High Quality Circuit-Based 3-SAT Mappings for Oscillator Ising Machines

Venkata Pavan Sumanth Sikhakollu[1(✉)], Shreesha Sreedhara[1], Rajit Manohar[2], Alan Mishchenko[1], and Jaijeet Roychowdhury[1]

[1] University of California Berkeley, Berkeley, CA 94720, USA
{pavan_sumanth,shreesha_sreedhara,alanmi,jr}@eecs.berkeley.edu
[2] Yale University, New Haven, CT 06520, USA
rajit.manohar@yale.edu

Abstract. 3-SAT is a class of NP-hard combinatorial optimization problems that Ising machines have had difficulty solving successfully. Solution success rate depends not only on the choice of Ising machine, but crucially, also on the mapping from 3-SAT to Ising form. We evaluate the performance of Oscillator Ising Machines (OIMs) on several existing 3-SAT-to-Ising mappings, finding that they yield mediocre or poor results. We propose two novel enhancements to logic-synthesis-based Ising mapping schemes that improve solution success rate significantly (from 0% to about 56% on SATLIB's `uf20` problem set). We then propose a new circuit- and clause-based 3-SAT-to-Ising mapping scheme that employs 3-input OR gates. Using this mapping increases OIM's success rate on `uf20` to 95.9%—we believe this is by far the best raw performance achieved on any 3-SAT problem class by any Ising machine scheme. We also present a comparison of OIM *vs.* simulated annealing on Ising-mapped 3-SAT problems, revealing that OIM's performance is significantly superior.

Keywords: Ising Machine · Oscillator Ising Machine · 3-SAT · Combinatorial Optimization · Ising Mapping

1 Introduction

In recent years, Ising machines (IMs) have emerged as promising hardware solvers for finding optimal or near-optimal solutions to combinatorial optimization (CO) problems [20]. Several IM schemes have been proposed, each leveraging specialized hardware for addressing CO problems. Notable examples include Quantum Annealers [3,14] that use qubits as spins, Coherent Ising Machines [10,11] that use modulated optical pulses to represent spins, and various other analog-based IMs [5,6]. Among these, Oscillator-based Ising Machines [25] stand out due to their cost-effectiveness, low energy consumption, and the ability to be fabricated on-chip using CMOS technologies. Importantly, they also produce high-quality results on various combinatorial problems [21,22,25].

D.-J. Cho and J. Kim (Eds.): UCNC 2024, LNCS 14776, pp. 269–285, 2024.
https://doi.org/10.1007/978-3-031-63742-1_19

In practice, CO problems require translation into Ising form to be implementable on Ising hardware. While some problem classes, such as MAX-CUT, have straightforward Ising mappings, mappings of many other problems are complex, requiring additional spins to represent the CO problem [17]. To fully leverage the advantages of analog Ising solvers, the CO problem class should feature an efficient Ising mapping that remains scalable even as problem size increases, while also adhering to hardware-related constraints such as sparsity and coupling resolution. As we demonstrate in this paper, the optimization performance of the IM is highly sensitive to the mapping used, underscoring the importance of selecting an appropriate mapping.

In this paper, we will analyze IMs for solving the *Boolean satisfiability (SAT)* problem. The SAT problem stands as one of the most fundamental and extensively studied problems in computer science. SAT has far-reaching implications in various fields, including formal verification, artificial intelligence, and optimization. In this study, we mainly focus on the translation of 3-SAT problems to Ising form and their solution by OIM, with comparisons against simulated annealing (SA). Applying our proposed techniques to the uf20 benchmark sourced from SATLIB [1], we achieve very good results, surpassing any other IM to the best of our knowledge, while consuming similar hardware and computational resources compared to other mappings.

The main contributions of this paper can be outlined as follows. First, we propose two techniques aimed at improving the performance of OIM on 3-SAT by leveraging logic-synthesis-based Ising mapping methods [13,23]. These enhancements result in a significant increase in the average success rate for uf20 problems, rising from 0% to 55.6%. Secondly, we present a simple yet novel circuit-based mapping approach utilizing 3-input OR Ising gates. As detailed in subsequent sections, this mapping achieves the highest average success rate of 95.9% for uf20 problems among various mapping schemes documented in the literature. Lastly, we conduct a comparative analysis between the performance of OIM and SA, demonstrating the superiority of OIM in solving 3-SAT instances. Note that all the results presented in the paper are based on simulations.

The remainder of the paper is organized as follows. Section 2.1 provides an overview of the Ising model. Section 2.2 outlines the 3-SAT problem. Existing methods for transforming a 3-SAT instance into an Ising network are discussed in Sect. 3. Our novel approaches that yield improved results are presented in Sect. 4. In Sect. 5, we discuss the proposed novel 3OR mapping technique. Finally, Sect. 6 contains a comparative analysis of our results with other mapping techniques.

2 Background

2.1 The Ising Model

The Ising model is a physics-oriented formalism that was first devised for explaining domain formation in ferro-magnets [12]. It comprises a group of discrete variables $\{s_i\}$, referred to as spins, each taking a binary value ± 1, such that an

associated "energy function" known as the Ising Hamiltonian is minimized. The Ising Hamiltonian is given by

$$H \triangleq -\sum_{1\le i<j\le n} J_{ij}\cdot s_i\cdot s_j - \sum_{i=1}^{n} h_i\cdot s_i, \tag{1}$$

where n is the number of spins; $\{J_{ij}\}$ and $\{h_i\}$ are real coefficients. To accommodate the second summation term (known as "magnetic field" or "Zeeman effect" terms) within the first summation term (quadratic in spins), an extra spin can be introduced with its value fixed at "+1". We term this additional spin the REF spin.

