

Motzkin-Straus Optimization on an Entropy-Computing Platform

PoJen Wang¹, Sutapa Samanta¹, Yuntai Song^{1,2},
Mohammad-Ali Miri¹

¹*Quantum Computing Inc (QCi), 5 Marine View Plaza, Hoboken,
07030, NJ, USA.

²*Department of Physics, University of Illinois Urbana-Champaign,
1110 West Green Street, Urbana, 61801, IL, USA.

Contributing authors: pwang@quantumcomputinginc.com;

Abstract

We introduce a framework for combinatorial optimization using sum-constrained continuous quadratic programs solvable by QCi’s Dirac-3S photonic entropy computer. This is enabled by the Motzkin-Straus theorem which provides a powerful bridge between discrete clique problems and optimization over the probability simplex. We demonstrate this framework’s versatility by solving constraint satisfaction problems, providing extensive benchmarks on the DIMACS suite. The Dirac-3S platform matches or outright leads two independently implemented classical baselines on more than four-fifths of the benchmark instances, reaching the best known solution on nearly all structured graph families, even outperforming both classical solvers on several of the largest instances tested. On the other hand, well-tuned classical continuous optimizers retain an edge only on the hardest planted-clique instances. This work establishes a viable pathway for solving combinatorial optimization problems using natively analog unconventional computing platforms, while positioning entropy computing as a competitive approach for navigating non-convex landscapes and providing rigorous baselines for an emerging computational paradigm.

Keywords: Combinatorial Optimization, Maximum Independent Set, Maximum Clique, Quantum Computing, Unconventional Computing, Entropy Computing

1 Introduction

Combinatorial optimization problems in the NP-hard class pose a formidable barrier to progress in fields ranging from logistics and bioinformatics to artificial intelligence. The Maximum Independent Set (MIS) problem, a canonical example, underscores this challenge, with applications central to resource allocation, scheduling, and network analysis [1]. As the scale of these problems grows, the limitations of traditional computing architectures become increasingly apparent. Exact methods can become computationally prohibitive on large or difficult instances, while heuristics, though practical, surrender the guarantee of optimality.

The transformation of the maximum clique problem into a continuous optimization framework via the Motzkin-Straus theorem [2] has opened a rich and challenging field of study (see [3] for a recent review spanning classical, neural-network, and quantum approaches). The historical development of classical solvers for this formulation is a testament to the persistent difficulty of its non-convex optimization landscape. The original 1965 formulation, while theoretically elegant, presents a landscape replete with spurious local optima [4, 5], which serve as traps for deterministic, gradient-based algorithms and lead to poor convergence. The computational difficulty is further underscored by recent results showing that even convex standard quadratic programs on the simplex become NP-hard under sparsity constraints [6]. In response, subsequent research introduced sophisticated countermeasures, each with its own inherent trade-offs. Regularization techniques [7, 8], for instance, were developed to reshape the objective function and eliminate these non-clique solutions, but this success came at the cost of introducing problem-dependent parameters that require careful and often computationally expensive tuning.

Alternative approaches sought to bypass this issue through complete problem reformulation. The symmetric rank-one nonnegative matrix approximation [9], for example, establishes a direct one-to-one correspondence between its global optima and the maximum clique, but in doing so, it elevates the problem’s complexity to a quartic objective that demands more specialized solvers. Even established, general-purpose algorithms for simplex-constrained optimization face significant hurdles [7]. The widely used projected gradient descent method, for instance, must be augmented with computationally intensive multi-restart strategies to effectively explore the solution space and escape the numerous local maxima. More recently, parallelized differentiable quadratic formulations have been proposed that incorporate clique-informed penalty terms and momentum-based gradient descent to improve convergence and exploration on the MIS problem, though these methods still rely on running many parallel initializations to compensate for the non-convex landscape [10]. Generative approaches based on discrete diffusion models have also been applied to MIS and maximum clique, sampling from the Boltzmann distribution of the problem Hamiltonian without requiring explicit likelihood computation [11]. Ultimately, the classical treatment of the Motzkin-Straus problem is characterized by a series of sophisticated yet compromised strategies, where overcoming the fundamental challenge of a complex landscape invariably introduces new costs in parameter sensitivity, algorithmic complexity, or computational overhead.

Recent advances in quantum computing have opened promising, albeit challenging, new avenues for tackling these problems [12–14]. However, recent benchmarks on Rydberg-atom hardware suggest that demonstrating a computational advantage for MIS may require scaling to thousands of atoms, as classical methods remain highly competitive on current problem sizes [15]. On superconducting hardware, quantum-enhanced greedy algorithms that use shallow QAOA circuits to guide classical heuristics have demonstrated scalable MIS solving on graphs larger than the physical qubit count [16]. Trapped-ion platforms have also been explored for MIS through digital-analog counterdiabatic protocols that reduce circuit depth [17]. Despite these advances across multiple platforms, a distinct approach has emerged in the form of entropy computing, as realized in QCI’s Dirac-3S system [18]. This paradigm marks a departure from mainstream quantum computing’s reliance on fragile, isolated quantum systems. Instead, it operates by conditioning a quantum reservoir to stabilize ground states through controlled dissipation and measurement-feedback loops [19–21]. Entropy computing paradigm harnesses quantum effects, such as fluctuations and shot noise, not as sources of error to be suppressed, but as computational resources to escape local minima. This offers a unique and potentially powerful approach for navigating the complex, non-convex landscapes characteristic of NP-hard problems, which is distinct from the algorithmic heuristics like multi-start framework [22].

This work explores the intersection of classical optimization theory and emerging entropy computing hardware, establishing several significant contributions to the field of quantum-enhanced combinatorial optimization. Our primary contribution lies in the design and implementation of Motzkin-Straus oracles operating on an entropy computer, as illustrated in Fig. 1. The oracle implementation exploits the natural quadratic structure of the Motzkin-Straus theorem, which provides an exact relationship $\omega(G) = \frac{1}{1-2M^*}$ between the clique number and the optimal value M^* of the continuous optimization problem. By implementing this optimization directly on the entropy computing’s Dirac-3S platform, we eliminate the approximation errors and computational overhead associated with discrete-to-continuous problem translations that characterize most quantum approaches to combinatorial optimization. Our experimental methodology provides a rigorous comparative performance analysis through comprehensive benchmarking on the standard DIMACS graph suite [23], employing highly optimized classical implementations including projected gradient descent with Armijo backtracking (Armijo-PGD) [24, 25] and Hexaly [26], a commercial general-purpose mathematical programming solver, as independent classical baselines for the same unregularized quadratic program. Among related photonic optimization platforms, time-multiplexed measurement-feedback systems such as coherent Ising machines [28–31] share architectural similarities with the Dirac-3S approach. Notably, a recent CIM demonstration solved the MIS problem on dense graphs with up to 40,000 nodes, outperforming optimized simulated annealing in time-to-solution for large instances [32]. The hardware architecture of the Dirac-3S platform and its natural mapping to the Motzkin-Straus formulation are described in detail in Section 3.1.

