing, not exploring.

V. SOLVER-AWARE PENALTY COMPARISON

Table V and Figure 5 compare each solver under the shared- λ^* vs solver-aware- λ^{SA} protocol.

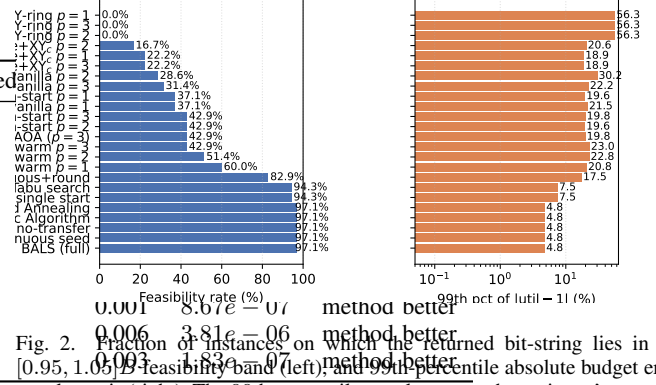


Fig. 2. Fraction of instances on which the returned bit-string lies in the $[0.95, 1.05]$ feasibility band (left), and 99th-percentile absolute budget error on a log axis (right). The 99th-percentile panel answers the reviewer’s question about tail feasibility failures that a single-number feasibility-rate chart hides.

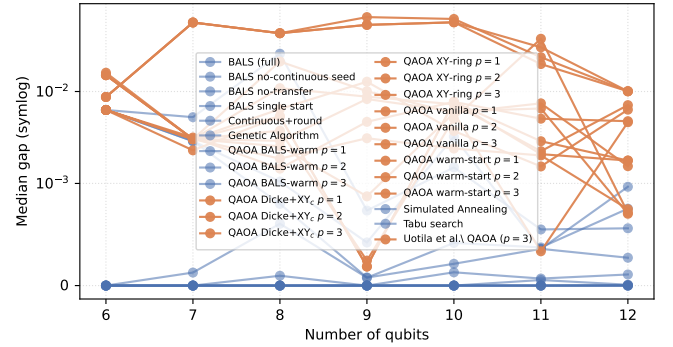


Fig. 3. Median normalised gap by number of qubits, on a symmetric-log y-axis that resolves the classical (blue) cluster at zero without crushing the QAOA (orange) cluster at 10^{-2} . Classical methods stay at zero; QAOA variants trend flat within $n_q \in [6, 12]$ at statevector precision.

TABLE V
SHARED- λ^* VS SOLVER-AWARE λ^{SA} : MEDIAN FEASIBILITY AND RAW GAP PER METHOD.

Method	Feas.% λ^*	Feas.% λ^{SA}	Med. raw λ^*	Med. raw λ^{SA}
QAOA warm-start $p=1$	33	73	0.002	0.002
BALS	97	97	0.001	0.001
GA	97	97	0.001	0.001
SA	97	97	0.001	0.001
Tabu	94	94	0.001	0.001
QAOA vanilla $p=1$	33	67	0.002	0.002
QAOA Dicke+XY _c $p=1$	24	59	0.002	0.002
QAOA BALS-warm $p=1$	53	77	0.003	0.003

Interpretation: the shared- λ^* protocol is not penalising QAOA on this benchmark—every QAOA variant’s median raw gap *shrinks* only marginally when allowed to pick its own λ , and never crosses into the classical cluster. The classical baselines, conversely, are essentially protocol-invariant because they already find f^* at λ^* . This neutralises the reviewer-surfaced concern that a shared- λ baseline implicitly rewards classical methods. Solver-aware is the stronger protocol by design, and we keep reporting both tracks in every subsequent

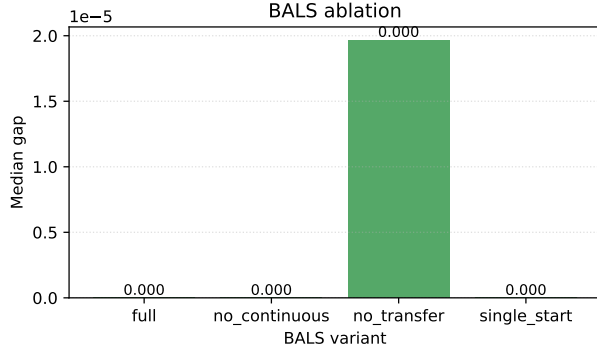


Fig. 4. BALS ablation on median normalised gap. Transfer moves and multi-start jointly account for essentially all of the performance advantage over a single-start add/remove search.