Finding the ground state[1] of the Ising model is a classic example of a combinatorial optimization problem. All of Karp's 21 NP-complete CO problems can be mapped to the Ising model by assigning appropriate values to the coefficients [15]. Physical systems that can directly minimize the Ising Hamiltonian are termed Ising machines [3,10,11,14].

Oscillator Ising Machines (OIMs): When an oscillator with a natural frequency f_0 experiences a small periodic external input at a frequency $f_1 \approx f_0$, the oscillator's response can lock on to the input in both frequency and phase [2]. This phenomenon is called injection locking, or more precisely, fundamental harmonic injection locking (FHIL). Sub-harmonic injection locking (SHIL) is a related phenomenon in which an oscillator is perturbed by a periodic input at approximately twice its natural frequency, say $2f_1$. This external signal, known as a synchronization signal (SYNC), causes the oscillator to lock to exactly half the frequency of SYNC, *i.e.*, f_1, and feature bistable phase locks separated by 180° [19]. Consequently, the oscillator functions as a logic latch capable of storing a phase-based binary bit.

It has been shown in [25] that a network of coupled oscillators under SHIL can function as an Ising machine. The dynamics of the network of coupled oscillators with SYNC inputs to each oscillator are described by the generalized Kuramoto equation [24]. However, in a simplified scenario where the oscillations are sinusoidal, and with other approximations, the dynamics can be modeled using the Kuramoto model [24], *i.e.*,

$$\frac{d}{dt}\phi_i(t) = -K_c\sum_j J_{ij}\cdot\sin(\phi_i(t)-\phi_j(t)) - K_s\cdot\sin(2\phi_i(t)) + K_n\cdot\xi_i(t). \tag{2}$$

In (2), $\{J_{ij}\}$ are the weights of oscillator couplings; the underlying combinatorial optimization problem is encoded in these weights. The operation of oscillator-based Ising machines modeled by (2) is controlled by several parameters: K_c, a scalar representing the coupling strength; K_s, a scalar modeling the coupling strength from SYNC; and K_n, a scalar representing the magnitude of noise $\xi_i(t)$, which is a Gaussian white noise with zero mean. While K_c, K_s, and K_n can all be

[1] The term "ground state" means a state that achieves the minimum Hamiltonian.

time-varying, resulting in various "annealing profiles", our experiments indicate that varying K_s significantly affects the performance of the OIM, whereas the others do not have much impact.

***K_s* Parameter Cycling:** A periodic square pulse is assumed for the K_s parameter, alternating between a positive high value and a negative low value over a certain time period. It is important to note that if K_s is static at high positive value, then the oscillator phases will settle to either 0 or π after a few oscillator cycles [25]. Essentially, this means that for each cycle of K_s, we have one set of settled phases. Due to dynamics and noise, the settled phases may not be the same in the next K_s cycle. Therefore, after initializing the OIM, for N K_s cycles, we obtain N samples. These samples are then utilized to evaluate whether the OIM has effectively solved the underlying Ising problem, *i.e.*, settled to the minimum Hamiltonian or possibly a nearby value.

2.2 The Satisfiability Problem: 3-SAT

Given a Boolean formula, the goal of the satisfiability problem is to determine whether there exists an assignment of truth values to variables that evaluates the given Boolean formula to "true". The SAT problem is a classically difficult NP-Hard CO problem [15] with a wide variety of practical applications [18], and it can be transformed into any other NP-complete problem through a polynomial-time transformation [15].

SAT problems are often expressed in Conjunctive Normal Form (CNF), where the Boolean formula is given as a conjunction of multiple clauses, with each clause representing a disjunction of k literals or their negations. This formulation is known as k-SAT, and in the case where k=3, it is specifically referred to as 3-SAT. A Boolean function $f()$ expressed in CNF form with 3 literals in each clause takes the form

$$\begin{aligned} f(x_1, x_2, \cdots, x_n) &= C_1 \wedge C_2 \wedge \cdots \wedge C_m, \\ &\text{with } C_i = x_p \vee x_q \vee x_r;\ p, q, r \in [1, n], 1 \leq i \leq m, \end{aligned} \tag{3}$$

where n is the number of input variables, m is the number of clauses, C_i represents the i^{th} clause, and x_p represents the p^{th} input variable. f is deemed satisfiable if there exists a set of input assignments from $\{0,1\}^n$ such that $f(x_1, x_2, \cdots, x_n)$ evaluates to 1. The 3-SAT problem can be formulated in Ising form using various formulations, as described in the Sect. 3.

3 SAT to Ising Transformation

In this section, we discuss several prior 3SAT-to-Ising formulations. SAT-to-Ising transformations can be broadly categorized into two types: circuit-based transformations and generic transformations. A circuit-based transformation directly maps the structure of the Boolean circuit representing the SAT problem onto an Ising model, preserving the logical relationships and dependencies inherent

to the circuit. Unlike generic transformations, which operate at the clause level, circuit-based transformations can map any SAT expression, not just those in CNF form. The MIS-based [8] and Chancellor [7] formulation can be considered generic mappings, while logic-synthesis-based mappings [23] are circuit-based.

