

Variational matrix product states for combinatorial optimization

Guillermo Preisser^{✉,*}, Conor Mc Keever[✉], and Michael Lubasch^{✉,†}

Quantinuum, Partnership House, Carlisle Place, London SW1P 1BX, United Kingdom



(Received 24 December 2025; revised 3 April 2026; accepted 9 June 2026; published 6 July 2026)

To compute approximate solutions for combinatorial optimization problems, we describe variational methods based on the product state (PS) and matrix product state (MPS) ansätze. We perform variational energy minimization with respect to a quantum annealing Hamiltonian and utilize randomness by embedding the approaches in the metaheuristic iterated local search (ILS). The resulting quantum-inspired ILS algorithms are benchmarked on maximum cut problems of up to 50 000 variables. We show that they can outperform traditional (M)PS methods, classical ILS, the quantum approximate optimization algorithm, and other variational quantum-inspired solvers.

DOI: [10.1103/x7s3-ykb8](https://doi.org/10.1103/x7s3-ykb8)

I. INTRODUCTION

Remarkable algorithmic discoveries have enabled combinatorial optimization to solve problems of relevance to several industries, including manufacturing [1], transportation [2,3], and telecommunication [4]. A key ingredient of certain successful classical approaches, such as iterated local search (ILS) [5], is the strategic use of randomness to escape local optima and efficiently explore the solution space in search of better optima. From the perspective of quantum computing, promising proposals for combinatorial optimization are quantum annealing [6–10] and variational quantum algorithms [11–13], in particular, the quantum approximate optimization algorithm (QAOA) [14,15]. However, current quantum hardware suffers from experimental noise to such an extent that certain quantum computations can be amenable to classical simulations, for which the variational matrix product state (MPS) ansatz [16–20] has proven to be a particularly powerful tool [21,22]. This naturally raises the following question: By making use of both classical and quantum concepts, can MPS algorithms outperform established classical solvers for combinatorial optimization?

In this paper, we combine MPS algorithms for quantum annealing with ILS and show that the resulting quantum-inspired ILS (QiILS) approaches outperform both classical ILS and standard MPS methods. In the context of maximum cut (Max-Cut) problems [23,24], which are a paradigmatic benchmark in combinatorial optimization, we also show that QiILS can find better solutions more efficiently than QAOA, the variational quantum-inspired algorithm local quantum annealing (LQA) [25], and its variant based on generalized group-theoretic coherent states (GCS) [26–28] (see Appendix A

for concise descriptions of ILS, LQA, and GCS). Furthermore, we introduce an embarrassingly parallel alternative to QiILS: Quantum-inspired iterative global search (QiIGS) employs global gradient updates that enable parallelization across problem size. We show that QiIGS can efficiently utilize GPU-based parallel computing, exhibit a runtime nearly independent of problem size for instances with up to 50 000 variables, and provide an order-of-magnitude speedup over QiILS for the largest, 20 000-variable, MaxCut instance (G81) from the Gset dataset [29].

Figure 1 illustrates the typical advantage of QiILS over other solvers on the Gset problems. In this example, we compare QiILS with LQA, GCS, and ILS on G12, a standard benchmark graph (see figure caption for details). Performance is evaluated in terms of the relative error, defined as $1 - r$, where r denotes the approximation ratio between the obtained and optimal (or best-known) cut values. After an initialization phase of $\approx 10^4$ sweeps, used to tune a hyperparameter, QiILS consistently outperforms the competing methods and achieves more than an order-of-magnitude improvement.

Our study is motivated by earlier works on discrete optimization using (M)PSs [25,30–47] and more complex tensor networks [32,48–62]. Most closely related to our proposals are approaches that propagate (M)PSs along adiabatic paths [25,30–33,37,39,47] or in imaginary time [34]. Compared with these prior methods, our algorithms more directly target the solution space and deliberately exploit randomness to find better solutions. They also facilitate a comparison of the utility of randomness and limited quantum entanglement (accessible via MPS) in combinatorial optimization.

The paper is organized as follows. Section II contains background information. We present and analyze QiILS based on MPS and product states (PSs) in Secs. III and IV, respectively. Then, we investigate the performance mechanism underlying QiILS in Sec. V, discuss the selection of the main QiILS hyperparameter in Sec. VI, perform a comparison with QAOA in Sec. VII, present additional Gset results in Sec. VIII, and introduce and study QiIGS in Sec. IX. Finally, we conclude with Sec. X. Additional information is provided in the Appendices.

*Contact author: guillermo.preisser@quantinuum.com

†Contact author: michael.lubasch@quantinuum.com

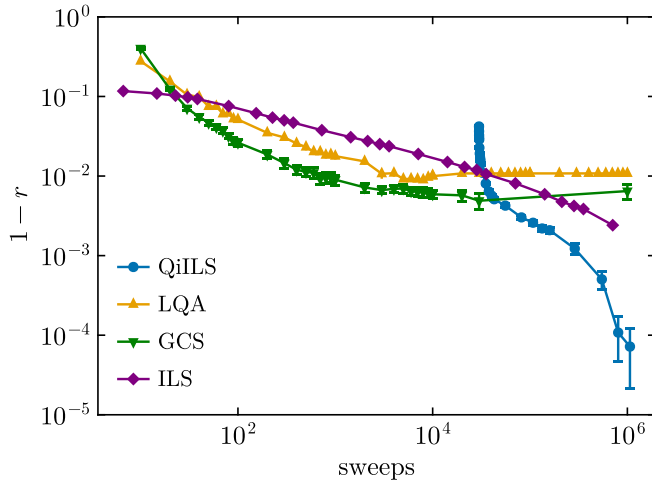


FIG. 1. Performance comparison for G12. Average relative error $1 - r$, averaged over 100 random initial states, for G12 [29], a toroidal weighted MaxCut problem of 800 variables, and weights from $\{-1, 1\}$. If not visible, error bars are smaller than their associated symbols. The offset for QiILS results from additional sweeps used to tune a hyperparameter that aims to obtain more consistent and faster improvement per sweep. Despite this offset, QiILS achieves the best performance. For each method, results correspond to the best hyperparameter choice within the explored hyperparameter search space. Each sweep of QiILS and ILS runs in a similar amount of time, whereas LQA (GCS) is roughly 7 (9000) times slower. Further details on hyperparameters as well as runtimes per sweep are provided in Appendixes B and C.

II. BACKGROUND

For a quantum system of n qubits, an MPS with open boundary conditions takes the form $|\Psi\rangle = \sum_{\{s_j\}} A_1^{s_1} A_2^{s_2} \cdots A_n^{s_n} |s_1, s_2, \dots, s_n\rangle$ [16], where $s_j \in \{0, 1\}$, each $A_j^{s_j}$ is a matrix of size $\chi_{j-1} \times \chi_j$, $\chi_0 = 1 = \chi_n$, and $\chi_j = \chi$ for $0 < j < n$. The so-called bond dimension χ determines the maximum amount of entanglement the MPS can capture. For $\chi = 1$, the MPS represents an unentangled PS. Larger χ increases the expressivity of the MPS ansatz but also leads to higher computational costs. Throughout this work, we compute approximate ground states using the variational MPS implementation of the density matrix renormalization group (DMRG) algorithm [17–20] provided by the ITensor library [63,64]. For the quantum annealing Hamiltonians considered in this paper, we use matrix product operator (MPO) representations [17–20] within DMRG that are numerically calculated by ITensor routines; however, it is worth highlighting that optimal analytical MPO constructions for these types of Hamiltonians exist [65].

The MaxCut problem [23,24] is defined as follows. We assume that a graph is given with vertices $\{1, 2, \dots, n\}$, edges $\{(j, k) : w_{j,k} \neq 0\}$, and weights $w_{j,k}$ between vertices j and k . The goal is to find a partition of the vertices into two disjoint subsets (equivalently, an assignment of spin values $\in \{-1, 1\}$ to the vertices) that maximizes the weight of the cut, i.e., the sum of the weights associated with the edges crossing the partition. We consider three types of graphs: unweighted d -regular graphs (udR), where $w_{j,k} = 1$, weighted d -regular

graphs (wdR), where $w_{j,k}$ are drawn uniformly at random from $[0, 1]$, and Gset instances [29], where $w_{j,k} \in \{-1, 1\}$.

To evaluate performance, we use the so-called approximation ratio, defined as

$$r = \frac{C(\mathbf{b})}{C_{\max}}, \quad (1)$$

where $\mathbf{b} = (b_1, b_2, \dots, b_n)$ is a bitstring of bit values $b_j \in \{0, 1\}$ representing the spin values of the vertices, $C(\mathbf{b})$ is the total weight of edges crossing the partition corresponding to bitstring $\mathbf{b}$, and $C_{\max} = \max_{\mathbf{b} \in \{0,1\}^n} C(\mathbf{b})$ denotes the optimal cut value.

A common approach to solving combinatorial optimization problems with quantum methods maps the classical problem onto a quantum spin system. In this mapping, each classical bit b_j corresponds to the eigenvalues of the Pauli-Z operator σ^Z and the optimization objective is encoded in an Ising Hamiltonian $H_f = \sum_{(j,k)} w_{j,k} \sigma_j^Z \sigma_k^Z$. Here, the couplings $w_{j,k}$ define the edges of the underlying graph, making the correspondence between spin interactions and the original combinatorial structure explicit. The cut value C is directly related to the ground state energy E of H_f via $C = (\sum_{j < k} w_{j,k} - E)/2$ [66,67].

