Supporting Higher-Order Interactions in Practical Ising Machines

Nafisa Sadaf Prova, Hüsrev Cılasun, Abhimanyu Kumar, Ahmet Efe,
Sachin S. Sapatnekar, Ulya R. Karpuzcu
University of Minnesota, Twin Cities
{prova026, cilas001, kumar663, efe00002, sachin, ukarpuzc}@umn.edu

***Abstract*—Ising machines as hardware solvers of combinatorial optimization problems (COPs) can efficiently explore large solution spaces due to their inherent parallelism and physics-based dynamics. Many important COP classes such as satisfiability (SAT) assume arbitrary interactions between problem variables, while most Ising machines only support pairwise (second-order) interactions. This necessitates translation of higher-order interactions to pair-wise, which typically results in extra variables not corresponding to problem variables, and a larger problem for the Ising machine to solve than the original problem. This in turn can significantly increase time-to-solution and/or degrade solution accuracy. In this paper, considering a representative CMOS-compatible class of Ising machines, we propose a practical design to enable direct hardware support for higher order interactions. By minimizing the overhead of problem translation and mapping, our design leads to up to 4 × lower time-to-solution without compromising solution accuracy.**

I. Introduction

Solutions to combinatorial optimization problems (COP) common in scheduling, logistics, or resource allocation, come from a finite discrete search space, and minimize or maximize an objective function under constraints. COPs are amongst the hardest optimization problems; most of the COPs belong to the computational complexity classes NP-complete or NP-hard. At the same time, problem sizes of practical interest prohibit exhaustive search in the discrete solution space. As a result, classical solvers typically rely on approximations and cannot always guarantee to find solutions.

Ising machines as COP solvers can efficiently explore large solution spaces by directly leveraging the inherent parallelism and natural dynamic behavior of Ising-model compliant physical systems which stabilize at minimum energy states. The Ising model, originally developed in statistical mechanics to describe ferromagnetism, represents each solution as a binary vector, where each element maps to a spin state in such a way that the extremes of the objective function match the minimum energy of the system [1]. As the system naturally favors spin configurations corresponding to the minimum energy states, it essentially solves a COP by converging on a minimum energy state upon perturbation.

Ising machines can primarily handle pairwise (second-order) interactions between spins. This limitation arises from the original formulation, which simplifies spin interactions to second-order terms. However, many real-world COPs from computational biology [2], machine learning [3] or financial modeling [4] involve higher-order interactions between three or more variables. Such higher-order interactions are typically approximated or transformed into equivalent pairwise interactions through a process known as quadratization, by introducing auxiliary variables which do not correspond to problem variables. Quadratization can also increase the range of core model parameters (such as interaction coefficients quantifying the strength of interactions) to exceed physical representation capabilities of the underlying Ising machine. This results in a larger, and often more complex, problem for the Ising machine to solve than the original problem, which in turn can significantly increase time-to-solution and/or degrade solution accuracy. In this paper we propose a practical design to enable High-Order Ising Machines (HOIM) which natively support higher order interactions.

Depending on the underlying physical system, Ising machines come in different flavors. In this paper, we confine our study to practical Ising machines based on conventional technologies and operating at the room temperature: Oscillatory Ising Machines (OIM) which represent a network of coupled oscillators [5]–[7]. These oscillators are engineered such that their interactions encode the constraints and objectives of a given combinatorial optimization problem. As the system evolves, the collective dynamics of the oscillators guide it toward a stable configuration, ideally corresponding to the optimal or near-optimal solution. Each oscillator maps to a spin, where two distinct values of the oscillator phase encode the two distinct spin states (spin-up, +1 or spin-down, -1). To determine whether the oscillator phase at a given point in time encodes spin-up or spin-down, we need a reference – a pre-specified fixed phase against which each oscillator's phase is compared. OIMs typically use a set of phase-locked reference oscillators to this end.

Dedicated hardware elements called *couplers* implement pairwise spin interactions such that oscillators can influence each other's phase angles. A key challenge for augmenting OIM couplers with support for higher-order interactions is distributing the reference phase to each coupler across the network, which requires increasingly complex circuitry and interconnects at scale. As a result, OIMs from the literature only support pairwise coupling – and must rely on problem translation to be able to solve COPs featuring higher-order interactions, which incurs forbidding overheads at scale. Throughout the paper, we will refer to these designs as Pairwise Oscillatory Ising Machines (POIM).

We will next demonstrate how the proposed High-order Oscillatory Ising Machine (HOIM) eliminates the need for a reference in implementing higher-order couplers, by simply tracking the bitwise parity among spin states (i.e., oscillator phases). In mapping higher-order COPs, such higher-order couplers facilitate seamles problem translation which prevents the introduction of ancillary variables or the expansion of interaction coefficient ranges beyond machine capability. By minimizing the overhead of problem translation and mapping, when compared to POIM (which can only directly support pair-wise interactions and must use problem translation to solve COPs featuring higher-order interactions), HOIM can lead to up to 4 × lower time-to-solution without compromising solution accuracy. In the following, we will cover the background in Section II; basic operation principle of HOIM in Section III-B; quantitative analysis in Sections IV and V; related work in Section VI; and a summary and discussion of our findings in Section VII.

II. Background

Ising Model [8]–[10] as a mathematical abstraction represents a material as a collection of molecules, and describes any physical system composed of pairwise interacting discrete elements (e.g., spins) [11]. Each molecule has a spin that can be either aligned or anti-aligned with an external magnetic field. The spins can interact with each other in pairs. A Hamiltonian function is used to describe the energy of an n-spin system with $s = [s_1, s_2, ..., s_n]$, as:

$$H(s) = -\sum_{<i \neq j>} J_{ij} s_i s_j - \sum_i h_i s_i \quad \text{where} \quad i, j \in [1, 2, ..., n]$$

s_i, the spin of the i^{th} molecule, can be either -1 or $+1$. h_i captures the strength of the *local* field at the i^{th} molecule; J_{ij}, of the *interaction* field between neighboring spins i and j. h_i and J_{ij} are real-valued constants.