3.1 MIS^{3m}-Based Formulation

The maximal independent set (MIS) formulation, also known as the Choi formulation [8], provides a translation of 3-SAT to Ising by assigning one Ising spin for each literal. For a 3-SAT instance with m clauses and n variables, the Choi formulation requires a total of $3m$ spins, excluding the REF spin. In this formulation, the literals within each clause are interconnected in a triangular configuration. Additionally, conflicting edges are established between two literals that represent negated versions of the same variable. The triangular structure aims to satisfy the SAT condition, while the conflicting edges penalize dissimilar assignments for the same variable.

It is worth noting that in the context of the OIM, if different values are assigned to spins copies representing the same variable, conventionally this would be considered illegitimate. However, to achieve more effective results, we relaxed this condition by adopting a majority vote approach among spins to assign the variable value [9].

3.2 The Chancellor^{n+m} Mapping

The Chancellor formulation [7] maps an n-variable m-clause instance using $n+m$ Ising spins, excluding the REF spin. The formulation establishes a 1-1 correspondence between the n SAT variables and the first n Ising spins and introduces one additional Ising spin for each of the m clauses.

3.3 Logic-Synthesis-Based Mapping

A detailed logic-synthesis-based SAT to Ising mapping is described in [13], which is inspired by [23]; a concise summary is provided in this section. Logic-synthesis-based transformation is circuit based and follows the flow shown in Fig. 1. First, the given 3-SAT problem in CNF is converted to a Boolean circuit using logic synthesis. Subsequently, the Boolean circuit is transformed into an Ising network by replacing each Boolean gate in the circuit with its Ising equivalent.

Logic synthesis is the process of converting a high-level (*e.g.*, CNF, RTL) description of a digital circuit into a lower-level representation composed of logic gates, such as AND, OR, and NOT gates, optimizing factors like performance, area, and power consumption. Widely available logic-synthesis tools (such as ABC [4]) utilize a user-specified gate library to synthesize a Boolean circuit for a given CNF. There is ample room for optimization at the synthesis stage, where, depending on the problem, one can control the number of gates, the gate depth, the fan-ins/outs of the gates, *etc.*, in the resulting Boolean network.

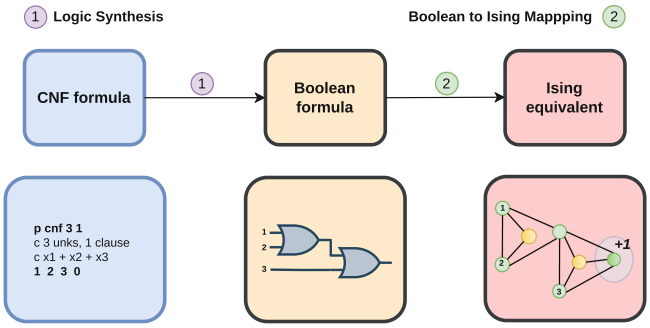


Fig. 1. Converting a CNF instance to an Ising instance.

These optimizations can be useful for hardware feasibility (*e.g.*, fan-ins/outs affect connection sparsity), but at the same time, can significantly impact the performance of the Ising machine.

After synthesis, each logic gate is converted into Ising form using its corresponding *Ising equivalent* network. In this network, the inputs and the output of each logic gate are represented by Ising spins. Coupling weights between the spins are carefully chosen to correspond to the specific type of logic gate in question (*e.g.*, see Fig. 2). The crucial property of these Ising equivalent gates is that if the gate's input-output spins satisfy the corresponding Boolean relationship, then the Hamiltonian of the Ising equivalent network reaches its global minimum. For any other spin combination, the Hamiltonian is strictly above the global minimum. Importantly, this property extends to arbitrary interconnections of these Ising equivalent gates. Thus, the Hamiltonian of the synthesized network achieves its global minimum if, and only if, the logical relationships of all the gates in the network are satisfied.

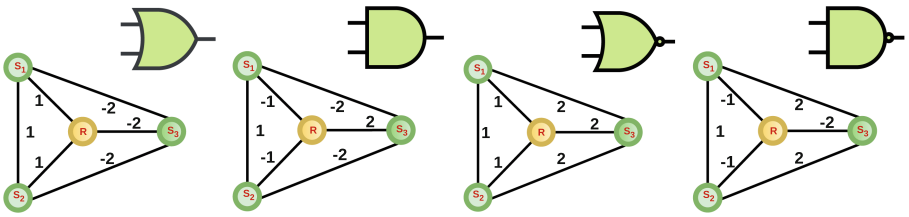


Fig. 2. Ising equivalent networks for 2-input Boolean gates; R is REF/+1 signal.

To enforce the SAT condition, *the output spin in the Ising version of the synthesized network is set to +1* (this is easy to implement in hardware). This constraint ensures that the network's minimum Hamiltonian solutions automat-

ically correspond to SAT solutions. Unlike many other combinatorial optimization problem mappings, for SAT, we already know what the minimum Hamiltonian is—since each individual gate's minimum Hamiltonian is known, and these are simply summed to find the global minimum. We use this information to determine whether the settled spin state is SAT or not. The Ising machine (IM) assumes the problem is SAT and attempts to find the corresponding set of inputs. If the Hamiltonian for the settled spins is not the same as the global minimum, then the IM deems the problem as UNSAT. Therefore, **if the IM indicates that the problem is SAT, then it is indeed SAT. However, if it indicates that the problem is UNSAT, then it may or may not be UNSAT.**

OIM with ABC Mapping: ABC is a widely used open-source tool for logic synthesis and formal verification of digital circuits. It offers a suite of algorithms and techniques for optimizing and verifying designs, making it a valuable resource for hardware designers. ABC can directly process CNF files and synthesize Boolean circuits based on user preferences, such as controlling the depth of the Boolean circuit or limiting fan-in/out. Since we have utilized ABC for logic-synthesis-based mapping, we refer to this mapping as the **ABC mapping**.

