



OPEN Augmenting an electronic Ising machine to effectively solve boolean satisfiability

Anshujit Sharma , Matthew Burns, Andrew Hahn & Michael Huang

With the slowdown of improvement in conventional von Neumann systems, increasing attention is paid to novel paradigms such as Ising machines. They have very different approach to solving combinatorial optimization problems. Ising machines have shown great potential in solving binary optimization problems like MaxCut. In this paper, we present an analysis of these systems in boolean satisfiability (SAT) problems. We demonstrate that, in the case of 3-SAT, a basic architecture fails to produce meaningful acceleration, largely due to the relentless progress made in conventional SAT solvers. Nevertheless, careful analysis attributes part of the failure to the lack of two important components: cubic interactions and efficient randomization heuristics. To overcome these limitations, we add proper architectural support for cubic interaction on a state-of-the-art Ising machine. More importantly, we propose a novel semantic-aware annealing schedule that makes the search-space navigation much more efficient than existing annealing heuristics. Using numerical simulations, we show that such an “Augmented” Ising Machine for SAT is projected to outperform state-of-the-art software-based, GPU-based and conventional hardware SAT solvers by orders of magnitude.

Semiconductor fabrication technology and advanced von Neumann computer architecture have been two key drivers for the explosive increase in computing system performance in the past 50 years or so. As both drivers are reaching a phase of diminishing returns, interest in non-von Neumann systems is increasing. One obvious example is quantum computing (QC). QC has become more widely discussed ever since Peter Shor proposed an exponentially faster quantum algorithm for factorization¹. However, we have yet to discover a wide array of applications where QC can change the complexity class of the solution algorithm. Grover’s algorithm is another popular quantum algorithm that theoretically provides a quadratic speedup for unstructured search². In practice, however, realities of error sensitivity in the “Noisy-Intermediate Scale Quantum” (NISQ) era devices limit such algorithms’ immediate usefulness³. More noise-resilient, “shallow-depth” algorithms have been proposed^{4–6}. So far, there is no indication that they can overcome NISQ era hurdles.

If we recall Feynman’s motivation of building quantum mechanical systems as computers⁷, we see that the argument applies to classical dynamical systems as well. Large-scale natural or artificial systems (e.g., weather or electronic circuits) can be modeled by ordinary or stochastic differential equations (ODE/SDE), but the modeling can be computationally very expensive. Yet nature seems to have little trouble “solving” these ODEs/SDEs effortlessly and often much more quickly than von Neumann systems. Hence building a *physical* dynamical system is in some sense building a “nature-based” computing system. One such type of dynamical systems that are gaining attention lately are often called *Ising machines*.

The Ising model is a physics-oriented model that describes the energy of a system of N coupled spins (See Methods section for more details). Finding the ground state of the Ising model is a classical example of combinatorial optimization problems (COP). Although early study focused on spins on 1D⁸ lattice, the prevailing notion of Ising model is not restricted to any graph topology, and is thus equivalent to the quadratic unconstrained binary optimization (QUBO) model. A generic QUBO problem is NP-complete and usually solved by physics-inspired Simulated Annealing (SA) algorithm. When an Ising machine maps such a problem, its dynamics can be extremely fast and much more efficient than SA on a von Neumann system. Quantum annealers (QA) marketed by D-Wave Systems⁹ are but early examples. The Coherent Ising machine (CIM) is another prominent example. CIM uses modulated optical pulses to represent spins¹⁰. In a recent work, NTT developed a 100,000-node CIM¹¹, thus demonstrating a scalable implementation. Since the working principle of CIM can be attributed to the Kuramoto model¹², researchers have proposed other electronic Oscillator-based Ising machines (OIM)¹³.

Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA. ✉email: ashar36@ur.rochester.edu

Other emerging technologies have also been proposed to build Ising machines like memristors¹⁴ and p-bits¹⁵. A more thorough survey of Ising machines can be found in¹⁶.

In this paper, we will take an application-centric approach to analysing Ising machines with *boolean satisfiability* (SAT) as a case study¹⁷ (See Methods section for more details). SAT is an excellent benchmark: While many applications can be mapped into the Ising formula, there are nuances in the case of SAT problems: *high-degree interaction* and *efficient searching heuristic*. High-degree Ising machines have been proposed in the past^{18–21}, but these completely ignored the second point. As we shall see, this has a serious implication on the machine's capabilities to achieve state-of-the-art performance. For concreteness, our experiments will focus on an electronics-based CMOS-compatible Bistable Resistively-coupled Ising machine (BRIM)^{22–25}. It has several key advantages compared to previous works. First, it uses the signs of the voltage across capacitors to represent Ising spins, thus making it easily realizable. Second, it uses programmable resistive coupling units, thus making it easy to implement all-to-all couplings. Third, it relies on natural dynamics of the system based on nano-scale capacitive spins, thus making it fundamentally more efficient than von Neumann accelerators.

The main contributions of this paper can be summarized as follows. First, we demonstrate that standard Ising machines are not competitive in solving SAT compared to state-of-the-art SAT solvers. Our analysis show that the main causes are lack of support for high-degree interaction and efficient search-space navigation. Second, motivated by these limitations, we extend a baseline Ising machine to support high-degree terms. Third, more importantly, we propose a new semantic-aware annealing heuristic to significantly improve Ising machines' navigational efficiency. Finally, using numerical simulation, we demonstrate that our proposed "augmented" Ising machine is projected to be orders of magnitude faster than existing state-of-the-art SAT solvers.

Results

Solving 3-SAT using Ising machines

While Ising machines have been shown to be competitive in solving generic QUBO (general MaxCut) problems^{22,23,26}, their application to SAT has not been as successful. This is partly due to the fundamental structure of k -SAT for $k \geq 3$: Given an N -variable, M -clause 3-SAT formula F , we can construct a minimization energy function (H) as follows:

$$F = \bigwedge_{i=1}^M (\ell_{i1} \vee \ell_{i2} \vee \ell_{i3})$$

$$H = \sum_{i=1}^M g(\ell_{i1})g(\ell_{i2})g(\ell_{i3}) \quad (1)$$

where,

$$g(\ell_{ij}) = \begin{cases} (1 - x_n) & \ell_{ij} = x_n \\ x_n & \ell_{ij} = \neg x_n \end{cases} \quad (2)$$

where, ℓ_{ij} is the j th literal in the i th clause and $n \in \{1, 2, \dots, N\}$. Some state vector x^* satisfies F if and only if $H(x^*) = 0$. Here, x^* is called the *ground state* of H (it is assumed that $x_n = 1$ means *True* and $x_n = 0$ means *False*). Any *unsatisfying* state x will have $H(x) > 0$.

