DaS: Implementing Dense Ising Machines using Sparse Resistive Networks

Naomi Sagan and Jaijeet Roychowdhury
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
{naomi.sagan,jr}@berkeley.edu

***Abstract*—Ising machines have generated much excitement in recent years due to their promise for solving hard combinatorial optimization problems. However, achieving physical all-to-all connectivity in IC implementations of large, densely-connected Ising machines remains a key challenge. We present a novel approach, DaS, that uses low-rank decomposition to achieve effectively-dense Ising connectivity using only sparsely interconnected hardware. The innovation consists of two components. First, we use the SVD to find a low-rank approximation of the Ising coupling matrix while maintaining very high accuracy. This decomposition requires substantially fewer nonzeros to represent the dense Ising coupling matrix. Second, we develop a method to translate the low-rank decomposition to a hardware implementation that uses only sparse resistive interconnections. We validate DaS on the MU-MIMO detection problem, important in modern telecommunications. Our results indicate that as problem sizes scale, DaS can achieve dense Ising coupling using only 5%-20% of the resistors needed for brute-force dense connections (which would be physically infeasible in ICs). We also outline a crossbar-style physical layout scheme for realizing sparse resistive networks generated by DaS.**

I. Introduction

Over the last decade or so, a new and exciting technology called Ising machines has arisen for solving hard combinatorial optimization (CO) problems. CO problems [1] are important in a wide variety of practical applications, including protein folding, optimal logistics for healthcare/military operations/-transportation, chip routing, cyber-security and cryptography, secure grids and communication networks, autonomous vehicles and robotics, *etc.*. These problems are generally very difficult to solve, *e.g.*, they are typically NP-hard/complete [2]. It has been shown that these problems can be mapped onto an equivalent NP-hard CO problem known as the Ising problem [3]. Ising machines, which use analog hardware to solve the Ising problem, show great promise for outperforming digital and software techniques for solving CO problems [4–13]. The Ising problem is described using a weighted, undirected graph where the nodes are called spins. As explained in Section II-A, solving this problem corresponds to finding spin values that minimize a quantity known as the Ising Hamiltonian.

Depending on the application, the Ising graph may be densely or sparsely connected, *i.e.*, either there is all-to-all (or nearly all-to-all) connectivity between the spins, or each spin is connected to only a few other spins. Dense connectivity is generally difficult to achieve at scale in hardware; this is especially true for IC implementations of Ising machines [11, 13] as the number of spins increases, due to the difficulty of routing $O(n^2)$ connections between n spins. This constitutes a serious barrier to the development of scalable on-chip Ising machines for problems that require dense connections. This problem does not appear to have been considered, far less addressed, in prior work on the subject, despite how important it is for practical realization of Ising machines.

One such dense problem, important in telecommunications, is the MU-MIMO (Multi-User Multiple-Input-Multiple-Output) detection problem [14, 15]. The efficacy of Ising machines for MU-MIMO has been demonstrated in simulation [16], but building hardware to realize this promise, especially for larger problems, runs into the issue of implementing dense connectivity. It is essential to achieve dense Ising connectivity at scale if practical Ising machine ICs that solve, *e.g.*, the MU-MIMO problem, are to become a reality.

ICCAD '22, October 30-November 3, 2022, San Diego, CA, USA

ACM ISBN 978-1-4503-9217-4/22/10.
https://doi.org/10.1145/3508352.3549467

In this paper, we devise a highly accurate method to represent a useful class of dense Ising problems using only sparse on-chip resistive networks. Such sparse networks are practical to implement at scale on ICs. Our method exploits low-rank structure in the matrix of Ising couplings (*i.e.*, weights of the Ising graph). This low-rank structure results in a singular value decomposition (SVD) [17] where most singular values are negligible. As a result, the matrix can be represented, with very little error, using a truncated SVD that requires far fewer nonzero values than the original dense matrix needs.

The next task is to map this SVD approximation onto a sparse on-chip resistive network. We first devise a technique for doing so when all required resistors are positive-valued. We use a small number of Kirchhoff's Current Law (KCL)-enforcing auxiliary nodes that form bipartite connections with the spins. We then show the current-voltage relationship between the spins is equivalent to that from the truncated SVD. There are as many auxiliary nodes as non-negligible singular values, so this constitutes a sparse connectivity fabric when the Ising coupling matrix has low-rank structure. We devise a mathematical procedure for generating this sparse bipartite resistive network from a truncated SVD of the Ising coupling matrix.

Typical Ising problems require a mix of positive and negative resistors,[1] the latter being complicated to implement physically. We show that even negative values in the truncated SVD mapping can be elegantly implemented using only positive resistors. If each spin has a differential output, any connectivity pattern can be realized using only positive resistors between the auxiliary nodes and either the positive or negative spins. Our sparse synthesis algorithm is thus able to treat any Ising coupling matrix using only positive physical resistors.

We validate DaS on a range of MU-MIMO detection problems, produced using MATLAB's Phased Array System Toolbox [19, 20]. For large problem sizes, the MU-MIMO Ising coupling matrix typically has low effective rank, *i.e.*, many negligible singular values. We achieve density reductions of ~ 80 to 95%, depending on the number of scatterers that interfere with the communication link. The SVD approximation has very little impact on the coupling matrix; the absolute sum of the element-wise approximation error is less than 10^{-8} times the absolute sum of the Ising weights.

We also illustrate a simple layout technique, based on a crossbar-

[1] Negative resistors [18] have a negative R in the Ohm's Law relation $V = IR$.

style physical architecture, that is suitable for implementing the sparse, bipartite resistive networks generated by DaS. The technique allows programmable physical interconnectivity, *i.e.*, any arbitrary Ising problem with a low-rank structure can be programmed onto the chip.

We believe the ideas embodied in DaS will be crucial for feasibly implementing on-chip Ising machines for MU-MIMO, especially as their sizes scale to address the needs of 6G and future standards, and potentially for other dense Ising problems as well. The remainder of the paper is organized as follows:

1) In Sections II-A and II-B, we outline the Ising problem and the Ising formulation of MU-MIMO.
2) In Section II-C, we provide a brief summary of DaS and the core results.
3) In Section III-A, we describe a densely-connected Ising connectivity mesh and its current-voltage relationship.
4) In Section III-B, we describe the sparsification achieved by a truncated SVD approximation of a matrix with low effective rank. We show how to improve the sparsity achieved for certain classes of matrices in Sections III-B1 and III-B2.
5) In Section III-B3, we obtain a bound on the approximation error of the Ising Hamiltonian due to taking the truncated SVD of the Ising coupling matrix.
6) In Section III-C, we map the truncated SVD onto a sparse resistive network, describing how to implement negative values in the SVD mapping in Section III-C1. We describe various implementation details in Sections III-C2 to III-C4.
7) In Section III-D, we describe how to implement DaS as a programmable network with a crossbar architecture.
8) In Section IV, we show results for various MU-MIMO problems, demonstrating its effectiveness as the number of spins scales.

II. Background and Overview

A. The Ising Problem

The Ising problem involves a weighted, undirected graph of n spin nodes, where there are no self-loops. The value of spin i is denoted s_i, and the weight between spins i and j is J_{ij}. $\vec{s}$ is the vector of spin values and J is the symmetric matrix of Ising weights, known as the Ising coupling matrix.

The objective of the Ising problem is to minimize the Ising Hamiltonian,

$$H(\vec{s}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i} J_{ij}s_is_j, \tag{1}$$

where the spins are restricted to be $\in \{\pm 1\}$.

B. The MU-MIMO Detection Problem

The MU-MIMO (Multi-User Multiple-Input-Multiple-Output) detection problem is an important problem in telecommunications that can be mapped to the Ising problem. In modern wireless communication, multiple users, each with one or more transmit antennas, simultaneously transmit to multiple receive antennas. The received signals are, therefore, a noisy combination of each user's transmitted symbols. Because the transmitted symbols are discrete, recovering them from the received signals turns out to be a hard combinatorial optimization problem, *i.e.*, the discrete maximum-likelihood estimation (MLE) problem. The subsequent paragraphs summarize how to map the MU-MIMO problem to the Ising problem, allowing it to be solved by an Ising machine [15, 16].

