CMOS Ising Machines with Coupled Bistable Nodes

Richard Afoakwa, Yiqiao Zhang, Uday Kumar Reddy Vengalam, Zeljko Ignjatovic, and Michael Huang
University of Rochester, Rochester, NY 14627 USA
{richard.afoakwa, yiqiao.zhang, u.venagala, zeljko.ignjatovic, michael.huang}@rochester.edu

Abstract—Ising machines use physics to naturally guide a dynamical system towards an optimal state which can be read out as a heuristical solution to a combinatorial optimization problem. Such designs that use nature as a computing mechanism can lead to higher performance and/or lower operation costs. Quantum annealers are a prominent example of such efforts. However, existing Ising machines are generally bulky and energy intensive. Such disadvantages might lead to intrinsic advantages at some larger scale in the future. But for now, integrated electronic designs allow more immediate applications. We propose one such design that uses bistable nodes, coupled with programmable and variable strengths. The design is fully CMOS compatible for on-chip applications and demonstrates competitive solution quality and significantly superior execution time and energy.

Index Terms—Ising machine, optimization, CMOS accelerators, max-cut problem

I. INTRODUCTION

The power of computing machinery has improved by orders of magnitude over the past decades. At the same time, the need for computation has been spurred by the improvement and continues to require better mechanisms to solve a wide array of modern problems. For a long time, the industry focused on improving general-purpose systems. In recent years, special-purpose designs have been increasingly adopted for their efficacy in certain type of tasks such as encryption and network operations [1], [2]. More recently, machine learning tasks have become a new focus and many specialized architectures are proposed to accelerate these operations [3]. Much of this work is to construct a more efficient architecture where the control overhead as well as the cost of operation becomes much lower than traditional designs.

In a related but different track of work, researchers are trying to map an entire algorithm to physical processes such that the resulting state represents an answer to the mapped algorithm. Quantum computers marketed by D-Wave Systems are prominent examples. Different from circuit model quantum computers [4], [5], D-Wave machines performs quantum annealing [6]. The idea is to map a combinatorial optimization problem to a system of qubits such that the system’s energy maps to the metric of minimization. Then, when the system is controlled to settle down to the ground state, the state of qubits can be read out, which corresponds to the solution of the mapped problem.

It is as yet not definitive whether D-Wave’s systems can reach some sort of quantum speedup. But one thing is clear: machines like these can indeed find some good solutions of an optimization problem, and in a very short amount of time too. Indeed, a number of alternative designs have emerged recently all showing good quality solutions for non-trivial sizes (sometimes discovering better results than the best known answer from all prior attempts) in milli- or micro-second latencies [9]–[11]. These systems all share the property: that a problem can be mapped to the machine’s setup and then the machine’s state evolves according to the physics of the system. This evolution has the effect of optimizing a particular formula called the Ising model (more on that later). Reading out the state of such a system at the end of the evolution thus has the effect of obtaining a solution (usually a very good one) to the problem mapped.

For example, in some systems, the Hamiltonian is closely related to the Ising formula. Naturally, the system seeks to enter a low-energy state. In other systems, a Lyapunov function of the system can be shown to be related to the Ising formula. In general, these systems can be thought of as optimizing an objective function (in the form of the Ising formula) due to physics. Hence, they are generally referred to as Ising machines. Clearly, unlike in a von Neumann machine, there is no explicit algorithm to follow. Instead, nature is effectively carrying out the computation. Ising machines have been implemented in a variety of ways with very different (and often complex) physics principles involved. It is unclear (to us at least) whether any particular form has a fundamental advantage that will manifest in a very large scale.

Note that these systems can not guarantee reaching the ground state in practice[2]. Nonetheless, some systems find a good answer with high speed and a good energy efficiency, as we shall see later with concrete examples. In this paper, we propose a novel CMOS-based Ising machine which uses circuit elements’ physical properties to achieve nature-based computation. This design is completely different from other efforts of using CMOS circuit to build machines that simulate an annealer. We perform a detailed analysis of the design and show that it is a compelling design and superior in many respects to existing Ising machines and accelerators of

1 Recent theoretical works have claimed increasingly strong equivalence between the two modes of quantum computing [7], [8].

2 Theoretical guarantee in some ideal setup may exist. For instance, adiabatic quantum computing theory says that when the annealing schedule is sufficiently slow and in the absence of noise (zero kelvin) the system is guaranteed to stay in the ground state [12].
II. BACKGROUND AND RELATED WORK

We first explain the background of Ising machines and discussed the state of the art in implementations.

A. Ising model

The Ising model is used to describe the Hamiltonian (sum of the energies of a system, e.g., kinetics and potential) of a system of spins. The model is a general one that describes a system with many nodes (e.g., atoms), each with a spin (σ_i) which takes one of two values (+1, −1). The energy of the system is a function of pair-wise coupling of the spins (J_{ij}) and the interaction of some external field (µ_i) with each spin (h_i). The resulting Hamiltonian is as follows:

\[ H = - \sum_{i<j} J_{ij} \sigma_i \sigma_j - \mu \sum_i h_i \sigma_i, \]  

(1)

If we ignore the external field, the Hamiltonian simplifies to

\[ H = - \sum_{i<j} J_{ij} \sigma_i \sigma_j \]  

(2)

This simplified version is more useful for the purpose of our discussion. Henceforth, when we refer to the Ising model or formula, we mean Eq. 2.

A physical system with such a Hamiltonian naturally tends towards low-energy states and thus serves as a convenient machine to solve a problem with a formulation equivalent to the Ising Hamiltonian – provided we can configure parameters (e.g., J_{ij}) to match that of the problem.

B. Optimization problems

A group of optimization problems naturally map to an Ising machine. Perhaps the most straightforward problem to map is Max-Cut. Given a graph, \( G = (V, E) \), where \( V \) is a set of vertices and \( E \) is a set of edges, a cut is a partition of vertices into two sets of, say, \( V^+ \) and \( V^- \), where; \( V^+ \cup V^- = V \) and \( V^+ \cap V^- = \emptyset \).

The Max-Cut problem tries to find a cut such that the combined weight of the edges spanning the two sets of vertices is maximum. In other words, the best cut is

\[ \arg \max \sum_{(i,j) \in E} W_{ij} \]  

(3)

where \( W_{ij} \) is the weight of edge \((i, j)\). (We will refer to the resulting \( \sum W_{ij} \) as the cut value in this paper.)

