Custom CMOS Ising Machine Based on Relaxed Burer-Monteiro-Zhang Heuristic

Aditya Shukla, Mikhail Erementchouk, and Pinaki Mazumder, *Fellow, IEEE*

***Abstract*—Determining the maximum cut of large graphs may require impractically long time, necessitating approximate algorithms and/or specialized computing platforms. A heuristic by Burer, Monteiro and Zhang for max-cut has not only been shown to be advantageous in many respects, but is also applicable to other NP-complete problems. From the perspective of accelerated computing, the heuristic's implementational challenge lies in its gradient-descent dynamics, which could be reduced to several *sinusoidal* kernel operations applied to each edge of the graph. We had previously established the theoretical underpinnings of a relaxed dynamical heuristic for max-cut similar to the one proposed by Burer et al. but suited for accelerated computing on custom analog CMOS. In this work, we present the first fully custom analog integrated circuit implementing the dynamics of our heuristic on 130-nm CMOS technology. In an era of increasing specificity of computing machines, our algorithm-circuit co-design, originally for max-cut, introduces a versatile approach applicable to a diverse set of practical large-scale NP-complete problems.**

***Index Terms*—Application-specific integrated circuits, burer-monteiro-zhang heuristics, CMOS, combinatorial optimization, goemans-williamson's algorithm, maximal cut, NP-complete, quadratic unconstrained binary optimization (QUBO), semi-definite programming.**

I. Introduction

COMBINATORIAL optimization finds its application in a vast majority of fields, such as commerce, resource allocation, semiconductor chip-design [1], and medicine [2]. A large proportion of these problems requires computing resources that exponentially scale with the number of variables, meaning that a solvable problem may become practically unsolvable by just doubling its size [3], [4]. This has motivated (1) specialized algorithms that provide the solution for either simplified problems, or answers with various degrees of approximation [5], [6], and (2) application-specific accelerators aimed at reducing the time-to-solution for large problems [7], [8], [9], [10].

Finding the maximum cut of a generic graph, or the max-cut problem, is a well-studied problem in the context of the design of approximate algorithms for NP-hard problems. To tackle a general max-cut problem's exponential complexity, several polynomial-time approximate algorithms exist that guarantee bound on the approximate solution [11], [12]. So far, the algorithms providing the tightest bounds are based on the semi-definite programming (SDP) relaxation [13]. However, these may become impractical for graphs with a large number of vertices as these involve operations on vectors with an equally large number of elements.

The heuristic by Burer, Monteiro and Zhang [14] (henceforth called BMZ heuristic) uses rank-2 SDP. Thanks to the greatly reduced dimensionality, this heuristic is much faster than the rank-N SDP. For max-cut, this heuristic has been shown to be one of the best performing ones [11]. Although the rank-2 SDP is not guaranteed to perform like the full-rank SDP (see [15] for some recent results in this direction), the BMZ heuristic provides good quality solutions with a more manageable computational resource scaling. Circut, which performs very well in comparison to other max-cut solvers, is based this heuristic [11].

For accelerated solving of combinatorial optimization problems, several dedicated efficient CMOS-based accelerators, simulating Ising (binary) spin-models, have been proposed in [16], [17], [18], [19] (review in [8]). While the matrix-vector multiplication heavy annealing algorithms have greatly benefited from hardware-related schemes, acceleration techniques for the BMZ heuristic have been limited. This originates from a relatively complex dynamics of the heuristic, which comprises a system of pair-wise non-linearly coupled real variables.

In [20], we proposed an almost linear dynamical model for solving max-cut that works on similar underlying principles as that of the BMZ heuristic, but adapted for the CMOS-based computing. We empirically demonstrated its performance as a max-cut solver in polynomial time and discussed the accompanying hardware requirements.

In this work, we present a fully custom CMOS accelerator for the relaxed rank-2 SDP dynamics, designed using 130 nm commercial technology. We co-design all levels of the system – algorithm, architecture and circuit. At the algorithm-level, rank-2 SDP dynamics is relaxed from the sinusoidal to triangular coupling, causing the kernel operation to be realizable using analog computing methodologies, with only a slight compromise in the solution quality. Computer architecture-wise, continuous variables are represented by analog voltages while an array of analog vertices supports all-to-all connectivity of graphs, and extends to post-SDP operations. The coupling arithmetic is conducted on a shared arithmetic unit while the

Manuscript received 5 July 2022; revised 14 February 2023; accepted 23 April 2023. Date of publication 19 July 2023; date of current version 6 September 2023. The work has been supported by the US National Science Foundation under Grant 1710940 and by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-16-1-0363. Recommended for acceptance by A. Rubio. *(Corresponding author: Aditya Shukla.)*

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48104 USA (e-mail: aditshuk@umich.edu; merement@umich.edu; mazum@umich.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TC.2023.3272278, provided by the authors.

Digital Object Identifier 10.1109/TC.2023.3272278

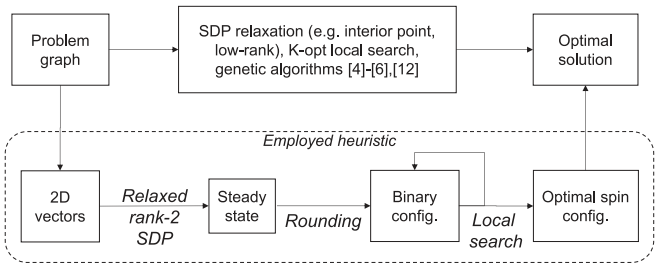


Fig. 1. This work *vis-à-vis* other approximate algorithms.

switch-capacitor-based accumulation mechanism updates the states. At the circuit/transistor-level, our design ensures that each vertex comprises the bare minimum transistors to save area. The capacitor is realized using upper metal-layers to serve as the analog memory, while a transconductance cell (henceforth called a g_M-cell) beneath enables its reading.

As the first prototype, we design an integrated circuit comprising 64 nodes – each with a pitch of 45 μm and an area 38 μm $\times$ 27 μm – and the power dominated by its static component at 40 μW per vertex. It replicates the gradient-descent dynamics of the relaxed rank-2 SDP, while the fully digital post-dynamics stages are outsourced to a general-purpose computer. Fig. 1 puts into perspective our heuristic versus the others, and also shows the constituent sub-steps.

This article is organized as follows. In Section II, we provide the necessary background for this work. In Section III, we give the description of the architecture and the pseudo-code that translate the original heuristics for implementation on the proposed hardware. In Section IV, we provide circuit-level details of all components of the system. Post-layout simulations are presented in Section V and conclusion in Section VI.

II. Background

A. Ising Models and Maximum Cut

Ising model is a set of coupled binary variables (commonly represented by ± 1) or spins (symbolically by $\uparrow\downarrow$). Each spin is coupled to a subset of other spins which makes one spins-state more energetically favorable than the other. Let the coupling between the spins of Ising model be represented by a undirected graph $\mathcal{G}$ with N vertices, M edges and adjacency matrix $\mathbf{A} = \{A_{ij}\}$. For spin-state $\boldsymbol{\sigma} = \{\sigma_i\}$ with $\sigma \in \{-1, +1\}$, the corresponding Hamiltonian H is

$$H(\boldsymbol{\sigma}) = \frac{1}{2}\sum_{i,j} A_{ij}\sigma_i\sigma_j. \tag{1}$$

The ground state of the Ising model is the spin-state yielding the minimum value of $H(\boldsymbol{\sigma})$.

For generic graphs, finding the ground state is an NP-complete problem [21], [22]. For example, all of Karp's NP-complete problems can be re-formulated as the ground state search problem of a specially constructed Ising models [23].

Finding the ground state of the Ising model based on $\mathcal{G}$ is equivalent to determining the max-cut of $\mathcal{G}$. Indeed, any spin-state partitions the vertices into two sub-sets: one with only positive spins and another with only negative spins. The corresponding cut-size is defined as the number (or the net weight) of the edges from one partition to the other:

$$\begin{aligned}\chi_1(\boldsymbol{\sigma}) &= \frac{1}{4}\sum_{i,j} A_{ij}(1-\sigma_i\sigma_j)\\ &= \frac{M}{2} - \frac{1}{2}H(\boldsymbol{\sigma}).\end{aligned} \tag{2}$$

Thus the minimum of $H(\boldsymbol{\sigma})$ corresponds to the maximum of $\chi_1(\boldsymbol{\sigma})$ and vice-versa. Alternatively, the max-cut problem can be presented in terms of a $N \times N$ positive semi-definite matrix $\mathbf{S}$:

$$\begin{aligned} C_{\mathcal{G}} = \max_{\mathbf{S}} \quad & \frac{M}{2} - \frac{1}{4}(\mathbf{A}\cdot\mathbf{S}) \\ \text{s.t.} \quad & \text{rank}(\mathbf{S}) = 1 \\ & \text{diag}(\mathbf{S}) = \{1\}^N \\ & \mathbf{S} \succeq 0 \end{aligned}, \tag{3}$$

where $\mathbf{A}\cdot\mathbf{S} = \sum_{i,j} A_{ij}S_{ij}$ and the last condition denotes positive semi-definiteness. The spin-state $\boldsymbol{\sigma}$ and the positive semi-definite matrix $\mathbf{S}$ are related by $\mathbf{S} = \boldsymbol{\sigma}\boldsymbol{\sigma}^T$.