III. QUANTUM-INSPIRED ITERATED LOCAL SEARCH

To approximate the ground state of the Ising Hamiltonian within the MPS framework, we employ techniques inspired by ILS [5] and quantum annealing [6–10]. In particular, we consider an interpolation between the initial Hamiltonian $H_i = -\sum_{j=1}^n \sigma_j^X$ with Pauli-X operator σ^X and the problem Hamiltonian:

$$H(\lambda) = (1 - \lambda)H_i + \lambda H_f, \quad \lambda \in [0, 1]. \quad (2)$$

Using the Hamiltonian of Eq. (2) at a fixed value of λ , we perform an iterative procedure indexed by ι , where each iteration consists of a DMRG optimization followed by a perturbation step, in analogy with ILS. Starting from a randomly initialized MPS with bond dimension χ , we perform DMRG sweeps at the chosen λ until convergence. Once converged, we generate a spin configuration (bitstring solution) $\mathbf{b}$ according to the probability distribution encoded in the MPS. The configuration is constructed sequentially by selecting, at each site, the spin value most probable given the previously chosen spins—a greedy procedure similar to that used in Refs. [35,68]. We evaluate the corresponding energy with respect to the spin Hamiltonian H_f and keep the bitstring if it yields the lowest energy found so far.

To enable further progress and prevent the DMRG optimization from reconverging to the same local minimum, we perturb the converged MPS by randomly applying σ^X operators to a fraction p of the spins, where p denotes the perturbation strength. This is realized by randomly selecting a fraction p of the MPS tensors one after another and, for each selected tensor at site j , by assigning

$$A_j^{s_j} = A_j^{-s_j}, \quad (3)$$

where $-0 = 1$ and $-1 = 0$. The perturbed state then serves as the initial condition for iteration $\iota + 1$, and the process is

repeated. A more detailed description of the MPS algorithms is given in Appendix D.

In all figures, we report the best energy obtained up to iteration ι .

In Fig. 2(a), we assess the performance of QiILS as a function of λ , using a fixed perturbation value and the two bond dimensions $\chi = 1$ and 2. For nearly all values of λ , the performance systematically improves with both the iteration index ι and the bond dimension χ . Significant improvements over the traditional ILS baseline ($\lambda = 1 = \chi$) appear when $\lambda \neq 1$.

In Fig. 2(b), we compare the performance scaling with the number of iterations and with the bond dimension. For smaller problems ($n = 50$), both approaches yield comparable performance, with relative errors around $10^{-2.5}$. However, as the problem size increases, their behavior diverges. For larger instances ($n = 200$), the iterative method ($\chi = 1$) achieves relative errors below 10^{-2} that decrease systematically with ι , whereas increasing the bond dimension (with only one iteration) performs worse. Since the bond dimension dominates the computational cost of a DMRG sweep, scaling as $O(\chi^3)$ (see Appendix D), and its benefits diminish with increasing problem size, an iterative procedure with $\chi = 1$ offers a more efficient strategy for larger MaxCut instances.

While u3R graphs are analyzed in Figs. 2(a) and 2(b), analogous results for w3R graphs are provided in Appendix E, leading to similar conclusions.

The performance of QiILS is highly sensitive to the perturbation level. A very small perturbation may prevent the state from escaping a local minimum, while an excessively large one effectively randomizes the configuration and has a high probability of undoing any progress made toward the ground state. As shown in Fig. 2(c), for unweighted graphs, a perturbation strength of $p = 0.5$ yields the best performance. For the weighted graphs studied, moderate perturbations perform better in the short term—with $p = 0.3$ giving the lowest error for a small number of iterations—whereas smaller perturbations ($p = 0.1$) exhibit better scaling and ultimately achieve the best performance as the number of iterations increases.

IV. QUANTUM-INSPIRED ITERATED LOCAL SEARCH FOR $\chi = 1$

The strong performance of QiILS at $\chi = 1$ motivates the development of a dedicated PS version of the method, which can be implemented more efficiently. We start with a random PS of the form $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$, where

$$|\psi_j\rangle = \cos(\theta_j)|0\rangle + \sin(\theta_j)|1\rangle, \quad \theta_j \in \left[0, \frac{\pi}{2}\right]. \quad (4)$$

For the energy expectation value $\langle \Psi | H | \Psi \rangle$ as a function of θ_j , assuming all other angles θ_k for $k \neq j$ are fixed, we obtain the expression

$$\begin{aligned} E(\theta_j) &= A \cos(2\theta_j) - B \sin(2\theta_j) \\ &= -\sqrt{A^2 + B^2} \sin[2\theta_j - \text{atan2}(A, B)], \end{aligned} \quad (5)$$

where $A = \lambda a$,

$$a = \sum_{k < j} w_{k,j} \cos(2\theta_k) + \sum_{k > j} w_{j,k} \cos(2\theta_k), \quad (6)$$

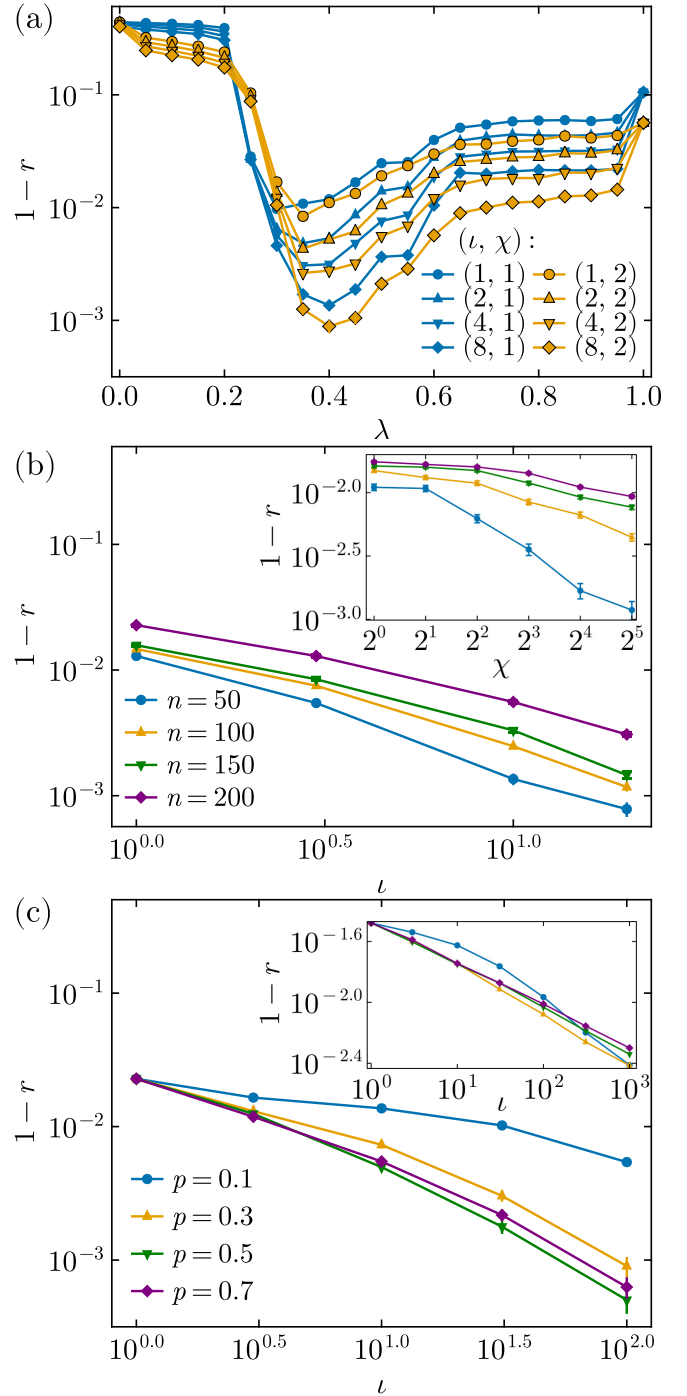


FIG. 2. Study of QiILS hyperparameters. Performance is measured by the average relative error $1-r$, averaged over 1000 random graph instances. (a) For u3R graphs of size $n=50$, we consider various values of λ , numbers of iterations ι , and bond dimensions χ . (b) For u3R graphs of different sizes, we study the performance as a function of the iteration index ι using PS ($\chi=1$) and fixing $\lambda=0.4$. The inset shows the performance for the same u3R graphs when varying χ while fixing the iteration index to $\iota=1$. (c) For $n=200$ and several perturbation strengths, we compare the performance for u3R graphs (main panel) and w3R graphs (inset). (Fixed hyperparameters: sweeps = 80; in panels (a) and (b) $p=0.5$; in panel (c) $\lambda=0.4$ for u3R and $\lambda=0.5$ for w3R).

and $B = 1 - \lambda$. Equation (5) is minimized by

$$\theta_j^{\text{new}} = \frac{\pi}{4} + \frac{1}{2} \text{atan2}(A, B). \quad (7)$$

We update the angles sequentially, going through each θ_j in turn. Once all angles have been updated in this manner, we refer to the full pass as a sweep. After each sweep, we check for convergence using the condition

$$\max_j |\theta_j^{\text{new}} - \theta_j| < \frac{\epsilon}{n} \sum_{j=1}^n \left| \theta_j^{\text{new}} - \frac{\pi}{4} \right|, \quad (8)$$

where ϵ is a user-defined tolerance. Once convergence is achieved, we obtain a bitstring by projecting the PS onto the computational basis:

$$b_j = \text{round}\left(\frac{2\theta_j}{\pi}\right) \in \{0, 1\}, \quad (9)$$

where the rounding function assigns $b_j = 0$ for $\theta_j \leq \pi/4$ and $b_j = 1$ otherwise. The energy associated with this configuration is evaluated with respect to the spin Hamiltonian H_f and recorded if it yields the lowest value obtained so far. In this setting, the subsequent perturbation, implemented by randomly applying σ^X operators, corresponds to updating the angles as $\theta_j^{\text{new}} = -\theta_j + \pi/2$.