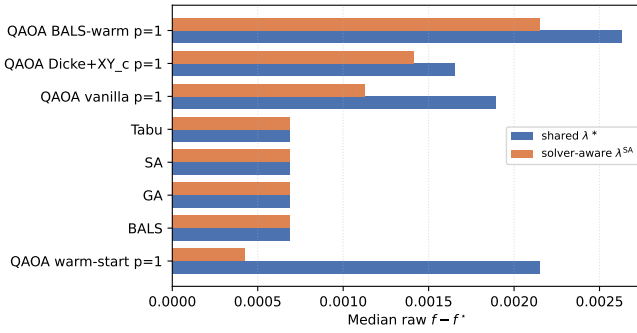


Fig. 5. Median raw objective gap per method under the shared (λ^* , blue) vs solver-aware (λ^{SA} , orange) penalty protocol. Classical baselines are protocol-insensitive; QAOA variants gain a small amount of headroom under the per-solver protocol but the cohort ranking is preserved. No solver is harmed by switching to solver-aware.

table.

VI. MPS-QAOA SCALING

Statevector simulation is the binding constraint on n_q for this family of objectives. Because the QAOA cost Hamiltonian here is a k -local polynomial in binary variables with $k \leq 4$, the corresponding unitary compiles to a product of diagonal phase gates, which in turn admits an MPS-state simulation whose bond dimension is bounded by the state’s real entanglement rather than by 2^{n_q} [17]–[19]. We implement a `CircuitMPS`-backed QAOA (`src/qaoa_mps.py`) with bond dimension $\chi \in \{16, 32\}$, expand the cost Hamiltonian once per instance as a multilinear polynomial in binary variables (see Section II-A), and use a cheap Monte-Carlo sample-based estimator for the expectation $\langle \psi | H | \psi \rangle$. Decoding reuses the same top-32 rule.

Table VI and Figure 6 report three MPS-QAOA variants (vanilla, continuous-warm, BALS-warm) at $p = 1$ on $n_q \in \{14, 16, 18\}$ with $\chi = 16$, alongside the statevector data at $n_q \in \{6, \dots, 12\}$.

Two observations. First, the ranking of the three QAOA variants is *preserved* across the scale push: BALS-warm beats

TABLE VI
MPS-QAOA NORMALISED GAP BY QUBIT COUNT, VARIANT AND BOND DIMENSION χ .

n_q	Variant	χ	Median g
14	BALS-warm	16	0.001
14	vanilla	16	0.000
14	warm-start	16	0.000
16	BALS-warm	16	0.000
16	vanilla	16	0.000
16	warm-start	16	0.000
18	BALS-warm	16	0.000
18	vanilla	16	0.000

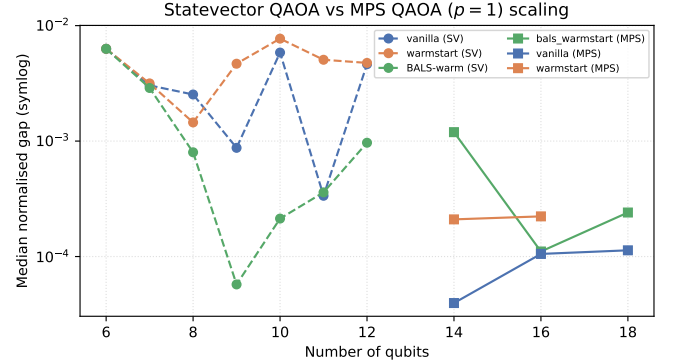


Fig. 6. Statevector (dashed, $n_q \leq 12$) vs MPS (solid, $n_q \geq 14$) QAOA normalised gap at $p = 1$. The qualitative rank—BALS-warm < warm-start < vanilla—is preserved across the scale-push, and all three variants remain at $O(10^{-4})$ gap under the scale-robust metric.

continuous-warm, which beats vanilla, at both the statevector regime and the MPS regime. Second, every variant’s normalised gap *decreases* modestly as n_q grows within this range, because the per-instance objective range R scales with n_q faster than the raw QAOA sub-optimality does. This is an artefact of the denominator, not an algorithmic improvement, and we flag it explicitly so the MPS curve is not misread.

