A Multilevel Approach For Solving Large-Scale QUBO Problems With Noisy Hybrid Quantum Approximate Optimization

Filip B. Maciejewski*†, Bao G. Bach‡, Maxime Dupont§, P. Aaron Lott*†, Bhuvanesh Sundar§, David E. Bernal Neira¶†, Ilya Safro‡, Davide Venturelli*†

* Quantum Artificial Intelligence Laboratory (QuAIL), NASA Ames Research Center, CA, USA
† USRA Research Institute for Advanced Computer Science (RIACS), CA, USA
‡ Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA
§ Rigetti Computing, Berkeley, CA
¶ Davidson School of Chemical Engineering, Purdue University, West Lafayette, IN, USA
Email: fmaciejewski@usra.edu

***Abstract*—Quantum approximate optimization is one of the promising candidates for useful quantum computation, particularly in the context of finding approximate solutions to Quadratic Unconstrained Binary Optimization (QUBO) problems. However, the existing quantum processing units (QPUs) are of relatively small size, and canonical mappings of QUBO via the Ising model require one qubit per variable, rendering direct large-scale optimization infeasible. In classical optimization, a general strategy for addressing many large-scale problems is via multilevel/multigrid methods, where the large target problem is iteratively coarsened and the global solution is constructed from multiple small-scale optimization runs. In this work, we experimentally test how existing QPUs perform when used as a sub-solver within such a multilevel strategy. To this aim, we combine and extend (via additional classical processing steps) the recently proposed Noise-Directed Adaptive Remapping (NDAR) and Quantum Relax & Round (QRR) algorithms. We first demonstrate the effectiveness of our heuristic extensions on Rigetti's superconducting transmon device Ankaa-2. We find approximate solutions to 10 instances of fully connected 82-qubit Sherrington-Kirkpatrick graphs with random integer-valued coefficients obtaining normalized approximation ratios (ARs) in the range $\sim 0.98-1.0$, and the same class with real-valued coefficients (ARs $\sim 0.94-1.0$). Then, we implement the extended NDAR and QRR algorithms as subsolvers in the multilevel algorithm for 6 large-scale graphs with at most $\sim 27,000$ variables. In practice, the QPU (with classical post-processing steps) is used to find approximate solutions to dozens of at most 82-qubit problems, which are iteratively used to construct the global solution. We observe that quantum optimization results are competitive in terms of the quality of solutions when compared to classical heuristics used as subsolvers within the multilevel approach.**

Reproducibility: source code and data are available at [TBA upon acceptance].

I. Introduction

Quadratic Unconstrained Binary Optimization (QUBO) problems are a powerful framework to formulate numerous industrially relevant computational challenges in various fields such as logistics [1], finance [2], aerospace applications [3]. As such, much recent effort has been focused on empirically assessing whether quantum heuristics can provide any speedup over well-established classical heuristics [4]–[16]. One of the approaches to optimizing QUBO problems with quantum processing units (QPUs) is quantum approximate optimization – an umbrella term encompassing various heuristic techniques to find approximate solutions to (usually) combinatorial optimization problems. This includes the original, well-known Quantum Approximate Optimization Algorithm (QAOA) [17], but also a multitude of its variations, see, e.g., recent review [10].

To solve a QUBO problem on a quantum device, one typically maps it to an instance of an Ising model that describes a two-body Hamiltonian constructed from Pauli σ_Z operators. However, in the standard mapping, a single QUBO variable corresponds to a single qubit, restricting the usage of current quantum devices to solving only relatively small problems – limited by the size of the quantum device, which typically falls into range of dozens and hundreds. Indeed, to the best of our knowledge, the biggest-scale experimental demonstrations of quantum approximate optimization for QUBO have been limited to at most few-hundred-variables sparse graphs in Refs. [4]–[6], or few-dozen-variables dense graphs in Refs. [7]–[9], see also [10] for review.

In this work, we numerically and experimentally investigate the possibility of circumventing those scale

limitations by employing a multilevel (MLVL) approach to solve QUBO problems using quantum approximate optimization. We note that this is not the only known decomposition method that breaks down the global problem into smaller instances [11], [18]. Another related approach is to use dense encodings of the target variables [19]–[21].

We summarize here our main novel contributions.

- We improve upon the recently proposed Noise-Directed Adaptive Remapping (NDAR) [22] and Quantum Relax & Round (QRR) [8] algorithms. NDAR is a noise-tailored meta-algorithm involving an external loop, where each step requires running quantum optimization on a suitably gauge-transformed Hamiltonian. QRR is a classical post-processing scheme that uses results of quantum approximate optimization to propose better solutions. We combine and modify both algorithms by implementing a hardware-efficient ansatz (Time-Block QAOA from Ref. [9]), by improving parameter optimization, and by augmenting with additional classical processing steps. Our improvements lead to at least $\sim$ 3x speedup and increased solution quality compared to prior work.
- We improve and apply the methods introduced in recent works [23]–[25], where a novel multilevel solver for the Max-Cut (graph partitioning) problem was designed and numerically tested in small-scale QAOA simulations (IBM superconducting and DWave architectures). We find approximate solutions to 6 large-scale QUBO problems (up to $\approx 27,600$ variables), including the first time the method was applied to problem instances with negative/positive weighted coefficients, using Rigetti's superconducting QPU Ankaa-2. We use the extended NDAR+QRR algorithm to solve the subsidiary QUBOs. In practical terms, the QPU is used to find approximate solutions to multiple subsidiary instances of up to 82-qubit problems, demonstrating a performance that is competitive with state-of-the-art classical heuristics within the multilevel setting.

By improving and combining these two best-of-class approaches, our results are placed among the most complex experimental demonstrations of applied quantum optimization to date[1].

II. Preliminaries

A. Problem formulations

In this work, we define an n-variable QUBO problem as *maximization* over vector of binary variables $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})^{\mathrm{T}} \in \mathbb{B}^n$ (with $x_i \in \mathbb{B} = \{0,1\}$) specified by real-valued, upper-triangular matrix Q via $\max_{\mathbf{x}} \mathbf{x}^{\mathrm{T}} Q \mathbf{x} = \max_{\mathbf{x}} \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} Q_{i,j} x_i x_j$.