V. PERFORMANCE MECHANISM

Let us investigate the mechanism underlying the performance of QiLS. Specifically, let us address the question of why there exists a λ regime—visible, e.g., in Fig. 2(a)—where the algorithms perform best. We show that this regime corresponds to values of λ for which the optimization landscape contains only a small number of minima. At the same time, a large fraction of these minima map, after rounding according to Eq. (9), to the lowest-energy spin configurations. Moreover, the basins of attraction associated with these optimal-rounding minima are large and well conditioned. Together, these conditions enable QiLS to rapidly converge to high-quality solutions. In the following, we present evidence supporting the proposed mechanism.

For 50 randomly generated u3R graphs of size $n = 50$, we probe the optimization landscape as a function of λ . To this end, we minimize the energy using SciPy's Sequential Least Squares Programming (SLSQP) optimizer. We use 1.2×10^4 randomly initialized spin configurations as starting points. The optimization is bounded such that all angles remain within $[0, \pi/2]$. We employ a stopping criterion of $\text{ftol} = 10^{-8}$ and allow up to $\text{maxiter} = 5 \times 10^3$ iterations per initialization. The ground-state energy E^* is found using an exact solver. We denote quantities associated with minima that, after rounding, yield E^* with a superscript $*$.

For each set of minima identified, we record the total number of distinct minima N_{minima} . We also record the size N_{minima}^* of the subset of those distinct minima that, after rounding, yield configurations with energy E^* . These data, including their mean values over the u3R graph instances, are shown in Fig. 3(a). We observe that, as λ increases, there is a transition from a single global minimum to multiple minima, many of which round to spin configurations with energy E^* . Minima

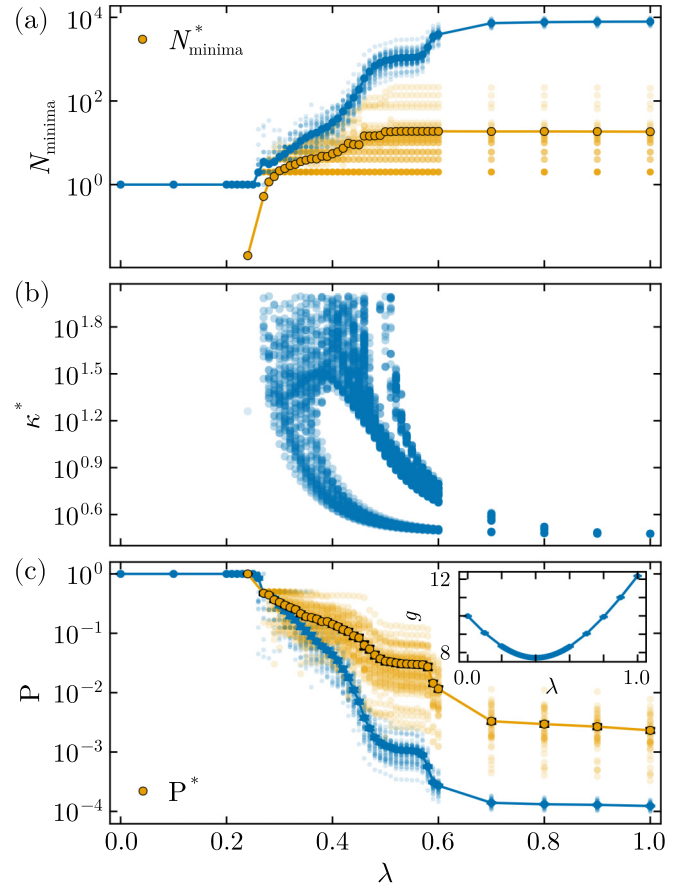


FIG. 3. Optimization landscape analysis. Shown are (a) the total number N_{minima} of distinct minima identified and the size N_{minima}^* of the subset yielding optimal energies E^* after rounding; (b) the condition number κ^* of the landscape evaluated at the best minima, capped at a maximum value of 100; and (c) the probability that the randomly initialized SLSQP optimizer converges to each distinct minimum, where the inset shows the mean expectation value of the gradient norm, g . Additional details are discussed in the main text.

occur in symmetry-related pairs connected by the transformation $\theta_j \rightarrow -\theta_j + \pi/2$. After the transition, N_{minima} grows exponentially, whereas N_{minima}^* increases at a much slower rate.

For each of the N_{minima}^* distinct optimal-rounding minima, we report the condition number κ^* of the corresponding Hessian matrix in Fig. 3(b). Immediately after the transition, these minima exhibit large condition numbers. As λ increases further, the condition numbers decrease rapidly, such that these minima become well conditioned in the region of λ where QiLS and QiGS perform best. For the u3R graphs used in this analysis, increasing λ leads to the appearance of many additional distinct optimal-rounding minima. These newly emerging minima contribute to N_{minima}^* and are initially characterized by large condition numbers that typically decrease as λ increases.

For each distinct minimum, we also compute the probability, P or P^* , that the randomly initialized optimizer converges to that minimum. These probabilities and their mean values, averaged over all u3R graphs, are shown in Fig. 3(c). While the mean probability across all minima is proportional to

TABLE I. Performance comparison for Gset problems [29]. For each graph and algorithm, we perform ten trials; i.e., we run each algorithm starting from ten different initial states. In each trial, the considered algorithm is run for 1000 iterations. The subscripts “best”, “avg”, and “solved” stand for the best solution obtained, the average solution value achieved, and the fraction of times the best-known solution was obtained, respectively. (Fixed hyperparameters: for QiLS, sweeps = 200, $p = 0.3$; for ILS, sweeps = 200, $p = 0.2$; and for LQA, $\gamma = 0.5 = \eta$).

Graph	Solution	QiLS _{best}	QiLS _{avg}	QiLS _{solved}	LQA _{best}	LQA _{avg}	LQA _{solved}	ILS _{best}	ILS _{avg}	ILS _{solved}	GCS _{best}	GCS _{avg}	GCS _{solved}
G1	11 624	11 624	11 623.6	9/10	11 624	11 621.2	1/10	11 622	11 614	0/10	11 624	11 591.4	1/10
G2	11 620	11 620	11 617	5/10	11 608	11 600.8	0/10	11 612	11 606.5	0/10	11 600	11 586.5	0/10
G3	11 622	11 622	11 619.6	6/10	11 621	11 605.7	0/10	11 615	11 608.9	0/10	11 622	11 595.7	1/10
G4	11 646	11 646	11 643.44	6/10	11 646	11 630.4	1/10	11 638	11 632	0/10	11 641	11 608.6	0/10
G5	11 631	11 631	11 630.55	9/10	11 623	11 622.1	0/10	11 623	11 619.2	0/10	11 622	11 593.8	0/10
G6	2178	2178	2177.9	9/10	2175	2173	0/10	2176	2171.9	0/10	2160	2141.6	0/10
G7	2006	2006	2001.1	5/10	2001	1998	0/10	2005	1993.4	0/10	1994	1979.8	0/10
G8	2005	2005	2002.4	4/10	2005	1988.3	1/10	2001	1991.8	0/10	2005	1977.5	1/10
G9	2054	2054	2048.88	2/10	2045	2042.5	0/10	2052	2041	0/10	2045	2027.9	0/10
G10	2000	2000	1987.44	1/10	1993	1981	0/10	1997	1989.1	0/10	1994	1978.4	0/10

$N_{\minima}^{-1}$ as expected, the values of P^* for the optimal-rounding minima are larger on average.

The inset of Fig. 3(c) shows the mean expectation value of the gradient norm, g , averaged over the 50 u3R graphs. Here, the expectation value is estimated by randomly sampling 10^4 spin configurations. We conclude from Fig. 3(c) that the mean gradient expectation value remains sufficiently large across all values of λ . Therefore, the optimization landscape is navigable throughout the entire λ range.

VI. SELECTION OF HYPERPARAMETER λ

Ideally, one seeks to choose the value of λ that maximizes the rate of improvement per iteration. We propose two methods for selecting λ . The first is to construct a diagram similar to Fig. 2(a) to visually identify the optimal value. Alternatively, leveraging the convex profile observed in Fig. 2(a), one can determine λ using the so-called golden-section search (GSS) [69], a standard technique for locating the extremum of a function within a bounded interval. In this approach, different λ values are evaluated via GSS, and the value that maximizes the decay rate m in the fit $E(\iota) = c_0 + c_1 e^{-m \times \iota}$ is selected, where $E(\iota)$ denotes the mean energy at iteration ι averaged over multiple initial states, and c_0 and c_1 are fitting parameters.

VII. COMPARISON WITH QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

Figure 4 presents a performance comparison between QiLS and QAOA, using QAOA results taken from Fig. 4 of Ref. [15]. We compare the performance of QiLS, evaluated either as a function of the number of iterations ($\chi = 1$) or as a function of the bond dimension (for a single iteration), to that of QAOA, whose performance is quantified by the number of QAOA quantum circuit layers. Although these metrics are not directly comparable, they reflect analogous notions of computational effort and enable a meaningful assessment of relative efficiency. In the unweighted case [Fig. 4(a)], we observe an exponential decay in $1 - r$ with respect to both iteration count and bond dimension. Our method attains higher approx-

imation ratios than those reported in Ref. [15], successfully solving all 1000 instances in fewer than 15 iterations. For weighted graphs [Fig. 4(b)], QiLS also outperforms QAOA, achieving higher approximation ratios while exhibiting a comparable rate of improvement with increasing iterations.