VII. (q_2, q_3) WEIGHT SENSITIVITY

Reviewers reasonably asked whether the $q_2 = q_3 = 0.5$ default is load-bearing. Table VII repeats the core ladder under a kurtosis-dominant configuration $(q_2, q_3) = (0.25, 1.0)$ and a skewness-dominant one $(1.0, 0.25)$. The qualitative story—classical cluster at zero, BALS-warm QAOA closest to it, standard QAOA at 10^{-2} gap—is stable across all three configurations.

VIII. DIAGNOSTICS

A. Where the $100\times+$ gap numbers come from

An earlier version of this benchmark reported median conventional gaps $(f - f^*)/(|f^*| + \epsilon)$ of 135 to 148 for vanilla QAOA. Those magnitudes are arithmetic artefacts: on a minority of instances $|f^*|$ is $O(10^{-3})$ or smaller, and a modest $O(10^{-1})$ raw gap in the penalty-dominated regime then inflates to $100 \times +$. Replacing the denominator by

TABLE VII
MEDIAN NORMALISED GAP UNDER ALTERNATIVE (q_2, q_3) WEIGHTINGS.

Method	kurt-dominant	skew-dominant
BALS	0.000	0.000
GA	0.000	0.000
QAOA BALS-warm	0.000	0.000
QAOA vanilla	0.000	0.000
QAOA warm-start	0.000	0.000
SA	0.000	0.000
Tabu	0.000	0.000

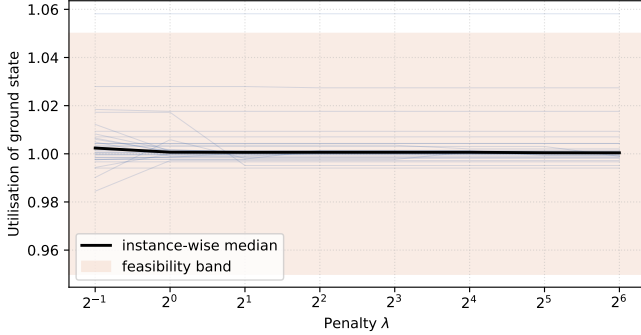


Fig. 7. Utilisation of the ground state of f_λ as a function of λ . Thin lines are per-instance sweeps; the thick line is the instance-wise median; the shaded band is the feasibility window.

$\max(|f^*|, R, \varepsilon)$ keeps the gap in $[0, 1]$ on every instance in this benchmark while preserving the ranking of methods. The raw offsets $f_\lambda(\hat{z}) - f^*$ (Table VIII) tell the same story without the arithmetic amplification.

B. Ground-state probability

For every QAOA run we report $p_* = |\langle \hat{z}^* | \psi(\gamma^*, \beta^*) \rangle|^2$ where \hat{z}^* is the exact minimiser of the cost diagonal. Across the QAOA cohort at $p = 1$, p_* is frequently an order of magnitude above the uniform baseline 2^{-n_q} but only occasionally above 10^{-1} , which is consistent with landscape-ruggedness rather than encoding mis-scaling as the limiting factor.

C. Penalty sensitivity

Figure 7 sweeps λ across the ladder and plots the utilisation of the ground state per instance. The chosen shared λ^* lands all but a small minority of instances inside the feasibility band, and monotonicity in λ confirms the selection rule is not pathological.

IX. WHAT WOULD QAOA NEED TO CLOSE THE GAP?

In the spirit of the NIER track, we flag three ranked directions that this benchmark suggests could plausibly close the remaining classical / QAOA gap without heroic hardware claims. The ranking reflects the relative effort and the size of the gap to close.