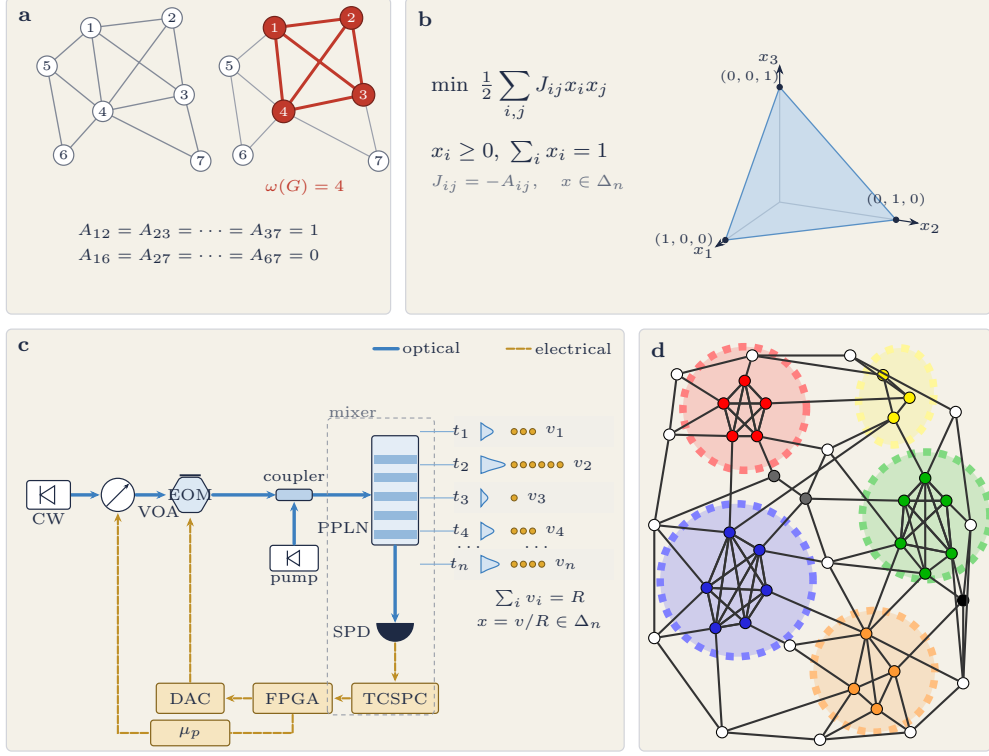


Fig. 1 A schematic illustration of Motzkin-Straus optimization through entropy computing. (a) The maximum clique problem. (b) The Motzkin-Straus mapping of the maximum clique problem to the corresponding quadratic optimization problem over non-negative variables subject to a sum constraint. (c) The entropy computing architecture utilizing photon number encoding in time bins respecting the non-negativity and sum constraints for faithful Motzkin-Straus optimization. (d) Applications of the maximum clique problem in community detection [27].

2 Max-clique Problem

2.1 Problem Formulation

The mathematical cornerstone of our work is the Motzkin-Straus theorem [2], which reveals a profound connection between discrete graph theory and continuous optimization. For a graph $G = (V, E)$ with adjacency matrix A , the theorem states:

$$\max_{x \in \Delta_n} \frac{1}{2} x^T A x = \frac{1}{2} \left(1 - \frac{1}{\omega(G)} \right) \quad (1)$$

where $\Delta_n = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq 0\}$ is the standard probability simplex and $\omega(G)$ is the size of the maximum clique in G . For a scaled simplex $\Delta_n^R = \{y \in \mathbb{R}^n :$

$\sum_{i=1}^n y_i = R, y_i \geq 0\}$, Eqn. 1 is modified to

$$\max_{y \in \Delta_n^R} \frac{1}{2} y^T A y = \frac{R^2}{2} \left(1 - \frac{1}{\omega(G)} \right) \quad (2)$$

This equivalence allows for the creation of oracles that can determine the clique number of a graph, a canonical NP-hard problem, by solving a continuous quadratic program. By leveraging the fundamental relationship $\alpha(G) = \omega(\overline{G})$, where $\alpha(G)$ is the independence number of G and $\overline{G}$ is its complement, we establish a complete algorithmic framework for the Maximum Independent Set problem through optimization on the complement graph.

2.2 Clique Extraction from Continuous Solutions

Given a graph $G = (V, E)$ with adjacency matrix A , we minimize $f = \frac{1}{2} y^T J y$, with $J = -A$. Let f^* be the minimum, then

$$f^* = -\frac{R^2}{2} \left(1 - \frac{1}{\omega(G)} \right), \quad \text{equivalently} \quad \omega(G) = \frac{1}{1 + 2f^*/R^2}. \quad (3)$$

Because the formula divides the energy by R^2 , it is invariant to the choice of R .

Any feasible solution therefore yields a continuous estimate for ω which takes non-integer value in general. The continuous estimate of ω might not be clique at all as there are spurious maximisers supported on non-cliques [33]. We must extract discrete integer value of max-cut from the continuous estimation of ω . We employ the following extraction methods. First, we use the Motzkin-Straus relationship in 3 to estimate the clique number directly from the continuous optimum by rounding to closest integer. Then the *support-based* method identifies the support set $S = \{i : x_i^* > \tau\}$ for a threshold $\tau > 0$ and extracts a valid clique from the subgraph induced by S . For small supports ($|S| \leq 20$), all subsets are enumerated to find the largest clique. For larger supports, greedy peeling which iteratively removes the vertex with the most non-adjacencies, and greedy building which iteratively adds the heaviest vertex adjacent to all selected vertices are applied, and the best result is retained. All candidate cliques are extended to maximal cliques before reporting. The full algorithmic details are provided in the supplementary information.

3 Solvers

3.1 Dirac-3S

The Dirac-3 platform [18] implements the entropy computing paradigm through a hybrid quantum-classical measurement-feedback system that encodes optimization variables in time-multiplexed photon qudits. Dirac-3S is the second generation Dirac-3 solver with higher variable count. A schematic of Dirac-3S hardware is illustrated in Fig. 1(c). The system minimizes a polynomial cost function E over non-negative variables v_i [18]:

$$E = \sum_i C_i v_i + \sum_{i,j} J_{ij} v_i v_j + \sum_{i,j,k} T_{ijk} v_i v_j v_k + \sum_{i,j,k,l} Q_{ijkl} v_i v_j v_k v_l + \sum_{i,j,k,l,m} P_{ijklm} v_i v_j v_k v_l v_m, \quad (4)$$

where C_i , J_{ij} , T_{ijk} , Q_{ijkl} , and P_{ijklm} are fully programmable weight tensors symmetric under permutation of indices, supporting up to fifth-order interactions. Two constraints arise naturally from the photonic hardware: (i) *non-negativity*: $v_i \geq 0$ for all i , since photon counts are intrinsically non-negative; and (ii) *sum constraint*: $\sum_i v_i = R$, where R is set by the total photon flux.