VIII. GSET BENCHMARK

To assess the performance of our algorithms on larger problems, we consider additional Gset problems [29] with 800 variables. As in Fig. 1, we compare the algorithms QiLS, LQA, ILS, and GCS. We benchmark them in the context of solving Gset instances G1–G10, which are based on random graph instances. Table I contains our results. We observe that QiLS consistently outperforms the other solvers, in line with our conclusions drawn from Fig. 1 for G12.

IX. QUANTUM-INSPIRED ITERATED GLOBAL SEARCH

The QiIGS algorithm replaces the sequential QiLS update of Eq. (7) by a global update based on gradient descent:

$$\theta_j^{\text{new}} = \theta_j - \tau \frac{\partial E}{\partial \theta_j}, \quad (10)$$

where τ denotes the step size,

$$\frac{\partial E}{\partial \theta_j} = -2\lambda a \sin(2\theta_j) - 2(1 - \lambda) \cos(2\theta_j) \quad (11)$$

and a is defined in Eq. (6). This QiIGS formulation enables all angles to be updated in parallel, naturally lends itself to GPU parallelization, and therefore may be more efficient than QiLS in solving larger problem instances, involving tens of thousands of vertices or more.

Indeed, Fig. 5 demonstrates that for the G81 instance [29]—a toroidal weighted graph with $n = 20\,000$ vertices and weights in $\{-1, 1\}$ —QiIGS achieves accuracy comparable to that of QiLS while offering an order-of-magnitude speedup. For problem sizes up to 50 000 vertices, Fig. 5 (inset) shows that GPU acceleration enables QiIGS to achieve approximately constant per-iteration times as n grows.

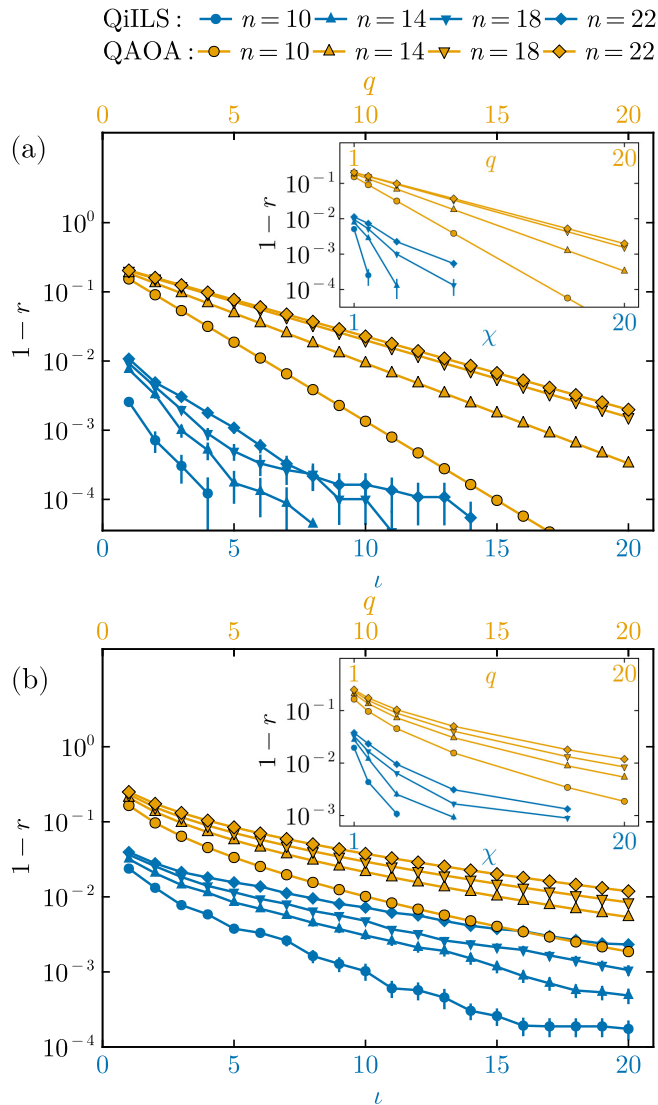


FIG. 4. Comparison with QAOA. Performance is measured by the average relative error $1 - r$, averaged over 1000 random graph instances. We consider u3R graphs [panel (a)] and w3R graphs [panel (b)]. The main panels show the performance of QiILS with $\chi = 1$ as a function of ι . The insets show the performance as a function of the bond dimension χ , using a single iteration ($\iota = 1$). Closed markers are results from Fig. 4 of Ref. [15], where the model function is $1 - r \propto \exp(-q/q_0)$ for unweighted graphs and $1 - r \propto \exp(-\sqrt{q/q_0})$ for weighted graphs, with q denoting the number of QAOA quantum circuit layers. The quantity on the horizontal axis (ι in the main panels or χ in the insets) plays an analogous role to the QAOA circuit depth q . (Fixed hyperparameters: sweeps = 80; in panel (a) $\lambda = 0.55$ and $p = 0.5$; in panel (b) $\lambda = 0.75$ and $p = 0.3$. Further details on the selection of λ are provided in Appendix F).

X. CONCLUSIONS

In this paper, we demonstrate that the combination of variational MPS methods—powerful classical tools for simulating certain quantum computations—with ILS—an established combinatorial optimization solver—can outperform state-of-the-art classical heuristics (ILS, LQA, and GCS) and a popular variational quantum algorithm (QAOA). From a

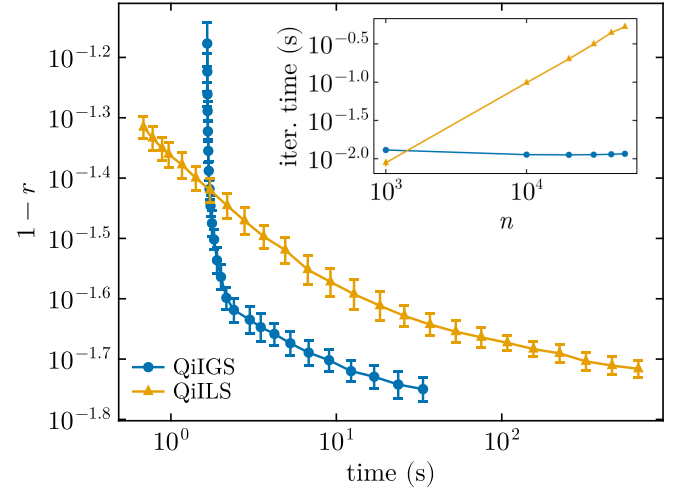


FIG. 5. Comparison of QiIGS with QiILS. In the main panel, we consider G81 [29] and performance is measured by the average relative error $1 - r$ as a function of wall-clock time (in seconds), computed over 10 000 iterations and averaged over 20 random initial states. The inset shows the average time per iteration for problem sizes $n = 1000, 10\,000, 20\,000, 30\,000, 40\,000, 50\,000$ across 20 random instances of u3R graphs. For QiIGS, angle updates were executed in parallel on an NVIDIA L4 GPU (24 GB VRAM, Ada Lovelace architecture) via Lightning AI. The CPU computations were performed on a 2023 MacBook Pro with an Apple M2 chip (8-core CPU, 10-core GPU) and 16 GB of unified memory, running macOS 15.6.1. (Fixed hyperparameters: sweeps = 200, $p = 0.15$, $\lambda = 0.35$, $\tau = 0.1$).

tensor-network perspective, we find that increasing the entanglement of the optimized MPSs systematically improves the results; however, optimizing unentangled PSs over an increasing number of ILS iterations can be significantly more efficient and yield better final solutions. From a combinatorial optimization viewpoint, the introduced QiILS and QiIGS algorithms strictly generalize ILS, recovering it as a special case for $\lambda = 1$, while enabling parallelization strategies that fundamentally extend beyond the sequential update structure of standard ILS. Finally, from a quantum-computing perspective, the proposed techniques are conceptually simple, naturally amenable to GPU parallelization, and therefore provide a useful classical framework to benchmark and challenge current and future quantum algorithms for combinatorial optimization.

ACKNOWLEDGMENTS

We thank Marco Ballarin and Lewis Wright for valuable feedback on the article. We are also grateful to Pranav Kalidindi for assistance with the GPU service.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this

research project. The data are available from the authors upon reasonable request.

APPENDIX A: OVERVIEW OF BENCHMARK OPTIMIZATION METHODS

Here, we provide an overview of the combinatorial optimization methods used as baselines for comparisons in the main text. Each subsection summarizes one method, highlighting its key features and implementation details. In the final subsection, we present hyperparameter exploration simulations used to select the settings employed in the benchmarks.

1. Iterated local search

Iterated local search [5] is a metaheuristic that repeatedly invokes a local search procedure, each time starting from a perturbed version of a previously found solution. In our case, the local search step consists of finding the ground state of the Hamiltonian H_f . Once a local minimum is reached, we apply a perturbation—specifically, a set of σ^X operators—to escape the local region of convergence and continue the search. This process is iterated until a satisfactory solution is obtained.

2. Local quantum annealing

Local quantum annealing [25] is an approach based on a parametrized product state $|\Psi\rangle$ and takes inspiration from quantum annealing. Starting in the ground state of the initial Hamiltonian $H_i = -\sum_j \sigma_j^X$, LQA sequentially minimizes the energy of the PS with respect to the Hamiltonian

$$H(t) = \left(1 - \frac{t}{T}\right) H_i + \frac{t}{T} \gamma H_f, \quad (\text{A1})$$

at a discrete set of times $0 < t_1 < t_2 < \dots < t_M = T$. In Eq. (A1), the ground state(s) of the Hamiltonian H_f encode(s) the optimal solution(s) to the considered combinatorial optimization problem, and the scalar γ rescales H_f . The energy minimization is performed via gradient descent with a step size η . In particular, at each time step t_j , the gradients of the energy expectation value $\langle \Psi | H(t_j) | \Psi \rangle$ are evaluated, and the parameters of the PS are updated once in the direction of steepest descent.

