

Cosm: Collective Switched Motion for Fast and Accurate Sparse Ising Optimization

Kenneth M. Zick, Nikhil Shukla, Alexander Marakov

Abstract—We introduce Collective Switched Motion (Cosm), a heuristic algorithm for solving sparse Ising-type optimization problems. Cosm combines locally interacting continuous circular variables with global coordination rules that facilitate collective dynamics. Pairwise interactions occur sequentially over a set of conflict-free edge partitions, resulting in an interaction network that switches periodically. Unlike conventional gradient-based approaches, Cosm enables structured, non-gradient dynamics that promote exploration beyond local minima. A correlated perturbation mechanism helps enable collective variable rotations. On the three largest Gset instances, which have 10,000–20,000 variables and represent 2D spin glasses, Cosm attains improved solutions that are verified as optimal using an exact solver. On two large bounded-degree Gset instances, a CPU-based implementation of Cosm reduces the state-of-the-art times-to-target from hundreds of hours to 36–303 s, reductions of 2–4 orders of magnitude. Additional tests on planted-solution benchmark instances show a lower scaling exponent than previous dynamical systems heuristics. These results highlight the effectiveness of Cosm in harnessing collective computation for improved sparse combinatorial optimization.

Index Terms—Binary optimization, collective computation, combinatorial optimization, dynamical system, heuristic algorithm, Ising, Max-Cut, QUBO, switched network, time-varying network.

I. INTRODUCTION

HIGHLY sparse binary optimization problems have applications such as task scheduling, low-density parity-check (LDPC) decoding, spin glass simulations, and power systems optimization [1] [2] [3]. Such problems often entail pairwise interactions and formulations that include Ising models, Max-Cut, and QUBO (quadratic unconstrained binary optimization). These sparse “Ising-type” problems often have rugged landscapes and frustration that challenge exact [4] [5] and approximate solvers such as semidefinite programming [6]. Heuristics such as simulated annealing, parallel tempering [7], and Breakout Local Search [8] are frequently employed. More recently, physics-inspired approaches have gained attention, particularly the Simulated Bifurcation Machine (SBM) introduced by Goto et al. [9], which has demonstrated state-of-the-art performance on dense problem instances [10] [11]. Although SBM and several other physics-inspired approaches, such as the Coherent Ising Machine (CIM), show promising characteristics, they find sparse optimization very challenging, as seen in a recent comparative

study by Hou et al. using planted-solution nearest-neighbor benchmark problems [12].

A complementary direction that has gained traction in recent years explores the use of quantum mechanical resources—such as superposition, tunneling, and entanglement—to accelerate combinatorial optimization. Two prominent approaches are quantum annealing [13] [14] [15] [16] and the Quantum Approximate Optimization Algorithm (QAOA) [17] [18]. Despite their conceptual appeal, it should be noted that there is currently no clear evidence that quantum algorithms for combinatorial optimization provide exponential speedups for real-world NP-hard problems, nor compelling evidence that they yield consistent practical advantages in heuristic settings. Moreover, many sparse optimization problems involve graphs with arbitrary connectivity and tens of thousands of variables, posing a significant engineering challenge for quantum processors.

Further motivation for new approaches comes from the field of spin glasses, which are disordered and frustrated magnetic systems. Parisi’s discoveries around 1980 of patterns in disordered materials [19] led to a better understanding of rugged energy landscapes and to the influential parallel tempering algorithm. Many hard combinatorial optimization problems can be formulated as Ising models, often with disordered and frustrated couplings characteristic of spin glasses. In particular, 2D and 3D spin glasses have bounded graph degree and are a canonical benchmark problem for solvers [20]. However, upon surveying the field in 2025, Parisi and authors Dahlberg et al. report that, beyond parallel tempering, algorithmic progress in this domain has been largely stagnant over the past two decades and that new algorithms are needed for both physics and engineering applications [21].

Taken together, these observations highlight a critical gap: while both classical and emerging quantum approaches have made progress, there remains a lack of fundamentally new algorithms that can effectively address large-scale, sparse Ising-type optimization problems. In this work, we present a novel heuristic approach for such problems, called Collective Switched Motion (Cosm). The main contributions of this paper are summarized as follows:

- We propose a switched dynamical system approach called sequential conflict-free search using edge-colored updates enabling efficient exploration of high-dimensional circular (phase) variable spaces.
- We introduce a novel correlated perturbation mechanism termed dual window twist that enhances cluster mobility along the unit circle and significantly improves solution quality.

Kenneth M. Zick is with the University of Southern California, Information Sciences Institute, Arlington, Virginia, USA.

Nikhil Shukla is with the University of Virginia, Charlottesville, Virginia, USA.

Alexander Marakov is with Northrop Grumman Systems Corporation, Linthicum, Maryland, USA.

- We present experimental results demonstrating substantial gains over existing approaches on three types of sparse, bounded-degree Ising benchmark problems.

II. COLLECTIVE SWITCHED MOTION – COSM

In this section, we cover the basics of Cosm, present the full pseudocode, and describe in detail two defining features.

A. Baseline Algorithmic Components

The Ising Hamiltonian is given by

$$H(\mathbf{s}) = - \sum_{(i,j) \in E_G} J_{ij} s_i s_j - \sum_i h_i s_i \quad (1)$$

with spin variables $s_i \in \{-1, +1\}$, edges (i,j) in the set E_G , interaction terms J_{ij} , and local fields h_i . In defining Cosm, we first relax the binary constraint on the variables and instead define variables x_i that live on a unit circle (i.e., a 1-sphere S^1), where $x_i \in [0, 360)$ degrees with $0 \sim 360$ (wrapped interval). Updates use modular arithmetic on $[0, 360)$. Circular variables have been employed in approaches such as oscillator Ising machines [22] [23] [24], XY machines [25], and semidefinite programming. Next, we replace the quadratic $s_i s_j$ term with a relaxed coupling term $g(x_i, x_j)$ that is a piecewise linear function of the minimum angular difference between the two variables. This sets up an inverted V-shaped potential [24] [26] for each coupling term, as illustrated in Fig. 1. The coupling energy $-J_{ij}g(x_i, x_j)$ is thus minimized when a ferromagnetically coupled (anti-ferromagnetically coupled) pair is at the same (diametrically opposite) position along the unit circle. The function $g(\cdot)$ is expressed as

$$g(x_i, x_j) = \frac{1 - |((x_i - x_j + 180) \bmod 360) - 180|}{90} \quad (2)$$

where mod returns a nonnegative remainder and $g(\cdot)$ takes values in the continuous range $[-1, 1]$. We then define how two coupled variables move as a result of a pairwise interaction. We use the slope of the coupling potential to control the direction and magnitude of the variable moves. Given the two constant slopes (Fig. 1), there are just two possible outcomes,

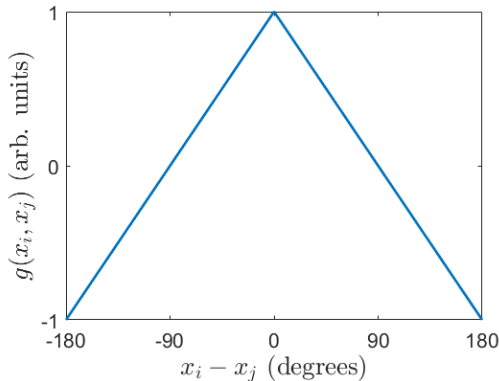


Fig. 1. Relaxed Pairwise Coupling Energy Function Used in Cosm. The quantity $x_i - x_j$ represents the minimum angular difference between two coupled circular variables.

analogous to bang-bang-like control policies which have two extremal control actions. While $g(\cdot)$ is not differentiable at 0 or ± 180 , we define a simple biased $\text{sgn}()$ function

$$\text{sgn}_+(x_i, x_j) = \begin{cases} 1 & \text{for } x_i - x_j \geq 0 \\ -1 & \text{for } x_i - x_j < 0 \end{cases} \quad (3)$$

that returns a magnitude of 1 in all cases, even when the input is 0. Sgn-based coupling interactions have been employed in other recent models [24] [26] [27]. In a Cosm pairwise interaction, variable x_i rotates by an amount proportional to $\alpha_t J_{ij} \text{sgn}_+(x_i, x_j)$ where α_t is a time-dependent step size. Due to symmetry, the same calculation applies to each variable in the pair but with opposite signs. As a result, one variable in a pair always rotates clockwise by $\alpha_t J_{ij}$ and the other counterclockwise by the same amount.