Consider a MU-MIMO problem with N_t transmitters and N_r receivers. Let the transmitted signal be denoted by a vector $\vec{x}$ of length N_t, such that $x_i \in \{\pm 1\}$. The resulting vector of received signals is

$$\vec{y} = H_C\vec{x} + \vec{n}, \tag{2}$$

where $H_C \in \mathbb{C}^{N_r\times N_t}$ is the channel transmission matrix and $\vec{n} \in \mathbb{C}^{N_r}$ represents additive white Gaussian noise (AWGN).

H_C, $\vec{y}$, and $\vec{n}$ are complex, so we represent them using real numbers by vertically stacking their real and imaginary components:

$$H_{C,\mathfrak{Re}} = \begin{bmatrix}\mathfrak{Re}\{H_C\}\\ \mathfrak{Im}\{H_C\}\end{bmatrix} \in \mathbb{R}^{2N_r\times N_t}, \vec{y}_{\mathfrak{Re}} = \begin{bmatrix}\mathfrak{Re}\{\vec{y}\}\\ \mathfrak{Im}\{\vec{y}\}\end{bmatrix} \in \mathbb{R}^{2N_r}, \quad \vec{n}_{\mathfrak{Re}} = \begin{bmatrix}\mathfrak{Re}\{\vec{n}\}\\ \mathfrak{Im}\{\vec{n}\}\end{bmatrix} \in \mathbb{R}^{2N_r}. \tag{3}$$

In the detection problem, we would like to determine the symbols $\vec{x}$ that minimize the mean-squared error, given by

$$||H_{C,\mathfrak{Re}}\vec{x} - \vec{y}_{\mathfrak{Re}}||^2 = (H_{C,\mathfrak{Re}}\vec{x} - \vec{y}_{\mathfrak{Re}})^T(H_{C,\mathfrak{Re}}\vec{x} - \vec{y}_{\mathfrak{Re}}). \tag{4}$$

To reach an Ising formulation for this minimization problem, we want to write $H_{C,\mathfrak{Re}}\vec{x} - \vec{y}_{\mathfrak{Re}}$ as a single matrix-vector product. To do so, we define

$$\hat{H}_{C,\mathfrak{Re}} = [H_{C,\mathfrak{Re}} \quad -\vec{y}_{\mathfrak{Re}}], \text{ and } \hat{x} = \begin{bmatrix}\vec{x}\\ 1\end{bmatrix}. \tag{5}$$

With these definitions, (4) becomes

$$\hat{x}^T\hat{H}^T_{C,\mathfrak{Re}}\hat{H}_{C,\mathfrak{Re}}\hat{x}. \tag{6}$$

As the objective of the Ising problem is to minimize

$$-\frac{1}{2}\sum_{i,j} J_{ij}s_is_j = -\frac{1}{2}\vec{s}^TJ\vec{s},\ s_i \in \{\pm 1\}, \tag{7}$$

the MU-MIMO problem matches the Ising problem if

$$J = -2\hat{H}^T_{C,\mathfrak{Re}}\hat{H}_{C,\mathfrak{Re}}, \text{ and } \vec{s} = \hat{x}. \tag{8}$$

Ising machines, therefore, can be used to solve the MU-MIMO detection problem.

The channel transmission matrix, and therefore the Ising coupling matrix, is invariably dense. Although on-chip implementations of dense Ising solvers are viable for smaller problems, routing becomes infeasible as the problems scale. Therefore, we wish to sparsify the MU-MIMO problem to make it more amenable to large on-chip Ising solver implementations.

C. Overview of Our Method

We show, in Section III-B, that a singular value decomposition (SVD) of a low-rank dense coupling matrix can yield a sparse representation. To reach this sparse representation, we compute the truncated SVD of coupling matrix $J \in \mathbb{R}^{n\times n}$, $J = U_m\Sigma_mV_m^T$. $U_m \in \mathbb{R}^{n\times m}$, $\Sigma_m \in \mathbb{R}^{m\times m}$, and $V_m \in \mathbb{R}^{m\times n}$, where m is the number of singular values above a certain threshold. This threshold can be chosen such that the Ising coupling matrix from the truncated SVD is almost identical to the Hamiltonian of the original coupling matrix. If $m \ll n$, U_m, Σ_m, and V_m have far fewer values than those needed to represent a dense coupling matrix; this is the provenance of the desired sparsification.

To use this sparse representation for on-chip Ising solvers, it is crucial to map it onto a sparse resistive network. This process, however, is non-trivial, especially if there are negative entries in U_m and V_m and if $U_m \neq V_m$ (*i.e.*, J is not positive semi-definite), both of which are invariably the case in problems of interest. We devised a general procedure to translate the SVD representation to a sparse resistive network via a set of m auxiliary nodes, as described in Section III-C. We also show, in Section III-D, the basic idea behind how our scheme can be laid out using a programmable crossbar-style architecture.

In Section IV, we show that Ising coupling matrices for many MU-MIMO problems have a low-rank structure, especially for large numbers of transmitters and receivers. As a result, they are amenable to sparsification using DaS. In general, the sparsity of a MU-MIMO problem increases as the size of the problem (number of spins) increases, as the number of scatterers obstructing the communication channel decreases, or as the spacing between individual transmitters or receivers decreases. That sparsity increases as problem sizes increase is, in particular, a desirable feature from the standpoint of scalable Ising machine implementation.

III. Dense as Sparse

A. Dense Connectivity

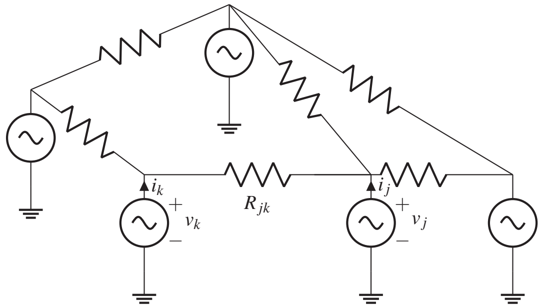

Fig. 1: Example of an Ising Machine connectivity mesh.

In Figure 1, we illustrate a resistive mesh for implementing analog Ising machine connectivity. If there is all-to-all connectivity, this mesh has $\frac{n(n-1)}{2}$ resistors. Each voltage source represents a spin node; the Ising weights are encoded in the conductances $J_{jk} = \frac{1}{R_{jk}}$ of the network [11–13]. Applying Kirchhoff's Current Law (KCL) at the output of every spin gives us

$$\left(\begin{bmatrix}\sum_j J_{1j} & \cdots & 0\\ & \ddots & \\ 0 & \cdots & \sum_j J_{nj}\end{bmatrix} - \begin{bmatrix}J_{11} & \cdots & J_{1n}\\ \vdots & \ddots & \vdots\\ J_{n1} & \cdots & J_{nn}\end{bmatrix}\right)\begin{bmatrix}v_1\\ \vdots\\ v_n\end{bmatrix} = \begin{bmatrix}i_1\\ \vdots\\ i_n\end{bmatrix};\ \text{or} \qquad (D-J)\vec{v} = \vec{i}, \tag{9}$$

where $D-J$ is the Laplacian matrix of the network.

We are concerned with the current-voltage relationship because, in analog implementations of Ising machines, the mapping $v_i \mapsto i_j$ prominently defines the coupling from spin i to spin j [11–13]. In Oscillator Ising Machines (OIM), for instance, the current injection at spin j due to spin i (the transconductance from spin i to spin j) determines the Ising coupling via the oscillator's phase response [11].

As D is diagonal, the $D\vec{v}$ term does not represent any interactions between spins. Instead, it constitutes resistive loading on each spin node. The $-J\vec{v}$ term encodes the weights of the Ising problem that is being solved; the transconductance matrix J is, in fact, the Ising coupling matrix.