3. Generalized group-theoretic coherent states

The algorithm of Ref. [28], based on generalized group-theoretic coherent states [26,27], is a quantum-inspired variational method for combinatorial optimization. Similar to LQA, it employs a parametrized ansatz to mimic the time-dependent evolution of quantum annealing, but within a more expressive variational manifold. Starting with the PS $|+\rangle^{\otimes n} = [(|0\rangle + |1\rangle)/\sqrt{2}]^{\otimes n}$, the GCS ansatz is constructed by applying two types of unitaries: single-qubit rotations

$$\mathcal{U}(x) = \bigotimes_{j=1}^n \exp \left(-i \sum_{K \in \{X,Y,Z\}} x_{j,K} \sigma_j^K \right) \quad (\text{A2})$$

and two-qubit rotations

$$\mathcal{V}(B) = \exp \left(-i \sum_{j \neq k} B_{j,k} \sigma_j^Z \sigma_k^Z \right). \quad (\text{A3})$$

These operations are combined to produce the state

$$|\Psi(x, B, y)\rangle = \mathcal{U}(y) \mathcal{V}(B) \mathcal{U}(x) |+\rangle^{\otimes n}. \quad (\text{A4})$$

The variational parameters consist of two real $n \times 3$ matrices x and y , and a symmetric $n \times n$ matrix B . Setting all variational parameters equal to 0 initializes the state in $|+\rangle^{\otimes n}$. To emulate quantum annealing, we use the interpolating Hamiltonian (A1) of LQA, however, without the rescaling parameter γ . As in LQA, the total evolution time T is discretized into points $0 < t_1 < t_2 < \dots < t_M = T$. At each time t_j , the parameters (x, B, y) are updated via one steepest descent step using the gradient of $\langle \Psi(x, B, y) | H(t_j) | \Psi(x, B, y) \rangle$.

4. Hyperparameter exploration

In the following, we discuss the results for different settings of the hyperparameters for the methods LQA, QiLS, and ILS. Experiments are conducted on Gset instances G1, G6, and G12 [29], and the results are shown in Fig. 6.

For LQA, we focus on the hyperparameters γ and η , which control the rescaling of the problem Hamiltonian and the gradient descent step size, respectively. For each pair of hyperparameters, we fix one while sweeping over a range of values for the other to assess its influence on performance. Across the tested range, the hyperparameter γ does not lead to significant variations in performance. In contrast, the step size η has a noticeable effect: Large values (e.g., $\eta = 1.0$) can cause the method to diverge from the solution when the number of sweeps is large, as illustrated in Figs. 6(b), 6(f), and 6(j). For G1, we conclude from Figs. 6(a) and 6(b) that all studied hyperparameter choices yield similar systematic improvements of the results with increasing number of sweeps, except for $\eta = 0.8$ and 1.0 , where larger numbers of sweeps can lead to worse results. For G6 and G12, we conclude from Figs. 6(e), 6(f), 6(i), and 6(j) that across all hyperparameter settings the results plateau ($\eta < 1.0$) or deteriorate ($\eta = 1.0$) for larger numbers of sweeps.

For QiLS and ILS, we vary the perturbation strength p over a range of values. In both methods, the perturbation strength has a significant impact on performance, as shown in Figs. 6(c), 6(d), 6(g), 6(h), 6(k), and 6(l). We observe that achieving the best performance requires smaller perturbations in ILS than in QiLS, indicating that ILS is more sensitive to strong perturbations.

APPENDIX B: HYPERPARAMETERS USED

Table II summarizes the hyperparameters employed in all numerical simulations presented in this paper. Note that the value of sweeps in Table II is an upper bound, as we stop an iteration when the optimization has converged. Also, in Fig. 1, to find λ , we employ 50 random initializations, 25 iterations, and a maximum of 200 sweeps per iteration. For the LQA results of Fig. 1 and related to the Gset instances, we use $\gamma = 0.5 = \eta$. To find the optimal λ values for the Gset calculations,

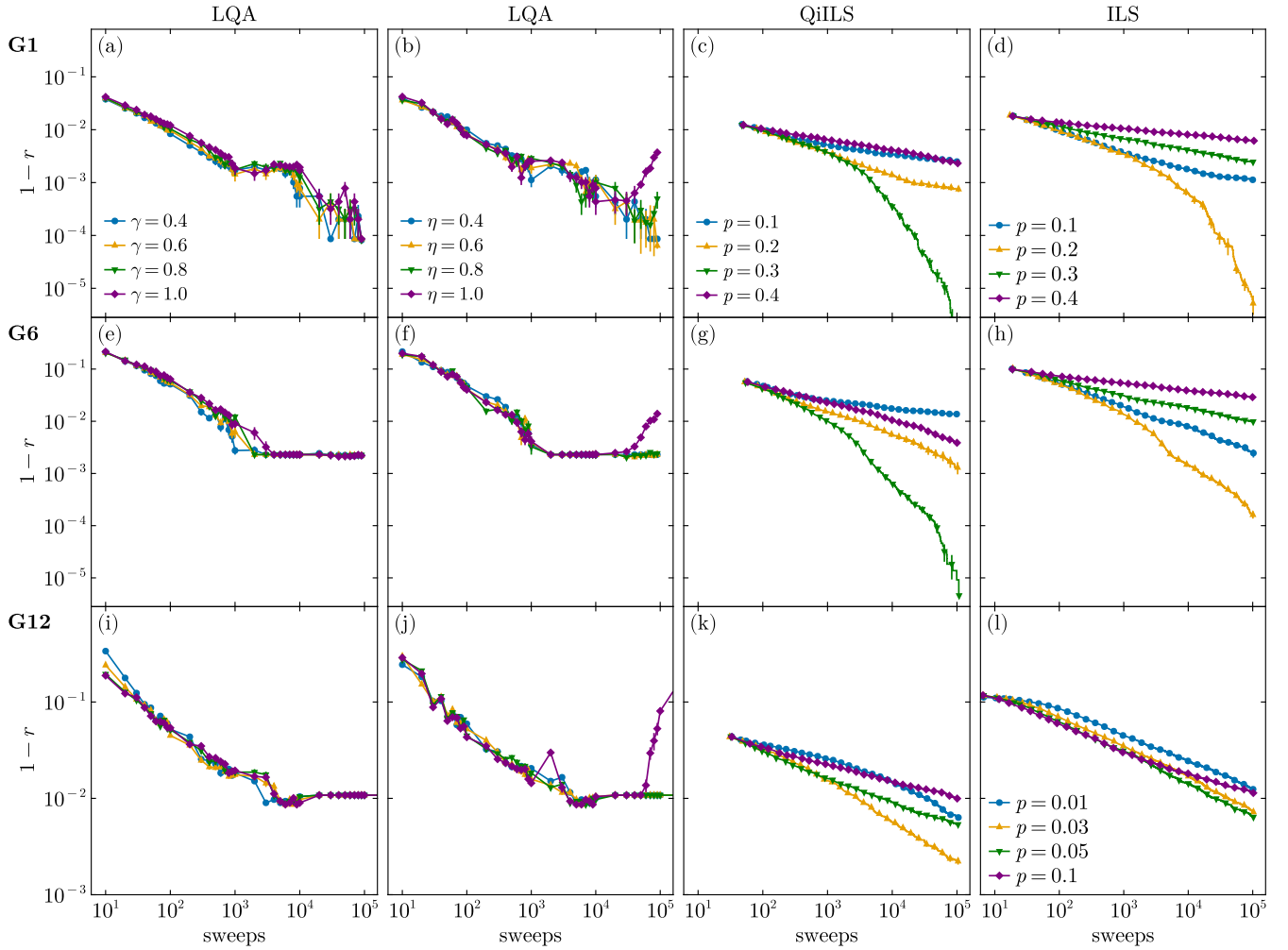


FIG. 6. Hyperparameter exploration for LQA, QiILS, and ILS applied to G1 [panels (a)–(d)], G6 [panels (e)–(h)], and G12 [panels (i)–(l)] [29]. Average performance is measured by the average relative error $1 - r$ and various hyperparameter choices are considered. In LQA, we fix $\eta = 0.5$, vary $\gamma \in \{0.4, 0.6, 0.8, 1.0\}$, and average over 20 [panels (a) and (e)] or 10 [panel (i)] random initial states. In LQA, we fix $\gamma = 0.5$, vary $\eta \in \{0.4, 0.6, 0.8, 1.0\}$, and average over 20 [panels (b) and (f)] or 10 [panel (j)] random initial states. In QiILS, we vary the perturbation strength $p \in \{0.1, 0.2, 0.3, 0.4\}$, use $\lambda = 0.4$ [panel (c)] or $\lambda = 0.3$ [panels (g) and (k)], and average over 100 [panels (c) and (g)] or 20 [panel (k)] random initial states. In ILS, we vary the perturbation strength $p \in \{0.1, 0.2, 0.3, 0.4\}$ [panels (d) and (h)] or $p \in \{0.01, 0.03, 0.05, 0.1\}$ [panel (l)], and average over 100 [panels (d) and (h)] or 20 [panel (l)] random initial states.

10 random initializations and 50 iterations were used for each graph.

APPENDIX C: TIMES PER SWEEP FOR EACH METHOD

Table III reports the median wall-clock time per sweep for each optimization method considered in Fig. 1. All timings were obtained under identical hardware and software conditions. We can see in Table III that QiILS and ILS exhibit identical per-sweep efficiency, whereas LQA and GCS show progressively higher computational overhead due to their more complex update mechanisms.