[1]Reproducibility: Link to our source code and data will be added here upon acceptance.

An n-variable (or n-node) simple undirected graph Max-Cut problem is specified by upper-triangular adjacency positive/negative edge weighted matrix W with zeros on the diagonal and involves *maximization* of the cost function $\max_{\mathbf{x}} \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} W_{i,j} (x_i + x_j - 2x_i x_j)$. An n-variable QUBO problem can be reformulated as $(n+1)$-variable Max-Cut problem. Up to multiplicative factor and a constant (irrelevant to optimization), this is done via mapping $W_{i,j} = -Q_{i,j}$ for $i \neq j$, and $W_{i,n} = Q_{i,i} + \sum_{j=i}^{n-1} Q_{i,j}$ for $i < n$ (edges between an additional node and rest of the graph), see, e.g., Refs. [26], [27]. If $\mathbf{y}$ is the corresponding Max-Cut solution, the solution to the original QUBO problem is recovered via $x_i = y_i \oplus_2 y_n$, where $\oplus_2$ denotes addition modulo 2.

An n-qubit Ising model is specified by Hamiltonian $H = \sum_{i=0}^{n-1} h_i \sigma_Z^{(i)} + \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} J_{i,j} \sigma_Z^{(i)} \sigma_Z^{(j)}$, where $\sigma_Z^{(k)}$ is a Pauli σ_Z operator acting on kth qubit, h_k and $J_{k,l}$ are real-valued coefficients called local fields and couplings, respectively. The corresponding Ising optimization problem is the *minimization* of the Hamiltonian over classical states $|\mathbf{x}\rangle = \otimes_{i=0}^{n-1} |x_i\rangle$ (with $|x_i\rangle$ denoting standard computational basis states), as in $\min_{|\mathbf{x}\rangle} \langle \mathbf{x}| H |\mathbf{x}\rangle = \min_{\mathbf{x}} \sum_{i=0}^{n-1} h_i (1-2x_i) + \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} J_{i,j} (1-2x_i)(1-2x_j)$. Note that since $(1-2x_i)(1-2x_j) = 1 - 2x_i - 2x_j + 4x_i x_j = 1 - 2(x_i + x_j - 2x_i x_j)$, in this convention the Max-Cut optimization (maximization) corresponds to Ising optimization (minimization) by identifying $J_{i,j} = W_{i,j}$, and setting local fields to 0. Mapping of QUBO to Ising can be thus done, for example, by first mapping n-variable QUBO to $(n+1)$-variable Max-Cut and identifying it with corresponding $(n+1)$-qubit Ising Hamiltonian, and this is the approach we take in this paper.

B. Figures of merit

To assess the quality of the solution, we compare the obtained cost C_i to the size of the spectrum via renormalized approximation ratio (AR) (see, e.g., [10]) $\mathrm{AR}_i = \frac{C_i - C_{\min}}{C_{\max} - C_{\min}}$, where $C_{\min}$ and $C_{\max}$ are the smallest and the highest known values of the cost function. Note that for maximization problems, this AR is equal to 1 for the optimal solution and to 0 for the worst (minimal) solution. Note that in the familiar case of unweighted Max-Cut, we have $C_{\min} = 0$, and the above is reduced to the standard approximation ratio. For minimization, the convention for AR is reversed to preserve the above properties w.r.t. the lowest cost state.

C. Multilevel solvers

A general strategy for addressing many large-scale computational problems on different hardware architectures, including various optimization problems on

graphs, is using multilevel algorithms (also known as multiscale, multiresolution, and multigrid-inspired methods) [28]–[32]. Specifically, in the quantum context, this multilevel framework has been explored for graph partitioning, clustering, and the Max-Cut problem, with the Quantum Approximate Optimization Algorithm (QAOA) serving as the main local processing component [23]–[25]. The motivation for combining the multilevel method with quantum optimization arises from the current state of quantum computers, which are constrained by a limited number of qubits. The multilevel framework addresses this limitation by coarsening the original graph into a hierarchy of reduced-size next coarser graphs, enabling the problem to be solved within the constraints imposed by quantum hardware. Each next coarser graph approximates the previous finer one with respect to the optimization problem but requires less resources to solve it.

In essence, the multilevel approach begins by coarsening the original problem to create a series (also known as a hierarchy) of progressively simpler, related problems at coarser levels. At each coarse level i, the best-found solution serves as an initialization for the next finer solution at level $i-1$. This initialization is enhanced through what is commonly referred to as "local processing" (also known as a refinement), a cost-effective series of fast steps that involve only a few variables at a time but collectively revisit all variables of that level multiple times. We refer the Reader to Ref. [28] for a pedagogical introduction to the subject.

Various coarsening-uncoarsening schedules exist in the multilevel algorithms to achieve a better optimization quality (e.g., W-cycle, and FMG [28]), but in this work, we explore the most basic single V-cycle to minimize the effect of classical processing. This setting involves sequentially generating a hierarchy of the next coarser graphs $\{G_l = (V_l, E_l, w_l)\}_{l=0}^{L}$, where l is the index of level, G_0 is the original large-scale graph, and G_L is the coarsest graph. The coarsening process consists of (1) relaxation-based grouping pairs of nodes based on the recently introduced maximization version of the algebraic distance [33] for graphs and (2) edge weight accumulation in each coarse level. Initially, every node is first placed in a random position on the surface of d-dimensional sphere. Following the initialization, several node-wise correction iterations are applied to maximize the total weighted distance between each node and its neighbors within the sphere, which maximizes the contribution of each node to the total energy of the system [23].

After the hierarchy is created, the Max-Cut instance at the coarsest level is solved, and the solution is gradually interpolated level-by-level all the way up to the finest level. At each level of the uncoarsening process, the lth level solution is initialized from level $l+1$ and further refined via sub-solvers (quantum or classical). During this refinement stage, sub-problems are iteratively generated and solved. Importantly, the maximal size of sub-problems (**MSS**) can be controlled to allow for implementation on limited-size QPU. If the solution of a generated sub-problem instance contributes to improving the final solution, a new sub-problem instance is produced and solved. The refinement stage ends when a specified maximal number of unsuccessful consecutive refinements (**MUR**) is reached. Note that the MUR parameter allows to control total runtime of the algorithm to achieve excellent quality/time trade-off, one of the main benefits of the multilevel algorithms.

