



Article submitted to journal

Subject Areas:

Combinatorics, Optics, Electrical Engineering

Keywords:

Photonics, Ising, QUBO, Max-Cut, Number Partitioning

Author for correspondence:

Anil Prabhakar

e-mail: anilpr@ee.iitm.ac.in

Optimization with photonic wave based annealers

A. Prabhakar¹, P. Shah¹, U. Gautham¹,
V. Natarajan¹, V. Ramesh¹,
N. Chandrachoodan¹ and S. Tayur²

¹Indian Institute of Technology Madras, Chennai 600036, India.

²Carnegie Mellon University, Pittsburgh, PA 15213, USA

Photonic Ising Machines (PIMs) offer alternatives to quantum annealing and simulated annealing. An NP-hard problem is cast as a quadratic unconstrained binary optimization (QUBO) where the final spin configuration in the Ising model is adiabatically arrived at as a solution to a Hamiltonian, given a known set of interactions between spins. We describe two PIMs and compare their performance against classical and quantum solvers. The temporal multiplexed Ising machine uses the bistable response of an electro-optic modulator to mimic the spin up and down states, and solves the Max-Cut problem on par with Gurobi for up to 1000 spins. In a second construction, a spatial photonic Ising machine convolves the wavefront of a coherent laser beam with the pixel distribution of a spatial light modulator (SLM) to adiabatically achieve a minimum energy configuration, and solve a number partitioning problem. The PIM easily partitions an array of 2^{14} integers, vastly outperforming both Gurobi and the state-of-the-art D-Wave annealer. The components used in our PIM implementations continue to improve, making them viable alternatives to other non von Neumann computing architectures.

1. Introduction

Many quadratic unconstrained binary optimization (QUBO) problems can be cast as Ising Hamiltonians. Solving these problems is motivated by a wide range of applications in supply chain management, portfolio optimization, operations management and machine learning.

© The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License <http://creativecommons.org/licenses/by/4.0/>, which permits unrestricted use, provided the original author and source are credited.

The problem of minimising the Ising Hamiltonian is known to be an NP-complete problem, which allows all other NP-complete problems to be recast in this form [1].

There are many examples of photonic Ising solvers, from injection locked lasers [2,3] and optical parametric oscillators [4–10], to degenerate cavity lasers that solve the phase retrieval problem [11] and simulate the XY model Hamiltonian [12]. In this article, we show improvements on two photonic implementations, the time multiplexed coherent Ising machine [10] and the spatial photonic Ising machine [13], and benchmark their performance against classical solver Gurobi [14] on the Max-Cut and the number partitioning problem, respectively.

2. The Ising Model

The mathematical model consists of discrete variables that represent magnetic dipole moments or atomic spins that can be in one of two states. The spins are arranged on an undirected graph, $G = (V, E)$, with a set of vertices or nodes V and a set of edges E . A vertex is denoted as $i \in V$ and an edge as $\langle i, j \rangle \in E$. A binary variable $s_i \in \{\pm 1\}$ associated with each vertex i , represents spin direction. The Hamiltonian (or energy function, or cost function) of the model is

$$H(\mathbf{s}) = - \sum_{\langle i, j \rangle \in E} J_{ij} s_i s_j - \sum_{i \in V} h_i s_i, \quad (2.1)$$

where J_{ij} is the pairwise interaction constant or coupling constant and h_i is the external magnetic field acting on the spins. We attempt to find a particular spin configuration \mathbf{s} , for which the Hamiltonian energy becomes a minimum. (2.1) can also be cast as a problem without local fields by adding an ancillary spin, s_a , and coupling it to each spin:

$$H' = - \sum_{ij} J_{ij} s_i s_j - \sum_i h_i s_i s_a. \quad (2.2)$$

The first term in (2.2) is invariant to flipping all spins. Hence, the final solution is obtained as $\{s_a s_i\}$ (Each spin is flipped based on the value of s_a).

For a graph with N vertices representing the spin configuration $\mathbf{s} = (s_1, s_2, \dots, s_V)$, the coupling constant or connectivity matrix J_{ij} is a square matrix of dimension $N \times N$, with each element representing the interaction strength between the pairs of spins (s_i, s_j) . The external magnetic field h_i , of dimension N , represents an external magnetic field exerted on s_i .

3. Temporal Multiplexed Ising Machine

The Poor Man's Coherent Ising Machine (CIM) uses a suitably biased Mach-Zender-Modulator (MZM) to produce bistable photonic states [9]. These are time multiplexed to represent the spins s_i , and the optical output is detected, sampled and fed back to the modulator via a field programmable gate array (FPGA). Given J_{ij} , the FPGA estimates $H(\mathbf{s})$ and updates the value of s_i . This is run iteratively until all the values of s_i are unchanged.

There have been many CIMs benchmarked against commercially available state-of-the-art solvers [15–18]. We use our CIM to solve different instances of the Max-cut problem [19], generated from **rudyl** [20]. We then solve the same problem instances using **Gurobi** [14], and compare its performance against that of our CIM.

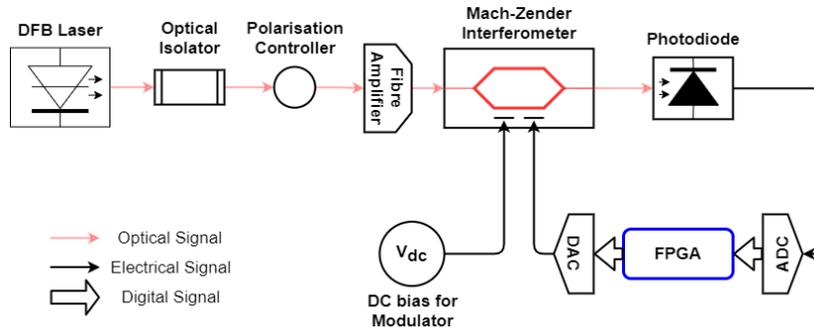


Figure 1. A schematic of the experimental setup used.

(a) Setup and Methods

Fig. 1 shows a schematic of our experimental setup which is based on the work by Bohm et al [9,21], having a transfer function

$$x_n[k+1] = \cos^2(f_n[k] - \frac{\pi}{4} + \zeta_n[k]) - \frac{1}{2} \quad (3.1)$$

$$f_n[k] = \alpha x_n[k] + \beta \sum_m J_{mn} x_m[k], \quad (3.2)$$

where k represents the iteration index. The “spin” values are encoded in the sign of the **time multiplexed** photo-voltage $\sigma_n[k] = \text{sign}(x_n[k])$, where x_n is a particular time bin of the feedback signal (see Fig. 2).