A gradient descent approach would consist of a relaxed objective function

$$f(\mathbf{x}) = - \sum_{(i,j) \in E_G} J_{ij} g(x_i, x_j) \quad (4)$$

(local fields omitted for clarity) and the update rule

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \nabla f(\mathbf{x}_t) \quad (5)$$

where ∇f is the generalized gradient using the previously defined biased sgn_+ function $\text{sgn}_+(x_i, x_j)$. We find that with circular variables and the model described by Eqs. 2 and 3,

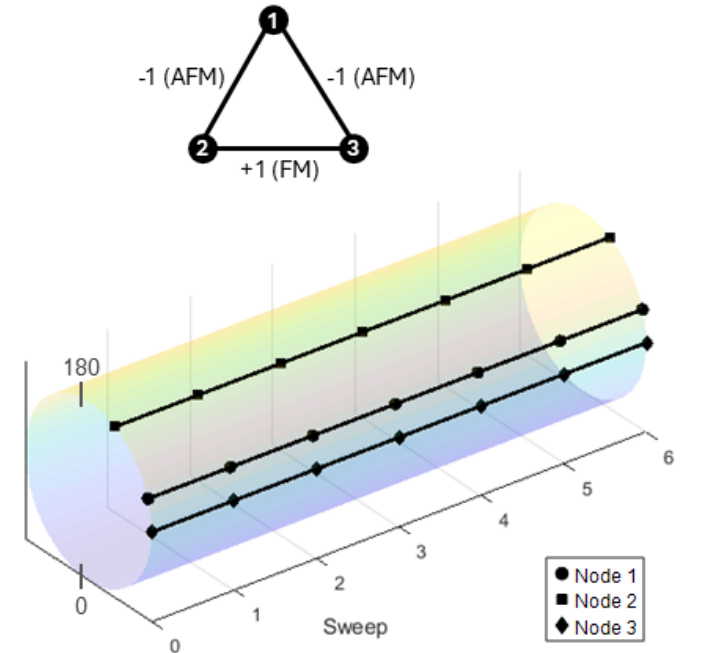


Fig. 2. Example of a Circular Conflict Trap. (top) Seemingly trivial 3-variable Ising problem with no frustration and two ground states (up-down-down and down-up-up, Ising energy -3). (bottom) Variable trajectory using gradient descent. The variable interval wraps around from 0 to 360 degrees in a circle as with phase variables. From the particular random initial state shown, forces cancel out and the gradient is 0, leading to frozen dynamics and a suboptimal solution (Ising energy -1) no matter where the final circular space is divided into two equal halves to binarize the system. While this minimal example shows a single trapped variable, one can envision much larger sets of variables.

standard gradient descent does not provide competitive accuracy on the sparse problems we test. Absent a formal proof, we provide an intuition for one of the difficulties. Unlike the non-periodic variables used in many methods such as SBM, circular variables have two alternative paths (clockwise and counterclockwise) to move around the circle. With gradient descent, a variable moves by an amount that is the sum of all of its interactions. If the contributions are in opposite directions, the sum can approach zero and the variable is stationary, even if a lower energy configuration were possible by rotating 180 degrees. The scenario is analogous to a coordination failure in multi-agent systems, when agents agree on a goal but do not achieve a consensus on which of two directions to pursue. This is a particular type of sub-optimal, meta-stable state we refer to as a *circular conflict trap*. A minimal example in Fig. 2 illustrates how, given Eqs. 2 and 3, even a very small system can get trapped. While this example is illustrative, similar cancellation effects can arise in larger sparse graphs due to competing local interactions. Cosm institutes two key mechanisms designed to improve the search dynamics in sparse Ising-type landscapes involving circular variables.

B. Cosm Pseudocode

The Cosm algorithm is described in pseudocode in Alg. 1. It consists of an edge coloring step, a main loop of sweeps, and a binarization step. With 2D and 3D lattice graph instances, an edge coloring can be trivially assigned using a regular pattern. With non-lattice instances, a variety of polynomial-time heuristics are possible for assigning a unique edge coloring to each instance. In this work, we employ a simple greedy heuristic that loops over the edges and assigns the lowest color number available. Within each sweep are two key features (sequential conflict-free search and dual window twist perturbations) that are covered in Sections II-C and II-D. Cosm employs a time-dependent continuous step size α_t so the system can be annealed [13]; an example annealing schedule is a linear ramp from α_0 down to 0. After annealing, the unit circle is bisected into two halves, and variables in one half are assigned -1 and in the other $+1$. Problems that have local fields have defined semicircular sections and thus this binarization step is trivial. For problems that do not have local fields, a procedure is needed to find an effective bisector. Here, we evaluate a set of equally-spaced bisectors and select the best solution (an alternative algorithm for binarization is provided in Ref. [28]).

C. Sequential Conflict-free Search (SCS)

To improve the efficacy of the search process, Cosm assigns the input graph G a proper edge coloring with C colors (Fig. 3a), such that the edge set E_G is partitioned as

$$E_G = E_1 \cup E_2 \cup \dots \cup E_C, \quad (6)$$

where each subset E_c corresponds to a matching (i.e., no two edges share a common vertex) associated with color c . A minimum coloring of the graph is not required.

Algorithm 1 Cosm

```

1: Assign a proper edge coloring of the input graph
2: Set  $\alpha_0$ 
3: Set  $x_i$  random
4: for  $sweep = 1$  to  $numSweeps$  do
5:   for each colored set of independent edges do
6:     for each edge in colored set do
7:       Calculate  $sgn_+(x_i, x_j)$ 
8:       Update  $x_i, x_j$  by  $\alpha_t J_{ij}$  in assigned directions
9:     end for
10:   end for
11:   if sweep is a multiple of  $DWTperiod$  then
12:     Select random reference angle  $\theta_{ref}$ 
13:     Define dual opposing windows centered at  $\theta_{ref}$ 
14:     for all  $x_i$  do
15:       Rotate  $x_i$  by  $DWTratio \times \alpha_t$ 
16:     end for
17:   end if
18:   Update  $\alpha_t$  according to step size schedule
19: end for
20: for  $p = 1$  to  $numBisectors$  do
21:   Select line bisecting the unit circle
22:   Convert  $x_i$  to two-valued  $s_i$  using the two semi-circles
23:   Evaluate the original binary objective function
24: end for
25: Return best solution

```

The objective function is correspondingly decomposed as

$$f(\mathbf{x}) = \sum_{c=1}^C f_c(\mathbf{x}), \quad (7)$$

where

$$f_c(\mathbf{x}) = - \sum_{(i,j) \in E_c} J_{ij} g(x_i, x_j). \quad (8)$$

Rather than updating the system using the full gradient $\nabla f(\mathbf{x})$, the Cosm update rule cyclically applies gradients of the individual components f_c . Specifically, at iteration t , the update is given by

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \nabla f_{c(t)}(\mathbf{x}_t), \quad (9)$$

where the active partition index is defined as

$$c(t) = ((t - 1) \bmod C) + 1 \quad (10)$$

and $\nabla f_{c(t)}$ is the generalized gradient using the previously defined biased sgn function $sgn_+(x_i, x_j)$. Thus, the algorithm performs a deterministic rotation over the C edge partitions. Effectively, the system evolves on a network for which the edge set changes in time. This is a particular type of switched (time-varying) network we refer to as a *periodically switched network*. At each point in time a variable is connected to at most one other variable and thus each edge set is free of potentially conflicting interactions. The concept is illustrated in Fig. 3b. We refer to the dynamical systems strategy utilizing such a network as *sequential conflict-free search* (SCS).

In SCS, the processing of all edges of a given color is referred to as a sub-sweep, and processing all edges in the problem graph is a *sweep* consisting of C sub-sweeps. SCS leads to *intra-sweep variable fluctuations* in which variable movements (phase rotations) accumulate from sub-sweep to sub-sweep. These lead to unique dynamics that can empirically

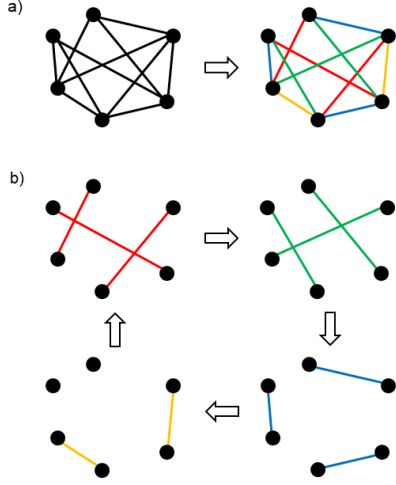


Fig. 3. Edge-Coloring for Sequential Conflict-Free Search. a) Cosm first assigns a proper edge coloring of the input graph, either by loading a standard coloring or generating one using a polynomial-time heuristic. b) The edge colored graph is separated by color into C matchings (graphs for which degree ≤ 1), ensuring there are no conflicting interactions. The system evolves according to the pairwise interactions of one color at a time and repeats in a sequence, representing a dynamical system on a conflict-free periodically switched network.