D. Quantum Approximate Optimization

1) Quantum Approximation Optimization Algorithm: While there are multiple approaches to quantum approximate optimization [10], here we focus mainly on the canonical Quantum Approximate Optimization Algorithm (QAOA) introduced in [17] and its extensions (described later in this section). In this setting, the input state is $|+\rangle^{\otimes n}$ that is a tensor product of $+1$ eigenstates of Pauli σ_X operator. The quantum circuit applied to the input state is constructed from parametrized mixer operator $\exp(-i\beta_i H_M)$ with Hamiltonian $H_M = \sum_{i=0}^{n-1} \sigma_X^{(i)}$ and phase separation operator $\exp(-i\gamma_i H)$, with H being the Ising cost Hamiltonian (recall Section II-A). In the above, γ_i and β_i are elements of p-dimensional real-valued parameters vectors γ and β. The effective quantum state is obtained via application of the unitaries of the form $|\psi(\gamma,\beta)\rangle = (\prod_{i=1}^{p} \exp(-i\beta_i H_M)\exp(-i\gamma_i H))\,|+\rangle^{\otimes n}$. The aim of p-depth QAOA is to find p-dimensional γ and β that minimize the expected value of variational state evaluated on the cost Hamiltonian $\langle H\rangle_{\gamma,\beta} = \langle\psi(\gamma,\beta)|\,H\,|\psi(\gamma,\beta)\rangle$. This can be achieved by using various parameter setting strategies, as used for classical black-box optimization.

2) Noise-Directed Adaptive Remapping: In this work, we implement an improved version of an extension of QAOA called Noise-Directed Adaptive Remapping (NDAR). To the best of our knowledge, it is among the most performant quantum approximate optimization protocols to date, far outperforming standard QAOA in recent experimental demonstrations. The method was introduced in Ref. [22] for improved quantum optimization in the presence of certain types of hardware noise. NDAR is a meta-algorithm applied on top of standard QAOA (or other quantum approximate optimization). The main assumption of NDAR is that the noisy device has a special classical "attractor" state, towards which the dissipative processes push the overall system dynamics. This attractor state is described by a bitstring

$|\mathbf{x}_{att}\rangle$. We note that this assumption was demonstrated experimentally to be a good approximation for superconducting devices, where the noise attractor is typically $|\mathbf{x}_{att}\rangle = |0\ldots0\rangle$ due to amplitude damping/dissipation [22]. Each step of the NDAR loop involves performing quantum approximate optimization, identifying the best-cost bitstring, and re-mapping the cost Hamiltonian H in a way that aligns $|\mathbf{x}_{att}\rangle$ with that best-found solution. This is done via bitflip transformations (also known as spin-reversal transforms [34], [35]) that exchange the definition of $|0\rangle$ and $|1\rangle$ from the point of view of the Hamiltonian. Specifically, in each NDAR step, the gauge transformation specified by high-quality bitstring $\mathbf{y}$ is applied to the cost Hamiltonian $H \to H^{\mathbf{y}}$ (this is done in pre- and post-processing) in a way that $\langle\mathbf{y}|\, H\, |\mathbf{y}\rangle = \langle\mathbf{x}_{att}|\, H^{\mathbf{y}}\, |\mathbf{x}_{att}\rangle$. This effectively causes the solutions' distribution in the current optimization to be centered around high-quality solutions from the previous step. The iterative re-mapping of the Hamiltonian is done until the convergence criterion is met. In Ref. [22], NDAR was experimentally demonstrated to highly outperform original QAOA in 82-qubit experiments on fully-connected random graphs (so-called Sherrington-Kirkpatrick model).

3) Time-Block QAOA Ansatz: Another modification of the original QAOA we use is a Time-Block ansatz introduced in Ref. [9] for hardware-efficient optimization. The Time-Block (TB) QAOA uses the SWAP network structure of typical QAOA circuits, which is necessary to implement phase separator operators of dense Hamiltonians on limited connectivity hardware. Specifically, a TB k-QAOA is a circuit ansatz constructed from a standard QAOA circuit by dividing it into batches of physical depth k that are parametrized jointly and adding additional mixer operators between such new layers. This effectively parametrizes a subset of interactions of the original Hamiltonian as a single-phase separator, like in standard QAOA. The Time-Block QAOA with small values of k offers shallow-depth variational circuits that can perform similarly to the original QAOA, with a gain of easier experimental implementation (see results in Ref. [9]). Indeed, a depth p Time-Block k-QAOA corresponds to a circuit of physical depth $\sim pk$, compared to $\sim nk$ for standard QAOA.

4) Quantum Relax & Round: Finally, the bit strings sampled from the quantum computers are used to estimate a two-point correlation matrix Z with entries $\mathsf{Z}_{ij} = (\delta_{ij} - 1)\langle\sigma_z^{(i)}\sigma_z^{(j)}\rangle$, where δ_{ij} is the Kronecker delta. Then, an eigendecomposition of Z is performed and its eigenvectors are sign-rounded, entrywise. Considering each of the sign-rounded eigenvectors as a candidate solution to the original problem, we compute their cost C and keep the best one as the final solution. This algorithm, known as the quantum relax-and-round (QRR) algorithm, was developed in Ref. [8]. It was shown [8], [11] that when sampling bit strings from the standard QAOA with p layers, QRR converges asymptotically to the optimal solution with p. At $p = 1$, the performance of QRR matches that of a classical relax-and-round algorithm performed on the adjacency matrix of a graph problem with entries W_{ij}, as defined in Sec. II-A. This was proven analytically in the large n limit for several problem classes such as Sherrington-Kirkpatrick spin glasses, unit-weight random 3-regular graphs, as well as circulant graphs, and supported numerically on other problems. We note that the resulting average performance at $p = 1$ is much higher than that of the standard QAOA algorithm, for the same quantum resources.

III. Improved Noise-Directed Adaptive Remapping

In this section, we discuss our modifications to the original NDAR proposal and demonstrate their effectiveness in experiments on 82 qubits on Rigetti's superconducting transmon device Ankaa-2.