It is not difficult to see that after expanding the summation, H will have cubic terms, $x_i x_j x_k$. However, the Ising/QUBO formula only supports upto quadratic (2nd-degree) terms. To get around this problem, the standard approach is to *quadratize* these high-degree terms^{27–29} to get a QUBO problem.

Figure 1 shows the Time-To-Solution (TTS) of BRIM and D-Wave's QA after quadratization, compared against a state-of-the-art SAT solver, KISSAT³⁰. TTS³¹ is defined as the expected time to find a solution with

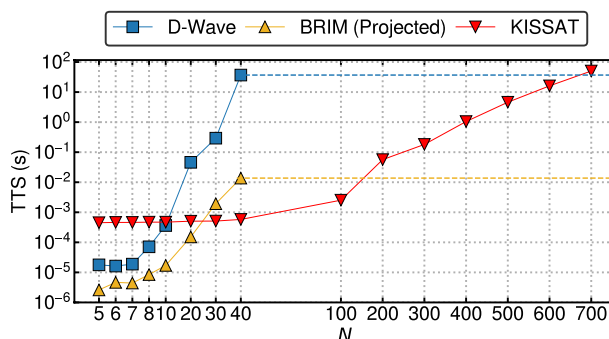


Figure 1. Time-To-Solution (TTS) of BRIM and D-Wave annealer compared against a state-of-the-art SAT solver, KISSAT³⁰ for uniform random problems with varying number of variables, N and clauses, $M = 4.25N$. The yellow and blue horizontal dashed line depicts how big of a problem KISSAT can solve for the same TTS taken by D-Wave and BRIM respectively while solving 40-variable problems.

99% probability. Over multiple instances each with runtime of T_{run} and a success probability of P , TTS can be calculated using the following formula:

$$TTS = \begin{cases} T_{run} \times \frac{\log_{10}(0.01)}{\log_{10}(1-P)} & P < 0.99 \\ T_{run} & \text{otherwise} \end{cases} \quad (3)$$

For SAT, success probability P is the proportion of runs that found satisfying states out of all the runs for a given problem. Each data point in the figure is the geometric mean TTS of 10 different problem instances. We can observe that while BRIM's trend is much better than that of D-Wave, compared to KISSAT, the initial speed advantage of both the Ising machines is quickly lost with only a small increase in problem size. The horizontal dashed lines show that for the same TTS that D-Wave and BRIM takes to solve a 40-variable problem, KISSAT can solve ~ 700 and ~ 150 -variable problems respectively. There are two reasons for this disappointing performance of Ising machines:

1. **Lack of support for high-degree interactions:** Fig. 2a shows the resultant number of nodes after using Kolmogorov-Zabin-Freedman-Drineas (KZFD)^{28,29} and Rosenberg²⁷ quadratization for increasing number of variables. We can observe that roughly, there is a 4–5x increase in the number of nodes compared to the number of variables, resulting in an increased number of phase points from 2^N to $\sim 2^{5N}$. This is because both quadratization techniques introduce roughly 1 extra variable per clause. Therefore, the resultant number of nodes is $\sim (N + M)$, where $M \approx 4.25N$ for hard uniform random 3-SAT instances. Such a massive disadvantage has been demonstrated in earlier works which motivated many researchers to propose Ising machines with high-degree interactions^{18–21}. Figure 2b shows the TTS of SA³² on the QUBO formulations (SAq) and with cubic interaction support (SAc) compared against KISSAT. We can observe that SAc significantly outperforms SAq, thus showing the importance of high-degree interactions. However, it still scales poorly when compared to KISSAT, which brings us to the next point.
2. **Lack of efficient searching heuristics:** After converting to the Ising/QUBO formulation, insights from the original problem formulation may not be obvious. However, such insights are exploited by software SAT solvers like KISSAT to effectively search the landscape³³. This gap between SAc and KISSAT is the result of efficient searching heuristics. Previous works on high-degree Ising machines completely ignored the vast amount of insights from SAT competitions³⁴ and literature^{33,35,36}.

These two shortcomings motivate us to propose hardware support for cubic interaction and a semantic-aware heuristic to effectively guide the Ising machine towards a solution as we will discuss in the next section.

Apart from these limitations, D-Wave QA also suffers from an additional drawback: lack of support for all-to-all coupling. The limited connectivity of the hardware topology necessitates a costly graph embedding process (which is itself NP-hard) that greatly increases the final graph size³⁷. This is the major reason for its worse performance scaling compared to BRIM (which supports all-to-all couplings) in Fig. 1. Any Ising machine only supporting near-neighbor couplings will suffer from this fundamental limitation²².

BRIM with support for cubic interaction

Let us rewrite the 3-SAT objective function (Eq. 1) by collecting linear, quadratic, and cubic terms:

$$H = \text{const} - \sum_n l_n x_n - \sum_{n < j} q_{nj} x_n x_j - \sum_{n < j < k} c_{njk} x_n x_j x_k \quad (4)$$

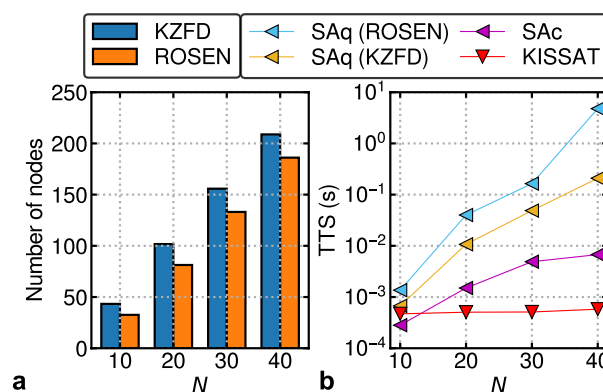


Figure 2. Illustration of drawbacks of Ising machines. (a) The average number of nodes after quadratization for increasing number of variables. (b) Performance of SA on the QUBO formulations (SAq) and on the original hypergraph with support for cubic interaction (SAc) compared against KISSAT.