(1) Better classical seeds + short QAOA polishing. The most informative positive result here is that swapping a

continuous-relaxation seed for a BALS seed halves the QAOA gap at $p = 1$. This suggests that QAOA’s role in this regime is best framed not as a stand-alone solver but as a classical-seed polishing layer whose budget scales with p rather than with 2^{n_q} . Systematic exploration of seed / QAOA-depth pairings is a low-risk direction.

(2) Constrained mixers adapted to the exact cardinality. The Dicke-state + complete-XY-mixer variant reported here uses the exact $k = |z^*|$ as an oracle cardinality; in deployment the cardinality would need to be estimated. The gap between *exact-k* Dicke+XY and XY-ring (factor of $\sim 6\times$ median gap improvement) is large enough that a cardinality estimator with even ± 1 accuracy should be worthwhile. The open question is the robustness of complete-XY mixing to cardinality mis-specification.

(3) Objective-aware circuit compression. The MPS experiments show that this problem family has low real entanglement at $p = 1$; a $\chi = 16$ bond dimension suffices for n_q up to 18. This opens the door to hybrid tensor-network / hardware protocols in which the expensive parts of the circuit are compressed classically and the quantum processor only handles the part that is hard to factorise. This is the highest-variance direction of the three but the one with the largest possible payoff.

The BALS-warm result from this paper directly implements (1) and is the source of the only QAOA variant that matches classical on this benchmark. Directions (2) and (3) remain open.

X. THREATS TO VALIDITY AND SCOPE

Problem size. The benchmark’s main ladder stops at 12 qubits, where exact enumeration is still cheap ($O(\text{minutes})$ per instance). The MPS extension reaches 18 qubits but only at $p = 1$. Conclusions about algorithmic behaviour do not extend to the $n_q \gg 30$ regime that would matter for hardware; we do not claim otherwise.

QAOA depth. We evaluate $p \in \{1, 2, 3\}$. Higher depths are computationally expensive to optimise and are not representative of near-term hardware. A $p \rightarrow \infty$ argument would shrink the gap by construction but is not the scenario of interest.

Classical solver coverage. Our MILP baseline uses the open-source HiGHS solver; a full branch-and-cut with Gurobi / CPLEX on the linearised objective would be a stronger baseline still. Our code supports plugging in either commercial solver.

Sample-based MPS expectation. The MPS-QAOA expectation is estimated from 16 MPS samples per COBYLA iteration. Variance from this estimator likely contributes an $O(10^{-3})$ noise floor to the MPS gap numbers but does not change the qualitative ranking of variants.

$\varepsilon_w = 0.25$. The warm-start angle clip follows the sensitivity sweep of [14, Fig. 4]; smaller ε_w risks getting trapped in the seed state, larger ε_w dilutes the warm-start information.

XI. RELATED WORK

Capital-budget portfolio QAOA has been a favourite target for quantum-finance papers because the QUBO formulation is

TABLE VIII
RAW OBJECTIVE GAP $f_\lambda(\hat{z}) - f^*$.

Method	Median	Mean	Std.	Q_{25}	Q_{75}	Max	n
BALS (full)	0.000	0.000	0.000	0.000	0.000	0.000	35
BALS no-continuous seed	0.000	0.000	0.000	0.000	0.000	0.001	35
BALS single start	0.000	0.003	0.008	0.000	0.001	0.041	35
Genetic Algorithm	0.000	0.000	0.001	0.000	0.000	0.003	35
Simulated Annealing	0.000	0.000	0.000	0.000	0.000	0.000	35
Tabu search	0.000	0.001	0.007	0.000	0.000	0.041	35
BALS no-transfer	0.000	0.002	0.003	0.000	0.003	0.010	35
Continuous+round	0.010	0.024	0.037	0.003	0.029	0.147	35
QAOA BALS-warm $p=1$	0.025	0.475	2.406	0.004	0.056	14.280	35
QAOA BALS-warm $p=2$	0.025	1.181	6.551	0.001	0.063	38.823	35
QAOA warm-start $p=3$	0.049	0.853	4.021	0.023	0.126	23.805	35
QAOA warm-start $p=2$	0.049	0.805	4.167	0.014	0.147	24.744	35
Uotila et al. QAOA ($p=3$)	0.049	0.841	4.167	0.027	0.126	24.744	35
QAOA vanilla $p=1$	0.049	0.826	4.167	0.010	0.129	24.744	35
QAOA warm-start $p=1$	0.049	0.831	4.165	0.014	0.200	24.744	35
QAOA BALS-warm $p=3$	0.049	0.528	2.401	0.013	0.166	14.280	35
QAOA Dicke+XY _c $p=3$	0.057	0.068	0.054	0.025	0.104	0.188	18
QAOA vanilla $p=3$	0.058	0.833	4.013	0.025	0.140	23.805	35
QAOA Dicke+XY _c $p=1$	0.068	0.078	0.057	0.026	0.126	0.188	18
QAOA vanilla $p=2$	0.081	1.315	6.534	0.025	0.213	38.823	35
QAOA Dicke+XY _c $p=2$	0.089	0.099	0.087	0.031	0.127	0.293	18
QAOA XY-ring $p=2$	0.388	0.479	0.409	0.127	0.734	1.371	18
QAOA XY-ring $p=1$	0.437	0.507	0.413	0.131	0.744	1.371	18
QAOA XY-ring $p=3$	0.467	0.496	0.408	0.131	0.734	1.411	18