B. Max-Cut Via Relaxed Burer-Monteiro-Zhang Heuristics

The NP-hardness of the max-cut problem prompted the development of algorithms that guarantee a lower-bound on their approximate solution in polynomial time [24], [25], [26], [27]. Among these, the algorithm by Goemans and Williamson [27] provides the tightest bound on the expected solution. Their algorithm has two stages. The first stage replaces each of the N spins, σ_i, with N-dimensional unit vector $\vec{s}_i$ and the product of spins with their dot-product. Then, an analogue of the cut χ_n is defined

$$\chi_n(\vec{\boldsymbol{s}}) = \frac{1}{4}\sum_{i,j} A_{ij}(1-\vec{s}_i.\vec{s}_j). \tag{4}$$

For $\mathbf{S} = \{\vec{s}_i.\vec{s}_j\}$, the following rank-$N$ SDP is solved:

$$\begin{aligned} C_{\mathcal{G}} = \arg\max_{\mathbf{S}} \quad & \frac{M}{2} - \frac{1}{4}(\mathbf{A}\cdot\mathbf{S}) \\ \text{s.t.} \quad & \text{diag}(\mathbf{S}) = \{1\}^N \\ & \mathbf{S} \succeq 0 \end{aligned}. \tag{5}$$

The solution vectors of the SDP ($\vec{s}_i$), distributed over the N-dimensional unit sphere, are mapped to +1 or -1 by comparing their orientation w.r.t. a random hyperplane through the origin. The cut so obtained was shown to be within 87% of the maximum cut for random graphs with non-negative weights.

Burer et al. [14] proposed to use rank-2 SDP, which is equivalent to limiting the dimensionality of $\vec{s}$ to two in (4). Each two-dimensional unit vector is represented by polar coordinate θ, s.t. $\vec{s} = [\cos(\theta), \sin(\theta)]$. The rank-2 cut analogue χ_2 is defined as

$$\chi_2(\boldsymbol{\theta}) = \frac{1}{4}\sum_{i,j} A_{ij}(1-\cos(\theta_i-\theta_j)), \tag{6}$$

where $\boldsymbol{\theta} = \{\theta_1, \theta_2 \ldots \theta_n\}$. This heuristic was implemented in the max-cut solver Circut, where the dynamical realization of

rank-2 SDP was followed by finding the optimal rounding and post-processing based on 1-opt and 2-opt local search.

The dynamical realization of SDP ensures that $\chi_2(\boldsymbol{\theta})$ monotonously increases with time. By updating $\boldsymbol{\theta}$ according to $\dot{\boldsymbol{\theta}} = \nabla\chi_2(\boldsymbol{\theta})$, the equations of motion governing the gradient descent of the system are:

$$\dot{\theta}_i = \sum_j A_{ij} \sin(\theta_i - \theta_j). \tag{7}$$

It can be shown [14] that if the stable steady state $\boldsymbol{\theta}'$ is such that $\theta'_i - \theta'_j = k_{ij}\pi$, where k_{ij} is an integer, then $\boldsymbol{\theta}'$ represents the maximum cut. However, generally the elements of $\boldsymbol{\theta}'$ are distributed across $[0, 2\pi)$, and consequently each θ_i (now one-dimensional) is rounded by comparing its orientation with a reference coordinate. Specifically, the following equation may be used to round $\boldsymbol{\theta}'$ to $\boldsymbol{\sigma}'$:

$$\boldsymbol{\sigma}'(\theta_Y) = \text{sgn}(\sin(\boldsymbol{\theta}' - \theta_Y)), \tag{8}$$

where θ_Y is the reference coordinate. In practice, several θ_Y are chosen from the range $[0, 2\pi)$. For each $\boldsymbol{\sigma}'(\theta_Y)$, the corresponding cut is evaluated using (2). The configuration leading to the largest cut is selected as the solution of rounding. During post-processing, a local search for the maximal cut is performed, ensuring that the cut cannot be increased by inverting any single spin or a pair of spins.

In contrast with the Goemans and Williamson's rank-N algorithm, rank-2 SDP-based heuristic does not have proven approximation guarantee. The landscape described by $\chi_2(\boldsymbol{\theta})$ is non-convex and the solution's quality depends on the quality of the minima of the Hamiltonian. Nevertheless, the algorithm was empirically shown to obtain competitive results in polynomial time [11].

C. Relaxing BMZ Heuristic to an Almost Linear Dynamics

The dynamical rank-2 SDP of the BMZ heuristic can be viewed as a system of vertices with sinusoidal coupling: each vertex i has a dynamical state θ_i that depends non-linearly on states of the other vertices. An exact realization of the nodal coupling dynamics requires computing sinusoidal function over multiple periods. Since there is a separate sine-evaluation corresponding to each edge, this computational effort may greatly shoot up for large and/or dense graphs. Instead, we consider a triangular coupling function, as a simpler piece-wise linear approximation of the sine. This reduces a complex non-linear coupling function to one composed of basic arithmetic operations - pair-wise additions and sign inversions, which are directly realizable using analog computing methods.

In (6), if the cosine cut-counting function is replaced with a more general but periodic Φ and the state vector by $\boldsymbol{v}$, then the new cut χ_Φ is,

$$\chi_\Phi(\boldsymbol{v}) = \frac{1}{4}\sum_{i,j} A_{ij}(1 - \Phi(v_i - v_j)) \tag{9}$$

and correspondingly, the new dynamical equations are

$$\dot{v}_i = \sum_j A_{ij}\phi(v_i - v_j), \tag{10}$$

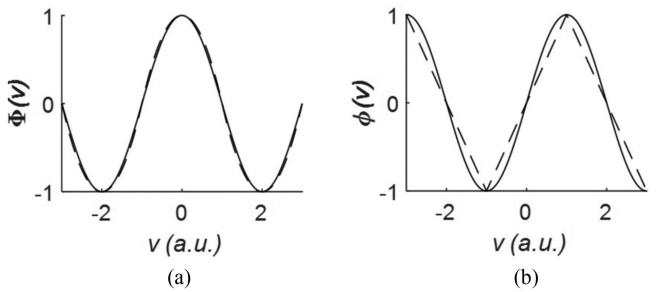


Fig. 2. Comparison of (a) the Hamiltonian kernel $\Phi(v)$ for the rank-2 relaxation (solid line) and the triangular model (dashed line) (b) the dynamic coupling function $\phi(v)$.

Algorithm 1: Nodal Coupling Dynamics (Input: $\mathbf{A}$; Output: $\mathbf{x}$).

1: **for** $i \in \{1, 2 \ldots N\}$ **do**
2: $x_i \leftarrow \mathcal{N}(0, \gamma P)$ ▷ Normally distributed
3: **end for**
4: **while** $t < T_{max}$ **do**
5: **for** $i \in \{1, 2 \ldots N\}$ **do**
6: $S \leftarrow \sum_j A_{i,j}\phi(x_i - x_j)$
7: $x_i(t+1) \leftarrow x_i(t) + \eta S$ ▷ η is time-discretization factor
8: **end for**
9: $t \leftarrow t + 1$
10: **end while**

where $\phi(v) = -\frac{1}{2}\frac{\Phi(v)}{dv}$. For a piece-wise linear coupling function, Φ is a periodic function with period P and $\Phi(kP/2) = (-1)^k$:

$$\Phi(v) = \begin{cases} 1 - v^2, & v \in \left(-\frac{P}{4}, \frac{P}{4}\right] \\ \left(|v| - \frac{P}{2}\right)^2 - 1, & |v| \in \left(\frac{P}{4}, \frac{P}{2}\right] \end{cases}, \tag{11}$$

so that

$$\phi(v) = \begin{cases} -v, & v \in \left(-\frac{P}{4}, \frac{P}{4}\right] \\ v - \frac{P}{2}, & v \in \left(\frac{P}{4}, \frac{3P}{4}\right] \end{cases}. \tag{12}$$

Fig. 2 compares Φ and ϕ with the original ones for $P = 4$ (a.u.).

We addressed the questions regarding the capability of the proposed heuristics to approximately solve the intended problem in polynomial time in [20]. The integrality gap [27] due to the random rounding following the new dynamics was shown to be about 85%, versus 87% for the full rank SDP.