It is easy to see the resemblance between Eq. 2 and 3. In fact, if we set the coupling weight (J_{ij}) to be the negative of edge weight (−W_{ij}) then the Ising Hamiltonian is simply the negative cut value plus a constant as follows:

\[ H = \sum_{(i<j)} W_{ij} \sigma_i \sigma_j + \sum_{(i<j)} W_{ij} \sigma_i \sigma_j \]

(4)

Hence if the machine finds the ground state of the Hamiltonian, we have the max-cut. To find out the max-cut of an arbitrary graph is an NP-hard problem. Practical algorithms only try to find a good answer. Similarly, existing Ising machines (including our design) are all Ising sampling machines that attempt to find a good answer with no guarantee of optimality.

Finally, we note that we will only focus on the Max-Cut problem when evaluating the design. This is because Max-Cut is NP-complete [13] and thus all other NP-complete problems can be transformed as a Max-Cut problem with polynomial complexity. This means other NP-complete problems can be solved with either additional pre- and post-processing time or additional nodes for mapping. Both time and space overheads are bound by a polynomial complexity [14].

C. Quantum mechanical and optical Ising machines

There are many natural systems that can be described by the Ising model. Take two existing systems with relatively large footprints for example. D-Wave’s quantum annealers use superconducting qubits as the basic building block. These bits are then coupled together with couplers forming a connection topology known as the Chimera graph. This is an important architectural constraint that limits the typology of the problem that can be mapped to the machine. In practice, an abstract problem for D-Wave has to go through a transformation (called minor embedding) [15] to ensure that it can be mapped to the machine. This process involves mapping a logical node onto multiple physical nodes that are themselves coupled together strongly. In this way, in the solutions found, these physical nodes are (almost) always parallel with each other and thus can be considered as one logical node. We will see later on that this connection topology limits the number of effective nodes (spins) that a machine can offer. In the extreme example of problems with fully connected graphs, the number of physical nodes needed to map the problem grows quadratically with the number of logical nodes. Another disadvantage of the system is the cryogenic operating condition (15mK) needed for the quantum annealer. This requirement consumes a significant portion of the 25KW power of the machine [16].

Coherent Ising machines (CIMs) are another recent example of Ising sampling machines [9], [17]–[20]. In a CIM, an optical device called OPO (optical parametric oscillator) is used to generate and manipulate the signal to represent one spin. Unlike a D-Wave Ising machine, the coupling between spins is relatively straightforward in principle. As a result,
CIM implementations have always supported all-to-all coupling. The authors emphasized that the 2000-node CIM is therefore far more capable than D-Wave 2000Q which can only map problems of size 64 (or 61 after discounting defective nodes) [21]. In practice, not all problems of actual interest are on complete graphs, so the capability difference is less extreme. CIM is not without its disadvantages. To support 2000 spins, kilometers of fibers are needed. Temperature stability of the system is thus an acute engineering challenge. Efforts to scale beyond the currently achieved size (of about 2000) have not been successful as the system runs into stability problems. Also worth noting is that the coupling between nodes is – at least in the current incarnation – implemented via computation external to the optical cavity. There is a rather intensive computational demand (100s of GFLOPS) [22]. Every pulse’s amplitude and phase are detected and its interaction with all other pulses calculated on an auxiliary computer (FPGA). The computation is then used to modulate new pulses that are injected back into the cavity. Strictly speaking, the current implementation is a nature-simulation hybrid Ising machine. Thus, beyond the challenge of constructing the cavity, CIM also requires a significant supporting structure that involves fast conversions between optical signal and electrical signals.

These room-sized Ising machines are certainly worthwhile creations for the sake of science. In particular, it is uncertain whether the theoretical underpinning for these machines is relevant in practice. As we shall see later, both models have significant room for improvements.

D. Electronic oscillator-based Ising machines

A network of coupled oscillators is another physical implementation of an Ising machine. After sufficient time, the coupled oscillators will synchronize forming stable relative phase relationship. While many factors (e.g., amplitude, stochastic noise) will influence the phase of each oscillator, the following formula is a simplified steady-state description of phase relationship.

\[
\frac{d}{dt} \phi_i(t) = \sum_{j=1}^{N} J_{ij} \sin(\phi_j(t) - \phi_i(t))
\]

(5)

Note that this simplified model ignores certain elements (e.g., diffusion due to noise) and is thus an approximation of a more complicated reality. Given such a differential equation describing a dynamic system, it can be shown that a Lyapunov function in the following form exists:

\[
H(\Phi(t)) = - \sum_{i<j} J_{ij} \cos(\phi_j(t) - \phi_i(t))
\]

(6)

This means that the system will generally evolve along a trajectory that minimizes the Lyapunov function. In other words, if we build a network of coupled oscillators with certain coupling strengths \((J_{ij})\), the system’s stable states represent good solutions that minimizes the right hand side of Eq. 6.

On a closer inspection, we see the resemblance of Eq. 6 and the Ising model (Eq. 2). Specifically, when all phases \((\phi_j)\) are either 0 or \(\pi\), the two formulae are the same. A number of oscillator-based Ising machines have been recently proposed [10], [24], [25]. All these examples use LC tank oscillators. While this is a common practice for analog circuit designers and relatively straightforward for discrete-element prototypes, the use of LC tanks introduce non-trivial practical challenges in integrated circuit (IC) designs. The lack of high quality inductors and the usually high area costs of incorporating them are common challenges for integrated RF circuitry. These desktop Ising machines are a significant improvement (at least in size) over room-sized Ising machines. But, for genuine wide-spread applications, we believe a clean-slate IC-focused design is a valuable direction to pursue. Needless to say, we believe there will be significant cross-pollination of different approaches and future practice may very well be a confluence of multiple styles of Ising machines.

E. Accelerated simulated annealing

While a physical substrate Ising machine is in principle fast and efficient, an Ising machine can be emulated by a conventional von Neumann machine. Indeed, the classical technique of simulated annealing [26] is a good example. The principles behind simulated annealing has been broadly adopted in a variety of algorithms from Hopfield network to Boltzmann machines. A most relevant example is a number of recent CMOS designs using traditional memory and relatively simple logic specifically to accelerate simulated annealing [27], [28]. In these machines, there is no physical mechanism that naturally guides the system states towards some energy ground state. Instead, system state and its transitions are modeled by explicit computations on functional units customized to a particular algorithm (simulated annealing). We will see later that when done right, nature-based Ising machines understandably have significant speed and energy advantages over accelerators. For conceptual clarity, we refer to machines like [27], [28] as accelerated simulated annealers.