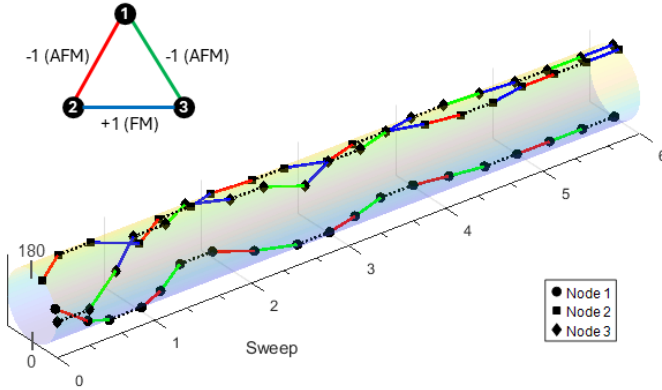


Fig. 4. Example of Cosm's Sequential Conflict-Free Search (SCS) Dynamics. (upper left) Three-variable problem instance after edge coloring. (bottom) System evolution in a periodic sequence of red, green, and blue interactions. Line segments are illustrated with the color of the associated interaction; when a variable is not involved in an interaction, it maintains its state (black dotted lines). Unlike the frozen dynamics shown in Fig. 2, the system escapes from the circular conflict trap and very quickly (1 sweep) reaches a configuration corresponding to the optimal solution (Ising energy -3).

improve performance. A minimal example of SCS dynamics and fluctuations is shown in Fig. 4. Unlike the baseline gradient descent example shown in Fig. 2, the system avoids a circular conflict trap and quickly reaches an optimal configuration.

D. Dual Window Twist (DWT) Perturbations

Complementing the deterministic SCS rules, Cosm institutes a novel type of perturbation of the spin variables. At occasional times during system evolution, Cosm selects two diametrically opposed equal-size windows centered around a random reference angle θ_{ref} along the unit circle. All variables found to be within either of the two windows rotate their

positions together by the same magnitude and in the same direction. We refer to this as a *dual window twist* (DWT) perturbation. An illustration is given in Fig. 5 and pseudocode in Alg. 1. DWT serves as a cluster-preserving perturbation. By rotating dual opposing windows rather than a single window, the alignment and anti-alignment relationships among a set of coupled variables within the windows are maintained. DWT can facilitate the separation of variable clusters in phase space and allow entire clusters to rotate if energetically favorable. DWT is a type of correlated perturbation that incorporates a stochastic aspect due to the random reference angle along the unit circle. The width of the windows is a tunable parameter; we find empirically that a width of 90 degrees (leading to exactly half of the phase space being included in the twist) is effective on the problems tested. The magnitude of rotation changes over time according to a schedule; for instance, the magnitude is typically set to a constant factor (e.g., 1.0) of the current step size α .

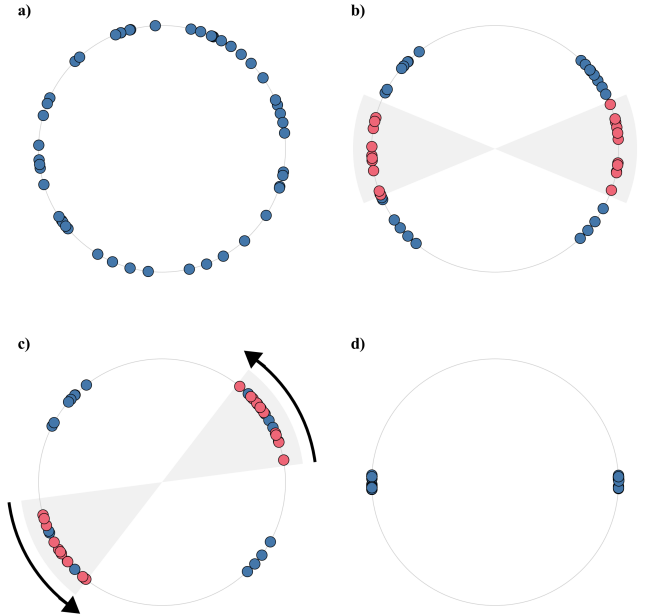


Fig. 5. Visualization of Cosm Dynamics with a Dual Window Twist (DWT) Perturbation. Variables are shown as filled-in circles residing on a unit circle. a) The system is initialized to a uniform random state. b) In addition to variable movements due to Sequential Conflict-free Search, at regular intervals a pair of dual opposing windows (shaded) is defined at a random angle. Variables within the windows are selected for perturbation. c) Variables within the dual windows are rotated by an amount proportional to the step size schedule. d) After an annealing cycle (solver run) with multiple sweeps and DWT perturbations, the system (shown here with two tight variable clusters for clarity) is partitioned to generate a binary solution.

In concert with Cosm's other features, the DWT mechanism is meant to promote energy-improving many-variable moves. Achieving useful cluster moves has long been an aim in binary optimization. Among spin glass simulation methods, there are replica cluster moves such as Houdayer [29] and isoenergetic cluster moves [7]. These require simulating multiple replicas of a system, comparing binary states, and flipping the binary states of selected clusters. Such moves randomize the config-

urations and can in some cases greatly accelerate simulations. Cosm, assisted by DWT, is an alternative approach in which clusters of variables in a continuous circular space emerge, separate, and move in a single dynamical system.

E. Cosm Implementation

Cosm possesses several properties that make it amenable to efficient implementation. First, no double-precision arithmetic is needed, enabling reduced computational and hardware overhead. Second, it primarily employs deterministic rules and mechanisms and consumes a low rate of pseudorandom bits. Aside from random initialization of the variable vector, only a single global random value is applied once every few sweeps during system evolution, to define the location of the DWT perturbation. This contrasts with methods that require random bits for each variable and update. Third, by construction, Cosm operates on sets of edge-disjoint interactions, allowing all edges within a given color class to be processed in parallel. This structure supports parallel implementations on platforms such as GPUs, wafer-scale engines [30], FPGAs, and custom ASICs.

In this work, we implement Cosm in C and evaluate its performance on a CPU, as described in Section III.

III. EXPERIMENTAL RESULTS

We test Cosm on three types of sparse benchmark problems: large Gset 2D lattice graph spin glasses, large Gset non-lattice graph problems with bounded degree, and a set of synthetic 2D lattice graph instances with tunable hardness. We implement Cosm in C and run it on a server CPU (Intel Xeon Gold 6544Y). We refer to this solver setup as Cosm-CPU. Implementation details are given in Appendices B and C.

A. Performance on the Largest Gset 2D Spin Glasses

For binary quadratic optimization, the Gset suite [31] constitutes a canonical benchmark collection of Max-Cut and Ising-type problems. Originally developed between 1999 [32] and 2000 [6], it has since been widely adopted to assess the performance of both conventional solvers and physics-inspired Ising machine models.

First, we select the three largest instances in the Gset suite. These instances—G72, G77, and G81—are equivalent to 2D spin glasses with degree-4 nearest neighbor connectivity, bimodal ($\pm J$) couplings, and a toroidal graph structure (periodic boundary conditions in both dimensions). Searching for the ground state of an Ising spin glass defined on a lattice graph (Edwards-Anderson spin glass) is a canonical problem in condensed matter physics [21] as well as an established benchmark for binary quadratic optimization [20]. While many nonplanar Ising problems are NP-hard, the toroidal square grid problem with $\pm J$ couplings and no local fields is an exceptional case [33] and admits a fast heuristic method [34]. However, these three instances with 10,000–20,000 variables have not been easily solved. Before Cosm, the best reported cuts were 7006, 9938, and 14056.

For each instance, we find that Cosm finds a better solution than reported by any other method. The new best known cuts

are 7008, 9940, and 14060. Cosm solution bitstrings for all three problems are posted in Refs. [35] [36] to allow the solutions to be independently validated.

Here we also supply the three instances to an exact solver, Gurobi 12.0.3. The Gurobi runs certify that the 7008, 9940, and 14060 solutions are in fact the *optimal* solutions for these 25-year old problems. Ref. [37] similarly found that a recent version of Gurobi could prove optimality for these instances (though the solutions were not provided). Cosm finds that the associated Ising ground state energies are -14022 , -19672 , and -28086 . Table I summarizes the problem instance properties and Cosm solutions.

TABLE I
COSM FINDS THE CERTIFIED OPTIMAL SOLUTIONS TO THE LARGEST GSET INSTANCES. N : NUMBER OF VARIABLES, E_G : SET OF EDGES.

Gset Instance	N	$ E_G $	Max degree	Best previous solution	Cosm solution	Cosm solution quality
G72	10000	20000	4	7006 [38]	7008	Certified optimal
G77	14000	28000	4	9938 [38]	9940	Certified optimal
G81	20000	40000	4	14056 [39]	14060	Certified optimal

One measure of heuristic performance is sweeps-to-solution (STS), which is the number of sweeps required to ensure reaching the optimal solution with 99% confidence (Appendix A). The Cosm STS values are shown in Table II. There are not yet any comparable STS results in the literature for these instances. As one point of comparison, a recent study tested eight different quantum-inspired and physics-inspired heuristics on instance G81. After 1000 trials of 10^5 steps/trial, the highest cuts attained were roughly 13800 to 14000 [15]. Direct comparison is difficult due to potentially different notions of steps and sweeps. However, we note that in 1000 trials of 10^5 sweeps, Cosm attains a G81 cut of 14056. Additional data on Cosm's highest and average G81 solutions for different run lengths is provided in Ref. [36].