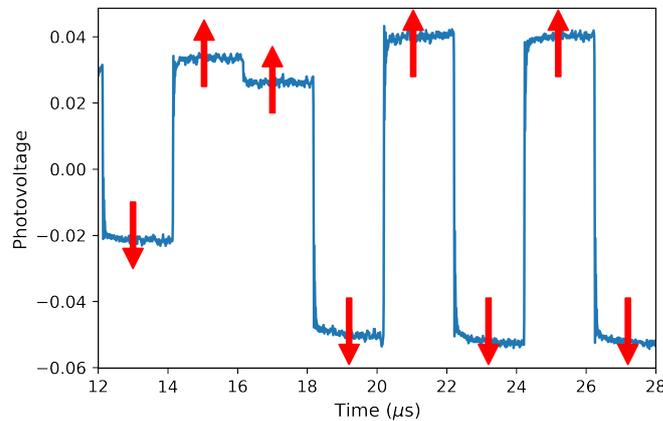


Figure 2. Piece-wise constant values of photo-voltage correspond to different spins, where the spin is up if $V > 0$ and down if $V < 0$. This implementation gives rise to multiplexing of spin values in time.

J_{mn} in (3.2) is the coupling coefficient in the Ising Hamiltonian we want to minimize, and α and β are gain parameters that need to be adjusted for each problem instance to reach near optimal solutions. $\zeta_n[k]$ represents Gaussian noise added to the signal in the FPGA, to ensure initial randomness. The non-linear $\cos^2(\cdot)$ and the bias of $\frac{\pi}{4}$ are implemented using a biased MZM, while $f_n[k] + \zeta_n[k]$ is calculated in the FPGA.

The dynamical equations of this setup create a pitchfork bifurcation [9] as the parameter α is varied. The bifurcation point with $\beta = 0$ is α_{bif} , that is, when $\alpha < \alpha_{\text{bif}}$, only values near 0 are stable. At values of $\alpha > \alpha_{\text{bif}}$, the spins randomly choose values $\pm a_0$, which are the new bistable points (0 is now an unstable point, as seen in Fig. 3). When $\beta \neq 0$, it can be shown [21] that only near-optimal (low energy) solutions to the Ising Hamiltonian are stable for a given α, β .

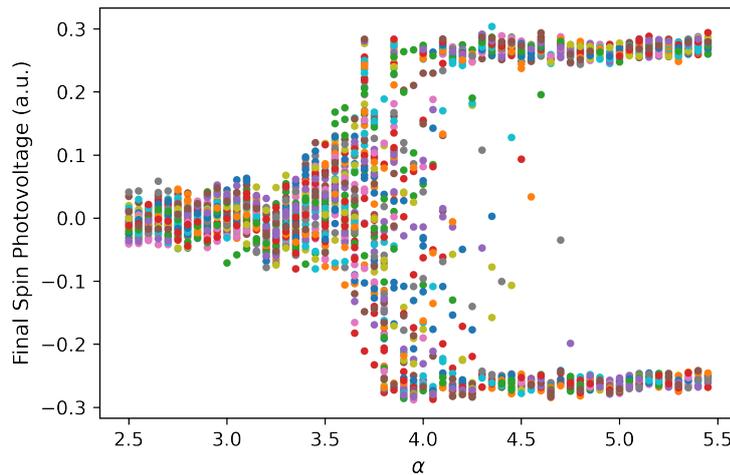


Figure 3. A bifurcation of the final values that spins settle to, as we vary the gain parameter α , with $\beta = 0$. For $\alpha > \alpha_{\text{bif}} \approx 3.7$, we have two bistable points, which is a suitable parameter space to operate in. At each value of α , we allow 60 spins (different colours) to evolve according to (3.1),(3.2) and plot their final value.

For each instance of the Max-cut problem, which we elaborate on below, our choice of α, β will affect the quality of our solution. Instead of doing a brute-force search through a grid of different values of α and β , we derive a bound that helps us to search only through different values of β instead. Consider a positively weighted graph with adjacency matrix J . The goal of a Max-cut problem is to partition the N vertices of the graph into two sets, P^+ and P^- , such that the cut value C (the sum of the weights of edges connecting the P^+, P^-) is maximized. This also divides the edges of the graph into three sets: E^+, E^- and E^C , which are the edges between vertices in P^+, P^- and those between P^+ and P^- respectively. The sum of the edges in E^C gives us the cut,

$$C = \sum_{ij \in E^C} J_{ij}. \quad (3.3)$$

If we label all the spins in P^+ with $\sigma = 1$ and all the spins in P^- with $\sigma = -1$, the above problem can be recast as the Ising Problem, with the Hamiltonian

$$\begin{aligned} H &= - \sum_{ij} J_{ij} \sigma_i \sigma_j = - \sum_{E^+} J_{ij} - \sum_{E^-} J_{ij} - \sum_{E^C} J_{ij} + 2 \sum_{E^C} J_{ij} \\ &= -D + 2 \sum_{E^C} J_{ij}. \end{aligned} \quad (3.4)$$

Here, D is the sum of the weights of the graph and $\sum_{E^C} J_{ij}$ gives you the cut value. Thus, if we set $J_{ij} = -J_{\text{graph}}$, minimizing the Ising Hamiltonian gives us the Max-cut. However, the solutions

obtained are stable only when we satisfy the condition [21]

$$\alpha - \alpha_{\text{bif}} \leq \frac{\beta}{N} \sum_{mn} J_{mn} \sigma_m \sigma_n. \quad (3.5)$$

Since we set J_{mn} to be negative, we can change the sign of the LHS and the inequality. Note that this requires $\alpha < \alpha_{\text{bif}}$ for stability. The summation on the RHS is now positive (due to the sign change), and we can upper bound it by the sum of edges of the graph. With that modification, we can sweep β as

$$\beta = N \frac{\alpha_{\text{bif}} - \alpha}{f|E|}, \quad f = 1, 0.9, 0.8, \dots \quad (3.6)$$

where $|E|$ is the sum off all edges in the graph. After we fix α to around 0.8 times α_{bif} (chosen empirically), we set $f = 1$ and extract β as a function of α . We plot this in a landscape of solution accuracy with α, β , to get the red line in Fig. 4. The proximity of the yellow (stable) region to the red line indicates that our bound is accurate enough for application.

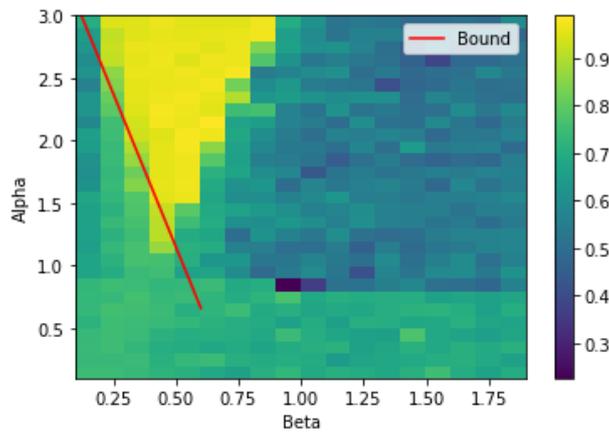


Figure 4. Variation of solution accuracy with α and β for a random 100 spin unweighted graph. The red line shows the lower bound of β , i.e. we sweep β starting from a point on the red line and move right (since we keep α constant).