The system evolves toward the ground state, represented by binary vector $\mathbf{s}$, which minimizes the energy function $H(\mathbf{s})$ at equilibrium. The interaction between neighboring spins depends on the sign of the coupling coefficient J_{ij}. Specifically, if J_{ij} is positive (ferromagnetic), neighboring spins tend to align such that $s_i s_j = +1$; if J_{ij} is negative (antiferromagnetic), they prefer to anti-align, resulting in $s_i s_j = -1$. In both cases, this alignment contributes by a negative term $-|J_{ij}|$ to the total energy, thereby reducing $H(\mathbf{s})$. The spins are situated on a finite-dimensional lattice, often visualized as a graph. The Ising model is mathematically equivalent to a Quadratic Unconstrained Binary Optimization (QUBO) problem, which is characterized by

$$H(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} \quad \text{where} \quad x_i = (s_i + 1)/2$$

$$h_i = Q_{ii}/2 + \sum_{j=1}^{n} (Q_{ij} + Q_{ji})/4 \quad \text{and} \quad J_{ij} = Q_{ij}/4$$

for $\mathbf{x} = [x_1, \cdots, x_n]^T \in \{0, 1\}^n$ and $Q \in \mathbb{R}^{n \times n}$ for n spins.

Ising model based solvers employ distinct approaches to implement convergence dynamics. As combinatorial optimization problems grow in size, exhaustive search – i.e., evaluating $H(s)$ for all possible configurations of s within the search space – becomes computationally infeasible. Software solvers typically utilize probabilistic methods such as simulated annealing [12], which involves iteratively flipping selected spin states according to predefined convergence criteria and stochastic transition rules to explore the energy landscape. Hardware solvers can be broadly categorized into two classes. The first class [13]–[17] treats the Ising model purely as a mathematical abstraction, mirroring software-based approaches but with hardware acceleration. The second class, Ising machines, implements physical systems that inherently follow Ising model dynamics, including quantum [18], [19] and quantum-inspired architectures [20]–[23]. In these systems, the evolution of spin states is governed directly by the physics of the underlying medium. This paper focuses on the latter category.

Formulating a COP using the Ising model involves representing problem variables as binary-valued spins and encoding constraints through pairwise interaction strengths (J_{ij}) and local fields (h_i). A wide range of NP-complete and NP-hard problems, including all 21 of Karp's NP-complete problems, have been successfully mapped to this model [1]. The goal is to construct an energy function whose minimum-energy state(s) correspond to the optimal solution(s) of the original problem. When this mapping is done effectively, a physical system governed by the Ising model can, with high probability, solve complex combinatorial optimization problems by evolving toward equilibrium, even under the influence of thermal fluctuations [24]. However, to comprehensively characterize the system's behavior, it is essential to specify the dynamical evolution of the energy function $H(\mathbf{s})$ and the spin configurations over time, as the static formulation alone does not capture the temporal aspects of the system's trajectory toward equilibrium.

For a given optimization problem, multiple Ising or QUBO formulations may be possible. Both Ising and QUBO models inherently represent pairwise interactions between spins, and by extension, between problem variables. However, many combinatorial optimization problems involve interactions among more than just two variables. A notable example is SAT (Boolean satisfiability), a foundational problem in computer science [25], which was the first problem proven to be NP-complete and served as the starting point for the theory of NP-completeness. Any other NP-complete problem can be transformed into SAT using a polynomial-time reduction [26]. Accordingly, we use SAT to benchmark the proposed designs in this paper.

SAT problems ask whether there exists an assignment of truth values (true or false) to n Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$ that satisfies a given Boolean formula $f(x_1, x_2, \ldots, x_n)$, which is typically given in Conjunctive Normal Form (CNF). Each clause in the CNF formula expresses higher-order interactions between problem variables by correlating literals (variables or their negations) using Boolean OR ($\vee$). For 3-SAT

$$f(x_1, \cdots, x_n) = C_1 \wedge C_2 \wedge \cdots \wedge C_m \text{ with } X = \{x_1, \cdots, x_n\}$$

applies, where each clause $C_i = l_1 \vee l_2 \vee l_3$ can have at most 3 literals $l_1, l_2, l_3 \subset X \cup \neg X$ (and $\wedge$ depicts Boolean AND). Generally, clauses in a k-SAT problem can contain at most k literals. 3-SAT (k = 3) is most commonly studied, and any k-SAT instance with k $>$ 3 can be efficiently reduced to an equivalent 3-SAT problem in polynomial time.

Different mathematical formulations vary in how they translate higher-order (third order or *cubic* for 3-SAT) interactions among problem variables into pairwise interactions among spins. This is commonly achieved by introducing ancillary variables, which can significantly increase the size of the translated problem compared to the original problem. For example, one of the most compact 3-SAT QUBO formulations [27] results in an (m+n) variable problem for an original problem with m clauses and n variables, by adding an ancillary variable x_a for each clause to capture energy levels corresponding to third-order interactions.

III. High Order Oscillatory Ising Machines

A. Macroscopic View

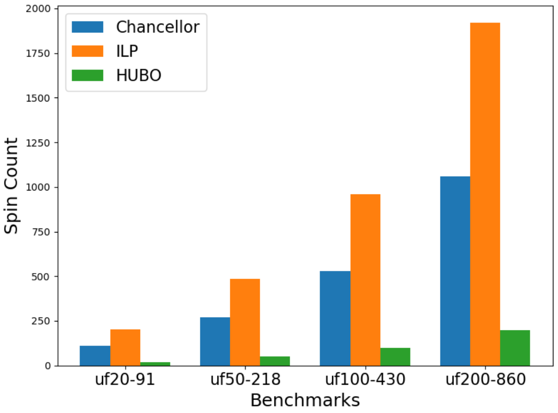


Fig. 1. Ising spin (equivalently, QUBO variable) count in the translated problem as a function of problem size, using different formulations for mapping higher-order SAT interactions. x-axis covers 3-SAT problems from SATLIB [28] of increasing problem size.

We will start our discussion with an illustrative example using SAT as a representative COP featuring higher-order interactions between problem variables. For problem translation, we first establish a correspondence between Boolean variables in the SAT formulation and binary or spin variables in QUBO or Ising models. Each SAT clause induces a constraint, which should be expressed within the pairwise interaction limits of QUBO or Ising formulations as a *quadratic* objective function over binary variables (which boils down to a clause specific Hamiltonian, with the order of interactions being limited to at most two), and satisfying assignments must minimize this objective function, by construction. This is when ancillary variables are introduced.

Considering the generic 3-SAT clause $(x_1 \vee x_2 \vee x_3)$, translating the cubic interaction among the problem variables x_1, x_2, and x_3 into the equivalent quadratic expression (i.e., *quadratization*), typically results in new binary variables (ancillaries) to express the cubic interaction through extra terms which penalize unsatisfying assignments [29]. How such extra terms look like and how many ancillary variables each necessitate to encode problem-specific correlations among problem variables depend on the mathematical formulation. A typical problem contains many such clauses with possibly overlapping higher-order interactions, which complicates mathematical formulation further.