The Motzkin-Straus quadratic program maps to the Dirac-3S hardware with exceptional directness. The maximum clique formulation requires maximizing the quadratic objective $\frac{1}{2}x^T A x$ subject to the simplex constraint $x \in \Delta_n$, i.e., $\sum_i x_i = 1$ and $x_i \geq 0$. Each of these three requirements is natively satisfied by the photonic architecture without any encoding overhead:

1. The quadratic objective $x^T A x$ maps directly to the two-body interaction term $\sum_{i,j} J_{ij} v_i v_j$ in Eq. (4) by setting $J_{ij} = -A_{ij}$ and all higher-order tensors to zero. Here A_{ij} is the adjacency matrix
2. The simplex constraint $\sum_i x_i = 1$ is provided by the hardware-enforced photon conservation $\sum_i v_i = R$ with $R = 1$, arising from the fixed total photon flux in the feedback loop.
3. The non-negativity constraint $x_i \geq 0$ is intrinsic to photon counting. As a result no slack variables or penalty terms are required.

This stands in contrast to approaches based on the Ising Hamiltonian, where encoding the Motzkin-Straus problem would require penalty terms to enforce the simplex constraint, quadratization of any higher-order terms, and minor embedding to handle dense connectivity. The entropy computing paradigm thus provides a direct physical realization of the mathematical structure of the maximum clique problem, without any translation layer between the optimization formulation and the hardware.

Quantum fluctuations serve as a built-in mechanism for escaping local optima in the non-convex Motzkin-Straus landscape. The photon counting process follows Poisson statistics, producing a quantum fluctuation coefficient of $1/\sqrt{N}$, where N is the number of photons accumulated per time bin. Lower mean photon number μ increases the relative magnitude of shot noise, thereby enhancing stochastic exploration of the solution space. Empirical evidence from the original Dirac-3 experiments demonstrates that operating in the deep single-photon regime, where the probability of multi-photon events is below 1%, yields significantly improved solution quality on non-convex problems [18]. This noise-driven exploration is functionally analogous to temperature in simulated annealing, but arises directly from quantum measurement physics rather than from an artificial noise schedule. A trade-off exists: lower photon numbers improve exploration but increase the measurement time required for photon accumulation, creating a tunable balance between solution quality and computational speed.

In its current implementation, the Dirac-3S platform supports up to 9980 independent variables under summation constraints, with configurable sampling parameters `num_samples` $\in [1, 100]$ and `relaxation_schedule` $\in \{1, 2, 3, 4\}$ that control the rate of fluctuation annealing across feedback iterations. The experiments reported in this work use graphs with up to approximately 4000 vertices, which fall within the comfortable operating range of the device.

3.2 Classical Solvers

We selected two classical optimization approaches that represent the current performance frontier while offering distinct computational characteristics: Projected Gradient Descent with Armijo backtracking [24, 25], and Hexaly [26]. These methods were selected as they represent state-of-the-art techniques for the direct optimization of the Motzkin-Straus quadratic program [4, 34]. Other classical approaches to maximum clique and related graph problems include local-search and tabu-based heuristics [35, 36], parallel clique algorithms [37], differentiable quadratic formulations [10], and unsupervised graph neural network methods [38, 39]. These approaches employ different formulations or search strategies and are not evaluated here. Our objective is to compare Dirac-3S, Armijo-PGD, and Hexaly on the same unregularized Motzkin-Straus quadratic program. The comparison therefore assesses the tested solver configurations on a common formulation, rather than establishing their competitiveness against the strongest specialized maximum-clique algorithms.

3.2.1 Armijo-PGD

The Projected Gradient Descent method with Armijo line search provides our primary classical baseline, combining theoretical rigor with robust practical performance across diverse graph structures [9, 25]. The method addresses the non-convex Motzkin-Straus optimization landscape through adaptive step-size selection and strategic multi-restart initialization.

At iterate $x_k \in \Delta_n$ with gradient $g_k = \nabla f(x_k) = Ax_k$ for the maximization problem, the algorithm forms a trial point through gradient step followed by simplex projection:

$$\tilde{x}(\alpha) = \Pi_{\Delta_n}(x_k + \alpha g_k) \quad (5)$$

where Π_{Δ_n} denotes the Euclidean projection onto the probability simplex, computed efficiently in $O(n \log n)$ time using the sorting-based algorithm of Duchi et al. [40].

The Armijo condition provides adaptive step-size selection by accepting step size α when the sufficient decrease inequality [24, 25]

$$f(\tilde{x}(\alpha)) \geq f(x_k) + c_1 \alpha \nabla f(x_k)^T (\tilde{x}(\alpha) - x_k) \quad (6)$$

holds for fixed parameter $c_1 \in (0, 1)$. Starting from initial step size α_{init} , the backtracking procedure reduces the step by factor $\beta \in (0, 1)$ until the condition is satisfied. This adaptive mechanism guarantees monotone increase while automatically adjusting to the local geometry of the optimization landscape.

The multi-restart framework employs Dirichlet initialization with concentration parameter $\alpha = 0.5$ to sample diverse starting points across the probability simplex.

This initialization strategy addresses the sensitivity to starting points that characterizes non-convex optimization by ensuring comprehensive exploration of promising regions. Our implementation executes multiple independent optimization trajectories and selects the solution achieving the highest objective value.

Our implementation employs 100 independent restarts with maximum 20,000 iterations per restart. Implementation leverages JAX [41] for Just-In-Time compilation and automatic differentiation, enabling efficient vectorized execution of multiple restarts. The configuration employs parameters $c_1 = 10^{-4}$, $\beta = 0.5$, and $\alpha_{\text{init}} = 1.0$, chosen based on empirical validation across DIMACS benchmark instances to optimize the trade-off between convergence speed and solution quality.

3.2.2 Hexaly

Hexaly is a commercial general-purpose mathematical programming solver built on a portfolio of local-search and direct-search heuristics with exact components. We use Hexaly 15.0 [26] to solve the problem instances. We declare n continuous decision variables $y \geq 0$, one linear equality constraint $\sum_i y_i = R$, and minimise the objective. The quadratic form is supplied as a black-box external function together with its analytic gradient:

$$f_{\text{ext}}(y) = y^T J y, \quad \nabla f_{\text{ext}}(y) = 2Jy. \quad (7)$$

Supplying the gradient rather than letting Hexaly finite-difference helps it scale well at large number of variables.

Hexaly solver has a user defined `time.limit` parameter. For non-convex problems, Hexaly tends to keep running until the limit expires, irrespective of whether objective function is saturated to a value or not. We set the time limit to be 300 seconds for our experiments.

4 Results

4.1 Benchmark On DIMACS

We evaluate our approach on standard DIMACS benchmark graphs [23] that provide comprehensive coverage of the diverse structural characteristics encountered in real-world maximum clique problems. The Brock series consists of structured graphs with known optimal solutions spanning from brock200-1 through brock800-4, providing systematic scalability analysis across increasing problem sizes while maintaining known ground truth values for precise performance assessment.