(b) Hardware

Computation of (3.1) is performed optically using the MZM, while (3.2) occurs either on an FPGA or on the CPU. FPGAs are capable of performing large numbers of parallel multiplications, which we can make use of to speed up the computation. In addition, the FPGA can be used to interface with the high speed ADCs and DACs.¹ We studied the use of two FPGA boards: A Zynq7020 chip on a RedPitaya SIGNALlab 250-12 (250 MHz, 12 bit ADC and DAC) and a RedPitaya STEMLab 125-14 (125 MHz, 14 bit ADC and DAC). Only the SIGNALlab board was used for the experiment. The STEMLab board was used to obtain solution accuracy and for comparisons with **Gurobi**.

The DAC and ADC buffers consist of 16384 samples, of which we use 500 to synchronise the signals; the rest are uniformly distributed to each “spin”. The MZM (an Optilab IM-1550-12) is biased at $V_{\pi/4}$. The output power of the MZM at 0 bias is around -4 dBm, which yields a mean photovoltage of 150 mV. To synchronize the time multiplexed waveforms, we send known waveforms to calibrate for delays.

Our main bottleneck in solution time is the matrix multiplication in (3.2). We optimize this using a system of parallel multipliers on the FPGA to speed up the matrix vector multiplication. Each matrix vector multiplication is decomposed into a set of parallel vector-vector products, or

¹ADC, DAC: Analog to Digital and Digital to Analog Converters respectively

dot products. Each such dot product is handled by a single DSP slice, which essentially acts as a multiply accumulate unit. We parallelize as many dot products as we can by using multiple DSP slices in parallel.

Each DSP slice handles one or many spins, depending on how many slices are in parallel. If the number of DSP slices is less than the number of spins, we “fold” multiple operations onto a single DSP slice. For example, for 16 spins and 8 DSP slices, the first DSP slice will handle spin 1 and spin 9. To allow for parallel access to the adjacency matrix J , we distribute the matrix across multiple Block-RAMs present inside the FPGA. A single row’s elements are stored contiguously in the Block-RAM, but different rows are distributed across multiple RAMs. A single Block-RAM is associated with a single DSP slice, and hence may contain more than one row if the DSP slices handles more than one spin. This promises a reduction in the clock cycles taken (originally $\approx N^2$) by a factor of the number of multipliers present in the FPGA board. Our results are showcased in Fig. 6.

(c) Solving the Max-Cut Problem

We present the results of our CIM on various Max-Cut instances and compare them with **Gurobi** [14], run on an Intel Core i3 processor, versus the CIM run with a ARM Cortex-A9 processor on the Zynq7020. We show the variation of results with Graph Size N and Graph Density D . The graph instances were generated using **rudyl** [20]. Performance variation with size is shown in Fig. 5 and compared with the performance of the DWave quantum annealer. We can see that the CIM performance is robust with graph size, while the Quantum Annealer is limited by its size. Optimizing the anneal schedule will also improve the quantum annealer, which is currently run for an anneal time of $20 \mu s$.

In Tables 1 and 2, we have shown the performance of the CIM against **Gurobi**. We measure the time taken by our CIM (T) and normalize all results against the solution found by **Gurobi**, when run for time $10T$. We also study the accuracy of the solutions obtained by **Gurobi** when restricted to $0.1T, 0.5T, T, 5T$. We see that the CIM performs well for under-constrained systems (graphs with lower density) and loses performance slightly as the density increases (See Table 1). Since finding the right set of parameters becomes harder for larger graphs, the CIM performance is lower in Table 2. However, better bounds on the Max-Cut problem will alleviate this issue.

In Fig. 6, we compare the optimized version of matrix multiplication in our setup with different numbers of multiplier units and we see the expected scaling.

Table 1. Cut values for a 100 node graph, with random unweighted edges. The cut values are normalized against that obtained using Gurobi ($10T$).

Graph Density (%)	CIM (T)	Gurobi ($0.1T$)	Gurobi ($0.5T$)	Gurobi (T)	Gurobi ($5T$)
10	1.014	0.975	0.975	0.975	0.992
20	1.006	0.873	0.984	0.989	0.995
30	1.008	0.862	0.862	0.862	0.991
40	1.019	0.884	0.884	0.884	0.993
50	0.994	1	1	1	1
60	0.986	1	1	1	1
70	0.991	1	1	1	1
80	0.951	0.999	0.999	0.999	0.999
90	0.927	1	1	1	1
100	0.974	1	1	1	1

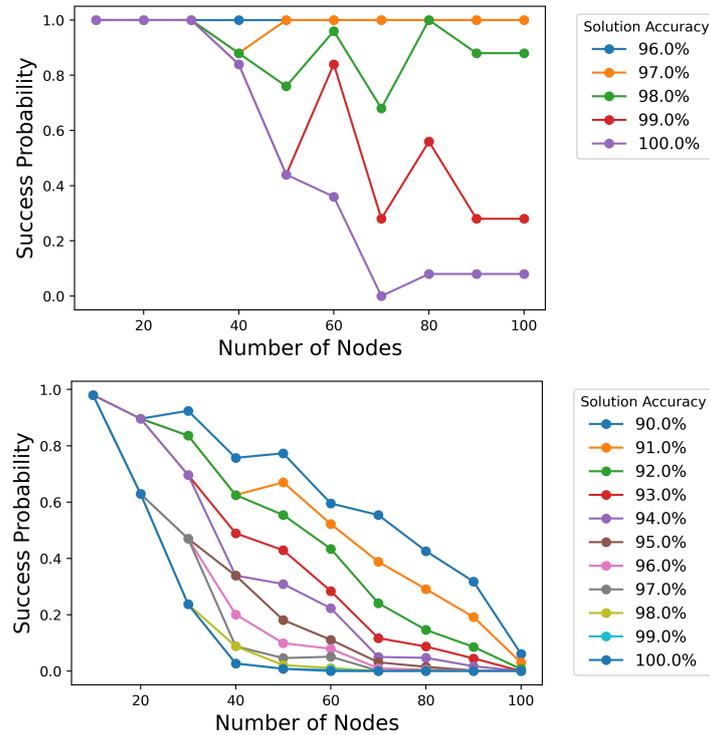


Figure 5. Success Probability of reaching the solution provided by Gurobi, when Gurobi is given a large time using (a) our CIM (b) DWave-Advantage with an anneal time of 20 μ s.

Table 2. Cut values obtained for a 100-1000 node graphs, using randomly generated integral weighted edges from 1 to 10. The cut values obtained by the CIM and different runs of Gurobi are normalized against Gurobi ($10T$).