To translate an n-variable m-clause 3-SAT instance, even the highly compact formulation proposed by Chancellor [27] introduces an ancillary variable for each clause, resulting in a total of n+m variables in the translated problem. Alternatively, in the Integer Linear Programming (ILP)-based formulation proposed by [29], the encoding uses two ancillary variables per clause to model clause Hamiltonians as equality constraints, yielding a total of n+2m variables in the translated problem. These additional variables ensure that multi-variable logical expressions are correctly transformed into pairwise (quadratic) energy penalties, making the SAT-to-Ising (or equivalently QUBO) mapping both physically implementable and mathematically correct. In contrast to these formulations, mapping a SAT instance to HOIMs requires to solve a Polynomial Unconstrained Binary Optimization (PUBO) [30] problem instead of QUBO – which we will generally refer to as Higher-order Unconstrained Binary Optimization (HUBO) thoughout the paper, and which eliminates the need for ancillary variables by definition.

Figure 1 shows spin count in the translated problem as a function of problem size, using different formulations for mapping higher-order SAT interactions. x-axis covers 3-SAT problems from SATLIB [28] of increasing problem size – 20, 50, 100, and 200 variables. We consider QUBO (Chancellor, ILP) and HUBO formulations. The plot clearly shows that as the problem size increases from 20 to 200 variables, HUBO formulation significantly reduces the overall spin count by eliminating the need for ancillary spins. Specifically, HUBO requires 5.5×(5.3×) fewer spins than the most efficient QUBO formulation (Chancellor) for 20 (200) variable problems, highlighting its scalability and encoding efficiency for larger SAT instances.

B. Microscopic View

We next take a closer look into 3-SAT as an illustrative example. The first step in translating a 3-SAT problem to QUBO or Ising model is converting it to CUBO (Cubic Unconstrained Binary Optimization) or CUSO (Cubic Unconstrained Spin Optimization) formulations [27], which are mathematically equivalent. CUSO translates Boolean SAT variables to spin variables $s_i \in \{-1, +1\}$; CUBO, to binary variables $x_i \in \{0, 1\}$. Both formulations can describe cubic optimization problems such as 3-SAT, which feature at most third-order interactions among problem variables. There is a

TABLE I

TRUTH TABLE FOR AN EXAMPLE 3-SAT CLAUSE $(x_1 \vee x_2 \vee x_3)$ EXPRESSED IN CUBO VS. CUSO FORMAT. HAMILTONIAN ENERGY (H_B OR H_S RESPECTIVELY) REACHES ITS MAXIMUM WHEN THE BINARY OR SPIN VARIABLES FAIL TO SATISFY THE CLAUSE.

CUBO (Cubic Unconstrained Binary Optimization)			
x_3	x_2	x_1	H_B
0	0	0	0
0	0	1	-1
0	1	0	-1
0	1	1	-1
1	0	0	-1
1	0	1	-1
1	1	0	-1
1	1	1	-1

CUSO (Cubic Unconstrained Spin Optimization)			
s_3	s_2	s_1	H_S
-1	-1	-1	+7
-1	-1	+1	-1
-1	+1	-1	-1
-1	+1	+1	-1
+1	-1	-1	-1
+1	-1	+1	-1
+1	+1	-1	-1
+1	+1	+1	-1

one-to-one correspondence between the SAT variables and the respective CUSO spins or CUBO variables. CUBO and CUSO formulations convert each SAT clause to a Hamiltonian, H_B or H_S[1], respectively:

$$H_B = -x_1x_1 - x_2x_2 - x_3x_3 + x_1x_2 + x_2x_3 + x_1x_3 - x_1x_2x_3.$$

$$H_S = -s_1 - s_2 - s_3 + s_1s_2 + s_2s_3 + s_1s_3 - s_1s_2s_3.$$

Only assignments of binary or spin variables that satisfy the respective clause (i.e., make it evaluate to logic one) can minimize the clause-specific H_B or H_S. As illustrated in Table I, the Hamiltonians H_B or H_S reach the maximum for the binary or spin assignments which fail to satisfy the clause.

This formulation gives rise to three different types of terms in the Hamiltonian equations: *diagonal* terms in the form of $(x_1x_1$ or $s_1, \ \ldots)$; *quadratic* terms in the form of $(x_1x_2$ or $s_1s_2, \ldots)$; and *cubic* terms in the form of $(x_1x_2x_3$ or $s_1s_2s_3)$. Quadratic and cubic terms represent quadratic (two-order or pairwise) and cubic (three-order) spin or variable interactions. Ising machines rely on dedicated hardware elements called *couplers* to implement such spin or variable interactions in hardware.

For typical oscillatory Ising machines (OIMs) where oscillator phases represent spin states, however, couplers can only directly map diagonal or quadratic terms. In mathematical terms, quadratic couplers correlate two oscillators i and j by adjusting the degree of correlation between their phases according to the (actual or translated) value of the interaction coefficient J_{ij} from the original Ising formulation. Couplers are critical design components, as the OIM system can only evolve to a solution based on the interaction between the oscillator phases. In other words, interactions between the oscillator phases guide the OIM toward a global solution over time.

Quadratic couplers in state-of-the-art OIMs enforce spin interactions as follows: When two oscillators are connected via a coupler, a positive coupling ($J_{ij} > 0$) causes the oscillators to lock into the same phase (s_i = s_j), while a negative coupling ($J_{ij} < 0$) makes them settle into opposite phases (s_i = -s_j). This behavior mirrors the minimization of the Ising Hamiltonian term H = $-J_{ij} \cdot s_i s_j$, effectively implementing spin logic in hardware without explicit digital computation. The most efficient state-of-the-art OIM couplers use transmission gates to implement the quadratic coupling logic accordingly [7], [31].

Figure 2 demonstrates how such OIM couplers implement quadratic (pairwise) coupling between two ring oscillator blocks – each depicted by a ring of five inverters without loss of generality. To implement positive coupling ($J_{ij} > 0$), the transmission gate connects oscillator blocks at (inverter ring) nodes of the same polarity. Essentially the transmission gate locks the phases of this pair of oscillators to each other. To implement negative coupling ($J_{ij} < 0$), the transmission gate connects oscillator blocks at (inverter ring) nodes of the opposite polarity. This basic principle can be generalized to support a specific range and precision for J_{ij} values [7]. The idea is correlating interaction strength J_{ij} between two oscillators i and j with the phase delay between the respective oscillators by letting transmission gates tap different nodes across the rings.