APPENDIX D: QUANTUM-INSPIRED ITERATED LOCAL SEARCH USING MATRIX PRODUCT STATES

Here, we describe the QiILS algorithm based on MPS in more detail than is provided in the main text. In

particular, we go beyond the main text by explaining MPS basics and the established MPS algorithms used in QiILS. We note that these topics are covered in review articles [17–20], which the reader can consult for further details.

The MPS-based QiILS procedure starts from a randomly initialized MPS and iterates the following three steps:

(1) The expectation value of the quantum annealing Hamiltonian is minimized using the MPS formulation of the density matrix renormalization group.

(2) From the resulting MPS, a bitstring is extracted as a candidate solution for the combinatorial optimization problem.

(3) The MPS obtained after the DMRG optimization is perturbed by applying Pauli-X matrices, after which the procedure returns to step 1.

Let us first explain the MPS and MPO ansätze. We assume open boundary conditions. An MPS of n qubits is

TABLE II. Numerical values of perturbation, sweeps, iterations, and λ used for all figures and tables of this paper.

	p	Sweeps	Iterations	λ
Fig. 1 (QiLS)	0.2	200		
Fig. 1 (ILS)	0.03	200		1.0
Fig. 2(a)	0.5	80	1, 2, 4, 8	
Fig. 2(b)	0.5	80		0.4
Fig. 2(b) (inset)		80	1	0.4
Fig. 2(c)	0.1, 0.3, 0.5, 0.7	80		0.4
Fig. 2(c) (inset)	0.1, 0.3, 0.5, 0.7	80		0.5
Fig. 4(a)	0.5	80		0.55
Fig. 4(a) (inset)		80	1	0.55
Fig. 4(b)	0.3	80		0.75
Fig. 4(b) (inset)		80	1	0.75
Fig. 5	0.15	200	10 000	0.35
Fig. 5 (inset)		200	1	0.35
Fig. 6(c)	0.1, 0.2, 0.3, 0.4			0.4
Fig. 6(d)	0.1, 0.2, 0.3, 0.4			1.0
Fig. 6(g)	0.1, 0.2, 0.3, 0.4			0.3
Fig. 6(h)	0.1, 0.2, 0.3, 0.4			1.0
Fig. 6(k)	0.1, 0.2, 0.3, 0.4			0.3
Fig. 6(l)	0.01, 0.03, 0.05, 0.1			1.0
Fig. 8(a)	0.3	80	1, 2, 4, 8	
Fig. 8(b)	0.3	80		0.6
Fig. 8(b) (inset)		80	1	0.6
Fig. 9(a)	0.5	80	1, 2, 4	
Fig. 9(b)	0.3	80	1, 2, 4, 8	
G1	0.3	200	1000	0.38
G2	0.3	200	1000	0.41
G3	0.3	200	1000	0.38
G4	0.3	200	1000	0.38
G5	0.3	200	1000	0.3
G6	0.2	200	1000	0.42
G7	0.2	200	1000	0.41
G8	0.2	200	1000	0.38
G9	0.2	200	1000	0.41
G10	0.2	200	1000	0.38
G12	0.2	200		
G81	0.15	200	10 000	0.35

given by

where $s_j \in \{0, 1\}$ and $\alpha_j \in \{1, 2, \dots, \chi\}$. An MPO has the form

$$|\Psi\rangle = \sum_{\{s_j, \alpha_j\}} A_{1, \alpha_1}^{s_1} A_{2, \alpha_1, \alpha_2}^{s_2} \cdots A_{n, \alpha_{n-1}}^{s_n} |s_1, s_2, \dots, s_n\rangle, \quad (\text{D1})$$

$$\mathcal{O} = \sum_{\{s_j, r_j, \alpha_j\}} B_{1, \alpha_1}^{s_1, r_1} B_{2, \alpha_1, \alpha_2}^{s_2, r_2} \cdots |s_1, \dots, s_n\rangle \langle r_1, \dots, r_n|, \quad (\text{D2})$$

TABLE III. Median CPU time in milliseconds (ms) per one sweep over all variables for various algorithms solving the G12 problem [29], where the median is calculated across 100 runs of each algorithm starting from different random initializations. The “median” indicates the typical time required per sweep, while the “IQR (interquartile range)” represents the variability between the 25th and 75th percentiles across runs. All experiments were performed on an Apple M2 CPU (8-core, 16 GB RAM) without GPU acceleration.

Algorithm	Median (ms)	IQR (ms)	Relative to QiLS	Remarks
QiLS	0.05	0.001	$\times 1.0$	Baseline (iterative local sweeps)
ILS	0.05	0.001	$\times 1.0$	Same per-sweep update cost as QiLS
LQA	0.34	0.00	$\times 6.8$	PyTorch-based; heavier local update rule
GCS	463.18	11.93	$\times 9260$	Variational updates; large matrix overhead

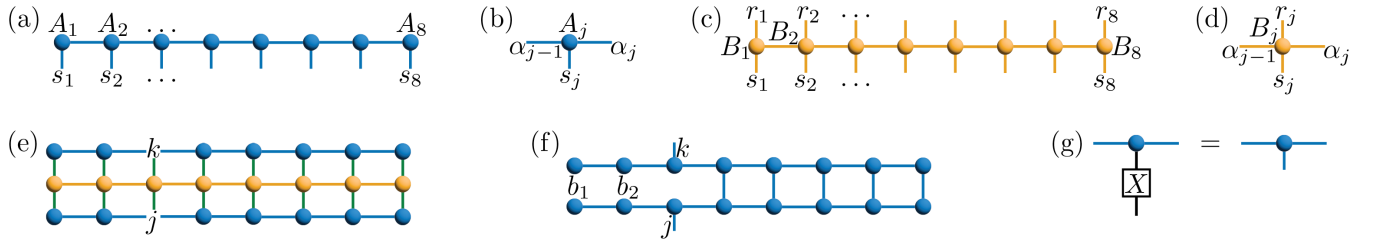


FIG. 7. Graphical tensor-network representations. Tensors are depicted as geometric shapes (spheres and rectangles) and their indices are represented by edges. Connecting two tensors by an edge indicates summation over the corresponding index. An open edge—attached to a tensor but not connected to another—represents an uncontracted degree of freedom. Shown are (a) an MPS for $n = 8$ qubits; (b) an internal MPS tensor A_j for $1 < j < n$; (c) an MPO for $n = 8$ qubits; (d) an internal MPO tensor B_j for $1 < j < n$; (e) the effective Hamiltonian $H_{j,k}^{\text{eff}}$ at site 3, assuming the MPO represents the Hamiltonian; (f) the reduced density matrix $\rho_{j,k}^{(3)}$, obtained after projecting the first two physical indices onto b_1 and b_2 , respectively; and (g) the application of a Pauli-X matrix to an MPS tensor.

where $r_j \in \{0, 1\}$. The MPS (MPO) ansatz consists of a product of n tensors, contracted via their so-called virtual indices $\{\alpha_j\}$. The indices $\{r_j, s_j\}$ are referred to as physical indices, as they correspond to the physical degrees of freedom of the qubits. The so-called bond dimension χ determines the expressivity of the ansatz as well as the computational cost of working with it. In practice, explicit mathematical equations are rarely used in the MPS and MPO formalism, as keeping track of the many indices is cumbersome. Instead, one typically adopts a graphical representation, illustrated in Figs. 7(a)–7(d) for an MPS and MPO.

The MPS variant of DMRG seeks to minimize the energy expectation value

$$E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \quad (\text{D3})$$

of a Hamiltonian H by variationally optimizing the MPS parameters. We consider the quantum annealing Hamiltonian of Eq. (2) and make use of its MPO representation, for which an analytical expression exists with minimum bond dimension $\tilde{\chi} \approx n/2$ [65]. The DMRG approach performs sweeps over the MPS tensors and for each tensor determines the optimal

entries that minimize E , while keeping all other tensors fixed. The optimal tensor elements are given by the eigenvector associated with the lowest eigenvalue of an effective Hamiltonian. This effective Hamiltonian is obtained from the tensor network corresponding to $\langle \Psi | H | \Psi \rangle$ by removing the considered tensor and its adjoint. Figure 7(e) shows the graphical notation for the effective Hamiltonian associated with the third MPS tensor. In practice, the effective Hamiltonian is not constructed explicitly; instead, only its action on a vector is evaluated. As a result, the computational cost per sweep of the DMRG method scales as $O(n\chi^3\tilde{\chi}) + O(n\chi^2\tilde{\chi}^2)$.