The detailed steps of the proposed heuristic are provided in Algorithms 1-3. In Algorithm 1, the N analog spins are first initialized to normally distributed values after which the states are let to evolve for T_{max} time-steps. In Algorithm 2, K uniformly distributed rounding centers are chosen. For each choice, the analog spins are rounded to ± 1 and the cut evaluated. The algorithms then finds the the largest cut and corresponding spin-state. In Algorithm 3, the spin-state is updated according to the 1-opt local search rule.

In Fig. 3, we demonstrate our heuristic on an example graph with 25 vertices, shown in Fig. 3(a). Fig. 3(b) plots the states of vertices as they evolve following (10) with the period of ϕ set to 2. Fig. 3(c) and (d) plot the initial and final states as phasors

Algorithm 2: Randomized Rounding (Input: $\mathbf{x}$, K; Output: $\boldsymbol{\sigma}$).

for $i \in \{1, 2 \ldots K\}$ **do**
  $y_i \leftarrow \mathcal{U}(-P/2, P/2)$
end for
$C_{max} \leftarrow 0$
$y_{max} \leftarrow y_1$
for $i \in \{1, 2 \ldots K\}$ **do**
  for $j \in \{1, 2 \ldots N\}$ **do**
    $\sigma_j \leftarrow \text{sgn}(\phi(x_j - y_i))$
  end for
  $C \leftarrow \frac{M}{2} - \sum_m \sum_n A_{m,n} \frac{\sigma_m \sigma_n}{4}$
  if $C > C_{max}$ **then**
    $C_{max} \leftarrow C$
    $y_{max} \leftarrow y_i$
  end if
  for $j \in \{1, 2 \ldots N\}$ **do**
    $\sigma_j \leftarrow \text{sgn}(\phi(x_j - y_{max}))$
  end for
end for

Algorithm 3: Local Search (Input: $\boldsymbol{\sigma}$; Output: $\boldsymbol{\sigma}$).

while $I < I_{max}$ **do**
  for $i \in \{1, 2, \ldots N\}$ **do**
    $S \leftarrow \sum_j A_{i,j} \sigma_i \sigma_j$
    if $S < 0$ **then**
      $\sigma_i \leftarrow -\sigma_i$
    end if
  end for
  $I \leftarrow I + 1$
end while

with period of 2. The solid black diameter at angle α rounds continuous states to binary spins, ± 1. The mapping clearly depends on α and this dependence is plotted in Fig. 3(e) and (f) for the initial and final states. Fig. 3(f) shows the cut evolution over time. It is divided into two stages: the first shows a coarse optimization from the relaxed heuristics and the second shows a finer increment in the cut from the local search procedure.

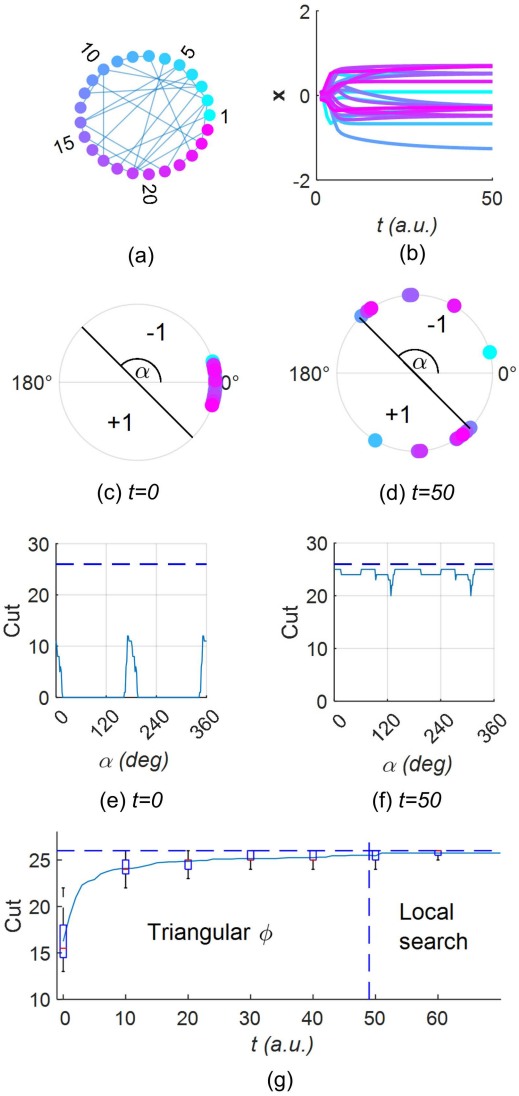


Fig. 3. Illustration of the relaxed BMZ heuristic on an example graph. (a) The example graph with 25 vertices. (b) State evolution based on the relaxed dynamics with period of ϕ to 2 for 50 time-steps. (c) and (d) Polar plot of the states at $T = 0$ and $T = 50$. (e) and (f) Cut versus polar angle at $T = 0$ and $T = 50$, and dashed line indicating the max-cut. (g) Average cut evolution from 20 random initial conditions as obtained using the relaxed heuristic and the local search procedure.

III. Ising Machine Based on the BMZ Heuristic

A. Motivation

Significant progress has been made towards accelerating the ground-state search problem of Ising models and more generally, solving combinatorial optimization problems via dynamical Ising-like models [8]. Binary Ising machines, in most cases, are designed to parallelize the thermal annealing of a large number of semi-independent Ising models. Most binary spin annealing systems are directly challenged by the need for an extremely large samples of the spin-states. On the other hand, analog models are hard to implement in a continuous time fashion due to sensitivity to analog parameters, difficulty in replicating their complex dynamics exactly using analog components, and dependence on alternatives to the widely-used CMOS devices.

The BMZ heurisitic was shown to give competitive max-cut results for artificially constructed graphs among 26 other heurisitcs in [11]. So far, to the best our knowledge, no hardware Ising machines (or generally, hardware combinatorial optimizers) based on the heuristic have been reported. The polynomial scaling and an overall reduced run-time of the relaxed heuristic due to the triangular approximation was demonstrated on a general-purpose computer in [20]. There, the max-cut values for graphs in G-set were found to be only slightly (about 2%) reduced than the max-cut obtained from Circut. With higher level of customization, one expects better scaling performance than achievable by a general-purpose computer.

TABLE I
RELAXED HEURISTIC'S CONSTITUENT OPERATIONS

		Algebraic expression	Input type	Output type
Nodal coupling dynamics		$\sum_j A_{ij}\phi(x_i - x_j)$	$\mathbb{R}^N$	$\mathbb{R}^N$
Binarization	Thresholding	$\text{sgn}(\phi(x_i - y)), \forall i$		$\{0,1\}^N$
	Cut evaluation	$\sum_i \sigma_i \sum_j A_{ij}\sigma_j$	$\{0,1\}^N$	
Local search		$\text{sgn}\left(\sigma_i \sum_j A_{ij}\sigma_j\right), \forall i$		

B. Kernel Operations of the Heuristic

Table I summarizes the key computational operations in the algorithm presented in Section II-C and identifies the corresponding input and output types. The table reveals two key composite operations that need to be efficiently conducted by an accelerator. The first operation is the modulo, defined for any real a and $b\,(\neq 0)$ as $\text{mod}(a,b) = a - |b|\lfloor a/|b|\rfloor$, where $\lfloor x \rfloor$ is the largest integer less than, or equal to x. The function ϕ in (12) may be expressed as the modulo with respect to ϕ's period P, followed by conditional sign-inversions. Binarization of analog spins requires a modulo of the analog state with respect to P, followed by 1-bit quantization.

The second operation is the multiplication-and-accumulation (MAC). Both cut-evaluation and local search steps involve the product of spin-vector with the adjacency matrix. The vector output determines a binary spin vector either for the binarization or for Hamiltonian reducing spin-flips.

C. Analog-Digital Mixed-Mode Computation

In this work, we base operations discussed above around a mix of analog-digital computing methodologies for two reasons. First, MAC is efficiently accelerated through a simultaneous multi-element product and sum [28], where the sub-operations of the MAC (usually a product of time-variant vector with time-invariant ones) are spatially unfolded and simultaneously conducted. Each analog/digital variable input of the MAC supplies a proportional current signal onto an accumulating wire/bus and thus physical movement of data is avoided. Higher the number of simultaneous summations, more efficient is the accumulation operation via unfolding. Such methods have been extensively studied for applications involving artificial neural networks and signal-processing [29].

Second, the proposed heuristic involves the evolution of analog variables via a series of accumulations conducted over time. Charge, as an analog variable, can be stored and naturally accumulated on a capacitor, within the limits of supply voltage and leakage [30]. Hence a capacitor can serve as a stationary analog memory element to store the spin-state. Whereas, in a purely digital memory, updating states without the movement of data would require adders in each vertex. A comparative study of digital versus analog approach (and the study other potential architectures for this heuristic), however, is beyond the scope of this paper.