Graph Size	Time taken (CIM) (s)	CIM (T)	Gurobi ($0.1T$)	Gurobi ($0.5T$)	Gurobi (T)	Gurobi ($5T$)
100	15.500	1.019	0.884	0.884	0.884	0.993
200	21.300	0.985	0.996	0.996	0.998	1
300	31.400	0.982	0.999	1	1	1
400	46	0.990	0.998	0.998	0.998	1
500	65.100	0.979	0.997	0.999	0.999	1
600	85.900	0.960	1	1	1	1
700	112.600	0.961	0.999	0.999	1	1
800	143.800	0.940	0.999	1	1	1
900	179.700	0.977	1	1	1	1
1000	220.200	0.936	0.999	1	1	1

Our line search heuristic is robust against instance variability. This can be further improved by using bounds on Max-cut values in (3.5), instead of using the sum off all edges in the graph. Many studies with bounds have been extensively studied [22,23]. The algorithm can be further improved by adaptively terminating it when the spins settle, instead of running it for a constant number of iterations.

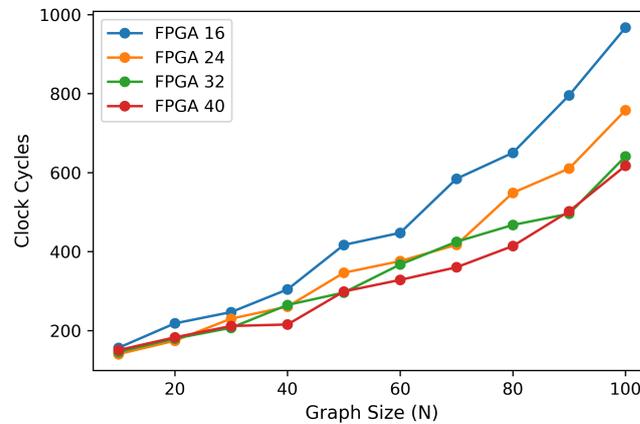


Figure 6. A plot of the number of clock cycles (clock frequency of 100 MHz) taken for matrix multiplication versus instance size, for different numbers of parallel units on the FPGA. The timing is done using the FPGA's clock. In these runs, all the attempts hit the perfect cut value, since the graph instance was small. Hence, the optimality is not represented.

Even without very careful optimization, our CIM is able to perform to reasonably high standards, with accuracies comparable to the Gurobi optimizer. In fact, we do better than Gurobi for graphs with 100 spins and densities less than 50%, as seen in Table 1.

4. Spatial Photonic Ising Machine

The Mattis spin glass [24,25] is a special case of the general Ising model, with the Hamiltonian

$$H = \sum_{i,j} \zeta_i \zeta_j \sigma_i \sigma_j, \quad (4.1)$$

where σ_i is the binary spin at the i^{th} lattice point and ζ_i is the amplitude contributing to the coupling between spins. We constructed a spatial photonic Ising machine (SPIM) to solve (4.1) and map it onto the number partitioning problem (NPP) for problem sizes up to over 16000 spins. We benchmark the working of our system with other QUBO solvers, and show that adiabatically tuning the Hamiltonian provides solutions which are on par with if not better than those achieved by the D-Wave system. Our solution strategy scales linearly at worst with the problem size on our current setup, and opens the door to parallel computing using photonic architectures.

(a) Setup and Methods

The experimental setup employed comprises a laser source, linear optical elements and an imaging system, as shown in a schematic given in Fig. 7. A Gaussian laser beam from a 633 nm He-Ne laser is aligned with the help of 2 mirrors in kinematic mounts. The light is linearly polarized by a polarization beam splitter (PBS) aligned to the axis of the SLM. The beam profile is then expanded as it passes through 2 convex lenses. The intensity of the beam is reduced after passing through a beam splitter (BS). The resulting light is incident onto the screen of the SLM (Holoeye PLUTO-2-VIS-016). The wavefront at the SLM plane undergoes a Fourier Transform upon reflection by passing through a convex lens of focal length 50 mm, and this Fourier object (captured at the back-focal plane of the lens) is then magnified using a second convex lens of the same focal length. Finally, the magnified object is captured on a screen and imaged by a CMOS camera (Basler acA2000-165um).

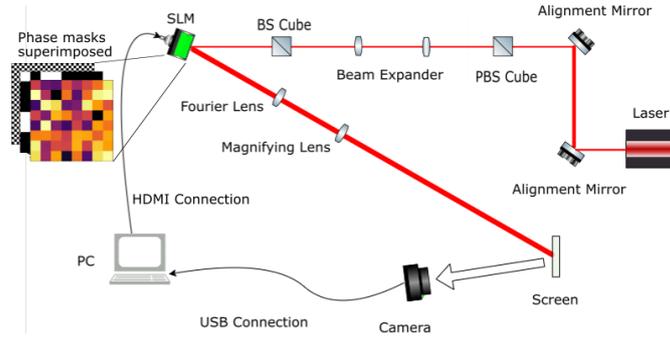


Figure 7. Schematic of the experimental setup. The laser is linearly polarized by a PBS cube and using a beam expander, we fill out the active area of the SLM. The light reflected from the SLM undergoes a Fourier Transform after passing through a convex lens and subsequent magnification by a second convex lens increases the spatial resolution of the Fourier plane. At the top left of the figure the superimposed phase masks are indicated. These represent the binary checkerboard ($c_j = \pm 1$), phase mask of spins ($s_j = \pi/2$ or $3\pi/2$), and phase mask corresponding to the normalised numbers in the set ($\alpha_j = \cos^{-1} \zeta_j$). These phase masks are superimposed as $\theta_j = s_j + c_j \alpha_j$.

Let us take the electric field at the plane of the SLM to be $\vec{E}(\vec{r})$. Assume a polarized incident laser beam such that the incoming wavefront is

$$\vec{E}_{\text{in}}(\vec{r}) = \sum_{j=0}^{N^2-1} \zeta_j \text{rect}_j(\vec{r}) \hat{x}, \quad (4.2)$$

where ζ_j gives the complex amplitude of the electric field at the j^{th} pixel of the SLM. We also assume that the active area of the SLM comprises $N \times N$ pixels of side length L , and that the laser spot is approximately a plane wave of constant amplitude E_0 over this active area and zero elsewhere. The x -component of the electric field at the SLM plane is written as

$$E_x(\vec{r}) = E_0 \sum_{j=0}^{N^2-1} \phi_j \text{rect}_j(\vec{r}), \quad (4.3)$$

where $\text{rect}_j(\vec{r})$ is the rectangular function and $\phi_j = \exp(i\theta_j)$. Here, θ_j is the phase delay imparted by the j^{th} SLM pixel to the laser beam. The field given in (4.3) undergoes a Fourier Transform to become $\tilde{E}_x(\vec{k})$ at the camera plane [26]:

$$\tilde{E}_x(\vec{k}) = E_0 \int \sum_{j=0}^{N^2-1} \phi_j \text{rect}_j(\vec{r}) \exp(i\vec{k} \cdot \vec{r}) d^2r. \quad (4.4)$$