Our goal is to utilize low-rank structures in the J matrix to represent the current-voltage relationship in (9) sparsely, and to translate the sparse mathematical representation to a physically sparse network of resistive connections.

B. Low-Rank Decompositions of Resistive Connections

The Ising coupling matrix J may have low effective rank, *i.e.*, a large proportion of its singular values may be much smaller than its largest singular value. Assuming that J has m non-negligible singular values, its truncated SVD [21] is $J \approx U_m \Sigma_m V_m^T$, where $U_m, V_m \in \mathbb{R}^{n\times m}$ and $\Sigma_m \in \mathbb{R}^{m\times m}$.

As J is a dense, symmetric matrix with zeros on the diagonals, we require $\frac{n(n-1)}{2}$ values to fully characterize J. Using the truncated SVD, we can represent J using $m(2n+1)$ values (mn each for U_m and V_m and m for Σ_m). If $m \ll n$, then this is a sparse representation: we can represent J using much fewer values than $\frac{n(n-1)}{2}$, with high accuracy, as seen in the examples in Section IV. Here, we define sparsity as the number of values (later, number of resistors) needed to represent J accurately compared to the $\frac{n(n-1)}{2}$ resistors required in Section III-A.

1) Improved Sparsity of the Truncated SVD for Positive and Negative Semi-Definite Matrices

We can decrease the number of values required to represent the truncated SVD of J by making J positive semi-definite (PSD) or negative semi-definite (NSD), *i.e.*, having only positive or only negative eigenvalues [22]. If J is PSD, then its truncated SVD is $J = U_m \Sigma_m U_m^T$. A symmetric NSD matrix can be represented as the negative of a symmetric PSD matrix, so its truncated SVD is $J = -U_m \Sigma_m U_m^T$.

As V_m is the same as or the negative of U_m, J can be represented using only $m(n+1)$ values.

While J is always symmetric in Ising problems, it is generally neither PSD nor NSD. We can, however, add a constant to the diagonals of J to get a PSD or NSD matrix. For instance,

$$J_{\text{PSD}} \triangleq J + \alpha I, \ \text{or} \ J_{\text{NSD}} \triangleq J + \beta I \tag{10}$$

Additions to the diagonal of J do not affect the Ising problem, since the diagonal contributes nothing to the Ising Hamiltonian (1). So, we can replace J with J_{PSD} or J_{NSD} without changing the parameters of the Ising problem.

Consider the case where J is PSD. We wish to minimize the magnitude of α to increase the likelihood that J_{PSD} has low effective rank. As shown below, we achieve this by setting $\alpha = -\lambda_{\min}(J)$, where $\lambda_{\min}(J)$ is the smallest eigenvalue of J.

A matrix $J \in \mathbb{R}^{n\times n}$ is PSD if and only if $\vec{x}^T J \vec{x} \geq 0, \forall \vec{x} \in \mathbb{R}^n$. Without loss of generality, let us consider $\vec{x}$ such that $||\vec{x}||_2 = 1$. Then, it is guaranteed that $\vec{x}^T J \vec{x} \geq \lambda_{\min}(J)$, with equality if $\vec{x}$ is in the span of the eigenvector corresponding to $\lambda_{\min}(J)$.

Setting $\alpha = -\lambda_{\min}(J)$,

$$\vec{x}^T J_{\text{PSD}} \vec{x} = \vec{x}^T J \vec{x} + \alpha \vec{x}^T \vec{x} \geq \lambda_{\min}(J) + \alpha = 0. \tag{11}$$

Thus, $J + \alpha I$ is PSD when $\alpha = -\lambda_{\min}(J)$. This is the minimum value of α such that $J + \alpha I$ is PSD: if $\alpha < -\lambda_{\min}(J)$ and $\vec{x}$ is in the span of the eigenvector corresponding to $\lambda_{\min}(J)$,

$$\vec{x}^T J_{\text{PSD}} \vec{x} = \vec{x}^T J \vec{x} + \alpha \vec{x}^T \vec{x} = \lambda_{\min}(J) + \alpha < 0. \tag{12}$$

This scheme is generally effective if the largest negative eigenvalue of J is sufficiently smaller than its largest positive eigenvalue. Otherwise, adding αI to J will result in a substantial increase in m, making the truncated SVD no longer sparse.

Likewise, to make J NSD, we set $\beta = -\lambda_{\max}(J)$. In general, making J NSD improves the sparsity if the low-rank decomposition of the largest negative eigenvalue of J is much larger than the largest positive eigenvalue.

2) Scaling U and V for Increased Sparisty

As we will see in Section III-C, our hardware implementation requires fewer resistors than the $m(2n+1)$ (for an arbitrary matrix) or $m(n+1)$ (for a PSD or NSD matrix) values in the truncated SVD.

Let us consider an arbitrary, *i.e.*, not PSD or NSD, Ising coupling matrix. Instead of decomposing the coupling matrix as $J \approx U_m \Sigma_m V_m^T$, we $J \approx G_{\text{in}} D_c G_{\text{out}}^T$, where D_c is the diagonal matrix of the column sums of G_{out} (see Section III-C, specifically Section III-C2 for implementation details, as well as motivation behind the naming of these matrices). The columns of V_m and Σ_m are scaled to obtain G_{out} and D_c such that $U_m \Sigma V_m^T = G_{\text{in}} D_c G_{\text{out}}^T$ and D_c is the diagonal matrix of the column sums of G_{out}. So, we represent J using only $2mn$

values, as D_c is obtained from G_{out}.

For a PSD matrix, we represent J_{PSD} as GD_cG^T, where the columns of G are scaled such that $U_m\Sigma_mV_m^T = GD_cG^T$ and D_c consists of the column sums of G. In doing so, we represent J_{PSD} using mn values. Likewise, we can also represent an NSD matrix using mn values.

3) Impact of the Truncated SVD Approximation on the Ising Hamiltonian

In choosing m, the number of singular values to use, we must examine the impact of representing J with a truncated SVD on the Ising Hamiltonian. Let $\hat{J} = U_m\Sigma_mV_m^T$ be the rank-m approximation of the coupling matrix. Define the Ising Hamiltonians for the original and SVD-approximated coupling matrices as

$$H(\vec{s}) = -\frac{1}{2}\sum_{i\neq j} J_{ij}s_is_j \text{ and } \hat{H}(\vec{s}) = -\frac{1}{2}\sum_{i\neq j} \hat{J}_{ij}s_is_j, \tag{13}$$

where $\vec{s}, s_i \in \{-1, 1\}$ is a vector of spin values and J_{ij} is the element of J at row i, column j.

The error between $H(\vec{s})$ and $\hat{H}(\vec{s})$ can be bounded as follows:

$$\begin{aligned} |H(\vec{s}) - \hat{H}(\vec{s})| &= \left| -\frac{1}{2}\sum_{i\neq j}(J_{ij} - \hat{J}_{ij})s_is_j \right| \\ &\leq \frac{1}{2}\sum_{i\neq j}|(J_{ij} - \hat{J}_{ij})s_js_j| = \frac{1}{2}\sum_{i\neq j}|J_{ij} - \hat{J}_{ij}|. \end{aligned} \tag{14}$$

Thus, if we would like to choose m to enforce some bound on the approximation error of the Ising Hamiltonian, $|H(\vec{s}) - \hat{H}(\vec{s})| < \varepsilon$, we can choose the minimum m such that

$$\frac{1}{2}\sum_{i\neq j}|J_{ij} - \hat{J}_{ij}| = \frac{1}{2}\sum_{i\neq j}|J_{ij} - (U_m\Sigma_mV_m^T)_{ij}| < \varepsilon. \tag{15}$$

C. Low-Rank Decompositions as Sparse Resistive Networks

In this section, we devise a method to represent the truncated SVD of a matrix using a resistive network. This method involves two sets of nodes with bipartite connections: spins and auxiliary nodes, as shown in Figure 2. There are n spin nodes, where n is the size of the Ising graph, and m auxiliary nodes, where m is the effective rank of J, as defined in Section III-B.