In BRIM, each variable x_n is represented by a continuous capacitive voltage $v_n \in [0, 1]$. Moreover, the output nodal voltage is *quantized* to $\{0, 1\}$ V with 0.5 V being the threshold. A real quantizer can be represented as a scaled and shifted-tanh function $Q(v_n)$ with a steep slope, $S > 0$. Hence, we get a continuous variable energy function $\tilde{H}$ from Eq. (4):

$$\tilde{H} = \text{const} - \sum_n I_n Q(v_n) - \sum_{n < j} q_{nj} Q(v_n) Q(v_j) - \sum_{n < j < k} c_{njk} Q(v_n) Q(v_j) Q(v_k) \quad (5)$$

where, $Q(v_n) = 0.5 \cdot \tanh\{S \cdot (v_n - 0.5)\} + 0.5$; $x_n = \lim_{S \rightarrow \infty} Q(v_n)$

Now, the incoming current for each node is given the following equation:

$$\frac{dv_n}{dt} = \alpha \left(I_n + \sum_j q_{nj} Q(v_j) + \sum_{j,k} c_{njk} Q(v_j) Q(v_k) \right) = -\alpha \frac{\partial \tilde{H}}{\partial Q(v_n)} = -\frac{\alpha}{0.5 \cdot S \cdot \text{sech}^2\{S \cdot (v_n - 0.5)\}} \cdot \frac{\partial \tilde{H}}{\partial v_n}; \alpha > 0 \quad (6)$$

We modify the baseline BRIM as proposed by Afoakwa et al.²² for bit- $\{0, 1\}$ representation as shown in the blue block in Fig. 3a. This supports the first two terms ($I_n + \sum_j q_{nj} Q(v_j)$) in the parenthesis of Eq. (6). There are N bi-stable capacitive nodes whose voltages range $\in [0, 1]$. The bit value of each node is simply the *quantized* voltage $\in \{0, 1\}$. Thus, each node represents a variable in the SAT formula. Every node is connected to its neighbors via programmable resistive coupling units q . This resistance between node n and node j is given by $R_{nj} = \frac{R}{|q_{nj}|}$, where R is a constant resistance and q_{nj} is normalized to $[-1, +1]$. Thus, strong coupling means lower resistance. The sign of q_{nj} is implemented as follows: when $q_{nj} > 0$, the nodes are coupled in a parallel fashion (output of one node is connected to the positive input terminal of the other, via resistance R_{nj} : $\text{Out}_n \rightarrow \text{In}_j^+$, $\text{Out}_j \rightarrow \text{In}_n^+$); when $q_{nj} < 0$, they are coupled in an anti-parallel fashion ($\text{Out}_n \rightarrow \text{In}_j^-$, $\text{Out}_j \rightarrow \text{In}_n^-$); nodes are disconnected if $q_{nj} = 0$. Each node n can also have a linear bias I_n .

To support the last term ($\sum_{j,k} c_{njk} Q(v_j) Q(v_k)$), we require 3 steps: ① use a multiplier to produce a voltage $v_{jk} = Q(v_j) \times Q(v_k)$, ② apply v_{jk} across a resistor proportional to $\frac{1}{c_{njk}}$, and ③ feed the current to node n . In practice, we make 3 optimizations as we discuss next:

1. **Multiplier:** A “true” analog multiplier is a rather expensive circuit due to accuracy and linearity concerns. Since the differential equation (Eq. 6) itself is a continuous approximation of discrete variables and the output voltage of nodes is *quantized*, we use a simple AND gate as a multiplier.

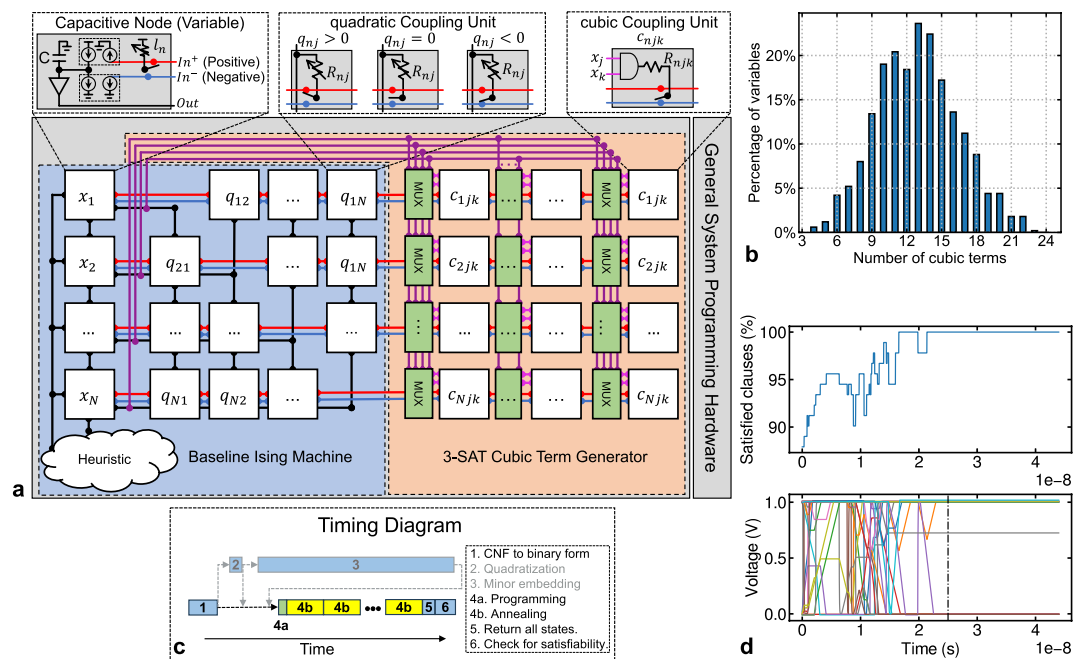


Figure 3. Illustration of cBRIM. (a) High-level cBRIM diagram extending the baseline BRIM with MUXs and cubic coupling units. (b) Distribution of number of variables participating in cubic terms. (c) Timing diagram of using cBRIM. The greyed-out steps are not required for cBRIM (but may be required for other Ising machines). (d) The evolution of the number of satisfied clauses (%) [top] and nodal voltages for all nodes in cBRIM [bottom] while solving a 20-variable problem. The vertical dash line indicates that a solution is found when the voltages have settled.