The performance of OIM with ABC-mapped 3-SAT problems is good for very small-scale SAT problems. However, it encounters notable challenges with larger instances, such as DIMACS uf-20 problems (see Sect. 6). To enhance performance, the present work introduces two novel ideas, discussed in the following sections.

Success Rate: A metric termed the *success rate* is important for assessing OIM performance on SAT problems. To calculate the success rate of OIM on a problem, we execute OIM on the problem many times with random initial conditions and the fraction of runs resulting in success determines the success rate. For a given Ising-mapped SAT instance, we begin the OIM simulation with random initial conditions for oscillator phases. Following the procedure outlined in Sect. 2.1, we cycle the K_s parameter. Throughout our simulations, we maintain a standard of 100 cycles of K_s, ensuring consistency across all reported experiments in this paper. This yields 100 sets of settled oscillator phase samples per simulation. A simulation is considered successful if at least one of these samples achieves the global minimum Hamiltonian or, equivalently, satisfies the SAT condition.

4 Level-Based Gate Scaling (LGS)

As outlined in the previous section, the objective of an Ising machine is to minimize the system's "energy", or more precisely the Hamiltonian, as defined in (1). In the context of SAT, the problem is deemed satisfiable only if each Ising

gate within the Ising equivalent network maintains a logically consistent input-output pairing, which ensures that the Ising Hamiltonian of the Ising equivalent network is the minimum possible.

In cases where the problem remains unsolved, indicating logically inconsistent gates in the overall Ising network, we argue that it is preferable to have these logically inconsistent gates closer to the inputs than to the output. To understand why, consider two scenarios: one where the inconsistent gate is close to the output and another where it is close to the primary inputs. If the gate is closer to the inputs, then the number of primary inputs that affect the input-set of the inconsistent gate is relatively small compared to the other case, *i.e.*, when the gate is far from the inputs. For example, in a "balanced" circuit with 2-input gates, the number of primary inputs affecting the output of a gate that is at a distance of m from the primary inputs is 2^m; so, the smaller the m, the smaller the number of critical primary inputs. Therefore, the Hamming distance (number of positions at which corresponding bits are different) from the actual SAT solution can be expected to be less when the erring gate is closer to the inputs. If the solution given by the OIM is close to the actual solution (smaller Hamming distance), then we can potentially improve the OIM result by using local search algorithms like single bit flip search. Hence, inconsistent gates closer to the primary inputs lead to better solutions. However, *the Hamiltonian achieved by the OIM is the same regardless of where in the circuit the inconsistent gate is located.*

To address this problem, we propose *scaling the Hamiltonian contribution of gates* in the network based on their "proximity" from the primary inputs, or alternatively, their proximity to the output. Higher priorities are assigned to gates closer to the output. While traditionally, the digital design literature measures levels from the primary inputs, we adopt a perspective where levels are measured from the output. The gate attached to the global output is assigned level 0. A gate is assigned level n if its output net serves as an input to a gate of level $(n-1)$. In cases where a gate is connected to multiple gates, priority is given to the one closest to the output. For example, in Fig. 3, gate G_1's output net is connected to a *level-1* gate, hence G_1 is considered a *level-2* gate. Gate G_2's output net is connected to gates at *level-2* and *level-3*; therefore, G_2 is considered a *level-3* gate because *level-2* is closer to the output and receives priority.

We use the term *sizing*[2] to refer to scaling all the edge weights within the Ising equivalent network of a particular gate. In Fig. 4b, a size-K OR gate is depicted. Scaling an Ising equivalent gate by a factor of K implies that the Hamiltonian contribution of that Ising gate is also scaled by K. For example, the minimum Hamiltonian for an unscaled 2-input OR gate is *-3*; if scaled by K=3, the minimum Hamiltonian become -9. As discussed in Sect. 2.1, the dynamics of OIM tend to minimize the Hamiltonian of the network. Therefore, OIM will tend to solve the scaled gates correctly, even at the expense of the unscaled ones. This stems from the fact that resolution of scaled gates yields a more significant reduc-

[2] Note that sizing in this context does not mean physical scaling.

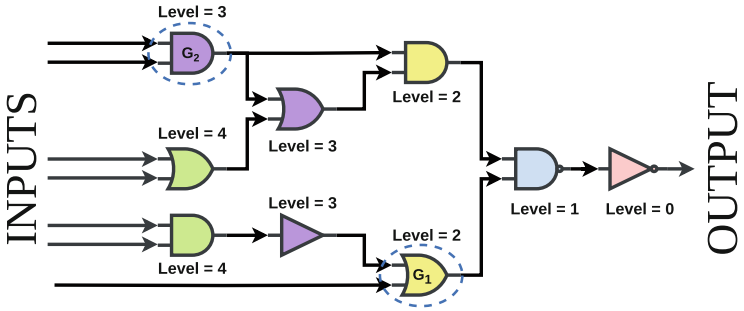
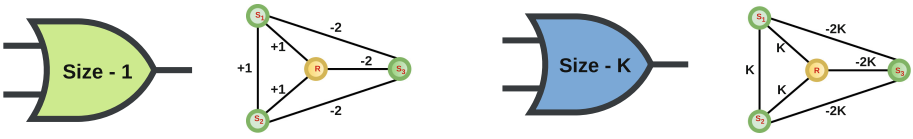