The resistor between spin j and auxiliary node k, R_{jk}, has conductance $G_{jk} = R_{jk}^{-1}$.

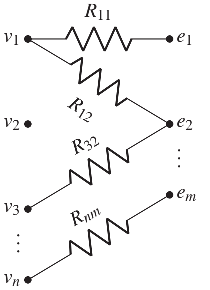

Fig. 2: Bipartite network.

Define the transconductance matrix

$$G \triangleq \begin{bmatrix} G_{11} & \cdots & G_{1m} \\ \vdots & \ddots & \vdots \\ G_{n1} & \cdots & G_{nm} \end{bmatrix}, \tag{16}$$

and define D_c and D_r as the diagonal matrices of the row and column sums of G, respectively. As in Section III-A and Figure 1, define i_j to be the current leaving the j^{th} spin node.

To see how this formulation encodes the truncated SVD of a matrix, let us examine the equations produced by applying KCL at the the auxiliary and spin nodes. At the auxiliary nodes,

$$\begin{bmatrix} \sum_j G_{j1} \cdots & & 0 \\ & \ddots & \\ 0 & & \cdots \sum_j G_{jm} \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} - \begin{bmatrix} G_{11} & \cdots & G_{n1} \\ \vdots & \ddots & \vdots \\ G_{1m} & \cdots & G_{nm} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \vec{0}; \text{ or} \tag{17}$$

$$\vec{e} = D_c^{-1}G^T\vec{v}.$$

At the spin nodes,

$$\begin{bmatrix} \sum_k G_{1k} \cdots & & 0 \\ & \ddots & \\ 0 & & \cdots \sum_k G_{nk} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} - \begin{bmatrix} G_{11} & \cdots & G_{1m} \\ \vdots & \ddots & \vdots \\ G_{n1} & \cdots & G_{nm} \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} i_1 \\ \vdots \\ i_n \end{bmatrix}; \text{ or} \tag{18}$$

$$D_r\vec{v} - G\vec{e} = \vec{i}.$$

Using (17) in (18), we obtain

$$\vec{i} = (D_r - GD_c^{-1}G^T)\vec{v}. \tag{19}$$

As in Section III-A, we incorporate the loading terms, $D_r\vec{v}$, into each spin unit.[2] Comparing (19) with (9), we have $J = GD_c^{-1}G^T$. This matches the form of the SVD of a positive semi-definite matrix, $J_{\text{PSD}} = U_m\Sigma_mU_m^T$.

1) Implementation of Negative Entries in G

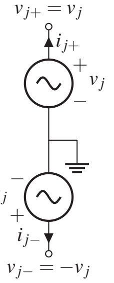

Fig. 3: Differential spin node.

Although U_m of the SVD of J can contain positive and negative entries, G is comprised of conductances and is thus restricted to positive values.[3] In order to properly implement the SVD, we must revise our formulation.

To do so, we use differential spin nodes, *i.e.*, each spin has a positive and negative output. In the bipartite network (Figure 2), we replace each spin v_j with the spin pair v_j and $-v_j$. A connection between $\pm v_j$ and auxiliary node e_k is positive if there is a resistor $R_{jk+} = G_{jk+}^{-1}$ between $v_{j+} = v_j$ and e_k and negative if there is a resistor R_{jk-} between $v_{j-} = -v_j$ and e_k.

The transconductance matrix is now

$$G \triangleq G_+ - G_- = \begin{bmatrix} G_{11+} & \cdots & G_{1m+} \\ \vdots & \ddots & \vdots \\ G_{n1+} & \cdots & G_{nm+} \end{bmatrix} - \begin{bmatrix} G_{11-} & \cdots & G_{1m-} \\ \vdots & \ddots & \vdots \\ G_{n1-} & \cdots & G_{nm-} \end{bmatrix}.$$

We define $D_c = D_{c+} + D_{c-}$, where D_{c+} and D_{c-} are the column sums of G_+ and G_-, respectively. Equivalently, $D_r = D_{r+} + D_{r-}$, where D_{r+} and D_{r-} are the row sums of the respective transconductance matrix.

For this setup, we examine the differential current[4] $\vec{i}_d = \vec{i}_+ - \vec{i}_-$, where $\vec{i}_+$ is the vector of currents leaving the positive spin nodes and $\vec{i}_-$ is the vector of currents leaving the negative spin nodes. Performing KCL at the auxiliary nodes,

$$D_c\vec{e} - (G_+^T\vec{v} - G_-^T\vec{v}) = D_c\vec{e} - G^T\vec{v} = \vec{0}; \text{ or } \vec{e} = D_c^{-1}G^T\vec{v}. \tag{20}$$

At the positive spin outputs, KCL produces the equation

$$\vec{i}_+ = D_{r+}\vec{v} - G_+\vec{e}, \tag{21}$$

and, at the negative outputs, we have

$$\vec{i}_- = -D_{r-}\vec{v} - G_-\vec{e}. \tag{22}$$

Taking the differential current,

$$\begin{aligned} \vec{i}_d = (D_{r+} + D_{r-})\vec{v} - (G_+ - G_-)\vec{e} &= D_r\vec{v} - G\vec{e} \\ &= (D_r - GD_c^{-1}G^T)\vec{v}. \end{aligned} \tag{23}$$

This is equivalent to (19), except that G can now hold negative entries. As $G = G_+ - G_-$, positive entries in G correspond to positive connections and negative entries to

[2] These loading terms do not affect the Ising problem or its Hamiltonian, but circuits implementing Ising spins need to take the loading into account.

[3] While it is possible to approximate negative resistors using nonlinear electronic circuits [18], these greatly increase the complexity of hardware implementation and cannot be perfectly linear.

[4] It is the differential current that constitutes the coupling input into a spin from another, as we show later in this section.

negative connections.

Now, we must show that setting the differential current as in (23) actually relates to the weights of the Ising Hamiltonian. To do so, we linearize the positive and negative spin units around their steady state waveforms, with $\vec{i}_+$ and $\vec{i}_-$ as current inputs. We examine the pair of spin nodes v_{j+}, v_{j-}, looking at the current inputs i_{j+} and i_{j-} in superposition.

First, we will determine the effect of i_{j+} on $v_j(t)$, setting i_{j-} to 0. To demonstrate the core concept, we represent the positive spin by the differential equation

$$\frac{d}{dt}v_j(t) = f(v_j(t)) + bi_{j+}(t), \tag{24}$$

for some nonlinear function $f(\cdot)$ and constant b.[5] Assume that the spin unit's differential equation has steady state $v_{js}(t)$. We then linearize (24) around this steady state to get

$$\begin{aligned} \frac{d}{dt}\Delta v_j(t) &= \frac{d}{dt}(v_j(t) - v_{js}(t)) \\ &= \left.\frac{\partial f}{\partial v_j}\right|_{v_{js}(t)} \Delta v_j(t) + bi_{j+}(t), \end{aligned} \tag{25}$$

where $\Delta v_j(t)$ represents the change in spin voltage from the steady state due to small perturbation $bi_{j+}(t)$.

Now, we examine the effect of i_{j-}, setting i_{j+} to 0. Assume the differential equation governing the negative spin is

$$\begin{aligned} &\frac{d}{dt}(-v_j(t)) = f(-v_j(t)) + bi_{j-}(t); \text{ or} \\ &\frac{d}{dt}v_j(t) = -f(-v_j(t)) - bi_{j-}(t). \end{aligned} \tag{26}$$

Linearizing around the steady state $-v_{js}(t)$,

$$\begin{aligned} \frac{d}{dt}\Delta v_j(t) &= -\left(-\left.\frac{\partial f}{\partial v_j}\right|_{-v_{js}(t)}\right)\Delta v_j(t) - bi_{j-}(t) \\ &= \left.\frac{\partial f}{\partial v_j}\right|_{-v_{js}(t)}\Delta v_j(t) - bi_{j-}(t). \end{aligned} \tag{27}$$

Applying superposition, we add (25) and (27) to get

$$\begin{aligned} \frac{d}{dt}\Delta v_j(t) &= \frac{\left.\frac{\partial f}{\partial v_j}\right|_{v_{js}(t)} + \left.\frac{\partial f}{\partial v_j}\right|_{-v_{js}(t)}}{2}\Delta v_j(t) + b\frac{i_{j+} - i_{j-}}{2} \\ &= \frac{\left.\frac{\partial f}{\partial v_j}\right|_{v_{js}(t)} + \left.\frac{\partial f}{\partial v_j}\right|_{-v_{js}(t)}}{2}\Delta v_j(t) + b\frac{i_{jd}}{2}, \end{aligned} \tag{28}$$

where i_{jd} is the differential current of spin pair v_j, $-v_j$.