The intensity $\tilde{I}(\vec{k})$ at the readout plane is hence given by

$$|\tilde{E}_x(\vec{k})|^2 = L^2 \text{sinc}^2 \left(\frac{\vec{k} \cdot \vec{L}}{2} \right) \sum_{m,n=0}^{N^2-1} \zeta_m \zeta_n \phi_m \phi_n \exp[i\vec{k} \cdot (\vec{r}_m - \vec{r}_n)]. \quad (4.5)$$

We now set a target intensity and define the cost function

$$\text{Cost} = \sum_{x', y'} [I(x', y') - I_{\text{Target}}(x', y')]^2, \quad (4.6)$$

where x' and y' are spatial coordinates in the camera plane. These represent the components of $\vec{k} \propto x' \hat{x} + y' \hat{y}$. The constant of proportionality is given by $\frac{1}{\lambda f}$, where λ is the wavelength of the laser and f is the focal length of the Fourier lens. Since $I_{\text{Target}}(x', y')^2 = \text{constant}$ and

$\sum_{x', y'} I(x', y')^2 \approx \text{constant}$ over iterations, we can take the Hamiltonian to be the cross product term such that

$$H = -2L^2 \int \text{sinc}^2 \left(\frac{\vec{k} \cdot \vec{L}}{2} \right) \sum_{m, n=0}^{N^2-1} \zeta_m \zeta_n \phi_m \phi_n \exp [i\vec{k} \cdot (\vec{r}_m - \vec{r}_n)] I_{\text{Target}}(k) d^2k. \quad (4.7)$$

If we consider a binary phase modulation by the SLM, i.e. $\phi_m = \pm 1$, then H represents an all-to-all coupling in the Ising model, with coupling constants

$$J_{mn} = -2L^2 \mathcal{F} \left[\tilde{I}_{\text{Target}}(k) \text{sinc}^2 \left(\frac{\vec{k} \cdot \vec{L}}{2} \right) \right] \zeta_m \zeta_n. \quad (4.8)$$

The coupling constant of two sites on the SLM depends on the chosen target intensity that we desire to settle to. To encode problems of our choice, we choose the target intensity to be a 2D delta function with peak at the central pixels of the camera, so that its effect after a Fourier transformation is constant, and we have a coupling term that depends on $\zeta_m \zeta_n$.

(b) Hardware

Using 256×256 pixels as the active area of the SLM, we group adjacent pixels as a spin. This aggregation is done to create sufficient contrast to be detected by the camera. We choose an active area of 256×256 pixels as it fills the laser spot on the SLM. Initializing a random spin distribution within the active area, we keep a constant binary checkerboard in the inactive area. Using this setup, we ran iterative algorithms to move from the random spin distribution to the target distribution. Flipping $d = 1$ spins within the active area at each iteration, we capture the resulting pattern with a CMOS camera.

The camera exposure time is set to give a maximum range for the intensity detection. The cost function, (4.6), of this pattern is then calculated with respect to the target image. At the i^{th} iteration, $\text{Cost}[i] - \text{Cost}[i-1] = \Delta E$, and a Metropolis-Hastings (M-H) algorithm [27] is run with an annealing schedule until the cost function converged to a minimum [28,29].

We see a steady decrease in the cost function as the experiment ran. As shown in Fig. 8, the cost function for the M-H algorithm decrease to within a 100^{th} of its initial value within 800 iterations for a 16×16 spin lattice.

(c) Solving the number-partitioning problem

The Mattis Hamiltonian can be exactly mapped to the objective function to be minimized to solve the NPP [1], subject to a proportionality constant. With the same experimental setup from Sec. 4, we take the electric field at the plane of the SLM to be $\vec{E}(\vec{r})$, polarized along \hat{x} . Suppose the SLM active area comprises $N \times N$ pixels of side length l . Let $M \times M$ pixels be one spin, so we have a lattice of $S \times S$ spins, with $N = MS$. As shown in Sec. 4, and using the same approximations and notations, the x -component of the electric field at the camera plane is given by

$$\tilde{E}_x(\vec{k}) = E_0 \int \sum_{j=0}^{N^2-1} \phi_j \text{rect}_j(\vec{r}) \exp(i\vec{k} \cdot \vec{r}) d^2r. \quad (4.9)$$

$\vec{k} = \vec{0}$ gives the field at the origin of the readout plane as in [30]:

$$\tilde{E}_x(\vec{0}) \propto -E_0 l^2 \sum_{j=0}^{N^2-1} \phi_j. \quad (4.10)$$

where $\phi_j = \sigma_j e^{i(-1)^j \alpha_j}$. A similar idea was used in [31]. Here σ_j gives the spin value and $\alpha_j = \cos^{-1} \zeta_j$, with ζ_j as the numbers normalized by dividing over the largest number in the set. These

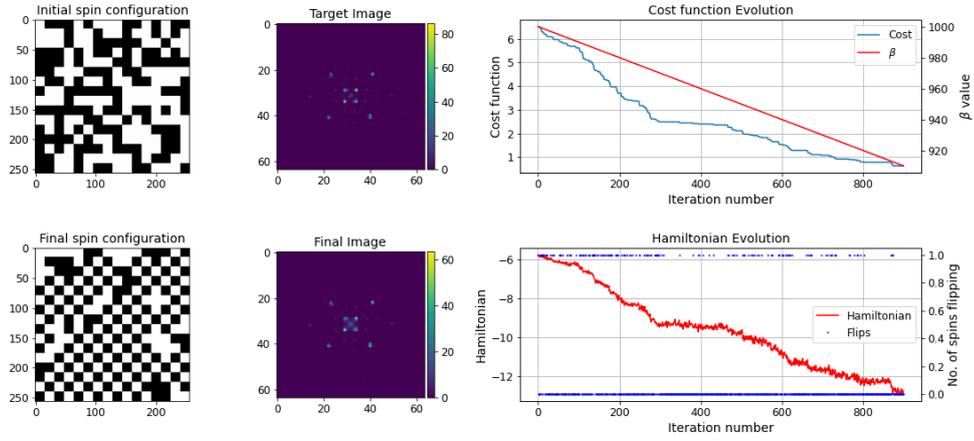


Figure 8. Ising machine results for 16×16 spins. We flip $d = 1$ spins at each iteration. The checkerboard is almost entirely reproduced, as the cost function decreases to 0.04 of its initial value within around 800 iterations. The Hamiltonian function is correlated with the cost function defined in (4.6) and similarly decreases. On the top right graph, the variable $\beta = \frac{1}{k_B T}$, where $k_B = 1.38 \times 10^{-23} \text{JK}^{-1}$ and T is the temperature of the system.