The C-fat series encompasses dense clique-rich graphs including c-fat200-1 and c-fat500-10, designed specifically to challenge maximum clique algorithms through high edge density and multiple large cliques that create complex optimization landscapes with numerous competitive solutions. These instances test algorithmic ability to distinguish between near-optimal and optimal solutions in highly connected graphs.

Johnson graphs represent combinatorial designs with well-understood mathematical properties that enable theoretical analysis of algorithmic performance, providing instances where structural characteristics can be leveraged to validate theoretical predictions about optimization behavior. Hamming graphs encode error-correcting

code structures that arise naturally in coding theory applications, offering practical relevance beyond academic benchmarking.

Keller graphs provide geometric constructions with high chromatic numbers that challenge algorithms through complex constraint interactions arising from geometric embedding properties.

We evaluated the accuracy of our oracles on a broad selection of 75 DIMACS benchmark graphs, with sizes ranging from 28 to 4000 vertices. The primary goal of this benchmark is to understand the solution quality of the emergent entropy computing approach compared to highly refined classical solvers for the same continuous optimization problem. For this purpose, we compare the Dirac-3S 3.1 solver against Armijo-PGD 3.2.1 and Hexaly 3.2.2.

Table 1 reports the results for all 75 graphs. The ‘Best Known’ column lists the accepted optimal clique size from the literature or the best known lower bound, marked $\geq$, where optimality has not been proven. We report the integer clique size extracted from the solutions provided by the Dirac-3S, the classical Armijo-PGD, and the Hexaly oracles.

Table 1: Benchmark Results on Selected DIMACS Graphs. Bold indicates the solver that beat others. Star indicates when Dirac-3S solution is the best.

Graph	Vertices	Edges	Best Known	Dirac-3S	Armijo-PGD	Hexaly
C125.9	125	6,963	34	34	34	34
C250.9	250	27,984	44	44*	42	42
C500.9	500	112,332	≥ 57	53*	52	52
C1000.9	1000	450,079	≥ 68	63*	61	62
C2000.5	2000	999,836	16	15*	14	14
C2000.9	2000	1,799,532	≥ 80	70*	68	67
C4000.5	4000	4,000,268	18	15	15	12
DSJC500.5	500	62,624	13	12	12	12
DSJC1000.5	1000	249,826	15	14	13	14
brock200.1	200	14,834	21	20	20	20
brock200.2	200	9,876	12	10	10	10
brock200.3	200	12,048	15	13	13	13
brock200.4	200	13,089	17	16	15	16
brock400.1	400	59,723	27	24	23	24
brock400.2	400	59,786	29	24	24	23
brock400.3	400	59,681	31	25*	23	24
brock400.4	400	59,765	33	24	23	24
brock800.1	800	207,505	23	19	19	19
brock800.2	800	208,166	24	20	19	20
brock800.3	800	207,333	25	21*	19	19
brock800.4	800	207,643	26	20*	19	19
c-fat200-1	200	1,534	12	12	12	12
c-fat200-2	200	3,235	24	24	24	24
c-fat200-5	200	8,473	58	58	58	58
c-fat500-1	500	4,459	14	14	14	14
c-fat500-2	500	9,139	26	26	26	26
c-fat500-5	500	23,191	64	64	64	64
c-fat500-10	500	46,627	126	126	126	126
gen200-p0.9-44	200	17,910	44	38	39	38
gen200-p0.9-55	200	17,910	55	42	47	40
gen400-p0.9-55	400	71,820	55	51*	50	50

gen400-p0.9-65	400	71,820	65	54*	49	51
gen400-p0.9-75	400	71,820	75	52	53	44
hamming6-2	64	1,824	32	32	32	32
hamming6-4	64	704	4	4	4	4
hamming8-2	256	31,616	128	128	128	128
hamming8-4	256	20,864	16	16	16	16
hamming10-2	1024	518,656	512	512	512	512
hamming10-4	1024	434,176	≥ 40	36*	35	33
johnson16-2-4	120	5,460	8	8	8	8
johnson32-2-4	496	107,880	≥ 16	16	16	16
johnson8-2-4	28	210	4	4	4	4
johnson8-4-4	70	1,855	14	14	14	14
keller4	171	9,435	11	9	9	9
keller5	776	225,990	27	19*	18	16
keller6	3361	4,619,898	59	35*	33	31
p_hat300-1	300	10,933	8	8	8	8
p_hat300-2	300	21,928	25	25	25	25
p_hat300-3	300	33,390	36	36*	34	35
p_hat500-1	500	31,569	9	9	9	9
p_hat500-2	500	62,946	36	36	36	36
p_hat500-3	500	93,800	50	49	50	49
p_hat700-1	700	60,999	11	11	11	9
p_hat700-2	700	121,728	44	44	44	44
p_hat700-3	700	183,010	62	62	62	62
p_hat1000-1	1000	122,253	10	10	10	9
p_hat1000-2	1000	244,799	46	46*	45	45
p_hat1000-3	1000	371,746	68	64	66	64
p_hat1500-1	1500	284,923	12	11	11	10
p_hat1500-2	1500	568,960	65	65	65	63
p_hat1500-3	1500	847,244	94	92	92	92
sanr200_0.7	200	13,868	18	18	18	18
sanr200_0.9	200	17,863	42	41	41	41
sanr400_0.5	400	39,984	13	13*	12	12
sanr400_0.7	400	55,869	21	21*	20	20
san200_0.7_1	200	13,930	30	16	23	16
san200_0.7_2	200	13,930	18	12	15	12
san200_0.9_1	200	17,910	70	47	50	47
san200_0.9_2	200	17,910	60	42	60	60
san200_0.9_3	200	17,910	44	36	36	35
san400_0.5_1	400	39,900	13	7	13	7
san400_0.7_1	400	55,860	40	20	40	20
san400_0.7_2	400	55,860	30	15	30	15
san400_0.7_3	400	55,860	22	14	16	12
san400_0.9_1	400	71,820	100	100	100	100

Across the 75 graphs, all three solvers return the same clique size on 33 instances, and on 25 of those the common value is the best known one. On a further 13 instances two of the three solvers tie for the highest value. Dirac-3S and Armijo-PGD tie ahead of Hexaly on 7, Dirac-3S and Hexaly tie ahead of Armijo-PGD on 5, and Armijo-PGD and Hexaly tie ahead of Dirac-3S on 1. So on 46 of the 75 graphs no single solver stands alone at the top. On the remaining 29 graphs Dirac-3S achieves best on 17, Armijo-PGD on 12, and Hexaly on none. Altogether, Dirac-3S matches or exceeds both classical solvers on 62 of the 75 graphs, i.e., on more than four-fifths of the benchmark. Where Dirac-3S is the sole leader, its lead is only 1-3 vertices. Armijo-PGD's leads are likewise 1-2 vertices outside the planted-clique families, apart from gen200-p0.9-55 (5 vertices), but on several **san** instances it recovers the planted clique

outright (san400.0.5.1, san400.0.7.1, san400.0.7.2) with margins of up to 20 vertices. Dirac-3S reaches the best known value on 33 of the 75 graphs, Armijo-PGD on 33, and Hexaly on 26.