Therefore, the effective current input that determines the dynamics of the voltage response $\Delta v_j(t)$ of each differential spin unit is proportional to the differential current i_{jd}.

2) Ensuring Column Sums of G Match Singular Values

To make the current-voltage relationship in (19) match that of dense connectivity mesh in (9), we must have $GD_c^{-1}G^T = U_m\Sigma_m U_m^T$, where $J = U_m\Sigma_m U_m^T$ for a PSD Ising coupling matrix. Now that we are able to set G to match U_m, we must ensure that D_c^{-1} matches Σ_m, the diagonal matrix of singular values.

The most straightforward solution is to add a resistor from each auxiliary node to ground. Denote the resistor from node e_k to ground by R_{e_k}, its conductance by G_{e_k}, and the diagonal matrix of these conductances by D_e. Now, the KCL equations at the auxiliary nodes are

$$\begin{aligned} D_c\vec{e} + D_e\vec{e} - G^T\vec{v} = (D_c + D_e)\vec{e} - G^T\vec{v} = \vec{0}; \text{ or} \\ \vec{e} = (D_c + D_e)^{-1}G^T\vec{v}. \end{aligned} \tag{29}$$

The equation for the differential current $\vec{i}_d$ is now

$$\vec{i}_d = (D_r - G(D_c + D_e)^{-1}G^T)\vec{v}. \tag{30}$$

Thus, we can set D_e such that $(D_c + D_e)^{-1} = \Sigma_m$, or

$$G_{e_k} + \sum_j G_{jk+} + \sum_j G_{jk-} = \sigma_k^{-1}, \tag{31}$$

where σ_k is the k^{th} singular value of J.

If $\sigma_k^{-1} > \sum_j G_{jk+} + \sum_j G_{jk-}$, then this method allows (30) to match the truncated SVD of J without any added resistive loading on the spin nodes. However, if $\sigma_k^{-1} < \sum_j G_{jk+} + \sum_j G_{jk-}$, then G_{e_k} must be a negative resistor. This, while possible to implement, is design- and area-intensive.

Alternatively, we can scale the elements of G such that $GD_c^{-1}G^T = U_m\Sigma_m U_m^T$, without any added resistors. Let us denote the column sums of U_m by $D_{c,U}$. We will set the transconductance matrix G to U_mD_β, where D_β is a diagonal matrix of scaling factors $\text{diag}(\beta_1,\ldots,\beta_m) = D_\beta$. In this configuration, β_k multiplies the k^{th} column of U_m, so the column sums of G are $D_c = D_{c,U}D_\beta$.

To have $GD_c^{-1}G$ match $U_m\Sigma_m U_m^T$, we must have

$$U_mD_\beta D_\beta^{-1}D_{c,U}^{-1}D_\beta U_m^T = U_mD_{c,U}^{-1}D_\beta U_m^T = U_m\Sigma_m U_m^T, \tag{32}$$

so we must define D_β such that

$$D_{c,U}^{-1}D_\beta = \Sigma_m \implies D_\beta = D_{c,U}\Sigma_m. \tag{33}$$

One disadvantage of this method is that, if the elements of $D_{c,U}\Sigma_m$ are too large, it could lead to excessive loading on some spin units. This can be mitigated by scaling J by some constant, positive factor γ.[6] If $\Sigma_m' = \gamma\Sigma_m$ is the singular value matrix after scaling, we have $D_\beta = \gamma D_{c,U}\Sigma_m$. The resistive loading term on the spins is $D_r\vec{v}$, so we can choose γ to be small enough such that the row sums of $G = \gamma U_mD_{c,U}\Sigma_m$ do not constitute excessive loading.

3) Representing Negative Semi-Definite (NSD) Matrices

If the coupling matrix is negative semi-definite, we have $J_{\text{NSD}} = -U_m\Sigma_mU_m$. So, at the spin nodes, we must have the current-voltage relationship

$$\vec{i}_d = (D - J_{\text{NSD}})\vec{v} = (D + U_m\Sigma_mU_m^T)\vec{v}, \tag{34}$$

where D is a diagonal loading matrix.

To implement this formulation as a resistive network, we add unity gain buffers between the spin nodes and the resistors connecting them to the auxiliary nodes. We refer to the outputs of these buffers as "buffered nodes". The current output of these buffers can then be reflected to the spin nodes via current mirrors, such that the current leaving spin node v_{j+} is $-i_{bj+}$. For the purpose of this paper, we represent this process using current-controlled current sources rather than showing the detailed transistor-level implementation we have devised. As the number of buffers and current-controlled current sources scales linearly with the number of spins, they do not significantly impede on-chip layout.

Applying KCL to the buffered nodes in Figure 4, we get the same result as in (23):

$$\vec{i}_{b+} - \vec{i}_{b-} = (D_r - GD_c^{-1}G^T)\vec{v}_b, \tag{35}$$

where D_r and G are defined as in Section III-C1, $\vec{i}_{b+}$ is the vector of currents leaving the positive buffered spin nodes, and

[5] Any analog Ising machine spin can be represented by such a differential equation [23].

[6] As this modification scales the Ising Hamiltonian by a constant, positive number, it does not affect Hamiltonian minimization.

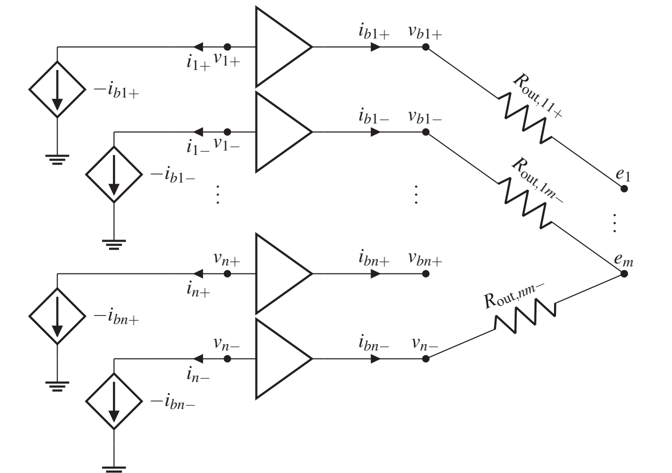

Fig. 4: Formulation for representing NSD matrices, with buffers and current-controlled current sources at each spin node.

$\vec{i}_{b-}$ is the currents leaving the negative buffered spin nodes.

As a result, the differential current $\vec{i}_d$ at the spin nodes is

$$\vec{i}_d = -(D_r - GD_c^{-1}G^T)\vec{v}_b = (-D_r + GD_c^{-1}G^T)\vec{v}. \qquad (36)$$

This matches the desired relation in (34).

4) Representing Arbitrary Matrices

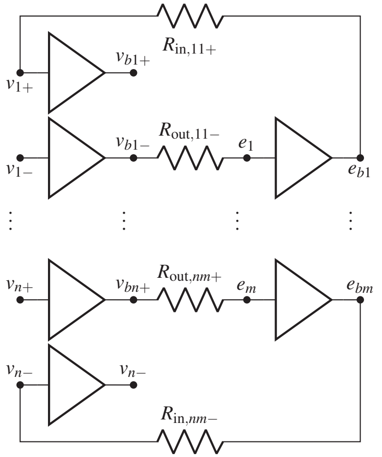

Fig. 5: Bipartite network with buffers at every spin and auxiliary node, allowing for unidirectional connections.