two terms are constant over an area of adjacent $M \times M$ pixels on the SLM, or within one spin. Hence, the electric field at the center of the camera plane is

$$\tilde{E}_x(\vec{0}) \propto -E_0 l^2 \sum_{a=0}^{S^2-1} \sigma_a \left[e^{i\alpha_a} + e^{-i\alpha_a} \dots M^2 \text{ terms} \right] \quad (4.11)$$

We choose M to be even, which allows us to group pairs of exponentials and obtain:

$$\tilde{E}_x(\vec{0}) \propto -E_0 l^2 M^2 \sum_{a=0}^{S^2-1} \sigma_a \cos(\alpha_a). \quad (4.12)$$

The intensity at the center of the camera plane becomes

$$\tilde{I}(\vec{0}) \propto \sum_{m=0}^{S^2-1} \sum_{n=0}^{S^2-1} \sigma_m \sigma_n \cos(\alpha_m) \cos(\alpha_n). \quad (4.13)$$

Comparing (4.1) and (4.13), we find that $\tilde{I}(\vec{0})$ maps onto H .

With the experimental setup given in Fig. 7, the laser intensity is recorded at the central 64×64 pixels on the image plane. The aim here is to minimize the total intensity captured at each iteration. A recurrent feedback loop is therefore setup between the camera and SLM through a Python program on a computer.

The Hamiltonian given by (4.1) is adiabatically changed from a problem instance of all equal numbers, to the desired problem instance, according to:

$$H(t) \propto \sum_{m=0}^{S^2-1} \sum_{n=0}^{S^2-1} \sigma_m \sigma_n \cos\left(\frac{t\theta_m}{T}\right) \cos\left(\frac{t\theta_n}{T}\right). \quad (4.14)$$

The interpretation here is that the Hamiltonian initially represents a scenario where all the numbers to be partitioned are equal when $t = 0$; hence, the checkerboard pattern for the spins is a ground state solution. These spins are rotated in phase on the SLM, which causes a change in

H to shift to represent the original Hamiltonian as we rotate the phase. For $t = T$, H is the same as in (4.13). Since the rotation here is cosine and not linear, the intermediate problem instances of the Hamiltonian are different from the target Hamiltonian. This phase imparted is to create the amplitudes that encode the number, and is superposed with the phase mask representing the spins as described in Sec. (c). The value of t is not changed continuously, but in steps determined by the precision of the SLM. At each instance where the phases are changed, few iterations are given to let the SLM settle to the changed Hamiltonian, as can be observed in Fig. 9.

The SLM active area is chosen as 256×256 pixels or 512×512 pixels, depending on the size of the problem. Within this active area, we utilize the full analogue range of the SLM (8 bits) to simultaneously perform amplitude and phase modulation to get a Mattis model Hamiltonian at the readout plane. In our implementation, the range of problem instances depends on the size of the floating point variables sent to the SLM, which is 24 bits after the decimal place. For any number $\zeta = \cos \alpha$, we have $d\zeta = -\sin \alpha d\alpha$. Since the increment limit for the floating point is $d\alpha = 10^{-8}$, and $|\sin \alpha| \leq 1$, we get the upper limit as $d\zeta \leq 10^{-8}$. Hence, we can generate problem instances where each number has 8 significant digits at best.

Taking $N \times N$ adjacent pixels as a spin, a ground state is initialized, which for our initial Hamiltonian is any configuration that is 50% phase $\pi/2$ and 50% phase $3\pi/2$. The SLM is also divided into macropixels of size 2×2 pixels each. An additional phase of $(-1)^j \cos^{-1} \zeta_m$ is applied onto the j^{th} macropixel of the SLM active area, where m is the index of the spin. Since the coupling constants are all positive in the problem instances we consider, adiabatically tuning them results in a dip in the cost function. The exposure time of the camera is chosen for each problem instance as the value required to just reach saturation of the intensity reading upon initializing a problem instance. This allows us to maximize the range of intensity values and hence the cost function change that the camera can detect, which leads to improved results.

At each stage in the adiabatic process [32], the M-H algorithm is run by flipping d spins at each iteration and deciding whether to keep the flip based on the change in intensity, where the value of d depends on the number of spins. Generally, we expect a lower value of d to provide better convergence. We find an adiabatic solution by minimizing $H \propto I(\vec{0})$ continuously as $\beta = (k_B T)^{-1}$ decreases, until a solution is reached. This way of encoding the problem onto a SPIM gives us an efficient means to get approximate solutions to the NPP.

Adiabatically tuning the Mattis coupling coefficients improves the results, and the system consistently performs well for problem sizes ranging from 16 spins to 16384 spins. A sample plot for a problem instance of size 16384 spins is shown in Fig. 9. The plot at the top right of the figure shows the cost function decreasing throughout the experiment, with sharp dips whenever the coupling constants are changed. The quality of our solution is quantified by a fidelity,

$$\eta = \left| \frac{\sum_j \zeta_j \sigma_j}{\sum_j \zeta_j} \right|, \quad (4.15)$$

where ζ_j and σ_j are the values of the j^{th} number and spin. Squaring the fidelity gives:

$$\eta^2 = \left| \frac{\sum_{i,j} \zeta_i \sigma_i \zeta_j \sigma_j}{\sum_{i,j} \zeta_i \zeta_j} \right|. \quad (4.16)$$

The denominator in the above equation remains constant through all the iterations. Therefore, the square of the fidelity is proportional to the Mattis model Hamiltonian given in (4.1) and by extension, the intensity at the center of the camera plane as well. The fidelity is thus positively correlated with the Hamiltonian throughout the energy landscape. In the plot shown on the bottom left of Fig. 9, we can see that the fidelity escapes a local minima quite easily; hence, we are able to sample a large solution space. This is partly due to the intrinsic intensity noise of the laser source, and partly due to the algorithm employed. The same plot also reveals that the number of accepted flips is a lot lower than the number of rejected flips. This is due to the Fourier transform, an all-to-all operation, which results in a spin flip causing only a tiny change to the detected intensity. A plot of the solution quality with problem size is shown in Fig. 10, where the