Among the 14 graphs with at least 1000 vertices, ties account for 7 of the 14. All three solvers agree on hamming10-2 and p_hat1500-3, Dirac-3S and Armijo-PGD tie ahead of Hexaly on C4000.5, p_hat1000-1, p_hat1500-1 and p_hat1500-2, and Dirac-3S and Hexaly tie ahead of Armijo-PGD on DSJC1000.5. Dirac-3S is the sole leader on the remaining large graphs C1000.9, C2000.5, C2000.9, hamming10-4, keller6 and p_hat1000-2, and Armijo-PGD is the sole leader only on p_hat1000-3. Dirac-3S is therefore at or above both classical solvers on 13 of the 14 large graphs. Taken together, an emerging hardware architecture is competitive with two independently implemented classical solvers on a substantial part of the benchmark. On the Brock series all solvers struggle, which is consistent with this difficulty being a property of the Motzkin-Straus optimization landscape on these graphs rather than an artifact specific to the entropy computing hardware or to the classical algorithm implementation. The planted-clique **gen** and **san** instances are also hard for all three solvers, but here Armijo-PGD’s many independent restarts give it a clear advantage over Dirac-3S and Hexaly.

Figure 2 summarizes these head-to-head counts over the full benchmark. The three groups of bars measure progressively relative notions of success. The first counts agreement with the literature, the second counts outright wins, and the third counts instances on which a solver is not beaten by either competitor. Dirac-3S and Armijo-PGD reach the best known value equally often, so the absolute measure alone does not separate them. The difference appears in the relative measures. Dirac-3S is the sole leader more often than Armijo-PGD (17 versus 12), and it is at or above both competitors on 62 instances, compared with 53 for Armijo-PGD and 39 for Hexaly.

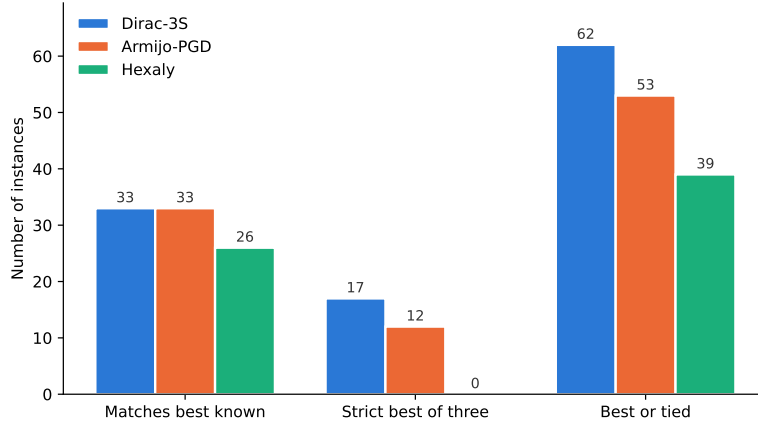


Fig. 2 Summary of solver performance on the 75 DIMACS benchmark graphs. For each solver, the bars show the number of instances on which it (i) *matches best known*, i.e. returns a clique size equal to the best known value (or best known lower bound) in Table 1; (ii) is the *strict best of three*, i.e. returns a clique strictly larger than those of both other solvers; and (iii) is *best or tied*, i.e. returns a clique at least as large as those of both other solvers, including ties. The counts in each category are taken directly from Table 1.

4.2 Energy Histogram of Specific Examples

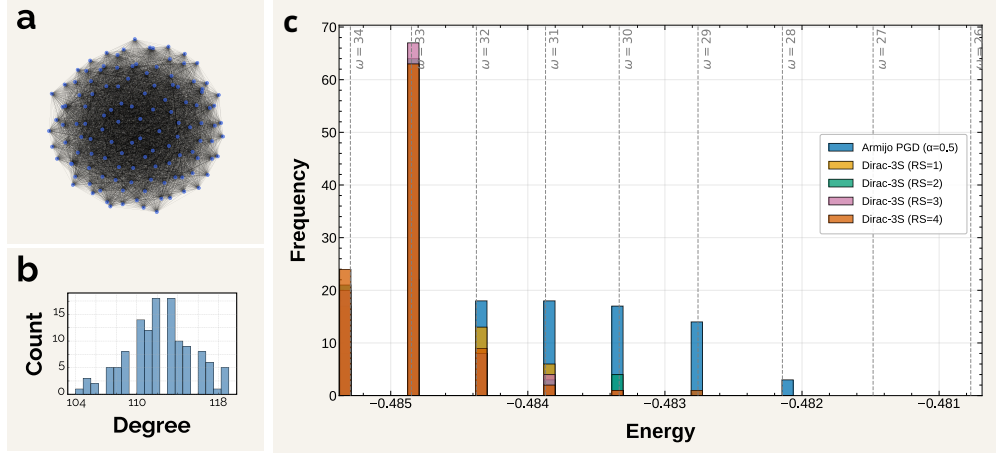


Fig. 3 Energy distribution of Dirac-3S and Armijo-PGD solutions for the DIMACS graph C125.9. (a) Visualization of the C125.9 graph. (b) Degree distribution of the C125.9 graph. (c) Histogram of the Motzkin-Straus energy values returned across all runs, colored by solver: Armijo-PGD with initial points sampled from a Dirichlet distribution with $\alpha = 0.5$, and Dirac-3S at each of its four `relaxation_schedule` (RS) settings. Vertical dashed lines mark the theoretical energy $-\frac{1}{2} \left(1 - \frac{1}{l}\right)$ predicted by the Motzkin-Straus formula for a clique of size l , with the corresponding l labeled above each line; the best known clique number for C125.9 is 34.

Figures 3 and 4 reveal a qualitative difference in how the two solvers’ solutions are distributed relative to the optimum. On the easy instance C125.9 (Fig. 3), Dirac-3S’s energy distribution is sharply concentrated at the theoretical energies corresponding to $l = 33$ and the best known clique size $l = 34$, with the large majority of runs across all relaxation-schedule settings landing on one of these two values. Armijo-PGD’s distribution is comparatively spread out, with a non-trivial fraction of restarts terminating at suboptimal energies corresponding to $l = 28$ through 34, so that a larger number of independent restarts is needed before the optimum is reliably recovered. The same pattern persists, and is more pronounced, on the harder C250.9 instance (Fig. 4): Dirac-3S’s mass is shifted toward the higher clique sizes near the optimum, with samples reaching all the way to the best known value $l = 44$, whereas Armijo-PGD’s distribution remains concentrated at the comparatively lower $l = 34 - 42$ range and does not reach $l = 44$ in the runs shown. This indicates that the entropy computing hardware’s measurement-feedback dynamics explore the Motzkin-Straus landscape in a way that concentrates solutions near the optimum, whereas Armijo-PGD’s gradient-based restarts are more broadly distributed across suboptimal basins and consequently require a larger sample size to reach the same solution quality.