III. ARCHITECTURE OF THE CMOS IC ISING MACHINES

In this section, we describe the functional micro-architecture of our design. We start with a broad overview of our approach (Sec. III-A), then use a small discrete component implementation as a concrete example of a functioning system (Sec. III-B); and proceed to discuss a chip-scale design (Sec. III-D) and III-E.

3Indeed, the formulation of Eq. 6 is similar to the classic XY spin model (again ignoring external field): each spin can point to any direction along an “XY” plane and thus can be represented by a phase \((\phi_i)\). Ising model is thus a special case of the XY model. In other words, a system of coupled oscillators form an “XY machine” (not an Ising machine). An XY state can be quantized into an Ising state \((\phi_i = 0, \pi)\) in a number of different ways. For the sake of this paper, let us simply imagine direct quantization which rounds the phase to the nearest multiple of \(\pi\).
A. Overview & intuition

As already discussed before, existing designs have different strengths and weaknesses. The room-sized machines are good vehicles for continued scientific exploration of the underlying principles. But at the moment, they have no tangible benefits for immediate application. Electronic, oscillator-based Ising machines have already shown good problem solving capabilities but present real technical challenges for IC implementations. For example, a machine requiring an LC-tank in each node for its operation might not be suitable for integration in advanced CMOS technologies due to challenges associated with inductor scaling. Even though it is possible to scale on-chip inductors to smaller sizes in theory, this comes at the expense of requiring higher resonant frequencies (e.g., GHz range). A large-scale Ising machine necessarily contains many nodes spread over long distances with concomitant parasitics of the interconnect lines. Proper coupling at such high operating frequencies while preserving phase coherence is a real engineering challenge, if possible at all. In addition, it might be difficult to achieve purely resistive coupling of oscillator at GHz operating frequencies. Hence, we want to explore an IC-focused designs that have good performance characteristics and easy for CMOS integration.

Intuition: We start with a simple intuitive foundation. In the Ising model, when two nodes (say, i, and j) are strongly and positively coupled (i.e., \( J_{ij} \) is large and positive) their spins are likely to be parallel (\( \sigma_i = \sigma_j \)). In this way, the term \(-J_{ij} \sigma_i \sigma_j\) will contribute to lowering the energy. Conversely, a strong negative coupling (\( J_{ij} \) is large and negative) will likely lead to anti-parallel spins (\( \sigma_i = -\sigma_j \)). Finally, weak coupling (\( J_{ij} \) is small) suggests that the two spins are more likely to be independent.

This behavior can be easily mimicked with resistively coupled capacitors. Consider representing a node with capacitors in a differential manner where the polarity of the voltage represent the spin of the node. We can then connect nodes with different conductance/resistance. A strong coupling means a high conductance (lower resistor) value so that voltages of two nodes can more easily equilibrate. The sign of coupling can also be achieved with connecting either the same or opposite polarity in the differential circuit. Once initialized with random voltages, these coupled capacitors can indeed seek some temporary equilibrium – temporary because the energy stored in the capacitors will eventually dissipate through the coupling resistors and all nodes will decay to value 0, rather than staying at the desired \( \pm 1 \). To induce and maintain the nodes at \( \pm 1 \), we can introduce a local feedback unit to make the node voltages bistable. For brevity, we will refer to such a Bistable, Resistively-coupled Ising Machine as BRIM.

To show that this is a viable approach, we next discuss a concrete example using discrete components. We will then delve into the principle behind the operations and explore what characteristics of some circuit elements are needed for the machine to function as expected. Later (Sec. III-D and III-E) we will detail architecture and circuit design of a complete Ising machine for integrated circuit.

B. Example design with discrete electronics

In this section, we discuss an implementation of BRIM in discrete electronics with operational amplifiers and passive components such as capacitors as well as resistors, as described below. The topology overview of a discrete BRIM is shown in Fig. 1. At the heart of the proposed discrete implementation system is an array of bi-stable nodes (e.g., \( N_i, i = 1,2,\ldots,n \)) coupled through an all-to-all resistive coupling network with coupling units \( CU_{ij} \). Each bistable node \( N_i \) provides a differential output (\( v_i^+ \) and \( v_i^- \)) to the mesh of coupling units. Each coupling unit \( CU_{ij} \) has a pair of resistors \( R_{ij} \) connecting the differential outputs from two nodes (e.g., \( N_i \) and \( N_j \)). For positive coupling coefficients \( J_{ij} \), we couple them in parallel: \( v_i^+ \) to \( v_j^+ \) and \( v_i^- \) to \( v_j^- \). For negative coupling coefficients \( J_{ij} \), the coupling is "anti-parallel": \( v_i^+ \) to \( v_j^- \) and \( v_i^- \) to \( v_j^+ \). The resistor values in the coupling units are set to \( R_{ij} = R_C/|J_{ij}| \), where \( R_C \) is a constant resistance whose value is chosen appropriately to allow each node to converge to one of its bi-stable states.

Fig. 1. Overall architecture of the proposed discrete design BRIM for 4-node. \( N_i \) are nodes, \( CU_{ij} \) are coupling units, and blue lines are interconnects.

1) Bi-stable nodes: An example bi-stable node \( N_i \) implemented with discrete electronic components is depicted in Fig. 2. The circuit comprises of one energy storage element (capacitor \( C \)) giving rise to a state variable \( v_i(t) \) whose trajectory is described by the ordinary differential equations as shown in Eq. 7. Where \( I_X \) is coupling current from each other node connected to the current node, \( I_R \) is current through the node resistor, \( R \), \( I_{ZIV} \) is current through the diode, \( ZIV \).

---

6In fact, their theoretical foundation can be used to understand why they are necessarily sub-optimal. For instance, according to quantum adiabatic theory, the system’s Hamiltonian needs to be changed sufficiently slowly to guarantee that the system stays in ground state. In the interface that we were offered, the total annealing time is fixed, not problem dependent. Thus the system no longer guarantees to be adiabatic even assuming the absence of noise.
The circuit also includes a balanced diode implemented with two operational amplifiers as shown in Fig. 3. An example I-V curve is shown in Fig. 5, where the coupling resistors are chosen as \( R = 9.1k \Omega \), \( R_2 = 2.6k \Omega \), and \( LF412 \) opamps powered at +/-9V.

Fig. 2. Example of a BRIM node \((N_i)\) with coupling to two other nodes \((N_j \text{and } N_k)\).