A. Modifications

Recall that the original NDAR proposal consists of an external loop, where each iteration involves adaptively solving gauge-transformed cost function Hamiltonian via QAOA or another quantum approximate optimization method. We modify the original NDAR algorithm by implementing the following changes:

1) **(Pre-processing at iteration** 0**)** NDAR at step 0 implements quantum approximate optimization for the original Hamiltonian H. Its representation is always provided with some implicit, "default" bitflip gauge choice. Here, we generate 10^4 random solutions, evaluate them on the cost Hamiltonian, and choose the best-cost solution $\mathbf{y}$ to specify the initial bitflip gauge. For the gauge-transformed Hamiltonian $H^{\mathbf{y}}$, the noise attractor state $|0\ldots0\rangle$ becomes a higher-quality solution, in a sense that $\langle\mathbf{y}|\, H\, |\mathbf{y}\rangle = \langle 0\ldots0|\, H^{\mathbf{y}}\, |0\ldots0\rangle$ (see II-D).

2) **(Shallow-depth circuit ansatz)** Instead of using original QAOA, here we implement a Time-Block k-QAOA from [9], see discussion in Section II-D. We choose $k = \frac{n}{5}$ to reduce physical depth approximately five-fold compared to standard QAOA. We set algorithmic depth (the number of parametrized layers) to $p = 1$.

3) **(Optimization over multiple gauges)** We allow the optimizer to choose one of 4 best-cost bitstrings from the previous step to specify allowed gauge transformations in the next step, as opposed to the single best bitstrings in the original NDAR.

4) **(Quantum Relax & Round and extension)** At each iteration step, we take the results of quantum optimization and evaluate the QRR algorithm on them to find a better approximate solution (recall Section II-D).

Moreover, we implement an alternative version of QRR that constructs weighted correlators of the form $J_{i,j}\left\langle\sigma_z^{(i)}\sigma_z^{(j)}\right\rangle$, and otherwise works the same as QRR (see Section II-D. The algorithm, which we call w-QRR (weighted-QRR) aims to resemble the classical relax and round strategies. At each optimization, we implement both QRR and w-QRR on the results from QPU and choose a better solution.

5) **(Hamming Distance Quadratic Local Search)** At the end of each step, we additionally explore the neighborhood of the best-cost bitstring by generating and evaluating all solutions within Hamming distance 2. We refer to this step as HDQLS.

B. Experimental results on 82-qubit dense SK problems

We now present the results of benchmarks of modified Noise-Directed Adaptive Remapping and compare it with the original version for the same 10 instances of the Ising model as in Ref. [22]. Those Hamiltonians correspond to fully connected graphs, with integer-valued interactions taken randomly as ± 1, so-called Sherrington-Kirkpatrick (SK) model [39] on 82 qubits.

We implement NDAR experiments with the adaptive optimizer known as Tree-Structured Parzen Estimators (TPE) [40], as implemented in package Optuna [41]. Each iteration step involves implementing Time-Block 8-QAOA with TPE using $t = 150$ cost function evaluations and gathering $s = 1000$ samples at each evaluation, same as in Ref. [22]. Following Refs. [22], [42], we also allow the TPE to optimize over categorical variable controlling the gates' ordering of the ansatz.

In the top plot of Fig. 1, we present the joint result from Ref. [22] and our implementation. We observe that the introduced modifications of the protocol lead to both better performance and much faster convergence of Noise-Directed Adaptive Remapping (at least $\sim$ 3x - although a rigorous parameter setting cost analysis should be conducted to determine the comparison [43]). In the middle plot, we investigate in more detail how much each step of the extended NDAR (recall Section III) contributes to the final solution quality. We observe that the pre-processing allows to start already from approximation ratios around 0.7, and then, in the first iteration, the results obtained directly from QPU (via Time-Block QAOA) improve it to around 0.8. Further improvement by 0.1 is provided by weighted QRR that turns out to perform better than standard QRR (that does not improve upon w-QRR). Finally, additional small improvements are gained by local search in the Hamming distance. Further iterations of NDAR refine the solution quality, until we obtain $\sim 0.98 - 1.0$ AR for all tested instances with at most 4 NDAR iterations.

The bottom plot of Fig. 1 shows analogous investigation for 10 random SK model instances with *real-valued* interactions taken from range $[0, 1]$. We obtain approximation ratios in the range of $\sim 0.941 - 1.0$ after at most 10 iterations. As such, the above results are placed among the most complex and most-performant applied quantum optimization experiments to date (see, e.g., Table IV in Ref. [10]).

IV. Multilevel approach to solving QUBO

A. Benchmark graphs

In this work, we consider a set of 6 benchmark graphs summarized in Table I. The biggest considered graph has 26588 nodes with an average degree of ~ 4, while the densest graph is a 1750-node problem with ~ 875 average degree. We chose graphs that are fairly small compared to typical problems of industrial relevance, so we can benchmark the multilevel method on the real QPU with limited resources. The large-scale graphs are first solved using *non-multilevel* (non-MLVL) heuristics chosen from the Mqlib library [27], where the optimization is performed on the whole graph, as opposed to coarsened sub-problems generated by MLVL approach. For each graph, we implemented around 25 solvers (with a timeout of at most 15 minutes) that use various heuristics including simulated annealing [44], tabu-search [45], [46], genetic algorithms [47], and greedy local-search [48]. We note that the best or second-best solver for all graphs proved to be the Burer-Monteiro algorithm from Ref. [49], which is a rank-2 relaxation of the famous Goemans-Williamson algorithm [50].

In Table I, we show absolute values of the best cost function (cut) values found across all classical solvers. Those values will be the reference when assessing the performance of the multilevel approach (via approximation ratio w.r.t. the non-MLVL optimizer) in the following sections.

B. Classical heuristics sub-solver: MLVL algorithm

We start by investigating the performance of the multilevel approach with classical subsolvers. Recall from Section II-C that the MLVL algorithm from Ref. [23] is specified by, among others, the following hyperameters – Maximal Subproblem Size (MSS) and Maximal consecutive Unsuccessful Refinements (MUR). MSS controls the biggest subproblems the solver is allowed to generate and thus can be set to the size of a given quantum device, allowing hybrid quantum-classical approximate optimization of large graphs using small-scale hardware. MUR affects the run-time, it can thus be adjusted to the QPU availability. In short, increasing MSS/MUR increases the space/time requirements of the multilevel approach.