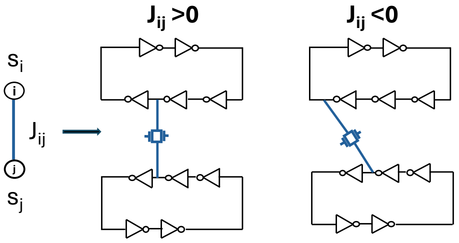


Fig. 2. Transmission gate based state-of-the-art OIM coupler implementing quadratic (pairwise) coupling between two ring oscillator blocks (each depicted by a ring of 5 inverters).

How to augment OIM couplers with support for higher-order interactions such as cubic? Figure 3 (Figure 4) provides the functional specification for a generic cubic coupler considering CUBO (CUSO) formulation – along with functional specifications for generic quadratic and diagonal couplers for reference. For each coupler, the functional specification tabulates conditions for spins or variables to get coupled or decoupled as the system is evolving – ideally towards a global solution. The coupler logic is in charge of resolving this operation.

Diagonal couplers establish coupling with a reference binary variable (Figure 3) or spin (Figure 4) – which takes the form of a reference phase in OIM – typically assuming a positive coupling coefficient and a reference state of +1. Accordingly, the system favors states that correspond to 0 (Figure 3) or −1 (Figure 4) aligning the variables or spins with the expected minimum-energy configurations.

In the absence of a reference, a challenge arises for OIM quadratic couplers when configurations requiring different coupling actions cannot be distinguished from each other. For the quadratic coupler from Figure 3, for example, this is the case for configurations $x_ix_j = 00$ and $x_ix_j = 11$ which

[1] The Hamiltonians are related by $H_B = (H_S - 7)/8$.

indicate that the phases of the respective coupled oscillators are identical. However, while $x_i x_j = 00$ does not require any action, $x_i x_j = 11$ calls for decoupling the respective oscillator phases. These configurations correspond to fully synchronous states and produce indistinguishable oscillatory patterns. Without a reference oscillator, the coupler cannot identify which state the system is in (i.e., which $x_i x_j$ configuration applies) and take the corresponding action accordingly. Addressing this requires additional logic to facilitate comparison of the instantaneous state to the state of a reference oscillator. For example, the decoupling condition for $x_i x_j = 11$ can be expressed as (x_i XNOR reference) AND (x_j XNOR reference). This discrepancy does not apply for the quadratic coupler from Figure 4, because the coupling actions for $s_i s_j = -1-1$ and $s_i s_j = 11$ are identical.

Diagonal Coupler

x_i	Action
0	None
1	Decouple x_i

Quadratic Coupler

x_i	x_j	Action
0	0	None
0	1	None
1	0	None
1	1	Decouple x_i, x_j

Cubic Coupler

x_i	x_j	x_k	Action
0	0	0	None
0	0	1	None
0	1	0	None
0	1	1	None
1	0	0	None
1	0	1	None
1	1	0	None
1	1	1	Decouple x_i,x_j,x_k

Fig. 3. Functional specification for Diagonal, Quadratic, and Cubic Couplers (assuming CUBO formulation).

Diagonal Coupler

s_i	Action
-1	None
+1	Decouple s_i

Quadratic Coupler

s_i	s_j	Action
-1	-1	Decouple s_i, s_j
-1	+1	None
+1	-1	None
+1	+1	Decouple s_i, s_j

Cubic Coupler

s_i	s_j	s_k	Action
-1	-1	-1	None
-1	-1	+1	Decouple s_i,s_j,s_k
-1	+1	-1	Decouple s_i,s_j,s_k
-1	+1	+1	None
+1	-1	-1	Decouple s_i,s_j,s_k
+1	-1	+1	None
+1	+1	-1	None
+1	+1	+1	Decouple s_i,s_j,s_k

Fig. 4. Functional specification for Diagonal, Quadratic, and Cubic Couplers (assuming CUSO formulation).

The cubic coupler from Figure 3 and Figure 4 face a similar ambiguity. As an example, Figure 5 reveals how spin configurations $s_i s_j s_k = -1+1+1$ and $s_i s_j s_k = +1-1-1$ from Figure 4 produce indistinguishable oscillatory patterns, making it impossible for the cubic coupler to determine the correct action to take: 011, 100, 011, ... applies for $s_i s_j s_k = -1+1+1$; and 100, 011, 100, ... for $s_i s_j s_k = +1-1-1$. This occurs because the Hamiltonian terms are symmetric under spin inversion. Comparison of the instantaneous state to the state of a reference oscillator can resolve the ambiguity. Distributing the reference phase to each coupler across the densely packed oscillator network, however, requires increasingly complex circuitry at scale.

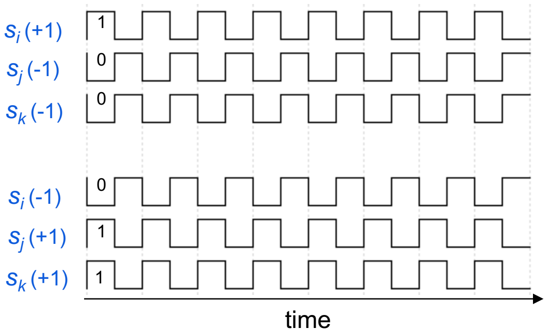
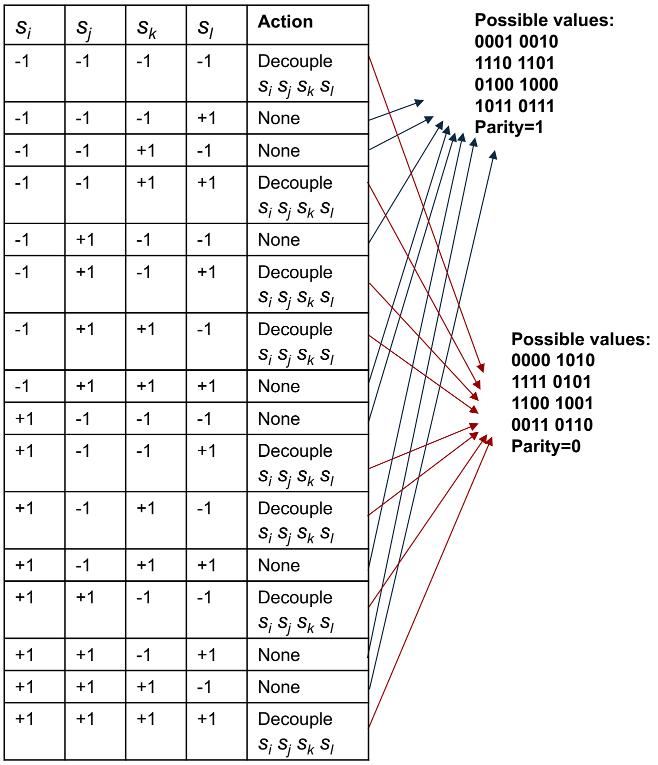


Fig. 5. Spin configurations $s_i s_j s_k = -1+1+1$ and $s_i s_j s_k = +1-1-1$ from Figure 4 produce indistinguishable oscillatory patterns, making it impossible for the cubic coupler to determine the correct action to take.