Table 2 compares three solver paradigms on representative DIMACS instances spanning easy (C125.9), moderate (C250.9), and hard (brock400.2) difficulty levels. We run Dirac-3S for 100 `num_samples`, and report total time to run those 100 samples.

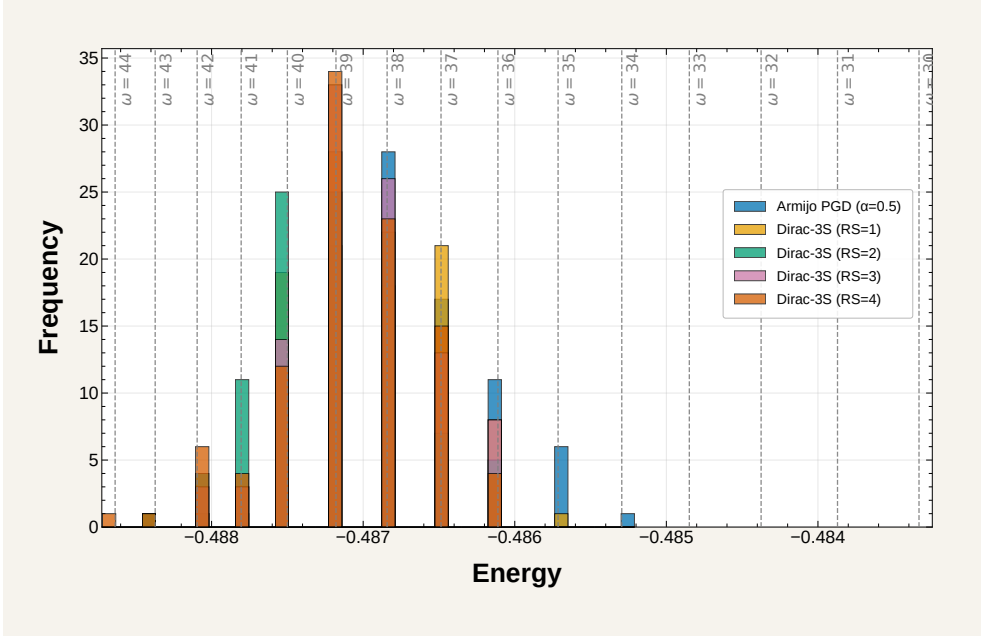


Fig. 4 Energy distribution of Dirac-3S and Armijo-PGD solutions for the DIMACS graph C250.9. Histogram of the Motzkin-Straus energy values returned across all runs, colored by solver: Armijo-PGD with initial points sampled from a Dirichlet distribution with $\alpha = 0.5$, and Dirac-3S at each of its four `relaxation_schedule` (RS) settings. Vertical dashed lines mark the theoretical energy $-\frac{1}{2}(1 - \frac{1}{l})$ predicted by the Motzkin-Straus formula for a clique of size l , with the corresponding l labeled above each line; the best known clique number for C250.9 is 44.

Hexaly is run with its fixed 300-second `time_limit`, which it consumes in full regardless of instance difficulty. The results reveal distinct performance profiles: all three solvers reach the optimum on the easy instance, C125.9; on the moderate instance, C250.9, only Dirac-3S reaches the optimal 44, while Armijo-PGD and Hexaly fall short; and on the hard instance, brock400_2, none of the three reaches the optimal 29.

5 Discussion

5.1 Graph Family Analysis

Different graph families exhibit varying oracle performance characteristics that reflect the underlying structure of the optimization landscape. Approximation ratios are computed against the best known clique size or best known lower bound where marked $\geq$ in Table 1, and are summarized by family in Table 3. Structured instances (Hamming, Johnson, c-fat) and the `p_hat` family are solved essentially exactly by all three solvers, with average ratios of 95 - 99%. Performance degrades on graphs with less exploitable structure. For example the dense C-series and the `sanr`/DSJC/Keller group are solved to 85 - 93% on average. The Brock series, constructed specifically to hide its maximum clique among low-degree vertices, drops further to 81 - 84%. The planted-clique `gen`

Table 2 Comparative Solver Performance on Representative DIMACS Instances

Solver	Graph	Clique Found / Best	Ratio	Avg. Time (s)
Armijo-PGD (GPU)	C125.9	34 / 34	100%	~15
Dirac-3S	C125.9	34 / 34	100%	~65
Hexaly	C125.9	34 / 34	100%	~300
Armijo-PGD (GPU)	C250.9	42 / 44	95.5%	~10
Dirac-3S	C250.9	44 / 44	100%	~120
Hexaly	C250.9	42 / 44	95.5%	~300
Armijo-PGD (GPU)	brock400_2	24 / 29	82.8%	~150
Dirac-3S	brock400_2	24 / 29	82.8%	~200
Hexaly	brock400_2	23 / 29	79.3%	~300

and **san** families are the hardest for Dirac-3S and Hexaly, averaging only 71%, whereas Armijo-PGD reaches 86.5% on this family. Dirac-3S attains the highest average ratio in five of the six families, with Armijo-PGD and Hexaly trailing by small margins. The exception is the planted-clique family, where Armijo-PGD leads by a wide margin; this single family is enough to give Armijo-PGD the highest average ratio over all instances (91.2% versus 89.2% for Dirac-3S and 87.4% for Hexaly). Apart from this, the ranking of family difficulty is broadly consistent across the three mechanistically unrelated solvers, which supports the conclusion that this difficulty gradient largely reflects the structure of the Motzkin-Straus landscape itself rather than a weakness specific to any one solver.

Table 3 Average Approximation Ratio by Graph Family. Ratios are computed against the best known clique size (lower bound where marked $\geq$ in Table 1).

Graph Family	Instances	Dirac-3S Avg. Ratio	Armijo-PGD Avg. Ratio	Hexaly Avg. Ratio
Structured (Hamming, Johnson, c-fat)	17	99.4%	99.3%	99.0%
Dense random (C-series)	7	92.9%	90.3%	88.0%
Brock (200–800)	12	84.3%	81.3%	82.7%
Planted clique (gen, san)	15	71.0%	86.5%	70.8%
p_hat	15	98.8%	98.6%	95.8%
Random (sanr, DSJC, keller)	9	88.3%	85.4%	84.9%
All instances	75	89.2%	91.2%	87.4%

5.2 Interpreting Benchmark Results Through Platform Characteristics

The benchmark results can be interpreted through the lens of the solver’s property. Classical approaches handle the non-convex Motzkin-Straus landscape through regularization [7], alternative formulations [9], population-level dynamics [42, 43], or

general-purpose local-search portfolios such as Hexaly, each introducing complexity through parameter tuning or convergence trade-offs. The Dirac-3S platform instead leverages quantum shot noise as an exploration mechanism, providing a physically distinct alternative to multi-restart strategies. Whether this constitutes a qualitative advantage or is better understood as a hardware-implemented form of stochastic optimization remains an open question [44]. Recent work has shown that for certain photonic approaches to graph problems, efficient classical algorithms can replicate quantum sampling distributions [45], reinforcing the importance of benchmarking against strong classical baselines – a role that both Armijo-PGD and Hexaly serve in this study, since they are built on entirely different optimization principles (multi-restart projected gradient ascent versus a black-box heuristic portfolio) yet converge on similar qualitative conclusions about which graph families are hard.