The circuit also includes a balanced diode implemented with discrete components as shown in Fig. 4. The I-V curve is shown in Fig. 5. Because of the (slanted) "Z" shape of the IV curve, we call it a ZIV diode. The I-V curve has three equilibrium points \((i = 0)\) including the unstable one at the origin \((i = 0 \text{ and } v = 0)\) and two stable equilibrium points \((e.g., v = -2.3V \text{ and } v = +2.3V \text{ for load resistance of } 1.5K)\) giving rise to the bi-stable behaviour of the BRIM nodes.

An example discrete BRIM prototype implementing six bistable nodes is shown in Fig. 5. This particular prototype circuit implements a BRIM Ising machine for 6-vertex graph shown in Fig. 5 where the coupling resistors are chosen as \( R_{ij} = \frac{\text{[20k]}}{W_{ij}} \), where \( W_{ij} \) is the edge weight between the \( i^{th} \) and \( j^{th} \) node in the graph. After the circuit is powered up, the voltages \( v_i(t) \) converge to one of the stable states as shown in Fig. 6 depicting voltages from the prototype unit in Fig. 5. After the completion of the convergence, the voltages are compared against a threshold of 0V to measure a polarity and the results is presented as spins to the user.

C. Theoretical analysis

We now provide a Lyapunov stability analysis of the BRIM system which shows why BRIM has the ability to perform nature-based optimum-seeking. First, Eq. 7 can be simplified into two terms, as shown in Eq. 8, where \( g_{total} \) is a function of \( v_i \) combining the last two terms in Eq. 7.

\[
\frac{dv_i}{dt} = \sum_{j=1}^{N} J_{ij} v_j - J_i g_{total}(v_i)
\]

We are now able to construct a Lyapunov function \( H(v) \) of the BRIM system, where \( v(t) = [v_1(t) v_2(t) ... v_N(t)]^T \), by imposing the following construction rule.

\[
\frac{\partial H(v)}{\partial v_i} = -\frac{v_i(t)}{dt}, \quad i = 1, 2, ..., N
\]

One choice of Lyapunov function satisfying the conditions in Eq. 9 is shown in Eq. 10 where \( P(v_i(t)) \) is obtained from \( g_{total}(v_i) \) by integration over \( v_i \). It is important to notice that if \( g_{total} \) preserves the shape of the ZIV diode’s IV characteristic (i.e., retains negative gradient region around \( v_i = 0 \)), the term \( P(v_i(t)) \) will exhibit a double-well energy profile with two stable equilibrium points at voltages corresponding to two non-trivial zero-crossings of the IV curve (e.g., \( v_i = \pm 1V \)) and a saddle point at \( v_i = 0V \).

\[
H(v(t)) = - \sum_{i<j} J_{ij} v_j(t) v_i(t) - \phi_i(t) + \sum_{i} J_i P(v_i(t))
\]

Since function \( H(v) \) satisfies the conditions in Eq. 9 for all \( i \)'s representing a sufficient condition for \( \frac{dH(v)}{dt} = -\sum_i \frac{\partial H(v_i)}{\partial v_i} \frac{dv_i}{dt} \leq 0 \), the system is stable in the Lyapunov sense and it is guaranteed to converge to a minimum of \( H(v) \).
Fig. 5. Top: An example 6-vertex graph including both positive and negative edge weights $W_{ij}$ with the max-cut = 18.2. Bottom: 6-nodes discrete BRIM implementation of the graph above using LF412 opamps powered at $\pm9V$, $R = 1.5K$, $C = 50nF$, $R_1 = 2.6K$, $R_2 = 9.1K$, and $R_{ij} = \frac{120K}{|W_{ij}|}$.

– note that the minimum might not necessarily be the global minimum of function $H(v)$ and the system may converge to a local minimum depending on the energy landscape and initial conditions.

D. Architecture of an integrated design

We now discuss a more complete system and its components. While this design shares a general structure with the simplified example using discrete components, we introduce a number of variations that can be useful in improving the system’s flexibility. The system illustrated in Fig. 7 can be viewed as a group of components as follows:

1) Nodes and couplers:
At the left of the diagram are the bistable nodes $N_i$. Each of them contains two capacitors, two resistors, and a special diode to form a bi-stable, differential BRIM node with two terminals ($V^+_i$ and $V^-_i$). The nodes are connected to each other through a mesh of coupling units each with four terminals, two connected to two input nodes and two to the output nodes. Note here that the coupling is directed/unidirectional: this is achieved with a voltage buffer inside the node unit. In principle, an undirected/bidirectional coupling has similar effects. But empirically, we found directed coupling produces better solution quality at the expense of increased circuit area.

2) Programming units: Both the initial nodal values and the coupling resistance are programmable. Programming
Annealing scheduler: The coupling strength is adjustable over time for annealing. We use exponential annealing both because it is a common practice and it can be conveniently achieved using a discharging capacitor as the global annealing scheduler. At the end of the annealing, the state of the nodes will be read out from the nodes. With the stable voltage adjusted appropriately, the read out can be achieved with a simple flip-flop. The circuit details are discussed in Sec. III-E.

4) Perturbation unit: Finally, we find it useful to have the ability to flip the state of a selected node. This gives the system the ability to escape the current basin of attraction. We note that this is a form of introducing perturbation. An alternative is to add circuit level noise. While both can achieve similar results, introducing analog noise is harder to control and leads to more discrepancies between simulation and actual hardware.

Our BRIM is used similarly to other Ising machines: first programming the weights; then select the annealing length; and finally read out the state of the nodes. With the elements discussed, the system can be used in a number of different ways: the annealing time can be adjusted; the perturbation unit can be turned on with different frequency; the machine can be used with a software-based search algorithm (e.g., simulated annealing). We show some examples in Sec. IV.

E. Design of key circuits

The CMOS implementation of one BRIM node in 45nm technology is shown in Fig. 8. This circuit is fully balanced, therefore can be understood by analyzing either half of it. Transistors $M_3$ to $M_4$ are an integrated realization of the ZIV diode. It captures the overall IV characteristic of its discrete counterpart but with much less circuitry, namely, two cross-coupled inverters. Together with the parallel combination of resistors $R$ and capacitors $C$ connected on both sides of the ZIV diode, the basic structure of the CMOS BRIM node is formed. The rest of the circuit on each side consists of a variable gain buffer (VGB) and a current-controlled current source (CCCS) that together implement a direct coupling to other nodes in BRIM. Transistors $M_5$ to $M_6$ form a single-ended differential amplifier configured in a unity-gain topology. $M_{10}$ operates as a variable resistor when applying an annealing voltage $V_{\text{anneal}}$ to its gate. As the channel resistance of $M_{10}$ is changed from small to large by decreasing $V_{\text{anneal}}$, the effective output impedance of the VGB changes from almost zero (disconnecting the node from the coupling network) to its maximum value of about 0.9 V/V in this technology. Transistors $M_{13}$ to $M_{15}$ compose an integrated CCCS, which is modified from an active cascode summing circuit in [29]. $M_{13}$ is used to force the input voltage of the CCCS at a relatively fixed level. $M_{14}$ and $M_{15}$ form a current mirror to reflect current changes at the input of the node to the capacitor $C$ that stores the node’s state. Transistors $M_{11}$ and $M_{12}$ provide constant-current biasing for the CCCS unit. The same analysis applies to the circuitry on the left side of the ZIV diode.