After convergence of the DMRG energy minimization, we extract a bitstring $(b_1, b_2, \dots, b_n)$ from the resulting MPS by choosing the most likely values of the physical MPS indices one after another. We begin with the first MPS site and compute the corresponding reduced density matrix

$$\rho^{(1)} = \text{tr}_{\{s_2, \dots, s_n\}}(|\Psi\rangle\langle\Psi|), \quad (\text{D4})$$

where $\text{tr}_S(\cdot)$ denotes the partial trace over the index set S . We assign to b_1 the bit value 0 or 1 maximizing $\langle b_1 | \rho^{(1)} | b_1 \rangle$. Next, we fix the first physical MPS index to b_1 and compute the

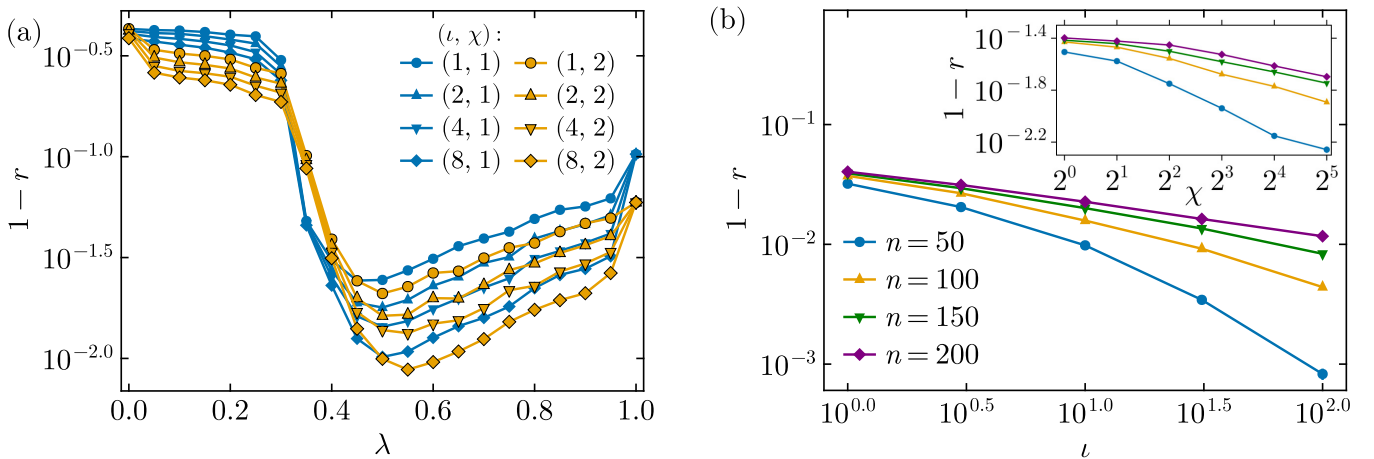


FIG. 8. Study of QiLS hyperparameters for w3R graphs. Performance is measured by the average relative error $1 - r$, averaged over 500 random w3R graph instances. (a) For $n = 50$, we consider several values of λ , numbers of iterations ι and bond dimensions χ . (b) For graphs of various sizes, we investigate the performance as a function of the iteration index ι using $\chi = 1$ at $\lambda = 0.6$. The inset shows the performance for the same graphs as a function of χ and fixing $\iota = 1$. (Fixed hyperparameters: sweeps = 80, $p = 0.3$).

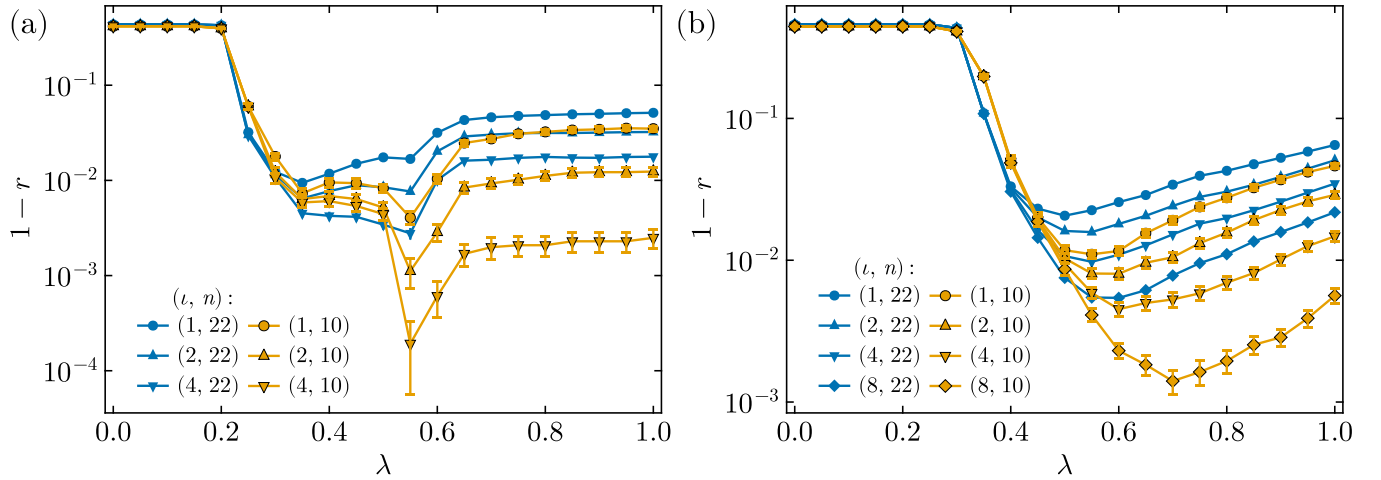


FIG. 9. Selection of λ . Performance is measured by the average relative error $1 - r$, averaged over 800 random instances, for different values of λ . (a) For u3R graphs, we find that the optimal value for λ is ≈ 0.55 . (b) For w3R graphs, the optimal value of λ is ≈ 0.75 . (Fixed hyperparameters: sweeps = 80; in panel (a) $p = 0.5$; in panel (b) $p = 0.3$).

reduced density matrix for the second MPS site,

$$\rho^{(2)} = \text{tr}_{\{s_3, \dots, s_n\}}[(\langle b_1 | \otimes \mathbb{1}) |\Psi\rangle \langle \Psi| (|b_1\rangle \otimes \mathbb{1})], \quad (\text{D5})$$

where $\langle b_1 |$ and $|b_1\rangle$ act on the first MPS site and the identity $\mathbb{1}$ acts on the remaining MPS sites. The bit value b_2 is chosen as the value maximizing $\langle b_2 | \rho^{(2)} | b_2 \rangle$. We continue by fixing the first and second physical MPS indices to b_1 and b_2 , respectively, and computing the reduced density matrix for the third MPS site,

$$\rho^{(3)} = \text{tr}_{\{s_4, \dots, s_n\}}[(\langle b_1, b_2 | \otimes \mathbb{1}) |\Psi\rangle \langle \Psi| (|b_1, b_2\rangle \otimes \mathbb{1})]. \quad (\text{D6})$$

Figure 7(f) depicts the tensor-network representation of $\rho^{(3)}$. We choose the most likely bit value for b_3 and continue the scheme until all $\{b_i\}$ have been assigned. Overall, the computational cost of this approach scales as $O(n\chi^3)$. The resulting bitstring $(b_1, b_2, \dots, b_n)$ serves as a candidate solution for the combinatorial optimization problem. We compute its energy with respect to the final Hamiltonian and keep track of the lowest energy achieved.

Having obtained a bitstring, we next perturb the previously converged MPS. We sequentially and randomly select a total of pn MPS tensors, where p denotes the perturbation strength. Each selected tensor is multiplied via its physical index with a Pauli- X matrix. The total computational cost of the perturbation step scales as $O(pn\chi^2)$. An example of a perturbation application is shown in Fig. 7(g). The perturbed MPS serves as the initial state for the subsequent DMRG run.

In MPS-based QiLS, this process—DMRG sweeps, bitstring extraction, perturbative updates—is iterated. The dominant computational cost arises from the DMRG sweeps and scales as $O(n^2\chi^3) + O(n^3\chi^2)$ per sweep. For fixed n , the leading asymptotic scaling is $O(\chi^3)$.

APPENDIX E: ADDITIONAL RESULTS FOR w3R GRAPHS

Here, we evaluate QiLS's performance across w3R graphs of sizes $n = 50, 100, 150$, and 200 . These graphs are the weighted counterparts of the u3R graphs considered in Figs. 2(a) and 2(b). Figure 8 shows our results. We can see in Fig. 8(a) that the improvement achieved for $\lambda \neq 1$ over $\lambda = 1$ is not as large as for our unweighted graph results of Fig. 2(a). However, the characteristic convex profile with a well-defined optimal λ region is clearly visible, as in Fig. 2(a). Figure 8(b) shows that increasing the number of iterations with a PS ansatz performs better than using a larger bond dimension, similar to what we can see in Fig. 2(b).

APPENDIX F: SELECTION OF λ FOR QAOA COMPARISON

To select the hyperparameter λ used in Fig. 4, we perform an analysis analogous to the one related to the diagram shown in Fig. 2(a). Such diagrams provide a practical guide for tuning λ for specific problem instances. Our results are shown in Fig. 9.

- [1] C. Zhang, Y. Wu, Y. Ma, W. Song, Z. Le, Z. Cao, and J. Zhang, A review on learning to solve combinatorial optimisation problems in manufacturing, *IET Collab. Intel. Manuf.* **5**, e12072 (2023).
- [2] V. Guilhaire and J.-K. Hao, Transit network design and scheduling: A global review, *Transp. Res. A* **42**, 1251 (2008).
- [3] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, A review of urban transportation network design problems, *Eur. J. Oper. Res.* **229**, 281 (2013).

- [4] Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. Man-Cho So, E. A. Jorswieck, and W. Yu, A survey of recent advances in optimization methods for wireless communications, *IEEE J. Sel. Areas Commun.* **42**, 2992 (2024).
- [5] H. R. Lourenço, O. C. Martin, and T. Stützle, Iterated local search: Framework and applications, in *Handbook of Metaheuristics*, edited by M. Gendreau and J.-Y. Potvin (Springer International Publishing, Cham, 2019), pp. 129–168.