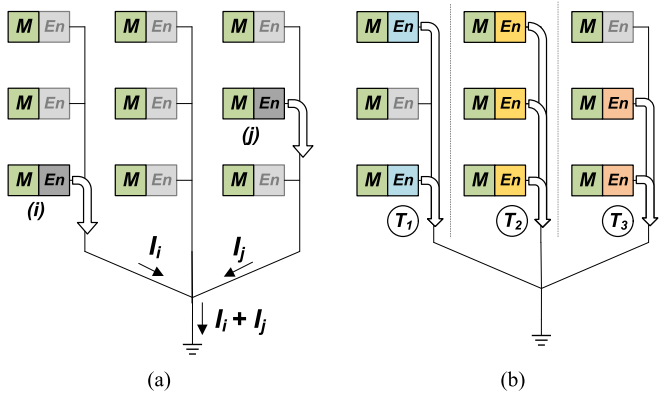


Fig. 4. Illustration of the organization of operations on 3×3 vertex array. (a) Evaluating modulo for computing $\phi(v_i - v_j)$. (b) Temporally multiplexed multiply-accumulate. Vertices' being read simultaneously have the same color.

D. Organization of Operations

In this work, analog memory cells are in one-to-one correspondence with vertices of the graph. Each cell comprises a capacitor-based memory and means to communicate with the computing element of the system. A shared current-bus enables a multi-vertex read via superposition of individual currents and establishes connection with the ϕ computing unit. Such sharing of the bus, on the one hand limits the speed of the nodal coupling dynamics, on the other, enhances the speeds of the more computationally linear cut-evaluation and local search. Fig. 4(a) depicts the modulo operation on a pair of vertices i and j selected from array of vertices. Fig. 4(b) shows the time-multiplexed MAC operation, divided into multiple column-wise simpler accumulations due to restrictions on the selectability of the vertices. Fig. 5 illustrates the organization of the N vertex array along with a read-write signal modulator to read and update vertex states, an external adjacency list/memory and global controller that manages the operations occurring in all the other blocks in the system.

E. Implementing Nodal Coupling Dynamics on the Proposed System

Fig. 5 shows the implemented computational system of vertices and the read-write modulator in the architectural schematic. The entire system of equations in (10) is realized by processing one vertex, one edge at a time. Each such processing is essentially divided into two phases: reading and writing. During the

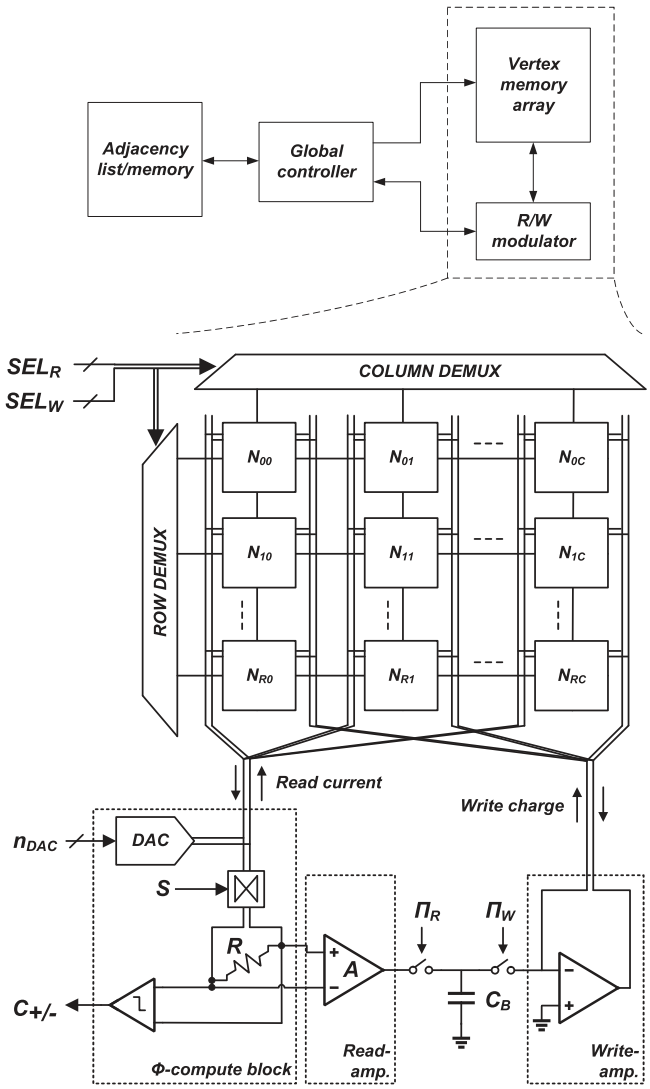


Fig. 5. High-level organization comprising the vertex memory array, computational unit, global controller and graph's adjacency memory with internals of the vertex-array and computational unit. $SEL_{R,W}$ are the vertex selection bits, n_{DAC} is the DAC's digital input, $C_{+/-}$ is the comparator's output, $\Pi_{R,W}$ are read/write switch's controlling signals, and S is read-bus inversion input.

reading phase vertices are simultaneously read by superposing multiple read currents onto a voltage-pinned current carrying bus. The net current is stepped up/down by superposing digitized current from a digital-to-analog converter (DAC). This is followed by sign-inversions from a 2-input and 2-output (2:2) multiplexer, to get a current proportional to ϕ. During the writing phase, a voltage proportional to ϕ is buffered onto capacitor C_B. The charge so stored is sent up to the updated vertex, through a separate write-bus, leading to the evolution of the state by long-term accumulation.

Consider the array of vertices in Fig. 5. Let vertex i be chosen for updating, based on an adjacent vertex j. It's updated by first selecting the vertices via the row and column selection multiplexers and closing the read-switch R or asserting Π_R, while keeping the common-mode bus voltage pinned. This gives an output differential current through the read bus

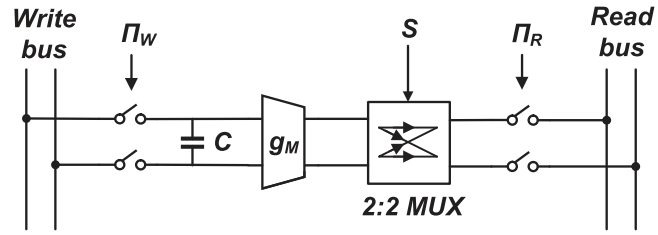


Fig. 6. Vertex block-diagram; *Key:* Π_W – write-switch control signal, Π_R – read-switch control signal, S – sign-inversion signal.

equal to

$$I_i - I_j = g_M v_{c,i} - g_M v_{c,j} = g_M (v_{c,i} - v_{c,j}) \tag{13}$$

The sign of differential current I_j above is reversed using 2:2 multiplexer local to the vertex, which reverses the current direction by exchanging the differential components. If the DAC output is represented by $n_{DAC} I_0$, then it is adjusted so that the net current lies within the peak values of ϕ:

$$-\frac{I_0}{2} < I_i - I_j - n_{DAC} I_0 < \frac{I_0}{2}. \tag{14}$$

Finally, the input to the multiplexer S is set to n_{DAC} to give a net modulated current

$$\phi(v_i - v_j) = (-1)^{n_{DAC}} \left(g_M (v_i - v_j) - n_{DAC} I_0 \right). \tag{15}$$

For writing, the moment Π_W is asserted, the voltage on the read-bus stored in C_B is pushed to the vertex being updated, akin to the standard switched-capacitor accumulator. The change in $v_{c,i}$ is given by

$$\delta v_{c,i} = \frac{C_B}{C} AR\phi(v_{c,i} - v_{c,j}), \tag{16}$$

where C is vertex's capacitance (Fig. 6), A is the voltage gain from read current to C_B and R is the net resistance seen the by the read-current bus (latter two shown in Fig. 5). Similarly, the other vertices are updated in sequence. Each such cycle of state updates, for all vertices, constitutes a time-step.

Algorithms 4–6 (listed in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TC.2023.3272278) provide detailed steps for realizing the proposed heuristics on the system depicted in Fig. 5. The algorithms are suited for sequential code-blocks in a hardware-description language (e.g., Verilog), as these involve reading and asserting real-time signals. All key input and output signals, referenced in the Algorithms 4–6 are depicted in Fig. 5. Two key sub-routines are repeatedly used through the algorithms – COMPUTE-ϕ and READ-WRITE, the exact sequence of steps of which are provided in Appendix A.1, available in the online supplemental material.

The algorithm for the nodal coupling dynamics is provided in Algorithm 4. The execution time for the nodal coupling dynamics scales as $O(M)$, where M is the number of edges. Though a vertex-stationary array realizes the dynamics of (12), a shared bus restricts the number of state updates in a given period of time, as only one edge is processed at a time.