2. *Cubic terms fan-in*: For a general support of cubic interaction, we need $O(N^3)$ couplers. However, for 3-SAT, the number of cubic terms is determined by the number of clauses, M which, for difficult uniform random problems, is typically about $4.2N^{38}$. Therefore, on average, each variable is involved in about a dozen cubic terms—regardless of the size of the problem. Figure 3b shows the distribution of variables occurring in cubic terms for uniform random problems. As we can observe, the plot doesn't have a long tail and a fixed degree fan-in of $d = 20$ would work. Every multiplier (AND gate) is fed by two $N:1$ multiplexers to program which voltages to multiply. This keeps the hardware complexity at $O(N^2)$. Given a fan-in limit chosen for a particular system, users can pre-process excess clauses with quadratization techniques.
3. *Coupler programmability*: Finally, another problem-specific feature of 3-SAT is that the vast majority of cubic term coefficients are ± 1 . For an M -clause 3-SAT problem, there are at most M cubic terms produced by Eq. (1), one for each clause with 3 literals. On rare occasions, when the magnitude of a coefficient is greater than 1, that cubic term is simply treated as multiple instances that take up multiple couplers (fan-in). So we eliminate the programmability on the coupler strength—it will be fixed at 1. Instead, the programmability is on which two inputs to multiply, and the polarity of the coupling.

The resulting architecture is shown in Fig. 3a. We name this design, cubic BRIM or cBRIM in short. By minimizing $\tilde{H}$, cBRIM maximises the number of satisfied clauses and thus, solves Max-3-SAT. When there is no random perturbation, using Eq. (5) and Eq. (6), we can show the Lyapunov stability analysis as follows:

$$\frac{d\tilde{H}}{dt} = \sum_n \frac{\partial \tilde{H}}{\partial v_n} \cdot \frac{dv_n}{dt} = \sum_n \frac{\partial \tilde{H}}{\partial Q(v_n)} \cdot \frac{\partial Q(v_n)}{\partial v_n} \cdot \frac{dv_n}{dt} = -\frac{1}{\alpha} \sum_n \left(\frac{dv_n}{dt} \right)^2 \cdot \frac{\partial Q(v_n)}{\partial v_n} = -\frac{1}{\alpha} \sum_n \left(\frac{dv_n}{dt} \right)^2 \cdot 0.5 \cdot S \cdot \text{sech}^2\{S \cdot (v_n - 0.5)\} \quad (7)$$

Now, v^* is a solution to a 3-SAT problem if and only if $\tilde{H}(v^*) = 0$. Otherwise, $\tilde{H} > 0$. Moreover, since $\alpha, S > 0$ and $v_n \in [0, 1]$, from Eq. (7), $(\frac{d\tilde{H}}{dt} \leq 0)$ for all $v \neq v^*$. Therefore, the system is Lyapunov stable: it naturally seeks local minima of $\tilde{H}$ and stays there when found. Figure 3d shows the evolution of the number of satisfied clauses (%) [top] and nodal voltages in cBRIM [bottom] while solving a 20-variable problem. The voltages settle at about 25 ns and the quantized voltages indeed satisfies the 3-SAT formula.

Figure 3c shows the timing diagram of using cBRIM. All the blue boxes represent von Neumann computation. Steps 2 and 3 are not required for cBRIM and hence, greyed out. In step 1, the given 3-SAT problem in Conjunctive Normal Form (CNF) is pre-processed via $O(3d_{\max} \cdot M + N)$ time von Neumann computation (where $d_{\max}$ is the maximum degree of the graph) to turn it into a binary formulation (Eq. 4). The formulation is organized as a matrix, and programmed into the Ising machine hardware column by column in step 4a. For linear and quadratic terms, the programmed values represent the coefficient of terms. For cubic terms, it identifies the variables to be multiplied together. The Ising machine is then annealed multiple times, each time storing the final state (step 4b). After the annealing process, the system returns all the binary state vectors to the von Neumann computer (step 5) which then checks each state for satisfiability (step 6). The Ising machine can also store snapshots of the spin vector in a run which can be used to get the evolution of number of satisfied clauses.

Semantic-aware randomization

A simple yet effective annealing schedule is to randomly flip a spin with a certain probability and gradually reduce this probability²². Such a problem-agnostic heuristic works well to escape local minima for problems that lack any structure. However, SAT is a well-studied problem whose structure is exploited by solver heuristics. Conflict Driven Clause Learning (CDCL) solvers learn new clauses using Boolean resolution to actively store information on the problem at hand, and Variable State Independent Decaying Sum (VSID) has been shown to exploit large scale problem structures³⁹. Stochastic Local Search (SLS) algorithms such as WalkSAT⁴⁰ include a veritable buffet of different heuristics for variable selection based on various scoring and ranking mechanisms. With this motivation, we propose a new heuristic.

Tanh-make-break (TMB) heuristic

The *tanh-make-break* (TMB) heuristic assigns probability, p_n to flip each variable x_n as follows:

$$p_n = \tanh(c_m \times \mathcal{M}_n) \cdot [1 - \tanh(c_b \times \mathcal{B}_n)] \quad (8)$$

where, $c_m > 0$ and $c_b > 0$ are parameters. Given an assignment of variables a , $\mathcal{M}_n$ represents the number of newly satisfied clauses after flipping a variable x_n (*make count*), while newly unsatisfied clauses contribute to $\mathcal{B}_n$ (*break count*). Our main motivation to use the tanh function are: First, it is easily implementable in hardware. Second, it directly resembles probability without any need for normalization. Third, it is non-linear as experimentally we found it to work better. Finding the most optimal function is left as future work. The main intuition for TMB is that, stochastically flipping nodes that satisfy more clauses makes the landscape navigation more efficient as we shall see later. An illustrative example is shown in the supplemental. A key point to note here is that, unlike SLS algorithms, TMB allows multiple nodes/variables to flip in parallel without any synchronization. This makes the hardware design much simpler.