Our results provide some preliminary evidence for this view. Dirac-3S’s measurement-feedback dynamics produce diverse solution candidates without explicit multi-start procedures, and the platform is the outright best of the three solvers on a handful of instances that are not confined to small graphs, including six with at least 1000 vertices (Table 1). On these larger graphs, Dirac-3S is at or above the level of the other two solvers on 13 of the 14 instances, more often than either classical baseline. This asymmetry suggests that the platform’s exploration dynamics may complement classical gradient-based methods on some large instances in a way that a third, unrelated classical heuristic does not reproduce. The margins involved are typically only one or two vertices, however, so this evidence is suggestive rather than conclusive, and part of the apparent advantage may simply reflect the generic benefit of consulting a second, independent solver. Characterizing when and why the platform’s physical dynamics offer practical benefits over gradient-based methods, particularly near energy barriers, is a key direction for future work.

5.3 Assessment and Practical Considerations

Our benchmarks show that the Dirac-3S platform already matches or leads two independently implemented classical solvers – Armijo-PGD and Hexaly – on more than four-fifths of the DIMACS instances and attains the highest average approximation ratio in five of six graph families. The remaining gap in raw solution quality is confined mainly to the planted-clique instances, which places Dirac-3S’s overall average approximation ratio between Armijo-PGD’s and Hexaly’s (Table 3).

Compared to quantum annealing approaches, the entropy computing architecture offers a significant practical advantage for dense graph problems: the Motzkin-Straus quadratic program is implemented directly on the hardware without embedding overhead. On quantum annealers such as D-Wave, embedding dense graph structures onto sparse hardware connectivity graphs introduces substantial overhead that degrades solution quality [46]. The Dirac-3S platform avoids this entirely by satisfying the simplex constraint at the hardware level (see Section 3.1).

Furthermore, as a near-term technology deployed on existing hardware rather than future fault-tolerant systems, entropy computing provides a concrete platform for

iterative benchmarking and improvement. This positions it as complementary to gate-based quantum algorithms, which require extensive error correction infrastructure that may take years to mature.

5.4 Current Limitations and a Roadmap for the Future

The main limitations of this study lie in the method and the benchmark rather than in the platform. Solution quality is weakest on the planted-clique **gen** and **san** instances and on the Brock graphs, where the Motzkin-Straus landscape hides the maximum clique among many competing local optima. The comparison is also restricted to the unregularized quadratic formulation and to two classical baselines (Armijo-PGD and Hexaly), and does not include specialized clique heuristics. Finally, each run currently yields a single clique extracted from the highest-weight solution. Future work should therefore focus on hybrid quantum-classical algorithms.

The true power of this paradigm may lie in intelligent hybrid algorithms that combine the complementary strengths of quantum and classical approaches. One compelling direction involves using the Dirac-3S entropy computing oracle to quickly identify promising regions of the search space through natural exploration mechanisms, followed by refinement using classical solvers like projected gradient descent that excel at local optimization. A similar hybrid philosophy has been demonstrated on neutral-atom hardware, where a quantum-informed reduction algorithm uses the quantum processor to resolve the hard core of the graph when classical kernelization stalls, achieving exact solutions on instances where standalone quantum approaches fail entirely [47]. Developing more sophisticated techniques for extracting and interpreting the solution distributions from quantum measurements represents a rich area for future research, potentially enabling access to multiple high-quality solutions from single quantum computations rather than extracting only the highest-weight solution.

6 Conclusion

We have presented a quantum-classical hybrid framework that leverages the Motzkin-Straus theorem to map the NP-hard maximum clique problem into a continuous quadratic program natively suited to QCI’s Dirac-3S entropy computer. The key insight is that the Motzkin-Straus formulation’s three requirements – a quadratic objective, simplex constraint, and non-negativity – are all satisfied at the hardware level by the photonic architecture, eliminating the encoding overhead that characterizes other quantum approaches. Through comprehensive benchmarking on 75 DIMACS instances against two independently implemented classical baselines – Armijo-PGD and the commercial solver Hexaly – we demonstrated that the Dirac-3S platform is competitive with both classical baselines overall and matches or leads them on the majority of instances, while well-tuned classical solvers retain an advantage on the hardest planted-clique benchmarks (Table 3).

This work should be viewed as a foundational experiment establishing that entropy computing can be meaningfully applied to canonical NP-hard problems. The main limitation which is the reduced accuracy on planted-clique instances, is

a matter of algorithm design rather than a fundamental barrier. Hybrid quantum-classical algorithms offer a clear pathway toward closing the gap with classical solvers and ultimately demonstrating practical quantum advantage for combinatorial optimization.

Acknowledgements. The authors would like to thank Lac Nguyen and other scientists at Quantum Computing Inc for their valuable discussions and engineering support on the Dirac-3S platform.

References

- [1] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
- [2] Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics* **17**, 533–540 (1965)
- [3] Marino, R., Buffoni, L., Zavalnij, B.: A Short Review on Novel Approaches for Maximum Clique Problem: from Classical algorithms to Graph Neural Networks and Quantum algorithms (2024)
- [4] Gibbons, L.E., Hearn, D.W., Pardalos, P.M., Ramana, M.V.: Continuous characterizations of the maximum clique problem. *Mathematics of Operations Research* **22**(3), 754–768 (1997) <https://doi.org/10.1287/moor.22.3.754>
- [5] Beretta, G., Torcinovich, A., Pelillo, M.: On generalized KKT points for the Motzkin-Straus program. *Journal of Global Optimization* **91**(3), 535–557 (2025)
- [6] Bomze, I.M., Peng, B., Qiu, Y., Yıldırım, E.A.: On tractable convex relaxations of standard quadratic optimization problems under sparsity constraints. *Journal of Optimization Theory and Applications* **204**(3) (2025)
- [7] Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: Maximum weight cliques. *Handbook of combinatorial optimization* **4**, 1–74 (1999)
- [8] Bomze, I.M.: Evolution towards perfection. *Games and economic behavior* **20**(2), 227–249 (1997)
- [9] Belachew, M.T., Gillis, N.: Solving the Maximum Clique Problem with Symmetric Rank-One Nonnegative Matrix Approximation (2015)
- [10] Alkhouri, I., Le Denmat, C., Li, Y., Yu, C., Liu, J., Wang, R., Velasquez, A.: Differentiable quadratic programming layers for maximum independent set. In: *International Conference on Learning Representations (ICLR)* (2025). arXiv:2406.19532
- [11] Sanokowski, S., Hochreiter, S., Lehner, S.: A diffusion model framework for unsupervised neural combinatorial optimization. In: *International Conference on*