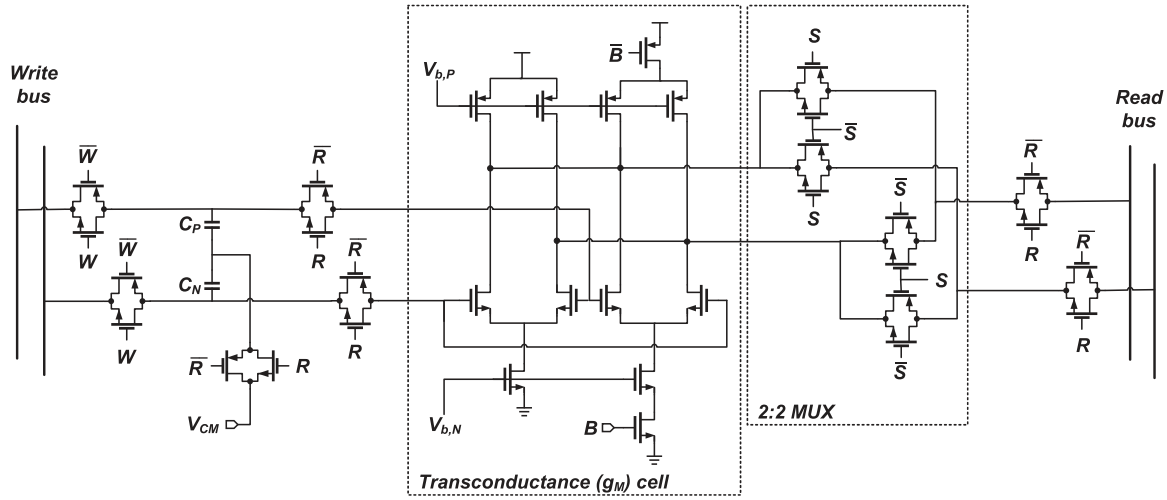


Fig. 7. Vertex schematic; logic transistors are not shown.

F. Extending the System to Incorporate Binarization and Local Search

As the binarization and local search components are primarily composed of binary variables and have relaxed circuit-related requirements than the purely analog nodal coupling dynamics, these will be incorporated in the future designs. Detailed algorithms for binarization with K rounding centers and cut-evaluation are provided in Algorithms 5 and 6, respectively in Appendices A.3 and A.4. We skip the description of the local search step as it essentially involves the same steps as cut-evaluation. The additional step is that the sum $\sum A_{ij}\sigma_j$ is evaluated for each vertex i. Depending upon the relation of the sign of net charge thus accumulated in C_B with the spin of the vertex i, the spin is flipped or preserved.

IV. Circuit-Level Details

A. Vertex

The vertex is primarily a capacitor-based analog memory with a differential g_M-cell that converts the capacitor's voltage to a proportional current as the output of a read. Additionally, the vertex has a 2:2 multiplexer to invert the sign of the current, and a minimal logic to select itself for read/write. Fig. 6 depicts the block diagram of the proposed vertex, with the capacitor C, and the other components accordingly labelled. The detailed schematic is shown in Fig. 7.

The analog memory is implemented using metal-insulator-metal (MIM) capacitor with a capacitance of 100 fF. This allows us to place the capacitor above the substrate containing the actual transistors, thus saving area. The capacitor's minimum allowed size depends on (1) ability to undergo writing cycles without loosing substantial existing charge, and (2) the size of the CMOS components below, which it should not significantly exceed. To set the common mode of the state voltages to a pre-determined value, we split the capacitor in a series of two identical C_P and C_N, and assert their common node during every read-cycle.

The g_M-cell (Figs. 6 and 7 converts the capacitor's analog charge to a proportional current. A differential amplifier with a tail current-source ensures that the transconductance is largely independent of input common-mode voltage. Larger transistors lead to less variability in transistor threshold, and a more uniform g_M spatially across the array. The g_M-cell's size also determines the minimum time needed to read the states. A faster cell would mean faster reads, and a faster overall chip operation. Therefore, we used big transistors for the g_M-cell ($W = L = 15\lambda$). Despite the static power consumption, the g_M-cell is maintained in its DC conducting state as the vertex read-time was found to be smaller than its turn-on time.

All the analog switches of the cell are implemented using transmission gates. The design requires that all the bus voltages be between 0 and the V_{DD}, even when writing. Read-enable switches turn on the g_M-cell's current output to the current bus. Write-enable switches enable charge from the write bus to pass through. A 2:2 multiplexer serves as an output sign-selection switch for inverting the output differential current.

A minimally sized logic determines the following local signals from the global row and column signals: (1) select-enable of the vertex, (2) g_M-cell's one-bit weight, (3) sign of the output current, and (4) binary spin-state.

The layout of the vertex' integrated circuit with all the discussed components is shown in Fig. 8(a). The majority of the substrate's area is occupied by the g_M-cell, followed by the logic gates and then write-enable switches. An equal area is assigned to the MIM capacitor, as shown in the orthographic view of a separated upper metal layers and the lower layers in Fig. 8(b).

B. Read-Write Amplifier

The read-write amplifier buffers the net read-bus voltage onto the buffer capacitor C_B and pushes the charge thus stored to the updated vertex. This block may be divided into two components: read-buffer and write-buffer (Fig. 9). The read-buffer is a differential amplifier with a current-mirror load [30] and has a small but linear gain. Extra sinking loads are used for operating

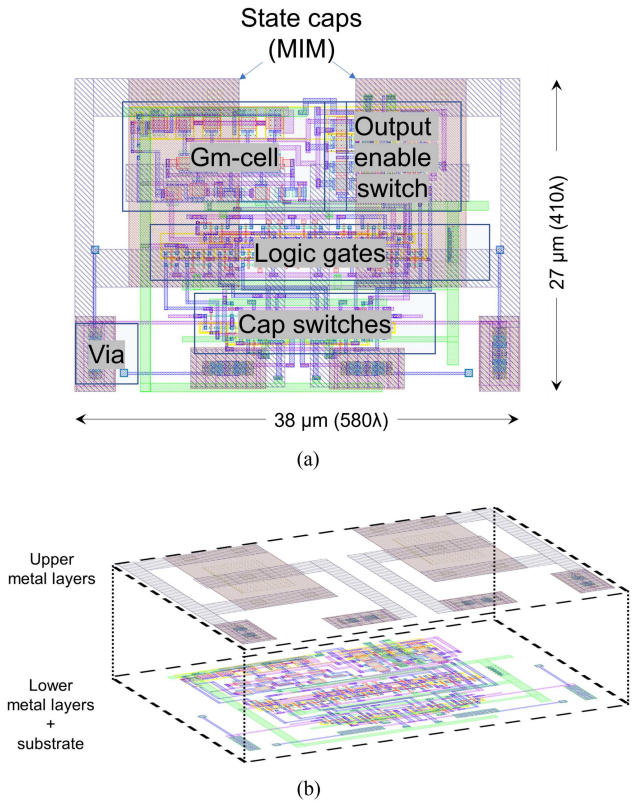


Fig. 8. Vertex layout. (a) Occupancy by various portions of the vertex. (b) Layout visualized in 3D to differentiate the state capacitors from the lower layers.

point's adjustment, setting it closer to $V_{DD}/2$. C_B stores the read-bus voltage and supplies its charge to the capacitor of the vertex being written onto. Since the peak increment in the state is generally smaller than the maximum state voltage, C_B is smaller than C_P and C_N. For this reason, we use lower metals layers for C_B.

The write-stage is essentially a high gain amplifier with differential input and single-ended output (Figs. 5 and 9). Its primary gain stage is a current-mirror loaded differential amplifier [30]. We design its gain to be large enough so that the amplifier can push charges reliably over the long write-bus, but not so much as to cause any oscillatory side-reaction due to positive feedback. Its secondary/output stage is an inverting amplifier with resistive load to limit the gain, provide high voltage swing, and predictably set the output operating point. Ideally, the charge increments provided by write-stage must be homogeneous with respect to the read-bus voltage and must cause zero increments for zero inputs. This trend may be easily disrupted post-fabrication due to the single-endedness of its input (the voltage across C_B). To limit any such non-homogeneity, we *balance* the output amplifier with respect to its input. This is done using a three-step process once during the entire IC's operation. First, we remove any differential input to the read bus, or effectively ensuring $V_{i,N} = V_{i,P}$ (Fig. 9). Then we store the voltage across C_B onto C' by asserting Π_o. Lastly, all the vertex capacitors (C) are simultaneously preset to the output of the write stage, $V_{o,N} - V_{o,P}$.