Quartic Coupler



s_i	s_j	s_k	s_l	Action
-1	-1	-1	-1	Decouple s_i s_j s_k s_l
-1	-1	-1	+1	None
-1	-1	+1	-1	None
-1	-1	+1	+1	Decouple s_i s_j s_k s_l
-1	+1	-1	-1	None
-1	+1	-1	+1	Decouple s_i s_j s_k s_l
-1	+1	+1	-1	Decouple s_i s_j s_k s_l
-1	+1	+1	+1	None
+1	-1	-1	-1	None
+1	-1	-1	+1	Decouple s_i s_j s_k s_l
+1	-1	+1	-1	Decouple s_i s_j s_k s_l
+1	-1	+1	+1	None
+1	+1	-1	-1	Decouple s_i s_j s_k s_l
+1	+1	-1	+1	None
+1	+1	+1	-1	None
+1	+1	+1	+1	Decouple s_i s_j s_k s_l

Fig. 6. Functional specification for the proposed Quartic coupler, which eliminates any ambiguity in oscillatory patterns that necessitates a resolution with the reference.

Our solution to support higher order interactions without compromising hardware complexity is the quartic coupler from Figure 6. The quartic coupler by definition can represent (up-to) fourth-order interactions. Most importantly, the quartic coupler effectively eliminates any ambiguity in oscillatory patterns that necessitates a resolution with the reference. Specifically, the quartic coupler can distinguish configurations that require different coupling actions from each other by solely checking the parity of the configuration. As shown in Figure 6, each of the two possible parity values is uniquely associated with a specific coupling action. There is no need

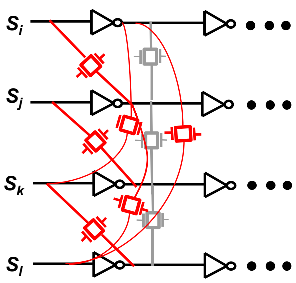


Fig. 7. A transmission gate based implementation of the quartic coupler, assuming an all-to-all connected OIM without loss of generality. Parity resolution logic enables transmission gates according to the functional specification in Figure 6.

for a reference for this operation. The quartic coupler only needs to decouple spins when their combined parity is 0, and it does not take any action when their combined parity is 1. The quartic coupler can thereby enable practical Higher Order Oscillatory Ising Machines (HOIM) which can seamlessly map COPs featuring higher order interactions. *We should also note that the core principle behind our quartic coupler design naturally extends to any even number of interactions beyond four.*

At the core of a transmission gate based implementation of the quartic coupler is the parity resolution logic. Figure 7 depicts an example implementation, assuming an all-to-all connected OIM without loss of generality. Parity resolution logic (which boils down to four input XOR operations) would enable transmission gates according to the functional specification in Figure 7. Such transmission gates effectively mimic programmable interaction coefficients. The downside is that the parity resolution logic may increase the control delay for the transmission gates, which careful design optimizations can compensate for.

A quartic coupler can realize all lower order couplings than fourth-order by using *dummy* spins. Let us revert to an SAT instance as an illustrative example. The idea is using an extra (termed dummy) spin to encode a lower-order clause to a higher-order one. Consider the 3-SAT clause $(x_1 + x_2 + x_3)$. The objective is to transform this into a 4-SAT clause by introducing a dummy spin d in a way that preserves the original satisfiability of the clause. This can be achieved by expressing the clause as the product of two terms: $(x_1 + x_2 + x_3 + d)$ $(x_1 + x_2 + x_3 + \neg d)$, ensuring that the dummy spin does not influence the logical outcome of the clause. The transformation is valid because, regardless of the truth value assigned to the dummy spin d, at least one of the two resulting clauses preserves the satisfiability of the original 3-SAT clause $(x_1 + x_2 + x_3)$. In this case the total number of spins required becomes $n + 1$ where n is the total number of (original) problem variables. Similarly, second order problems would require 2 dummy spins.

Any hardware solver has a limited number of spins, and OIMs do not represent an exception. At the same time the number of spins cannot increase indefinitely due to fundamental physical limits. However, COP sizes of practical importance keep growing. The ancillary variables introduced in the process of converting higher-order problem interactions to lower-order (pairwise) machine interactions exacerbate this by shrinking the effective number of available hardware spins. Large problems that cannot fit into a given Ising machine must be *decomposed*, i.e., broken down into smaller subproblems that match the available spin count in hardware. The complication comes from such subproblems not being independent. As a result, decomposition can degrade solution accuracy and/or increase the time to reach a solution. Using SAT as an example, the subproblems are solved iteratively until either all clauses are satisfied (reaching the *All-SAT* condition) or when a predefined iteration limit is reached. By construction, HOIM, as enabled by higher-order couplers detailed in Figure 6, reduces the pressure on decomposition and its adverse impact on solver performance.

Typical OIMs are not just limited by the number of spins. The range and precision for interaction coefficients are also limited. The process of converting higher-order problem interactions to lower-order (pairwise) machine interactions boils down to another layer of problem translation and typically affects the range as well the precision and may result in coefficients smaller or larger than the underlying machine can support. This may necessitate excessive scaling (and rounding) of coefficient values, which in turn may degrade the solution accuracy significantly. By simplifying problem translation, HOIM can reduce the likelihood of excessive scaling (and rounding), as well.

IV. Evaluation Setup

Simulation Framework: To evaluate the effectiveness of HOIM as enabled by higher-order couplers, we simulate OIM by a QUBO/HUBO solver based on the the Dual Annealing optimization algorithm [32]. This stochastic method integrates the general principles of Classical Simulated Annealing (CSA) and Fast Simulated Annealing (FSA), enhanced with a local search strategy applied to accepted candidate solutions. For our experiments, 3-SAT instances were formulated using the Chancellor formulation [27] to construct the corresponding QUBO models, while the HUBO model was derived from H_B as outlined in Section III-B.