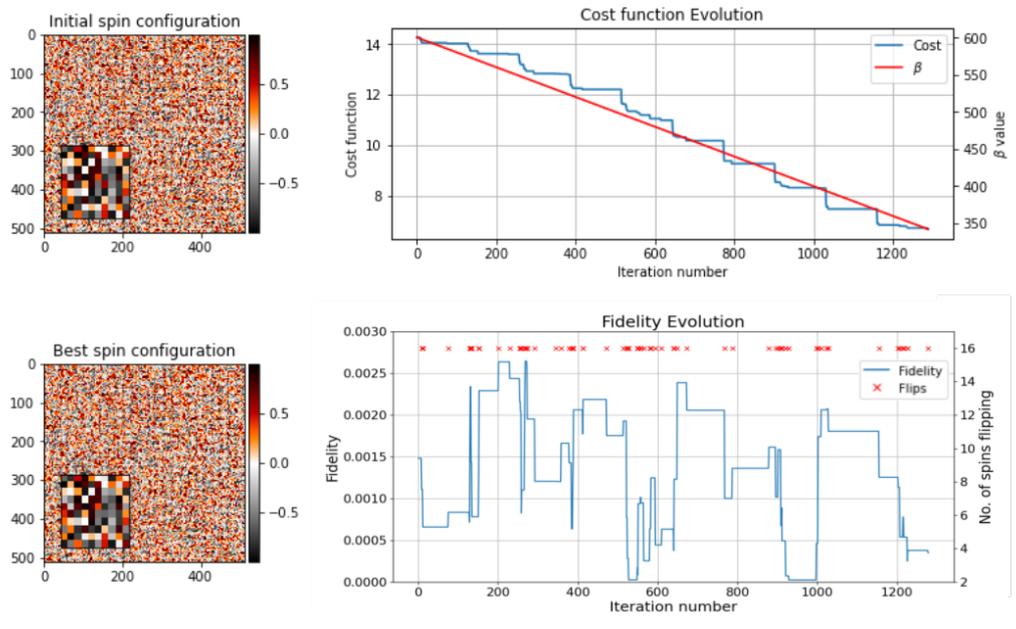


Figure 9. Results for a problem instance of size 16384 spins. The periodic sudden dips in the cost function are a signature of the adiabatic tuning of the coupling constants. As shown in the plot on the bottom right of the figure, the fidelity escapes a local minima fairly easily, allowing us to sample a large energy landscape. Further, we can see that the number of accepted flips is lower than the number of rejected flips. Insets on the colourmaps shown in the left show an expanded view of a section of 10×10 spins.

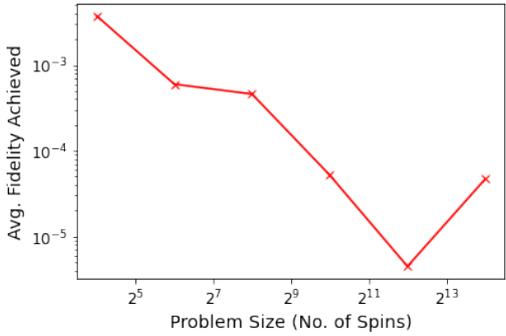


Figure 10. Scaling of the solution quality with problem size. The mostly downward trend, which shows that solution quality increases on average with problem size. The experimental scheme is hence favourable in time complexity.

solution is averaged over several randomly generated problem instances. Unlike similar work done previously, we do not use the Hamming distance as a metric for the solution quantity [33]. The NPP typically has several approximate solutions which may be degenerate, and yet of starkly different spin configurations. The Hamming distance gives us information on how many spins need to be flipped to achieve the true ground state, but from an application point of view, the aim here is to benchmark the quality and utility of the achieved solution against other solvers.

The performance of our SPIM in solving the NPP is compared with that of the D-Wave 5000+ qubit Advantage_system1.1 Annealer (DWA) [34] and the classical solver, Gurobi [14], by running multiple problem instances on both devices.

- Table 3 compares the best fidelity achieved over a single run for different problem instances of a fixed size.
- Table 4 shows average fidelity achieved and its corresponding runtime, for the maximum problem size.

Table 3. Performance of size 64 spins on different problem instances, with the best fidelity achieved in each trial.

Trial	Gurobi	DWA	SPIM
1	4.38E-05	2.89E-05	5.34E-04
2	3.58E-05	1.78E-04	1.28E-04
3	4.53E-05	2.41E-03	2.74E-04
4	1.07E-05	1.27E-05	1.47E-04
5	9.74E-05	1.16E-04	5.89E-04

Table 4. Benchmarking the SPIM with the DWA and Gurobi. *The SPIM runtime indicated is the maximum required for all problem sizes. **The DWA runtime indicated includes the embedding time for 121 spins for different problem instances.

Solver	Max. Problem Size	Avg. Fidelity for 64 spins	Runtime
SPIM	16384	6E-04	9 min.*
DWA	121	5.49E-04	~ 10 min.**
Gurobi	1024	4.66E-05	< 1 min. for 64 spins ~ 10 min. for 1024 spins

We observe from Table 3 that the SPIM achieves fidelity values of $\mathcal{O}(10^{-4})$ for different problem instances as opposed to the D-Wave device. Despite the fact that D-Wave has a 5000+ qubit system, the number of spins that can be embedded is capped at a 11×11 grid for the NPP. This is due to the large overhead in embedding a coupling between spins for a graph of density 100%. The runtime of the SPIM also scales favourably for larger problem instances when compared to other systems as we see in Table 4. For smaller problem sizes, up to 1024 spins, the performance of Gurobi is consistently better than both the D-Wave system and the SPIM and serves as a good benchmark. However, as the size of the problem gets larger, Gurobi is unable to fetch a solution, and the D-Wave annealer can't embed problem sizes greater than a 11×11 grid, of 121 spins, due to limited connectivity. For larger problem sizes, the SPIM provides us a scalable method to solve the NPP.

5. Summary

We have successfully demonstrated two photonic Ising machines with off the shelf optical components. The time multiplexed coherent Ising machine and the spatial multiplexed Ising machine were each benchmarked against the Gurobi optimizer.

The CIM gave better results than Gurobi for the Max-Cut problem with 100 nodes (spins) with a graph density less than 50%. Its performance was on par with Gurobi for graphs with up to 1000 nodes. Over an order of magnitude speed-up can be expected from further optimization of the FPGA used in the CIM.

The SPIM could tackle the number partition problem with 2^{14} spins, whereas both Gurobi and D-Wave were limited to running with 2^{10} spins. Future implementations of SPIMs will be more compact, with a digital lens programmed into the SLM [35]. They can also operate at higher speeds using fast adaptive optics technologies such as digital micromirror devices (DMDs) [36].

Data Accessibility. Data underlying the results presented in this paper are available under a Creative Commons License [37]

Authors' Contributions. This manuscript was written by A. Prabhakar and S. Tayur. P. Shah and U. Gautham built the TMIM, while V. Natrajgan and V. Ramesh built the SPIM. A. Prabhakar and N. Chandrachoodan supported and mentored the students on both hardware builds.

Funding. This work was supported in part by a research grant from KLA.

Acknowledgements. We are thankful to IIT Madras for supporting this research during the pandemic. The student authors on this paper (Gautham, Parth, Vighnesh and Vikram) were recipients of the 2020 Tayur Prize.