We now numerically investigate how those parameters affect the optimization of our benchmark graphs when the sub-solver is *classical* heuristics. As sub-solver

Graph name (U/W)	Id	Nodes	Edges (av. degree)	Best cut (non-MLVL)	Ref.
soc-epinions (U)	**s-e (U)**	26588	100120 (3.77)	70112	[36]
soc-epinions (W)	**s-e (W)**			11398	here
Karloff 16-7-1 (U)	**Krl (U)**	11440	3363360 (294)	2522520	[37]
Karloff 16-7-1 (W)	**Krl (W)**			59685	here
Rolling Stock Assignment 3 (W)	**RLS3**	1750	1530375 (874.5)	6829	[38]
Queens 16 (W)	**Q16**	1035	535095 (517)	14428	[38]

TABLE I: Large-scale graphs considered in this work, with reference solutions obtained via classical global heuristics *not* employing our multilevel approach. "U" ("W") stands for "unweighted" ("weighted"), meaning equal (non-equal) weights at each edge. For the soc-epinions and the Karloff graph, the original graphs are unweighted and to obtain a weighted (general QUBO) version, we add random weights from $[-1, 1]$ range. The two graphs from Ref. [38] are already general QUBO – here we map them to the Max-Cut problem (see Section II-A). The "Best cut" column denotes the best-found cut using classical heuristics solvers from the MQLib library [27]. For each graph, multiple heuristics are tested (see text description), and the shown results are the best cut found across at most 100 random initialization of the solver (with allowed running time of at most 15 minutes each).

heuristics, we used 2-3 solvers that performed the best in solving the original, large graphs. This included the Burer-Monteiro algorithm [49] for all graphs, and other solvers [44]–[48] varied between graphs (see discussion in the previous Section); all sub-solvers were allowed to run for at most 10 seconds per sub-problem. The best-found results are presented in Table II (we note that the Burer-Monteiro algorithm proved to be usually the most performant also in the multilevel setting). The mean approximation ratio is estimated empirically by running the solver for $10 - 100$ independent random initialization (10 for all cases except for the pair $\mathrm{MSS} = 82$ and $\mathrm{MUR} = 3$ with 100 runs, where we increased statistics to have a better comparison with QPU implementation in the next section). We observe a monotonic increase in solution quality for both hyperparameters. We observe that for 5000-variable subproblems, the multilevel solver starts to outperform global heuristics for the biggest considered graphs (soc-epinions), which corresponds to a five-fold size reduction compared to the original problem. For other graphs, it becomes competitive around 500-variable subproblems, except for the weighted Karloff graph where 2000 variables are needed to obtain approximation ratios around 0.99. In the case of smaller QUBOs (RLS3 and Q16 from Ref. [38]), the multilevel solver finds very good solutions already for 82-node subproblems. However, in that case, it is worth noting that the initial coarsening phase of the algorithm already finds high-quality solutions, and the subproblem refinements improve them only slightly, indicating that the solutions of subproblems (either from classical or quantum sub-solver) do not contribute much to the final solution.

C. 82-qubit QPU NDAR sub-solver: MLVL algorithm

For subproblems of size 82 (the size of our experiments on Rigetti's QPU), the multilevel approach with classical sub-solver always finds worse solutions than non-MLVL heuristics, thus for those system sizes we do not expect to observe any advantage over those state-of-the-art heuristics. Yet, it is worthwhile to benchmark how current QPUs perform compared to classical sub-solvers within the multilevel setting, and possibly extrapolate the classical results from Table II to estimate what we can expect from future generations of QPUs. The objectives are to gain insights on the following questions in order to guide further R&D: (i) how does the MLVL decomposition scores against the best classical non-MLVL methods? (ii) are the solutions obtained by the quantum solvers similar to the ones obtained by the best classical method (multiscale or not) (iii) how does the performance when using the quantum method compare against a competitive non-quantum subsolver? We implement the extended NDAR (recall Section III) as a multilevel sub-solver with at most 82-qubit subproblems. The results are presented in Fig. 2, together with corresponding results from Table II with classical subsolvers.

The top plot shows the Jaccard similarity coefficient (i.e. the size of the intersection divided by the size of the union) between the set of edges participating in a given cut (obtained via the multilevel approach) and the optimal cut (from non-MLVL solvers, see Table I). We observe that the structural difference (measured by this coefficient) between solutions obtained from QPU and classical are minor, suggesting that QPU finds similar solutions to classical MLVL solvers. We note that their coefficients are significantly different than 1.0 which suggests that all solutions have different structure than the best-found solution to the global problem.

The middle plot shows the approximation ratio (compared to global solver solutions in Table. I), while the bottom plot shows the total sub-solver calls, which gives an estimated runtime measured in the number of implemented local optimizations on 82-variable subproblems. We notice that for both unweighted and weighted soc-