Our simulator emulates all-to-all (A2A) connected OIMs. Ising machines that support A2A connectivity [7], [33] have a physical link between any pair of spins and can directly map arbitrary J_{ij}. In contrast, an Ising machine with nearest neighbor connectivity [34], [35] lacks this ability. We adjust each mapped Hamiltonian to conform to the constraints of the simulated OIM – particularly the requirement that interaction coefficients be restricted within a specific range – which we model after the A2A connected OIM from [7] as a recent representative design. Based on the specifications in [7], each iteration on the reference OIM takes approximately 171.9 μs and consumes 10mW of power. The simulations operate without the non-idealities and noise inherent in physical systems, thus represent an upper bound on achievable performance.

To demonstrate the advantages of the quartic coupler (Figure 6), we solve SAT problems using a simulated OIM with a quartic (fourth-order) coupler. We consider 4-SAT as well 3-SAT problems. For 3-SAT, we use dummy spins as explained in Section III-B to effectively map 3-SAT clauses onto 4-SAT clauses. Dummy spins do not alter the logical structure of the original 3-SAT clauses.

Benchmarks: To provide a comparative analysis, we use benchmark instances from the SATLIB uf20-91 suite [28]. All benchmarks are satisfiable and each includes 20 variables/91 clauses. The clauses-to-variables ratio represents problems that lie close to the phase transition region [36] where the hardest SAT problems reside. There are 1000 instances in total under this benchmark. We select 10 problem instances to have a closer look at the performance differences for each problem. We also use randomly generated uniform 4-SAT instances of variable size 20 with 200 clauses for a side by side comparison with 3-SAT. These problems are at the phase transition region, as well, where each problem is satisfiable and each possible literal is selected with the same probability of $1/2n$ (where n is the number of problem variables). We also consider larger SATLIB benchmarks as well as randomly generated uniform 4-SAT 50 variable problems with 500 clauses to assess scalability.

V. Evaluation

In this section, we quantify the performance of Higher Order Oscillatory Ising Machines (HOIM) as enabled by our higher-order couplers (such as the quartic coupler from Figure 6). We start our analysis with the comparison of average iteration count required to reach All-SAT between HOIM and the state-of-the-art OIM which can only support pairwise interactions (POIM) across 10 problem instances from SATLIB 3-SAT uf20-91 suite in Figure 8.

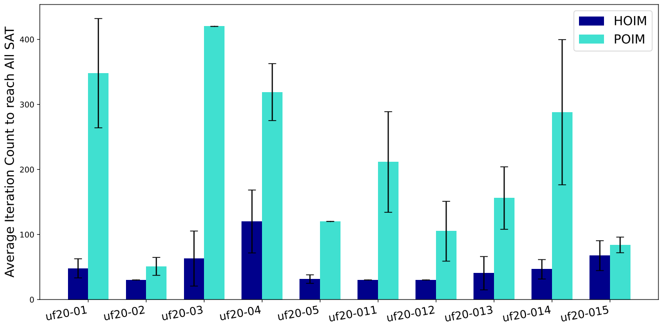


Fig. 8. Comparison of average number of iterations required to reach All SAT across different problem instances from uf20-91.

We observe that HOIM consistently outperforms POIM by requiring significantly fewer iterations in most cases. For example, for uf20-01 and uf20-03, POIM requires approximately 350–420 iterations, while HOIM only needs about 50–70 and the only notable exception is uf20-015. Overall, HOIM requires about 4.3× fewer iterations on average compared to POIM. Additionally, HOIM shows lower variance, as indicated by its smaller error bars, suggesting more stable performance.

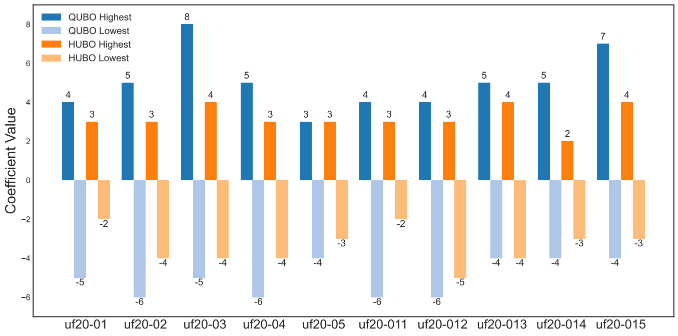


Fig. 9. Coefficient ranges across different problem instances from uf20-91.

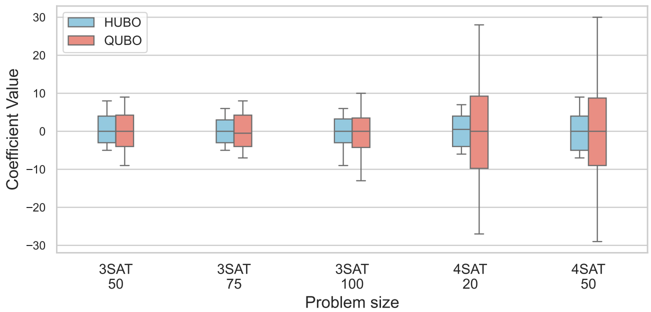


Fig. 10. Coefficient ranges across different problem sizes.

Next we perform a comparison of interaction coefficient ranges considering 20-variable 3-SAT problems in Figure 9, and larger (50 to 100 variable) 3-SAT as well 4-SAT problems in Figure 10. We observe that the QUBO formulation using Chancellor [27] exhibits a broader coefficient range compared to HUBO for 20 variable problems. For example, QUBO reaches highs of 8, 7, and 5, while HUBO's highest values are mostly around 4 or 3. As problem size increases, the same trends persists. Even for denser problems like 4-SAT where clause to variable ratio is 2× greater than 3-SAT, QUBO range is significantly higher, with coefficients stretching from about –30 to 30 for 50-variable problems, compared to HUBO's more restrained range of –10 to 9. Even in smaller 20-variable 4-SAT instances, QUBO coefficients span –30 to 28, while HUBO coefficients remain between –8 and 10. We conclude that QUBO – with persistently and significantly larger coefficient ranges – is more likely to hit hardware limits for interaction coefficient range and precision than HUBO.