References

1. A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.
2. S. Utsunomiya, K. Takata, and Y. Yamamoto, "Mapping of Ising models onto injection-locked laser systems," *Opt. Express*, vol. 19, pp. 18091–18108, Sep 2011.
3. M. Babaeian, D. T. Nguyen, V. Demir, M. Akbulut, P.-A. Blanche, Y. Kaneda, S. Guha, M. A. Neifeld, and N. Peyghambarian, "A single shot coherent Ising machine based on a network of injection-locked multicore fiber lasers," *Nature Communications*, vol. 10, p. 3516, 2019.
4. P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, and Y. Yamamoto, "A fully programmable 100-spin coherent Ising machine with all-to-all connections," *Science*, vol. 354, pp. 614–617, 2016.
5. 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, pp. 937–942, 2014.
6. T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, pp. 603–606, 2016.
7. T. Inagaki, K. Inaba, R. Hamerly, K. Inoue, Y. Yamamoto, and H. Takesue, "Large-scale Ising spin network based on degenerate optical parametric oscillators," *Nature Photonics*, vol. 10, pp. 415–419, 2016.
8. J. D. Hart, D. C. Schmadel, T. E. Murphy, and R. Roy, "Experiments with arbitrary networks in time-multiplexed delay systems," *Chaos: An Interdisciplinary J. of Nonlinear Science*, vol. 27, p. 121103, 2017.
9. F. Böhm, G. Verschaffelt, and G. Van der Sande, "A poor man's coherent Ising machine based on opto-electronic feedback systems for solving optimization problems," *Nature Comm.*, vol. 10, p. 3538, 2019.
10. F. Böhm, T. Inagaki, K. Inaba, T. Honjo, K. Enbutsu, T. Umeki, R. Kasahara, and H. Takesue, "Understanding dynamics of coherent Ising machines through simulation of large-scale 2D Ising models," *Nature Comm.*, vol. 9, p. 5020, 2018.
11. C. Tradonsky, I. Gershenzon, V. Pal, R. Chriki, A. A. Friesem, O. Raz, and N. Davidson, "Rapid laser solver for the phase retrieval problem," *Science Advances*, vol. 5, p. eaax4530, 2019.
12. I. Gershenzon, G. Arwas, S. Gadasi, C. Tradonsky, A. Friesem, O. Raz, and N. Davidson, "Exact mapping between a laser network loss rate and the classical XY Hamiltonian by laser loss control," *Nanophotonics*, vol. 9, pp. 4117–4126, 2020.
13. I. Vellekoop and A. Mosk, "Phase control algorithms for focusing light through turbid media," *Optics Comm.*, vol. 281, p. 3071, 2008.
14. "Gurobi optimization, llc." <https://www.gurobi.com/>, 2021.
15. R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, E. Rieffel, M. M. Fejer, S. Utsunomiya, H. Takesue, and Y. Yamamoto, "Quantum vs. optical annealing:

- Benchmarking the opo Ising machine and d-wave," in *Conference on Lasers and Electro-Optics*, p. FTu4A.2, Optical Society of America, 2018.
16. R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, K. Enbutsu, T. Umeki, R. Kasahara, S. Utsunomiya, S. Kako, K. ichi Kawarabayashi, R. L. Byer, M. M. Fejer, H. Mabuchi, D. Englund, E. Rieffel, H. Takesue, and Y. Yamamoto, "Experimental investigation of performance differences between coherent Ising machines and a quantum annealer," *Science Advances*, vol. 5, no. 5, p. eaau0823, 2019.
 17. R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, D. R. Englund, and Y. Yamamoto, "Synchronously-pumped OPO coherent Ising machine: Benchmarking and prospects," in *AI and Optical Data Sciences* (B. Jalali and K. ichi Kitayama, eds.), vol. 11299, pp. 42 – 48, International Society for Optics and Photonics, SPIE, 2020.
 18. Y. Haribara, H. Ishikawa, S. Utsunomiya, K. Aihara, and Y. Yamamoto, "Performance evaluation of coherent Ising machines against classical neural networks," *Quantum Science and Technology*, vol. 2, p. 044002, aug 2017.
 19. R. M. Karp, *Reducibility among Combinatorial Problems*, pp. 85–103. Boston, MA: Springer US, 1972.
 20. G. Rinaldi, "Rudy, a software for graph generation."
 21. F. Böhm, T. V. Vaerenbergh, G. Verschaffelt, and G. Van der Sande, "Order-of-magnitude differences in computational performance of analog Ising machines induced by the choice of nonlinearity," *Communications Physics*, vol. 4, Jul 2021.
 22. E. Boros and P. L. Hammer, "The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds," *Annals of Operations Research*, vol. 33, pp. 151–180, Mar 1991.
 23. C. M. Li, F. Manyá, and J. Planes, "Detecting disjoint inconsistent subformulas for computing lower bounds for Max-SAT," in *AAAI*, vol. 6, pp. 86–91, 2006.
 24. D. Mattis, "Solvable spin systems with random interactions," *Physics Letters A*, vol. 56, pp. 421–422, 1976.
 25. H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, 2001.
 26. J. Goodman, *Introduction to Fourier Optics*. Electrical Engineering Series, McGraw-Hill, 1996.
 27. N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, vol. 44, pp. 335–341, 1949.
 28. R. Pathria and P. D. Beale, *Statistical Mechanics*, ch. 16. Boston: Academic Press, 3 ed., 2011.
 29. K. Binder and D. Heermann, *Monte Carlo Simulation in Statistical Physics: An Introduction*. Graduate Texts in Physics, Berlin Heidelberg: Springer-Verlag, 5 ed., 2010.
 30. O. Mendoza-Yero, G. Mínguez-Vega, and J. Lancis, "Encoding complex fields by using a phase-only optical element," *Optics Letters*, vol. 39, pp. 1740–1743, 2014.
 31. Y. Fang, J. Huang, and Z. Ruan, "Experimental observation of phase transitions in spatial photonic Ising machine," 2021. arXiv:2011.02771.
 32. D. Pierangeli, G. Marcucci, and C. Conti, "Adiabatic evolution on a spatial-photonic Ising machine," *Optica*, vol. 7, p. 1535, 2020.
 33. D. Pierangeli, G. Marcucci, D. Brunner, and C. Conti, "Noise-enhanced spatial-photonic Ising machine," *Nanophotonics*, vol. 9, p. 4109, 2020.
 34. "D-Wave systems." <https://www.dwavesys.com/>.
 35. L. Zhao, N. Bai, X. Li, L. S. Ong, Z. P. Fang, and A. K. Asundi, "Efficient implementation of a spatial light modulator as a diffractive optical microlens array in a digital Shack-Hartmann wavefront sensor," *Appl. Opt.*, vol. 45, pp. 90–94, 2006.
 36. S. A. Goorden, J. Bertolotti, and A. P. Mosk, "Superpixel-based spatial amplitude and phase modulation using a digital micromirror device," *Opt. Express*, vol. 22, pp. 17999–18009, 2014.
 37. <https://github.com/vighneshn/SLM-code>.