To compute p_n , we need to compute $\mathcal{M}_n$ and $\mathcal{B}_n$. It turns out, the incoming electric current for each node in our system is related to its $\mathcal{M}_n$ and $\mathcal{B}_n$ as follows:

$$\frac{dv_n}{dt} = \alpha(1 - 2x_n) \cdot (\mathcal{M}_n - \mathcal{B}_n) \quad (9)$$

The derivation is shown in the Supplemental material. With this relation, we opt to compute $\mathcal{M}_n$ since it is much easier to obtain (refer to the supplemental for more details) and recover $\mathcal{B}_n$ from the incoming electric current in Eq. (9).

Figure 4 shows how we integrate TMB into our dynamical system. For each node, after every *TMB period* time, the heuristic gets the electric currents proportional to $\mathcal{M}_n$ and $\mathcal{B}_n$. This information is used to select the respective node to flip determined by the probability p_n as in Eq. (8). The selected nodes are then temporarily connected to V_{DD} /ground to charge/discharge the capacitor respectively. Note that, while TMB is in the works, the system dynamics keeps evolving. If the system found a satisfying solution before the cutoff time, then $\forall n (\mathcal{M}_n = 0)$. In this case, the system stops annealing and returns the spin vector (*latching*).

High-level analyses of Augmented Ising Machine for SAT (AIMS)

With the discussed architectural innovations: cubic interaction and TMB heuristic, we propose Augmented Ising Machine for SAT (AIMS). We tuned the parameters c_m and c_b in our TMB heuristic that works the best, in general, for all problems. Sensitivity analysis is discussed in the supplemental. All the results are obtained using numerical simulations with realistic device parameters, assuming a 45nm technology. AIMS is first compared against state-of-the-art software solvers and cBRIM. Note that, existing high-degree Ising machines have been either evaluated with tiny problems¹⁹ or with poor solution quality and no direct report of the solver time¹⁸. In contrast, cBRIM is able to find solutions with higher success rates and hence, is compared here. Figure 5a shows the geometric mean TTS for each benchmark suite with error bars and also the overall TTS for various solvers.

As we can observe, for these benchmarks, AIMS is consistently faster with a geometric mean speedup of about 2 and 3 orders of magnitude over software solvers WalkSAT⁴⁰ and KISSAT³⁰ respectively. The performance of the GPU-based SAT solver, ParaFrost⁴² lies between WalkSAT and KISSAT, thus it is slower than AIMS by about 517x. With contributions from our TMB heuristic, AIMS outperforms cBRIM by at least an order of magnitude. Some of the data for cBRIM are missing as it could not solve the problems in our tested annealing times. The horizontal dashed line shows the TTS of baseline BRIM solving the QUBO formulation of a 100-variable problem. The speed of AIMS compared to both cBRIM and BRIM emphasizes the importance of our proposed architectural innovations.

Now, we look at the execution times and success probabilities for various solvers on 500-variable uniform random and scale-free problems as shown in Fig. 5b. The results of WalkSAT and KISSAT are shown for reference. AnalogSAT⁴³ is a recently proposed GPU-based solver that accelerates a simulation of a dynamical system. We can observe that AIMS takes the least execution times of $\sim 100 \mu s$ on average while still maintaining high

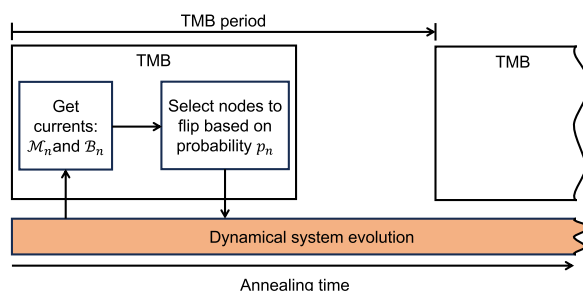


Figure 4. The high-level flow of how TMB works with our dynamical system.

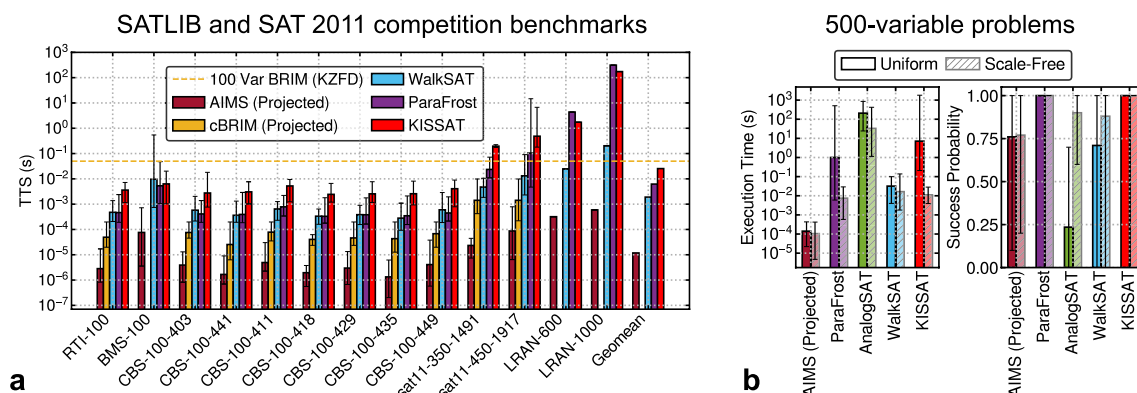


Figure 5. Performance comparison of AIMS against existing SAT solvers. (a) Comparing TTS of various solvers for benchmark suites from SATLIB⁴¹ and SAT 2011 competition. Each benchmark suite is named in this format: (Benchmark)–(Vars)–(Clauses (if stated)). (b) Plots showing the execution time [Left] and success probability [Right] for various solvers. All solvers run 20 problems each with uniform random and industrial-like scale-free distribution of 500 variables.

success probability of 0.75. For both the problem types, AnalogSAT performed the worst as it took the longest times with low success probability for uniform random problems. KISSAT slightly outperformed WalkSAT in scale-free problems which corroborates its success in industrial SAT instances. In terms of TTS, AIMS is faster than the better performing GPU-based solver, ParaFrost, by about 3 orders of magnitude for uniform random problems and by 27× for scale-free problems.