Figure 11 provides a direct performance comparison of HOIM to POIM in terms of Time to Solution (TTS) and Energy to Solution (ETS). We observe that enabling higher-order coupling (HOIM) reduces TTS and ETS by up-to 4× in 3-SAT problems. For 4-SAT, the reduction in TTS and ETS reaches by up-to 7.89×.

Furthermore, due to hardware constraints [7], even a 20-variable 3-SAT problem cannot be directly mapped to the

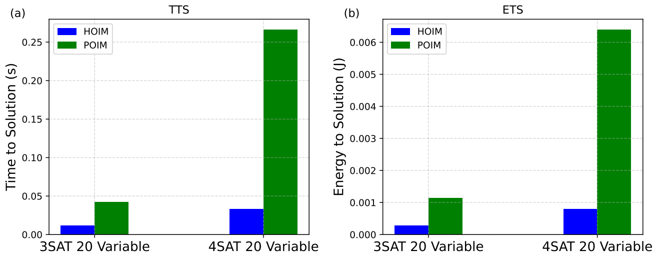


Fig. 11. (a) Time-to-solution (TTS), (b) Energy-to-solution (ETS) for 3-SAT and 4-SAT problems.

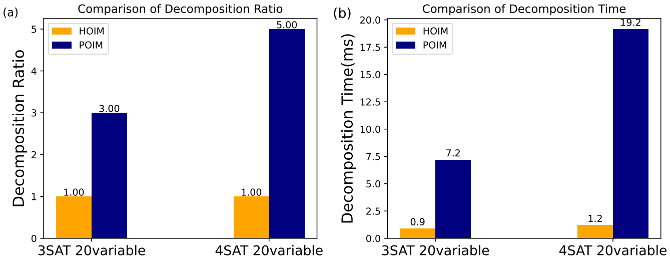


Fig. 12. Decomposition (a) ratio and (b) time for 3-SAT and 4-SAT.

OIM chip after translation to the QUBO format, necessitating decomposition into smaller sub-problems. In contrast, formulating the problem using higher-order interactions (HUBO) avoids the introduction of ancillary variables, substantially reducing the need for decomposition by 3× to 5×, as captured by the decomposition ratio in Figure 12 (a). The decomposition ratio quantifies how many more times a QUBO formulation must be decomposed compared to the corresponding HUBO. For example, if a HUBO problem can be solved on hardware as a single 20-variable problem without decomposition, while the equivalent QUBO problem requires splitting into a minimum three subproblems, the decomposition ratio becomes 3×. The corresponding reduction in decomposition time by 8× to 16× from Figure 12 (b) further highlights a key practical advantage of supporting higher-order interactions natively in Ising hardware.

VI. Related Work

In recent years, optimization using higher-order Ising machines has gained attention for solving complex combinatorial problems. The work presented in [37] investigates the use of Hopf oscillators and their amplitude dynamics, specifically mapping the SAT problem into a constraint satisfaction framework grounded in energy states. Here, computing with higher-order interactions requires a state variable to form and accumulate the partial derivatives of all terms in the total energy it participates in. The accumulation of partial derivatives can be a communication bottleneck eventually, and unlike our approach, they do not provide comprehensive hardware compatible solutions for their implementation.

Another article, [38] explores the use of higher-order spin interactions in OIMs for solving combinatorial optimization problems such as NAE-k-SAT and max-k-cut. The authors formulate a dynamical system and the corresponding energy function based on phase dynamics, enabling the encoding of complex optimization problems. However, this implementation has a potential challenge with the system's tendency to become trapped in local minima. As the size of the problem graph increases, the role of these local minima in the high-dimensional phase space becomes more significant, posing potential limitations on the system's ability to efficiently solve larger-scale problems. Our implementation avoids this problem by using programmable couplers instead of altering the phase dynamics.

In this line of research, [39] investigated the enhancement of cubic interactions in the Bistable Resistively-Coupled Ising Machine (BRIM) through additional architectural support, [40] implemented third-order interactions in a large-scale FPGA-based p-computer using hypergraph coloring to achieve parallelism, [41] proposed a reconfigurable higher-order Ising machine that utilizes SRAM to store spin interaction coefficients and employs a multiply-and-accumulate (MAC) unit for spin operations. However, none of these solutions cater to OIMs as opposed to ours and the operation cost is higher as these implementations rely on multi-cycle spin operations which is not feasible for larger problems.

VII. Conclusion

In this paper, we propose a practical higher-order coupler to enable Higher Order Oscillatory Ising Machines (HOIM) which can directly support interactions between an arbitrary number of problem variables. We quantitatively compare the performance to state-of-the-art Oscillatory Ising Machines (OIM) which can only support pairwise interactions, using representative combinatorial optimization problems. We demonstrate how enabling higher-order coupling can reduce time- and energy-to-solution significantly. HOIM, by construction, lifts the pressure on problem decomposition and reduces decomposition-induced adverse effects on solver performance. At the same time, by simplifying problem translation, HOIM can reduce the likelihood of excessive scaling (and rounding) of interaction coefficients, which can also degrade solver performance significantly.

The key contribution of our paper is a practical higher-order coupler design, which can enable seamless problem translation for any even number of variable interaction, by eliminating the reliance on external reference spins and by using a simple, parity-based logic. Importantly, our coupler's ability to model all lower-order interactions further enhances its versatility. The core principle behind our quartic coupler design naturally extends to any even number of interactions beyond four. This adaptability makes the proposed design well-suited for future implementations of higher-order oscillatory Ising machines in complex and resource-constrained environments.

Finally, apart from k-SAT problems, there exist other classes of COPs – such as Max-k-Cut and hypergraph coloring – that incorporate higher-order problem variable interactions. As the core principle behind our coupler design is not limited to any specific problem mapping or Ising machine, it can also support these problems in the context of a generalized COP solver.

References

[1] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[2] S. Izawa, K. Kitai, S. Tanaka, R. Tamura, and K. Tsuda, "Continuous black-box optimization with an ising machine and random subspace coding," *Phys. Rev. Res.*, vol. 4, p. 023062, Apr 2022.

[3] J. Laydevant, D. Marković, and J. Grollier, "Training an ising machine with equilibrium propagation," *Nature Communications*, vol. 15, p. 3671, 2024.

[4] W. R. S. Guimaraes and L. S. Lima, "Self-organizing three-dimensional ising model of financial markets," *Phys. Rev. E*, vol. 103, p. 062130, Jun 2021.