For coupling matrices that are not PSD or NSD (and cannot be made PSD or NSD without sacrificing their low-rank structure), we must implement $J = U_m\Sigma_m V_m^T$, where $U_m \neq V_m$. To achieve this relationship, we add buffered nodes to each spin and auxiliary node, as shown in Figure 5. The number of buffers scales as $n+m$, so, as in Section III-C3, they do not form a barrier to on-chip implementation.

We have resistors $R_{\text{out},jk+} = G_{\text{out},jk+}^{-1}$ from buffered spin node v_{bj+} to auxiliary node e_k, and an analogous setup for a negative connection.[7] Likewise, we have resistors $R_{\text{in},jk+} = G_{\text{in},jk+}^{-1}$ from buffered auxiliary node e_{bk} to spin v_{j+}, and an analogous setup for a negative connection. Define

$$G_{\text{in}} \triangleq \begin{bmatrix} G_{\text{in},11+} & \cdots & G_{\text{in},1m+} \\ \vdots & \cdots & \vdots \\ G_{\text{in},n1+} & \cdots & G_{\text{in},nm+} \end{bmatrix} - \begin{bmatrix} G_{\text{in},11-} & \cdots & G_{\text{in},1m-} \\ \vdots & \cdots & \vdots \\ G_{\text{in},n1-} & \cdots & G_{\text{in},nm-} \end{bmatrix},$$

$$G_{\text{out}} \triangleq \begin{bmatrix} G_{\text{out},11+} & \cdots & G_{\text{out},1m+} \\ \vdots & \cdots & \vdots \\ G_{\text{out},n1+} & \cdots & G_{\text{out},nm+} \end{bmatrix} - \begin{bmatrix} G_{\text{out},11-} & \cdots & G_{\text{out},1m-} \\ \vdots & \cdots & \vdots \\ G_{\text{out},n1-} & \cdots & G_{\text{out},nm-} \end{bmatrix},$$

the row sum matrices as $D_{r,\text{in}}$ and $D_{r,\text{out}}$, and the column sum matrices as $D_{c,\text{in}}$ and $D_{c,\text{out}}$.

Performing KCL at the (non-buffered) auxiliary nodes, we have

$$D_{c,\text{out}}\vec{e} = G_{\text{out}}^T\vec{v} \implies \vec{e} = D_{c,\text{out}}^{-1}G_{\text{out}}^T\vec{v}, \qquad (37)$$

as $v_{bj} = v_j$. At the (non-buffered) spin nodes, we get

$$\vec{i}_d = (D_{r,\text{in}} - G_{\text{in}})\vec{e} = D_{r,\text{in}}\vec{v} - G_{\text{in}}D_{c,\text{out}}^{-1}G_{\text{out}}^T\vec{v}. \qquad (38)$$

This matches the desired relationship $J = U_m\Sigma_m V_m^T$, if we set G_{in} to be proportional to U_m, $D_{c,\text{out}}$ to be proportional to Σ_m, and G_{out} to be proportional to V_m.

D. Analog DaS Crossbar Programmable Implementation

DaS can be implemented in a programmable fashion using variable resistors and a crossbar switch architecture. Programmability is essential in order to use the same chip to solve multiple problems that fit within a maximum number of spins and feature low-rank structure.

A simple physical layout scheme for programmable connectivity is depicted in Figure 6. The spins are laid out horizontally, and the horizontal wires represent the auxiliary nodes.[8] Each square is a switch potentially connecting a spin to a variable resistor leading to an auxiliary node. There are M layers of switches and M auxiliary nodes, where M is the maximum effective rank we choose to allow.

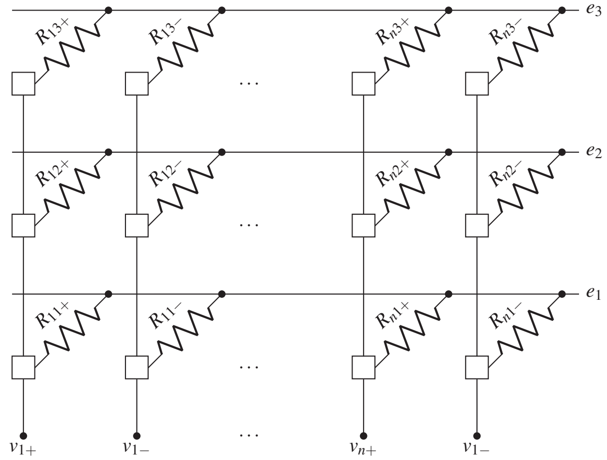

Fig. 6: DaS crossbar implementation for 3 possible auxiliary nodes.

IV. Results

We demonstrate results on a range of MU-MIMO detection problems, where the channel transmission matrix is produced via MATLAB's `scatteringchannelmtx` function in the Phased Array System Toolbox [19, 20] and the Ising coupling matrix is defined as in (8).

There are several parameters that define a MU-MIMO detection problem. First, the number of transmitters and receivers determine the shape of the channel transmission matrix and therefore the size (number of spins) of the Ising problem. The number of spins is equal to the number of transmitters, plus one. The behavior of the channel transmission matrix is primarily shaped by the spacing between transmitters and receivers (in number of wavelengths) and the number of obstructors (*i.e.*, scatterers) in the channel. In general, the channel transmission

[7] The "in" subscript refers to resistors with current flowing into the spin nodes, and the "out" subscript refers to current flowing out of the (buffered) spin nodes into the auxiliary nodes.

[8] This simple layout is to illustrate the concept; in practice, the spin nodes are typically laid out in a rectangular grid, and a few layers of interconnect will be needed. Also, though Figure 6 shows the crossbar architecture concept for a PSD matrix, it is easily extended to support arbitrary coupling matrices.

matrix is smoother for smaller spacing between transmitters or receivers and for fewer obstructors.

In Figures 7 to 9, we plot the Ising coupling matrices for 65-spin MU-MIMO problems (64 transmitters, 128 receivers) with different sets of parameters. For the sake of plotting, we omit the last row and column of J, as these tend to be several times larger in magnitude than the rest of the Ising weights. Figure 7 shows the coupling matrix produced in the case that the spacing is small and there are few scatterers (0.1-wavelength spacing and 5 scatterers). This coupling matrix is very smooth and follows an almost sinusoidal pattern. The coupling matrix in Figure 8 has the same spacing as Figure 7, but with 15 scatterers instead of 5. The matrix, although still smooth, has more variation in the magnitude and location of its peaks and troughs. In Figure 9, we show the opposite end of the spectrum: relatively spaced out transmitters or receivers, and many scatterers for the size of the problem (0.45-wavelength spacing and 50 scatterers). This matrix is not very smooth at all, due to the larger spacing and high number of scatterers.

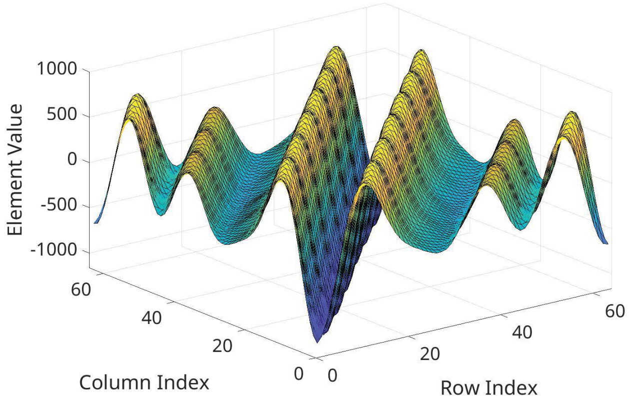

Fig. 7: Ising coupling matrix J for a MU-MIMO problem with 64 transmitters, 128 receivers, 5 scatterers, and 0.1 wavelengths between adjacent transmitters or receivers.