Apart from software solvers, we also compare AIMS with two other state-of-the-art hardware SAT solvers: FPGA-based HW AmoebaSAT⁴⁴ and Magnetic Tunnel Junction (MTJ)-based BRWSAT⁴⁵ as shown in Table 1. The benchmarks shown here are the only ones reported by the references and hence, can be directly compared. Based only on the overlapping benchmarks, AIMS has a geometric speedup of 7.4× and 10.3× over HW AmoebaSAT and BRWSAT, respectively. Note that HW AmoebaSAT scales rather poorly as TTS increases 3 orders from problem sizes of 50 variables to 225 variables.

Finally, using Cadence simulations, we estimate that a 500-variable AIMS chip will consume ~ 300 mW power and require ~ 13 × 13 mm² area. A detailed table of hardware parameters is shown in the supplemental. We believe that there is room for significant improvement for the various circuits used in this study.

Navigational efficiency

With such promising results demonstrated in the previous section, it is important to also look at how efficiently AIMS finds a solution. Figure 6 shows the geometric mean phase points visited to arrive at a solution by AIMS, cBRIM and WalkSAT. A phase point denotes a specific state of the quantized spin vector $\in \{0, 1\}^N$. Each such state traversed by AIMS in a run is counted as a phase point “visit”. Since KISSAT is a CDCL solver, the notion of a phase point is not well defined and hence is not shown. Note here that the number of phase points visited is also adjusted by solution probability as in TTS. AIMS is consistently more efficient than cBRIM as it could achieve a satisfying solution by visiting about 134× fewer phase points in general. Note that some of the benchmarks are not shown since cBRIM fails to find a solution for these, suggesting that it may require searching through many more phase points. WalkSAT still requires slightly fewer phase point visits than AIMS in general. However, the extremely fast dynamics of AIMS more than compensates for the larger phase point visits with

Benchmark	N	M	HW AmoebaSAT ⁴⁴	BRWSAT ⁴⁵	AIMS (projected)
uf50-0100	50	218	4.126	–	5.829
uf50-0410	50	218	4.360	–	1.329
uf50-0767	50	218	8.300	–	4.059
uf100-0285	100	430	356	–	73.81
uf150-0100	150	645	1832	–	73.16
uf225-028	225	960	3078	–	10.41
sat11-350-p11	350	1491	–	200	39.54
sat11-350-p23	350	1491	–	1289	44.56
sat11-350-p75	350	1491	–	280	22.81
sat11-350-p98	350	1491	–	50	7.389
sat11-450-p1	450	1917	–	80	6.448
sat11-450-p11	450	1917	–	1912	577.1
sat11-450-p14	450	1917	–	4650	786.2
sat11-450-p78	450	1917	–	353.1	8.082

Table 1. Comparison of TTS for various hardware solvers in units of μ s.

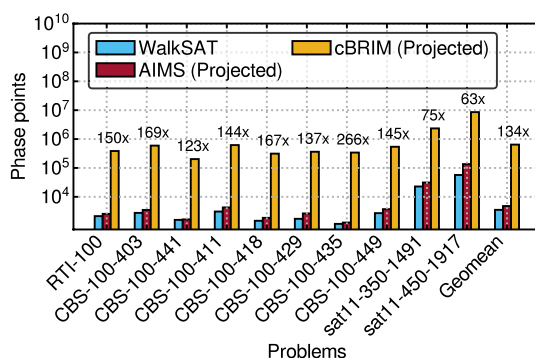


Figure 6. Geometric mean phase points visited by AIMS, cBRIM and WalkSAT for various benchmark suites from SATLIB and 2011 SAT competition. The labels show the improvement of AIMS over cBRIM.

an average flip-rate of 4250 flips/ μs compared to 6 flips/ μs for WalkSAT. This explains the observed speedup of AIMS over WalkSAT in Fig. 5.

Discussion

With diminishing speed of improvements for conventional computing systems, non-von Neumann systems such as Ising machines are receiving increased attention. While all COPs can be targeted by an Ising machine in theory, we have shown in this paper, using SAT as a case study, that there are still practical hurdles with such systems: high-degree interactions and efficient navigational heuristics.

Fortunately, these issues can be overcome with additional architectural support, at least for electronic Ising machines such as BRIM²². We have presented the design details to support cubic interaction natively in BRIM with 3 optimizations: ① using AND gate as a multiplier, ② exploiting SAT structure to still use $O(N^2)$ couplers and ③ using fixed resistors for the cubic couplers without much loss of generality.

More importantly, we have proposed a novel, semantic-aware annealing heuristic called TMB. To efficiently incorporate TMB into our dynamical system, we showed a relation between the difference ($\mathcal{M}_n - \mathcal{B}_n$) for each variable and its total incoming current. Then we derived an expression to easily compute $\mathcal{M}_n$ using analog circuits and extract $\mathcal{B}_n$ from the above mentioned relation. This reduced the extra circuit complexity and made the design more feasible. With the proposed architectural support, the resulting augmented Ising machine for SAT (AIMS) can achieve orders of magnitude speedup over state-of-the-art SAT solvers. Moreover, we also demonstrated that AIMS can find a solution by visiting significantly fewer phase points than cBRIM, thus emphasizing the importance of TMB.

Although we conducted some non-ideal simulations verified partly with Cadence, a real hardware-based study is required in the future. Overall, our study suggests that Ising machines offer a powerful substrate to explore novel non-von Neumann computing. At the same time, additional architectural support may be crucial to truly unlock their performance potential.

Methods

Ising model

The Ising model was originally used to describe the Hamiltonian (H) for a state of spins (σ_i) on some lattice. A modified version (Eq. 10) of the formula is more useful in the discussion of optimization problems.

$$H(\sigma) = - \sum_{(i,j)} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i = -\sigma^\top J \sigma - h^\top \sigma \quad (10)$$

It turns out, this optimization problem is NP-complete and all the Karp's original set of NP-complete problems have been transformed into the Ising model⁴⁶. A loose physical analogy is as follows. A system of spins ($\sigma_i \in \{-1, +1\}$) are subject to the influence of a set of parameters J_{ij} and h_i . J_{ij} describes the coupling between two spins (σ_i and σ_j), while h_i describes the influence of an external magnetic field on spin σ_i . Given this setup, the spins will naturally gravitate towards the lowest energy state (ground state). Technically, the probability of each state follows the Boltzmann distribution:

$$P(\sigma) \propto e^{-\frac{H(\sigma)}{T}} \quad (11)$$

Boolean satisfiability (SAT)