[5] T. Wang, L. Wu, P. Nobel, *et al.*, "Solving combinatorial optimisation problems using oscillator based ising machines," *Natural Computing*, vol. 20, pp. 287–306, 2021.

[6] J. Chou, S. Bramhavar, S. Ghosh, *et al.*, "Analog coupled oscillator based weighted ising machine," *Scientific Reports*, vol. 9, p. 14786, 2019.

[7] H. Lo, W. Moy, H. Yu, *et al.*, "An ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture," *Nature Electronics*, vol. 6, pp. 771–778, 2023.

[8] Y. Fu and P. W. Anderson, "Application of statistical mechanics to NP-complete problems in combinatorial optimisation," *Journal of Physics A Mathematical General*, vol. 19, pp. 1605–1620, June 1986.

[9] F. Barahona, "On the computational complexity of ising spin glass models," *Journal of Physics A: Mathematical and General*, vol. 15, p. 3241, oct 1982.

[10] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem," *Science*, vol. 292, p. 472–475, Apr. 2001.

[11] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik*, vol. 31, pp. 253–258, 1925.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[13] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "2.6 a 2× 30k-spin multichip scalable annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," 2019.

[14] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, and M. Motomura, "Statica: A 512-spin 0.25 m-weight annealing processor with an all-spin-updates-at-once architecture for combinatorial optimization with complete spin–spin interactions," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 165–178, 2020.

[15] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, "Physics-inspired optimization for quadratic unconstrained problems using a digital annealer," *Frontiers in Physics*, vol. 7, p. 48, 2019.

[16] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, "High-performance combinatorial optimization based on classical mechanics," *Science Advances*, vol. 7, no. 6, p. eabe7953, 2021.

[17] K. Tatsumura, M. Yamasaki, and H. Goto, "Scaling out ising machines using a multi-chip architecture for simulated bifurcation," *Nature Electronics*, vol. 4, no. 3, pp. 208–217, 2021.

[18] M. Johnson, M. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. Berkley, J. Johansson, P. Bunyk, E. Chapple, C. Enderud, J. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. Thom, E. Tolkacheva, C. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, "Quantum annealing with manufactured spins," *Nature*, vol. 473, May 2011.

[19] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, *et al.*, "Quantum optimization of maximum independent set using rydberg atom arrays," *Science*, vol. 376, no. 6598, pp. 1209–1215, 2022.

[20] S. Dutta, A. Khanna, A. Assoa, H. Paik, D. G. Schlom, Z. Toroczkai, A. Raychowdhury, and S. Datta, "An ising hamiltonian solver based on coupled stochastic phase-transition nano-oscillators," *Nature Electronics*, vol. 4, no. 7, pp. 502–512, 2021.

[21] W. Moy, I. Ahmed, P.-w. Chiu, J. Moy, S. S. Sapatnekar, and C. H. Kim, "A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving," *Nature Electronics*, vol. 5, no. 5, pp. 310–317, 2022.

[22] H. Lo, W. Moy, H. Yu, S. Sapatnekar, and C. H. Kim, "An ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture," *Nature Electronics*, vol. 6, pp. 771–778, Oct 2023.

[23] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, *et al.*, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, no. 6312, pp. 603–606, 2016.

[24] Z. Bian, F. A. Chudak, W. G. Macready, and G. Rose, "The Ising model: teaching an old problem new tricks," 2010. DWave Systems.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 4th ed., 2022.

[26] R. M. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958–2008* (M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, eds.), pp. 219–241, Berlin, Heidelberg: Springer, 2010.

[27] N. Chancellor, S. Zohren, P. Warburton, *et al.*, "A direct mapping of max k-sat and high order parity checks to a chimera graph," *Scientific Reports*, vol. 6, p. 37107, 2016.

[28] H. H. Hoos and T. Stützle, "Satlib benchmark problems." http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html, 2000. Accessed: 2025-04-11.

[29] H. Cılasun, Z. Zeng, R. S, *et al.*, "3sat on an all-to-all-connected cmos ising solver chip," *Scientific Reports*, vol. 14, p. 10757, 2024.

[30] S. Nagies, K. T. Geier, J. Akram, D. Bantounas, M. Johanning, and P. Hauke, "Boosting quantum annealing performance through direct polynomial unconstrained binary optimization," 2025.

[31] M. Graber and K. Hofmann, "An integrated coupled oscillator network to solve optimization problems," *Communications Engineering*, vol. 3, p. 116, 2024.

[32] S. Developers, "scipy.optimize.dual_annealing – dual annealing optimization." https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.dual_annealing.html, 2024. Accessed: 2025-04-19.

[33] M. K. Bashar, A. Mallick, D. S. Truesdell, B. H. Calhoun, S. Joshi, and N. Shukla, "Experimental demonstration of a reconfigurable coupled oscillator platform to solve the max-cut problem," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 6, no. 2, pp. 116–121, 2020.

[34] I. Ahmed, P.-W. Chiu, W. Moy, and C. H. Kim, "A probabilistic compute fabric based on coupled ring oscillators for solving combinatorial optimization problems," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2870–2880, 2021.

[35] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "A 20k-spin ising chip to solve combinatorial optimization problems with cmos annealing," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 303–309, 2016.

[36] W. Zhang, "Phase transitions and backbones of 3-sat and maximum 3-sat," in *Principles and Practice of Constraint Programming — CP 2001* (T. Walsh, ed.), (Berlin, Heidelberg), pp. 153–167, Springer Berlin Heidelberg, 2001.

[37] C. Bybee, D. Kleyko, D. E. Nikonov, and K. Camsari, "Efficient optimization with higher-order ising machines," *Nature Communications*, vol. 14, p. 6033, 2023.

[38] M. Bashar and N. Shukla, "Designing ising machines with higher order spin interactions and their application in solving combinatorial optimization," *Scientific Reports*, vol. 13, p. 9558, 2023.

[39] A. Sharma, M. Burns, A. Hahn, *et al.*, "Augmenting an electronic ising machine to effectively solve boolean satisfiability," *Scientific Reports*, vol. 13, p. 22858, 2023.

[40] S. Nikhar, S. Kannan, N. A. Aadit, *et al.*, "All-to-all reconfigurability with sparse and higher-order ising machines," *Nature Communications*, vol. 15, p. 8977, 2024.

[41] Y. Su, T. T.-H. Kim, and B. Kim, "A reconfigurable cmos ising machine with three-body spin interactions for solving boolean satisfiability with direct mapping," *IEEE Solid-State Circuits Letters*, vol. 6, pp. 221–224, 2023.