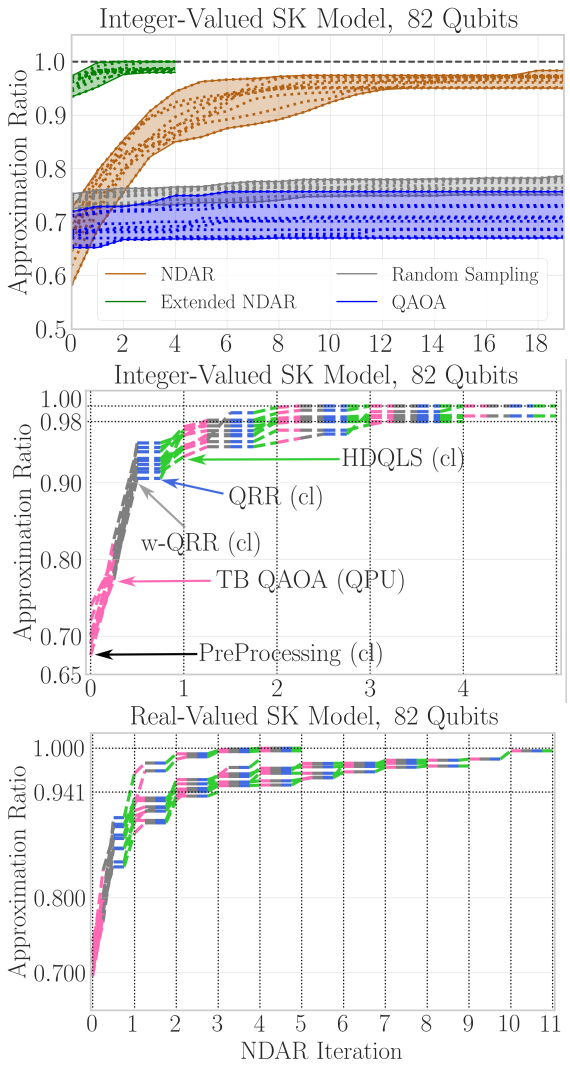

Fig. 1: The top plot shows the experimental performance of extended NDAR (green), compared to original NDAR (orange), original QAOA (blue), and random bitstrings sampling (grey). The data points other than green are taken from Ref. [22]. Those experiments were performed on 10 random instances of 82-qubit integer-valued Sherrington-Kirkpatrick Hamiltonians. The dotted lines within each colored region correspond to individual instances. The middle plot shows a zoomed-in region of the top plot that corresponds to extended NDAR, with more refined information. Here, the end of each colored region corresponds to AR resulting from applying the corresponding algorithm step ("cl" stands for classical processing). For example, the improvement gained directly from the QPU run (Time-Block QAOA) is indicated by the change in the AR value indicated by the pink line. Different lines are different Hamiltonian instances. The bottom plot shows the extended NDAR performance applied to real-valued Sherrington-Kirkpatrick models with the same convention. Note that y-axis scales do not start at 0.0 for improved clarity.

Fig. 2: Results of Noise-Directed Adaptive Remapping implemented on Rigetti's QPU Ankaa-2 applied as subsolver for the multilevel (ML) approach, compared to using classical subsolvers. The hyperparameters are set to MSS = 82 and MUR = 3. The top plot presents the Jaccard similarity coefficient between the edges participating in the cut for the optimal solution and the one from the multilevel approach. The middle plot shows the approximation ratio (AR, note that y scale starts at 0.9 for improved visibility), while the bottom plot shows the total number of sub-solver calls in the multilevel approach. For the classical approach, the error bars correspond to the range of obtained values (of best-found solutions) when running a multilevel algorithm with 100 independent random initializations. QPU results correspond to the best solution from a single experimental implementation. We did not implement the MLVL method multiple times due to limited QPU availability.

epinions, the QPU seems to converge faster than the classical sub-solver, while offering solutions of similar quality. For both Karloff graphs, QPU finds solutions with slightly worse ARs than the classical subsolvers, while converging faster for the weighted variant. For RLS3 and Q16 graphs, the QPU is similar in both quality and convergence speed.

Id	MSS	MUR	MEAN	MAX
s-e (U)	82	3	0.972 ± 0.008	0.978
		10	0.980 ± 0.007	0.983
	500	3	0.982 ± 0.009	0.986
		10	0.991 ± 0.003	0.993
	5000	3	1.00 ± 0.01	1.001
		10	1.001 ± 0.001	1.002
s-e (W)	82	3	0.96 ± 0.01	0.968
		10	0.969 ± 0.007	0.973
	500	3	0.98 ± 0.03	0.990
		10	0.992 ± 0.008	0.996
	5000	3	0.99 ± 0.02	1.001
		10	1.001 ± 0.001	1.002
Krl (U)	82	3	0.98 ± 0.02	0.992
		10	0.987 ± 0.006	0.990
	500	3	0.995 ± 0.003	0.997
		10	0.995 ± 0.004	0.997
	2000	3	0.999 ± 0.002	0.9994
		10	0.9996 ± 0.0005	0.9998
Krl (W)	82	3	0.94 ± 0.01	0.948
		10	0.950 ± 0.007	0.955
	500	3	0.96 ± 0.01	0.963
		10	0.976 ± 0.003	0.978
	2000	3	0.98 ± 0.02	0.984
		10	0.989 ± 0.003	0.991
RLS3	82	3	0.999990 ± 0.000005	0.999995
		10	0.999991 ± 0.000006	0.999995
	500	3	0.999996 ± 0.000007	0.999998
		10	0.999998 ± 0.000005	0.9999999
Q16	82	3	0.9989 ± 0.0009	0.99946
		10	0.9993 ± 0.0007	0.99970
	500	3	0.99997 ± 0.00007	0.999997
		10	0.99998 ± 0.00004	0.999997

TABLE II: Results of classical heuristics applied as subsolvers for the multilevel approach. "MSS" stands for Max Subproblem Size, i.e., the maximal number of variables for each solved refinement subproblem. "MUR" stands for "Max Unsuccesfull Consecutive Refinements", specifying how many non-improving refinements are performed before going to the next hierarchy level. See Section II-C for details. For each set of MSS and MUR, each graph is solved using the multilevel approach with $10 - 100$ independent initialization for the refinements (the embedding is fixed). The MEAN and MAX values refer to approximation ratios obtained in those independent runs. Error bars indicate 3 standard deviations estimated empirically. Approximation ratio (AR) is computed w.r.t. the results obtained with global solver, presented in Table I.