C. Other Components

The DAC plays a critical role in the realization of the heuristics by digitally stepping up/down the read-bus current. It is used in modulo/ϕ realization, random initialization of vertices and incrementally shifting the rounding center during binarization. We used a 5-bit DAC, which provides enough separation between the zeroes assuming a full (0 to V_{DD}) input range for ϕ. To realize DAC, we re-use the g_M-cell of the vertex with the multiplicities 1,2,4,8 and 16. This allows a direct control over the zeros of ϕ, e.g., if the first zero is to be placed at 0.1 V, one could set the input to the DAC's g_M cell to 0.1 V. By tuning the input voltage of the g_M-cell, one may also scale the current-step of the DAC, and alter the number of bits.

The comparator was implemented using a 5-stage high gain differential amplifier, with an absolute voltage gain of about 10^3. Thus, it could compare small but frequently occurring differential voltages of 1 mV. Read-bus pinning circuit (schematic in Appendix B, available in the online supplemental material), pins the common mode read-bus voltage to $V_{DD}/2$. It uses the principle of common-mode feedback, i.e., it feeds back current into the bus depending on the separation from its desired value of $V_{DD}/2$. Lastly, the sign-inverter is essentially a 2:2 multiplexer, similar to the one discussed in Section IV-A. The layout of the integrated circuit with 64 vertices is shown in Fig. 10.

V. Validation and Results

A. Realization of ϕ

The nodal coupling dynamics is fully analog and in comparison to other parts of the heuristics – rounding, cut-evaluation and local-search – it is most susceptible to non-idealities of analog computation. The computational errors may primarily stem from time-invariant causes such as parasitic impedances, non-linearity of components, devices mismatches, and time-variant ones such as noise.

The dynamical part comprises reads in proportion to the dynamical variables and writes in proportion to ϕ. Extent of accuracy, with which ϕ is realized, depends on how accurate the reading, DAC's application and writing processes are on the vertex array. Hence, we quantitatively analyze the implementation accuracy of ϕ by simulating several dummy writes on the post-parasitic extraction model of the layout of the circuit. The validation procedure may be split into two sub-parts - ϕ-computation and the proportional writing-process. The ϕ-computation involves the state-reads and the DAC application. The latter involves sensing the read-bus voltage and sending the charge to the vertex. To establish the validity of the overall ϕ, we separately test the two sub-processes.

Fig. 11(a) plots the ϕ-computation characteristics as obtained on the layout simulations. We sweep the state voltage (X-axis) and observe the final differential read-bus voltage (Y-axis) once the DAC stops counting. We see that unlike the intended ϕ of (12), peaks systematically decrease away from zero or equivalently, the zeros are non-uniformly positioned. Also segments after the fourth zero have observable curvature. The coupling

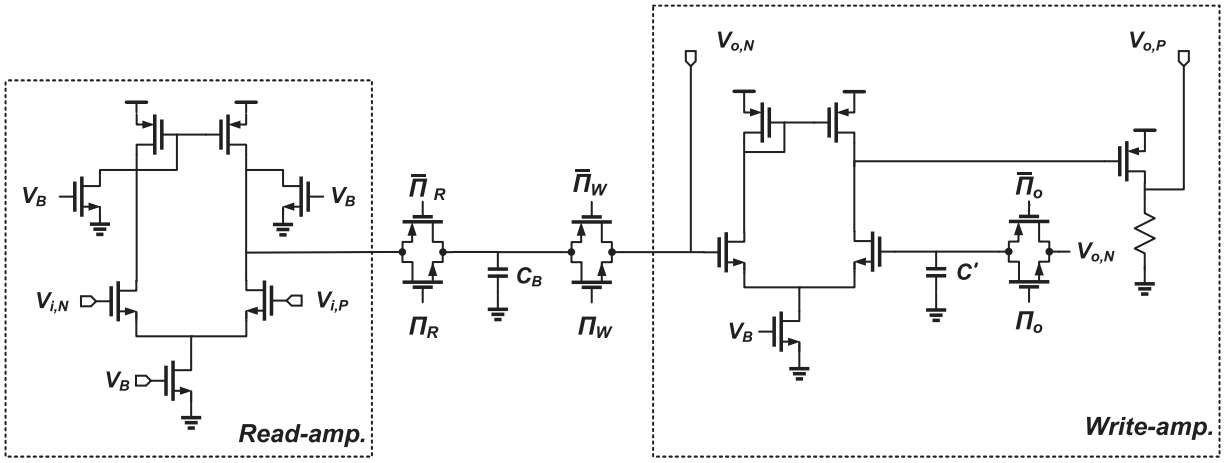


Fig. 9. Read-write amplifier's schematic.

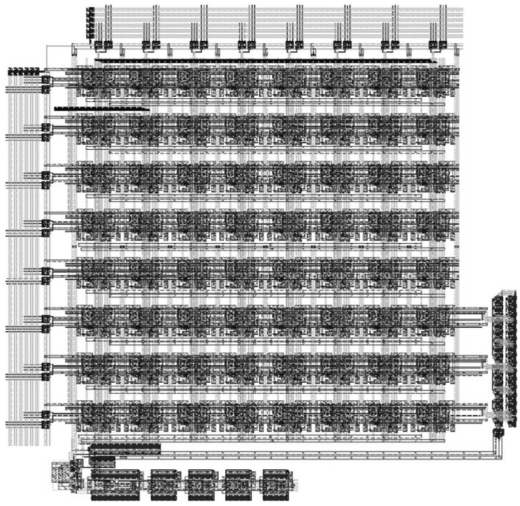


Fig. 10. Layout of the integrated circuit with 64 nodes.

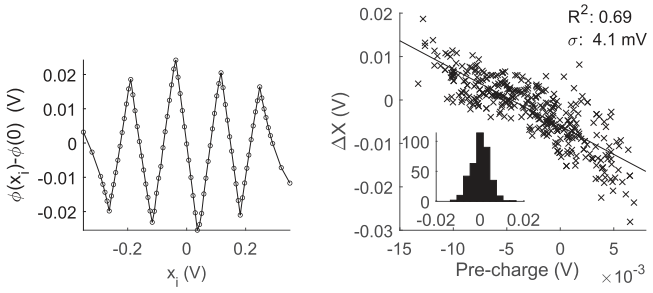


Fig. 11. (a) ϕ-realization. (b) Write-test; inset plots the histogram of ΔX with the mean-line as the reference.

function has a constant vertical shift, which is represented by an offset ϕ_0 along the Y-axis.

This systematic error is caused by the non-linearity of the g_M-cells, which affects the ϕ computation via inaccurate placement of the levels of the DAC and hence, the zeros of ϕ. Non-linearity of the g_M-cell also leads to curvature in the individual ϕ segments. This is apparent once we move away from the origin, i.e., for a large difference between the state charges. Both non-idealities arise from small output resistance of the transistors that makes it difficult to draw large current from them. Though this may be avoided by the use of buffered DACs, but as previously mentioned, we avoided this for an ease of design by re-using the same g_M-cells and hence directly controlling the zeros of ϕ.

Fig. 11(b) plots the charge-update characteristics, i.e., net change in the state-voltage for the corresponding read-bus voltage. This is obtained from the layout simulations of the nodal coupling dynamics of randomly connected graphs. We observe a random distribution of the net change around the mean line, with a standard deviation of 4 mV. A vertical offset at the zero-input is also observed, which would cause a collective temporal drooping of the state-voltages.

The real writing process may be modelled as:

$$\phi_n(v_i - v_j) = \phi(v_i - v_j) + w_n\mathcal{N}(0,1) + \phi(0), \qquad (17)$$

where, w_n is standard deviation of the noise in V.