A generic SAT problem is about determining whether a propositional logical formula F is satisfiable for some mapping of variables to truth values¹⁷. F consists of boolean-valued variables acted on by unary logical operator (*negation*: $\neg$) or binary ones (*and*: $\wedge$; *or*: $\vee$; and *implications*: $\rightarrow$, $\iff$). For example, $F = x_1 \vee (\neg x_2) \wedge (x_3 \rightarrow x_4)$ would be a formula in propositional logic. The statement is typically expressed in Conjunctive Normal Form (CNF)¹⁷, i.e. a logical conjunction of separate *clauses* (e.g. $F = C_1 \wedge C_2 \wedge C_3$). Each clause is a logical disjunction of a number of *literals*. Literals are boolean variables or their negations (e.g. $C = x_1 \vee \neg x_2 \vee x_3$). Any formula in propositional logic can be converted to CNF in linear time by introducing new variables and clauses⁴⁷. When the number of literals in any clause is no more than k , the problem is called k -SAT. When $k \geq 3$, the problem is NP-complete. 3-SAT can be used to formulate any NP-complete problem in standard CNF, with SAT solvers as an efficient means of solution⁴⁸.

D-Wave QA results

We report the results on D-Wave's 2000Q system with optimized annealing times of 2 μs , 20 μs and 2000 μs for 5–7, 8–20, and 30–40 variable problems respectively. The results of the newer D-Wave Advantage system (5000 qubits) were worse for our experiments although it could map larger 3-SAT problems. This is consistent with previous studies which conjectured that for sparse problems like SAT, the large number of unused couplers in the *Pegasus* topology of D-Wave Advantage might lead to more noise^{49,50}.

Software-based solver results

In all the experiments, CPU software solvers like WalkSAT⁴⁰, KISSAT³⁰ and SA³² are natively executed on Intel Xeon Platinum 8268 CPU at 2.90 GHz. WalkSAT is run using the `-best` heuristic and cutoff flips of 500,000 with 20 retries. KISSAT is allowed to run with no cutoff time. GPU-based solvers (ParaFrost⁴² and AnalogSAT⁴³) are executed on a system with AMD Ryzen Threadripper PRO 3975WX CPU with 16 GB of 3200 MHz DDR4 RAM and an NVIDIA GeForce RTX 3090 @ 1.70GHz GPU.

Simulation of dynamical systems

The evolution of various dynamical systems like BRIM, cBRIM and AIMS is modeled by solving differential equations using 4th-order Runge-Kutta method in C++. The behavioral simulation is verified to be accurate upto 32 nodes when compared against detailed circuit simulations in Cadence.

Benchmarks

For SAT benchmarks, we use those publicly available from SATLIB⁴¹ and the 2011 SAT competition. Moreover, we generated “hard” uniform random instances of varying sizes with $M/N = 4.25$. Since it has been established that industrial problems exhibit *scale-free* behaviour⁵¹, we generated some difficult industrial-like SAT instances with 500-variables using scale-free distribution with $M/N = 3.29$ and power-law exponent of 2.935⁵¹. All problems are verified as satisfiable.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Received: 12 August 2023; Accepted: 14 December 2023

Published online: 21 December 2023

References

- Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science* 124–134. <https://doi.org/10.1109/SFCS.1994.365700> (1994).
- Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* 212–219 (1996).
- Wang, Y. & Krstic, P. S. Prospect of using Grover’s search in the noisy-intermediate-scale quantum-computer era. *Phys. Rev. A* <https://doi.org/10.1103/physreva.102.042609> (2020).
- Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. <https://doi.org/10.48550/ARXIV.1411.4028> (2014).
- Boulebnane, S., & Montanaro, A. Solving boolean satisfiability problems with the quantum approximate optimization algorithm. <https://doi.org/10.48550/ARXIV.2208.06909> (2022).
- Bharti, K. *et al.* Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004. <https://doi.org/10.1103/RevModPhys.94.015004> (2022).
- Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488. <https://doi.org/10.1007/bf02650179> (1982).
- Ising, E. Beitrag zur Theorie des Ferromagnetismus. *Z. Phys.* **31**, 253–258. <https://doi.org/10.1007/BF02980577> (1925).
- D-Wave. The first and only quantum computer built for business—d-wave systems.
- Inagaki, T. *et al.* A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
- Honjo, T. *et al.* 100,000-spin coherent Ising machine. *Sci. Adv.* **7**, eabh0952. <https://doi.org/10.1126/sciadv.abh0952> (2021).
- Takeda, Y. *et al.* Boltzmann sampling for an XY model using a non-degenerate optical parametric oscillator network. *Quantum Sci. Technol.* **3**, 014004. <https://doi.org/10.1088/2058-9565/aa923b> (2017).
- Wang, T. & Roychowdhury, J. O. I. M. *Oscillator-Based Ising Machines for Solving Combinatorial Optimisation Problems* **1903** 07163 (2019).
- Cai, F. *et al.* Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **3**, 409–418. <https://doi.org/10.1038/s41928-020-0436-6> (2020).
- Kaiser, J. & Datta, S. Probabilistic computing with p-bits. *Appl. Phys. Lett.* **119**, 150503. <https://doi.org/10.1063/5.0067927> (2021).
- Mohseni, N., McMahon, P. L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379. <https://doi.org/10.1038/s42254-022-00440-8> (2022).
- Biere, A., Heule, M., van Maaren, H. & Walsh, T. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications* (IOS Press, NLD) (2009).
- Bybee, C. *et al.* Efficient optimization with higher-order Ising machines. arXiv preprint [arXiv:2212.03426](https://arxiv.org/abs/2212.03426) (2022).
- Bashar, M. K., Lin, Z. & Shukla, N. Oscillator-inspired dynamical systems to solve boolean satisfiability. *IEEE J. Explor. Solid-State Comput. Dev. Circuits* **9**, 12–20 (2023).
- Kanao, T. & Goto, H. Simulated bifurcation for higher-order cost functions. *Appl. Phys. Express* **16**, 014501 (2023).
- Bashar, M. K. & Shukla, N. Designing Ising machines with higher order spin interactions and their application in solving combinatorial optimization. *Sci. Rep.* <https://doi.org/10.1038/s41598-023-36531-4> (2023).
- Afoakwa, R., Zhang, Y., Vengalam, U. K. R., Ignjatovic, Z. & Huang, M. Brim: Bistable resistively-coupled Ising machine. *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* 749–760. <https://doi.org/10.1109/HPCA51647.2021.00068> (2021).
- Sharma, A., Afoakwa, R., Ignjatovic, Z. & Huang, M. Increasing ising machine capacity with multi-chip architectures. In *Proceedings of the 49th Annual International Symposium on Computer Architecture, ISCA ’22*, 508–521, <https://doi.org/10.1145/3470496.3527414> (Association for Computing Machinery, New York, NY, USA, 2022).
- Zhang, Y., Vengalam, U. K. R., Sharma, A., Huang, M. & Ignjatovic, Z. Qubrim: A CMOS compatible resistively-coupled Ising machine with quantized nodal interactions. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design, ICCAD ’22*, <https://doi.org/10.1145/3508352.3549443> (Association for Computing Machinery, New York, NY, USA) (2022).
- Pan, Z. *et al.* Ising-traffic: Using ising machine learning to predict traffic congestion under uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37 9354–9363. <https://doi.org/10.1609/aaai.v37i8.26121> (2023).
- Yamaoka, M. *et al.* A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing. *IEEE J. Solid-State Circuits* **51**, 303–309 (2015).
- Rosenberg, I. G. Reduction of bivalent maximization to the quadratic case. *Cahiers du Centre d’Etudes de Recherche Operationnelle* **17**, 71–74 (1975).
- Kolmogorov, V. & Zabin, R. What energy functions can be minimized via graph cuts?. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 147–159. <https://doi.org/10.1109/TPAMI.2004.1262177> (2004).
- Freedman, D. & Drineas, P. Energy minimization via graph cuts: Settling what is possible. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 2, 939–946. <https://doi.org/10.1109/CVPR.2005.143> (2005).
- Biere, A. & Fleury, M. Gimsatul, ISASAT, KISSAT entering the sat competition 2022. In *Proceedings of SAT Competition 2022: Solver and Benchmark Descriptions* (2022).