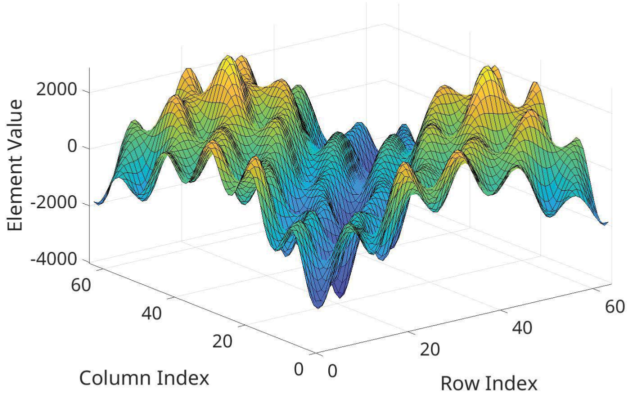

Fig. 8: J for 64 transmitters, 128 receivers, 15 scatterers, and 0.1 wavelengths between adjacent transmitters or receivers.

The MU-MIMO problem (for certain choices of parameters) is well-suited for DaS for two reasons. First, for relatively smooth MU-MIMO Ising coupling matrices, there are many negligible singular values, *i.e.*, the matrix has low effective rank. Second, the coupling matrix is nearly negative semi-definite: its largest negative singular value is much greater in magnitude than its largest positive singular value. So, we use the method in Section III-B1 to make J NSD before performing the truncated SVD. This means that we can implement an Ising machine for MU-MIMO on-chip using mn resistors, where n is the number of spins (number of transmitters, plus 1) and m is the effective rank of J. We choose m such that the approximation error on the Ising Hamiltonian is less than 10^{-8} times the absolute sum of the Ising weights (see Section III-B3).

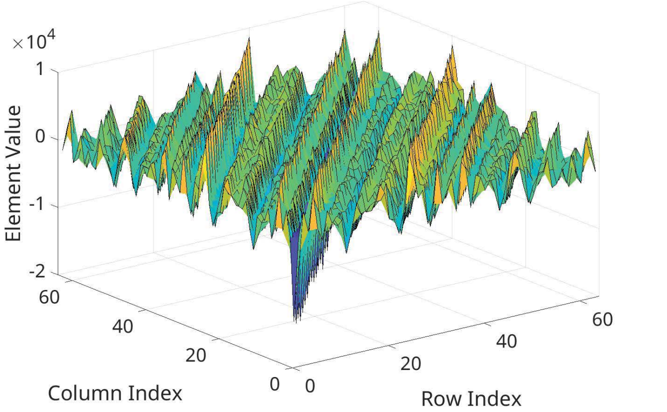

Fig. 9: J for 64 transmitters, 128 receivers, 50 scatterers, and 0.45 wavelengths between adjacent transmitters or receivers.

We performed DaS on MU-MIMO problems with different sets of parameters, choosing m as described above. We then calculated the relative density of each DaS representation: the number of resistors used in DaS divided by the number of resistors required for dense connectivity, *i.e.*, $\frac{nm}{n(n-1)/2} = \frac{2m}{n-1}$.

In Figure 10, we fixed the number of scatterers at 20 and calculated the density of DaS for spacings of 0.05, 0.25, and 0.45 wavelengths, varying the number of transmitters. Overall, the density decreases as the number of transmitters increases. For smaller numbers of transmitters, larger spacing results in higher density, but spacing makes little to no difference as the number of transmitters increases to several hundred. All three curves converge to $\sim 8\%$ density.

In Figure 11, we fixed the spacing to be 0.45 wavelengths and examined problems with 5, 25, and 45 scatterers. As in Figure 10, we varied the number of transmitters, and saw that the density decreases with more transmitters. The density increases with the number of scatterers. The curves for each number of scatterers get closer together as the number of transmitters increases, but they do not converge to the same value. The curves with 5, 25, and 45 scatterers converge to ~ 2, 9, 18% density, respectively.

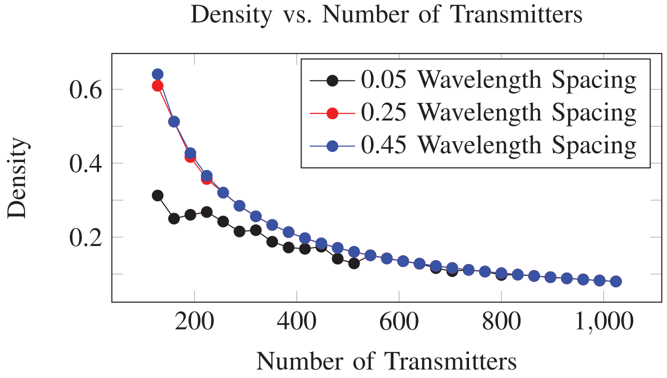

Fig. 10: Normalized number of resistors needed to represent J (1 is dense) vs. number of transmitters, for various spacing. The number of scatterers is fixed at 20.

A. Hierarchical Decomposition

Many matrices representing electrostatic and electromagnetic processes have Green's functions that are smooth in the far

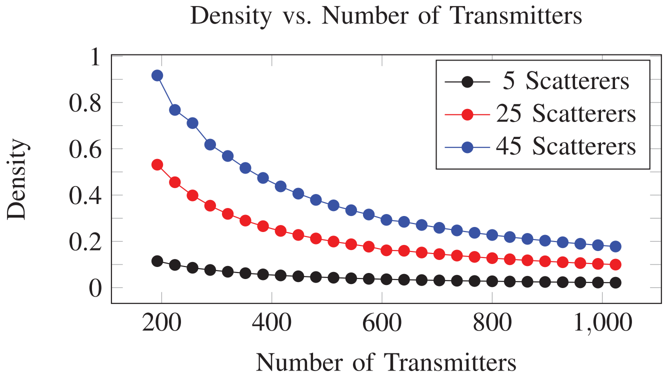

Fig. 11: Normalized number of resistors needed to represent J (1 is dense) vs. number of transmitters, for various numbers of scatterers. The spacing is fixed at 0.45 wavelengths.

field. As a result, although they do not have low effective rank, they have a low-rank structure in off-diagonal blocks.

For Ising coupling matrices derived for such processes, we compute the SVD of matrix blocks as follows:

1) We compute the SVD of J and take note of its effective rank m (the number of singular values above a certain threshold, determined based on the value of the largest singular value).
2) The number of values needed to represent the SVD of J is $N_v = mn$ if J is PSD or NSD, and $N_v = 2mn$ otherwise.
3) Divide J into four equally-sized blocks:
$$J\left(1:\frac{n}{2},1:\frac{n}{2}\right), \quad J\left(1:\frac{n}{2},\frac{n}{2}+1:n\right), \tag{39}$$
$$J\left(\frac{n}{2}+1:n,1:\frac{n}{2}\right), \quad J\left(\frac{n}{2}+1:n,\frac{n}{2}+1:n\right). \tag{40}$$
4) For each block, recursively compute its rank map.
5) Denote the number of values required to represent each sub-block as N_{v1}, V_{v2}, N_{v3}, and N_{v4}. If $N_{v1}+N_{v2}+N_{v3}+N_{v4} < N_v$,*i.e.*, sub-dividing J decreased the density of the representation, set $N_v = N_{v1}+N_{v2}+N_{v3}+N_{v4}$ and return the recursively-determined rank map. Otherwise, return the SVD of the full J matrix.

The SVD of each matrix block can be implemented in hardware via the process in Section III-C4.

We examine $J \triangleq A^T A$, where A is the discretized Green's function of some asymptotically smooth electromagnetic process. This discretized Green's function plays the same role as the scattering channel matrix H in the MU-MIMO problem, so we take the product $A^T A$ to produce an Ising coupling matrix.

For the sake of demonstration, we will approximate the Green's function by the sum of matrices of the following form:

$$A_\ell(i,j) = \begin{cases} \frac{1}{|i-j-\ell|}, & i-j \neq \ell \\ 0, & i-j = \ell. \end{cases} \tag{41}$$

In Figures 12 and 13, we show the results of recursively decomposing various size-512 matrices of the form $J = A^T A$. We measure the density of each block as the number of resistors required to implement it in sparse hardware divided by the number of resistors required for dense connectivity (essentially, $\frac{m}{n}$). Blocks of J are colored according to their density; cyan denotes sparser blocks and magenta denotes denser blocks.