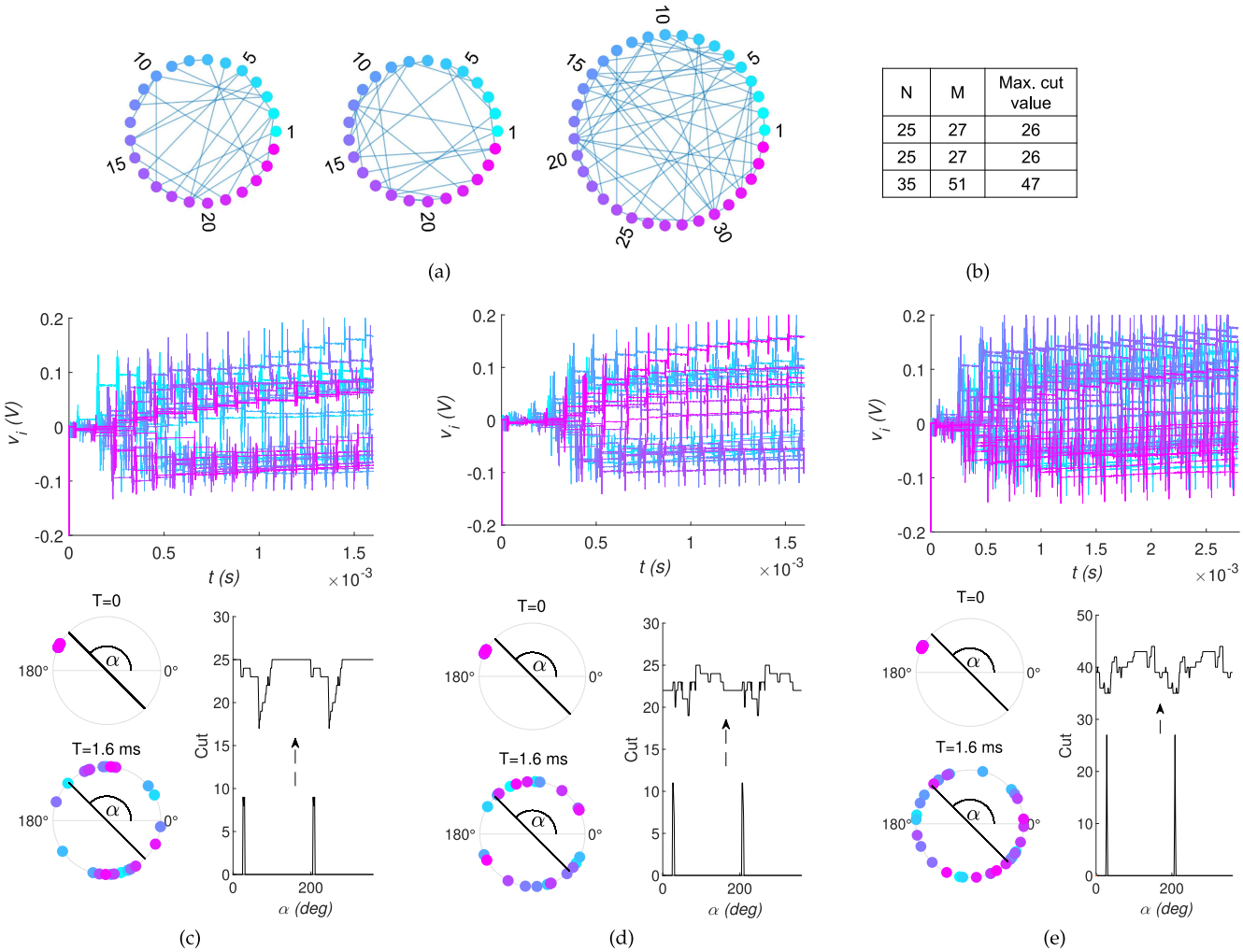
Write-bus impedance is one of the major causes for these errors in writing, because the differential voltage across the write-bus deviates from the actual write-amplifier's output. The bus may be modelled as a long wire with a distributed (often time-variant) line impedance, leading to a distribution in the charge increments. Other parasitic phenomena in the read-write amplifier, such as capacitive feed-through, lead to an offset in ϕ. Removal of this offset constitutes future work and a possible solution is discussed in Section VI.

B. Modelling Process-Related Variations

The process-related variations are expected to affect only the analog components of a vertex: the capacitor and the g_M-cell. Another (partly) analog component, the DAC is shared and tunable while in operation.

Let the actual vertex state-capacitance be C_i' and output conductance be g_M'

$$C_i' = C + \Delta C_i$$
$$g'_{M,i} = g_M + \Delta g_{M,i} \qquad (18)$$



N	M	Max. cut value
25	27	26
25	27	26
35	51	47

Fig. 12. Circuit simulations for illustration. (a) Three randomly connected $\{0, 1\}$-weighted graphs with the number of nodes, edge-count and the max-cut tabulated in (b). (c) and (d) Outputs of the three components of the heuristics, for each of the three graphs in (a). The temporal plot depicts the nodal coupling dynamics, in which each colored continuous curve represents the state-voltage of the correspondingly colored node. In the polar plot, each point angularly represents the phasor (w.r.t. $P = 140$ mV) of final state-voltage. Next, the cut is obtained using (2) for several rounding centers/angles and plotted to obtain the cut value.

Any deviation in the overall capacitance from the targeted manifests as a proportional scaling of the charge-increments. Any variation of the g_M-cell translates to a proportional change in the output current while reading. Expressing (15) with process variation effects incorporated, we obtain

$$\delta v_i = \frac{C_B}{C'_i} AR(-1)^{n_{DAC}} \left(g'_{M,i} v_i - g'_{M,j} v_j - n_{DAC} I_0 \right). \tag{19}$$

Introducing:

$$v'_i = \frac{g'_{M,i}}{g_M} v_i,$$

leads to:

$$\delta v'_i = \frac{g'_{M,i}}{g_M} \frac{C_B}{C'_i} AR(-1)^{n_{DAC}} \left(g_M (v'_i - v'_j) - n_{DAC} I_0 \right). \tag{20}$$

Comparing this with the original dynamics of (10), we see that the even though the charge-increment has the same relationship with the vertex' state as intended, each vertex' state gets updated at different rates. The new dynamics, modelling the PV, is:

$$\dot{v}'_i = \gamma_i \sum_j A_{ij} \phi(v'_i - v'_j). \tag{21}$$

Hence, the process variations can be modelled by a non-uniform state-update factor. It should be noted that at the steady state, one has $\dot{\mathbf{v}} = 0$, and therefore, the critical point of the system is independent of γ. Consequently, the system is expected to converge to the same set of stationary points, which diminishes the impact of process-related variations.

C. Max-Cut Results

To demonstrate the efficacy of the BMZ heuristic based max-cut solver, we simulate the dynamics at the transistor-level model of the integrated circuit. We limit our study to random $\{0, 1\}$-weighted graphs with upto 35 nodes. Fig. 12(a) shows the randomly connected graphs we considered for simulation on the proposed machine and the state of the system during/after each of supported sub-processes are shown in Fig. 12(c)–(e). For

TABLE II
MAX-CUT RESULTS WITH MAJOR NON-IDEALITIES

Graph	Mean final cut (100 trials)			
	Ideal conditions	P-V only	Write-noise only	Both P-V and noise
G1	11298	11292	11234	11212
G2	11307	11302	11244	11211
G3	11310	11302	11254	11220
G4	11324	11315	11261	11233
G5	11312	11298	11254	11220
G22	12764	12747	12673	12633
G23	12778	12754	12674	12629
G24	12779	12756	12678	12638
G25	12777	12746	12680	12630
G26	12764	12744	12670	12621
G43	6387	6373	6340	6323
G44	6383	6373	6339	6319
G45	6386	6374	6338	6319
G46	6388	6377	6341	6318
G47	6395	6383	6345	6327
G48	5147	5107	5093	5044
G49	5148	5106	5094	5046
G50	5148	5102	5091	5046
G51	3644	3639	3630	3620
G52	3654	3649	3639	3629
G53	3649	3644	3634	3627
G54	3658	3652	3640	3631

each of the graphs, we see an increase in the value of the cut at the end of the dynamics. Note that the global maximum for the cut expected to be found only through multiple such runs. For instance, the probability of finding the global maximum for the graph with 25 nodes in Fig. 3(a), over 100 trials of 20 time-steps each, is about 20%.

In Sections V-A and V-B we identified write-noise and process variation as main sources of non-idealities that may affect the performance of the proposed machine for larger graphs. To study the effect of these factors we simulate the dynamics under three conditions. For the first set, we assume a non-uniform rate of increments, with a Gaussian distribution of γ_i, i.e.,

$$\gamma_i = 1 + \epsilon\mathcal{N}(0,1), \tag{22}$$

where ϵ is the effective magnitude of variations of the parameters. In the second set of simulations, we add Gaussian noise during each increment of the state, given by (17). Finally, we add both effects to enable the most realistic simulation of the machine practically possible for large graphs.

Table II lists the mean max-cut values for G-set obtained from 100 runs of the relaxed dynamics through 100 discrete time-steps with a high-level scripting language. We used identical set of initial conditions across the columns for each trial of the heuristic. For simulating the effect of PV at practical levels, we used $\epsilon = 0.3$ in (22), and for the write-noise we added a random Gaussian noisy component during each write:

$$\delta x_i = \phi(v_i - v_j) + w_n\mathcal{N}(0,1), \tag{23}$$

with $w_n = 4$ mV. We see that mean final cut in the presence of process variations slightly decreases, which could be attributed to termination of the dynamics before the steady state was reached. In the case of the noisy writes (with or without P-V), we see a moderate decrease in the mean final cut.

VI. DISCUSSION

The offset $\phi(0)$ in (17) causes a write-frequency dependent drooping of the state voltage over time. It can be shown that such offsets, if not checked, lead to deviations large enough to disassociate the dynamics from the intended computational model. The effect of the offset may be eliminated by supplying an additional steady current to the read bus. First, dummy writes (with $V_{i,P} = V_{i,N}$) are performed on a free vertex. The offset causes non-zero write charges to accumulate over a period of time. Negatively feeding back this accumulation signal as a steady component to the read bus current in effect will remove any offset.