31. Hamerly, R. *et al.* Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Sci. Adv.* **5**, eaau0823. <https://doi.org/10.1126/sciadv.aau0823> (2019).
32. Isakov, S., Zintchenko, I., Rønnow, T. & Troyer, M. Optimised simulated annealing for Ising spin glasses. *Comput. Phys. Commun.* **192**, 265–271. <https://doi.org/10.1016/j.cpc.2015.02.015> (2015).
33. Marques Silva, J. & Sakallah, K. Grasp-a new search algorithm for satisfiability. In *Proceedings of International Conference on Computer Aided Design* 220–227. <https://doi.org/10.1109/ICCAD.1996.569607> (1996).
34. The international sat competition web page (2022).
35. Schoning, T. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*. <https://doi.org/10.1109/sfcs.1999.814612> (IEEE Comput. Soc).
36. Scheder, D. & Steinberger, J. P. Ppsz for general k-sat: Making hertli's analysis simpler and 3-sat faster. In *Proceedings of the 32nd Computational Complexity Conference, CCC '17 (Schloss Dagstuhl–Leibniz–Zentrum fuer Informatik, Dagstuhl, DEU)* (2017).
37. D-WAVE. minorminer (2014).
38. Nishimori, H. *Statistical Physics of Spin Glasses and Information Processing* (Oxford University Press) (2001).
39. Liang, J.H., Ganesh, V., Zulkoski, E., Zaman, A. & Czarnecki, K. Understanding VSIDS branching heuristics in conflict-driven clause-learning SAT solvers. In *Hardware and Software: Verification and Testing: 11th International Haifa Verification Conference, HVC 2015, Haifa, Israel, November 17–19, 2015, Proceedings* 11 225–241 (Springer, 2015).
40. Kautz, H. Walksat (2018).
41. Satlib benchmarks (2022).
42. Osama, M., Wijs, A. & Biere, A. Sat solving with GPU accelerated inprocessing. In *Tools and Algorithms for the Construction and Analysis of Systems: 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27–April 1, 2021, Proceedings, Part I* 27 133–151 (Springer, 2021).
43. Molnár, F., Kharel, S. R., Hu, X. S. & Toroczkai, Z. Accelerating a continuous-time analog sat solver using GPUS. *Comput. Phys. Commun.* **256**, 107469. <https://doi.org/10.1016/j.cpc.2020.107469> (2020).
44. Hara, K., Takeuchi, N., Aono, M. & Hara-Azumi, Y. Amoeba-inspired stochastic hardware sat solver. In *20th International Symposium on Quality Electronic Design (ISQED)* 151–156. <https://doi.org/10.1109/ISQED.2019.8697729> (2019).
45. Shim, Y., Sengupta, A. & Roy, K. Biased random walk using stochastic switching of nanomagnets: Application to sat solver. *IEEE Trans. Electron. Dev.* **65**, 1617–1624. <https://doi.org/10.1109/TED.2018.2808232> (2018).
46. Lucas, A. Ising formulations of many np problems. *Front. Phys.* **2**, 5 (2014).
47. Tseitin, G. S. *On the Complexity of Derivation in Propositional Calculus* 466–483 (Springer, Berlin Heidelberg, Berlin, Heidelberg, 1983).
48. Cook, S. A. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71* (151–158). <https://doi.org/10.1145/800157.805047> (Association for Computing Machinery, New York, NY, USA) (1971).
49. Willsch, D. *et al.* Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems. *Quantum Inf. Process.* **21**, 141. <https://doi.org/10.1007/s11128-022-03476-y> (2022).
50. Calaza, C. D. G., Willsch, D. & Michielsen, K. Garden optimization problems for benchmarking quantum annealers. *Quantum Inf. Process.* <https://doi.org/10.1007/s11128-021-03226-6> (2021).
51. Friedrich, T., Krohmer, A., Rothenberger, R. & Sutton, A. Phase transitions for scale-free SAT formulas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31 (2017).

Author contributions

A.S. designed the experiments, analyzed the data and wrote majority of the paper. M.B. proposed TMB and did the initial testings. A.H. designed all the circuits. M.H. wrote the introduction and gave guidance on the high-level understanding.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-49966-6>.

Correspondence and requests for materials should be addressed to A.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023