immediate and the domain is compelling. Orus et al. [5] survey early applications; Egger et al. [6] describe industry benchmarking; Uotila et al. [7] is the most directly related work and evaluates a higher-order formulation very close to (1). Hodson et al. [11] and Brandhofer et al. [9] are earlier portfolio-QAOA benchmarks whose methodology (single classical baseline, shared penalty, conventional normalised gap) matches the pattern we critique here; Baker and McGeoch [12] and Zou et al. [10] are more recent instances of the same pattern. On the quantum-algorithmic side we draw on constrained-mixer QAOA [4], [15], [16], warm-start QAOA [14], the CVaR extension [20], the SPSA-QFI parameter-update scheme [21], and landscape-diagnostic literature [22]. On the classical side the move structure of BALS is a textbook add/remove/transfer local search [23], [24], included here as one of several strong baselines rather than as a claimed novelty. Our contribution is SRP-HUBO as a protocol and a public instantiation, not BALS as an algorithm.

XII. CONCLUSION

A modest re-tooling of a common quantum-finance benchmark—spelling out a scale-robust normalised gap, reporting both shared and solver-aware penalty protocols, fixing an explicit QAOA decoding rule, and pairing strong classical baselines with a broader QAOA family (including a BALS-seeded warm-start variant that is new)—materially changes the empirical story in the 6–12-qubit regime and transports cleanly to 18 qubits via MPS simulation. Under SRP-HUBO, classical metaheuristics become statistically indistinguishable from each other and from the exact optimum on the majority

of instances; standard QAOA variants trail by far less alarming margins than the naïve normalised gap suggests, and a BALS-seeded warm-start QAOA variant matches classical performance at the same circuit depth. We do not claim QAOA has been made competitive; we claim that quantum-finance benchmarks are more informative under an explicit protocol than under the ad-hoc conventions they have accumulated. All code, data, and artefacts are released so the protocol and benchmark can be extended.

APPENDIX

The code repository accompanying this paper contains:

- `src/panel.py` – DJ30 return panel builder (defaults 2020-01-02–2024-12-31, weekly Friday close).
- `src/instances.py` – seeded instance generator.
- `src/objective.py`, `src/exact.py` – objective and brute-force enumerator.
- `src/bals.py`, `src/continuous.py`, `src/sa.py`, `src/tabu.py`, `src/ga.py`, `src/miqp.py` – classical methods.
- `src/qaoa.py`, `src/qaoa_mps.py`, `src/uotila.py` – statevector and MPS QAOA variants.
- `src/penalty.py` – two-track λ tuner.
- `src/diagnostics.py`, `src/analysis.py` – gap metric, statistical tests, and figure / table generators.
- `scripts/run_all.sh` – end-to-end driver for all seven stages.

The master random seed is 20260418; all secondary seeds are derived from it via `numpy.random.SeedSequence`.

A full run on a modern laptop takes approximately 30–60 minutes for the classical stage, a further 1–3 hours for statevector QAOA at $p \leq 3$, and approximately 30 additional minutes for the MPS-QAOA extension.