Fig. 3. An example network depicting the concept of gate level.

tion in the Hamiltonian, compared to unscaled/weakly-scaled ones. Therefore, **by applying a scaling function that decreases monotonically with the gate's level**, we encourage OIM to solve gates at the n^{th} level, even if it necessitates not satisfying those at $(n+1)^{th}$ level. As a result, this mechanism facilitates the migration of logically inconsistent gates towards inputs, since gates closer to the inputs have smaller scales, thus leading to solutions closer to SAT in terms of Hamming distance.



(a) Unit sized (b) K-sized

Fig. 4. Ising representation of 2-input OR gates; R represents the REF/+1 signal.

4.1 Iterative Gate Scaling (IGS)

The integration of OIM with ABC mapping using the level-based gate scaling (LGS) above suggests an iterative approach for solving SAT problems. As discussed previously, simply scaling a gate to be larger relative to the others effectively drives the scaled gate towards the correct input-output state. If we can identify the gates that are prone to failure, we can up-size these gates to potentially improve the success rate (defined in Sect. 3.3) of OIM on a particular problem. To find gates prone to failure, we simply use OIM and detect which gates are not logically consistent.

We start with an unscaled Ising network and run OIM on this network to identify the failed gates. Then, we up-size these gates and rerun OIM on the

modified network. We repeat this process iteratively for a few iterations, expecting an improvement in the success rate. Pseudo-code for IGS is provided in Algorithm 1.

Algorithm 1. Pseudocode for Iterative Gate Scaling (IGS)

```
iter ← 1, N_iter ← 100                                   ▷ Iteration count
parallel_runs(N_S,est) ← 100          ▷ #runs for estimating the success rate
failed_gates ← {}                                        ▷ list of failed gates
current_Ising_network ← unscaled_network
while iter ≤ N_iter do
    run OIM on current_Ising_network N_S,est times
    failed_gates ← {collection of all failed gates in this iteration across runs}
    for gate in failed_gates do
        gate.scale ← gate.scale + K · (1 - S) · P_f, gate
                                   ▷ S : success rate; P_f : failure rate
    end for
    current_Ising_network ← network_with_updated_gate_scale
end while
```

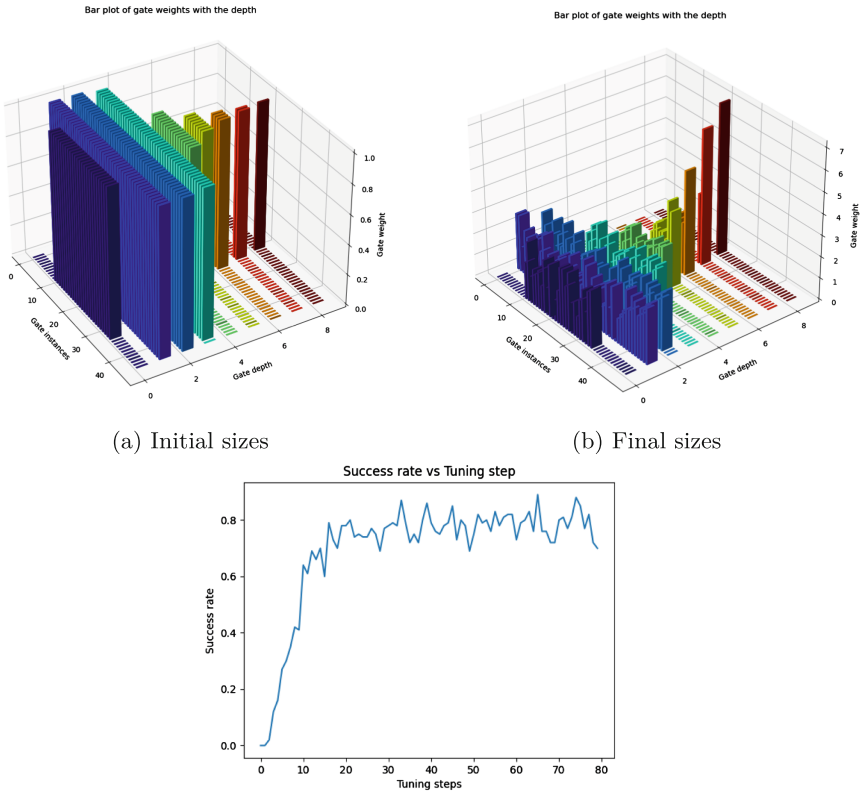
Gate-Scale Update Step: The update step in Algorithm 1 involves two quantities: the success rate of the network (S) and the failure rate of the gate ($P_{f,\text{gate}}$). The failure rate of a gate G is equal to the fraction of runs (used to estimate the success rate) in which the gate G is faulty. We will increase the scale of the gate if the failure rate is high. However, if the success rate is already high, there is no point in scaling the gate further. Therefore, the update quantity is proportional to $(1 - S) \cdot P_{f,\text{gate}}$.