$V_{b1}$ and $V_{b2}$ are used to bias the VGB and the CCCS, respectively. Their values are set by extra current sources which are not shown in Fig. 8. $V_{b1}$ produces an 8 $\mu$A tail current flowing into $M_5$ and $M_{16}$. While $V_{b2}$ produces four 10 $\mu$A current flowing into $M_{11}$, $M_{12}$, $M_{22}$, and $M_{23}$ separately. $V_{\text{CM}}$ is the common mode voltage of overall circuit and is used to bias $M_{10}$, $M_{21}$, $M_{13}$, and $M_{24}$.

The internal configuration of the coupling units is shown in Fig. 9. In each coupling unit, the differential inputs of one...
node (e.g., node $N_j$) and the differential outputs of another node (e.g., node $N_i$) can be coupled either in the same or the opposite polarity, thus four coupling resistors and two pairs of complementary switches are required. The coupling resistors are implemented as bootstrap switches/transistors commonly used in low-voltage switched-capacitor analog circuits. The parasitic capacitance $C_{gs}$ of the bootstrap transistors is used to store and maintain the biasing voltage $V_{gs}$ set by the row-level DAC, which regulates the channel resistance of the transistors. Switch $S_1$ and its complementary $S_1$ can be implemented as a single N-type transistor and single P-type transistor, respectively. The programming of coupling units is performed sequentially one column at the time. During the programming procedure of $i^{th}$ column, the column selector asserts the $CS$ switches in each of the coupling units in the column, which in turn connects the outputs of the row-level DACs to the selected coupling units. In addition, the column selector pull-down network connects the differential outputs in the $i^{th}$ column (and associated source terminals of the bootstrap transistor) to the common-mode voltage $V_{cm}$. The magnitude of $J_{ij}$ is then loaded into the DAC from the memory and converted to analog voltage supplied to the gate terminals of the bootstrap transistors setting their biasing voltage $V_{gs}$ to the desired value. The sign bit of $J_{ij}$ is passed from the MUX unit to the coupling units, which controls polarity switches $S_1$ and $S_1$. For example, if $J_{ij}$ is positive, the MUX turns on $S_1$ while keeping $S_1$ off, hence allowing DAC to charge gates of the bootstrap transistors connecting inputs to $N_j$ and outputs from $N_i$ with the same polarity. After bootstrap switches in $i^{th}$ column are pre-charged, the column selector moves to the subsequent column (e.g., $(i+1)^{th}$ column) and the procedure is repeated until coupling units in all columns are programmed.

The integrated BRIM circuit analyzed above is designed in 45nm technology node and simulated in Cadence Analog Design Environment. A test bench that maps to the 6-vertex graph in Fig. 5 is simulated and the example waveforms at the output of each node are shown in Fig. 10.

IV. EXPERIMENTAL ANALYSIS

A. Experimental setup

We compare BRIM to 4 other similar machines, 3 Ising machines and 1 accelerated simulated annealer, which we describe below. Except for the D-Wave machine, other tested Ising machines (including BRIM) are simulated using MATLAB (version R2019a) on a server cluster. The experimental characteristics of each machine are as follows:

1) D-Wave: We use Q2000 which is the latest quantum annealer [30]. We obtain results using software API provided by D-Wave [31]. Note that preprocessing is needed for this machine. A problem graph needs to go through minor embedding [15] in order to be able to map the hardware. After minor-embedding, the transformed graph is sent to the hardware for computation. Upon success, we receive final state spins, as well as computation times. For each graph problem, we collect 50 samples. In terms of timing, we do not specify any constraints, and adopt D-Wave default values. Specifically, 20 µs, 198 µs, 21 µs, and 11.7 ms respectively for annealing, data readout, inter-sample delay, and qubits programming [32].

2) CIM: We compare to the optical Coherent Ising Machine [9]. There is no known public access to the actual hardware and no model available for simulation. Thankfully, there is reported result for two commonly used benchmarks that allow us to make meaningful comparisons.

3) OIM: An Oscillator-based Ising Machine [10]. The machine is described in detail and allows us to construct differential equations for the RLC circuit model. This model is more realistic than the Kuramoto model (an approximate behavior model for generic synchronizations) used in their publications [10]. In any case, the two models of OIM largely agree in all cases.

$^7$This process currently takes significant time on a conventional computer and can fail. In our analysis, we ignore the time needed for this preprocessing.
4) **ASA:** Finally, a number of related proposals all try to accelerate simulated annealing using special hardware [27], [28]. Strictly speaking, these are not Ising machine, but customized accelerators for simulated annealing. They are, never the less, capable of solving the same optimization problems. These machines are fairly straightforward in design and can largely be modeled based on the description in literature. We focus on one with 30,000 nominal spins. In this design, the coupling following a near-neighbor pattern dubbed the King’s graph. All nodes are grouped into 4 groups. Every time step (0.22 µs), nodes in one of the groups will process in parallel; read off neighbor’s spin and the associated weights to compute which spin state lowers the energy in the neighborhood. Random bit flips similar to those in standard simulated annealing algorithms are also adopted. Similar to the D-Wave machine, a problem graph has to be preprocessed before it can map to the hardware. Given that the King’s graph is even more limited than the Chimera graph, the minor embedding process both takes longer time to complete and results in more physical nodes needed. Again, the time for the embedding process far dominate the actual annealing time. But in our analysis we ignore this time.

Both OIM and BRIM are modeled by solving their respective differential equations. We perform these simulations using MATLAB’s nonstiff, single step, 5th-order differential solver (ode45). Finally, as a reference, we also performed the classic Simulated Annealing (SA) [33] algorithm as a reference using MATLAB. For this simulation, we measure the execution time as the host machine time.