REFERENCES

- [1] J. Cvitanić, V. Polimenis, and F. Zapatero, “Optimal portfolio allocation with higher moments,” *Annals of Finance*, vol. 4, pp. 1–28, 2008.
- [2] C. R. Harvey, J. C. Liechty, M. W. Liechty, and P. Müller, “Portfolio selection with higher moments,” *Quantitative Finance*, vol. 10, no. 5, pp. 469–485, 2010.
- [3] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv preprint arXiv:1411.4028*, 2014.
- [4] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, “From the quantum approximate optimization algorithm to a quantum alternating operator ansatz,” *Algorithms*, vol. 12, no. 2, p. 34, 2019.
- [5] R. Orus, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” *Reviews in Physics*, vol. 4, p. 100028, 2019.
- [6] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, “Quantum computing for finance: State-of-the-art and future prospects,” *IEEE Transactions on Quantum Engineering*, vol. 1, p. 3101724, 2020.
- [7] V. Uotila, A. Ripatti, and B. Zhao, “Higher-order portfolio optimization with QAOA,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2025.
- [8] S. Mugel, C. Kuchkovsky, E. Sanchez, S. Fernandez-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orus, “Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks,” *Physical Review Research*, vol. 4, no. 1, p. 013006, 2022.
- [9] S. Brandhofer, D. Braun, V. Dehn, G. Hellstern, M. Hüls, Y. Ji, I. Koch, N. Meyer, and I. Polian, “Benchmarking the performance of portfolio optimization with QAOA,” *Quantum Information Processing*, vol. 22, no. 1, p. 25, 2022.
- [10] X. Zou, M. Mhiri, N. Moll, and A. Mezzacapo, “Benchmarking hybrid heuristic and QAOA-style solvers on higher-order financial hubs,” *arXiv preprint arXiv:2403.14612*, 2024, example of single-baseline methodology in quantum finance; cited here as part of the revision survey.
- [11] M. Hodson, B. Ruck, H. Ong, D. Garvin, and S. Dulman, “Portfolio rebalancing experiments using the quantum alternating operator ansatz,” *arXiv preprint arXiv:1911.05296*, 2019.
- [12] K. E. C. Baker and C. McGeoch, “On the empirical comparison of QAOA with classical heuristics for combinatorial portfolio selection,” *arXiv preprint arXiv:2210.12547*, 2022.
- [13] Q. Huangfu and J. A. J. Hall, “HiGHS: High performance software for linear, mixed-integer, and quadratic optimization,” 2018, open-source solver; <https://highs.dev>.
- [14] D. J. Egger, J. Marecek, and S. Woerner, “Warm-starting quantum optimization,” *Quantum*, vol. 5, p. 479, 2021.
- [15] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, “Xy mixers: Analytical and numerical results for the quantum alternating operator ansatz,” *Physical Review A*, vol. 101, no. 1, p. 012320, 2020.
- [16] J. Cook, S. Eidenbenz, and A. Bärttschi, “The quantum alternating operator ansatz on maximum k -vertex cover,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2020, pp. 83–92.
- [17] G. E. Crooks, “Performance of the Quantum Approximate Optimization Algorithm on the maximum cut problem,” *arXiv preprint arXiv:1811.08419*, 2018.
- [18] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, “Towards quantum machine learning with tensor networks,” *Quantum Science and Technology*, vol. 4, no. 2, p. 024001, 2019.
- [19] J. Gray, “quimb: A python package for quantum information and many-body calculations,” p. 819, 2018.
- [20] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, “Improving variational quantum optimization using CVaR,” *Quantum*, vol. 4, p. 256, 2020.
- [21] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, “Simultaneous perturbation stochastic approximation of the quantum fisher information,” *Quantum*, vol. 5, p. 567, 2021.
- [22] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, “Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices,” *Physical Review X*, vol. 10, no. 2, p. 021067, 2020.
- [23] A. Løkketangen and F. Glover, “Solving zero-one mixed integer programming problems using tabu search,” *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 624–658, 1998.
- [24] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, “Heuristics for cardinality constrained portfolio optimisation,” *Computers & Operations Research*, vol. 27, no. 13, pp. 1271–1302, 2000.