We tested IGS on the *uf20-02* 3-SAT problem; the results are shown in Fig. 5. Initially, in Fig. 5a, all gates in the ABC-mapped network are of unit size. After 80 iterations, as depicted in Fig. 5b, we observe that the sizes of gates closer to the output are larger, which is consistent with the discussion in Sect. 4. Note that, in Fig. 5c, the success rate improves and eventually saturates at close to 90% with iterations.

While IGS appears valuable for increasing the success rate, note that one needs to run the problem several times, updating the weights in between. From a hardware implementation standpoint, the time and overall energy consumption for solving the problem scale with the product of the number of iterations (N_{iter}) and the number of times the problem is executed to estimate success rate ($N_{S,\text{est}}$). Nevertheless, this approach can be a valuable addition to the repertoire of mappings to solve SAT problems.

5 3OR-Based Ising Mapping

In this section, we present $\boldsymbol{3OR}^{n+m}$, a 3SAT-to-Ising transformation technique that requires $n+m$ spins, excluding the REF spin. The number of spins required



(a) Initial sizes

(b) Final sizes

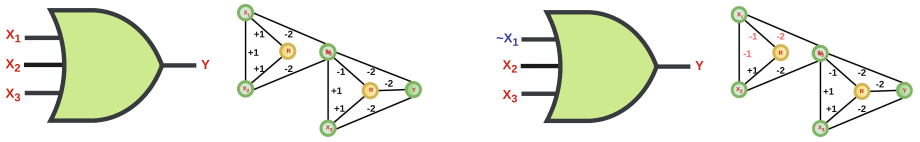
(c) Success rate *vs.* iteration

Fig. 5. IGS: evolution of gate sizes and success rate for uf-02 problem.

is the same as that for Chancellor^{n+m} mapping. However, the new 3OR mapping yields superior results with OIM (see Sect. 6). For each Boolean variable x_i, $0 \leq i \leq n-1$, occurring in the 3-SAT instance, we use one spin to encode its Boolean value. Additionally, we need one more spin for every clause in the 3-SAT instance.

Typically, 3-SAT problems are provided in Conjunctive Normal Form (CNF), where each clause consists of the OR of three literals. These clauses collectively feed into an m-input AND gate, where m represents the total number of clauses. This Boolean circuit structure, utilizing 3-input OR gates, provides a natural mapping for the CNF format. To translate the CNF problem into Ising form, this Boolean circuit representation is directly leveraged. Similar to Choi and Chancellor, the 3OR transformation preserves the notion of clauses even on the hardware.

3OR Gate Construction: A 3-input OR gate is constructed by cascading two 2-input Ising OR networks, as illustrated in Fig. 6a. To accommodate negated literals within the clauses, the weights of edges connected to the respective nodes in the Ising network can be inverted, as shown in Fig. 6b. Consequently, all nodes in the Ising formulation correspond to the positive literals. It must be noted that although Fig. 6a contains 5 spins (not including the REF/+1 signal), as discussed in Sect. 3.3, to solve the SAT problem, the output node is forced to "+1", making the output node the same as the REF signal. Thus, the output spin and the REF signal can be merged. Indeed, every +1 node can be merged into a single node.



(a) All positive inputs (b) One inverted input

Fig. 6. 3OR Ising formulations

Table 1 contains the weights of the coupling at each spin of the 3OR gate's Ising equivalent. Symbols s_a, s_b, s_c in Table 1 are Ising spins that correspond to the literals a, b, and c in the clause $(a \vee b \vee c)$, and s_m is the additional spin required by the 3OR mapping. The diagonal entries represent the coupling between the spin and the REF/+1 signal, whereas the upper triangular entries, W_{ij}, represent the weights of coupling between the spins s_i and s_j. To handle negative literals, the *sign()* function is used as follows:

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x \text{ is a positive literal, } e.g., a, \\ -1, & \text{if } x \text{ is a negative literal, } e.g., \bar{a}. \end{cases} \tag{4}$$

For this Ising network, the minimum Hamiltonian is -4, irrespective of the signs of the literals.

Table 1. Coupling weights for different types of clauses $(a \vee b \vee c)$; $H = -4$

	s_a	s_b	s_c	s_m
s_a	sign(a)	sign(a)·sign(b)	0	−2·sign(a)
s_b		sign(b)	0	−2·sign(b)
s_c			−sign(c)	sign(c)
s_m				−3

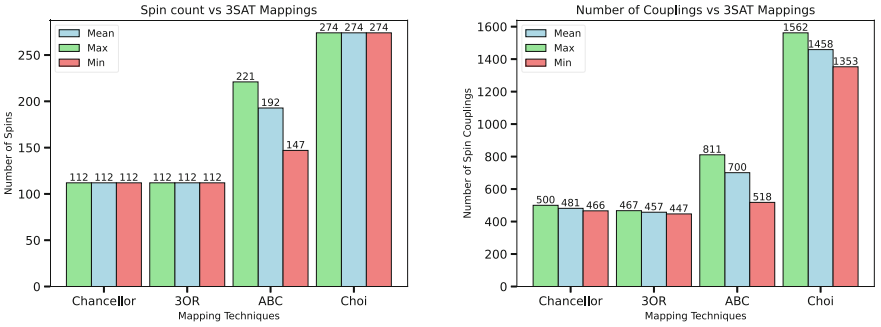
The output node not only serves as the output of the SAT network but also represents the output of the m-input AND gate. Consequently, ensuring the

output remains +1 mandates that each input of the AND gate be set to +1. Thus, constraining every clause's output to be +1 is equivalent to implementing an m-input AND gate with the output fixed at +1, making the m-AND implementation redundant.