TABLE II
COSM SWEEPS-TO-SOLUTION

Gset Instance	Sweeps per Trial ($\times 10^6$)	Cosm Solution	P_s	R_{99}	Sweeps-to-Solution ($\times 10^6$)
G72	1.0	7008	$405/1024 = 0.3955$	9.15	9.2
G77	1.0	9940	$237/1024 = 0.2314$	17.49	17.5
G81	2.0	14060	$20/1024 = 0.0195$	233.47	467.0

The wall clock times-to-solution using the Cosm-CPU setup depend on the assumed level of parallelism. Assuming use of all 32 CPU cores and two threads per core, the average execution times per trial for G72, G77, and G81 are 1.41, 1.98, and 5.68 s, respectively, giving wall clock time-to-solution (TTS) values of 12.9, 34.7, and 1326.1 s.

To gain empirical insight into dynamical system state changes, we previously monitored several Cosm G72 trials in detail and extracted the final variable changes [40] that led to Cosm exceeding a cut of 7006 and reaching the new best (optimal) solution of 7008. In some trials, variables in multiple regions of the toroidal graph changed state. In four trials, there was a single contiguous cluster that changed state to enable the

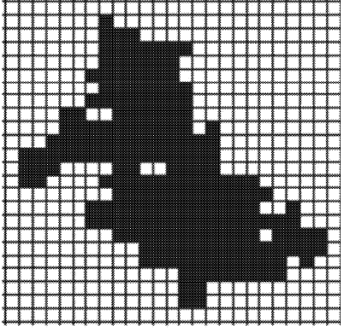


Fig. 6. Spatial Layout of a Large Cluster of Variables (shaded) Representing the Final Move Enabling the New Best and Optimal Solution to Gset Instance G72 (7008). The cluster is visualized on a 25x25 subset of the 100x100 toroidal problem graph. Each square cell represents a variable. The cluster size is 204 variables, representing over 2% of all problem variables. The cluster has 128 pairwise bonds at its interface (connections to non-shaded cells). Variables in the cluster changed their continuous states together to reach the opposite binary state (after binarization), increasing the number of satisfied bonds at the interface from 63 to 65 and improving the cut from 7006 to 7008.

optimal solution; those cluster sizes were 29, 35, 122, and 204 variables. These observations serve as a hint regarding Cosm’s ability to traverse the energy landscape with non-local moves. Note that a cluster of 204 variables represents more than 2% of the entire system. The 204-variable cluster layout on a portion of the problem lattice is illustrated in Fig. 6. The cluster has 128 pairwise bonds at its interface (i.e., with variables outside of the cluster). By changing their continuous states together to reach the opposite binary state (after binarization), the number of satisfied bonds at the interface increased from 63 to 65, improving the cut from 7006 to 7008.

B. Performance on the Largest Gset Bounded-Degree Non-Lattice Instances (G61, G70)

We test Cosm on the two largest Gset bounded-degree non-lattice instances, G61 and G70. Bounded degree refers to problems for which the coupling degree does not grow with the size of the problem, for instance due to physical constraints (e.g., in road networks or VLSI digital circuits). The two problem graphs have irregular structure and maximum node degrees of 14 and 9, respectively. Key properties of these instances are summarized in Table III.

TABLE III
PROBLEM CHARACTERISTICS OF THE LARGEST BOUNDED-DEGREE NON-LATTICE GSET INSTANCES. N : NUMBER OF VARIABLES, E_G : SET OF EDGES.

Gset Instance	N	$ E_G $	Max Degree	Avg Degree	Weights
G61	7000	17148	14	4.9	$\{-1, +1\}$
G70	10000	9999	9	2.0	Unweighted

This problem type is NP-hard and empirically difficult for heuristic optimizers to reach even near-best known solution quality. Recent efforts have reported incremental improvements on these benchmarks under highly specialized conditions. An evolutionary algorithm called the Large Population Island (LPI) framework by Goudet et al. reached a

new best known solution of 5799 for G61 during extensive calibration runs using an NVIDIA Tesla V100 GPU [41]. Subsequent evaluation runs reached a high of 5798. Similarly, a Monte Carlo gradient-based heuristic by Chen et al. reached a new best known solution of 9595 for G70 (first reported in 2023 [42]) after prolonged execution using an NVIDIA Tesla A100 GPU. During standard testing, it reached a maximum cut of 9572 [43].

We find that Cosm reaches the best known solutions for each of the instances—5799 for G61 and 9595 for G70—with regularity. A comparison of Cosm solution quality with other notable results is given in Table IV.

TABLE IV
HIGHEST CUTS ATTAINED

Solver Approach	G61	G70
Cosm (this work)	5799	9595
Large population island - calibration [41]	5799	9595
Large population island - test [41]	5798	9594
Fixstars Amplify Annealing Engine [44]	5796	9582
Simulated bifurcation machine [9]	5796	9578
Simulated annealing [14]	5788	9546
Breakout Local Search [8]	5789	9541
Monte Carlo policy gradient - calibration [43]	n/a	9595
Monte Carlo policy gradient - test [43]	5748	9572
D-Wave Hybrid [14]	5750	9516
CirCut [28]	5690	9529
Physics-inspired graph neural network [1]	n/a	9421

To the best of our knowledge, no prior work has reported success probabilities for reaching the current best known solutions on these instances. In contrast, the performance of Cosm is sufficiently consistent to permit such a characterization. Over 256 independent trials of 1.28×10^8 sweeps, Cosm reaches the best known solution in 25.4% of G61 trials and 75.8% of G70 trials (Table V).

TABLE V
COSM SUCCESS PROBABILITY, 128M SWEEPS

Gset Instance	Successes/Trials	P_s
G61	65/256	0.254
G70	194/256	0.758

We measure the speed of the Cosm-CPU setup using the wall clock time-to-target (TTT) metric, defined as the expected time required to reach a specified solution quality with 99% probability (Appendix A). Measurements of average execution time per trial cover the annealing time including the binarization step at the end of each trial. We do not include the time for edge coloring which is performed once per instance. Edge coloring is performed off-line using the greedy heuristic from Section II-B; using MATLAB, the edge coloring takes 3.2 s (G61) and 1.7 s (G70). Although a minimum coloring is not strictly necessary, the edge coloring heuristic colors these graphs using the minimum number (14 and 9 colors). The numbers of edges assigned each color are provided in Appendix B. For each benchmark instance, we conduct 64,000 independent trials. Empirically, we observe that the solver

achieves high efficiency when processing batches of $r \approx 256$ –512 trials concurrently. Accordingly, we employ short trial lengths (0.5M sweeps per trial for G61 and 0.2M for G70) and select $r \approx 512$ and $r \approx 256$, respectively, to minimize TTT.

Under this configuration, Cosm-CPU achieves a wall clock TTT of 303 s on G61 and 36 s on G70 for the best known solutions. To the best of our knowledge, TTT values for these solution qualities have not been previously reported.

To provide context for these results, we consider the best previous reported results. The only solver of which we are aware to have reached the best known solution on G61 (5799) is the Goudet et al. LPI algorithm [41], which reached this target during hundreds of hours of calibration runs on a V100 GPU. The target was not reached during 20-hour test runs. While the hardware used is not directly comparable, the Cosm-CPU TTT of 303 s on G61 is in a very different performance regime. To reach a target one unit below the best known values, Goudet et al. report times of 74,373 s for G61 (cut 5798), and 28,820 s for G70 (cut 9594). Thus, the Cosm-CPU results are approximately 2–3 orders of magnitude faster (303 and 36 s) and simultaneously more accurate. The Simulated Bifurcation Machine (SBM) attained a wall clock TTT of 31,599 s to reach a cut value of 9578 on G70 using an NVIDIA Tesla V100 GPU [9]; for this solution quality, Cosm-CPU requires ~ 2 s. A summary of Cosm TTT results is provided in Table VI.

TABLE VI
COSM-CPU WALL CLOCK TIME-TO-TARGET (TTT) RESULTS. R_{99} : REPETITIONS FOR 99% SUCCESS PROBABILITY.