- [6] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [7] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [8] A. Das and B. K. Chakrabarti, *Colloquium*: Quantum annealing and analog quantum computation, *Rev. Mod. Phys.* **80**, 1061 (2008).
- [9] T. Albash and D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [10] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, *Rep. Prog. Phys.* **83**, 054401 (2020).
- [11] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [12] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, *Rev. Mod. Phys.* **94**, 015004 (2022).
- [13] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: A review of methods and best practices, *Phys. Rep.* **986**, 1 (2022).
- [14] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [15] 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, *Phys. Rev. X* **10**, 021067 (2020).
- [16] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, Matrix product state representations, *Quantum Inf. Comput.* **7**, 401 (2007).
- [17] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, *Adv. Phys.* **57**, 143 (2008).
- [18] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys. (NY)* **326**, 96 (2011).
- [19] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, *Ann. Phys. (NY)* **349**, 117 (2014).
- [20] M. C. Bañuls, Tensor network algorithms: A route map, *Annu. Rev. Condens. Matter Phys.* **14**, 173 (2023).
- [21] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers? *Phys. Rev. X* **10**, 041038 (2020).
- [22] T. Ayrat, T. Louvet, Y. Zhou, C. Lambert, E. M. Stoudenmire, and X. Waintal, Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity, *PRX Quantum* **4**, 020304 (2023).
- [23] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, edited by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, The IBM Research Symposia Series, (Springer US, Boston, MA, 1972), pp. 85–103.
- [24] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* **42**, 1115 (1995).
- [25] J. Bowles, A. Dauphin, P. Huembeli, J. Martinez, and A. Acín, Quadratic unconstrained binary optimization via quantum-inspired annealing, *Phys. Rev. Appl.* **18**, 034016 (2022).
- [26] T. Guaita, L. Hackl, T. Shi, E. Demler, and J. I. Cirac, Generalization of group-theoretic coherent states for variational calculations, *Phys. Rev. Res.* **3**, 023090 (2021).
- [27] P. M. Schindler, T. Guaita, T. Shi, E. Demler, and J. I. Cirac, Variational ansatz for the ground state of the quantum Sherrington-Kirkpatrick model, *Phys. Rev. Lett.* **129**, 220401 (2022).
- [28] L. Fiononi and V. Savona, Entanglement-assisted variational algorithm for discrete optimization problems, *Commun. Phys.* **8**, 438 (2025).
- [29] Y. Ye, Gset dataset, 2003, <https://web.stanford.edu/%7Eyye/Gset/>.
- [30] M. C. Bañuls, R. Orús, J. I. Latorre, A. Pérez, and P. Ruiz-Femenía, Simulation of many-qubit quantum computation with matrix product states, *Phys. Rev. A* **73**, 022344 (2006).
- [31] J. A. Smolin and G. Smith, Classical signature of quantum annealing, *Front. Phys.* **2**, 52 (2014).
- [32] B. Bauer, L. Wang, I. Pižorn, and M. Troyer, Entanglement as a resource in adiabatic quantum optimization, [arXiv:1501.06914](https://arxiv.org/abs/1501.06914).
- [33] T. Hatomura and T. Mori, Shortcuts to adiabatic classical spin dynamics mimicking quantum annealing, *Phys. Rev. E* **98**, 032136 (2018).
- [34] S. Mugel, C. Kuchkovsky, E. Sánchez, S. Fernández-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orús, Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks, *Phys. Rev. Res.* **4**, 013006 (2022).
- [35] R. Sreedhar, P. Vikstål, M. Svensson, A. Ask, G. Johansson, and L. García-Álvarez, The quantum approximate optimization algorithm performance with low entanglement and high circuit depth, [arXiv:2207.03404](https://arxiv.org/abs/2207.03404).
- [36] M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, and M. J. Reagor, Entanglement perspective on the quantum approximate optimization algorithm, *Phys. Rev. A* **106**, 022423 (2022).
- [37] M. T. Veszeli and G. Vattay, Mean field approximation for solving QUBO problems, *PLoS One* **17**, e0273709 (2022).
- [38] M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, and M. J. Reagor, Calibrating the classical hardness of the quantum approximate optimization algorithm, *PRX Quantum* **3**, 040339 (2022).
- [39] G. Lami, P. Torta, G. E. Santoro, and M. Collura, Quantum annealing for neural network optimization problems: A new approach via tensor network simulations, *SciPost Phys.* **14**, 117 (2023).
- [40] J. Lopez-Piqueres, J. Chen, and A. Perdomo-Ortiz, Symmetric tensor networks for generative modeling and constrained combinatorial optimization, *Mach. Learn.: Sci. Technol.* **4**, 035009 (2023).
- [41] A. Batsheva, A. Chertkov, G. Ryzhakov, and I. Oseledets, PROTES: Probabilistic optimization with tensor sampling, in *37th Conference on Neural Information Processing Systems (NeurIPS)* (Curran Associates, Inc., Red Hook, New York, 2023).
- [42] J. Alcazar, M. Ghazi Vakili, C. B. Kalayci, and A. Perdomo-Ortiz, Enhancing combinatorial optimization with classical

- and quantum generative models, *Nat. Commun.* **15**, 2761 (2024).
- [43] J. Gardiner and J. Lopez-Piqueres, Tensor network estimation of distribution algorithms, *Neurocomputing* **680**, 133255 (2026).
- [44] J. Lopez Piqueres and J. Chen, Cons-training tensor networks: Embedding and optimization over discrete linear constraints, *SciPost Phys.* **18**, 192 (2025).
- [45] H. Nakada, K. Tanahashi, and S. Tanaka, Quick design of feasible tensor networks for constrained combinatorial optimization, *Quantum* **9**, 1799 (2025).
- [46] A. Morais, E. Osaba, I. Pastor, and I. Oregi, Comparative analysis of classical and quantum-inspired solvers: A preliminary study on the weighted max-cut problem, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'25 Companion* (Association for Computing Machinery, New York, NY, 2025), pp. 2449–2457.
- [47] D. Rattacaso, D. Jaschke, M. Ballarin, I. Siloi, and S. Montangero, Quantum algorithms for equational reasoning, *Sci. Adv.* **12**, eaec2736 (2026).
- [48] A. Garcia-Saez and J. I. Latorre, An exact tensor network for the 3SAT problem, *Quantum Inf. Comput.* **12**, 283 (2012).
- [49] J. D. Biamonte, J. Morton, and J. Turner, Tensor network contractions for #SAT, *J. Stat. Phys.* **160**, 1389 (2015).
- [50] Z. Zhu and H. G. Katzgraber, Do tensor renormalization group methods work for frustrated spin systems? [arXiv:1903.07721](https://arxiv.org/abs/1903.07721).
- [51] S. Kourtis, C. Chamon, E. R. Mucciolo, and A. E. Ruckenstein, Fast counting with tensor networks, *SciPost Phys.* **7**, 060 (2019).
- [52] J.-G. Liu, L. Wang, and P. Zhang, Tropical tensor network for ground states of spin glasses, *Phys. Rev. Lett.* **126**, 090506 (2021).
- [53] M. M. Rams, M. Mohseni, D. Eppens, K. Jałowiecki, and B. Gardas, Approximate optimization, sampling, and spin-glass droplet discovery with tensor networks, *Phys. Rev. E* **104**, 025308 (2021).
- [54] T. Hao, X. Huang, C. Jia, and C. Peng, A quantum-inspired tensor network algorithm for constrained combinatorial optimization problems, *Front. Phys.* **10**, 906590 (2022).
- [55] J.-G. Liu, X. Gao, M. Cain, M. D. Lukin, and S.-T. Wang, Computing solution space properties of combinatorial optimization problems via generic tensor networks, *SIAM J. Sci. Comput.* **45**, A1239 (2023).
- [56] N. Pancotti and J. Gray, One-step replica symmetry breaking in the language of tensor networks, [arXiv:2306.15004](https://arxiv.org/abs/2306.15004).
- [57] S. Yasuda, S. Sotobayashi, and Y. Minato, HOBOTAN: Efficient higher order binary optimization solver with tensor networks and PyTorch, [arXiv:2407.19987](https://arxiv.org/abs/2407.19987).
- [58] A. A. Gangat and J. Gray, Hyperoptimized approximate contraction of tensor networks for rugged-energy-landscape spin glasses on periodic square and cubic lattices, *Phys. Rev. E* **110**, 065306 (2024).
- [59] S. Patra, S. Singh, and R. Orús, Projected entangled pair states with flexible geometry, *Phys. Rev. Res.* **7**, L012002 (2025).
- [60] A. M. Ali, Explicit solution equation for every combinatorial problem via tensor networks: MeLoCoToN, [arXiv:2502.05981](https://arxiv.org/abs/2502.05981).
- [61] A. M. Dziubyna, T. Śmierzchalski, B. Gardas, M. M. Rams, and M. Mohseni, Limitations of tensor-network approaches for optimization and sampling: A comparison to quantum and classical Ising machines, *Phys. Rev. Appl.* **23**, 054049 (2025).
- [62] M. Tesoro, I. Siloi, D. Jaschke, G. Magnifico, and S. Montangero, Integer factorization via tensor-network Schnorr's sieving, *Phys. Rev. A* **113**, 032418 (2026).
- [63] M. Fishman, S. R. White, and E. M. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Phys. Codebases* **4** (2022).
- [64] M. Fishman, S. R. White, and E. M. Stoudenmire, Codebase release 0.3 for ITensor, *SciPost Phys. Codebases* **4-r0.3** (2022).
- [65] F. Fröwis, V. Nebendahl, and W. Dür, Tensor operators: Constructions and applications for long-range interaction systems, *Phys. Rev. A* **81**, 062337 (2010).
- [66] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt, An application of combinatorial optimization to statistical physics and circuit layout design, *Oper. Res.* **36**, 493 (1988).
- [67] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 5 (2014).
- [68] A. Chertkov, G. Ryzhakov, G. Novikov, and I. Oseledets, Optimization of functions given in the tensor train format, [arXiv:2209.14808](https://arxiv.org/abs/2209.14808).
- [69] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. (Cambridge University Press, Cambridge, UK, 2007).