Recursively subdividing these matrices, we achieved a density of 21.36% for $A = A_{25}$ and a density of 28.52% for $A = A_{400} + A_{100}$. Although we did not need this hierarchical low-rank decomposition method for MU-MIMO problems, it can serve as a useful component of our DaS toolbox of sparsification methods for other classes of dense real-world Ising problems.

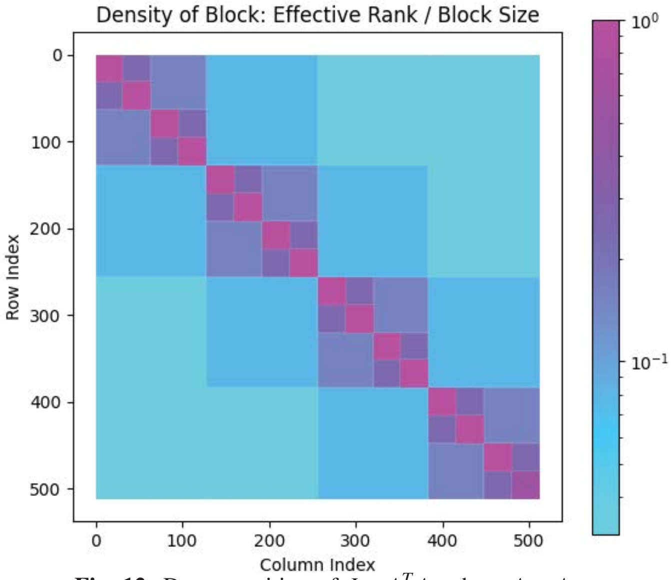

Fig. 12: Decomposition of $J = A^T A$, where $A = A_{25}$.

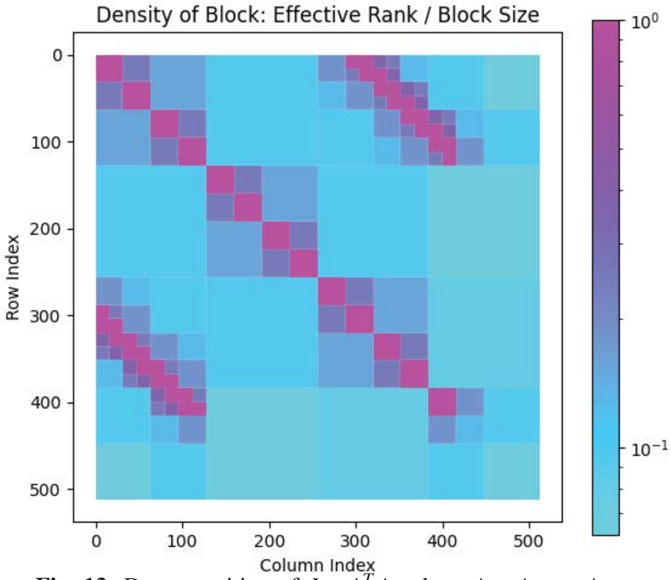

Fig. 13: Decomposition of $J = A^T A$, where $A = A_{400} + A_{100}$.

V. Conclusion

In this paper, we have presented Dense as Sparse (DaS), a method that uses techniques from linear algebra to implement a dense Ising coupling matrix as a sparse resistive network. DaS has two key innovations: using a truncated SVD to sparsify the coupling matrix and synthesising the sparsified form on-chip. This synthesis consists of a bipartite interconnection, with relatively few resistors, between spin and KCL-enforcing auxiliary nodes. DaS can utilize positive/negative semi-definiteness of the Ising coupling matrix to increase sparsity. Results on the MU-MIMO detection problem in telecommunications yield sparsities of ∼5%-20% for size-1000 problems, with essentially no loss of accuracy in coupling. Programmable physical fabrics implementing DaS-generated sparse resistive meshes are easily laid out physically on ICs. As such, DaS offers considerable promise for overcoming a critical roadblock in IC-based Ising machines: implementing dense interconnections at scale.

Acknowledgments

Support from Sandia National Laboratory and the U.S. National Science Foundation (NSF) is gratefully acknowledged.

References

[1] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, pp. 85–103, Springer, 1972.

[2] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[3] A. Lucas, “Ising formulations of many NP problems,” *Frontiers in Physics*, vol. 2, p. 5, 2014.

[4] Z. Bian, F. Chudak, W. G. Macready, and G. Rose, “The Ising model: teaching an old problem new tricks,” *D-Wave Systems*, vol. 2, 2010.

[5] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, *et al.*, “Quantum annealing with manufactured spins,” *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.

[6] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, “Next-Generation Topology of D-Wave Quantum Processors,” *arXiv:2003.00133*, 2020.

[7] Z. Bian, F. Chudak, R. Israel, B. Lackey, W. G. Macready and A. Roy, “Discrete optimization using quantum annealing on sparse Ising models,” *Frontiers in Physics*, vol. 2, p. 56, 2014.

[8] A. Marandi, Z. Wang, K. Takata, R. L. Byer and Y. Yamamoto, “Network of time-multiplexed optical parametric oscillators as a coherent Ising machine,” *Nature Photonics*, vol. 8, no. 12, pp. 937–942, 2014.

[9] T. Wang and J. Roychowdhury, “Oscillator-based Ising Machine,” *arXiv:1709.08102*, 2017.

[10] T. Wang, L. Wu and J. Roychowdhury, “New Computational Results and Hardware Prototypes for Oscillator-based Ising Machines,” in *Proc. IEEE DAC*, pp. 239:1–239:2, 2019.

[11] T. Wang, L. Wu, P. Nobel, and J. Roychowdhury, “Solving combinatorial optimisation problems using oscillator based Ising machines,” *Natural Computing*, pp. 1–20, April 2021.

[12] R. Afoakwa, Y. Zhang, U. K. R. Vengalam, Z. Ignjatovic, and M. Huang, “BRIM: Bistable Resistively-Coupled Ising Machine,” in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 749–760, IEEE, 2021.

[13] J. Roychowdhury, “Bistable Latch Ising Machines,” in *Proc. UCNC*, LNCS sublibrary: Theoretical computer science and general issues, October 2021. Web link.

[14] A. K. Singh, K. Jamieson, D. Venturelli, and P. McMahon, “Ising Machines’ Dynamics and Regularization for Near-Optimal Large and Massive MIMO Detection,” *arXiv:2105.10535v1*, May 2021.

[15] M. Kim, D. Venturelli, and K. Jamieson, “Leveraging quantum annealing for large MIMO processing in centralized radio access networks,” in *Proceedings of the ACM Special Interest Group on Data Communication*, pp. 241–255, ACM, 2019.

[16] J. Roychowdhury, J. Wabnig, and K. P. Srinath, “Performance of Oscillator Ising Machines on Realistic MU-MIMO Decoding Problems,” *Research Square preprint (Version 1)*, 22 September 2021. Web link to preprint.

[17] J. Bisgard, *Analysis and Linear Algebra: The Singular Value Decomposition and Applications*. American Mathematical Society, 2021.

[18] L. O. Chua, J. Yu, and Y. Yu, “Negative resistance devices,” *International Journal of Circuit Theory and Applications*, vol. 11, no. 2, pp. 161–186, 1983.

[19] “MATLAB Phased Array System Toolbox.” Available at https://www.mathworks.com/help/phased/index.html.

[20] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, “An overview of signal processing techniques for millimeter wave MIMO systems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, 2016.

[21] T. F. Chan and P. C. Hansen, “Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations,” *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 3, pp. 519–530, 1990.

[22] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[23] J. Roychowdhury, “Numerical simulation and modelling of electronic and biochemical systems,” *Foundations and Trends in Electronic Design Automation*, vol. 3, pp. 97–303, December 2009.