Gset Instance	Highest Cut Achieved	P_s (64,000 trials)	R_{99}	Cosm Sweeps-to-Target ($\times 10^6$)	Cosm-CPU Wall Clock TTT (s)
G61	5799	0.008328	550.7	275	303.3
G70	9595	0.01872	243.7	48.7	35.9

To assess the extent to which two of Cosm’s key features contribute to performance, we conduct an ablation study comparing Cosm against three alternative algorithms for which SCS is disabled, DWT is disabled, and both SCS and DWT are disabled, respectively. In all four cases, 1000 trials of Gset problem G70 are performed, with 1.5M sweeps/trial. The results are shown in Table VII. With this run length of 1.5M sweeps, Cosm achieves the best known solution in 16.3% of trials and an average cut of 9593.39. In the three other cases, the highest achieved cuts are well below the best known, and the average cuts are dramatically lower than Cosm. This result provides evidence that both SCS and DWT are essential to Cosm’s performance.

C. 2D Tile-Planted Instances

To test Cosm on an additional sparse problem type outside of the Gset suite, we consider a set of 2D lattice instances that were crafted in Ref. [12] using the Chook benchmark instance generator [46]. The weights come from the set $\{\pm 1, \pm 2\}$; the use of two weight magnitudes allows tuning of the problem hardness [47]. The problem graphs have degree-4

TABLE VII
ABLATION EXPERIMENT RESULTS. 1000 TRIALS OF INSTANCE G70. 1.5M SWEEPS/TRIAL FOR EACH ALGORITHM VARIATION. COSM FINDS THE BEST KNOWN SOLUTION ON 16% OF TRIALS. PERFORMANCE DROPS PRECIPITOUSLY WHEN DISABLING SEQUENTIAL CONFLICT-FREE SEARCH (SCS) OR DUAL WINDOW TWIST (DWT) PERTURBATIONS.

Algorithm	Highest Cut	Avg Cut	P_s
Cosm	9595	9593.39	0.163
without SCS	9551	9528.00	0
without DWT	9544	9501.40	0
no SCS, no DWT	9481	9432.81	0

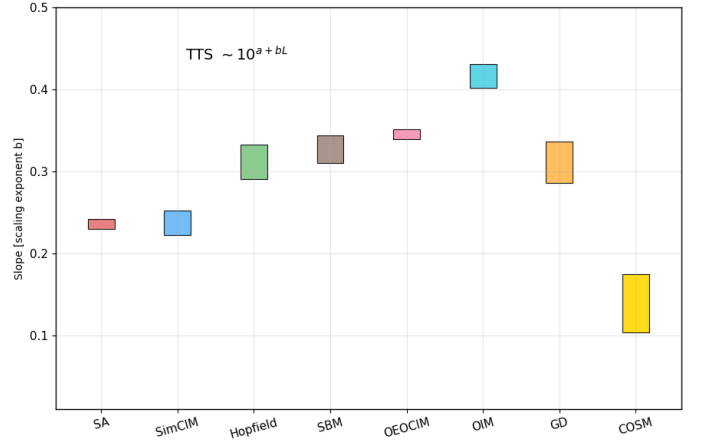


Fig. 7. Comparison of TTS Scaling Exponents of Eight Solvers on Tile-Planted Benchmark Instances. Error bars for the seven other solvers were extracted from Ref. [45] and are approximate. Cosm’s 95% confidence interval is shown on the right.

nearest neighbor connectivity and periodic boundaries in both dimensions. By construction, using the tile-based approach, each instance has a known ground state making the set suitable for benchmarking. The collection consists of 100 instances for each problem size ranging from 16 to 1024 variables, where the number of variables $N = L^2$ and $L \in \{4, 6, 8, \dots, 32\}$.

These instances were used in an extensive comparative study of stochastic driven non-linear dynamical systems by Hou et al. [12]. Seven prominent solver algorithms were tested including simulated annealing, simulated coherent Ising machine (SimCIM), OIM, and a version of SBM. Since many of the solvers were unable to attain ground states for the larger problem sizes, scaling characterization was limited to the five largest sizes shown in Fig. 1 of Ref. [12], corresponding to sizes of $L \in \{12, 14, 16, 18, 20\}$. The authors found that the median TTS data for simulated annealing and SimCIM fit to the lowest (best) scaling exponents, in the range of 0.22 to 0.25.

We obtained the Chook-generated instances from the authors and tested Cosm (details in Appendix C). We first find that, unlike many of the solvers in [12], Cosm attains the ground state of all instances and all sizes in the set within the tested sweep budgets, including the largest instances with $N = 1024$. Specifically, at $N = 1024$ with 10,000 trials and 1000 sweeps/trial, the Cosm median P_s across the 100 instances is $P_s = 2.09\%$ and the median STS is 2.18×10^5 sweeps.

Though absolute runtimes are not directly comparable due to different code bases and hardware, scaling exponent comparisons are more robust to implementation differences. Thus, we estimate the scaling of median TTS vs. L for Cosm over the same problem sizes as Ref. [12], namely $L \in \{12, 14, 16, 18, 20\}$ (144 to 400 variables). As with Ref. [12], we use a single CPU core when measuring execution time. From measured P_s values and average execution times over 10,000 trials for each of median instances, we calculate median TTS at each problem size and fit the data using the same method as Ref. [12] (linear fit to $a + bL$ in the logarithmic scale). We obtain an estimated value for the slope (scaling exponent b) of 0.139 (95% confidence interval: 0.104–0.175). This value is lower than that of the seven previously characterized solvers, including simulated annealing and Sim-CIM (0.22–0.25). The data are plotted in Fig. 7. Additionally, we characterize Cosm scaling over a wider range of sizes ($L \in \{12, 14, \dots, 32\}$), and obtain a scaling exponent of 0.128.

IV. RELATED WORK

Approaches such as SBM and OIM attempt to use the collective behavior of many pairwise interacting dynamical systems. SBM is a deterministic approach in which virtual particles move according to quantum-inspired equations of motion [9]. The discrete variant in particular, called dSB, uses sgn-based coupling as Cosm does, though unlike Cosm the variable range is non-periodic. A recent OIM approach also pursues sgn-based coupling [27]. Cosm is distinct from SBM and OIM due its SCS strategy and use of a stochastically-defined correlated perturbation (DWT).

Stochastic gradient descent—used widely for training deep neural networks—has a conceptual connection to Cosm’s SCS; in both cases search is performed not using the gradient of the loss (objective) function but rather a sequence of gradients of small portions of the function. However, Cosm updates are predominantly deterministic and cyclic, rather than stochastic, which preserves structure in the dynamics while still breaking the symmetry of simultaneous updates. SCS is related to block-coordinate descent methods in that updates are performed on subsets of the objective function. However, Cosm is a flexible approach; for lattice graphs, all variables are updated in each sub-sweep, making it a matching-based full-space method. For non-lattice graphs, different subsets of variables are updated in each subsweep. Some existing work in binary optimization relies on *vertex* coloring for variable updates (block coordinate descent), such as a Monte Carlo probabilistic bit approach [48]; we have found scarce work in binary optimization that exploits edge coloring.

Monte Carlo methods are primarily stochastic while Cosm can be thought of as primarily structured deterministic. A reinforcement learning approach has been proposed for ground state search in lattice spin glasses [49]; its applicability to non-lattice problems is not clear.

Analogous to Cosm’s use of global cues to facilitate bottom-up collective behavior, recent work in cyber-physical systems, swarm robotics, and other disciplines strives for guided self-

organization [50]. Note, however, that Cosm operates in open-loop fashion and does not require sensing and feedback to a global controller. Graph cellular automata and graph neural networks have some connections with Cosm given their simple rules, support for arbitrary graphs, and collective behavior. Related to Cosm’s SCS are works on dynamic neighborhoods in cellular automata [51] and the use of time-varying networks to enhance the ability of agents to synchronize [52] [53].

V. DISCUSSION

In this section, we offer qualitative insights summarizing Cosm’s dynamical operation and discuss some limitations of the approach.

A. Theoretical Intuition and Dynamical Perspective

While Cosm is introduced as a heuristic algorithm, its behavior can be interpreted through the lens of switched dynamical systems as follows.

1) Switched, Edge-Partitioned Dynamics. Cosm replaces a simultaneous update over all edges with a cyclic sequence of updates over edge-disjoint matchings. This induces a periodically switched dynamical system, where the active interaction set changes deterministically over time, shaping the system trajectory. The network is neither static nor chaotic—it is rhythmically reconfigured. One consequence of this construction is that the resulting dynamics are non-commutative: the final state after one sweep depends on the order in which the partitions are applied. This differs fundamentally from standard gradient descent, where all interactions are aggregated into a single update. With circular variables, where competing interactions can induce cancellation, this temporal decomposition allows individual couplings to act without interference during their respective sub-sweeps. As a result, SCS can mitigate the type of force cancellation described in Section II-A by ensuring that each interaction contributes a finite update over time rather than being averaged out instantaneously. From this perspective, SCS can be viewed as introducing a structured deterministic fluctuation induced by switching dynamics. The system no longer follows a smooth descent trajectory, but instead traverses a sequence of piecewise flows, which can enable transitions across energy barriers that would otherwise trap the basic gradient-based method.