The presented computing device is designed to accommodate a fully connected graphs and can be directly employed for a more general class of problems. Some graph topologies, such as the planar, are relatively easily accelerated using alternative Ising machines. Many area efficient accelerators have been proposed that support planar spin lattices [8]. However, assuming planarity of problems limits the applicability of the machine and requires embedding methods [31] that increase the effective number of nodes.

Many problems of practical importance demand weights to possess multiple bits. This can be achieved by modulating the capacitance that sends back the charge to the state capacitor (C_B). For multi-bits of weights, the charge increments can be scaled up by using larger C_B.

VII. CONCLUSION

A computing system replicating the relaxed BMZ heuristic is presented. It determines the maximal cut by using the heuristic, but adapted for CMOS analog-digital mixed-signal mode acceleration, with theoretically-backed performance. The key computing operations of the employed heuristic are the pairwise modulo and multiply-and-accumulate. The heuristic's dynamical component, which concerns this paper, consists of nodes (of a graph) corresponding to the vertices, and coupled, depending on their adjacency, to contribute to evolution of others' states. Computing the periodic coupling function requires several transistors, thus limiting the acceleration/parallelization schemes applicable to the heuristic.

The implemented application-specific computing system comprises an array of custom designed nodes/vertex corresponding to each vertex of the problem graph. Each such node implements analog memory cells with read, write-enables and selection logic. Non-linear state-increment signals are computed in the (analog) current domain using a successively approximating modulo-calculator, external to the vertex array. The area of the integrated circuit of the vertex is dominated by the state-capacitor. The entire integrated circuit with 64 nodes, when laid out, occupies an area of 0.45 mm $\times$ 0.43 mm, and consumes power of about 40 μW per vertex at 1.2 V. Our post-parasitic extraction model of layout shows that the circuit is able to determine the max-cut for small graphs. The key challenge in the design lies in precisely incrementing the states, which involves pushing often small charges through long wires.

REFERENCES

[1] A. Kahng, J. Lieneg, I. Markov, and J. Hu, VLSI physical design: From graph partitioning to timing closure, 2011. [Online]. Available: http://www.springer.com/gp/book/9789048195909

[2] N. A. Pierce and E. Winfree, "Protein design is NP-hard," *Protein Eng. Des. Selection*, vol. 15, no. 10, pp. 779–782, 2002. [Online]. Available: https://academic.oup.com/peds/article-lookup/doi/10.1093/protein/15.10.779

[3] T. Roughgarden, *Algorithms Illuminated: Algorithms for NP-Hard Problems, Part 4*. New York, NY, USA: Soundlikeyourself Publishing, 2020.

[4] S. Dasgupta, C. H. Papadimitriou, and U. Vazirani, *Algorithms*. New York, NY, USA: McGraw-Hill Higher Education, 2006.

[5] V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer, 2001.

[6] D. S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*. Boston, MA, USA: PWS Publishing Company, 1997.

[7] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey of machine learning accelerators," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2020, pp. 1–12.

[8] N. Mohseni, P. L. McMahon, and T. Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," *Nature Rev. Phys.*, vol. 4, no. 6, pp. 363–379, 2022.

[9] S. Che, J. Li, J. W. Sheaffer, K. Skadron, and J. Lach, "Accelerating compute-intensive applications with GPUs and FPGAs," in *Proc. Symp. Appl. Specific Processors*, 2008, pp. 101–107.

[10] A. Agrawal et al., "Approximate computing: Challenges and opportunities," in *Proc. IEEE Int. Conf. Rebooting Comput.*, 2016, pp. 1–8.

[11] I. Dunning, S. Gupta, and J. Silberholz, "What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO," *Informs J. Comput.*, vol. 30, no. 3, pp. 608–624, 2018.

[12] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, pp. 331–360, 2020.

[13] D. P. Williamson and M. Goemans, "Improved maximum approximation algorithms for using cut and satisfiability programming problems semidefinite," *Science*, vol. 42, no. 6, pp. 1115–1145, 1994.

[14] S. Burer, R. D. Monteiro, and Y. Zhang, "Rank-two relaxation heuristics for MAX-CUT and other binary quadratic programs," *SIAM J. Optim.*, vol. 12, no. 2, pp. 503–521, 2002.

[15] A. S. Bandeira, N. Boumal, and V. Voroninski, "On the low-rank approach for semidefinite programs arising in synchronization and community detection," *J. Mach. Learn. Res.*, vol. 49, pp. 361–382, 2016.

[16] N. A. Aadit et al., "Massively parallel probabilistic computing with sparse ising machines," *Nature Electron.*, 2022. [Online]. Available: https://www.nature.com/articles/s41928--022-00774-2

[17] S. Patel, P. Canoza, and S. Salahuddin, "Logically synthesized and hardware-accelerated restricted boltzmann machines for combinatorial optimization and integer factorization," *Nature Electron.*, vol. 5, no. 2, pp. 92–101, 2022. [Online]. Available: https://www.nature.com/articles/s41928--022-00714-0

[18] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "20k-spin ising chip for combinational optimization problem with CMOS annealing," in *Proc. IEEE Dig. Tech. Papers Int. Solid-State Circuits Conf.*, 2015, pp. 432–433.

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

[20] A. Shukla, M. Erementchouk, and P. Mazumder, "Scalable almost-linear dynamical ising machines," 2022. [Online]. Available: http://arxiv.org/abs/2205.14760

[21] F. Barahona, "On the computational complexity of ising spin glass models," *J. Phys. A: Math. Gen.*, vol. 15, no. 10, pp. 3241–3253, 1982.

[22] Y. Fu and P. W. Anderson, "Application of statistical mechanics to NP-complete problems in combinatorial optimisation," *J. Phys. A: Math. Gen.*, vol. 19, no. 9, pp. 1605–1620, 1986.

[23] A. Lucas, "Ising formulations of many NP problems," *Front. Phys.*, vol. 2, no. February, pp. 1–14, 2014. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fphy.2014.00005/abstract

[24] S. Poljak and F. Rendl, "Solving the max-cut problem using eigenvalues," *Discrete Appl. Math.*, vol. 62, no. 1–3, pp. 249–278, 1995.

[25] D. Haglin and S. Venkatesan, "Approximation and intractability results for the maximum cut problem and its variants," *IEEE Trans. Comput.*, vol. 40, no. 1, pp. 110–113, 1991. [Online]. Available: https://ieeexplore.ieee.org/document/67327/

[26] T. Hofmeister and H. Lefmann, "A combinatorial design approach to MAXCUT," *Random Structures Algorithms*, vol. 9, no. 1/2, pp. 163–175, 1996. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1098--2418(199608/09)9:1/2%3C163::AID-RSA10%3E3.0.CO;2-P

[27] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[28] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Summer 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8811809/

[29] W. Zhang et al., "Neuro-inspired computing chips," *Nature Electron.*, vol. 3, no. 7, pp. 371–382, 2020, doi: 10.1038/s41928–020-0435-7.

[30] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: Tata McGraw-Hill, 2002.

[31] D. Oku, K. Terada, M. Hayashi, M. Yamaoka, S. Tanaka, and N. Togawa, "A fully-connected ising model embedding method and its evaluation for CMOS annealing machines," *IEICE Trans. Inf. Syst.*, vol. E 102.D, no. 9, pp. 1696–1706, 2019.

Aditya Shukla received the bachelor's degree in electronics and communications engineering from the Indian Institute of Technology Roorkee, Uttarakhand, India in 2015, and the master's degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India in 2017. In 2017-18, he served as a research fellow for the Industrial Research and Consultancy Center, IIT Bombay. Since 2018, he has been working towards his doctoral degree with the Department of Electrical Engineering and Computer Science, University of Michigan. His research interests include post-Moore era computing machines and devices.

Mikhail Erementchouk received the PhD degree in physics from the City University of New York, New York, NY, USA, in 2005. He is currently a research area specialist Lead with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. His current research interests include quantum and classical transport in complex media, quantum communications, and novel computing architectures.

Pinaki Mazumder (Fellow, IEEE) received the PhD degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1988. Currently, he is a professor of electrical engineering and computer science with the University of Michigan where he has been teaching for the past 25 years. He spent 3 years with National Science Foundation serving as the lead Program Director. He had worked for 6 years in industrial R&D laboratories on VLSI chip design. He has published more than 350 technical papers and 8 books on various aspects of VLSI technology and systems. Prof. Mazumder was a recipient of Digital's Incentives for Excellence Award, BF Goodrich National Collegiate Invention Award, and DARPA Research Excellence Award. He is an AAAS Fellow (2007) and an IEEE Fellow (1999) for his distinguished contributions to the field of VLSI.