V. Conclusions

An ability to solve large-scale optimization problems on real quantum devices is crucial to demonstration of practical quantum advantage in this area. Here we present a complex hybrid quantum-classical algorithmic pipeline that is a candidate to achieve this goal. We introduced advanced modifications to our best-known approach for practical quantum optimization, by hybridizing the Noise-Directed Adaptive Remapping (NDAR) [22] and Quantum Relax & Round (QRR) [8] algorithms. We experimentally demonstrated that augmenting NDAR hardware-efficient Time-Block QAOA ansatz [9] runs with additional classical processing steps allows to find high-quality solutions to fully-connected random graphs on 82 qubits, improving upon previous experimental demonstrations. Equipped with the new quantum heuristics, we investigated how they perform when used as a sub-solver in a multilevel approach to solving large-scale QUBO problems up to $\sim 27,000$ variables. *Our results indicate that the real noisy QPU quantum solver is competitive w.r.t. state-of-the-art heuristics used as sub-solvers in the multilevel approach.* Note that we restricted this initial work to analyze approximation ratios and solution similarity for a fixed meta-parameter setting strategy without estimating time-to-solution. Future work will further optimize NDAR and the MLVL scheme and will quantify additional metrics such as time-to-solution and energy-to-solution. We believe that our results constitute a promising step towards employing limited-scale, noisy quantum hardware for tackling industrially relevant large-scale problems.

Acknowledgments

All authors acknowledge support by the Defense Advanced Research Projects Agency (DARPA) ON-ISQ program (for NASA/USRA: Agreement No. HR00112090058 and IAA8839, Annex 114). Authors from USRA also acknowledge support under NASA Academic Mission Services under contract No. NNA16BD14C.

References

[1] Frank Phillipson. Quantum computing in logistics and supply chain management-an overview. *arXiv preprint arXiv:2402.17520*, 2024.

[2] Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alexeev. Quantum computing for finance. *Nature Reviews Physics*, 5(8):450–465, 2023.

[3] Eleanor G Rieffel, Ata Akbari Asanjan, M Sohaib Alam, Namit Anand, David E Bernal Neira, Sophie Block, Lucas T Brady, Steve Cotton, Zoe Gonzalez Izquierdo, Shon Grabbe, et al. Assessing and advancing the potential of quantum computing: A nasa case study. *Future Generation Computer Systems*, 2024.

[4] Elijah Pelofske, Andreas Baertschi, and Stephan Eidenbenz. Short-depth qaoa circuits and quantum annealing on higher-order ising models (rev. 2). Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2023.

[5] Elijah Pelofske, Andreas Bärtschi, Lukasz Cincio, John Golden, and Stephan Eidenbenz. Scaling whole-chip qaoa for higher-order ising spin glass models on heavy-hex graphs, 2023.

[6] Natasha Sachdeva, Gavin S. Harnett, Smarak Maity, Samuel Marsh, Yulun Wang, Adam Winick, Ryan Dougherty, Daniel Canuto, You Quan Chong, Michael Hush, Pranav S. Mundada, Christopher D. B. Bentley, Michael J. Biercuk, and Yuval Baum. Quantum optimization using a 127-qubit gate-model ibm quantum computer can outperform quantum annealers for nontrivial binary optimization problems, 2024.

[7] Maxime Dupont, Bram Evert, Mark J Hodson, Bhuvanesh Sundar, Stephen Jeffrey, Yuki Yamaguchi, Dennis Feng, Filip B Maciejewski, Stuart Hadfield, M Sohaib Alam, et al. Quantum-enhanced greedy combinatorial optimization solver. *Science Advances*, 9(45):eadi0487, 2023.

[8] Maxime Dupont and Bhuvanesh Sundar. Extending relax-and-round combinatorial optimization solvers with quantum correlations. *Physical Review A*, 109(1), January 2024.

[9] Filip B. Maciejewski, Stuart Hadfield, Benjamin Hall, Mark Hodson, Maxime Dupont, Bram Evert, James Sud, M. Sohaib Alam, Zhihui Wang, Stephen Jeffrey, Bhuvanesh Sundar, P. Aaron Lott, Shon Grabbe, Eleanor G. Rieffel, Matthew J. Reagor, and Davide Venturelli. Design and execution of quantum circuits using tens of superconducting qubits and thousands of gates for dense Ising optimization problems, 2023.

[10] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini, Bryce Fuller, Julien Gacon, Constantin Gonciulea, Sander Gribling, Swati Gupta, Stuart Hadfield, Raoul Heese, Gerhard Kircher, Thomas Kleinert, Thorsten Koch, Georgios Korpas, Steve Lenk, Jakub Marecek, Vanio Markov, Guglielmo Mazzola, Stefano Mensa, Naeimeh Mohseni, Giacomo Nannicini, Corey O'Meara, Elena Peña Tapia, Sebastian Pokutta, Manuel Proissl, Patrick Rebentrost, Emre Sahin, Benjamin C. B. Symons, Sabine Tornow, Victor Valls, Stefan Woerner, Mira L. Wolf-Bauwens, Jon Yard, Sheir Yarkoni, Dirk Zechiel, Sergiy Zhuk, and Christa Zoufal. Quantum optimization: Potential, challenges, and the path forward, 2023.

[11] Maxime Dupont, Bhuvanesh Sundar, Bram Evert, David E. Bernal Neira, Zedong Peng, Stephen Jeffrey, and Mark J. Hodson. Quantum Optimization for the Maximum Cut Problem on a Superconducting Quantum Computer, 2024.

[12] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin. Quantum optimization of maximum independent set using rydberg atom arrays. *Science*, 376(6598):1209–1215, June 2022.

[13] Andrew Byun, Minhyuk Kim, and Jaewook Ahn. Finding the maximum independent sets of platonic graphs using rydberg atoms. *PRX Quantum*, 3:030305, Jul 2022.

[14] Minhyuk Kim, Kangheun Kim, Jaeyong Hwang, Eun-Gook Moon, and Jaewook Ahn. Rydberg quantum wires for maximum independent set problems. *Nature Physics*, 18(7):755–759, June 2022.

[15] Andrew D. King, Jack Raymond, Trevor Lanting, Richard Harris, Alex Zucca, Fabio Altomare, Andrew J. Berkley, Kelly Boothby, Sara Ejtemaee, Colin Enderud, Emile Hoskinson, Shuiyuan Huang, Eric Ladizinsky, Allison J. R. MacDonald, Gaelen Marsden, Reza Molavi, Travis Oh, Gabriel Poulin-Lamarre, Mauricio Reis, Chris Rich, Yuki Sato, Nicholas Tsai, Mark Volkmann, Jed D. Whittaker, Jason Yao, Anders W. Sandvik, and Mohammad H. Amin. Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature*, 617(7959):61–66, May 2023.