2) Correlated Perturbations and Cluster Motion. The DWT mechanism introduces correlated perturbations that act on extended subsets of variables at a random reference angle in the continuous phase space. Unlike independent noise or random bit flips, DWT applies a coherent rotation to variables within two opposing angular windows. Because these windows are diametrically opposed, alignment and anti-alignment relationships among coupled variables within the pair of windows are preserved during the perturbation. This structure enables DWT to act as a form of cluster-level motion. In energy landscapes with frustration, locally consistent groups of variables (clusters) may become separated in phase space but remain internally aligned. DWT facilitates such clusters rotating collectively, promoting energy-improving transitions

that would require coordinated multi-variable flips in a binary formulation.

3) **Combined Effect: Exploration via Structured Non-Equilibrium Dynamics.** Taken together, SCS and DWT induce a form of structured non-equilibrium dynamics. SCS provides a deterministic, time-varying interaction structure while DWT injects intermittent, correlated perturbations that promote large-scale rearrangements. The annealing schedule on the step size α_t gradually reduces the magnitude of both effects. As the step size decreases, the magnitude of intra-sweep fluctuations is reduced, leading to a contraction of variable clusters in phase space. This damping effect allows initially overlapping clusters to separate along the unit circle, resolving ambiguities in their relative positions and enabling a consistent binary partition to emerge. This combination can be interpreted as enabling the system to traverse the energy landscape through a sequence of collective, temporally coordinated moves, rather than through purely local or purely random updates. Empirically, as illustrated in Fig. 5, such dynamics can result in non-local transitions involving contiguous clusters of variables, suggesting a mechanism for escaping metastable configurations in large sparse systems.

B. Limitations

The preceding intuition is most applicable in regimes where the problem graph is sparse and of bounded degree, such that the magnitude of intra-sweep fluctuations also remains bounded. In Cosm, the overall trajectory over a sweep is the result of a sequence of sub-sweep updates. In bounded-degree graphs, each variable participates in a limited number of interactions per sweep, and the resulting intra-sweep fluctuations—i.e., the cumulative angular displacement induced by successive sub-sweeps—remain moderate.

In contrast, in higher-degree or dense graphs, a variable may be involved in many interactions within a single sweep. This can lead to large intra-sweep fluctuations, where the net angular displacement becomes substantial (e.g., approaching or exceeding 180 degrees). Such large rotations can cause a variable to move across energetically meaningful boundaries in the circular representation, potentially placing it on the “wrong” side of the unit circle relative to its optimal binary assignment. Thus, Cosm’s performance degradation in dense graphs is not merely due to the overhead of a large number of interaction sets.

From this perspective, the effectiveness of Cosm relies in part on a balance between structured exploration and controlled update magnitude. With its bang-bang-like coupling and intra-sweep fluctuations, Cosm dynamics can bear a qualitative resemblance to chaotic chattering in sliding mode control, where rapid switching dynamics can induce oscillatory behavior [54]. While Cosm operates in a different setting, the analogy highlights the potential importance of controlling update amplitudes in systems governed by switching dynamics.

More broadly, these observations suggest that the use of circular variables may impose an implicit constraint on the allowable magnitude of sequential updates. When this constraint is violated—e.g., through high-degree connectivity or

insufficiently damped step sizes—the representation may fail to faithfully encode the underlying binary structure. A more formal characterization of intra-sweep fluctuation bounds, and their relationship to graph structure and algorithm parameters, remains an important direction for future work.

VI. CONCLUSION

We have introduced Collective Switched Motion (Cosm), a new heuristic for solving Ising-type optimization problems in specialized regimes. Cosm departs from conventional gradient-based approaches by employing periodically switched, edge-partitioned dynamics combined with structured perturbations, enabling efficient exploration of specialized energy landscapes. Although Cosm is not competitive or applicable for dense problems, it excels at handling large-scale, bounded-degree graph structures. Cosm’s SCS feature is reminiscent of matching-based update rules employed in other contexts; a key finding is the extent of its effectiveness, when combined with features such as DWT, in a sparse Ising-type context.

Experimental results demonstrate that Cosm achieves optimal solutions to the three largest longstanding Gset instances, and on the two largest Gset bounded-degree non-lattice instances attains best known solutions with substantially reduced time-to-target compared to existing methods. Cosm attains these solutions with consistent success rates. Moreover, on a set of synthetic benchmarks, Cosm outperforms a wide range of dynamical systems heuristics. While direct runtime comparisons across different hardware platforms and experimental protocols should be interpreted with caution, the results indicate that Cosm operates in a substantially different performance regime on the tested benchmark instances. In particular, even under conservative assumptions, the observed improvements in TTS and TTT are large relative to prior methods.

The structure of Cosm is well suited for parallel implementation, as updates operate on edge-disjoint subsets that can be processed concurrently. This suggests a pathway toward further acceleration on massively parallel hardware platforms, including GPUs, wafer-scale systems, and custom ASICs.

Future directions include benchmarking [1] [2] [13] [16], linking Cosm to meta-heuristic frameworks, and seeking connections to neural network training for models with binary weights [55] or natively circular weights [56]. The Cosm findings may be useful for gaining insight into the structure of sparse energy landscapes [57] and how to efficiently traverse them. While the present work focuses on sparse pairwise interactions, the underlying framework naturally extends to more general settings. In particular, the partitioned update mechanism can be generalized to handle higher-degree nodes through adaptive scheduling of interaction subsets, and extended to k -local objective functions by operating on collections of hyperedges. These directions provide a pathway toward applying Cosm to a broader class of combinatorial optimization problems.

These results indicate that Cosm represents a distinct and effective computational framework for sparse combinatorial optimization using structured, non-gradient dynamics. More

broadly, it highlights the potential of novel syntheses of switched dynamical systems, global mechanisms, and simple, locally-interacting variables for specialized collective computation.

VII. ACKNOWLEDGMENTS

The authors thank David Ferguson, Mohammad Sakib, Itay Hen, Ryan Epstein, Bryan Jacobs, Jonathan Machta, Sergey Novikov, Aaron Pesetski, Marc Sherwin, Matt French, and Federico Spedalieri for helpful discussions and suggestions. The authors thank Helmut Katzgraber for discussions and for providing the tile-planted problem instances. Nikhil Shukla's contributions were supported in part by National Science Foundation grant #243387.

APPENDIX A BACKGROUND

The problems of interest include sparse Ising, Max-Cut, and quadratic unconstrained binary optimization (QUBO) formulations. The Ising Hamiltonian is given in Eq. 1 in the main text. In this particular work, the problem instances do not contain local fields. Ising problems can alternatively be formulated as weighted Max-Cut problems

$$\max \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij}(1 - x_i x_j) \quad (11)$$

with n binary variables $x_i \in \{-1, +1\}$ [28]. When converting between Ising and Max-Cut formulations, $w_{ij} = -J_{ij}$. Note that the Gset problem files use the Max-Cut sign convention for the weights.

Given the success probability $P_s < 1$ of reaching a solution, the number of repetitions R_{99} required to reach the solution with at least 99% probability is

$$R_{99} = \frac{\log(1 - 0.99)}{\log(1 - P_s)}. \quad (12)$$

If $P_s = 1$, r is taken to be 1 trial. As an intrinsic measure of algorithmic performance, we use sweeps-to-target which is defined as

$$STT(target) = S_{trial} \cdot R_{99} \quad (13)$$

where S_{trial} is the number of sweeps per trial. Sweeps-to-solution (STS) has the same form as STT; we use the term STS when the target is the optimal solution. To characterize our specific Cosm-CPU implementation, we employ wall clock time-to-target defined as

$$Wall\ clock\ TTT(target) = t_{trial} \cdot R_{99} \quad (14)$$

where t_{trial} is the average execution time per trial over a batch of parallel trials. Wall clock time-to-solution (TTS) has the same form as TTT; we use the term TTS when the target is the optimal solution.

APPENDIX B EXPERIMENTAL DETAILS – GSET EXPERIMENTS

Cosm was implemented in C and compiled with GNU gcc 8.5.0 (-O3 -ffast-math option). All floating-point variables in Gset experiments used single precision. The processor employed was an Intel Xeon Gold 6544Y CPU with 32 physical cores (16 cores \times 2 sockets), 2 TB of memory, and a 3600 MHz clock frequency. Gset test campaigns generally used 64 parallel threads. Multiple replicas (independent dynamical systems, each associated with a different trial) were laid out in an interleaved fashion in memory to amortize the penalties of memory transactions; the number of parallel dynamical systems processed in interleaved fashion per thread was eight for G61 and four for all other instances. Given 64 threads, 512 trials were in flight at a time for G61 and 256 trials for all other instances.

All 2D lattice instances used a 4-color edge coloring with the same regular pattern. For non-lattice instances, a greedy edge coloring heuristic was used; the number of edges in each edge color partition is shown in Table VIII. Alternative heuristics are possible that use additional colors or balance the number of edges per color.