- [12] Lucas, A.: Ising formulations of many np problems. *Frontiers in Physics* **2**, 5 (2014) <https://doi.org/10.3389/fphy.2014.00005>
- [13] Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm (2014)
- [14] Ebadi, S., Keesling, A., Cain, M., Wang, T.T., Levine, H., Bluvstein, D., Semeghini, G., Omran, A., Liu, J., Samajdar, R., Luo, X.-Z., Nash, B., Gao, X., Barak, B., Farhi, E., Sachdev, S., Gemelke, N., Zhou, L., Choi, S., Pichler, H., Wang, S., Greiner, M., Vuletic, V., Lukin, M.D.: Quantum optimization of maximum independent set using rydberg atom arrays. *Science* **376**(6598), 1209–1215 (2022)
- [15] Cazals, P., François, A., Henriët, L., Leclerc, L., Marin, M., Naghmouchi, Y., Silva Coelho, W., Sikora, F., Vitale, V., Watrigant, R., Garzillo, M.W., Dalyac, C.: Identifying hard native instances for the maximum independent set problem on neutral atoms quantum processors (2025)
- [16] Wybo, E., Rönkkö, J., Hirviniemi, O., Finžgar, J.R., Leib, M.: A scalable quantum-enhanced greedy algorithm for maximum independent set problems (2026)
- [17] Kumar, S., *et al.*: Digital-analog counterdiabatic quantum optimization with trapped ions. *Quantum Science and Technology* **10**, 015023 (2025)
- [18] Nguyen, L., Miri, M.-A., Rupert, R.J., Dyk, W., Wu, S., Vrahoretis, N., Huang, I., Begliarbekov, M., Chancellor, N., Chukwu, U., Mahamuni, P., Martinez-Delgado, C., Haycraft, D., Spear, C., Campanelli, M., Huffman, R., Sua, Y.M., Huang, Y.: Entropy Computing: A Paradigm for Optimization in an Open Quantum System (2024)
- [19] Verstraete, F., Wolf, M.M., Cirac, J.I.: Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics* **5**(9), 633–636 (2009)
- [20] Kraus, B., Büchler, H.P., Diehl, S., Kantian, A., Micheli, A., Zoller, P.: Preparation of entangled states by quantum markov processes. *Physical Review A* **78**, 042307 (2008)
- [21] Wiseman, H.M., Milburn, G.J.: Quantum Measurement and Control. Cambridge University Press, Cambridge (2010)
- [22] Hungerford, J.T., Rinaldi, F.: A General Regularized Continuous Formulation for the Maximum Clique Problem (2017)
- [23] Johnson, D.S., Trick, M.A. (eds.): Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics

- and Theoretical Computer Science, vol. 26. American Mathematical Society, Providence, RI (1996)
- [24] Armijo, L.: Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics* **16**, 1–3 (1966)
 - [25] Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* **10**(4), 1196–1211 (2000)
 - [26] Hexaly: Hexaly Optimizer. <https://www.hexaly.com>
 - [27] Fortunato, S.: Community detection in graphs. *Physics Reports* **486**(3-5), 75–174 (2010) [arXiv:0906.0612](https://arxiv.org/abs/0906.0612) [physics.soc-ph]
 - [28] Inagaki, T., Haribara, Y., Igarashi, K., Sonobe, T., Tamate, S., Honjo, T., Marandi, A., McMahon, P.L., Umeki, T., Enbutsu, K., Tadanaga, O., Takenouchi, H., Aihara, K., Kawarabayashi, K.-i., Inoue, K., Utsunomiya, S., Takesue, H.: A coherent ising machine for 2000-node optimization problems. *Science* **354**(6312), 603–606 (2016)
 - [29] McMahon, P.L., Marandi, A., Haribara, Y., Hamerly, R., Langrock, C., Tamate, S., Inagaki, T., Takesue, H., Utsunomiya, S., Aihara, K., Byer, R.L., Fejer, M.M., Mabuchi, H., Yamamoto, Y.: A fully-programmable 100-spin coherent ising machine with all-to-all connections. *Science* **354**(6312), 615–617 (2016)
 - [30] Yamamoto, Y., Leleu, T., Ganguli, S., Mabuchi, H.: Coherent Ising machines – Quantum optics and neural network perspectives (2020)
 - [31] Yamamura, A., Aihara, K., Yamamoto, Y.: Quantum model for coherent ising machines: Discrete-time measurement feedback formulation. *Physical Review A* **96**, 053834 (2017)
 - [32] Takesue, H., Inaba, K., Honjo, T., Yamada, Y., Ikuta, T., Yonezu, Y., Inagaki, T., Umeki, T., Kasahara, R.: Finding independent sets in large-scale graphs with a coherent Ising machine. *Science Advances* **11**(7), 7223 (2025)
 - [33] Jagota, A., Pelillo, M.: Feasible and infeasible maxima in a quadratic program for maximum clique. *Journal of Artificial Neural Networks* **2** (1995)
 - [34] Gibbons, L.E., Hearn, D.W., Pardalos, P.M.: A continuous based heuristic for the maximum clique problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **26**, 103–124 (1996)
 - [35] Grosso, A., Locatelli, M., Pullan, W.: Simple ingredients leading to very efficient heuristics for the maximum clique problem. *Journal of Heuristics* **14**(6), 587–612 (2008)

- [36] Wu, Q., Hao, J.-K.: A review on algorithms for maximum clique problems. *European Journal of Operational Research* **242**(3), 693–709 (2015)
- [37] McCreesh, C., Prosser, P.: Multi-threading a state-of-the-art maximum clique algorithm. *Algorithms* **6**(4), 618–635 (2013)
- [38] Karalias, N., Loukas, A.: Erdos Goes Neural: an Unsupervised Learning Framework for Combinatorial Optimization on Graphs (2021)
- [39] Acikalin, U.U., Ferber, A.M., Gomes, C.P.: Learning to explore and exploit with GNNs for unsupervised combinatorial optimization. In: *International Conference on Learning Representations (ICLR)* (2025)
- [40] Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l_1 -ball for learning in high dimensions. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279 (2008). ACM
- [41] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: composable transformations of Python+NumPy programs (2018)
- [42] Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation* **11**(8), 1933–1955 (1999)
- [43] Bomze, I.M., Budinich, M., Pelillo, M., Rossi, C.: Annealed replication: a new heuristic for the maximum clique problem. *Discrete Applied Mathematics* **121**, 27–49 (2002)
- [44] Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. *Physical Review E* **58**, 5355 (1998)
- [45] Oh, C., Fefferman, B., Jiang, L., Quesada, N.: Quantum-inspired classical algorithm for graph problems by Gaussian boson sampling. *PRX Quantum* **5**, 020341 (2024)
- [46] Gherardi, A., Leporati, A.: An Analysis of Quantum Annealing Algorithms for Solving the Maximum Clique Problem (2024)
- [47] Schuetz, M.J.A., et al.: qReduMIS: A Quantum-Informed Reduction Algorithm for the Maximum Independent Set Problem (2025)