Note that all optimization problems for an Ising machine can be ultimately expressed as a Max-Cut problem with a weighted graph. To compare these systems, we use a commonly used set of graphs with diverse node sizes and edge densities. The details of each graph set are as follows;

1) **Regular graphs:** We use the “Gset” graphs from Stanford [34]. These graphs have between 800 to 20,000 nodes. The edges as well as the weights of such edges, were generated probabilistically, sometimes between +1 and -1, and sometimes all +1. We only use those graphs with less than 2000 nodes in our experiments.

2) **Small graphs:** Although supporting nominally 2048 spins, D-Wave’s machine can not map even the smallest graph in Gset. We therefore generate graphs with smaller node sizes (e.g., 200) and/or edge densities so that they can be successfully mapped onto D-Wave. For this purpose, we used rudy, a machine-independent graph generator [35], which is the same generator used to produce the “Gset” graph suite.

3) **Tiny graphs:** Finally, we also generated graphs with random edge densities and node sizes ranging from 16 to 32. For these graphs, we are able to enumerate all possible spin combinations to determine actual max-cuts.

### B. High-level comparison

It is important to keep in mind that Ising machines are far from mature. Early designs and prototypes are necessarily experimental in nature and thus lack the polish of, say, a conventional architecture. Much of the performance difference may be due to the art of prototyping rather than the fundamental science of the mechanism being exploited. This is perhaps especially the case for DWave, as in our comparison, it is the only actual hardware that we have access to. (CIM and OIM both have hardware prototypes but are not accessible to us.)

We start with a crude, high-level comparison of different Ising machines. There are several practical factors that make this comparison crude and incomplete. First, there is no single problem that can be measured on all machines. This is primarily because CIM only reported raw data on a very specific set of benchmarks and we are unaware of any reliable model of the physics that is publicly available. Second, the workload of optimization can usually allow tradeoff between speed and quality of the solution. Ideally, we will fix one metric (say, execution time) and compare the other (quality of solution). But in some cases, such control is unavailable to us. Third, the execution result depends on initial conditions. So any single run is subject to random chances. The common practice of using these machines is doing multiple runs and taking the best solution. This value should still be regarded as a random variable.

#### TABLE I

| Parameters | Dwave | CIM | OIM | ASA | BRIM |
|------------|-------|-----|-----|-----|------|
| Power (W)  | 25 K  | 210 | -   | -   | -    |
| Area (mm²) | -     | -   | 4.3 x 5.5 | -   | ≈ 1 |
| G22 Dist.  | -     | 46  | 263 | -   | 46   |
| Time       | -     | 5 ms| 5 ms| -   | 0.25 µs|
| G39 Dist.  | -     | 47  | 386 | -   | 46   |
| Time       | -     | 5 ms| 5 ms| -   | 0.25 µs|
| Small Graphs Dist. | 12 | -   | -   | -   | 7  |
| Time       | 20 µs| -   | 20 µs| -   | 20 µs|
| Tiny Graphs Dist. | 6  | -   | 16  | -   | 0   |
| Time       | 20 µs| -   | 20 µs| -   | 20 µs|
| Gset Avg Graphs Dist. | - | -   | 134 | -   | 7.4 |
| Time       | -     | 22 ms| -   | -   | 0.22 µs|

1) **Room-sized machines:** With these caveats in mind, Table I shows the estimated power, chip area (when applicable), and the execution time and answer quality of a few workloads. In terms of solution quality, we report the distance from the best reported cut value anywhere (the lower the better). First, we look at CIM using G22 and G39 because these are tested on CIM and reported (Fig. 3 of reference [9]). These graphs can not be mapped on D-Wave. For other machines, we try to match solution quality and show execution time. We see that BRIM can obtain similar quality solutions 4 orders of magnitude faster with a power consumption also about 4 orders
of magnitude better. Here we model a 1000-node BRIM built with 45 nm CMOS technology with all-to-all connection.

The next two row show small and graph comparisons intended to contrast D-Wave to others. Here because D-Wave current configuration/interface allows only fixed annealing time, we use the same annealing time for all machines. We see that in these cases BRIM always achieve the best solution while D-Wave’s solutions are not reach ground state. Recall these are the best answers of 50 runs. For tiny graphs, we can verify that BRIM and ASA always reach the ground state. Yet even for these, D-Wave does not always reach the ground state, with an average solution distance of 6. Here is a concrete example where the theoretical ability to reach the ground state is just that: a theoretical ability.

To sum, we see that the room-sized machines (CIM and D-Wave) do not show any tangible advantage in solving optimization problems. D-Wave can only map the smaller problems due to the connection limits. Even on these smaller problems, its solution quality trails behind others. Recall this limited connection means additional compute time to perform minor embedding, which at the moment takes 5 s, on average. CIM is less power-hungry and can map bigger problems due to its all-to-all connections. Nevertheless, there is no tangible advantage in any figure of merit. Again, these machines may (or may not) prove useful for scientific exploration and may (or may not) perform the computation. Indeed, here we see that to get its all-to-all connections. Nevertheless, there is no tangible advantage in any figure of merit. Again, these machines may (or may not) prove useful for scientific exploration and may (or may not) show some qualitative superiority at some other scale or at a future time.

The most important take-away point is that just because the machine leverages nature to perform computation does not necessarily make it efficient. Much engineering diligence is needed to convert some theoretical possibilities to tangible practical benefits.

2) Electronic designs: Next, we look at OIM. While the current OIM prototype is a desktop machine, in principle it can be scaled down in size and up in frequency. The primary unknown is to what extent the nodes can scale down and still operate like ideal LC-tanks. For this study, we assume a 100MHz frequency. Admittedly, this is nothing more than a rough guess. We see that OIM is perhaps comparable to BRIM, though subjectively, we feel the challenge is far more daunting than in BRIM.

Finally, we compare ASA with BRIM. ASA is essentially a specialized computer doing algorithmic search for better cut values. Clearly, ASA takes advantage of the tremendous cumulative improvements of CMOS technology to relatively fast and efficient computation. However, it is still a modified von-Neumann machine (with its only program hardwired). In contrast, BRIM is a bona-fide Ising machine where nature is performing the computation. Indeed, here we see that to get similar quality solutions on small graphs, BRIM is still much faster while consuming a tenth of the power, and with 20x lower area cost.