TABLE VIII
NUMBER OF EDGES PER COLOR USING GREEDY EDGE-COLORING
HEURISTIC

Gset Instance	Number of Edges per Color ($ E_1 $, $ E_2 $, ... $ E_C $)
G61	3036, 2862, 2608, 2331, 1949, 1561, 1150, 777, 467, 251, 103, 37, 12, 4
G70	3440, 2747, 1925, 1135, 523, 170, 44, 11, 4

TABLE IX
COSM SOLVER PARAMETER SETTINGS COMMON TO ALL GSET
EXPERIMENTS

Parameter	Value
Initial step size α_0	14 degrees
Final step size	0 degrees
Size of each DWT window	90 degrees
Number of bisectors for binarization	100

TABLE X
DUAL WINDOW TWIST (DWT) PERTURBATION PARAMETER SETTINGS

Parameter	G61	G70	G72	G77	G81
Ratio of DWT step to α	1.5	1.0	0.5	0.5	0.5
Period of DWT (sweeps)	3	6	3	3	3

TABLE XI
SWEEPS/TRIAL PARAMETER SETTINGS FOR GSET STS (STT) AND TTS
(TTT) EXPERIMENTS

Parameter	G61	G70	G72	G77	G81
Sweeps/trial ($\times 10^6$)	0.5	0.2	1.0	1.0	2.0

The Cosm solver parameter settings are given in Tables IX, X, and XI. We employed step size schedules that started from

an initial value of 14 degrees and ramped down linearly to 0. The step size was updated once per sweep and stayed constant within a sweep. Solutions were read out at the end of each trial. Variables were binarized by performing 100 bisections of the unit circle (spaced by 1.8 degrees, covering an entire semicircle) and taking the best solution. We selected a value of 100 as a conservative value that empirically was sufficient to ensure high performance.

APPENDIX C

EXPERIMENTAL DETAILS – TILE-PLANTED PROBLEM EXPERIMENTS

Cosm was tested on the tile-planted problem instances generated by the chook tool [46] and described in Ref. [12]. The problem instances were provided by one of the authors (H.K.) of Ref. [12]. There are 100 instances at each problem size N from 16 to 1024 variables, where $N = L^2$ and $L \in \{4, 6, 8, \dots, 32\}$. The Ising ground state energy was provided for each instance.

Using a MATLAB-based implementation of Cosm, we confirmed that Cosm reaches the ground state for all instances and all problem sizes ($L \in \{4, 6, 8, \dots, 32\}$).

For characterizing scaling, the selected problem sizes ranged from 144 to 400 variables ($L \in \{12, 14, 16, 18, 20\}$) to match Ref. [12]. At each size, the number of sweeps was tuned to optimize sweeps-to-solution and an instance with median P_s was identified using MATLAB. The execution times of the five median instances were then measured using a C implementation of Cosm. (Note that while MATLAB was used for parameter tuning and identifying median instances, all reported timing results are based on a C implementation.) The hardware used was the same CPU described in Appendix B (Intel Xeon Gold 6544Y). Unlike the Gset experiments, C floating-point variables during these particular experiments defaulted to double precision. Solver parameter settings included the following: initial step size $\alpha_0 = 25$ degrees; ratio of DWT step to $\alpha = 1.0$; DWT period = 3 sweeps. Average execution times per trial were measured by performing 10,000 sequential trials of each median instance using a single job. For lattice instances, Cosm used a standard edge coloring that is trivial to assign; the time required to assign colors in pre-processing is not included in execution time measurements. Here, only a single trial was solved at a time (no interleaving of multiple trials); use of sequential trials and a single job matches Ref. [12] which used a single CPU core. The observed P_s and average execution time were used to calculate the median TTS. The measured TTS values for sizes $L = 12, 14, 16, 18, 20$ are 6.101 ms, 14.589 ms, 28.739 ms, 53.492 ms, and 78.159 ms, respectively. To match the methodology described in [12], we fit $\log_{10}TTS$ data to a linear fit $a + bL$ and obtain a slope of 0.139, representing the scaling exponent b . The coefficient of determination R^2 is 0.981 and the lower and upper 95% confidence bounds are 0.104 and 0.175.

REFERENCES

- [1] M. J. Schuetz, J. K. Brubaker, and H. G. Katzgraber, “Combinatorial optimization with physics-inspired graph neural networks,” *Nature Machine Intelligence*, vol. 4, no. 4, pp. 367–377, 2022.
- [2] J. Wurtz, P. L. Lopes, C. Gorgulla, N. Gemelke, A. Keesling, and S. Wang, “Industry applications of neutral-atom quantum computing solving independent set problems,” *arXiv preprint arXiv:2205.08500*, 2024.
- [3] R. Barrass, H. Nagarajan, and C. Coffrin, “Leveraging quantum computing for accelerated classical algorithms in power systems optimization,” in *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 52–67, Springer, 2025.
- [4] J. Charfreitag, M. Jünger, S. Mallach, and P. Mutzel, “McSparse: Exact solutions of sparse maximum cut and sparse unconstrained binary quadratic optimization problems,” in *2022 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pp. 54–66, 2022.
- [5] D. Rehfeldt, T. Koch, and Y. Shinano, “Faster exact solution of sparse MaxCut and QUBO problems,” *Mathematical Programming Computation*, vol. 15, pp. 445–470, 2023.
- [6] J. Choi and Y. Ye, “Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver,” tech. rep., Department of Management Sciences, University of Iowa, Iowa City, IA 52242, USA, 2000. <https://web.stanford.edu/~yye/yye/cgsdp1.pdf>.
- [7] Z. Zhu, A. Ochoa, and H. Katzgraber, “Efficient cluster algorithm for spin glasses in any space dimension,” *Physical Review Letters*, vol. 115, p. 077201, 2015.
- [8] U. Benlic and J.-K. Hao, “Breakout local search for the max-cut problem,” *Engineering Applications of Artificial Intelligence*, vol. 26, Mar. 2013.
- [9] H. Goto *et al.*, “High-performance combinatorial optimization based on classical mechanics,” *Science Advances*, vol. 7, no. 6, 2021.
- [10] N. Mohseni, P. McMahon, and T. Byrnes, “Ising machines as hardware solvers of combinatorial optimization problems,” *Nature Reviews Physics*, vol. 4, June 2022.
- [11] H. Goto, R. Hidaka, and K. Tatsumura, “Edge-of-chaos-enhanced quantum-inspired algorithm for combinatorial optimization,” *Phys. Rev. Appl.*, pp. –, Feb 2026.
- [12] J. Hou, A. Barzegar, and H. G. Katzgraber, “Direct comparison of stochastic driven nonlinear dynamical systems for combinatorial optimization,” *Physical Review E*, vol. 112, no. 3, p. 035301, 2025.
- [13] B. Tasseff, T. Albash, Z. Morrell, M. Vuffray, A. Y. Lokhov, S. Misra, and C. Coffrin, “On the emerging potential of quantum annealing hardware for combinatorial optimization,” *Journal of Heuristics*, vol. 30, no. 5, pp. 325–358, 2024.
- [14] J. Vodeb, V. Eržen, T. Hrga, and J. Povh, “Accuracy and performance evaluation of quantum, classical and hybrid solvers for the max-cut problem,” *arXiv preprint arXiv:2412.07460*, 2024.
- [15] Q.-G. Zeng, X.-P. Cui, B. Liu, *et al.*, “Performance of quantum annealing inspired algorithms for combinatorial optimization problems,” *Communications Physics*, vol. 7, p. 249, 2024.
- [16] H. Munoz-Bauza and D. Lidar, “Scaling advantage in approximate optimization with quantum annealing,” *Physical Review Letters*, vol. 134, 2025.
- [17] J. Weidenfeller, L. C. Valor, J. Gacon, C. Tornow, L. Bello, S. Woerner, and D. J. Egger, “Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware,” *Quantum*, vol. 6, p. 870, 2022.
- [18] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, “A review on quantum approximate optimization algorithm and its variants,” *Physics Reports*, vol. 1068, pp. 1–66, 2024.
- [19] Nobel Prize Outreach, “Giorgio parisi – facts – 2021.” <https://www.nobelprize.org/prizes/physics/2021/parisi/facts/>, 2026. Accessed: 2026-04-10.
- [20] H. G. Katzgraber, “Viewing vanilla quantum annealing through spin glasses,” *Quantum Science and Technology*, vol. 3, no. 3, p. 030505, 2018.
- [21] E. Dahlberg, I. González-Adalid Pemartín, E. Marinari, G. Parisi, F. Ricci-Tersenghi, V. Martin-Mayor, J. Moreno-Gordo, R. Orbach, I. Paga, J. Ruiz-Lorenzo, *et al.*, “Spin-glass dynamics: Experiment, theory, and simulation,” *Reviews of Modern Physics*, vol. 97, no. 4, p. 045005, 2025.
- [22] T. Wang and J. Roychowdhury, “Oim: Oscillator-based ising machines for solving combinatorial optimisation problems,” in *International Conference on Unconventional Computation and Natural Computation*, pp. 232–256, Springer, 2019.
- [23] G. Csaba and W. Porod, “Coupled oscillators for computing: A review and perspective,” *Applied physics reviews*, vol. 7, no. 1, 2020.
- [24] K. M. Zick, “Ring oscillator-based Ising machine system,” January 2023. US Patent US11545963B1.