6 Results

To evaluate the performance of OIM with the various mappings discussed earlier, we used the uf20 dataset of 3-SAT problems from SATLIB [1]. This dataset comprises 1000 randomly generated problems, with each problem referred to as a "3-SAT instance". Each 3-SAT instance consists of 20 variables and contains 91 clauses. We analyze the performance of different mappings using the following criteria.

Spin Count and Connectivity: In the context of solving the problem on hardware, especially IC implementations, an important feature of an efficient mapping is that the mapped Ising problem should use fewer resources, *i.e.*, fewer hardware spins and couplings. In Fig. 7, we show the number of spins and couplings required for uf20 problems with different 3-SAT-Ising mappings. From Fig. 7a, we observe that both Chancellor and 3OR require the minimum number of spins to represent the 3-SAT problem in Ising form. Additionally, Fig. 7b shows that 3OR requires fewer couplings compared to the other mappings. Therefore, of all the mappings, *the 3OR mapping consumes fewer hardware resources to represent the same 3-SAT problem in Ising form.*



(a) Spin counts (b) Coupling counts

Fig. 7. Comparison of spins and couplings needed to map uf20 problems into Ising form.

Comparing Mapping Techniques: In this experiment, we assess the effectiveness of various mapping techniques in terms of how easily the OIM can solve a given 3-SAT problem. We primarily consider two criteria to measure the effectiveness of the mappings: the fraction of solved instances and the success rate. To calculate the success rate of an instance, we execute the OIM on the instance 100 times with random initial conditions and record the fraction of runs that result in success, *i.e.*, a SAT solution is found. If the OIM fails to solve the problem in any of these 100 runs, we consider that instance to have failed. In the uf20 dataset, there are 1000 instances. We compute the success rate for all instances and report statistical quantities such as the mean and median for each mapping.

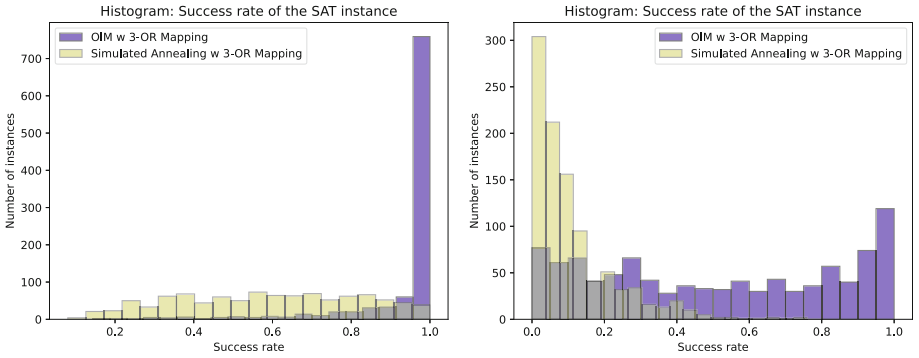
Table 2. Comparison between different mappings.

	Total Instances	Solved Instances	Fraction of Solved Instances	Mean Success Rate	Median Success Rate
ABC (no LGS)	1000	0	0	0	0
ABC (LGS)	1000	999	0.999	0.556	0.58
Choi	1000	990	0.99	0.393	0.29
Chancellor	1000	936	0.936	0.879	**1**
3OR	1000	**1000**	**1**	**0.959**	**1**

From Table 2, it is evident that level-based gate scaling (LGS, Sect. 4) significantly improves the performance of OIM on ABC-mapped 3-SAT instances, with the mean success rate increasing from 0 to 0.556. Overall, 3OR produced the best results, with a mean success rate of 0.959, which is notably higher than other mappings. Considering its requirement of fewer resources and excellent performance with OIM, *the 3OR mapping emerges as the ideal choice for 3SAT-Ising mapping.*

Table 3. Comparison between hardware solvers.

Benchmark	Solver	Total Instances	Solved Instances	Fraction of Solved Instances	Mean Success Rate	Median Success Rate
uf20	SA	1000	**1000**	1	0.587	0.59
uf20	OIM	1000	**1000**	**1**	**0.959**	**1**
uf50	SA	1000	920	0.92	0.109	0.07
uf50	OIM	1000	**984**	**0.984**	**0.501**	**0.48**



(a) uf20 benchmark results (b) uf50 benchmark results

Fig. 8. SATLIB benchmark results with 3-OR Ising mapping.

Comparing OIM and SA Using 3OR Mapping: We now compare the effectiveness of the OIM by comparing it to simulated annealing (SA) [16], a well-known method widely used for solving many combinatorial optimization problems. We solve 3OR-mapped uf20 and uf50 SAT problems with both OIM and SA, measure success rates for each problem, and plot a histogram of these in Fig. 8.

From Fig. 8a and 8b, it is evident that the histogram of OIM success rates is heavily weighted towards the right, implying that many uf20/uf50 instances have high success rates. With SA, however, the success rates are very low for most of the problems. As shown in Table 3, OIM achieves much better success rates and was able to solve greater number of instances compared to SA. *This indicates that OIM outperforms SA by a significant margin, highlighting its potential for solving practical and significant combinatorial optimization problems.*

7 Conclusion

In this paper, we introduced two novel techniques to improve the performance of OIM for logic-synthesis-based 3SAT-to-Ising mappings. Additionally, we presented a new circuit-based 3-SAT translation method called 3OR. Despite requiring the same number of spins and a similar number of couplings as the current state-of-the-art Chancellor mapping, we demonstrated that the success rate of OIM is higher with 3OR than with Chancellor. We also showed that OIM outperforms SA when it comes to solving Ising-translated 3-SAT instances.