It is tempting to think that the area comparison is unfair because ASA offers a large number of nominal spins. In reality, the limited nearest-neighbor connection means that we need far more physical nodes than that in the graph. For instance, the process to embed Gset graphs takes hours of computer time and often fail due to memory capacity issues. Here, we primarily test ASA using small graphs that can be more easily mapped. Later, we will idealize it for a more detailed analysis.

3) Recap: We see that: ① for the scale of problems we are discussing, room-sized machines are not advantageous; ② ASA, OIM, and BRIM are three possible candidates for chip-scale applications with perhaps different strengths. We look at these models in more detail next.

C. Detailed analyses

We first show one execution example where ASA, OIM, and BRIM are all solving one problem: G22. We add simulated annealing running on a workstation to this mix. Since the total annealing time is a user-set parameter for operating an Ising machine, we take different values for each machine and show the results in Fig. 11. Each dot’s vertical coordinate represents the final solution energy (lower is better) while its horizontal coordinate shows the annealing time. The range of annealing times are chosen such that the quality of the solution fits into roughly the same band.

![Fig. 11. Solution quality measured as Ising energy (lower is better) against total annealing time.](image)

The next two row show small and graph comparisons intended to contrast D-Wave to others. Here because D-Wave current configuration/interface allows only fixed annealing time, we use the same annealing time for all machines. We see that in these cases BRIM always achieve the best solution while D-Wave’s solutions are not reach ground state. Recall these are the best answers of 50 runs. For tiny graphs, we can verify that BRIM and ASA always reach the ground state. Yet even for these, D-Wave does not always reach the ground state, with an average solution distance of 6. Here is a concrete example where the theoretical ability to reach the ground state is just that: a theoretical ability.

To sum, we see that the room-sized machines (CIM and D-Wave) do not show any tangible advantage in solving optimization problems. D-Wave can only map the smaller problems due to the connection limits. Even on these smaller problems, its solution quality trails behind others. Recall this limited connection means additional compute time to perform minor embedding, which at the moment takes 5 s, on average. CIM is less power-hungry and can map bigger problems due to its all-to-all connections. Nevertheless, there is no tangible advantage in any figure of merit. Again, these machines may (or may not) prove useful for scientific exploration and may (or may not) show some qualitative superiority at some other scale or at a future time.

The most important take-away point is that just because the machine leverages nature to perform computation does not necessarily make it efficient. Much engineering diligence is needed to convert some theoretical possibilities to tangible practical benefits.

Finally, we compare ASA with BRIM. ASA is essentially a specialized computer doing algorithmic search for better cut values. Clearly, ASA takes advantage of the tremendous cumulative improvements of CMOS technology to relatively fast and efficient computation. However, it is still a modified von-Neumann machine (with its only program hardwired). In contrast, BRIM is a bona-fide Ising machine where nature is performing the computation. Indeed, here we see that to get similar quality solutions on small graphs, BRIM is still much faster while consuming a tenth of the power, and with 20x lower area cost.

It is tempting to think that the area comparison is unfair because ASA offers a large number of nominal spins. In reality, the limited nearest-neighbor connection means that we need far more physical nodes than that in the graph. For instance, the process to embed Gset graphs takes hours of computer time and often fail due to memory capacity issues. Here, we primarily test ASA using small graphs that can be more easily mapped. Later, we will idealize it for a more detailed analysis.

3) Recap: We see that: ① for the scale of problems we are discussing, room-sized machines are not advantageous; ② ASA, OIM, and BRIM are three possible candidates for chip-scale applications with perhaps different strengths. We look at these models in more detail next.

C. Detailed analyses

We first show one execution example where ASA, OIM, and BRIM are all solving one problem: G22. We add simulated annealing running on a workstation to this mix. Since the total annealing time is a user-set parameter for operating an Ising machine, we take different values for each machine and show the results in Fig. 11. Each dot’s vertical coordinate represents the final solution energy (lower is better) while its horizontal coordinate shows the annealing time. The range of annealing times are chosen such that the quality of the solution fits into roughly the same band.

![Fig. 11. Solution quality measured as Ising energy (lower is better) against total annealing time.](image)

We can see the general trend that a longer annealing generally provides a better answer. BRIM is significantly faster than other systems with about 8 orders of magnitude faster than simulated annealing on a general-purpose processor. For ASA, since we have trouble mapping the larger, Gset graphs to its topology, we create an idealized model where it supports all-to-all connection and can update all nodes once (sequentially) in one cycle (4.54MHz). We refer to this as $ASA^U$ in Fig. 11. In addition, we take the reported [28] improvement factor $(2.6 \times 10^{45})$ over simulated annealing and draw a dotted line $(ASA^U)$ to indicate what the real implementation’s performance roughly is.

Some simulation of longer runs are still in progress at the time of writing. Remember each additional dot to the right takes 10x longer to simulate.
Rather than thinking about these curves as fixed and precise, it is helpful to think about them as a general shape that can move horizontally. Their position only represents the effect of current design parameters. And it is instructive to understand what it takes to move them to the left by, say, one order of magnitude. For ASA and SA, this is equivalent to making the machine 10x faster. We can imagine the challenge of this given the difficulty of scaling computation speed, not to mention memory access time. For the analog implementation, the question is more subtle. By reducing capacitance and/or inductance, the curve can shift to the left. The real question becomes the impact of noise and parasitics both to speed and solution quality. Such investigation is our future work.

While Fig. 11 shows the comparison of one benchmark, the trend is very similar across all benchmarks. In Table II, we provide the detail of solution quality for benchmarks of Gset with no more than 1000 nodes. We compare BRIM, the idealized ASA, and SA. Again, the solution is measured as the distance from the best run out of 50 to the best known answer in each benchmark. Here, we assume both Ising machines are large enough to map all nodes of the graph. For BRIM, the distances range from 0 to 21, with a mean of 7.3 and a median of 7. In contrast, the distance for ASA (1ms) ranges from 0 to 42 with a mean of 15.10, and a median of 14. Additionally, the distance for SA (10 Minutes) ranges from 0 to 6 with a mean of 1.2, and a median of 0.

D. Perturbation

As already mentioned earlier, this paper highlights a different approach to Ising machine. The specific example used so far is but one design point in the space and much of the space remains to be explored in more detail. A perturbation unit is one example of optimizations to the basic machine. Fig. 12 illustrates the effect of the perturbation support. We compare the two runs of BRIM: with and without perturbation. The two runs are initialized with the same random starting state.

We show the energy as time progresses. Perturbation clearly helps the system explore more a single basin of attraction.