- [25] J. Ouyang, Y. Liao, X. Feng, Y. Li, K. Cui, F. Liu, H. Sun, W. Zhang, and Y. Huang, "Programmable and reconfigurable photonic simulator for classical xy models," *Physical Review Applied*, vol. 22, no. 2, p. L021001, 2024.
- [26] A. Shukla, M. Erementschouk, and P. Mazumder, "Non-binary dynamical ising machines for combinatorial optimization," *Physica D: Nonlinear Phenomena*, vol. 481, p. 134809, 2025.
- [27] S. Sreedhara and J. Roychowdhury, "A novel oscillator ising machine coupling scheme for high-quality optimization," in *International Conference on Unconventional Computation and Natural Computation*, pp. 203–218, Springer, 2024.
- [28] S. Burer, R. D. C. Monteiro, and Y. Zhang, "Rank-two relaxation heuristics for MAX CUT and other binary quadratic programs," *SIAM Journal on Optimization*, vol. 12, pp. 503–521, 2002.
- [29] J. Houdayer, "A cluster monte carlo algorithm for 2-dimensional spin glasses," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 22, no. 4, pp. 479–484, 2001.
- [30] D. Van Essendelft, H. Almolyki, W. Shi, T. Jordan, M.-Y. Wang, and W. A. Saidi, "Record acceleration of the two-dimensional ising model using a high-performance wafer-scale engine," *Computer Physics Communications*, vol. 315, p. 109734, 2025.
- [31] "Gset benchmark problems." <https://web.stanford.edu/~yyye/yyye/Gset/>. Retrieved April 2025.
- [32] S. J. Benson, Y. Ye, and X. Zhang, "Mixed linear and semidefinite programming for combinatorial and quadratic optimization," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 515–544, 1999.
- [33] F. Barahona, "On the computational complexity of ising spin glass models," *Journal of Physics A: Mathematical and General*, vol. 15, no. 10, pp. 3241–3253, 1982.
- [34] H. Khoshbakhht and M. Weigel, "Domain-wall excitations in the two-dimensional ising spin glass," *Physical Review B*, vol. 97, no. 6, p. 064410, 2018.
- [35] K. M. Zick, "Improved Sparse Ising Optimization," 2023.
- [36] K. M. Zick, "Performance report of heuristic algorithm that cracked the largest Gset Ising problems (G81 cut=14060)," 2025.
- [37] K. P. Kalinin, J. Gladrow, J. Chu, J. H. Clegg, D. Cletheroe, D. J. Kelly, B. Rahmani, G. Brennan, B. Canakci, F. Falck, *et al.*, "Analog optical computer for ai inference and combinatorial optimization," *Nature*, vol. 645, no. 8080, pp. 354–361, 2025.
- [38] V. Shylo *et al.*, "Teams of global equilibrium search algorithms for solving the weighted maximum cut problem in parallel," *Cybernetics and Systems Analysis*, vol. 51, pp. 16–24, 2015.
- [39] V. P. Shylo and O. V. Shylo, "Algorithm portfolios and teams in parallel optimization," in *Optimization Methods and Applications* (S. Butenko, P. Pardalos, and V. Shylo, eds.), vol. 130 of *Springer Optimization and Its Applications*, Cham: Springer, 2017.
- [40] K. M. Zick, "Systems and method for solving spin glasses and other sparse Ising problems," February 2024. US Patent application US2025025886A1.
- [41] O. Goudet, A. Goeffon, and J.-K. Hao, "A large population island framework for the unconstrained binary quadratic problem," *Computers & Operations Research*, vol. 168, p. 106684, 2024.
- [42] C. Chen, R. Chen, T. Li, R. Ao, and Z. Wen, "Monte carlo policy gradient method for binary optimization," *arXiv preprint arXiv:2307.00783*, 2023.
- [43] C. Chen, R. Chen, T. Li, R. Ao, and Z. Wen, "A monte carlo policy gradient method with local search for binary optimization: C. chen et al.," *Mathematical Programming*, pp. 1–57, 2025.
- [44] D. Liu, W. Yang, H. Wang, Y. Du, Y. Wang, Z. Lü, and J.-K. Hao, "Enhanced open-source scatter search algorithm for solving quadratic unconstrained binary optimization problems," *Computers & Operations Research*, vol. 182, p. 107137, 2025.
- [45] J. Hou, A. Barzegar, and H. G. Katzgraber, "Direct comparison of stochastic driven nonlinear dynamical systems for combinatorial optimization," *arXiv preprint arXiv:2503.15427*, 2025.
- [46] D. Perera, I. Akpabio, F. Hamze, S. Mandra, N. Rose, M. Aramon, and H. G. Katzgraber, "Chook—a comprehensive suite for generating binary optimization problems with planted solutions," *arXiv preprint arXiv:2005.14344*, 2020.
- [47] D. Perera, F. Hamze, J. Raymond, M. Weigel, and H. G. Katzgraber, "Computational hardness of spin-glass problems with tile-planted solutions," *Physical Review E*, vol. 101, no. 2, p. 023316, 2020.
- [48] S. Chowdhury, N. A. Aadit, A. Grimaldi, E. Raimondo, A. Raut, P. A. Lott, J. H. Mentink, M. M. Rams, F. Ricci-Tersenghi, M. Chiappini, *et al.*, "Pushing the boundary of quantum advantage in hard combinatorial optimization with probabilistic computers," *Nature Communications*, vol. 16, no. 1, p. 9193, 2025.
- [49] C. Fan, M. Shen, Z. Nussinov, Z. Liu, Y. Sun, and Y.-Y. Liu, "Searching for spin glass ground states through deep reinforcement learning," *Nature communications*, vol. 14, no. 1, p. 725, 2023.
- [50] C. Gershenson, "Guiding the self-organization of cyber-physical systems," *Frontiers in Robotics and AI*, vol. 7, p. 41, 2020.
- [51] S. Dantchev, "Dynamic neighbourhood cellular automata," *The Computer Journal*, vol. 54, no. 1, pp. 26–30, 2011.
- [52] I. V. Belykh, V. N. Belykh, and M. Hasler, "Blinking model and synchronization in small-world networks with a time-varying coupling," *Physica D: Nonlinear Phenomena*, vol. 195, no. 1–2, pp. 188–206, 2004.
- [53] X. Wang, Z. Wang, S. Dang, and J. Dai, "Frequency-dependent synchronization in blinking networks: insights from hindmarsh–rose, lorenz, and rössler systems," *The European Physical Journal B*, vol. 98, no. 8, p. 163, 2025.
- [54] A. Rehan, I. Boiko, and Y. Zweiri, "Chaotic chattering in sliding mode control systems," *IEEE Transactions on Automatic Control*, vol. 69, no. 11, pp. 7925–7931, 2024.
- [55] J. C. Villumsen and Y. Sugita, "Quadratic unconstrained binary optimization for training and regularisation of binary neural networks," *arXiv preprint arXiv:2601.00449*, 2026.
- [56] F. Léonard, A. S. Backer, E. J. Fuller, C. Teeter, and C. M. Vineyard, "Co-design of free-space metasurfaces optical neuromorphic classifiers for high performance," *ACS Photonics*, vol. 8, no. 7, pp. 2103–2111, 2021.
- [57] D. Dobrynin, A. Renaudineau, M. Hizzani, D. Strukov, M. Mohseni, and J. P. Strachan, "Energy landscapes of combinatorial optimization in ising machines," *Physical Review E*, vol. 110, no. 4, p. 045308, 2024.

BIOGRAPHY SECTION

Kenneth M. Zick, Ph.D. is the Research Director of Transformational Computing at the University of Southern California, Information Sciences Institute (USC ISI). He earned a Ph.D. in Computer Science and Engineering from the University of Michigan, USA. Dr. Zick serves on the Microsystems Exploratory Council.

Nikhil Shukla is an Associate Professor in the department of Electrical and Computer Engineering at the University of Virginia. His research interests lie in emerging devices and circuits as well as developing new approaches for energy efficient computing and storage.

Alexander Marakov is a physicist at the Microelectronics Design and Applications Business Area at Northrop Grumman Systems Corporation, Linthicum, MD, where he coordinates device and physical modeling of superconductive electronics and cryogenic systems across multiple programs. His broader technical interests include quantum heat engines, noise-biased qubits, numerical methods, and systems engineering.