Why one Ising mapping scheme performs better than another, not only in OIM but in other IM schemes, is poorly understood currently — it is an important direction for further exploration. We hope that empirical results, such as those presented here, will aid in unraveling this central question.

Acknowledgments. We gratefully acknowledge support from the US Defense Advanced Research Projects Agency (DARPA) and the US National Science Foundation (NSF). Additional support was provided by Berkeley's Bakar Prize Award.

References

1. SATLIB benchmark problems. https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html
2. Bhansali, P., Roychowdhury, J.: Gen-Adler: the generalized Adler's equation for injection locking analysis in oscillators. In: Proceedings of the IEEE ASP-DAC, pp. 522–227 (2009)
3. Bian, Z., Chudak, F., Israel, R., Lackey, B., Macready, W.G., Roy, A.: Discrete optimization using quantum annealing on sparse Ising models. Front. Phys. **2**, 56 (2014). https://doi.org/10.3389/fphy.2014.00056
4. Brayton, R., Mishchenko, A.: ABC: an academic industrial-strength verification tool. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 24–40. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14295-6_5
5. Cai, F., et al.: Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. Nature Electron. **3**(7), 409–418 (2020). https://doi.org/10.1038/s41928-020-0436-6
6. Camsari, K.Y., Faria, R., Sutton, B.M., Datta, S.: Stochastic p-bits for invertible logic. Phys. Rev. X **7**, 031014 (2017). https://doi.org/10.1103/PhysRevX.7.031014
7. Chancellor, N., Zohren, S., Warburton, P.A., Benjamin, S.C., Roberts, S.: A direct mapping of max k-SAT and high order parity checks to a chimera graph. Sci. Rep. **6**(1), 37107 (2016)
8. Choi, V.: Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3SAT problems. arXiv preprint arXiv:1004.2226 (2010)
9. Cılasun, H., et al.: 3SAT on an all-to-all-connected CMOS Ising solver chip. Sci. Rep. **14**, 10757 (2023)
10. Honjo, T., et al.: 100,000-spin coherent Ising machine. Sci. Adv. **7**(40), eabh0952 (2021). https://doi.org/10.1126/sciadv.abh0952
11. Inagaki, T., et al.: A coherent Ising machine for 2000-node optimization problems. Science **354**, 603–606 (2016). https://doi.org/10.1126/science.aah4243
12. Ising, E.: Beitrag zur theorie des ferromagnetismus. Zeitschrift für Physik **31**, 253–258 (1925). https://api.semanticscholar.org/CorpusID:122157319
13. Jagielski, T., Manohar, R., Roychowdhury, J.: FPIM: field-programmable Ising machines for solving SAT. arXiv preprint arXiv:2306.01569 (2023)
14. Johnson, M.W., et al.: Quantum annealing with manufactured spins. Nature **473**(7346), 194–198 (2011). https://doi.org/10.1038/nature10012
15. Karp, R.M.: Reducibility among Combinatorial Problems, pp. 85–103. Springer US, Boston, MA (1972). https://doi.org/10.1007/978-1-4684-2001-2_9
16. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
17. Lucas, A.: Ising formulations of many NP problems. Front. Phys. **2**, 74887 (2014). https://doi.org/10.3389/fphy.2014.00005
18. Marques-Silva, J.: Practical applications of boolean satisfiability. In: 2008 9th International Workshop on Discrete Event Systems, pp. 74–80 (2008). https://doi.org/10.1109/WODES.2008.4605925

19. Neogy, A., Roychowdhury, J.: Analysis and design of sub-harmonically injection locked oscillators. In: Proceedings of the IEEE DATE (2012)
20. Festa, P., Pardalos, P.M., Resende, M.G.C., Ribeiro, C.C.: Randomized heuristics for the max-cut problem. Optim. Methods Softw. **17**(6), 1033–1058 (2002). https://doi.org/10.1080/1055678021000090033
21. Roychowdhury, J., Wabnig, J., Srinath, K.P.: Performance of Oscillator Ising Machines on Realistic MU-MIMO Decoding Problems. Research Square preprint (Version 1) (2021). Web link to preprint
22. Sreedhara, S., Roychowdhury, J., Wabnig, J., Srinath, P.K.: MU-MIMO Detection Using Oscillator Ising Machines. In: Proceedings of the ICCAD, pp. 1–9 (2023)
23. Su, J., Tu, T., He, L.: A quantum annealing approach for boolean satisfiability problem. In: Proceedings of the IEEE DAC, pp. 1–6 (2016). https://doi.org/10.1145/2897937.2897973
24. Wang, T., Roychowdhury, J.: OIM: oscillator-based Ising machines for solving combinatorial optimisation problems. arXiv:1903.07163 (2019)
25. Wang, T., Roychowdhury, J.: OIM: oscillator-based Ising machines for solving combinatorial optimisation problems. In: Proceedings of the UCNC. LNCS sublibrary: Theoretical Computer Science and General Issues. Springer (2019). https://doi.org/10.1007/978-3-030-19311-9_19