[16] Minh-Thi Nguyen, Jin-Guo Liu, Jonathan Wurtz, Mikhail D. Lukin, Sheng-Tao Wang, and Hannes Pichler. Quantum optimization with arbitrary connectivity using rydberg atom arrays. *PRX Quantum*, 4:010316, Feb 2023.

[17] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv:1411.4028*, 2014.

[18] Tony Tran, Minh Do, Eleanor Rieffel, Jeremy Frank, Zhihui Wang, Bryan O'Gorman, Davide Venturelli, and J Beck. A hybrid quantum-classical approach to solving scheduling problems. In *Proceedings of the International Symposium on Combinatorial Search*, volume 7, pages 98–106, 2016.

[19] Marco Sciorilli, Lucas Borges, Taylor L Patti, Diego García-Martín, Giancarlo Camilo, Anima Anandkumar, and Leandro Aolita. Towards large-scale quantum optimization solvers with few qubits. *arXiv preprint arXiv:2401.09421*, 2024.

[20] Bryce Fuller, Charles Hadfield, Jennifer R Glick, Takashi Imamichi, Toshinari Itoko, Richard J Thompson, Yang Jiao, Marna M Kagele, Adriana W Blom-Schieber, Rudy Raymond, et al. Approximate solutions of combinatorial problems via quantum relaxations. *IEEE Transactions on Quantum Engineering*, 2024.

[21] Benjamin Tan, Marc-Antoine Lemonde, Supanut Thanasilp, Jirawat Tangpanitanon, and Dimitris G Angelakis. Qubit-efficient encoding schemes for binary optimisation problems. *Quantum*, 5:454, 2021.

[22] Filip B Maciejewski, Jacob Biamonte, Stuart Hadfield, and Davide Venturelli. Improving Quantum Approximate Optimization by Noise-Directed Adaptive Remapping. *arXiv preprint arXiv:2404.01412*, 2024.

[23] Anthony Angone, Xiaoyuan Liu, Ruslan Shaydulin, and Ilya Safro. Hybrid quantum-classical multilevel approach for maximum cuts on graphs. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2023.

[24] Bao Bach, Jose Falla, and Ilya Safro. Mlqaoa: Graph learning accelerated hybrid quantum-classical multilevel qaoa. *arXiv preprint arXiv:2404.14399*, 2024.

[25] Hayato Ushijima-Mwesigwa, Ruslan Shaydulin, Christian FA Negre, Susan M Mniszewski, Yuri Alexeev, and Ilya Safro. Multilevel combinatorial optimization across quantum architectures. *ACM Transactions on Quantum Computing*, 2(1):1–29, 2021.

[26] Endre Boros and Peter L. Hammer. The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, 33(3):151–180, Mar 1991.

[27] Iain Dunning, Swati Gupta, and John Silberholz. What works best when? a systematic evaluation of heuristics for max-cut and QUBO. *INFORMS Journal on Computing*, 30(3), 2018.

[28] Achi Brandt and Dorit Ron. Multigrid solvers and multilevel optimization strategies. *Multilevel optimization in VLSICAD*, pages 1–69, 2003.

[29] Ilya Safro, Dorit Ron, and Achi Brandt. Graph minimum linear arrangement by multilevel weighted edge contractions. *Journal of Algorithms*, 60(1):24–41, 2006.

[30] Dorit Ron, Ilya Safro, and Achi Brandt. Relaxation-based coarsening and multiscale graph organization. *Multiscale Modeling & Simulation*, 9(1):407–423, 2011.

[31] Sven Leyffer and Ilya Safro. Fast response to infection spread and cyber attacks on large-scale networks. *Journal of Complex Networks*, 1(2):183–199, 2013.

[32] Ilya Safro and Boris Temkin. Multiscale approach for the network compression-friendly ordering. *Journal of Discrete Algorithms*, 9(2):190–202, 2011.

[33] Jie Chen and Ilya Safro. Algebraic distance on graphs. *SIAM Journal on Scientific Computing*, 33(6):3468–3490, 2011.

[34] Sergio Boixo, Tameem Albash, Federico M Spedalieri, Nicholas Chancellor, and Daniel A Lidar. Experimental signature of programmable quantum annealing. *Nature communications*, 4(1):2067, 2013.

[35] Elijah Pelofske, Georg Hahn, and Hristo Djidjev. Optimizing the spin reversal transform on the d-wave 2000q. In *2019 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE, 2019.

[36] Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1), dec 2011.

[37] Howard Karloff. How good is the goemans-williamson max cut algorithm? In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 427–434, New York, NY, USA, 1996. Association for Computing Machinery.

[38] Fraunhofer FOKUS. Qubo problems dataset, 2024. Accessed: 2024-07-06.

[39] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical review letters*, 35(26):1792, 1975.

[40] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[41] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[42] Filip B Maciejewski, Flavio Baccari, Zoltán Zimborás, and Michał Oszmaniec. Modeling and mitigation of cross-talk effects in readout noise with applications to the quantum approximate optimization algorithm. *Quantum*, 5:464, 2021.

[43] David E Bernal Neira, Robin Brown, Pratik Sathe, Filip Wudarski, Marco Pavone, Eleanor G Rieffel, and Davide Venturelli. Benchmarking the operation of quantum heuristics and ising machines: Scoring parameter setting strategies on optimization applications. *arXiv preprint arXiv:2402.10255*, 2024.

[44] Talal M. Alkhamis, Mohamed A. Ahmed, and Vu Kim Tuan. Simulated annealing for discrete optimization with estimation. *European Journal of Operational Research*, 116(3):530–544, 1999.

[45] John E Beasley. Heuristic algorithms for the unconstrained binary quadratic programming problem, 1998.

[46] Zhipeng Lü, Fred Glover, and Jin-Kao Hao. A hybrid metaheuristic approach to solving the ubqp problem. *European Journal of Operational Research*, 207(3):1254–1262, 2010.

[47] Peter Merz and Bernd Freisleben. Genetic algorithms for binary quadratic programming. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1*, GECCO'99, page 417–424, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[48] Peter Merz and Bernd Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):197–213, Mar 2002.

[49] Samuel Burer, Renato D. C. Monteiro, and Yin Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521, 2002.

[50] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.