V. Conclusions

Ising machines can be programmed to map an abstract problem and let physics naturally guide the dynamical system towards some kind of optimal state. This state can then be translated back to be a heuristic solution to a combinatorial optimization problem. The use of nature suggests the possibility of significant speed and/or efficiency. Consequently, exploration of Ising machines has been gaining attentions. Quantum annealers and optical coherent Ising machines are prominent examples of such machines which drew particular interests from the physics community. However, these Ising machines are generally bulky and energy intensive. Continued exploration is certainly a worthwhile endeavor for scientific purposes. But for now, integrated circuit designs allow more immediate applications. We propose one such design we call BRIM that uses bistable nodes resistively coupled with programmable and variable strengths. The design is fairly CMOS compatible for on-chip implementations. Through our experimental analysis, we show that the machine is significantly smaller, faster, and more energy-efficient relative to some other designs. Compared to the room-sized Ising machines, it is about 2-3 orders of magnitude faster and consuming 5-6 orders of magnitude less energy. Compared to a recent chip-scale simulated annealing accelerator, it is also orders of magnitude faster with about 10x less in power. While there is some degree of uncertainty with these statistics, we also envision continued improvement and optimizations. On the other hand, there are also many challenges facing machines like BRIM. Scalable architecture, area-efficient connection networks, and circuit scaling in the presence of PVT variations are but a few examples of the challenges.

References

[1] C. Johnson, D. H. Allen, J. Brown, S. Vanderwiel, R. Hoover, H. Achilles, C. Cher, G. A. May, H. Franke, J. Xenedis, and C. Basso,
“A wire-speed powertm processor: 2.3ghz 45nm soi with 16 cores and 64 threads,” in 2010 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers, pp. 354–612, 2010. [Online]. Available: https://doi.org/10.1109/ISSCC.2010.5436124

[23] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam, “Dadiannao: A machine-learning super-computer,” in 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, Dec 2014, pp. 609–622.

[24] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, H. Jiang, P. Love, A. L. Su, D. Wang, P. OBrien, A. Peruzzo, C. Rubin, C. Rose, J. Wang, B. Colless, C. Crystal, N. Jones, T. Parameswaran, M. Troyer, M. Veitch, A. Vuletic, M. Widera, S. Yoneda, and Z. Su, “Quantum supremacy using a programmable superconducting processor,” Nature, vol. 549, no. 7671, pp. 222–225, 2017. [Online]. Available: https://doi.org/10.1038/nature23462

[25] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsonomiyu, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, and Y. Yamamoto, “A fully programmable 100-spin coherent ising machine with all-to-all connections,” Science, vol. 361, no. 6321, pp. 604–607, 2018. [Online]. Available: https://science.sciencemag.org/content/361/6321/604

[26] K. Takata, A. Marandi, R. Hamerly, Y. Haribara, D. Marioo, S. Tamate, H. Sakaguchi, S. Utsonomiyu, and Y. Yamamoto, “A 16-bit ising ising machine for one-dimensional ring and cubic graph problems,” Scientific Reports, vol. 6, no. 1, p. 34089, 2016. [Online]. Available: https://doi.org/10.1038/srep34089

[27] F. Boi, G. Verschaffel, and G. Van der Sande, “A poor man’s coherent ising machine based on opto-electronic feedback systems for solving optimization problems,” Nature Communications, vol. 10, no. 1, p. 3538, 2019. [Online]. Available: https://doi.org/10.1038/s41467-019-11484-3

[28] R. Hamerly, T. Inagaki, F. T. McMahon, D. Venturrelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, K. Enbutu, T. Umeke, R. Kasahara, S. Utsonomiyu, S. Kako, K. iichi Kawarabayashi, R. L. Byer, M. M. Fejer, H. Mabuchi, D. Engelund, E. G. Rieffel, H. Take- sue, and Y. Yamamoto, “Experimental investigation of performance differences between coherent ising machines and a quantum annealer,” in Science advances, 2019.

[29] H. Takesue, T. Inagaki, K. Inaba, T. Ikuta, and T. Honjo, “Large-scale coherent ising machine,” Journal of the Physical Society of Japan, vol. 88, no. 6, p. 061014, 2019. [Online]. Available: https://doi.org/10.7566/JPSJ.88.061014

[30] M. G. Rosenblum, A. S. Pikovsk, and J. Kurths, “Phase synchronization of chaotic oscillators,” Physical review letters, vol. 76, no. 11, p. 1804, March 1996.

[31] T. Wang, L. Wu, and J. Roychowdhury, “New computational results and hardware prototypes for oscillator-based ising machines,” in Proceedings of the 56th Annual Design Automation Conference 2019, ser. DAC’19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3316781.3322473

[32] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog, “Analog coupled oscillator based weighted ising machine,” Scientific Reports, vol. 9, no. 1, p. 14786, 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-49609-5

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” science, vol. 220, no. 4598, pp. 671–680, 1983.

[34] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, “24.3 20k-spin ising chip for combinatorial optimization problem with cmos annealing,” in 2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers, Feb 2015, pp. 1–3.

[35] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, “2.6 a 2 by 30k-spin multichip scalable annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems,” in IEEE International Solid-State Circuits Conference, February 2019.

[36] C. D. M. A. e. a. Comer, D. J., “A high-frequency cmos current summing circuit,” Analog Integrated Circuits and Signal Processing 36, 2003.

[37] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Croia, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose, “Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor,” Phys. Rev. B, vol. 82, p. 024511, Jul 2010. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.82.024511

[38] “dwave-system,” accessed 2020-04-13. [Online]. Available: https://docs.dwave.com/en/stable

[39] “Breakdown of qpu access time,” accessed: 2020-04-13. [Online]. Available: https://docs.dwave.com/docs/latest/c/timing.html

[40] T. G. J. Myklebust, “Solving maximum cut problems by simulated annealing,” 1982. [Online]. Available: https://web.stanford.edu/~yyye/yyye/Gset

[41] C. Hellberg and F. Rendi, “A spectral bundle method for semidefinite programming,” SIAM J. on Optimization, vol. 10, no. 3, p. 673696, Jul. 1999. [Online]. Available: https://doi.org/10.1137/S1052623497328897

[42] Q. Wu and J.-K. Hao, “Memetic search for the max-bisection problem,” Computers & Operations Research, vol. 40, no. 1, p. 166 – 179, 2013.

[43] F. Ma, J.-K. Hao, and Y. Wang, “An effective iterated tabu search for the maximum bisection problem,” Computers & Operations Research, vol. 39, no. 12, pp. 1095–1103, 2012.
